Universidade Federal de Minas Gerais

# Detecção de clusters espaciais através de otimização multiobjetivo

André Luiz Fernandes Cançado

Tese submetida à banca examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Minas Gerais como parte dos requisitos para a obtenção do título de Doutor em Engenharia Elétrica.

**Orientador:** Luiz Henrique Duczmal

**Co-Orientador:** Ricardo Hiroshi Caldeira Takahashi

*“ A meus pais, Lenira e Murilo. ”*

# Agradecimentos

Ao professor Luiz, pela orientação e incentivo. Por ter acreditado em mim e por ter estado sempre disponível e disposto a me ajudar. Por ter me ensinado a ter uma visão sempre voltada para os aspectos relevantes da ciência e a sempre buscar a pergunta correta.

Ao professor Ricardo, por ter abraçado esse trabalho e por tantas contribuições.

Ao professor Carlos, pelo acolhimento na Universidade do Algarve e apoio durante meu estágio de doutorado em Faro. Pela dedicação ao trabalho e ao nosso grupo da UFMG. Pelas inúmeras críticas, sugestões e contribuições.

Aos meus irmãos, Cláudia, Ricardo e Juliana, pelo apoio incondicional e pelos momentos de descontração.

Aos colegas do GOPAC, pela convivência.

Ao grupo de otimização, pelas críticas e sugestões durante nossas reuniões semanais, em especial aos colegas Beth, Carrano, Rodrigo e Gladston, e aos professores Oriane e Serjão.

Aos funcionários do PPGEE, em especial à Anete e à Arlete, sempre dispostas a quebrar qualquer galho burocrático.

Aos professores do PPGEE.

Aos colegas do Departamento de Estatística, em especial ao Anderson e ao Caicó, parceiros e amigos.

Ao professor Sabino, pelos papos sobre trabalho e sobre música, ambos essenciais durante o doutorado.

À CAPES por ter-me concedido uma bolsa de doutorado no Brasil e uma bolsa de estágio de doutorado no exterior.

À Iana, pelo companheirismo, deidicação e paciência.

# Resumo

*Clusters* espaciais irregulares ocorrem com frequência em estudos epidemiológicos, mas seu delineamento geográfico é mal definido. Os métodos atuais de detecção encontram somente uma dentre as várias soluções possíveis, com formas diferentes, da mais compacta até a mais irregular, correspondentes ao variados graus de penalização impostos à liberdade de forma. E mesmo quando um conjunto completo de soluções está disponível, a escolha do parâmetro mais adequado é deixada a cargo do analista, cuja decisão é subjetiva. Propomos um critério quantitativo para a escolha da melhor solução através de otimização multiobjetivo, encontrando o conjunto Pareto-ótimo. Dois objetivos conflitantes estão envolvidos na busca: regularidade da forma e avaliação da estatística *scan*. Ao invés de executar sequencialmente um algoritmo de detecção de *clusters* variando o grau de penalização, todas as soluções são encontradas em paralelo, através de um algoritmo genético multiobjetivo. O método é rápido e apresenta bom poder de detecção. A introdução do conceito de conjunto de Pareto nesse problema, seguido da escolha da solução mais significativa, permite que a escolha da melhor solução seja rigorosa, mas sem a necessidade de nenhum parâmetro arbitrário. O conceito de significância do *cluster* é estendido de maneira natural através do uso da função de aproveitamento, sendo empregado como critério de decisão para escolha da melhor solução. Os modelos de Gumbel e Weibull são utilizados para aproximar a distribuição empírica da estatística *scan*, aumentando a velocidade de estimação da significância. Essa metodologia é comparada ao algoritmo genético mono-objetivo. Uma aplicação na detecção de *cluster* de câncer de mama é discutida. Por fim, o problema de detecção de *clusters* é relaxado e modelado como um problema *knapsack*, permitindo que se obtenha uma cota superior, em contraste com a cota inferior obtida pelo algoritmo genético.

**Palavras-chave:** Otimização multiobjetivo, conjunto de Pareto, algoritmo genético, estatística espacial *scan*, *cluster* espacial, compacidade, penalização geométrica, distribuição de Gumbel, distribuição de Weibull.

# Abstract

Irregularly shaped spatial disease clusters occur commonly in epidemiological studies, but their geographic delineation is poorly defined. Most current spatial scan software usually displays only one of the many possible cluster solutions with different shapes, from the most compact round cluster to the most irregularly shaped one, corresponding to varying degrees of penalization parameters imposed to the freedom of shape. Even when a fairly complete set of solutions is available, the choice of the most appropriate parameter setting is left to the practitioner, whose decision is often subjective. We propose quantitative criteria for choosing the best cluster solution, through multi-objective optimization, by finding the Pareto-set in the solution space. Two competing objectives are involved in the search: regularity of shape, and scan statistic value. Instead of running sequentially a cluster finding algorithm with varying degrees of penalization, all solutions are found in parallel, employing a genetic algorithm. The method is fast, with good power of detection. The introduction of the concept of Pareto-set in this problem, followed by the choice of the most significant solution, is shown to allow a rigorous statement about what is a "best solution", without the need of any arbitrary parameter. The cluster significance concept is extended for this set in a natural way through the use of the attainment function, being employed as a decision criterion for choosing the optimal solution. The Gumbel and Weibull models are used to approximate the empirical scan statistic distribution, speeding up the significance estimation. The multi-objective methodology is compared with the single-objective genetic algorithm. An application to breast cancer cluster detection is discussed. Finally, a knapsack approach is proposed for a relaxed version of the problem, allowing an upper bound to be obtained, in contrast with the lower bounds obtained by the genetic algorithm.

**Keywords:** Multi-objective optimization, Pareto set, genetic algorithm, spatial scan statistic, spatial disease cluster, geometric compactness penalty correction, Gumbel distribution, Weibull distribution.

# Sumário

# Lista de Figuras

# Lista de Tabelas

# Capítulo 1.

# Introdução

Um *cluster*[1] espacial é uma parte de um mapa em que a ocorrência de casos de um fenômeno de interesse é discrepante do restante do mapa, isto é, alta demais ou baixa demais. Esse fenômeno é, muitas vezes, a infecção por alguma doença ou a ocorrência de algum crime. Daí a importância de se ter métodos eficientes de detecção de *clusters* espaciais nas áreas de epidemiologia, criminalidade e até em vigilância anti-terrorismo. Epidemiologia e vigilância sindrômica fazem uso intensivo de técnicas para detecção e inferência de *clusters* espaciais. O delineamento de *clusters* é uma ferramenta importante em estudos etiológicos (Lawson *et al.*, 1999), na detecção precoce de manifestações de doenças (Duczmal & Buckeridge, 2005, 2006; Kulldorff *et al.*, 2005, 2006, 2007) e na identificação de fatores ambientais relacionados à doença (Patil *et al.*, 2006). A estatística espacial *scan* (Kulldorff, 1997) disponível nos *softwares* SaTScan™ e ClusterSeer® é atualmente usada em vários departamentos de saúde para detecção de *clusters* circulares (Kulldorff & Nagarwalla, 1995). Em muitos cenários, no entanto, estamos interessados em detectar *clusters* que não estão necessariamente restritos à forma circular. As doenças podem estar concentradas ao longo de um rio, da costa do mar ou de um lago, ou ainda ao longo de rodovias ou de regiões poluídas. A idéia do SaTScan foi estendida para a detecção de *clusters* com forma elíptica (Kulldorff *et al.*, 2006), aumentando a versatilidade geométrica do SaTScan original e recentemente outros métodos foram propostos para a detecção de *clusters* com forma irregular (Duczmal & Assunção, 2004; Duczmal *et al.*, 2006; Iyengar, 2004; Tango & Takahashi, 2005; Assunção *et al.*, 2006; Neill *et al.*, 2005b; Patil & Taillie, 2004). Em Conley *et al.* (2005) foi apresentado um algoritmo

[1]Embora exista em português o termo *conglomerado*, optamos pelo termo em inglês por este já estar incorporado ao vocabulário científico.

genético baseado em dados pontuais para explorar a configuração espacial de aglomerados múltiplos de elipses. Em Sahajpal *et al.* (2004) também foi utilizado um algoritmo genético para detecção de *clusters* irregulares como interseções de círculos com raios e centros distintos. Em Duczmal *et al.* (2007) é apresentado um algoritmo genético para detecção de *clusters* irregulares em um mapa dividido em um certo número de regiões, maximizando a estatística *scan* com a utilização de uma penalização (Duczmal *et al.*, 2006) para as soluções altamente irregulares.

O delineamento geográfico de *clusters* irregulares apresenta algumas dificuldades. A liberdade geométrica ilimitada para a forma do *cluster* diminui o poder de detecção (Duczmal *et al.*, 2006). Isto acontece porque o conjunto de todas as soluções conexas, independente de forma, é muito grande. O máximo da função objetivo tende a estar associado a um *cluster* em forma de árvore, que simplesmente liga as regiões do mapa com maior verossimilhança, sem contribuir para a descoberta de soluções que fazem o delineamento correto do *cluster* verdadeiro. Em outras palavras, há uma grande quantidade de "ruído" sobre o qual o "sinal" da solução verdadeira não se sobressai. Este é um problema que ocorre em todos os métodos de detecção de *clusters* irregulares e pode ser contornado, em parte, limitando o número máximo de regiões que podem constituir cada solução. Outra solução, mais elegante, consiste em aplicar uma penalização usando o conceito de compacidade (Duczmal *et al.*, 2006, 2007), penalizando a avaliação da estatística *scan* de acordo com a irregularidade da forma da solução e generalizando uma idéia que foi utilizada no caso das elipses (Kulldorff *et al.*, 2006).

Variando a intensidade da penalização quanto à liberdade de forma, várias soluções-candidatas podem ser encontradas, da circular até a mais irregular. Os algoritmos atuais de detecção de *cluster* não permitem o controle da geometria e geralmente apenas uma solução é obtida. Mesmo quando um conjunto de soluções está disponível, executando o algoritmo várias vezes e alterando os parâmetros, como em Duczmal *et al.* (2006, 2007), a escolha da configuração de parâmetros mais adequada é deixada a cargo do analista, cuja decisão é, em geral, subjetiva.

## 1.1. Objetivos

O foco principal deste trabalho é apresentar um novo método para detecção e inferência de *clusters* espaciais, baseado em algoritmos genéticos multi-objetivo. Dois objetivos

estão envolvidos na busca pelo *cluster* verdadeiro: (i) valor da estatística *scan* e (ii) regularidade da forma. Propomos um critério quantitativo para escolher a melhor solução, encontrando o conjunto Pareto-ótimo no espaço de soluções, seguido de um critério de decisão que consiste em maximizar a significância sobre este conjunto. Dessa forma a escolha arbitrária e subjetiva da melhor solução é deixada de lado e substituída por uma metodologia teoricamente fundamentada para encontrar tal solução. O conceito de melhor solução passa a ser bem definido no contexto de detecção de *clusters* espaciais. Como subproduto dessa metodologia, um conjunto de soluções alternativas (o conjunto Pareto-ótimo) se torna disponível para o analista para efeito de comparação e análise da estrutura intrínseca do problema. Essas idéias são novas no contexto de detecção de *clusters* espaciais, apresentando similaridades com outros problemas de aprendizagem com estrutura multiobjetivo, como em Teixeira *et al.* (2000) e Nepomuceno *et al.* (2003).

Ao invés de executar a busca pela solução várias vezes, variando o grau de penalização, o algoritmo multiobjetivo proposto encontra um conjunto de soluções em paralelo. Como algoritmos genéticos trabalham com populações inteiras de soluções-candidatas, essa busca por várias soluções em uma única execução torna-se natural para essa classe de algoritmos e é o que faz com que algoritmos genéticos sejam particularmente eficientes na resolução de problemas multiobjetivo (Fonseca & Fleming, 1995). Além disso, algoritmos genéticos permitem que se consiga escapar de soluções que sejam ótimos locais, o que os torna ótimas ferramentas para a detecção de *clusters* (Duczmal *et al.*, 2007). Usando os conjuntos Pareto-ótimos, o conceito de significância dos clusters é estendido de maneira natural e sem a necessidade de uma escolha arbitrária do parâmetro de penalização.

## 1.2. Estrutura do texto

Esta tese está organizada em capítulos. No capítulo 2 introduzimos a estatística de teste na qual é baseada a busca de *clusters* - a estatística espacial *scan* - e apresentamos uma breve revisão dos métodos de detecção de *clusters* espaciais. Essa revisão abrange o método *Scan* Circular clássico, bem como métodos de detecção de *clusters* com geometria arbitrária. Iremos ainda dar uma motivação para o uso de uma penalização que é baseada na geometria dos *clusters*.

No capítulo 3 descrevemos uma estrutura genérica para algoritmos genéticos em geral. Em seguida o algoritmo genético utilizado para detecção de *clusters* espaciais é descrito detalhadamente em termos de seus operadores. Apresentamos uma motivação para abordar o problema de detecção de *clusters* espaciais como um problema de otimização bi-objetivo. Faremos uma introdução aos conceitos essenciais de otimização multi-objetivo e, em seguida, descrevemos as modificações aplicadas ao algoritmo genético para que obtivéssemos uma algoritmo capaz de atacar o problema bi-objetivo proposto.

No capítulo 4 fazemos uma discussão sobre técnicas de inferência usadas para se estimar o quão significativos são os *clusters* detectados e essas técnicas são estendidas para o caso bi-objetivo. Verificamos ainda a qualidade de ajuste de dois modelos paramétricos que podem nos auxiliar nessa estimativa. Por fim, o comportamento do algoritmo genético, em suas versões mono e bi-objetivo, é avaliado em termos de poder, sensibilidade e valor preditivo positivo.

No capítulo 5 aplicamos o algoritmo genético e as técnicas de inferência descritos nos capítulos anteriores a dados reais utilizados em trabalhos anteriores, de maneira que podemos comparar o desempenho dos métodos desenvolvidos nessa tese com resultados da literatura.

No capítulo 6 abordamos o problema tratado nessa tese sob um outro ponto de vista. A estrutura do problema é relaxada e mostramos que, assim, o problema pode ser reduzido a um problema clássico de otimização combinatória: o problema da mochila. Obtendo soluções exatas para o problema assim formulado, teremos condições de contrastá-las com as soluções obtidas pelo algoritmo genético, obtendo intervalos dentro dos quais garantidamente se encontram as soluções verdadeiras.

No capítulo 7 apresentamos as considerações finais desta tese e as propostas de continuidade de trabalho. São relacionadas ainda as publicações decorrentes do trabalho desenvolvido durante o doutorado.

# Capítulo 2.

# Detecção de clusters

Em muitas aplicações, como em epidemiologia, vigilância sindrômica e criminologia, é importante levar em conta a população em questão. Ao invés de encontrar regiões com grande número de casos, uma análise deveria encontrar regiões com número de casos maior do que o esperado. Nessa linha o trabalho Besag & Newell (1991) utilizava um método que localizava uma janela circular em cada região envolvida. O raio dessa janela era então expandido para incluir regiões vizinhas até que um número crítico de casos definido pelo analista se localizasse dentro da janela. Então a população dentro dessa janela era comparada àquela esperada sobre a frequência de casos. No entanto, levar em conta apenas a razão entre número de casos observados e a população (ou o número de casos esperado) pode levar ao problema de encontrar *clusters* que não têm nenhuma significância do ponto de vista estatístico.

Para exemplificar essa idéia, considere duas cidades $A$ e $B$ com populações de risco $N_A = 100$ e $N_B = 1.000.000$, respectivamente, inseridas em um mapa em estudo. Considere a população de risco total do mapa $N = 10.000.000$ e o número total de casos observados $C = 100.000$. Isto quer dizer que, caso não haja *cluster* no mapa, a freqüência de casos esperada deve ser de 1 caso para cada 100 habitantes em todas as regiões do mapa. Logo, o número de casos esperado na cidade $A$ deve ser $\mu_A = 1$ e na cidade $B$, $\mu_B = 10.000$. Suponha que os casos observado nas cidades $A$ e $B$ sejam, respectivamente, $c_A = 2$ e $c_B = 20.000$. Nesse caso, ambas as cidades apresentam risco relativo (número observado de casos dividido pelo número esperado de casos) $c_A/\mu_A = c_B/\mu_B = 2$. No caso da cidade $A$, a probabilidade de que o risco relativo em dobro tenha ocorrido por mero acaso é muito alta. Já na cidade $B$ há um grande motivo para haver preocupação. A chance de o número de casos passar de 10.000 para 20.000 não pode ser

encarada como uma simples flutuação estatística, e um estudo detalhado deve ser levado em consideração.

Na próxima seção será apresentada uma estatística capaz de distinguir regiões como as do exemplo anterior, a estatística espacial *Scan* . Nas seções seguintes apresentaremos alguns métodos que se utilizam dessa estatística como medida na detecção de *clusters* espaciais.

## 2.1. Estatística Espacial Scan de Kulldorff

A estatística espacial *scan* proposta em Kulldorff & Nagarwalla (1995) e em Kulldorff (1997) supera o problema de considerar simplesmente o risco relativo de uma maneira muito simples. Pelo exemplo anterior é fácil perceber que um aumento no risco relativo é tão mais significativo quanto maior é a população na região em estudo.

Vamos considerar um mapa dividido em $n$ regiões $R_1, ..., R_n$, cada uma delas com uma população $N_i$ e um número observado de casos $c_i$. Chamamos de *zona* qualquer subconjunto conexo de regiões do mapa. A Figura 2.1 ilustra algumas zonas distintas dentro do mesmo mapa. Denotaremos por $Z$ o conjunto de todas as zonas do mapa. Nesta tese de doutorado iremos adotar o modelo de Poisson[1] para a distribuição de casos no mapa. Isto quer dizer que o número de casos $C_i$ dentro da região $R_i$ é uma variável aleatória com distribuição de Poisson, cuja função de probabilidade é dada por

$$f_i(c) = \begin{cases} \dfrac{e^{-\lambda_i}{\lambda_i}^c}{c!} & \text{se } c \geq 0 \\ \\ 0 & \text{caso contrário.} \end{cases} \tag{2.1}$$

isto é, a probabilidade de que a variável aleatória $C_i$ assuma o valor $c$ é dada por $f_i(c)$. O parâmetro $\lambda_i$ é a média ou valor esperado da variável. A distribuição de Poisson é adequada para descrever o número de ocorrências de um evento em um determinado intervalo de tempo ou em uma determinada região. Assim, assumimos que o número de

[1]Há ainda a possibilidade de se adotar o modelo multinomial. No entanto, os dois modelos são assintoticamente equivalentes.

casos $C_i$ dentro da região $R_i$ segue uma distribuição de Poisson com média proporcional à sua população $N_i$, ou seja, $\lambda_i = p_i N_i$, onde $p_i$ é a probabilidade de que um indivíduo na região $R_i$ seja um caso. Denota-se a distribuição de Poisson por $C_i \sim Po(p_i N_i)$.

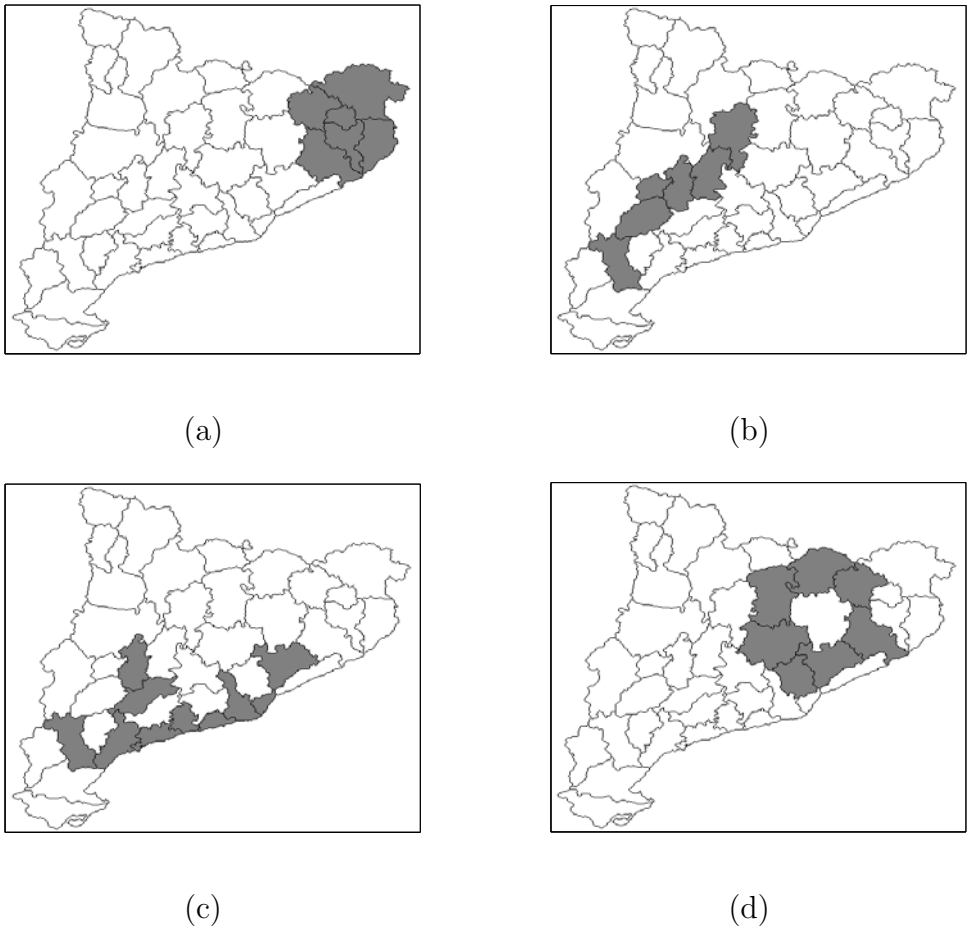



**Figura 2.1.:** Quatro diferentes zonas dentro de um mapa.

Sabe-se que a soma de variáveis aleatórias independentes com distribuição de Poisson é ainda uma variável aleatória com distribuição de Poisson cujo parâmetro é a soma dos parâmetros das distribuições das variáveis somadas. Vamos assumir, a princípio, que a probabilidade de que um indivíduo seja um caso seja a mesma em todas as regiões, isto é, $p_i = p$, $i = 1, ..., n$. Nessa situação o número de casos $C_z$ em uma zona $z$ será uma variável aleatória com distribuição de Poisson com parâmetro $pN_z$, onde $N_z$ é a população da zona $z$, ou seja, $C_z \sim Po(pN_z)$, $\forall z \in Z$. Note que, de acordo com essa suposição, não há *clusters* no mapa, uma vez que a probabilidade de um indivíduo vir a ser um caso é igual em qualquer parte do mapa. Essa é a nossa *hipótese nula* $h_0$.

A hipótese alternativa $h_a$ é de que exista uma zona $z^* \in Z$ que é um *cluster*. Nesse caso teríamos $C_{z^*} \sim Po(pN_{z^*})$ e $C_z \sim Po(qN_z)$, $\forall z \neq z^*$, com $p > q$. De maneira que estamos interessados no teste que confronta as hipóteses de $z^*$ ser ou não um *cluster*, ou seja

$$\begin{cases} h_0 : p = q \\ h_a : p > q \end{cases} \tag{2.2}$$

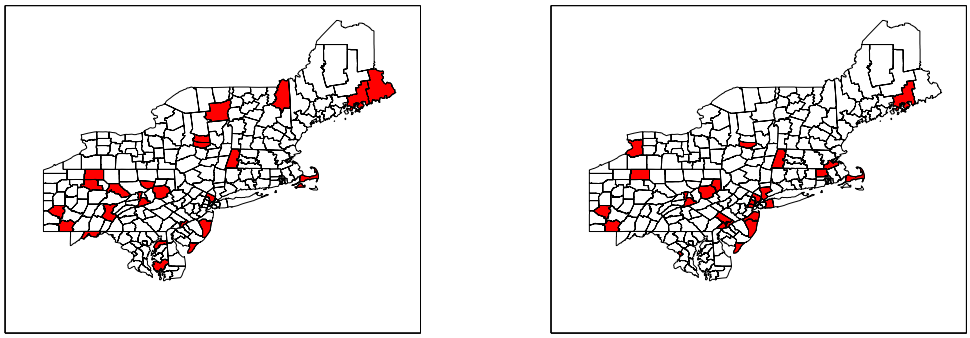
Sejam $N$ a população total do mapa e $C$ o número total de casos do mapa. Considere ainda $c_z$ como o número observado e $\mu_z$ como o número esperado de casos dentro de uma zona $z$. Definindo $L(z)$ como a função de verossimilhança sob a hipótese alternativa de que exista uma zona $z^*$ que é um *cluster*, e $L_0$ como a verossimilhança sob a hipótese nula de que não exista um *cluster*, foi mostrado em Kulldorff (1997) que a razão de verossimilhança (*likelihood ratio*) $LR = L(z)/L_0$ para o modelo de Poisson pode ser escrita como:

$$LR(z) = \begin{cases} \left(\frac{c_z}{\mu_z}\right)^{c_z} \left(\frac{C-c_z}{C-\mu_z}\right)^{C-c_z} & \text{se } c_z > \mu_z \\ 1 & \text{caso contrário.} \end{cases} \tag{2.3}$$

Esta razão maximizada sobre todas as zonas identifica o *cluster* $z^*$ mais verossímil. Daí temos a estatística de teste, dada por $T = \max_z LR(z)$.

A função $LR$ escrita como na equação (2.3) nos permite uma interpretação bastante intuitiva quanto ao seu significado. Observe que o risco relativo em uma zona $z$ é dado por $I(z) = c_z/\mu_z$, e o risco relativo fora dessa zona é dado por $O(z) = (C - c_z)/(C - \mu_z)$. Dessa forma, podemos escrever a função $LR$ como

$$LR(z) = I(z)^{c_z} O(z)^{C-c_z} \tag{2.4}$$



(a) (b)

**Figura 2.2.:** (a) As 10% maiores incidências de câncer de mama no mapa do nordeste dos Estados Unidos e (b) as 10% maiores verossimilhanças.

e considerá-la como uma função binomial, onde os pesos são dados pelos casos dentro e fora da zona $z$. Em geral é mais conveniente trabalhar com o logaritmo da razão de verossimilhança, $LLR$ (*logarithm of likelihood ratio*), já que a função $LR$ cresce muito rapidamente. Como o logaritmo é uma função estritamente crescente, se $z^*$ maximiza $LR(z)$, então $z^*$ maximiza $LLR(z)$. Note que no caso das duas cidades $A$ e $B$ do exemplo anterior, embora ambas apresentem o mesmo risco relativo, as respectivas avaliações de $LLR$ seriam:

$$LLR(A) = 2 \log\left(\frac{2}{1}\right) + 99.998 \log\left(\frac{99.998}{99.999}\right) \approx 3,8 \times 10^{-1}$$

$$LLR(B) = 20.000 \log\left(\frac{20.000}{10.000}\right) + 80.000 \log\left(\frac{80.000}{90.000}\right) \approx 4,4 \times 10^{3}$$

Maiores detalhes na derivação da estatística *scan* podem ser obtidos em Kulldorff (1997). A Figura 2.2(a) mostra as regiões de maior incidência de casos de câncer de mama em um mapa do nordeste dos Estados Unidos. Já a Figura 2.2(b) mostra as regiões de maior razão de verossimilhança (Duczmal *et al.*, 2006). Em ambas as figuras foram escolhidas as 10% maiores (de um total de 245 regiões). Observe que as regiões de maior incidência e maior $LLR$ coincidem apenas algumas vezes.

## 2.2. Métodos de detecção

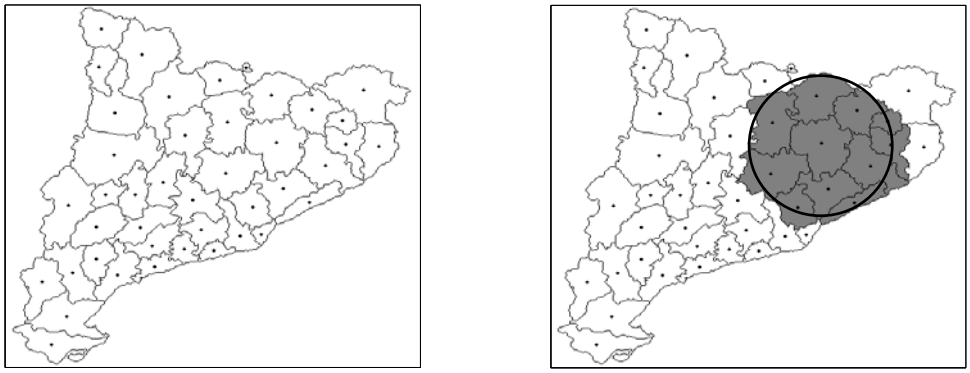
De posse de uma estatística que permita avaliar cada zona, nos resta encontrar aquela que apresenta avaliação máxima. Porém, a maior dificuldade da estimação de *clusters* reside exatamente na maximização da estatística $LLR(z)$ sobre o conjunto $Z$ de todas as zonas possíveis. Isto porque, embora seja finito, o conjunto $Z$ é em geral tão grande que torna a maximização de $LLR(z)$ impraticável através de uma busca exaustiva. Para contornar esse problema, existem basicamente duas técnicas:

- Redução do espaço de parâmetros $Z$ em outro espaço $Z'$, onde $Z' \subset Z$. O conjunto $Z'$ deve ser escolhido de modo que seu tamanho permita uma busca exaustiva. Esta técnica funciona bem se o conjunto $Z'$ contém a zona $z^*$ que maximiza $LLR(z)$, ou pelo menos uma boa aproximação para $z^*$.
- Utilização de métodos estocásticos de otimização. Ainda que esses métodos não analisem todo o espaço de busca eles podem, sob certas condições, convergir para o ótimo global.

Podemos ainda classificar os métodos de detecção quanto à geometria dos *clusters* encontrados:

- *Clusters* regulares são aqueles que têm uma forma pré-determinada, em geral circular. Os métodos analisam apenas *clusters* que tenham essa forma. Note que esses métodos, por definição, se utilizam da técnica de redução do espaço de parâmetros, transformando-o em um espaço que só contém *clusters*-candidatos com um formato específico.
- *Clusters* irregulares, que podem apresentar formas arbitrárias. Na classe de métodos que buscam *clusters* irregulares existem tanto aqueles que se utilizam da redução do espaço de busca quanto aqueles que se utilizam de regras heurísticas estocásticas.

Os primeiros métodos de detecção de *clusters* se utilizavam da técnica de redução do espaço de busca, já que essa é uma técnica mais imediata, menos elaborada e que requer menor esforço computacional. Os métodos estocásticos são, em geral, mais sofisticados e se utilizam de regras e heurísticas mais complexas, além de fazerem uso de computação mais intensiva. Na próxima seção faremos uma descrição do principal método de detecção



(a) (b)

**Figura 2.3.:** (a) Um mapa dividido em regiões e seus respectivos centróides e (b) uma zona obtida por uma janela circular.

utilizando a redução do espaço de parâmetros, o método *Scan* Circular. Em seguida faremos uma breve revisão dos principais métodos de detecção de *clusters* irregulares.

### 2.2.1. O método Scan Circular

O método *scan* circular proposto em Kulldorff (1997) pertence à primeira classe de técnicas, restringindo o espaço de busca apenas às zonas que têm formato circular. Para isso o método utiliza janelas circulares que varrem o mapa em busca da zona $z^*$. Para cada região do mapa definimos um centróide, que é um ponto arbitrário em seu interior. Assim, uma janela circular sobre o mapa em estudo define uma zona que é constituída pelas regiões cujos centróides se encontram dentro da janela. A Figura 2.3(a) mostra um mapa dividido em regiões e seus respectivos centróides. A janela circular ilustrada na Figura 2.3(b) determina a zona formada pelas regiões escuras.

Considere $d_{ij}$ a distância entre os centróides $c_i$ e $c_j$ (das regiões $R_i$ e $R_j$, respectivamente). O método *scan* circular escolhe as janelas da seguinte forma: selecione uma região $R_k$, $1 \leq k \leq n$. Ordene as demais $n-1$ regiões do mapa quanto à distância ao centróide $c_k$, em ordem crescente, obtendo a seqüência de regiões $\{R_{l_1}, R_{l_2}, ..., R_{l_{n-1}}\}$, onde $d_{kl_1} \leq d_{kl_2} \leq ... \leq d_{kl_{n-1}}$. As janelas são escolhidas como sendo círculos cujos centros coincidem com o centróide $c_k$ e com raios iguais a $d_{kl_1}$, $d_{kl_2}$, ..., $d_{kl_s}$, onde $s$ é tal que

$d_{kl_s} \leq r_{max} < d_{kl_{s+1}}$, sendo $r_{max}$ o raio máximo permitido. Cada janela gera uma zona e o processo é repetido para $k = 1, ..., n$.

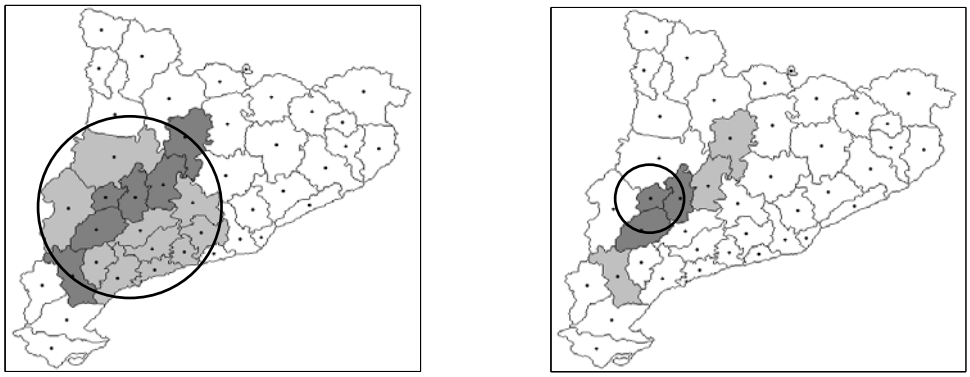
Para cada janela avaliamos a zona correspondente através da estatística *scan*. O *cluster* mais verossímil é aquele que maximiza $LLR(z)$. Note que a quantidade de raios utilizados é da mesma ordem de $n$. De fato, se $r_{max}$ é maior do que a maior distância entre centróides do mapa, o número de raios utilizado é $n$. Caso contrário, esse número é menor que $n$. Assim, no método *scan* circular temos que avaliar no máximo $n^2$ zonas distintas, o que é computacionalmente simples.

O método *scan* circular é hoje amplamente utilizado na detecção de *clusters* espaciais e, embora a idéia seja bastante simples, o método é eficiente e extremamente rápido. No entanto, o método falha quando o *cluster* verdadeiro apresenta uma forma que não seja circular. Imagine que o *cluster* verdadeiro apresente uma forma alongada, como a zona da Figura 2.1(b). O método circular não tem como encontrar essa solução e a solução por ele apresentada superestima ou subestima o *cluster* verdadeiro. No primeiro caso a solução do *scan* circular contém a solução verdadeira, no sentido de que todas as regiões que compõem o *cluster* verdadeiro estão também na solução encontrada. Porém, várias outras regiões também são incluídas na solução, simplesmente porque não existe um círculo que cubra a solução verdadeira sem que isso aconteça (Figura 2.4(a)). Por outro lado, o método pode incluir em sua solução apenas regiões que estão no *cluster* verdadeiro, mas deixar de fora outras regiões que também deveriam estar (Figura 2.4(b)).

Assim, embora em muitos casos o *cluster* verdadeiro possa apresentar um formato circular, estamos interessados em métodos que nos permitam encontrar soluções com outras formas. A próxima seção apresenta alguns métodos utilizados na detecção de *clusters* de geometria arbitrária.

### 2.2.2. Detecção de clusters irregulares

A extensão imediata para o método *scan* circular é a utilização de janelas de formato elíptico (Kulldorff *et al.*, 2006). A idéia desse método é análoga à do *scan* circular. Porém, ao invés de variarmos apenas o tamanho da janela, para cada centróide, podemos variar também sua orientação e sua excentricidade. Isso faz com que aumentemos o horizonte de soluções possíveis, permitindo que sejam detectados *clusters* com formas alongadas, por exemplo. Ainda assim, são muitos os casos em que o *cluster* verdadeiro



(a) (b)

**Figura 2.4.:** (a) A solução encontrada pelo *Scan* Circular superestima o *cluster* verdadeiro, incluindo regiões que não lhe pertencem. (b) A solução subestima o *cluster* verdadeiro, deixando de incluir regiões que pertencem ao *cluster* verdadeiro.

apresenta um formato que não se encaixa em nenhuma elipse. É o caso, por exemplo, de um cluster que acompanhasse um rio em forma de "L", ou um *cluster* que tenha um "buraco" como o da Figura 2.1(d) (ver página 7). Na tentativa de solucionar esse problema começam a surgir métodos que permitem a detecção de *clusters* com formato irregular.

Ainda na linha de redução do espaço de busca, os trabalhos de Iyengar (2004) e Tango & Takahashi (2005) propõem uma flexibilização do *scan* circular, utilizando janelas com várias geometrias diferentes, além da circular e da elíptica. No entanto, por mais que se flexibilize a geometria das janelas utilizadas, sempre é possível que o *cluster* verdadeiro não se encaixe em nenhum formato pré-determinado. Há ainda o trabalho de Patil & Taillie (2004) que utiliza a idéia de *upper level set* (conjunto de nível superior) que reduz o espaço $Z$ considerando apenas as zonas cujos riscos relativos estão acima de um determinado nível. No entanto, nenhuma discussão é feita sobre a escolha do parâmetro *nível* e resultados e comparações com outros métodos não são apresentados.

Como mencionado anteriormente, a maior dificuldade de se procurar *clusters* com formato qualquer é que isto se torna uma tarefa computacionalmente muito complexa. Em um mapa com $n$ regiões, se quiséssemos verificar todas as possibilidades, teríamos que descobrir quais dos $2^n$ subconjuntos de regiões são conexos e avaliá-los. Essa ordem de complexidade se torna proibitiva, mesmo para um mapa com poucas regiões.

Daí surgem as primeiras tentativas de se partir para métodos heurísticos estocásticos, que se aproximam de uma boa solução mas não garantem que a melhor solução será encontrada. Nessa linha o trabalho Duczmal & Assunção (2004) propõe um algoritmo *simulated annealing* que faz uma busca estocástica, sem limitar a geometria das soluções-candidatas analisadas e tentando se aproximar do que seria o *cluster* verdadeiro. Esse método faz uma busca aleatória em momentos em que o valor de $LLR$ é baixo e, à medida que a verossimilhança vai aumentando, aumenta também a chance de o algoritmo fazer uma busca gulosa. Essas incursões aleatórias são essenciais para que o método não fique preso em algum ótimo local. O maior problema desse algoritmo é que na maioria das vezes ele superestima o *cluster* verdadeiro. Isto é, o *cluster* verdadeiro está incluído na solução apresentada, mas várias outras regiões que não lhe pertencem também são incluídas na solução. Isto se deve justamente ao fato de o método permitir que a solução tenha uma forma qualquer, apenas exigindo que ela seja conexa, e fazendo com que a solução mais verossímil encontrada pelo *simulated annealing* seja simplesmente uma coleção de regiões de alta verossimilhança que se espalha em forma de árvore por todo o mapa. A Figura 2.5 mostra o *cluster* encontrado pelo *simulated annealing* no mapa da Nova Inglaterra, EUA, utilizando dados de câncer de mama. Obviamente não estamos interessados em soluções dessa natureza, uma vez que isso não nos acrescenta nenhuma informação geográfica a respeito da ocorrência do fenômeno em estudo.

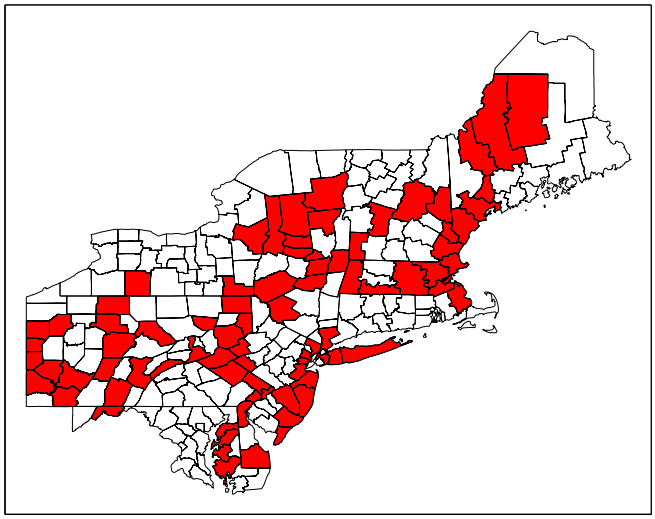


**Figura 2.5.:** *Cluster* encontrado pelo *simulated annealing* sem penalização.

Um outro método, proposto em Assunção *et al.* (2006), utiliza o conceito de árvore geradora para tentar estimar o *cluster* verdadeiro. Este método utiliza árvores geradoras que são cortadas em partes gerando vários candidatos a *cluster*. A vantagem desse método é que encontrar árvores geradoras do grafo que representa todo o mapa é relativamente barato, além de as partes da árvore gerarem soluções automaticamente conexas. Porém, este método apresenta o mesmo problema do *simulated annealing* de superestimação do *cluster* verdadeiro, encontrando *clusters* que se espalham por sobre todo o mapa.

Há ainda os métodos que se utilizam de dados pontuais, ao invés de dados distribuídos em regiões, caracterizando um outro tipo de abordagem. Nesse contexto também existem vários métodos, inclusive com a utilização de algoritmos genéticos. Podemos destacar os trabalhos Openshaw & Perrée (1996), Sahajpal *et al.* (2004) e Conley *et al.* (2005). Em todos eles foram apresentadas propostas de algoritmos genéticos onde cada indivíduo é uma janela circular ou elíptica e, portanto, esses algoritmos trabalham com populações de círculos ou elipses (ou aglomerações desses objetos). Como veremos, isso faz com que esses algoritmos genéticos, além de utilizarem dados pontuais, tenham estruturas e concepções completamente diferentes do algoritmo aqui utilizado.

Nesta tese de doutorado o método de detecção utilizado foi um algoritmo genético multiobjetivo baseado no algoritmo genético proposto em Duczmal *et al.* (2007). Esse algoritmo será descrito em detalhes no capítulo 3, em suas versões mono e multiobjetivo, respectivamente.

## 2.3. Penalização geométrica

Para contornar o problema de superestimação da solução foi proposto por Duczmal *et al.* (2006) a utilização de uma penalização geométrica que privilegia o *cluster* cuja forma se aproxima da forma circular e penaliza aquele cuja forma é muito irregular. Essa penalização é baseada no conceito de *compacidade*. Existem várias formas de se medir a compacidade geográfica (Selkirk, 1982). No artigo de Duczmal *et al.* (2006) a área do *cluster* era comparada à área do círculo cujo perímetro coincidisse com o perímetro do fecho convexo do *cluster*. O fecho convexo foi utilizado por duas razões principais. Muitas vezes os dados de contornos do mapa não estavam disponíveis, inviabilizando o cálculo do perímetro da região. Assim, era necessário obter-se uma estimativa do

perímetro. Essa estimativa pode ser feita baseada no fecho convexo, como a técnica descrita por Duczmal *et al.* (2006), ou por outros meios (diagrama de Voronoi, por exemplo). O segundo motivo é que o uso do perímetro real depende da resolução dos dados de contorno. O contorno de regiões pode ter uma natureza fractal, o que faz com que o perímetro verdadeiro seja grande demais. Uma maneira de contornar o problema da explosão fractal do perímetro é utilizar uma resolução suficientemente baixa, de forma que a região se torne um polígono cujo perímetro seja razoável. Nesta tese utilizamos a aproximação por fecho convexo e o perímetro "real" dado por uma certa resolução. Utilizamos, então, a seguinte definição de compacidade:

**Definição 1 (Compacidade)** *A compacidade $K(z)$ de uma zona $z$ é definida como*

$$K(z) = \frac{4\pi A(z)}{H(z)^2} \tag{2.5}$$

*onde $A(z)$ é a área e $H(z)$ é o perímetro da zona $z$.*

A expressão (2.5) pode ser reescrita como

$$K(z) = \frac{A(z)}{\pi\left(\frac{H(z)}{2\pi}\right)^2} \tag{2.6}$$

e, assim, interpretada como a área de $z$ dividida pela área do círculo cujo perímetro coincide com o perímetro de $z$. Note que a compacidade de uma zona depende de sua forma, mas não de seu tamanho. O objeto que apresenta a maior compacidade é o círculo, cuja compacidade é 1. A compacidade de um quadrado é $\pi/4$ e a de um retângulo $a \times 1$ é $\pi a/(1+a)^2$, de forma que quanto mais arredondada é a forma de um objeto, mais próxima de 1 estará sua compacidade. Por outro lado, quanto mais irregular a forma, mais próxima de 0 será a compacidade.

A penalização geométrica consiste então em substituir a avaliação $LR(z)$ por $LR(z)^{K(z)}$ ou, equivalentemente, $LLR(z)$ por $K(z) \cdot LLR(z)$. Isso faz com que zonas cuja compacidade seja próxima de 1 tenham a seu valor de $LLR$ pouco afetado, ams aquelas cujas

compacidades sejam muito pequenas terão seu valor bastante diminuído. Pode-se ainda dar maior ou menor importância à correção exercida pela compacidade sobre o valor de $LLR(z)$ utilizando a penalização na forma $K(z)^a \cdot LLR(z)$, com $a \geq 0$. Se $a \to 0$ então $K(z)^a \cdot LLR(z) \to LLR(z)$ e não se tem penalização. À medida que o valor de $a$ aumenta, aumenta-se a força da penalização sobre formas que diferem da circular. Em particular, quando $a \to \infty$, apenas formas circulares são permitidas.

Em Duczmal *et al.* (2006) o perímetro foi substituído pelo perímetro do fecho convexo da zona e verificou-se através de experimentos numéricos que algoritmos de detecção de *clusters* irregulares se beneficiam do emprego da penalização geométrica. Essa correção age como um filtro e restringe a presença de *clusters* em forma de árvore com valor de $LLR$ extremamente alto, permitindo a detecção de *clusters* com valores de $LLR$ um pouco menores, mas com significado geográfico real. Esses últimos são, em geral, menos irregulares que aqueles em forma de árvore.

Podemos considerar que a penalização geométrica é uma extensão para métodos de detecção de *clusters* irregulares se interpretarmos que o *scan* circular também aplica uma penalização que é intrínseca ao método. Sob esse ponto de vista, a penalização exercida sobre os *clusters* não-circulares no método *scan* circular é altíssima. De fato, é como se no *scan* circular a função $LLR$ fosse multiplicada por 1 caso o *cluster* seja circular, e por zero caso contrário. Nesse sentido, o que a penalização geométrica faz é relaxar a penalização aplicada pelo *scan* circular aos *clusters* irregulares.

# Capítulo 3.

# Algoritmo Genético para detecção de clusters

Neste capítulo iremos descrever o método de detecção empregado nesta tese de doutorado. O método consiste de um algoritmo genético (AG) desenvolvido especificamente para o problema de detecção de *clusters*. Os operadores desse AG foram desenvolvidos especificamente para esse problema e foram propostos em Duczmal *et al.* (2007). Nesta tese foi feita uma extensão multi-objetivo para esse AG e sua estrutura foi alterada de forma a se assemelhar à estrutura de um dos algoritmos mais utilizados atualmente: o NSGA-II (Deb *et al.*, 2002).

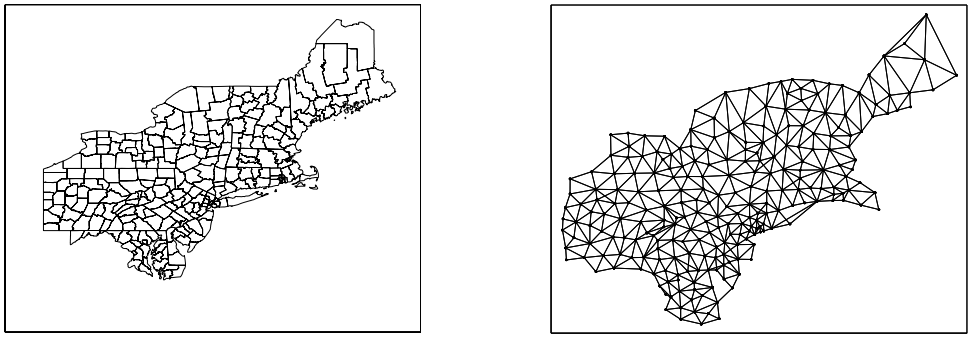
## 3.1. Aspectos estruturais

Uma forma simples de representar o mapa em estudo é através de um grafo.

**Definição 2 (Grafo)** *Um grafo $G$ é um par $G = (V, A)$, onde $V$=$\{v_1, v_2, ..., v_n\}$ é o conjunto de seus vértices e $A$ é o conjunto de todas as arestas $a_{i,j}$, onde $v_i$ e $v_j$ são adjacentes, com $v_i, v_j \in V$.*

Associamos um vértice $v_k$, $k = 1, ..., n$, a cada um dos $n$ centróides e, portanto, cada vértice está associado a uma região. Se duas regiões $i$ e $j$ têm uma fronteira em

comum[1], então os vértices $v_i$ e $v_j$ correspondentes são adjacentes e, portanto, ligados por uma aresta $a_{i,j}$. A Figura 3.1 mostra um exemplo de mapa e seu respectivo grafo associado. A representação do mapa através de um grafo apresenta algumas vantagens sobre outros tipos de estruturas. Conceitos de caminhos e conexidade estão bem definidos para estruturas de grafos. Além disso são conhecidos vários algoritmos de manipulação e busca eficientes sobre essas estruturas.



(a) (b)

**Figura 3.1.:** (a) Um mapa dividido em regiões e (b) o grafo associado.

Uma característica fundamental de toda solução-candidata é que ela deve ser conexa. Para entender o que é um grafo conexo, precisamos do conceito de *caminho*.

**Definição 3 (Caminho)** *Dois vértices $v_i$ e $v_j$ estão conectados por um caminho se existe uma sequência de p vértices $v_{l_1}, v_{l_2}, ..., v_{l_p}$ tal que $v_i = v_{l_1}$, $v_j = v_{l_p}$ e as arestas $a_{l_k,l_{k+1}} \in A$, $k = 1, ..., p-1$.*

Intuitivamente, existe um caminho entre dois vértices se é possivel partir de um deles e chegar ao outro passando somente pelas arestas existentes no grafo (e pelos vértices intermediários). Um grafo é conexo se qualquer par de vértices distintos $v_i$ e $v_j$ está conectado por um caminho. Assumimos que o mapa em estudo gera um grafo conexo, isto é, para duas regiões quaisquer $R_i$ e $R_j$ é sempre possível ir de $R_i$ a $R_j$ passando pelas fronteiras das regiões do mapa.

[1]Por fronteira em comum entende-se que haja alguma ligação entre as regiões. Uma ilha, por exemplo, terá uma fronteira em comum com uma região continental caso haja uma ponte, um túnel ou uma linha hidroviária entre elas.

Dado um conjunto $V_1 \subset V$ diremos que o grafo $G_1 = (V_1, A_1)$ é um subgrafo de $G = (V, A)$ induzido por $V_1$ se $A_1 \subset A$ é o conjunto de todas as arestas de $A$ com ambas as extremidades em $V_1$. Logo, o mapa todo é um grafo conexo e a cada zona corresponde um subgrafo conexo desse grafo, induzido[2] pelas regiões correspondentes. Os subgrafos $G_1 = (V_1, A_1)$ e $G_2 = (V_2, A_2)$ de $G$ são vizinhos se o conjunto $(V_1 \cup V_2) - (V_1 \cap V_2)$ possui exatamente um elemento. Por simplificação, usaremos a interseção $G_1 \cap G_2$ para designar $V_1 \cap V_2$. A Figura 3.2 mostra exemplos de zonas vizinhas.

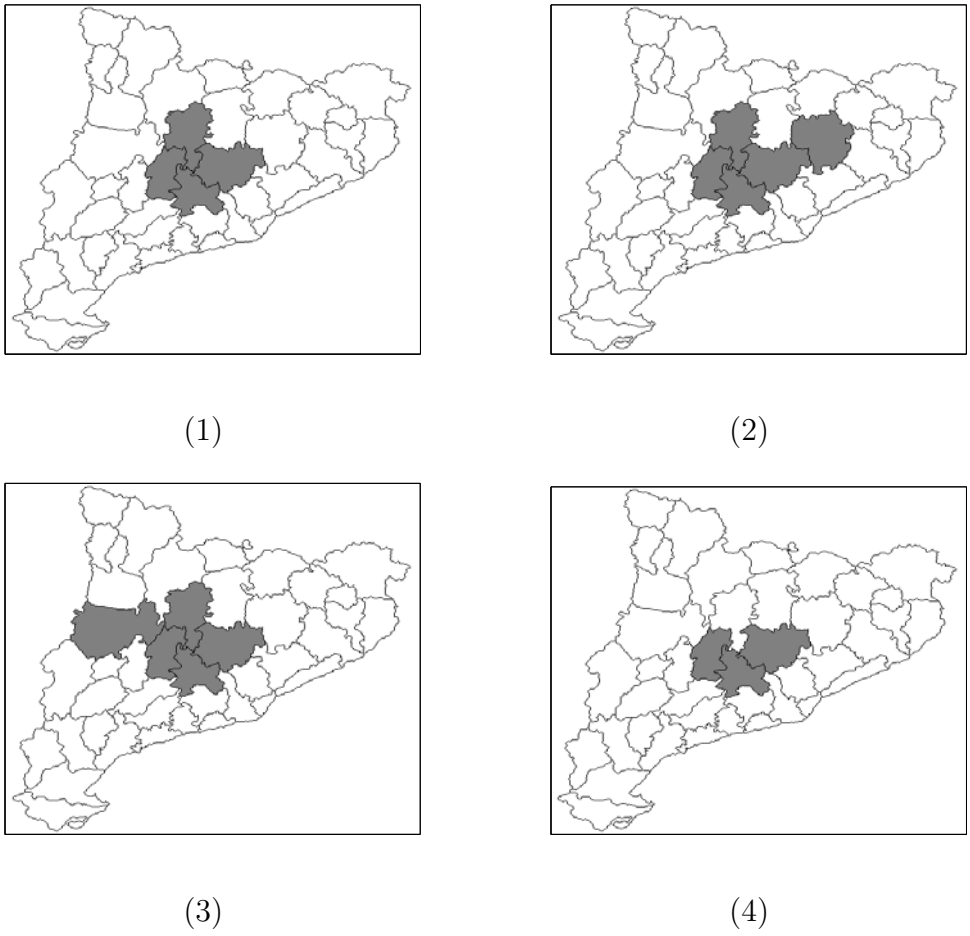



**Figura 3.2.:** As zonas 2, 3 e 4 são vizinhas da zona 1, mas não são vizinhas umas das outras.

[2]A definição de subgrafo induzido é mais forte que a de subgrafo. Em um subgrafo $G_1 = (V_1, A_1)$ não necessariamente todas as arestas de $A$ com ambas as extremidades em vértices de $V_1$ precisam estar em $A_1$. No entanto, como estamos interessados apenas em subgrafos induzidos, abandonamos essa caracterização e nos referimos às soluções apenas por subgrafos, muito embora todas as soluções sejam, na verdade, subgrafos induzidos.

## 3.2. O Algoritmo Genético

A evolução natural dos seres vivos pode ser considerada um processo de otimização. De fato, se indivíduos que são mais bem adaptados sobrevivem, ao passo que indivíduos menos adaptados tendem a desaparecer, espera-se que, após algumas gerações, a população seja composta por indivíduos que são, em geral, melhores que os das primeiras gerações. É essa mesma idéia que está por trás de um algoritmo genético. Ele tenta simular os mecanismos de variação aleatória e de seleção adaptativa da evolução natural. Os mecanismos (ou operadores genéticos) que constituem a base de um algoritmo genético são:

1. Um operador de *cruzamento*, que gera novos indivíduos a partir da combinação da informação contida em dois ou mais indivíduos;

2. Um operador de *mutação*, que utiliza a informação contida em um indivíduo para, estocasticamente, gerar outro indivíduo;

3. Um operador de *seleção*, que decide se um indivíduo terá a oportunidade de gerar descendentes para a próxima geração, baseado em sua aptidão.

Os operadores de cruzamento e mutação têm o objetivo de fazer uma "busca local". No entanto, o primeiro faz uma busca entre dois ou mais indivíduos ao passo que o segundo faz uma busca na vizinhança de um único indivíduo. Já o operador de seleção dá uma "direção" à busca. Muitas vezes esses operadores carregam algum componente estocástico, fazendo com que execuções consecutivas atinjam soluções diferentes.

Partindo de uma população inicial, constituída de soluções-tentativas, os algoritmos genéticos vão formando uma sequência de gerações. A cada iteração os operadores genéticos são aplicados à população corrente, e uma nova população é obtida. Essa estrutura faz com que os algoritmos genéticos sejam bastante robustos, no sentido de que não há necessidade de se fazer nenhuma suposição de diferenciabilidade, continuidade, convexidade ou unimodalidade da função a ser otimizada. Além disso, a função pode ser definida em espaços contínuos ou discretos (como no caso do estudo apresentado nessa tese). A única suposição que se espera ser válida a respeito da função objetivo a ser otimizada é que ela apresente uma tendência global em seu comportamento, e o desafio é fazer com que o algoritmo consiga captar essa tendência e "aprender" onde deve procurar as soluções.

Há um grande número de algoritmos genéticos conhecidos e o número de algoritmos possíveis pode ser bastante grande, já que cada operador genético pode ser implementado de várias formas diferentes bem como dispostos em estruturas diferentes. No entanto alguns algoritmos podem ser bem mais eficientes que outros sob o ponto de vista computacional (Takahashi *et al.*, 2003). Em particular, para problemas de natureza discreta sabe-se que o emprego de operadores de cruzamento e mutação específicos pode ser bem mais eficiente do que operadores genéricos que não levam em conta a estrutura específica do problema. O algoritmo genético proposto nesta tese de doutorado foi desenvolvido com operadores específicos, que exploram a estrutura do problema de se encontrar o *cluster* mais verossímil, como veremos adiante.

### 3.2.1. Geração da população inicial

É importante que a população inicial seja capaz de captar as informações do mapa como um todo. Não há razão para iniciarmos o algoritmo com os indivíduos concentrados em apenas uma parte do mapa, mesmo porque um *cluster* só pode ser identificado se possuir valor de $LLR$ discrepante das demais zonas, o que nos obriga a ter um mínimo de conhecimento sobre zonas espalhadas pelo mapa. Por esse motivo a população inicial deve ser constituída por subgrafos que estejam distribuídos de forma bastante homogênea dentro do mapa.

Assim, uma forma de gerar os indivíduos da população inicial é, a partir de cada vértice $v_i$ do grafo que representa todo o mapa, gerar um subgrafo conexo $G_i$. Aqui, usamos a idéia de um algoritmo guloso para gerar esses $G_i$'s. Considere o grafo $G_{i_0}$ formado apenas pelo vértice $v_i$. Escolha dentre os grafos vizinhos de $G_{i_0}$ o grafo $G_{i_1}$ cuja zona $z_1$ correspondente possua maior valor de $LLR$. Depois, escolha o vizinho $G_{i_2}$ de $G_{i_1}$ cuja zona $z_2$ correspondente possua maior valor de $LLR$, e assim sucessivamente, até encontrar o grafo $G_{i_n}$=$G_i$ cuja zona $\hat{z}$ correspondente possui valor de $LLR$ maior que todos os seus vizinhos, ou que tenha um número máximo de vértices pré-estabelecido. A cada passo, avaliamos todos os vizinhos do indivíduo atual (isto é, cada subgrafo que é formado pelos vértices do indivíduo atual, exceto um deles, e cada subgrafo formado pelos vértices do indivíduo atual mais alguma região vizinha). Na Figura 3.3 é possível ver a formação de um indivíduo a partir de uma única região inicial.

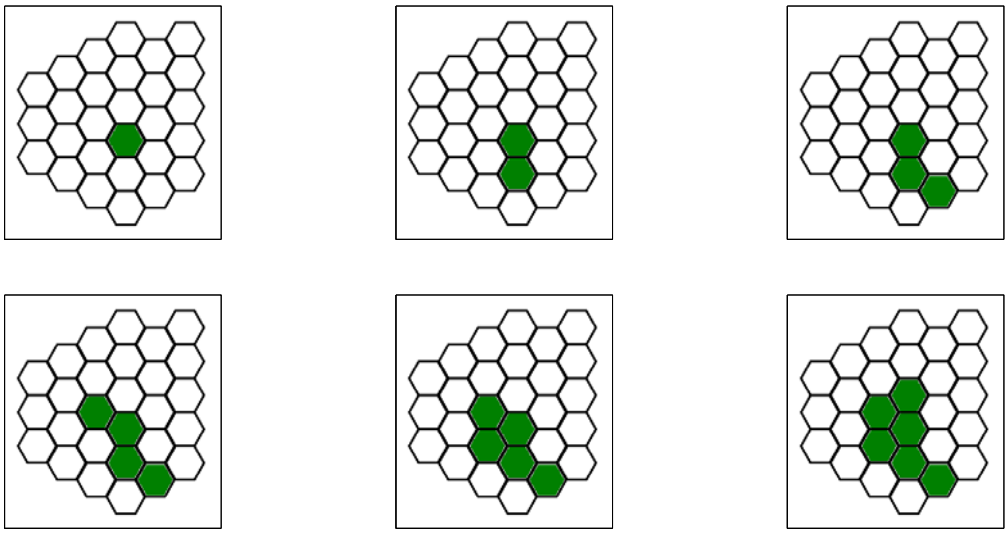


**Figura 3.3.:** Geração de um indivíduo via algoritmo guloso.

Repetindo esse procedimento a partir de cada um dos $N$ vértices teremos, ao final, uma população de $N$ zonas, cada uma obtida a partir de um vértice através dessa estratégia gulosa. É importante notar que este procedimento por si só, apesar de ser uma estratégia de otimização, em geral não encontra a solução ótima. Os indivíduos obtidos por algoritmos gulosos geralmente encontram soluções locais pois não levam em conta todo o espaço onde a função a ser otimizada está definida. Eventualmente, alguma dessas soluções locais pode coincidir com a solução global, mas não há garantia de que isso vá acontecer.

### 3.2.2. O operador de cruzamento

Como foi dito anteriormente, o objetivo do cruzamento é gerar novos indivíduos, denominados filhos, a partir da combinação das características de outros elementos, tipicamente dois, denominados pais. Como os filhos reúnem caracteristicas de ambos os pais, é natural imaginar que ele se encontra em algum ponto do "caminho" que os une. Alguns estarão eventualmente mais próximos de um dos pais do que de outro, mas espera-se que cada filho carregue consigo pelo menos uma pequena quantidade de características de cada um dos pais. Em problemas de variáveis contínuas é comum, por exemplo, a geração de filhos que estão no segmento de reta (o caminho mais curto, considerando a distância

Euclideana) que liga os dois pais. Num contexto de variáveis discretas, porém, o conceito de caminho entre soluções não está, na maioria das vezes, definido implicitamente ou intuitivamente, pela ausência da noção de vizinhança. Muitas vezes é necessário que se defina uma métrica adequada à natureza do problema, para que se possa trabalhar com o conceito de vizinhança. A partir daí é que será possível definir um caminho partindo de um pai, saltando de um indivíduo para um de seus vizinhos, e assim sucessivamente, até que se alcance o outro pai. Nesse sentido, a noção de vizinhança descrita na seção 3.1 será aplicada. O objetivo do nosso operador de cruzamento é, então, obter uma sequência de indivíduos que se encontram no caminho entre dois subgrafos pais. Para isso seguimos o procedimento descrito a seguir.

Dados dois subgrafos $A$ e $B$, tais que $A \cap B \neq \emptyset$, chamados pais, sejam $C = A \cap B$ e $D$ o maior subgrafo conexo cujos vértices estão em $C$, ou seja, $D$ é o maior subconjunto conexo dos vértices que formam o conjunto $C$. Atribuiremos um nível para cada vértice do pai $A$. Cada um dos $n_d$ vértices de $D$ (que também são vértices de $A$) recebe o nível zero. Escolhemos aleatoriamente um vértice $v_1$ adjacente a qualquer vértice de $A_0 = D$, com $v_1 \in A - A_0$, e a ele associamos o nível um. Depois, escolhemos aleatoriamente um vértice $v_2$ adjacente a qualquer vértice de $A_1 = D \cup \{v_1\}$, com $v_2 \in A - A_1$, e a ele associamos o nível 2. No $i$-ésimo passo, escolhemos aleatoriamente um vértice $v_i$ adjacente a qualquer vértice de $A_{i-1} = D \cup \{v_1, v_2, ..., v_{i-1}\}$, com $v_i \in A - A_{i-1}$. Repetimos esse passo até que todos os $n_a$ vértices de $A-D$ tenham sido escolhidos e tenham recebido seus respectivos níveis (veja o exemplo de atribuição de níveis na Figura 3.4, no meio). Note que a escolha dos níveis não é única.

Os $n_a$ vértices do pai $A$ mais o nó virtual $r$ (formado pela fusão dos vértices no conjunto $D$), juntamente com os segmentos orientados $(v_j, v_k)$, onde $v_k$ foi escolhido como adjacente a $v_j$ no $k$-ésimo passo $(j < k)$, mais os segmentos orientados $(r, v_k)$, onde $v_k$ é adjacente ao conjunto $D$, formam a árvore $T_A$ (veja Figura 3.5) que tem a seguinte propriedade:

**Lema 1** *Para cada vértice $v_i \in (A - D)$ existe um caminho do nó $r$ até o vértice $v_i$ que consiste apenas de vértices pertencentes ao conjunto $\{v_1, ..., v_{i-1}\}$.*

Demonstração: Siga o caminho orientado na árvore $T_A$, de $r$ até $v_i$.■

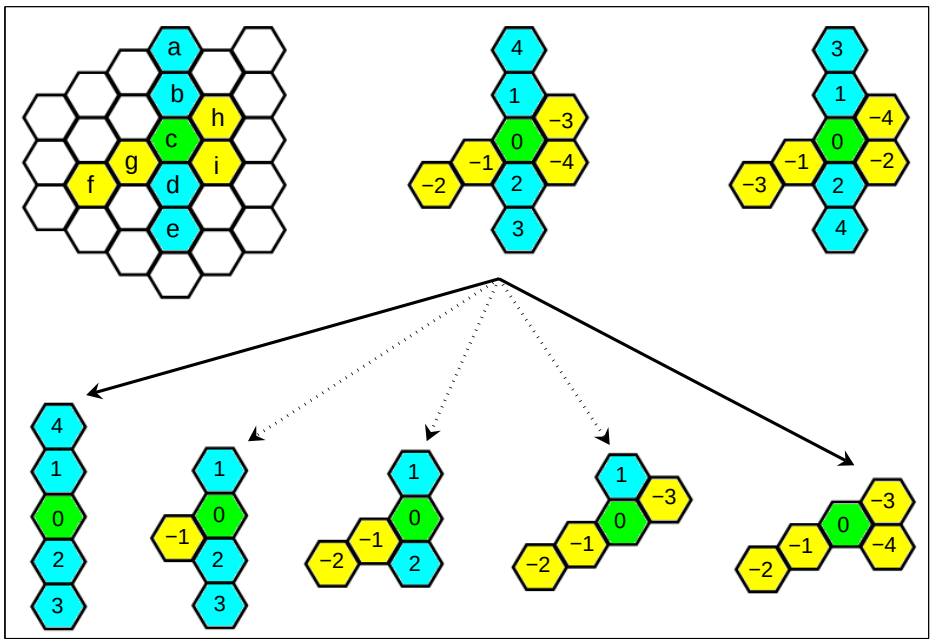



**Figura 3.4.:** Os pais $\{a, b, c, d, e\}$ e $\{c, f, g, h, i\}$ dentro do mapa (acima, à esquerda) têm a região $c$ em comum. A numeração dos níveis exemplificada (no meio, acima) gera os filhos $\{b, c, d, e, g\}$, $\{b, c, d, f, g\}$ e $\{b, c, f, g, h\}$ (apontados com setas pontilhadas). $\{a, b, c, d, e\}$ e $\{c, f, g, h, i\}$ (apontados com setas sólidas) são idênticos a seus pais, e, portanto, não são filhos. Outra numeração (dentre as várias possíveis) é exemplificada acima, à direita.

O processo descrito para determinação dos níveis dos vértices do pai $A$ é feito também para os $n_b$ vértices de $B - D$, porém usando níveis negativos ao invés de positivos. Se $C - D \neq \varnothing$ então os vértices $y \in C - D$ estão associados a dois níveis: um positivo e um negativo (ver Figura 3.6).

A partir daí contruímos os filhos de $A$ e $B$. Os níveis dos vértices do pai $A$ são $\{0,\ 1,\ 2,\ 3, ...,\ n_a\}$ e do pai $B$ $\{0,\ -1,\ -2,\ -3, ...,\ -n_b\}$. Suponha, sem perda de generalidade, que $n_a \geq n_b$. Então cada filho de $A$ e $B$ é formado pelos vértices associados aos níveis de cada uma das seguintes sequências, formadas a partir do pai $A$ e em cada passo, retirando o vértice de nível mais afastado de zero do pai $A$ (ou seja, o mais positivo) e adicionando o vértice de nível mais próximo de zero do pai $B$ (ou seja, o menos negativo):

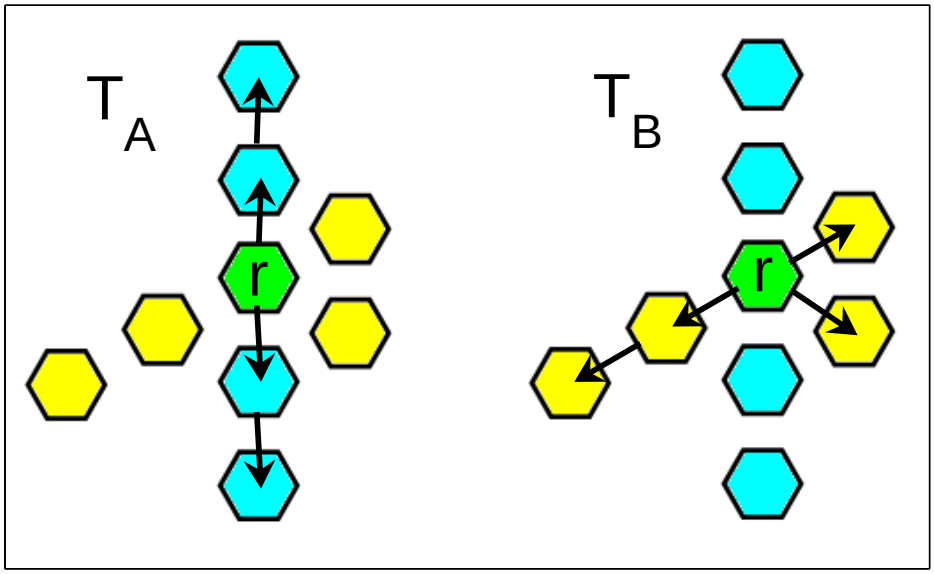



**Figura 3.5.:** Árvores $T_A$ e $T_B$.

$$
\begin{gathered}
\{n_a - 1,\ ...,\ 1,\ 0,\ -1\} \\
\{n_a - 2,\ ...,\ 1,\ 0,\ -1,\ -2\} \\
\vdots \\
\{n_a - n_b,\ ...,\ 1,\ 0,\ -1,\ -2,\ ...,\ -n_b\} \\
\{n_a - n_b - 1,\ ...,\ 1,\ 0,\ -1,\ -2, ...,\ -n_b\} \\
\vdots \\
\{2,\ 1,\ 0,\ -1,\ -2,\ ...,\ -n_b\} \\
\{1,\ 0,\ -1,\ -2,\ ...,\ -n_b\}
\end{gathered}
\tag{3.1}
$$

Se alguma sequência tem dois níveis correspondentes ao mesmo vértice (um positivo e outro negativo para vértices em $C-D$), então basta levar em conta apenas um dos níveis.

A cada vértice retirado ou adicionado saltamos de um grafo para seu vizinho. Como os filhos sempre são obtidos retirando e adicionando um vértice, o conjunto de filhos obtido no final constitui um caminho formado por passos de tamanho dois[3]. O próximo resultado representa uma grande vantagem desse processo de cruzamento.

**Lema 2** *Os filhos de $A$ e $B$ gerados pelas sequências (3.1) são conexos.*

Demonstração: Basta aplicar o lema 1 a cada vértice de cada filho e verificar que existe um caminho daquele vértice ao nó $r$.■

O fato de a transição entre a geração de um filho e outro ser apenas a retirada de um vértice e a adição de outro faz com que a avaliação da verossimilhança seja muito rápida: basta adicionar e subtrair a população e o número de casos das respectivas regiões adicionada e retirada da zona anterior. Além disso, o lema 2 garante que não precisamos verificar se os filhos gerados pelos pais $A$ e $B$ são conexos e, portanto, factíveis.

A idéia por trás dessa operação é que os filhos formam uma transição suave entre os pais $A$ e $B$. Note que o primeiro filho se parece bastante com o pai $A$ e que o último se parece bastante com o pai $B$.

Outro exemplo de cruzamento é mostrado na Figura 3.7. Nesse caso o cruzamento é feito entre um pai bastante alongado $A$ e outro pai bastante compacto $B$.

A cada geração, o algoritmo genético faz várias tentativas de cruzamento, uma vez que o cruzamento só é possível caso haja interseção não-vazia entre os pais. Essas tentativas cessam caso ele atinja o número máximo $ct_{max}$ de cruzamentos tentados ou $cbs_{max}$ de cruzamentos bem sucedidos.

### 3.2.3. O operador de mutação

Operar uma mutação em um indivíduo é simplesmente substituí-lo por um de seus vizinhos, aleatoriamente. Em outras palavras, um subgrafo que sofre uma mutação

[3]Obviamente poderíamos trabalhar com passos de tamanho 1, ou mesmo outros tamanhos. Essa escolha levou em conta que (1) a geração de todos os filhos pode nos conduzir a um número demasiadamente grande de soluções, consumindo muito tempo e (2) a avaliação incremental permite que avaliemos mais de dois filhos, sem aumento significativo do tempo. A escolha de passos de tamanho dois parece ser um bom compromisso entre tempo e número de soluções avaliadas.

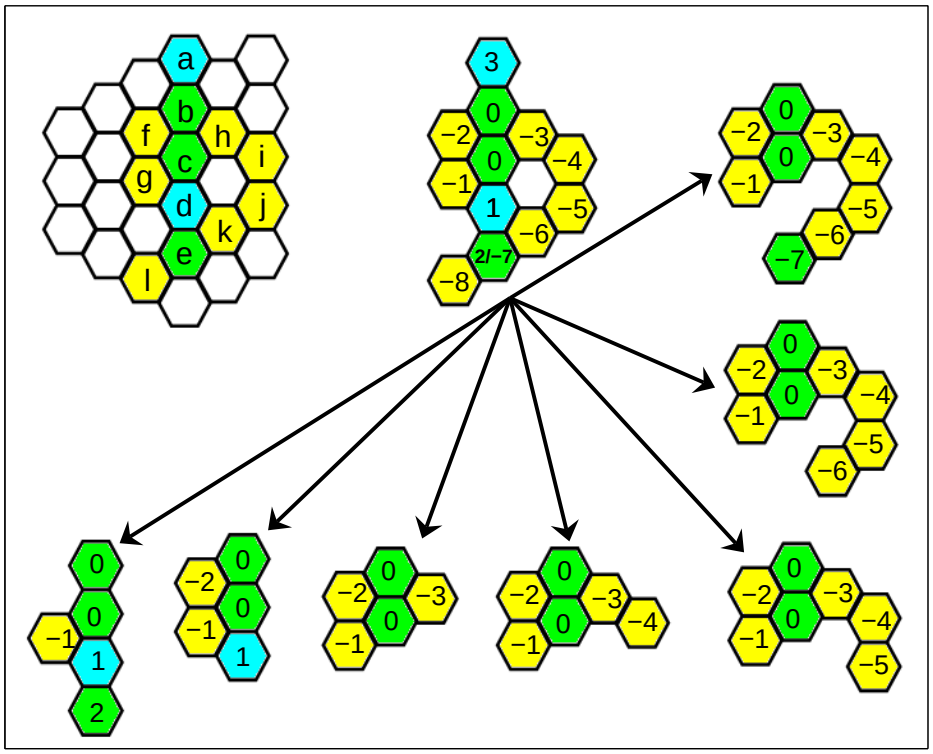



**Figura 3.6.:** Os pais $A = \{a, b, c, d, e\}$ e $B = \{b, c, e, f, g, h, i, j, k, l\}$ têm a parte em comum $C = \{b, c, e\}$. O maior conjunto conexo é escolhido $D = \{b, c\}$. Observe que o vértice $e$, pertencente ao conjunto $C - D$, recebe um nível positivo (2) e negativo (-7).

perde um de seus vértices, ou recebe um novo vértice, desde que permaneça conexo. A escolha de se retirar ou acrescentar um vértice é aleatória, bem como a escolha do vértice a ser retirado ou acrescentado. Note que a mutação constitui uma espécie de busca aleatória, no sentido de que um indivíduo que sofre uma mutação ao longo de algumas iterações segue um processo Markoviano.

A mutação é uma operação computacionalmente cara, caso o novo indivíduo seja obtido retirando-se uma região, uma vez que é necessário verificar a conexidade do subgrafo obtido pelo operador. Porém, em geral a mutação é aplicada apenas em uma pequena fração da população.

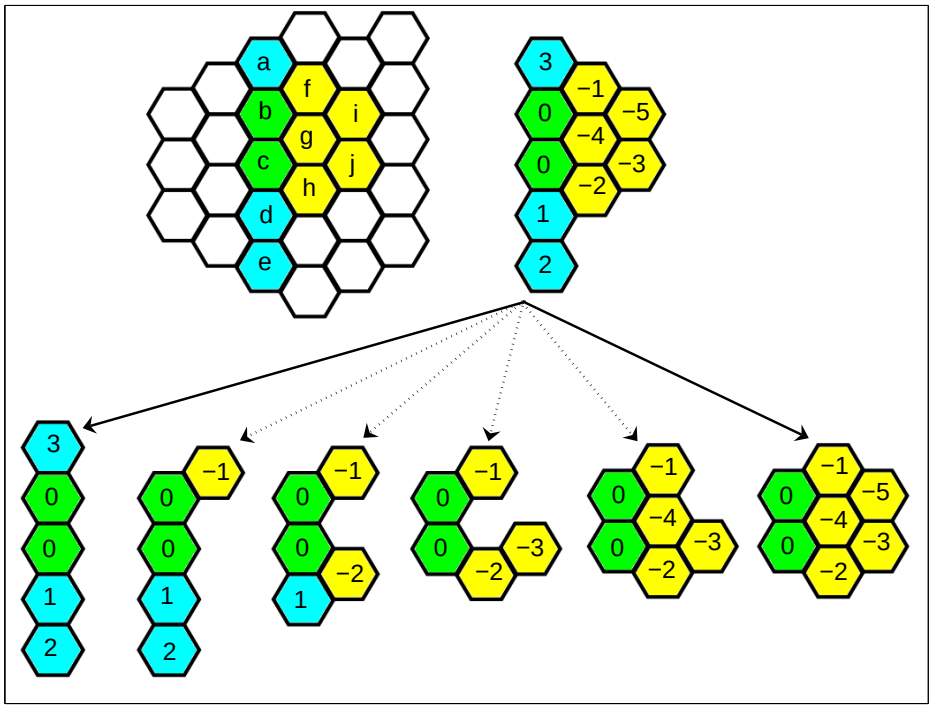



**Figura 3.7.:** Cruzamento entre um pai alongado $A = \{a, b, c, d, e\}$ e outro compacto $B = \{b, c, f, g, h, i, j\}$.

### 3.2.4. O operador de seleção

Antes de operar cruzamento e mutação, uma lista $L$ de $N$ indivíduos é escolhida a partir da população corrente $P$. Essa lista é obtida fazendo-se $N$ torneios binários. Cada torneio é feito sorteando-se dois indivíduos aleatoriamente entre os $N$ indivíduos da população corrente e comparando-os. Aquele com maior aptidão é adicionado à lista $L$. A cada passo o sorteio é feito com reposição. Assim, um indivíduo com alta aptidão tem maior probabilidade de ser colocado na lista $L$ mais de uma vez, enquanto os indivíduos com menor aptidão têm alta probabilidade de não aparecerem nenhuma vez na lista $L$. Os indivíduos da lista $L$ são então escolhidos aleatoriamente aos pares para sofrer cruzamento, até que se atinja o número máximo $ct_{max}$ de cruzamentos tentados ou $cbs_{max}$ de cruzamentos bem sucedidos.

Após operar o cruzamento os filhos são adicionados em uma subpopulação $Q$, a qual ainda sofre mutação. A partir daí é necessário decidir, dentre os pais e os filhos, quais

os $N$ indivíduos que formarão a próxima população. Essa escolha deve ser baseada na aptidão de cada indivíduo. Assim, ordenamos a população formada por $P \cup Q$ segundo a aptidão dos indivíduos e escolhemos, deterministicamente, os $N$ melhores. Esses indivíduos formarão a próxima população. Esse critério é, como veremos na seção 3.3.2, o caso particular do operador de seleção do algoritmo NSGA-II para problemas com apenas um objetivo.

### 3.2.5. Parâmetros e Estrutura do Algoritmo

Para que se consiga chegar a um desempenho satisfatório, o algoritmo genético deve ser ajustado através de alguns parâmetros que determinam seu funcionamento. Esses parâmetros são:

- $N$: tamanho da população
- $p_m$: probabilidade de mutação
- $cbs_{max}$: número máximo de cruzamentos bem sucedidos
- $ct_{max}$: número máximo de cruzamentos tentados
- $g_{max}$: número de gerações, utilizado como critério de parada do algoritmo

Especificamente para o algoritmo descrito nessa tese o tamanho da população $N$ está implicitamente definido como o número de regiões do mapa. O procedimento de geração da população inicial gera um indivíduo partindo de cada uma das $N$ regiões. A probabilidade de mutação $p_m$ utilizada ne maioria dos algoritmos genéticos está próxima de 0,05. No caso do nosso algoritmo, a mutação pode não ser possível, já que se a escolha for por retirar uma região pode não haver nenhuma região passível de ser retirada sem tornar o indivíduo desconexo. Por esse motivo, escolhemos uma probabilidade maior que 0,05. A taxa de mutação foi definida como 0,1. Um valor acima de 0,1 faria com que a busca se tornasse demasiadamente aleatória e o algoritmo se mostrou eficiente com esse parâmetro. Para o número máximo de cruzamentos bem sucedidos $cbs_{max}$ optamos por utilizar o número de cruzamentos em um algoritmo padrão, que é de $N/2$ cruzamentos. Com esse número de cruzamentos, um algoritmo normal (com um cruzamento que gerasse dois indivíduos por cruzamento) obteria $N$ novos indivíduos filhos. No nosso algoritmo, caso esse número de cruzamentos seja atingido, em geral

teremos mais que $N$ indivíduos filhos. Para isso, verificamos para o nosso problema que com um número de tentativas de cruzamento $ct_{max} = 2N$ raramente não atingimos $N/2$ cruzamentos bem sucedidos. Por fim, como critério de parada utilizamos o número máximo de gerações $g_{max}$ que foi fixado em 40. Esse número de gerações nos pareceu suficiente para que a população do algoritmo genético convergisse, na maioria das vezes.

Com esse conjunto de parâmetros, a estrutura do algoritmo genético pode ser descrita da seguinte forma:

```
P_0 ← GULOSO(N);
f_0 ← AVALIA(P_0);
i ← 0;
while i < g_max do
    L ← TORNEIO(P_i);
    ct ← 0;
    cbs ← 0;
    Q_i ← ∅;
    while (ct ≤ ct_max) & (cbs ≤ cbs_max) do
        a ← RANDi(N);
        b ← RANDi(N);
        if L(a) ∩ L(b) ≠ ∅ then
            Q_i ← Q_i ∪ (CRUZAM(L(a), L(b)));
            cbs ← cbs + 1;
        end
        ct ← ct + 1;
    end
    M_i ← SIZE(Q_i);
    for j = 1, ..., M_i do
        p ← RAND();
        if p < p_mut then
            Q_i(j) ← MUT(Q_i(j));
        end
    end
    P_i ← SORT(P_i ∪ Q_i);
    P_i ← P_i(1...N) g ← g + 1;
end
```

**Algoritmo 1**: Algoritmo genético mono-objetivo

No algoritmo 1 aparecem as seguintes funções:

- GULOSO($N$): gera $N$ indivíduos a partir das $N$ regiões do mapa através de um procedimento guloso.
- AVALIA($P$): avalia a população $P$ segundo o critério a ser otimizado.
- TORNEIO($P$): faz $N$ torneios binários com indivíduos escolhidos aleatoriamente, com reposição, a partir da população $P$.
- RANDi($N$): retorna um número aleatório inteiro de 1 a $N$.
- CRUZAM($L(a), L(b)$): opera o cruzamento entre os indivíduos $L(A)$ e $L(b)$, retornando todos os filhos obtidos avaliados.
- SIZE(P): retorna o número de indivíduos da população $P$.
- RAND(): retorna um número aleatório entre 0 e 1.
- MUT($P(i)$): opera mutação no indivíduo $P(i)$ e o avalia.
- SORT($P$): ordena os indivíduos da população $P$ de acordo com a aptidão.

## 3.3. Abordagem multiobjetivo

Nesta seção iremos introduzir alguns conceitos sobre otimização multiobjetivo e apresentar a versão multiobjetivo do algoritmo genético descrito anteriormente. Como o problema tratado nesta tese de doutorado é um problema de maximização, nossas definições levarão em conta que o problema de otimização a ser resolvido é um problema de maximização, ao contrário do que é feito normalmente na literatura.

### 3.3.1. Otimização multiobjetivo

Em muitos problemas reais de otimização há a necessidade de se otimizar simultaneamente duas ou mais funções-objetivo $f_1, f_2, ..., f_n$ (ou uma função-objetivo vetorial $f = (f_1, f_2, ..., f_n)$) sujeitas possivelmente às restrições $g_i(x) \leq 0,\ i = 1, ..., r$. Assim, o problema de otimização multiobjetivo pode ser escrito na forma

$$\begin{aligned} \max_{x} \quad & f(x) = (f_1(x), f_2(x), ..., f_n(x)) \\ \text{s.a.} \quad & g_i(x) \leq 0,\ i = 1, ..., r \end{aligned}$$

Na maioria das vezes os objetivos $f_1, f_2, ..., f_n$ são conflitantes, no sentido de que dificilmente uma mesma escolha de parâmetros $x$ otimiza todos os objetivos simultaneamente. Por essa razão a busca pela melhor solução em um problema com mais de um objetivo está intimamente ligada ao conceito de dominância, dado a seguir.

**Definição 4 (Dominância)** *Seja $f(x) = (f_1(x), ..., f_n(x))$ uma função definida em um espaço $X$. Um ponto $x_1 \in X$ domina outro ponto $x_2 \in X$ (denota-se $x_1 \succ x_2$) se $f_i(x_1) \geq f_i(x_2)$, $i = 1, ..., n$ e se existe pelo menos um índice $k \in \{1, ..., n\}$ tal que $f_k(x_1) > f_k(x_2)$.*

Em outras palavras, um ponto $x_1$ domina o ponto $x_2$ se a avaliação de $x_1$ for melhor que a avaliação de $x_2$ em um objetivo e não for pior em nenhum outro objetivo. Caso o problema seja de minimização, a definição para $x_1 \prec x_2$ vale trocando os sinais $\geq$ e $>$ por $\leq$ e $<$, respectivamente.

Para ilustrar o conceito de dominância, considere a situação onde se deseja maximizar os objetivos $f_1$ e $f_2$. Suponha que $f(x_1) = y_1$ e $f(x_2) = y_2$. Na Figura 3.8(a) $x_1 \succ x_2$ pois $x_1$ é melhor em ambos os objetivos. Já na Figura 3.8(b) os pontos $x_1$ e $x_2$ têm a mesma avaliação no objetivo $f_1$ mas $f_2(x_1) > f_2(x_2)$, portanto $x_1 \succ x_2$. Na Figura 3.8(c) os pontos $x_1$ e $x_2$ são tais que $f_1(x_1) > f_1(x_2)$ e $f_2(x_1) < f_2(x_2)$, de modo que $x_1$ não domina $x_2$, nem $x_2$ domina $x_1$. Neste caso dizemos que $x_1$ e $x_2$ são incomparáveis, ou indiferentes. O símbolo "$\succeq$" denotará "domina ou é indiferente".

Com o conceito de dominância podemos agora definir o objeto essencial na resolução de problemas de otimização multiobjetivo, a *solução Pareto-ótima*.

**Definição 5 (Solução Pareto-ótima)** *Diz-se que uma solução $x^* \in X$ é Pareto-ótima se não existe $x \in X$ tal que $x$ domina $x^*$.*

Note que dizer que uma solução é Pareto-ótima não significa dizer que ela é melhor que todas as (ou que algumas das) outras soluções, mas que ela não é pior que nenhuma outra. Uma solução Pareto-ótima pode ainda ser chamada de solução não-dominada

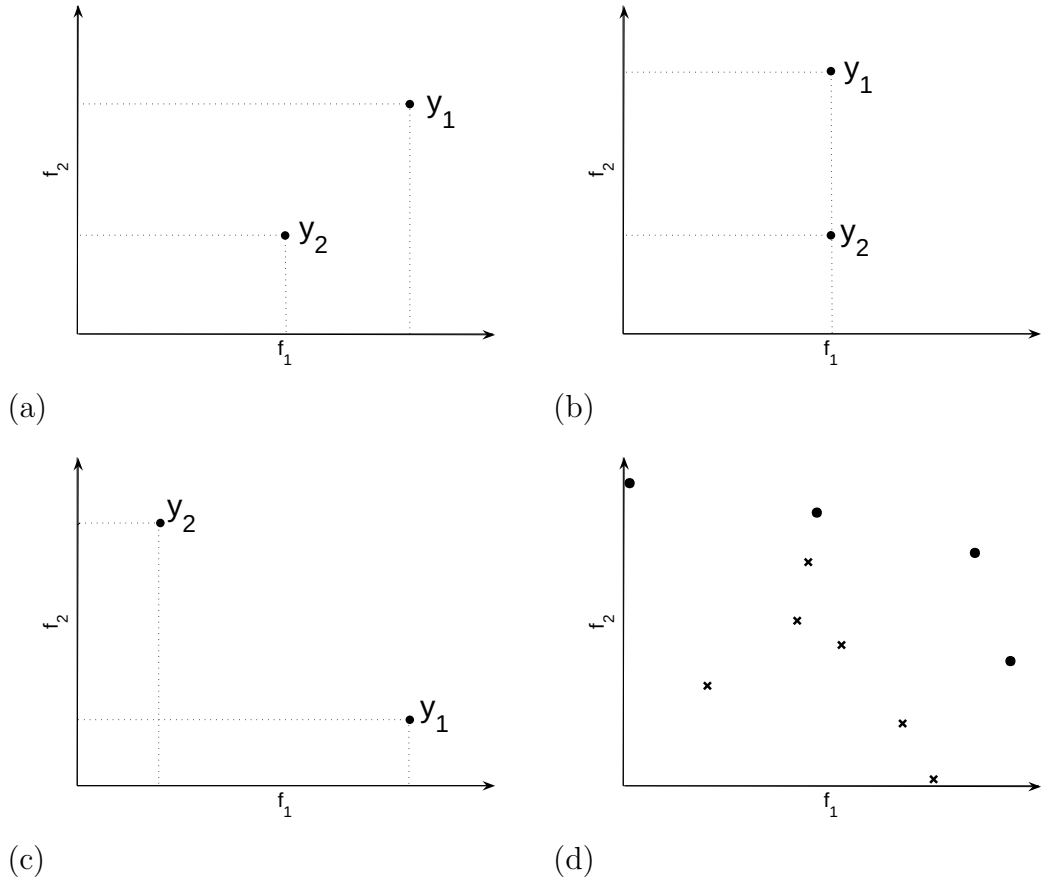



**Figura 3.8.:** (a) $x_1$ domina $x_2$, pois $f_1(x_1) > f_1(x_2)$ e $f_2(x_1) > f_2(x_2)$; (b) $x_1$ domina $x_2$, pois $f_1(x_1) = f_1(x_2)$ e $f_2(x_1) > f_2(x_2)$; (c) $x_1$ não domina $x_2$, e $x_2$ não domina $x_1$ pois $f_1(x_1) > f_1(x_2)$ mas $f_2(x_1) < f_2(x_2)$; (d) Pontos dominados (×) e conjunto de Pareto (•).

ou solução eficiente. O *conjunto Pareto-ótimo* é formado então por todas as soluções Pareto-ótimas. Assim, ao contrário do que ocorre em problemas de otimização com um único objetivo, aqui temos um conjunto de soluções que são, em um certo sentido, ótimas. A Figura 3.8(d) apresenta um exemplo com pontos dominados (×) e os pontos que formam o conjunto de Pareto (•).

O algoritmo genético descrito neste capítulo foi modificado para lidar simultaneamente com as duas grandezas: a compacidade $K$ (seção 2.3) e a estatística espacial *scan* (seção 2.1). A compacidade $K$ não será mais utilizada como uma penalização geométrica, mas como uma nova função objetivo.

### 3.3.2. Algoritmo genético multiobjetivo

Há várias formas de se tratar um problema multiobjetivo de maneira a se obter o conjunto de Pareto. Os métodos, em geral, transformam o problema multiobjetivo em vários sub-problemas mono-objetivo, de forma que a solução de cada sub-problema é um ponto do conjunto de Pareto. Os algoritmos genéticos constituem um método particularmente eficiente para lidar com otimização multiobjetivo, uma vez que trabalham com uma população de soluções-tentativas e, assim, podem encontrar o conjunto de soluções eficientes em uma única execução (Fonseca & Fleming, 1995). Isso é realizado fazendo com que a população toda vá convergindo geração a geração em direção ao conjunto de Pareto, de modo que a aproximação do conjunto de Pareto é obtida simplesmente tomando todos os indivíduos não-dominados encontradas em alguma altura da execução do algoritmo. Exemplos de algoritmos genéticos desenvolvidos para aplicações diferentes podem ser encontrados em Ramos *et al.* (2003), Takahashi *et al.* (2004) e Carrano *et al.* (2006). Os trabalhos Takahashi *et al.* (2004) e Carrano *et al.* (2006) apresentam situações onde o conjunto de Pareto pode ser empregado para a análise *a posteriori* do problema de uma maneira que nenhum algoritmo mono-objetivo poderia fazer.

As diferenças entre os AG's mono e multi-objetivo são muito pequenas. Os operadores de cruzamento e de mutação funcionam da mesma maneira, sem a necessidade de nenhum tipo de adaptação. A alteração fundamental se dá no operador de seleção, uma vez que agora a seleção deve ser feita levando-se em conta não uma, mas duas ou mais funções-objetivo.

No caso específico do algoritmo desenvolvido nesta tese de doutorado a construção da população inicial e os operadores de cruzamento e mutação são idênticos aos empregados no algoritmo descrito no capítulo 3. A diferença fica mesmo por conta da seleção que foi adaptada de forma que o algoritmo se encaixasse na estrutura do NSGA-II. Para isso vamos fazer uso de três procedimentos, descritos a seguir. A descrição detalhada desses procedimentos encontra-se em Deb *et al.* (2002).

1. Ordenação por não-dominância (*nondominated sorting*): consiste em atribuir um nível a cada indivíduo da população. Aos indivíduos da primeira camada de soluções não-dominadas é atribuído o nível 1. O nível 2 é atribuído àqueles presentes na segunda camada de soluções não-dominadas, isto é, aqueles que são dominados exclusivamente por indivíduos do nível 1, e assim sucessivamente.

2. Distância por ocupação (*crowding distance*): é baseada na soma das distâncias entre um indivíduo e seus vizinhos mais próximos em cada objetivo, sendo que os objetivos são normalizados para o cálculo das distâncias.

3. Torneio binário (*binary tournment*): consiste em sortear aleatoriamente dois indivíduos e compará-los de acordo com uma determinada função de ajuste. Aquele que apresentar melhor avaliação dessa função de ajuste é escolhido.

Com relação à ordenação por não-dominância, obviamente uma solução é melhor quanto menor for o seu nível. Um indivíduo do nível 1 não é dominado por nenhuma solução, enquanto um indivíduo do nível 2 é dominado por pelo menos um indivíduo (do nível 1). Já a distância por ocupação funciona como uma medida de ocupação do espaço de objetivos. Para cada indivíduo encontramos o maior hipercubo no espaço de objetivos normalizados que contém aquele indivíduo e mais nenhum outro. A distância por ocupação será a soma das arestas do hipercubo. Se um indivíduo está localizado na extremidade de algum objetivo, isto é, se possui o maior ou o menor valor em um objetivo, atribui-se uma distância infinita. Se uma solução tem alta distância por ocupação, então seus vizinhos se encontram longe dela. Isso dá a idéia de que aquela região é menos "povoada" ou está mais mal representada, e esse indivíduo deve ter mais chances de se manter presente. Por outro lado, um indivíduo com baixa distância por ocupação está numa área que já está muito povoada e, portanto, uma parte daquele nível que provavelmente já está sendo bem representada por outros indivíduos. No torneio binário comparamos o nível das soluções sorteadas. A que tiver menor nível é escolhida. Em caso de empate, a distância por ocupação é comparada e a que possuir maior valor é escolhida.

A estrutura do NSGA-II fica então:

1. A população inicial $P_0$ com $N$ indivíduos é gerada e avaliada. Calculam-se também o nível de não-dominância e distância por ocupação de cada indivíduo.

2. Enquanto o critério de parada não for atingido, a população $P_{i+1}$ da geração $i+1$ é obtida a partir da população $P_i$ seguindo os passos:

    - A partir de $P_i$ efetuam-se $N$ torneios binários de forma a se obter uma lista de $N$ indivíduos selecionados, sobre os quais se operam cruzamento e mutação, obtendo-se então uma lista $Q_i$ com $M_i$ novos indivíduos.

- Forma-se uma população combinada $C_i = P_i \cup Q_i$ com $N + M_i$ indivíduos e calcula-se níveis de não-dominância e distância por ocupação.

- Os indivíduos dos primeiros níveis vão sendo inseridos na próxima população $P_{i+1}$, até que se atinja a quantidade $N$ de indivíduos. Em geral, o último nível inserido, digamos $l$, não poderá ser totalmente adicionado à população $P_{i+1}$ por exceder o número $N$ de indivíduos que se deseja, pelo que eles devem ser inseridos por ordem de distância por ocupação. Os níveis dos indivíduos da população $P_{i+1}$ são preservados enquanto a distância por ocupação deve ser recalculada.

Com essa estrutura nosso AG se aproxima bastante do NSGA-II original. A diferença é devida à natureza do nosso operador de cruzamento. O algoritmo original opera $N/2$ cruzamentos em cada geração, obtendo uma lista de $N$ indivíduos, já que cada cruzamento gera dois novos indivíduos. Como nem sempre é possível operar nosso cruzamento para um dado par de indivíduos, em cada geração nós fazemos um máximo de $ct_{max}$ tentativas de cruzamento com pares aleatórios de indivíduos obtidos da lista resultante do torneio binário, ou um máximo de $n/2$ cruzamentos bem sucedidos. Assim não garantimos que $n/2$ cruzamentos irão ocorrer e, mesmo que ocorram, tipicamente não obteremos $N$ novos indivíduos, já que cada cruzamento gera um número variável de filhos. Isso explica porque a cada geração o conjunto $Q_i$ tem um número variável $M_i$ de indivíduos a cada geração. A estrutura apresentada na forma do algoritmo 2 é bastante parecida com o algoritmo 1. A diferença fica, como já vimos, por conta da seleção que, agora, é feita baseada não apenas na ordem dos indivíduos segundo um único critério (aptidão), mas com o auxílio da função NDS_CROWD, que atribui aos indivíduos um nível na ordenação por não-dominância e uma distância por ocupação.

Apresentamos um exemplo de execução do algoritmo utilizando dados do nordeste dos Estados Unidos (Kulldorff *et al.*, 1997). O mapa tem 245 regiões, população de 29.535.210 mulheres e 58.943 mortes por câncer de mama, durante o período de 1988-1992. Portanto, a taxa de mortalidade anual é de 39,91 a cada 100.000 mulheres. A Figura 3.9 mostra uma seqüência de gráficos $LLR \times$ *compacidade*, obtidos através do algoritmo genético multiobjetivo, para a população inicial e para as gerações 2, 4, 11, 14 e 20.

```
P_0 ← GULOSO(N);
f_0 ← AVALIA(P_0);
i ← 0;
while i < g_max do
    P_i ← NDS_CROWD(P_i);
    L ← TORNEIO(P_i);
    ct ← 0;
    cbs ← 0;
    Q_i ← ∅;
    while (ct ≤ ct_max) & (cbs ≤ cbs_max) do
        a ← RANDi(N);
        b ← RANDi(N);
        if L(a) ∩ L(b) ≠ ∅ then
            Q_i ← Q_i ∪ (CRUZAM(L(a), L(b)));
            cbs ← cbs + 1;
        end
        ct ← ct + 1;
    end
    M_i ← SIZE(Q_i);
    for j = 1, ..., M_i do
        p ← RAND();
        if p < p_mut then
            Q_i(j) ← MUT(Q_i(j));
        end
    end
    P_i ← P_i ∪ Q_i;
    g ← g + 1;
end
```

**Algoritmo 2**: Algoritmo genético multi-objetivo

Observamos o movimento dos pontos em direção a valores maiores de $LLR$ e $K$. Note que a convergência é muito rápida para pontos com alta compacidade. Nesse exemplo, um ponto isolado com alta avaliação de $LLR$ aparece na geração 14, sendo seguido por novos indivíduos com avaliações de $LLR$ ainda maiores na geração 20. Há uma evolução rápida nas primeiras gerações e mudanças sutis na população nas últimas gerações. A população pode conter múltiplas cópias de alguns indivíduos, principalmente nas gerações finais. O conjunto de Pareto da última geração é considerado a solução dada pelo algoritmo genético. Nesse exemplo não ocorrem mais mudanças no conjunto de Pareto nas gerações seguintes. As últimas populações vão se tornando cada vez mais próximas de seus respectivos conjuntos de Pareto, e essa proximidade pode ser utilizada como um critério de convergência.

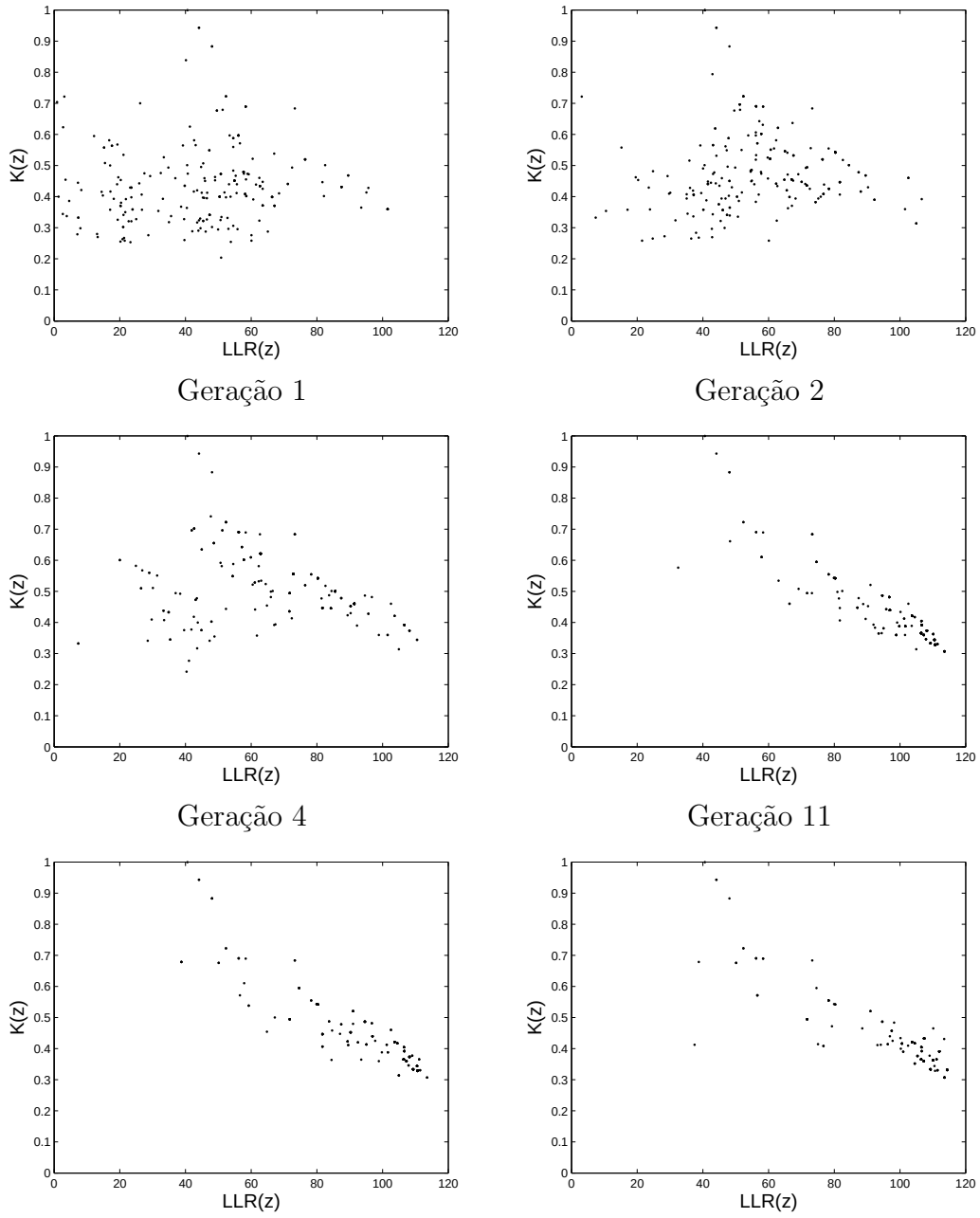



**Figura 3.9.:** Evolução da população no algoritmo genético multiobjetivo, ao longo de 20 gerações.

## 3.4. Discussão

Não se deve esperar que *clusters* espaciais sejam compactos. *Clusters* circulares foram adotados no *scan* circular por razões computacionais e por simplicidade. Apesar de

suas limitações, o *scan* circular funciona bem exatamente porque ele restringe fortemente a forma dos *clusters* analisados. Dado que a forma de todos os candidatos a *cluster* é a mesma, basta escolher aquele que apresenta maior avaliação de razão de verossimilhança. Como mencionado anteriormente, o *scan* circular impõe uma função de penalização fortíssima, eliminando completamente os *clusters* cujas formas não sejam circulares. Obviamente que, no contexto de *clusters* irregulares, o primeiro objetivo (regularidade da forma) não poderia ser considerado apropriado caso fosse o único objetivo a ser otimizado. Procedendo dessa maneira iríamos, inevitavelmente, obter uma solução circular, mas sem nenhum significado. Por outro lado, considere a situação inversa, onde o único objetivo a ser otimizado é a razão de verossimilhança, sem levar em conta a forma da solução. Como foi visto, isto também nos levaria a soluções que não têm nenhuma utilidade sob uma perspectiva geográfica. A maximização da regularidade da forma só faz sentido quando utilizada em conjunto com a maximização da verossimilhança, como desenvolvido na metodologia multiobjetivo. Isoladamente, nenhum dos objetivos é suficiente para guiar a busca do *cluster* mais verossímil quando temos liberdade para escolher *clusters* de forma arbitrária. Um *cluster* que tenha a forma mais arredondada geralmente tem muitas conexões entre suas regiões, comparado com o número de regiões que o compõem. Por outro lado, *clusters* irregulares se aproximam mais de árvores, de modo que o número de conexões entre as regiões tende a ser pequena comparada com o número de regiões. Em uma situação onde dois *clusters* têm a mesma avaliação de $LLR$, sendo um regular e outro irregular, damos preferência ao primeiro. A compacidade de um *cluster* está relacionada com a força com que suas regiões se conectam entre si. Sob esse aspecto, a compacidade é considerada uma medida de estabilidade do *cluster* como entidade geográfica. Podemos remover algumas regiões de um *cluster* circular sem que ele se quebre em partes desconexas entre si. Mas em um *cluster* extremamente irregular essa operação certamente nos levará a desconectar o *cluster*, na maior parte das vezes.

# Capítulo 4.

# Inferência Estatística

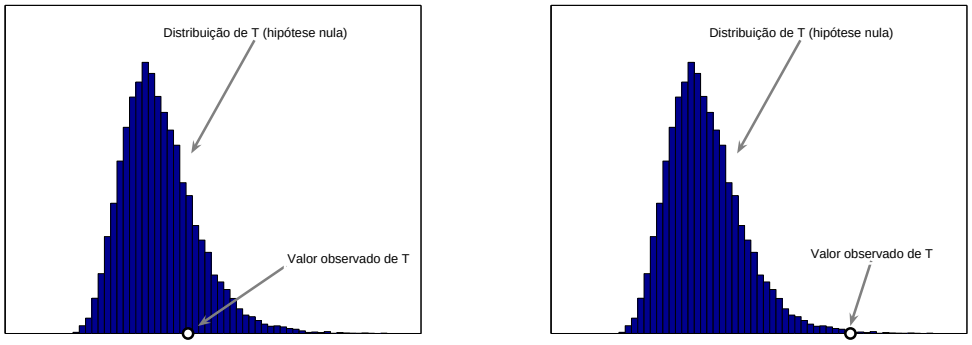
Como foi visto, os métodos de detecção são utilizados para se encontrar o *cluster* que maximiza a estatística *scan*. Como o número de candidatos a *cluster* é finito, é claro que, sempre que se fizer uma busca pelo *cluster* mais verossímil, independente do método aplicado, alguma solução será encontrada. Essa solução será a que, dentre todas as analisadas, apresentar a maior avaliação de $LLR$, penalizada ou não. Antes de podermos afirmar que essa solução é um *cluster*, devemos levar em conta que um *cluster* deve apresentar um número anormal de casos. Em outras palavras, não podemos afirmar que uma medida é discrepante das demais simplesmente porque ela é a maior dentre todas as avaliadas. Essa medida deve ser comparada com um universo de medidas. A partir desse universo é que será possível estabelecer uma medida crítica, acima da qual uma medida pode ser considerada anormal.

Da mesma forma, para considerarmos que a solução encontrada pelo método de detecção é um *cluster*, devemos comparar sua avaliação com as avaliações de soluções encontradas para vários cenários aleatórios. Só a partir dessa comparação é que será possível afirmar se a solução é ou não um *cluster*. A seguir vamos introduzir técnicas para se fazer essa inferência, utilizando um método clássico empírico e um método paramétrico, para o caso mono-objetivo e sua extensão para o caso bi-objetivo.

## 4.1. Caso mono-objetivo

Se conhecêssemos a distribuição de probabilidade seguida pela estatística $T = \max_z LLR(z)$ sob a hipótese de que não há *clusters* no mapa, poderíamos calcular o valor de $T_{crit}$

acima do qual poderíamos considerar, sob $h_0$, uma solução discrepante, simplesmente resolvendo a equação $P(T > T_{crit}) = \alpha$, onde $T$ é a estatística sob a hipótese nula e $\alpha$ é a probabilidade de que $T$ supere o valor crítico $T_{crit}$, chamado de *nível de significância*, tipicamente escolhido como $\alpha = 0,05$. Isso significa que $T_{crit}$ indica o valor que separa o que é normal do que é discrepante, supondo verdadeira a hipótese nula. Ou seja, um valor de $T$ abaixo de $T_{crit}$ pode ocorrer por mero acaso 95% das vezes, mas um valor acima de $T_{crit}$ só acontece por acaso com probabilidade menor que ou igual a 0,05 e, portanto, a solução pode ser considerada um *cluster*. Essa probabilidade de que o valor observado da estatística *scan* ocorra por mero acaso sob $h_0$ é chamada de probabilidade de significância do teste ($p$-valor). Assim, um $p$-valor pequeno indica que o valor observado da estatística de teste é pouco provável sob $h_0$, logo esta deveria ser rejeitada, indicando a provável existência de um *cluster*. Quando o $p$-valor da existência de um *cluster* é menor que o nível de significância $\alpha$ dizemos que a existência daquele *cluster* é significativa ao nível $\alpha$. A Figura 4.1(a) mostra um caso em que o valor observado $T_{obs}$ é um valor típico da estatística $T$ sob $h_0$ e portanto a solução mais verossímil possui alto $p$-valor. Já na Figura 4.1(b) o valor observado é atípico sob $h_0$ e o $p$-valor é pequeno. Essa solução deve ser considerada um *cluster*.




(a) (b)

**Figura 4.1.:** (a) O valor de $T_{obs}$ é típico sob $h_0$ e portanto o $p$-valor do *cluster* é alto. (b) O valor de $T_{obs}$ é atípico sob $h_0$ e portanto o $p$-valor do *cluster* é baixo.

O problema é que, a princípio, não conhecemos a distribuição da estatística *scan*. Por isso a inferência é feita a partir de simulações de Monte Carlo (Dwass, 1957). O método de Monte Carlo consiste em simular um sistema diversas vezes, cada uma delas com uma nova reamostragem aleatória para as entradas do sistema. Podemos fazer distribuições aleatórias dos casos sobre o mapa seguindo o modelo de Poisson e, para

cada um desses cenários, calcular a estatística *scan*, obtendo então sua distribuição empírica. Essas distribuições são feitas aleatória e independentemente através de um processo de Poisson não-homogêneo. Para tanto, supomos que o número de casos na região $r_j$ segue uma distribuição $Po(\mu_j)$, como descrito na seção 2.1. Assim geramos o número de casos em cada região. Fazendo isso várias vezes estaremos gerando vários cenários diferentes, todos sob a hipótese nula de que os casos estão distribuídos com probabilidades iguais para indivíduos em qualquer região. O valor observado $T_{obs}$ da estatística *scan* é então comparado ao valor critíco $T_{crit}$, que pode agora ser calculado empiricamente, simplesmente ordenando todos os valores de $T$ obtidos nas simulações e calculando o quantil amostral de ordem $1-\alpha$, $0 < \alpha < 1$. Em particular, se $\alpha = 0,05$, $T_{crit}$ será o quantil 0,95. Caso $T_{obs}$ exceda $T_{crit}$ considera-se que o *cluster* é significante ao nível $\alpha$. Nesse caso saberemos que o $p$-valor do *cluster* encontrado é menor que $\alpha$, mas note que se fizermos a contagem do número $w_0$ de valores de $T$ sob a hipótese nula que são maiores que o valor de $T_{obs}$, o $p$-valor da solução encontrada é dado por $(w_0 + 1)/w$, onde $w$ é o número de simulações sob a hipótese nula.

Agora, considere o conjunto $S = \{T_1, T_2, ..., T_w\}$, onde $T_j$ é o valor da estatística $T$ obtido na $j$-ésima simulação. Se $T_{obs} > T_j$ para $j = 1, ..., w$, então $w_0 = 0$ e o *cluster* encontrado é significante ao nível $1/w$. Porém, pode ser que o $p$-valor real seja menor que $1/w$, mas precisaríamos fazer mais simulações até que encontrássemos valores de $T$ sob a hipótese nula maiores que $T_{obs}$. Essa é uma desvantagem dessa técnica de cálculo de $p$-valor. O número de simulações necessárias para que se possa estimar com precisão um $p$-valor muito pequeno pode ser extremamente alto, o que pode tornar inviável o cálculo preciso do $p$-valor. Uma forma de extrapolar os dados obtidos pela simulação de Monte Carlo seria possível fazendo uso da técnica de núcleo estimador (Parzen, 1962), através da qual pode-se obter uma estimativa melhor para a distribuição empírica.

### 4.1.1. Cálculo paramétrico do $p$-valor

Como foi visto na seção anterior, utilizar simulações de Monte Carlo para estimar o $p$-valor é conveniente já que não conhecemos a distribuição da estatística de teste. No entanto, em Abrams *et al.* (2006) foi mostrado através de testes exaustivos que, para o *scan* circular, a estatística $T$ sob $h_0$ parece seguir uma distribuição conhecida como distribuição de Gumbel (Kotz & Nadarajah, 2000). Isso não acontece por acaso e pode ser explicado pelo fato de que a distribuição de Gumbel é uma distribuição de valores

extremos. Mais especificamente, pode-se mostrar (Johnson *et al.*, 1995; Coles, 2001) que, para uma dada seqüência de variáveis aleatórias $\{X_1, X_2, ..., X_n\}$ independentes e identicamente distribuídas, se $Y = \max\{X_1, X_2, ..., X_n\}$, então a distribuição assintótica de $Y$ quando $n \to \infty$, caso exista, é uma distribuição de valores extremos.

A função de densidade de probabilidade da distribuição de Gumbel é dada por:

$$f(x) = \frac{1}{\theta} exp\left\{-e^{(\frac{x-\mu}{\theta})} - \left(\frac{x-\mu}{\theta}\right)\right\}, \ x \in \mathbb{R} \tag{4.1}$$

onde $\mu \in \mathbb{R}$ é um parâmetro de locação (mais especificamente, a moda) e $\theta > 0$ é um parâmetro de escala. Desse modo, ao invés de se fazer um número enorme de simulações de Monte Carlo, pode-se fazer um número razoável de execuções e, a partir dos valores obtidos de $T$, estimar os parâmetros $\mu$ e $\theta$. Substituindo as estimativas $\hat{\mu}$ e $\hat{\theta}$ na expressão da função de densidade da distribuição de Gumbel dada em (4.1) obtemos a função de densidade estimada $\hat{f}$. Com $\hat{f}$ e o valor de $T_{obs}$ é fácil calcular o $p$-valor, simplesmente calculando a integral:

$$P(T > T_{obs}) = \int_{T_{obs}}^{\infty} \hat{f}(t)dt$$

que é a probabilidade de que $T$ seja maior que $T_{obs}$. Mais adiante iremos avaliar a qualidade do ajuste do modelo Gumbel e de outro modelo de valores extremos, o de Weibull, tanto para o *scan* circular quanto para o AG.

## 4.2. Caso multiobjetivo

Para calcular a significância estatística dos pontos do conjunto de Pareto do mapa de casos observados devemos compará-los com os conjuntos de Pareto obtidos para cada um dos mapas de casos simulados sob a hipótese nula, obtidos através de uma simulação de Monte Carlo, a exemplo do que acontece no caso de mono-objetivo. O algoritmo genético multiobjetivo é executado várias vezes para mapas contendo casos distribuídos aleatoriamente conforme a distribuição de Poisson sob a hipótese nula, onde a probabilidade de ocorrência de casos em cada região é proporcional à população naquela região.

O processo de obtenção do conjunto de Pareto é repetido para cada uma dessas alocações aleatórias de casos. Esses conjuntos de Pareto são agrupados, formando uma coleção de milhares de pontos distribuídos no espaço $LLR \times K$, que é a faixa $(0, \infty) \times (0, 1]$. Nesse caso, ao invés de encontrar o ponto crítico, acima do qual consideramos que um *cluster* é significativo, devemos encontrar uma curva crítica. Essa curva crítica divide o plano em duas regiões de maneira que um ponto do plano será considerado um *cluster* significativo se estiver acima dessa curva. Recaímos então na questão de como encontrar essa curva crítica. Apresentamos aqui diferentes formas de fazê-lo, na ordem em que foram propostas, até chegar na maneira que acreditamos ser a mais adequada. Nesta tese de doutorado propomos três técnicas para o cálculo do $p$-valor para o caso multiobjetivo. A seguir explicamos como inferir o $p$-valor das soluções utilizando cada uma das técnicas.

### 4.2.1. Descascamento

Considere $S_0$ o conjunto de todos os $n$ pontos obtidos pela simulação de Monte Carlo sob a hipótese nula. A curva de menor $p$-valor considerando os dados obtidos nessa simulação seria formada exatamente pelo conjunto de Pareto $P_0$ da união de todos os $n$ pontos. Se $P_0$ possui $n_0$ pontos, então o $p$-valor de um *cluster* observado não-dominado por nenhum ponto desse conjunto de Pareto seria $n_0/n$. Agora, fazendo $S_1 = S_0 - P_0$ podemos encontrar o conjunto de Pareto $P_1$ dos pontos do conjunto $S_1$. Se $P_1$ tem $n_1$ elementos esse conjunto forma a curva de $p$-valor $(n_0 + n_1)/n$. Podemos repetir o procedimento até alcançar o $p$-valor desejado, de forma que para encontrar a curva de $p$-valor $\alpha$ basta encontrar o conjunto $P_{i_{crit}}$ de forma que

$$\frac{\sum_{i=0}^{i=i_{crit}} n_i}{n} = \alpha \tag{4.2}$$

onde $n_i$ é o número de elementos do conjunto de Pareto $P_i$. Observe que este método constitui em se fazer um "descascamento" do conjunto de pontos através dos conjuntos de Pareto até que se atinja a "casca" com o $p$-valor desejado. A Figura 4.2 exibe o conjunto de Pareto crítico obtido por esse método. A nuvem de pontos é formada por 10.000 pontos e o conjunto de Pareto crítico divide essa nuvem no $p$-valor 0,0535, isto

é, o conjunto formado pelo conjunto de Pareto crítico mais os pontos não-dominados por ele tem 535 elementos. Esse conjunto foi obtido após 10 iterações, ou seja, após a retirada de 10 "cascas".

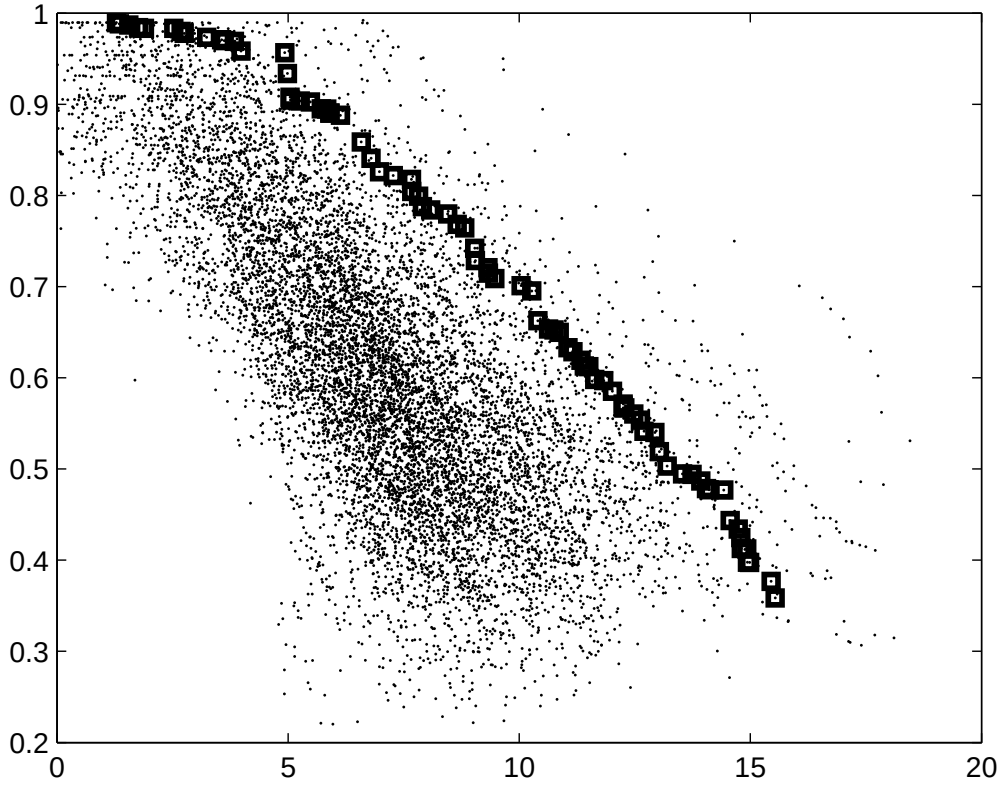



**Figura 4.2.:** Conjunto de Pareto crítico 0,0535. Um *cluster* observado não-dominado por esse conjunto tem $p$-valor menor que 0,0535.

A técnica do descascamento apresenta algumas características que desfavorecem seu uso. Em primeiro lugar, os pontos foram obtidos a partir de vários conjuntos Pareto-ótimos. No entanto, a análise não leva em conta essa dependência entre os pontos. Ao contrário, tratamos os pontos como entidades independentes. Além disso, como veremos adiante, uma abordagem paramétrica se torna um tanto inviável, uma vez que teríamos que aproximar uma distribuição das cascas, ou seja, uma distribuição conjunta de duas variáveis, o que seria muito difícil.

### 4.2.2. Faixas

A fim de viabilizar a extensão paramétrica para o caso multiobjetivo, vamos dividir o espaço $(0, \infty) \times (0, 1]$ em $m$ faixas horizontais paralelas, sendo a $j$-ésima faixa o espaço $(0, \infty) \times (s_j, s_{j+1}]$, $s_j < s_{j+1}$. Agora, para cada uma dessas faixas podemos utilizar a abordagem empírica ou paramétrica para o caso mono-objetivo, simplesmente utilizando os pontos obtidos na simulação de Monte Carlo que caem dentro de cada faixa (veja Figura 4.3). Os valores $s_j$ e $s_{j+1}$ devem ser escolhidos próximos o bastante de forma que a distribuição não mude muito para valores diferentes de compacidade no intervalo $(s_j, s_{j+1})$ e também que a faixa contenha um número suficiente de pontos que nos permita fazer inferência.

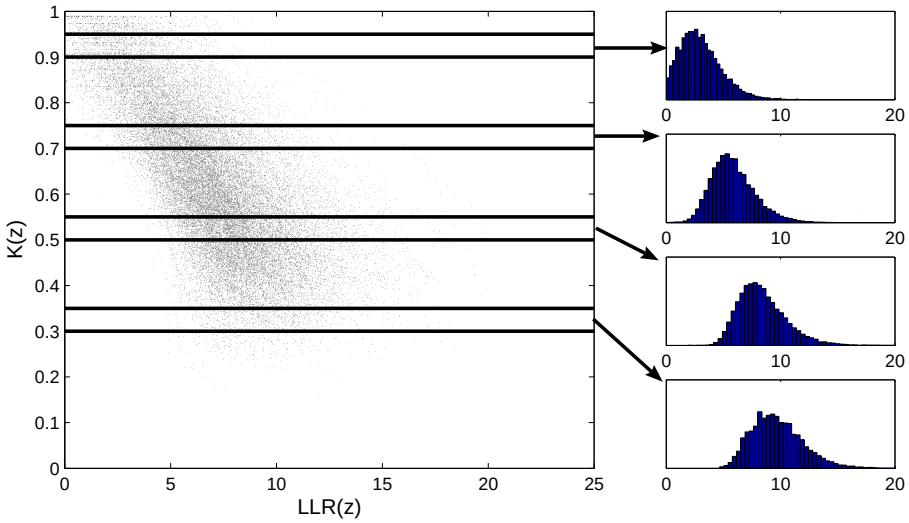


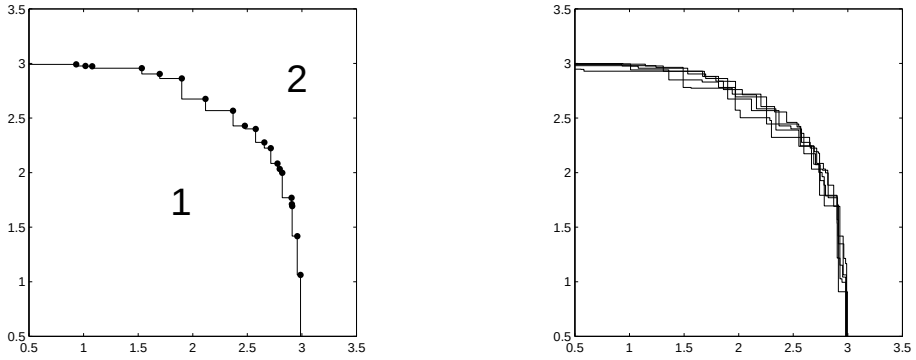

**Figura 4.3.:** O espaço $LLR$ vs. $K$ é dividido em faixas e a análise unidimensional é feita para cada faixa.

Uma grande vantagem desse método é que ele permite que nos utilizemos de um método paramétrico simples. Este método foi utilizado em Duczmal *et al.* (2008) e representa um avanço. No entanto ele apresenta outros problemas. O primeiro deles é a escolha de faixas de largura positiva que, além de constituir uma dificuldade em si (qual é a largura ideal?), impede a comparação do valor de $LLR$ do cluster detectado com a distribuição sob a hipótese nula para um particular valor de compacidade. Permite apenas que comparemos com toda uma faixa de compacidades. Além disso, tratar os conjuntos Pareto-ótimos como simples pontos independentes constitui perda de informação, uma vez que abandonamos a estrutura do conjunto. Isso permite, por exemplo, que

numa mesma faixa haja contribuição de vários pontos de uma das fronteiras e nenhum ponto de outra.

### 4.2.3. Função de aproveitamento

Considere um conjunto $\mathscr{Y} = \{y_j \in \mathbb{R}^d, j = 1, ..., M\}$ composto pelas soluções não-dominadas obtidas por uma execução de um algoritmo bi-objetivo. A esse conjunto podemos associar uma fronteira que separa o espaço de objetivos em duas regiões: (1) os pontos dominados por ou iguais a pelo menos um dos pontos do conjunto de Pareto e (2) os pontos que não são dominados por nenhum dos pontos do conjunto de Pareto (veja Figura 4.4(a)). Essa fronteria é chamada de fronteira de aproveitamento. Agora considere múltiplas execuções do algoritmo. Como cada execução produz conjuntos de Pareto diferentes iremos obter uma figura parecida com o que é mostrado na Figura 4.4(b). Pontos localizados acima e à direita de todas as superfícies de Pareto não foram atingidos em nenhuma execução. Por outro lado, os pontos localizados abaixo e à esquerda de todas as superfícies foram atingidos em todas as execuções. Pontos localizados entre as superfícies foram atingidos em algumas execuções e não em outras, de modo que podemos dividir o espaço em regiões de acordo com a frequência com que essas regiões são atingidas. Isso aproxima a probabilidade de um ponto no espaço de objetivos ser atingido.




(a) (b)

**Figura 4.4.:** (a) A superfície de aproveitamento divide o espaço em duas regiões. (b) A função de aproveitamento obtida por múltiplas execuções do algoritmo biobjetivo.

A função de aproveitamento[1] (da Fonseca *et al.*, 2001; Fonseca *et al.*, 2005) descreve a probabilidade de um conjunto de pontos não-dominados $\mathscr{Y}$, produzido pela execução do algoritmo bi-objetivo, atingir um ponto $y$ no espaço de objetivos, e é definida pela função $A_{\mathscr{Y}} : \mathbb{R}^d \to [0, 1]$, com:

$$A_{\mathscr{Y}}(y) = P(y_1 \trianglerighteq y \ \vee \ y_2 \trianglerighteq y \ \vee \ ... \ \vee \ y_M \trianglerighteq y)$$

onde o símbolo "$\vee$" é o "ou" lógico. A função $A_{\mathscr{Y}}(y)$ denota a probabilidade de pelo menos um elemento do conjunto $\mathscr{Y}$ atingir a meta $y$ em uma execução do algoritmo. A função de aproveitamento pode ser estimada a partir dos conjuntos $\mathscr{Y}_1, ..., \mathscr{Y}_n$ obtidos a partir de $n$ execuções independentes do algoritmo, como

$$A_n(y) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{I}(\mathscr{Y}_i \trianglerighteq y)$$

onde $\mathbf{I}$ é a função-indicador (igual a 1 se $\mathscr{Y}_i \trianglerighteq y$, e zero caso contrário) e o símbolo "$\trianglerighteq$" significa que pelo menos um elemento de $\mathscr{Y}_i$ domina ou é igual a $y$.

Consideremos o nosso espaço de objetivos $(LLR, K)$. Considere ainda um ponto $y_{obs} = (l, k)$ do conjunto Pareto-ótimo obtido para os casos observados e a reta $K = k$ paralela ao eixo das abcissas. Essa reta cruza todas as superfícies de aproveitamento e os pontos de cruzamento produzem uma distribuição univariada. Queremos usar essa distribuição para calcular a probabilidade de que o valor $l$ seja atingido, considerando os dados que estão acima da reta $K = k$, isto é, estamos interessados em calcular

$$P(LLR > l | K \geq k) = 1 - F(l | K \geq k)$$

Assim, dada uma solução $y_{obs} = (l, k)$, pertencente ao conjunto Pareto-ótimo obtido para os casos observados, o $p$-valor de $y_{obs}$ será dado por

$$1 - F(l | K \geq k) + P(LLR = l | K \geq k)$$

o que pode ser calculado impondo-se a condição $K \geq k$ para a função de aproveitamento empírica $A_n(y_{obs})$.

[1] Aqui, mais uma vez, fugimos à definição da literatura que leva em conta um problema de minimização e definimos a função de aproveitamento levando em conta que o problema em questão é um problema de maximização.

Com o uso da função de aproveitamento estendemos de forma mais natural o significado de $p$-valor para o espaço biobjetivo, ao mesmo tempo em que preservamos a "natureza de conjunto" dos pontos obtidos pela simulação de Monte Carlo, contrastando com o que foi proposto anteriormente, quando os conjuntos foram dissolvidos em pontos independentes, fazendo com que se perdesse informação sobre a distribuição dos conjuntos de Pareto sob a hipótese nula.

### 4.2.4. Cálculo paramétrico do $p$-valor

Tanto na abordagem das faixas quanto utilizando a função de aproveitamento é possível estender a aproximação paramétrica para o cálculo do $p$-valor. Novamente, a vantagem do cálculo paramétrico do $p$-valor das soluções é que podemos fazer inferência com um número razoável de simulações. Ainda que a nuvem de pontos obtidos sob a hipótese nula não avance o suficiente para envolver o conjunto de Pareto obtido para os casos observados, podemos estimar o $p$-valor para cada ponto do conjunto, utilizando uma distribuição ajustada.

No caso da abordagem por faixas, o espaço $(0,\infty)\times(0,1]$ é particionado pelas faixas $(0,\infty)\times(s_j,s_{j+1}]$, $s_j < s_{j+1}$. Para cada uma dessas faixas utilizamos os pontos que caem em seu interior para podermos estimar os parâmetros necessários para se chegar à distribuição que se ajusta aos dados daquela faixa. Seja $f_j$ a função de densidade de probabilidade da distribuição para a faixa $(0,\infty)\times(s_j,s_{j+1}]$ e seja $y_{obs} = (l,k)$ um ponto do conjunto de Pareto encontrado pelo algoritmo para os casos observados tal que $k \in (s_j,s_{j+1}]$. A função distribuição $F_j$ da faixa $j$ que contém o ponto $k$ é então utilizada para calcular seu $p$-valor, através da integral

$$\int_l^{\infty} f_j(t)dt \tag{4.3}$$

Como no caso não-paramétrico, a dificuldade aqui se encontra no fato de que, obviamente, as faixas possuem largura positiva, o que faz com que inevitavelmente utilizemos pontos com diferentes valores de compacidade na estimação, introduzindo um erro na distribuição obtida. A necessidade de escolha da largura das faixas também é uma desvantagem por ser um procedimento arbitrário, além do fato de que a contribuição de

cada fronteira de Pareto em cada faixa ser diferente: muitas vezes uma mesma fronteira contribui com vários pontos em algumas faixas e com nenhum ponto em outras.

Como foi visto, todos esses problemas são sanados com a introdução do conceito da função de aproveitamento e o cálculo paramétrico do $p$-valor pode ser feito parametricamente com a utilização da distribuição empírica $F(l|K \geq k)$ descrita anteriormente para se ajustar a distribuição paramétrica para cada valor $K_{obs}$.

## 4.3. Modelos paramétricos

Como mencionado na seção 4.1.1, a distribuição de Gumbel foi utilizada para modelar a distribuição da estatística de teste $T$ sob a hipótese nula para o *scan* circular. Vimos ainda que essa escolha é coerente com o fato de que a distribuição de Gumbel é uma distribuição de valores extremos. Vamos investigar a qualidade desse modelo, bem como outro modelo, o de Weibull (Johnson *et al.*, 1995). A distribuição de Weibull também é uma distribuição de valores extremos e já foi utilizada para modelar a distribuição do resultado de otimizadores estocásticos (Hüsler *et al.*, 2002). A seguir descrevemos as distribuições de Gumbel e Weibull em termos de suas funções de densidade de probabilidade e funções de probabilidade acumulada.

### 4.3.1. Modelo Gumbel

A função de densidade de probabilidade da distribuição de Gumbel para máximos é dada por

$$f_G(x;\mu,\theta) = \frac{1}{\theta} e^{-e^{\left(\frac{\mu-x}{\theta}\right)}} e^{\left(\frac{\mu-x}{\theta}\right)}, \ x \in \mathbb{R} \tag{4.4}$$

e a função de probabilidade acumulada é dada por

$$F_G(x;\mu,\theta) = e^{-e^{\left(\frac{\mu-x}{\theta}\right)}}, \ x \in \mathbb{R} \tag{4.5}$$

onde $\mu \in \mathbb{R}$ é o parâmetro de locação e $\theta > 0$ é o parâmetro de escala.

### 4.3.2. Modelo Weibull

A função de densidade de probabilidade da distribuição de Weibull de três parâmetros para máximos (ou Weibull reversa) é dada por

$$f_W(x;\alpha,\beta,\gamma) = \begin{cases} \frac{\gamma}{\beta}\left(\frac{\alpha-x}{\beta}\right)^{\gamma-1} e^{-\left(\frac{\alpha-x}{\beta}\right)^{\gamma}} & \text{se } 0 < x < \alpha \\ 0 & \text{se } x \geq \alpha \end{cases} \tag{4.6}$$

e a função de probabilidade acumulada

$$F_W(x,\alpha,\beta,\gamma) = \begin{cases} e^{-\left(\frac{\alpha-x}{\beta}\right)^{\gamma}} & \text{se } 0 < x < \alpha \\ 0 & \text{se } x \geq \alpha \end{cases} \tag{4.7}$$

onde $\alpha$, $\beta$ e $\gamma$ são números reais positivos e, respectivamente, os parâmetros de de locação, escala e forma.

### 4.3.3. Estimação de parâmetros

Para ambos os modelos consideramos os estimadores de máxima verossimilhança e de mínimos quadrados, mas os estimadores de máxima verossimilhança apresentaram resultados melhores, de maneira que apresentaremos os resultados apenas para esses estimadores. A estimação de máxima verossimilhança para ambos os modelos nos conduz a equações que não têm solução fechada, pelo que devemos resolver as equações numericamente. As estimativas foram computadas utilizando-se a função `fzero` do *software* MatLab®. Essa função utiliza o algoritmo de Dekker que é uma combinação dos métodos da secante, da bisseção e interpolação quadrática inversa (Dekker, 1969; Brent, 1973).

A estimação para o modelo de Gumbel é feita diretamente a partir das soluções numéricas de maximização da verossimilhança. Para o modelo de Weibull temos que

estimar três parâmetros. A estimação dos parâmetros de escala e forma, respectivamente $\beta$ e $\gamma$, é dependente do parâmetro de locação $\alpha$. Para estimar $\alpha$ restringimos nosso espaço de procura de acordo com a sugestão dada em Qiao & Tsokos (1995), onde mostra-se que, para uma dada amostra $\{x_1, ..., x_n\}$, a probabilidade de que o intervalo $[L,U]$) contenha o parâmetro $\alpha$ aumenta e tende para 1 à medida que o tamanho $n$ da amostra aumenta, onde:

$$L = \max(x_i)$$

$$U = 2\max(x_i) - \min(x_i)$$

Isto sugere a restrição do espaço de busca ao intervalo $[L,U]$. Essa busca é feita através de uma busca por seção áurea unidimensional. Para uma estimativa $\hat{\alpha}$ do parâmetro de locação podemos obter as estimativas de máxima verossimilhança $\hat{\beta}$ e $\hat{\gamma}$ para os outros parâmetros e calcular o erro quadrático, dado por

$$QE = \sum_{i=1}^{n}(F(x_i) - \hat{F}(x_i))^2$$

onde $\hat{F}(x_i) = F(x_i|\hat{\alpha};\hat{\beta};\hat{\gamma})$. A busca para quando este erro se torna menor que uma tolerância $\epsilon$ ou um número máximo de iterações é atingido.

## 4.4. Resultados experimentais

Nesta seção iremos comparar o desempenho dos modelos de Gumbel e Weibull para o algoritmo Scan Circular e para o AG mono e bi-objetivo. Para medir a qualidade da aderência dos modelos iremos utilizar o teste de Kolmogorov-Smirnov, que é baseado na maior diferença entre as distribuições acumuladas dos dados e da distribuição que está sendo presumida (ver Anexo B). No nosso caso a distribuição presumida é a distribuição cujos parâmetros foram estimados a partir dos dados, de forma que não podemos utilizar a distribuição de Kolmogorov-Smirnov para calcular valores críticos e $p$-valores (Keutelian, 1991). Por esse motivo, a distribuição da estatística de teste foi estimada através de simulações de Monte Carlo. A cada passo gera-se uma amostra da

distribuição presumida e em seguida re-estimamos os parâmetros a partir da amostra e calculamos a estatística de teste. Ao final, o $p$-valor será dado pela proporção de vezes que a estatítica de teste foi maior que a estatística de teste para os dados originais. Para todos os testes foram utilizados dados do mapa do nordeste dos Estados Unidos, com 245 regiões, população de risco $N = 29.535.210$ e número total de casos $C = 58.943$.

### 4.4.1. Scan Circular e AG mono-objetivo

Para o caso mono-objetivo usamos o método Scan Circular clássico e o AG. O qq-plot (quantil amostral vs. quantil ajustado) e a função de densidade de probabilidade ajustada para o Scan Circular usando os modelos de Gumbel e Weibull podem ser vistos nas Figuras 4.5 e 4.6, respectivamente. Um qq-plot indica um bom ajuste quando os quantis amostrais e ajustados coincidem, ou seja, quando o gráfico se aproxima da reta $y = x$, pelo que em todos os qq-plots essa reta foi traçada para ser tomada como referência.

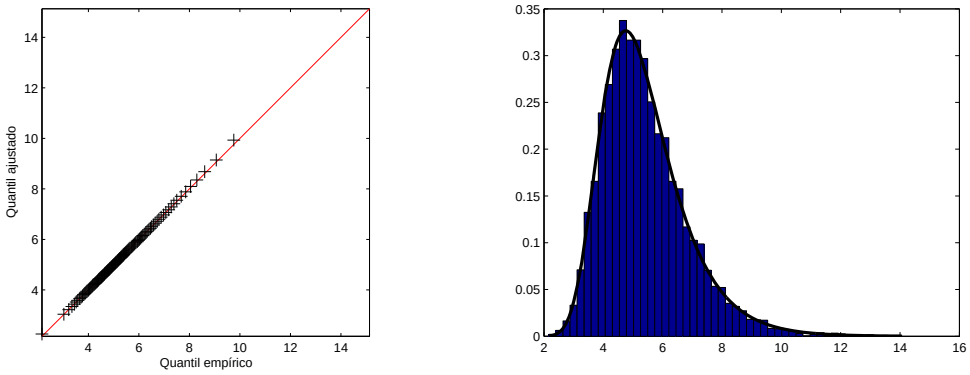



**Figura 4.5.:** qq-plot e histograma com o modelo de Gumbel ajustado para os dados do Scan Circular.

O qq-plot e a função de densidade de probabilidade ajustada para o AG usando os modelos de Gumbel e Weibull podem ser vistos nas Figuras 4.7 e 4.8, respectivamente.

Pelos qq-plots é possível notar que a distribuição de Gumbel adere bem aos dados do *Scan* circular, mas não aos dados do AG, ao passo que o modelo Weibull adere bem a ambos os conjuntos de dados. Essas observações podem ser comprovadas pelos $p$-valores obtidos para o teste de qualidade de ajuste de Kolmogorov-Smirnov, apresentados na Tabela 4.1. Pelo teste ambos os modelos se mostraram apropriados para os dados do

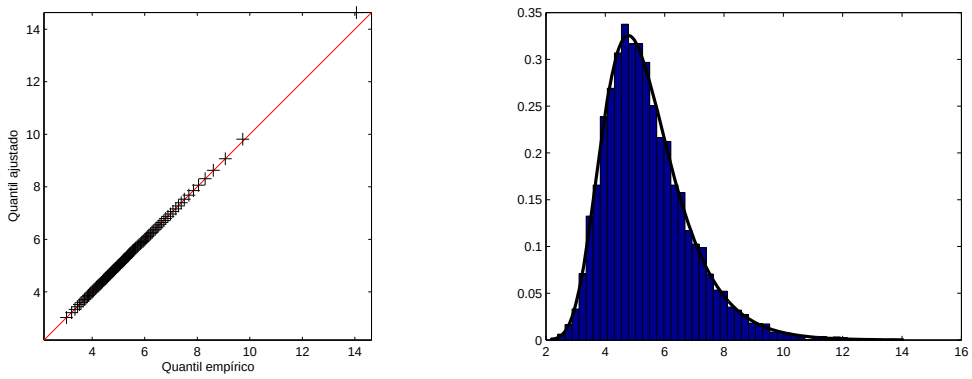



**Figura 4.6.:** qq-plot e histograma com o modelo de Weibull ajustado para os dados do Scan Circular.

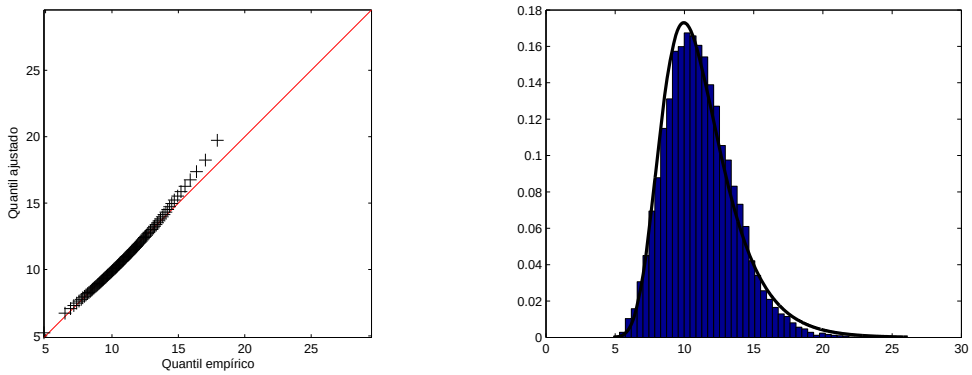



**Figura 4.7.:** qq-plot e histograma com o modelo de Gumbel ajustado para os dados do AG.

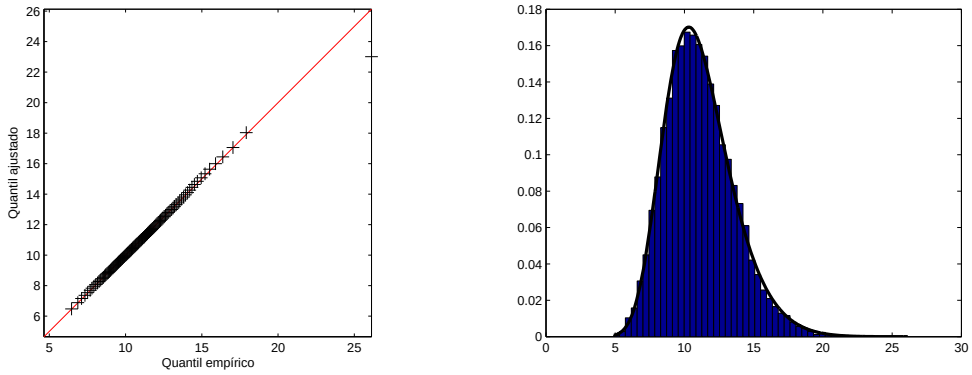



**Figura 4.8.:** qq-plot e histograma com o modelo de Weibull ajustado para os dados do AG.

Scan Circular. Porém, o modelo de Gumbel é rejeitado para os dados do AG. O fato de o modelo Gumbel apresentar um bom ajuste para os dados do *Scan* circular, mas não para os dados do AG, não está claro. Uma diferença entre os dois modelos é que a distribuição de Gumbel possui suporte infinito, ao passo que a distribuição de Weibull está definida em um domínio limitado inferiormente por zero e superiormente pelo parâmetro $\alpha$. Essa característica pode estar relacionada com o fato de que o *Scan* circular desempenha uma otimização exata, enquanto o AG encontra soluções sub-ótimas. Esses são fatores que podem influenciar a forma da distribuição, mas um estudo deve ser conduzido para se chegar a alguma conclusão.

**Tabela 4.1.:** $p$-valores para o teste Kolmogorov-Smirnov.

| Algoritmo | Weibull | Gumbel |
|---|---|---|
| Scan Circular | 0,643 | 0,472 |
| AG | 0,389 | $\mathbf{< 10^{-3}}$ |

### 4.4.2. Caso multiobjetivo

Para o AG multiobjetivo a compacidade de um cluster candidato envolve o cálculo de seu perímetro. Aqui utilizamos a aproximação desse perímetro pelo fecho convexo, como foi proposto em Duczmal *et al.* (2006) e um cálculo exato do perímetro, tendo em conta a disponibilidade dos dados apropriados. As Figuras 4.9 e 4.10 apresentam, respectivamente, os qq-plots obtidos para os modelos Weibull e Gumbel para a distribuição da $LLR$ em diferentes valores de $K$, usando a abordagem do fecho convexo para o cálculo do perímetro. Esses qq-plots, bem como os que se seguem, apresentam os quantis empíricos no eixo-$x$ e os quantis ajustados no eixo-$y$. Como no caso mono-objetivo, o modelo de Gumbel parece não apresentar bons resultados, uma vez que os qq-plots apresentam um viés que afasta o gráfico da linha $x = y$ usada como referência, o que não acontece para os ajustes utilizando a distribuição de Weibull.

A Tabela 4.2 apresenta os $p$-valores para o teste de Kolmogorov-Smirnov para ambos os modelos ajustados. $p$-valores em negrito indicam modelos rejeitados pelo teste. A tabela corrobora os resultados apresentados nos qq-plots, rejeitando o modelo Gumbel na maioria das vezes. O modelo Weibull só é rejeitado para $K = 1$.

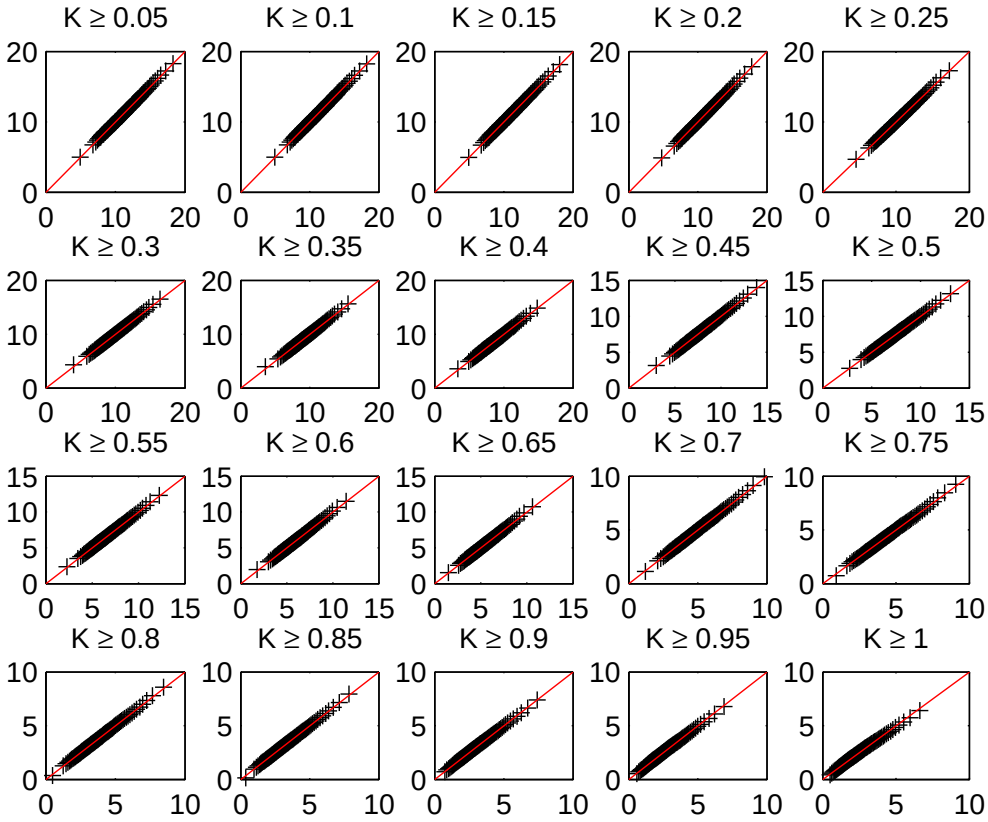



**Figura 4.9.:** qq-plots para o modelo Weibull em valores diferentes de $K$, usando a aproximação pelo fecho convexo.

As Figuras 4.12 e 4.13 apresentam, respectivamente, os qq-plots obtidos para os modelos Weibull e Gumbel para diferentes valores de $K$ usando a abordagem de fronteiras comuns para o cálculo do perímetro. Pode-se notar que os qq-plots obtidos para a distribuição de Gumbel apresentam um viés, como no caso do fecho convexo, enquanto os qq-plots relativos ao modelo Weibull mostram maior aderência deste modelo aos dados.

A Tabela 4.3 apresenta os $p$-valores para o teste de Kolmogorov-Smirnov para modelos ajustados usando dados cujos valores de $K(z)$ foram calculados usando as fronteiras comuns entre as regiões. $p$-valores em negrito indicam modelos rejeitados. Novamente o modelo de Gumbel não se mostra adequado, sendo rejeitado para a grande maioria de valores de $K$ escolhidos. O modelo Weibull, embora não tenha se saído tão bem quanto

**Tabela 4.2.:** $p$-valores dados pelo teste Kolmogorov-Smirnov usando aproximação pelo fecho convexo.

| K(z) | Weibull | Gumbel |
|---|---|---|
| 0,05 | 0,313 | $< \mathbf{10^{-3}}$ |
| 0,10 | 0,381 | $< \mathbf{10^{-3}}$ |
| 0,15 | 0,289 | $< \mathbf{10^{-3}}$ |
| 0,20 | 0,375 | $< \mathbf{10^{-3}}$ |
| 0,25 | 0,520 | $< \mathbf{10^{-3}}$ |
| 0,30 | 0,960 | $< \mathbf{10^{-3}}$ |
| 0,35 | 0,852 | $< \mathbf{10^{-3}}$ |
| 0,40 | 0,403 | $< \mathbf{10^{-3}}$ |
| 0,45 | 0,658 | $< \mathbf{10^{-3}}$ |
| 0,50 | 0,623 | **0,01** |
| 0,55 | 0,946 | 0,081 |
| 0,60 | 0,287 | 0,087 |
| 0,65 | 0,340 | **0,042** |
| 0,70 | 0,897 | 0,102 |
| 0,75 | 0,815 | $\mathbf{0,021}$ |
| 0,80 | 0,189 | $\mathbf{0,003}$ |
| 0,85 | 0,326 | $\mathbf{0,010}$ |
| 0,90 | 0,174 | $< \mathbf{10^{-3}}$ |
| 0,95 | 0,100 | $< \mathbf{10^{-3}}$ |
| 1,00 | $< \mathbf{10^{-3}}$ | $< \mathbf{10^{-3}}$ |

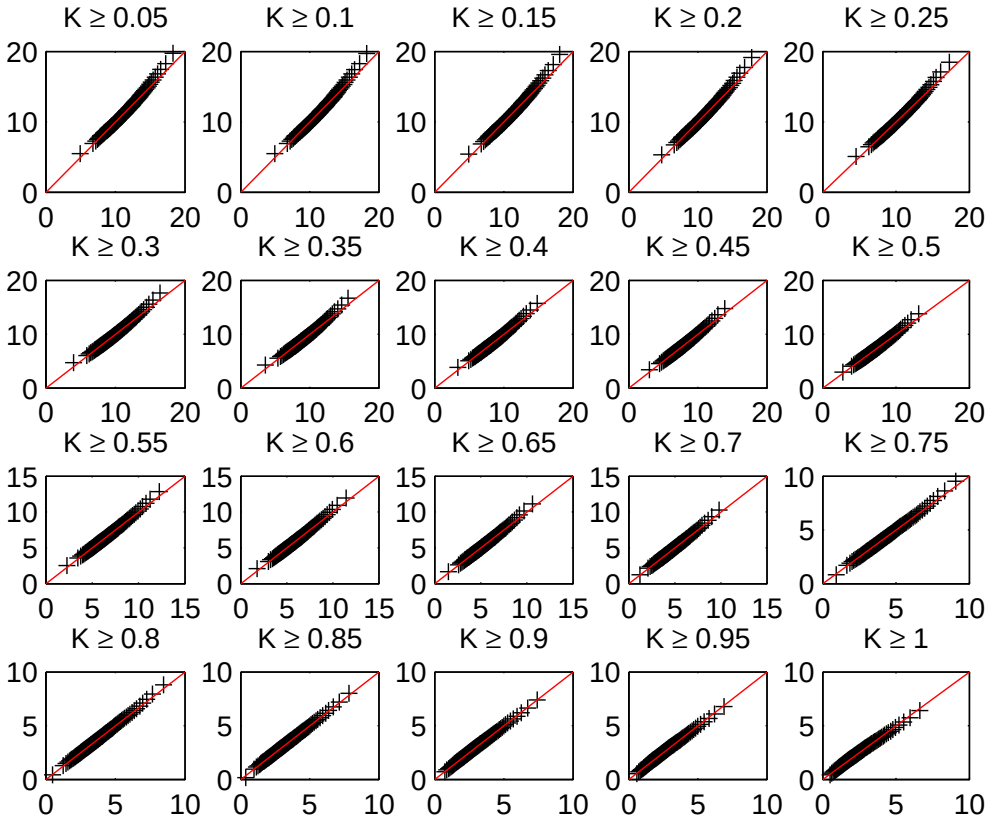
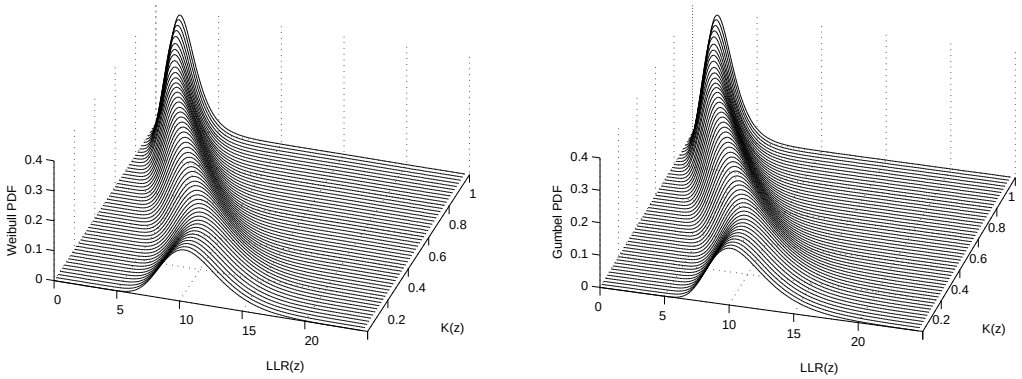



**Figura 4.10.:** qq-plots para o modelo Gumbel em valores diferentes de $K$, usando a aproximação pelo fecho convexo.




(a) (b)

**Figura 4.11.:** Modelos Weibull (a) e Gumbel (b) ajustados para valores de $K(z)$ fixos calculados usando a aproximação por fecho convexo.

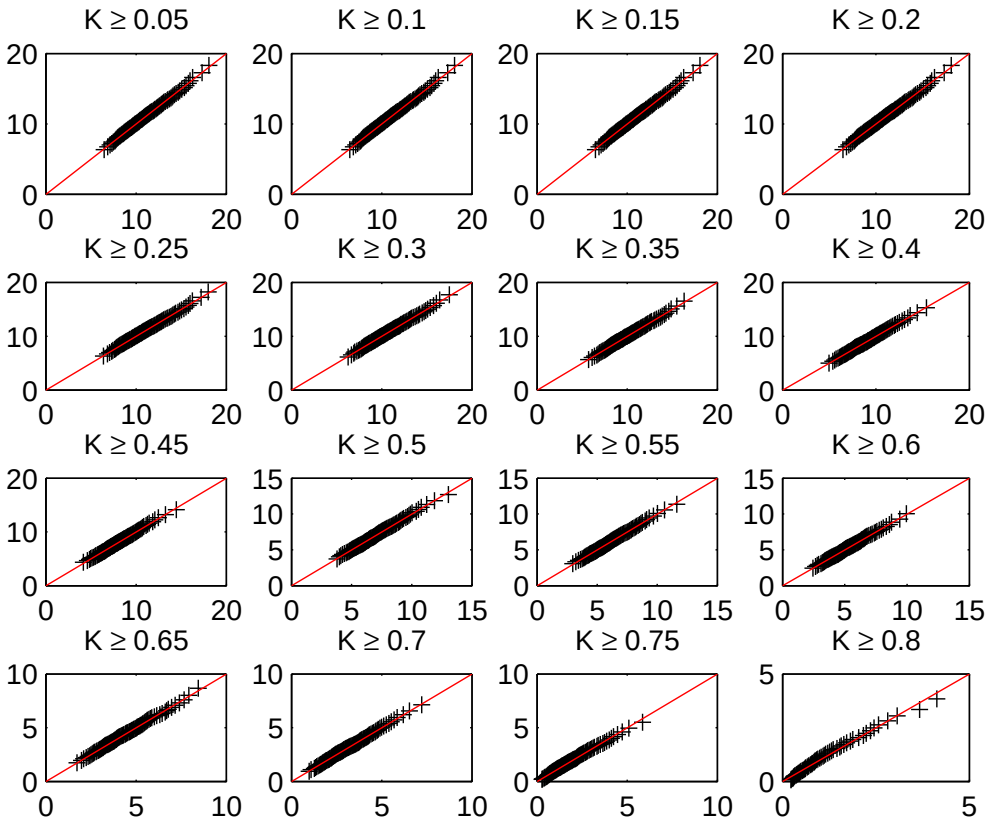



**Figura 4.12.:** qq-plots para o modelo Weibull para valores diferentes de $K$, usando aproximação por fronteiras comuns.

no caso da aproximação por fecho convexo, se mostra mais adequado que o Gumbel. Os $p$-valores são compatíveis com as observações dos qq-plots.

Em relação ao uso da função de aproveitamento, podemos destacar algumas vantagens em relação ao trabalho anterior. Aqui não há a necessidade da aproximação das distribuições condicionais através do uso de intervalos de largura positiva. Podemos ajustar a distribuição condicional $F(LLR|K \geq k)$ para qualquer valor de $K$ atingido ao longo das simulações. Além disso não há nenhuma perda de informação ou de estrutura dos conjuntos de Pareto, isto é, tratamos os conjuntos como conjuntos e não como pontos independentes. E por último, o conceito de probabilidade de se atingir uma solução tem uma base muito intuitiva e está bem definido.

**Tabela 4.3.:** $p$-valores dados pelo teste Kolmogorov-Smirnov usando a aproximação por fronteiras comuns.

| $K(z)$ | Weibull | Gumbel |
|---|---|---|
| 0,05 | 0,984 | $< \mathbf{10^{-3}}$ |
| 0,10 | 0,960 | $< \mathbf{10^{-3}}$ |
| 0,15 | 0,849 | $< \mathbf{10^{-3}}$ |
| 0,20 | 0,180 | **0,020** |
| 0,25 | 0,746 | **0,020** |
| 0,30 | 0,407 | 0,078 |
| 0,35 | 0,432 | **0,007** |
| 0,40 | 0,578 | **0,005** |
| 0,45 | 0,491 | **0,009** |
| 0,50 | 0,424 | 0,062 |
| 0,55 | $< \mathbf{10^{-3}}$ | $< \mathbf{10^{-3}}$ |
| 0,60 | $< \mathbf{10^{-3}}$ | $< \mathbf{10^{-3}}$ |
| 0,65 | $< \mathbf{10^{-3}}$ | $< \mathbf{10^{-3}}$ |
| 0,70 | $< \mathbf{10^{-3}}$ | $< \mathbf{10^{-3}}$ |
| 0,75 | 0,798 | $< \mathbf{10^{-3}}$ |
| 0,80 | 0,694 | $< \mathbf{10^{-3}}$ |

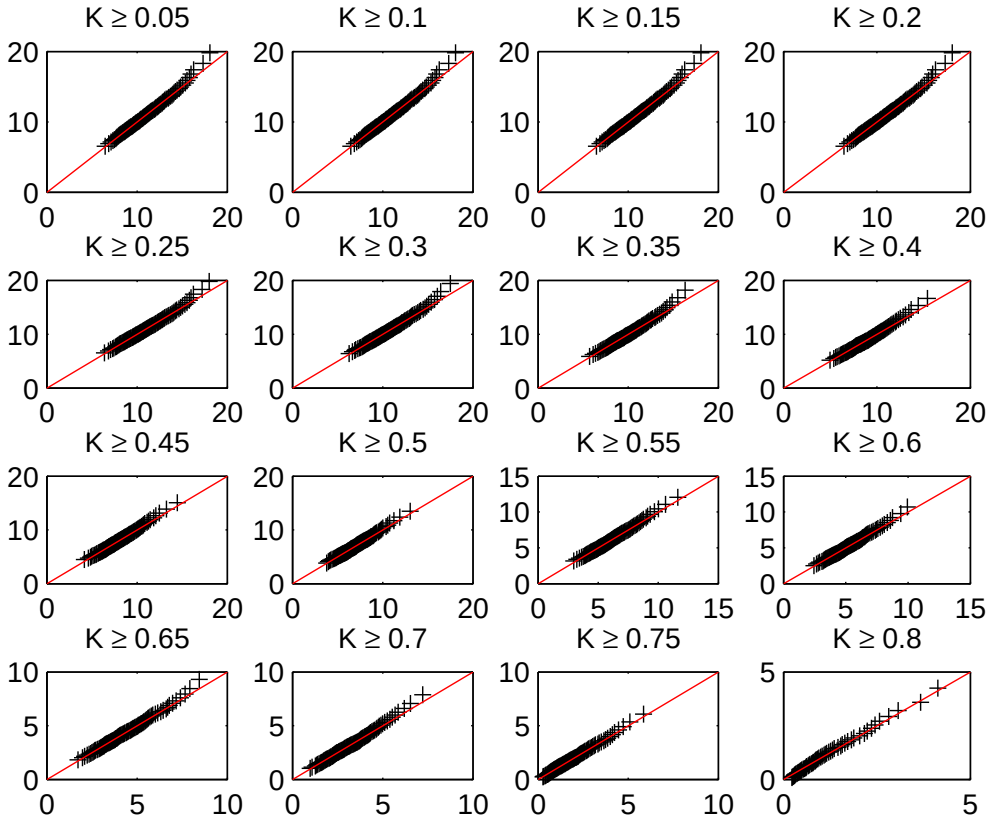
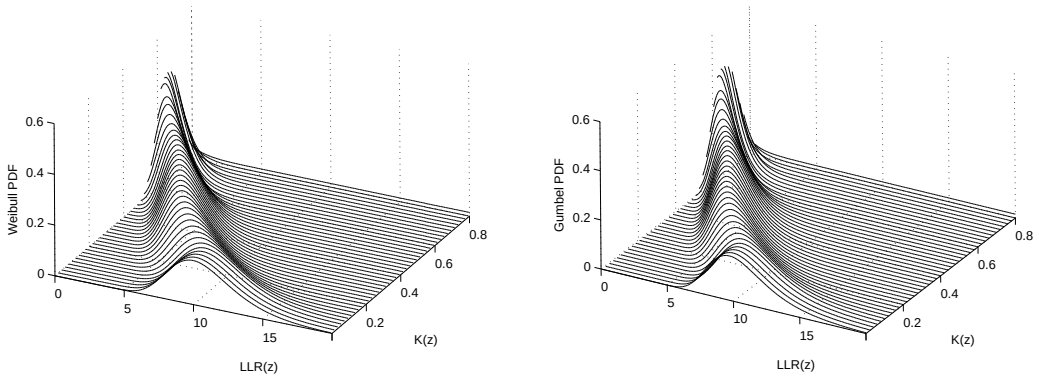



**Figura 4.13.:** qq-plots para o modelo Gumbel para valores diferentes de $K$, usando aproximação por fronteiras comuns.




(a) (b)

**Figura 4.14.:** Modelos Weibull (a) e Gumbel (b) ajustados para valores de $K(z)$ fixos computados usando aproximação por fronteiras comuns.

Quanto aos modelos paramétricos investigados devemos dizer que o modelo Gumbel é mais simples, já que envolve apenas dois parâmetros e as estimações se dão de forma simples. O modelo Weibull reverso é mais complexo, envolvendo a estimação de três parâmetros, sendo que a estimação de um deles (parâmetro de locação) é particularmente mais difícil. A estimação do parâmetro $\alpha$ mostrou-se um procedimento sensível. Por isso o modelo Gumbel deve ser preferido, sempre que possível. No entanto, o modelo Weibull se mostrou sistematicamente mais adequado para os dados obtidos pelo AG em suas versões mono e multiobjetivo.

## 4.5. Avaliação do poder

Uma característica desejável em um bom método de detecção é que ele seja sensível o suficiente para detectar um *cluster* quando este realmente existe. Uma maneira de avaliar a eficiência do algoritmo proposto nesta tese é calculando seu *poder* de detecção.

**Definição 6 (Poder do teste)** *O poder de um teste de hipóteses é definido como a probabilidade de que a hipótese nula seja rejeitada quando esta é, de fato, falsa.*

Pela definição acima, o poder do método será dado pela probabilidade de que ele detecte um *cluster* (ou seja, quando encontrar uma zona tal que $T_{obs} > T_{crit}$) quando este realmente existe. Podemos estimar o poder através de simulações de Monte Carlo, executando o algoritmo um número $n_{sim}$ de vezes em cenários artificiais, construídos de forma que sabemos que neles há a presença de um *cluster*. Assim basta fazer a contagem das $n_{detec}$ vezes em que um *cluster* foi detectado no mapa para estimar a probabilidade desejada, que será dada pela proporção $n_{detec}/n_{sim}$ de vezes que o *cluster* foi detectado em relação ao número de tentativas. Este será o poder estimado. No caso do algoritmo multiobjetivo, podemos fazer um procedimento parecido. A proporção de pontos (em relação ao número total de pontos obtidos em todas as simulações sob $h_a$) que estão à direita da isolinha de $p$-valor 0,05, a superfície crítica, nos dará o **poder médio** do algoritmo. A Figura 4.15 mostra a superfície crítica obtida pelos três métodos descritos na seção 4.2.

As superfícies obtidas pela função de aproveitamento e pelas faixas praticamente coincidem na parte inferior do gráfico, diferindo mais significantemente apenas para

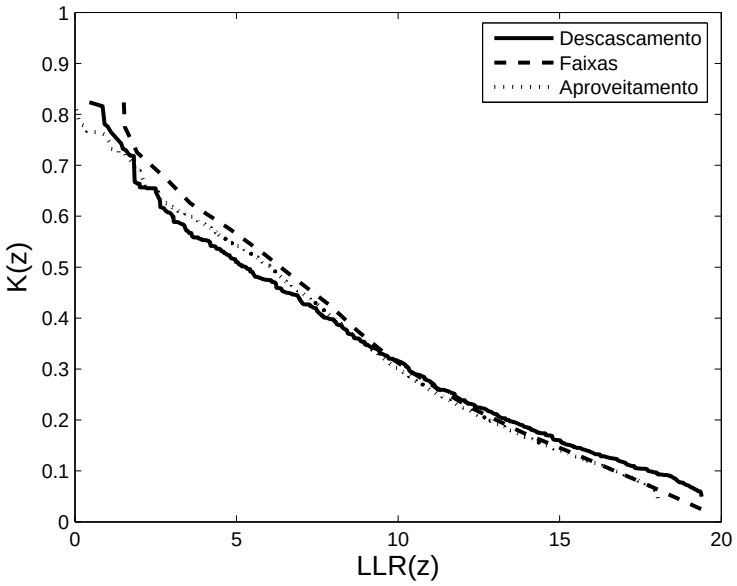



**Figura 4.15.:** Superfície crítica encontrada pelas técnicas de descascamento, faixas e função de aproveitamento.

altos valores de $K$. Já a superfície obtida pelo descascamento apresenta uma inclinação diferente das demais e tem um comportamento mais instável, principalmente para valores maiores de $K$. Pelas vantagens já citadas anteriormente, utilizaremos a superfície crítica obtida pela função de aproveitamento.

Além do poder, estimamos também a sensibilidade e o valor preditivo positivo (VPP) do algoritmo, que ajudam a medir a qualidade dos clusters detectados. Essas medidas são definidas a partir do seguintes eventos:

$V$={o indivíduo pertence ao *cluster* verdadeiro}
$D$={o indivíduo é detectado como pertencente ao *cluster* verdadeiro}

Com os eventos $V$ e $D$ podemos definir:

$$\text{Sensibilidade} = P(D|V) = \frac{P(D \cap V)}{P(V)}$$

$$= \frac{\text{População}(\textit{Cluster}\text{ Detectado} \cap \textit{Cluster}\text{ Verdadeiro})}{\text{População}(\textit{Cluster}\text{ Verdadeiro})}$$

$$\text{VPP} \quad = P(V|D) = \frac{P(D \cap V)}{P(D)}$$

$$= \frac{\text{População}(\textit{Cluster} \text{ Detectado} \cap \textit{Cluster} \text{ Verdadeiro})}{\text{População}(\textit{Cluster} \text{ Detectado})}$$

Para estimarmos o poder, sensibilidade e VPP do algoritmo multiobjetivo, utilizamos os 9 *clusters* artificiais no mapa do nordeste dos Estados Unidos, apresentados na Figura 4.16. Os seis primeiros, rotulados de $A$ a $F$ foram utilizados no trabalho Duczmal *et al.* (2006). Os outros três, rotulados de BOS, NYC e WAS, correspondem a regiões próximas às cidades de Boston, Nova Iorque e Washington, respectivamente. Em cada cenário foi fixado um risco relativo esperado igual para todas as regiões que não fazem parte do *cluster*, e um risco relativo $r > 1$ para as regiões que constituem o *cluster*. O valor de $r$ é escolhido de forma que se tenha uma probabilidade de 0,999 de que em cada distribuição aleatória se forme um *cluster* exatamente nas regiões com risco $r$, ou seja, $P(T_v > T_{crit}) = 0,999$, onde $T_v$ é o valor da estatística $T$ para o *cluster* verdadeiro (formado pelas regiões com risco $r$) (Kulldorff *et al.*, 2003). A estimação do poder foi baseada em 5.000 distribuições aleatórias de acordo com esses riscos, o que correponde a 5.000 *clusters* mais verossímeis para o algoritmo genético (AG) e 5.000 conjuntos de Pareto para o algoritmo genético multiobjetivo (AGM), para cada hipótese alternativa. O cálculo do valor crítico e da curva crítica foi baseado em outras 5.000 execuções dos AG's mono e multiobjetivo, respectivamente, sob distribuições aleatórias de casos sob a hipótese nula. O parâmetro de força da penalização geométrica $a$ foi fixado em 1. A Tabela 4.4 exibe a compacidade e o tamanho de cada *cluster*, bem como o poder, a sensibilidade e o VPP. estimados para cada método.

Na Figura 4.17 é possível ver a nuvem de pontos obtidos sob cada hipótese alternativa, bem como a superfície de aproveitamento crítica 95%. A proporção de pontos que estão à direita e acima da curva corresponde ao poder de detecção.

A Tabela 4.4 compara o poder de detecção dos algoritmos mono e multiobjetivo para os clusters artificiais utilizados, bem como a sensibilidade e o VPP. Pela Tabela 4.4 podemos concluir que:

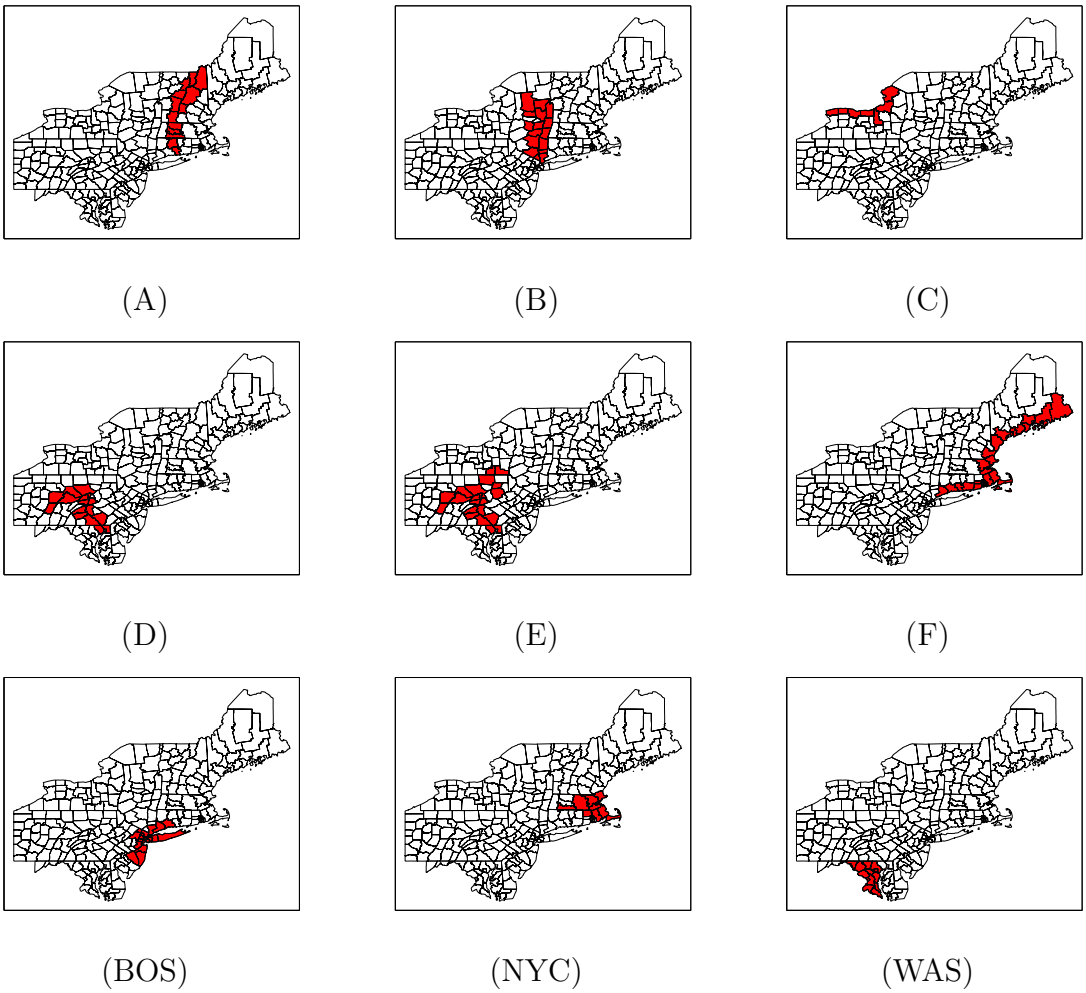



**Figura 4.16.:** *Clusters* artificiais $A - F$, BOS, NYC e WAS utilizados para estimar do poder do método.

- O método AGM supera o AG em termos de poder na maioria dos *clusters* (7 em 9). No entanto, levando em conta o erro associado às estimativas diríamos que o poder dos métodos é compatível.
- Assim como a análise de poder, em termos de sensibilidade os métodos se mostram semelhantes.
- O AGM tem um desempenho indiscutivelmente superior ao AG em termos de VPP, sendo que o VPP estimado para o AGM supera o do AG em todos os 9 cenários, sendo que a maior diferença se dá no *cluster* $B$ (aumento de 85,7%) e a menor no *cluster* $BOS$ (aumento de 15,9%).

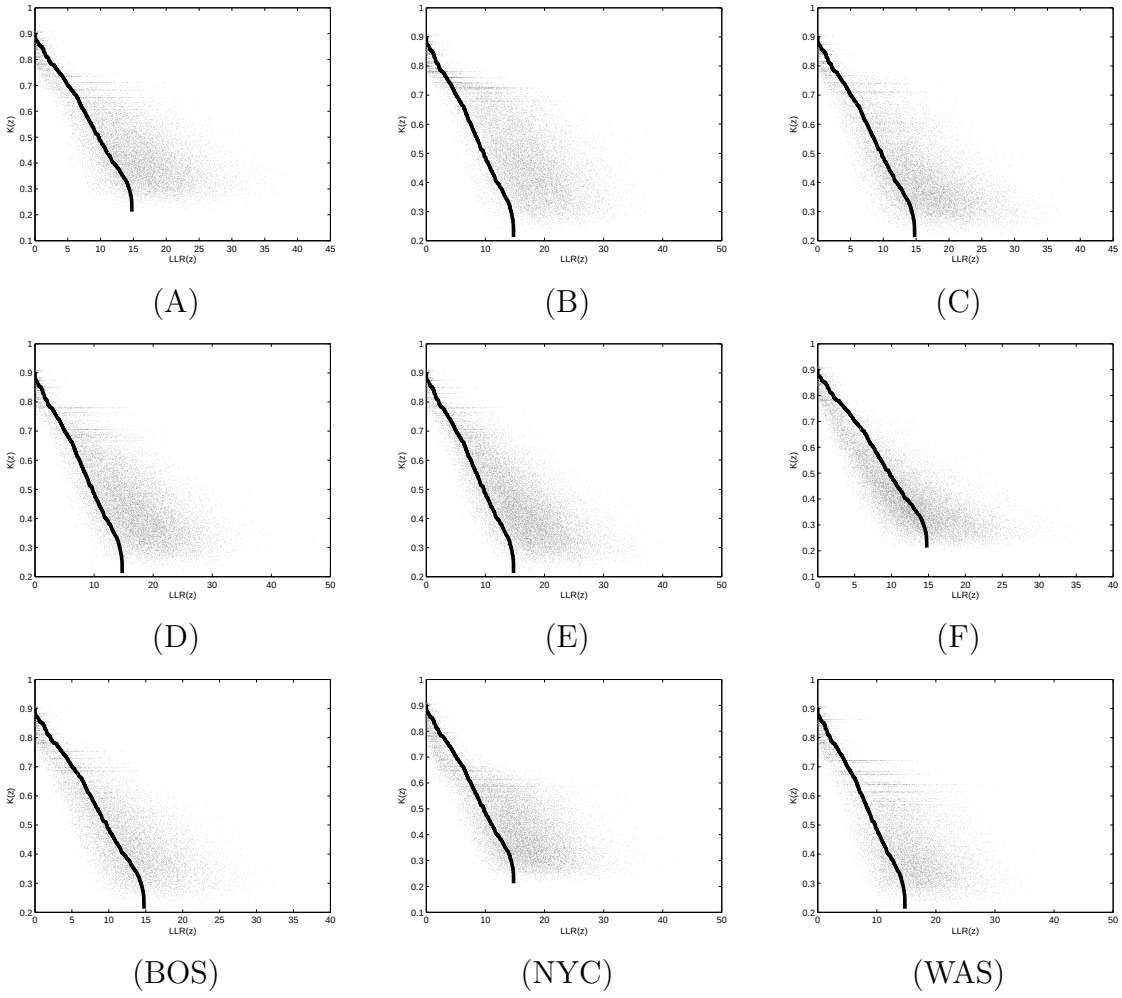



**Figura 4.17.:** Poder para os *clusters* $A - F$, BOS, NYC e WAS.

O aumento do valor preditivo positivo no algoritmo bi-objetivo sugere que este método, em média, superestima menos o *cluster* verdadeiro, isto é, não inclui na(s) solução(ões) muitas regiões que na verdade não estão na solução verdadeira. Esse é um bom sinal, já que os métodos de detecção de *clusters* com geometria arbitrária tendem, em sua maioria, a nos apresentar soluções superestimadas.

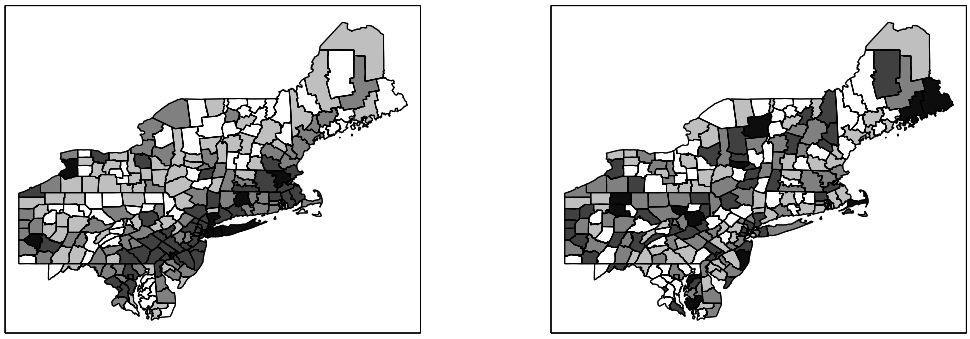
**Tabela 4.4.:** Poder, sensibilidade e vpp do algoritmo genético (AG) e do algoritmo genético multiobjetivo (AGM) estimados para os *clusters* artificiais.

| **Cluster** | **Tam** | **K** | **Poder** | | **Sensibilidade** | | **VPP** | |
|---|---|---|---|---|---|---|---|---|
| | | | **AG** | **AGM** | **AG** | **AGM** | **AG** | **AGM** |
| **A** | 13 | 0,38 | 0,85 | 0,83 | 0,84 | 0,86 | 0,51 | 0,81 |
| **B** | 16 | 0,50 | 0,83 | 0,86 | 0,78 | 0,83 | 0,49 | 0,91 |
| **C** | 7 | 0,35 | 0,79 | 0,87 | 0,89 | 0,83 | 0,44 | 0,77 |
| **D** | 15 | 0,39 | 0,88 | 0,91 | 0,76 | 0,76 | 0,54 | 0,83 |
| **E** | 21 | 0,31 | 0,80 | 0,89 | 0,67 | 0,61 | 0,60 | 0,83 |
| **F** | 25 | 0,17 | 0,46 | 0,50 | 0,69 | 0,65 | 0,60 | 0,76 |
| **BOS** | 20 | 0,29 | 0,70 | 0,78 | 0,68 | 0,54 | 0,82 | 0,95 |
| **NYC** | 10 | 0,36 | 0,70 | 0,60 | 0,80 | 0,85 | 0,66 | 0,91 |
| **WAS** | 13 | 0,37 | 0,69 | 0,76 | 0,78 | 0,89 | 0,66 | 0,98 |

# Capítulo 5.

# Aplicação

Como exemplo de aplicação vamos utilizar dados de casos de câncer de mama no mapa do nordeste dos Estados Unidos. O mapa está dividido em 245 regiões com população de risco (mulheres) total 29.535.210 e com um total de 58.943 casos no período de 1988 a 1992 (Kulldorff *et al.*, 1997). A Figura 5.1(a) mostra o mapa de população, com as regiões escuras representando as mais populosas e a mais clara as menos populosas. Na Figura 5.1(b) temos o mapa de incidências, com as regiões mais escuras tendo incidências mais altas.



(a) (b)

**Figura 5.1.:** (a) Mapa de população e (b) de incidência de casos de câncer de mama no nordeste dos Estados Unidos.

A Figura 5.2 mostra o conjunto Pareto-ótimo obtido pelo AG biobjetivo, constituído por 69 soluções que variam da mais irregular (com menor valor de $LLR$) até a mais compacta (com menor valor de $LLR$). Um resumo dessas soluções pode ser encontrado na

tabela 5.1, na qual encontram-se os valores de $LLR$ e $K$, a população e o número de casos, a qualidade do ajuste dos modelos (medida pelo $p$-valor do teste de Kolmogorov-Smirnov, aqui denotados por $k_W$ e $k_G$, para os modelos Weibull e Gumbel, respectivamente) e o $p$-valor da solução dada por cada modelo (denotados por $p_W$ e $p_G$). O modelo Weibull, como já era de se esperar, resulta em melhor ajuste. Isto pode ser constatado comparando as colunas $k_W$ e $k_G$, e observando que o modelo Weibull apresenta $p$-valores sistematicamente maiores, indicando a maior aderência desse modelo aos dados. No entanto, ambos os modelos apresentam resultados qualitativamente semelhantes quanto à significância. Ambos os modelos indicam que as soluções de maior valor de $LLR$ são mais significantes, embora os $p$-valores dos dois modelos não coincidam. Ainda que esses $p$-valores tão baixos possam estar influenciados por erros muito grandes de estimação, em termos de valores absolutos eles são extremamete úteis porque permitem, no mínimo, uma comparação qualitativa entre as soluções. Com essa análise, o que estamos fazendo é comparar a inclinação do conjunto Pareto-ótimo com a inclinação das "isolinhas de $p$-valor" (veja Figura 5.3). Essas isolinhas são obtidas ligando-se os pontos de mesmo $p$-valor para diferentes valores de $K$. Note que a inclinação do conjunto Pareto-ótimo confere um comportamento global tal que, à medida que o valor de $LLR$ aumenta, o conjunto atravessa as isolinhas no sentido da região de menor $p$-valor.

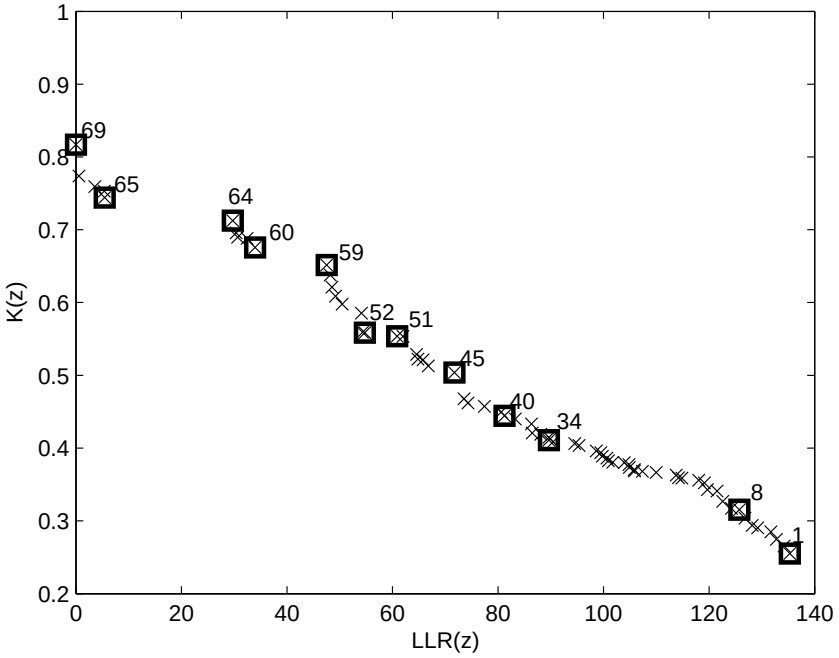



**Figura 5.2.:** Conjunto Pareto-ótimo encontrado para os casos de câncer de mama do Nordeste dos EUA.

**Tabela 5.1.:** Resumo dos clusters para os casos de câncer de mama do Nordeste dos EUA.

| Cluster | LLR | K | Pop | Casos | $k_W$ | $k_G$ | $p_W$ | $p_G$ |
|---|---|---|---|---|---|---|---|---|
| 1 | 135,29 | 0,26 | 4794221 | 11070 | 0,99 | 0,66 | 3,93e-37 | 3,42e-26 |
| 2 | 134,21 | 0,27 | 4827375 | 11134 | 0,98 | 0,65 | 7,28e-37 | 4,08e-26 |
| 3 | 132,74 | 0,27 | 4968126 | 11423 | 0,96 | 0,75 | 2,78e-36 | 7,45e-26 |
| 4 | 131,70 | 0,29 | 5001280 | 11487 | 0,99 | 0,65 | 4,96e-36 | 8,96e-26 |
| 5 | 129,19 | 0,29 | 4917260 | 11295 | 0,98 | 0,62 | 4,66e-35 | 2,48e-25 |
| 6 | 128,17 | 0,29 | 4950414 | 11359 | 0,98 | 0,64 | 1,06e-34 | 3,57e-25 |
| 7 | 126,81 | 0,30 | 4540741 | 10485 | 0,99 | 0,69 | 2,03e-34 | 4,32e-25 |
| 8 | 125,72 | 0,32 | 4999024 | 11447 | 0,95 | 0,59 | 4,03e-35 | 4,26e-25 |
| 9 | 124,22 | 0,32 | 4589351 | 10573 | 0,93 | 0,61 | 1,35e-35 | 7,60e-25 |
| 10 | 122,60 | 0,33 | 4423311 | 10212 | 0,70 | 0,59 | 2,40e-35 | 7,61e-25 |
| 11 | 121,52 | 0,34 | 4456465 | 10276 | 0,88 | 0,26 | 1,72e-35 | 4,77e-25 |
| 12 | 119,69 | 0,34 | 4571639 | 10509 | 0,90 | 0,41 | 9,66e-35 | 1,00e-24 |
| 13 | 119,11 | 0,35 | 4372445 | 10084 | 0,87 | 0,53 | 4,76e-35 | 5,41e-25 |
| 14 | 118,04 | 0,36 | 4405599 | 10148 | 0,85 | 0,48 | 1,22e-34 | 8,11e-25 |
| 15 | 114,83 | 0,36 | 4255849 | 9812 | 0,84 | 0,20 | 2,24e-33 | 2,98e-24 |
| 16 | 114,22 | 0,36 | 4574300 | 10482 | 0,78 | 0,20 | 4,29e-33 | 4,10e-24 |
| 17 | 113,77 | 0,36 | 4289003 | 9876 | 0,72 | 0,20 | 4,85e-33 | 4,06e-24 |
| 18 | 109,96 | 0,37 | 4457704 | 10210 | 0,57 | 0,24 | 1,10e-31 | 1,99e-23 |
| 19 | 107,36 | 0,37 | 4951152 | 11233 | 0,70 | 0,26 | 1,16e-30 | 6,37e-23 |
| 20 | 105,86 | 0,37 | 4952381 | 11226 | 0,67 | 0,24 | 5,10e-30 | 1,39e-22 |
| 21 | 105,81 | 0,37 | 4633930 | 10556 | 0,71 | 0,34 | 4,57e-30 | 1,27e-22 |
| 22 | 104,89 | 0,37 | 4667084 | 10620 | 0,75 | 0,25 | 9,22e-30 | 1,79e-22 |
| 23 | 104,81 | 0,38 | 4868493 | 11043 | 0,75 | 0,27 | 8,46e-30 | 1,67e-22 |

Cont. Tabela 5.1

| Cluster | LLR | K | Pop | Casos | $k_W$ | $k_G$ | $p_W$ | $p_G$ |
|---|---|---|---|---|---|---|---|---|
| 24 | 103,93 | 0,38 | 4901647 | 11107 | 0,84 | 0,25 | 1,41e-29 | 2,06e-22 |
| 25 | 101,74 | 0,38 | 5037194 | 11377 | 0,85 | 0,27 | 1,03e-28 | 5,96e-22 |
| 26 | 100,91 | 0,38 | 5070348 | 11441 | 0,88 | 0,23 | 1,78e-28 | 7,73e-22 |
| 27 | 100,63 | 0,39 | 4751897 | 10771 | 0,51 | 0,21 | 1,35e-28 | 5,98e-22 |
| 28 | 99,77 | 0,39 | 4785051 | 10835 | 0,61 | 0,25 | 2,90e-28 | 9,11e-22 |
| 29 | 99,49 | 0,39 | 4816881 | 10900 | 0,75 | 0,38 | 2,03e-28 | 6,67e-22 |
| 30 | 98,64 | 0,40 | 4850035 | 10964 | 0,76 | 0,20 | 3,36e-28 | 8,48e-22 |
| 31 | 95,37 | 0,40 | 4700285 | 10628 | 0,91 | 0,29 | 6,56e-28 | 1,89e-21 |
| 32 | 94,53 | 0,41 | 4733439 | 10692 | 0,96 | 0,41 | 1,16e-27 | 2,47e-21 |
| 33 | 90,42 | 0,41 | 4823083 | 10852 | 0,90 | 0,43 | 4,62e-26 | 1,94e-20 |
| 34 | 89,64 | 0,41 | 4856237 | 10916 | 0,97 | 0,35 | 8,91e-26 | 2,82e-20 |
| 35 | 89,63 | 0,41 | 4194270 | 9526 | 0,83 | 0,46 | 4,65e-26 | 2,32e-20 |
| 36 | 88,09 | 0,42 | 5312929 | 11859 | 0,87 | 0,36 | 1,60e-25 | 4,61e-20 |
| 37 | 86,49 | 0,42 | 4362971 | 9860 | 0,92 | 0,31 | 3,57e-25 | 1,09e-19 |
| 38 | 86,34 | 0,43 | 4144765 | 9400 | 0,99 | 0,77 | 2,95e-26 | 3,84e-20 |
| 39 | 83,27 | 0,44 | 4313466 | 9734 | 0,99 | 0,68 | 2,81e-25 | 1,26e-19 |
| 40 | 81,23 | 0,44 | 2904862 | 6740 | 0,85 | 0,41 | 1,19e-24 | 2,59e-19 |
| 41 | 80,60 | 0,45 | 3791365 | 8617 | 0,84 | 0,39 | 1,09e-24 | 2,19e-19 |
| 42 | 77,43 | 0,46 | 3960066 | 8951 | 0,82 | 0,54 | 1,02e-23 | 7,02e-19 |
| 43 | 74,32 | 0,46 | 4471234 | 10001 | 0,89 | 0,60 | 1,23e-22 | 2,93e-18 |
| 44 | 73,57 | 0,47 | 4436109 | 9922 | 0,75 | 0,42 | 1,41e-22 | 2,92e-18 |
| 45 | 71,73 | 0,50 | 4270069 | 9561 | 0,76 | 0,24 | 4,06e-24 | 4,18e-19 |
| 46 | 66,78 | 0,51 | 4531615 | 10070 | 0,64 | 0,20 | 1,85e-22 | 4,16e-18 |
| 47 | 65,78 | 0,52 | 4648150 | 10305 | 0,88 | 0,24 | 1,57e-22 | 3,08e-18 |
| 48 | 64,83 | 0,52 | 4815622 | 10646 | 0,92 | 0,28 | 3,98e-22 | 5,46e-18 |

Cont. Tabela 5.1

| Cluster | LLR | K | Pop | Casos | $k_W$ | $k_G$ | $p_W$ | $p_G$ |
|---|---|---|---|---|---|---|---|---|
| 49 | 64,58 | 0,53 | 2306408 | 5361 | 0,97 | 0,56 | 1,99e-22 | 3,32e-18 |
| 50 | 62,01 | 0,55 | 4766117 | 10520 | 0,85 | 0,28 | 3,06e-22 | 5,48e-18 |
| 51 | 60,87 | 0,55 | 2189812 | 5089 | 0,89 | 0,32 | 1,05e-21 | 1,18e-17 |
| 52 | 54,78 | 0,56 | 1914041 | 4461 | 0,98 | 0,51 | 3,99e-19 | 5,11e-16 |
| 53 | 54,35 | 0,56 | 1925021 | 4482 | 0,95 | 0,53 | 5,63e-19 | 6,35e-16 |
| 54 | 54,11 | 0,59 | 1911705 | 4452 | 0,65 | 0,36 | 4,40e-20 | 1,34e-16 |
| 55 | 50,45 | 0,60 | 1957230 | 4527 | 0,83 | 0,28 | 9,09e-19 | 1,00e-15 |
| 56 | 49,21 | 0,61 | 2018053 | 4649 | 0,82 | 0,44 | 9,96e-19 | 1,16e-15 |
| 57 | 48,52 | 0,62 | 2021074 | 4651 | 0,99 | 0,71 | 5,63e-19 | 7,25e-16 |
| 58 | 48,20 | 0,64 | 1992519 | 4588 | 0,99 | 0,44 | 1,24e-19 | 4,55e-16 |
| 59 | 47,52 | 0,65 | 1995540 | 4590 | 0,78 | 0,69 | 4,62e-20 | 2,11e-16 |
| 60 | 33,96 | 0,68 | 3921863 | 8514 | 0,99 | 0,66 | 4,35e-14 | 3,76e-12 |
| 61 | 32,44 | 0,69 | 1228541 | 2852 | 0,78 | 0,31 | 1,02e-13 | 6,81e-12 |
| 62 | 30,67 | 0,69 | 1444987 | 3303 | 0,74 | 0,31 | 5,84e-13 | 2,33e-11 |
| 63 | 30,36 | 0,70 | 1485475 | 3387 | 0,86 | 0,33 | 6,28e-13 | 3,95e-11 |
| 64 | 29,74 | 0,71 | 1488496 | 3389 | 0,93 | 0,49 | 9,56e-13 | 5,33e-11 |
| 65 | 5,48 | 0,74 | 297621 | 676 | 0,57 | 0,58 | 0,07 | 0,07 |
| 66 | 5,37 | 0,75 | 275486 | 628 | 0,64 | 0,35 | 0,05 | 0,05 |
| 67 | 3,61 | 0,76 | 44808 | 116 | 0,68 | 0,14 | 0,07 | 0,07 |
| 68 | 0,59 | 0,77 | 163960 | 347 | 0,61 | 0,19 | 0,35 | 0,36 |
| 69 | 0,00 | 0,82 | 66294 | 97 | 0,98 | 0,83 | 0,34 | 0,34 |

As Figuras 5.4 e 5.5 mostram 12 clusters selecionados entre os 69. Comparando esses *clusters* com os *clusters* detectados em Kulldorff *et al.* (1997) percebemos que os clusters 1, 8, 34, 40, 45, 51 e 60 estão localizados na região do *cluster* primário detectado pelo *scan* circular, localizado no entorno da cidade de Nova Iorque, enquanto os *clusters* 52, 59 e

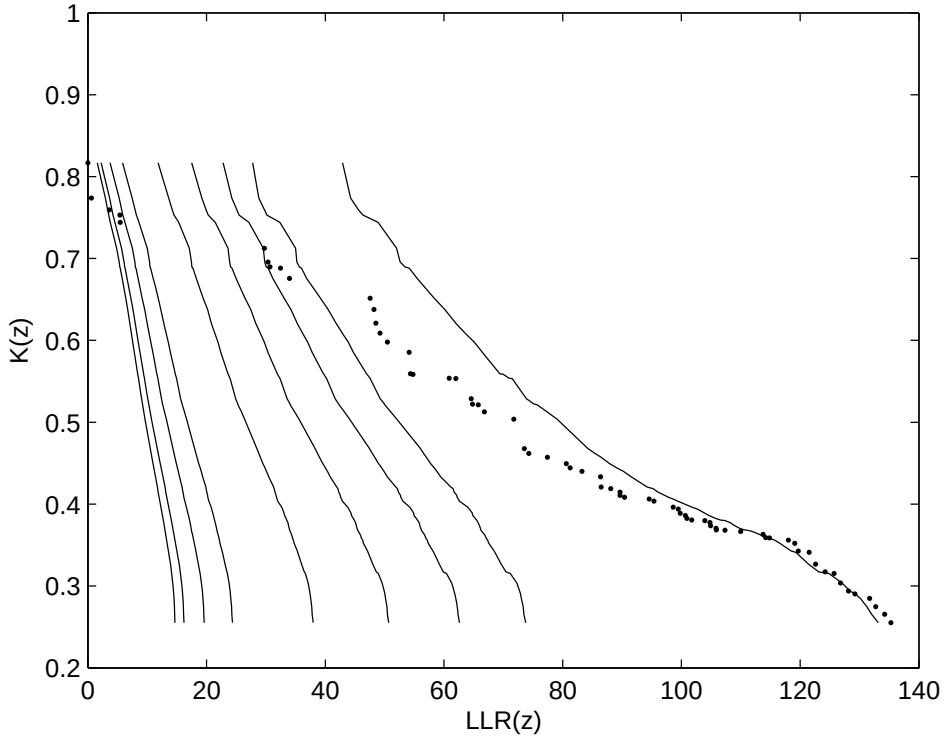



**Figura 5.3.:** Isolinhas de $p$-valor.

64 relacionam-se ao *cluster* secundário nas vizinhanças da cidade de Buffalo. Os *clusters* 65 e 69 não aparecem entre os *clusters* secundários. Isso é consistente com o $p$-valor dos *clusters* 65-69 na tabela 5.1. Esses *clusters* são espúrios pois aparecem na solução final apenas por apresentarem alto valor de compacidade, porém sem significância estatística.

A soluções de menor $p$-valor correspondem àquelas mais improváveis. No entanto, a decisão de qual deve ser a solução adotada ficará por conta de um especialista, que deverá levar em consideração os fatores que julgar adequados. De forma a complementar a Tabela 5.1, um gráfico com a superimposição dos *clusters* pode ser fornecido. Esse gráfico é feito de acordo com a frequência das regiões nas soluções significativas, sendo que as regiões mais frequentes aparecem em tons mais escuros e as menos frequentes em tons mais claros. Para as soluções da Tabela 5.1, esse gráfico é apresentado na Figura 5.6.

No mapa da Figura 5.6 as regiões mais escuras ocorrem com maior frequência nas soluções da Tabela 5.1, enquanto as mais claras ocorrem com menos frequência. As re-

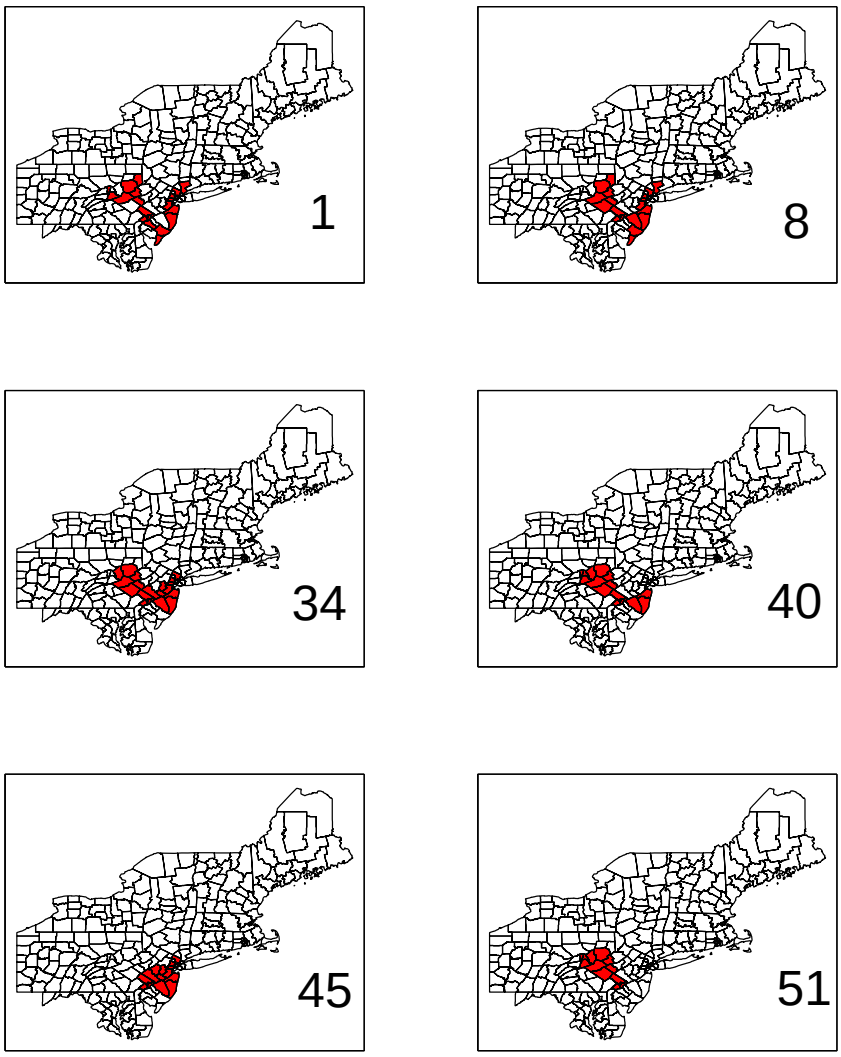



**Figura 5.4.:** Clusters detectados (1)

giões em branco não estão presentes em nenhuma solução. As soluções não-significativas estatisticamente (65-69) não contribuíram para a contagem de frequência.

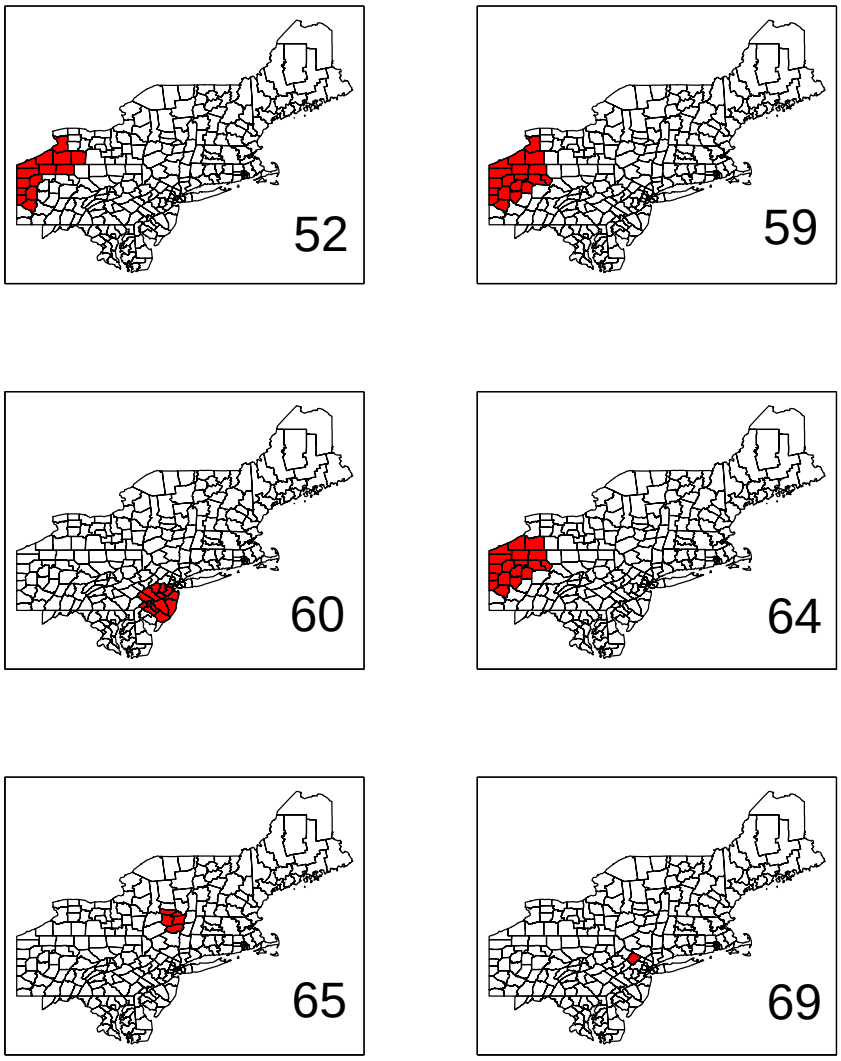



**Figura 5.5.:** Clusters detectados (2)

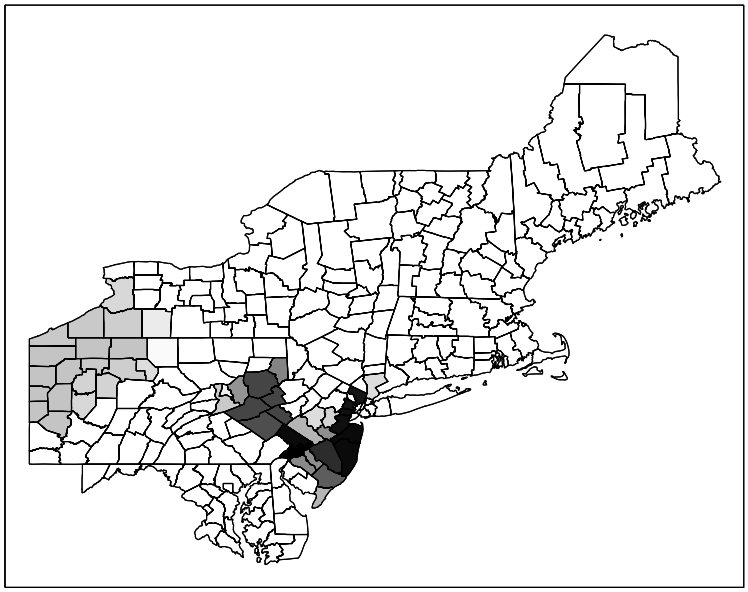


**Figura 5.6.:** Frequência de ocorrência nas soluções.

# Capítulo 6.

# Controlando o erro: Abordagem knapsack

O AG, como é típico dos algoritmos de otimização estocásticos, não garante que a solução encontrada seja ótima. Eventualmente poderemos encontrar soluções ótimas utilizando o AG, mas, mesmo que isso ocorra, não saberemos. Por esse motivo a distribuição obtida pela simulação de Monte Carlo sob a hipótese nula é enviesada, de modo que o valor crítico referente ao nível de significância adotado fica subestimado. Isso faz com que a rejeição da hipótese nula seja facilitada, aumentando a probabilidade do erro tipo I.

Alternativamente, iremos modelar o problema de maximização da $LLR$ e de $K$ como um problema de otimização combinatória conhecido como problema *knapsack* (problema da mochila) binário. O problema *knapsack* constitui-se em encontrar qual a combinação de itens que devem ser carregados na mochila que maximiza o valor total da escolha sujeita à capacidade da mochila. Sendo $x_i$ a variável que indica se o $i$-ésimo item está ou não na mochila, $v_i$ e $w_i$, respectivamente, o valor e o peso do $i$-ésimo item e $W$ a capacidade de carga total da mochila, o problema *knapsack* linear pode ser escrito na forma

$$\begin{aligned} \max_{x} \quad & \sum_i v_i x_i \\ \text{sujeito a} \quad & \sum_i w_i x_i \leq W \\ x_i \in \quad & \{0,1\} \end{aligned} \tag{6.1}$$

e o problema quadrático como

$$\begin{aligned} \max_{x} \quad & \sum_i v_i x_i + \sum_i \sum_{j \neq i} v_{ij} x_i x_j \\ \text{sujeito a} \quad & \sum_i w_i x_i \leq W \\ x_i \in \quad & \{0, 1\} \end{aligned} \tag{6.2}$$

Mostraremos que o problema de detecção de *clusters* pode ser modelado como um problema *knapsack* a menos da restrição de conexidade. Uma vez resolvido (exatamente) o problema *knapsack*, teremos a garantia de que a solução verdadeira do problema será menor que a solução do problema *knapsack* (ou, no máximo, igual). Isso nos permite ter uma cota superior para o problema que pode ser contrastada com a cota inferior dada pelo AG. Existem várias versões do problema *knapsack*, mas nesse texto trabalharemos com o problema clássico (binário, linear e com uma restrição de capacidade) e com uma versão quadrática, que será manipulada de forma a se chegar em uma versão que pode ser resolvida com as ferramentas disponíveis. Embora a fundamentação deste capítulo seja rigorosa, as simulações conduzidas até agora constituem apenas uma análise exploratória inicial e os resultados apresentados são preliminares.

## 6.1. Fundamentação

A fim de formular nosso problema como um problema *knapsack*, vamos assegurar que o conjunto das soluções obtidas para o problema nessa forma contenha o conjunto de soluções do problema original sem a restrição de conexidade. A função $LLR$ é obtida tomando-se o logaritmo na equação (2.3), e fica então:

$$LLR(z) = \begin{cases} c_z \log\left(\frac{c_z}{\mu_z}\right) + (C - c_z) \log\left(\frac{C - c_z}{C - \mu_z}\right) & \text{se } c_z > \mu_z \\ 0 & \text{caso contrário.} \end{cases} \tag{6.3}$$

Substituindo $\mu_z = Cp_z/P$ na equação 6.3, onde $P$ é a população total e $p_z$ é a população em $z$, a função $LLR$ pode ser reescrita como:

$$LLR(z) = \begin{cases} C \log\left(\frac{P}{C}\right) + c_z \log\left(\frac{c_z}{p_z}\right) + (C - c_z) \log\left(\frac{C-c_z}{P-pz}\right) & \text{se } \frac{c_z}{p_z} > \frac{C-c_z}{P-p_z} \\ 0 & \text{caso contrário.} \end{cases} \tag{6.4}$$

**Proposição 1** *Se* $\frac{c_z}{p_z} > \frac{C-c_z}{P-p_z}$ *a função LLR é estritamente crescente na variável* $c_z$ *e estritamente decrescente na variável* $p_z$*.*

Demonstração: Calculando as derivadas parciais:

$$\frac{\partial LLR}{\partial c_z} = \log\left(\frac{c_z}{p_z}\right) - \log\left(\frac{C - c_z}{P - p_z}\right) \tag{6.5}$$

$$\frac{\partial LLR}{\partial p_z} = \frac{C - c_z}{P - p_z} - \frac{c_z}{p_z} \tag{6.6}$$

Levando em conta que $\frac{c_z}{p_z} > \frac{C-c_z}{P-p_z}$ e que a função logaritmo é estritamente crescente, é fácil perceber que $\dfrac{\partial LLR}{\partial c_z} > 0$. Portanto a função $LLR$ é estritamente crescente na variável $c_z$. A demonstração é análoga para $p_z$. ■

Usaremos a proposição 1 para demonstrar a proposição 2 a seguir.

**Proposição 2** *Seja* $\mathscr{S}$ *o conjunto formado por todos os pares* $(\hat{c_z}, \hat{p_z})$ *tais que* $\nexists (c_z, p_z)$ *tal que* $c_z > \hat{c_z}$ *e* $p_z < \hat{p_z}$*. Se o máximo de LLR ocorre no ponto* $(c_z^*, p_z^*)$ *então* $(c_z^*, p_z^*) \in \mathscr{S}$*.*

Demonstração: Se $(c_z^*, p_z^*) \notin \mathscr{S}$, então existe um ponto $(c_z, p_z)$ tal que $c_z \geq c_z^*$ e $p_z \leq p_z^*$, com pelo menos uma das desigualdades sendo estrita. Se $c_z > c_z^*$ e $p_z = p_z^*$, e como a função $LLR$ é estritamente crescente em relação à variável $c_z$, temos

$$LLR(c_z, p_z) = LLR(c_z, p_z^*) > LLR(c_z^*, p_z^*)$$

Analogamente, se $p_z < p_z^*$ e $c_z = c_z^*$, e como a função $LLR$ é estritamente decrescente em relação a $p_z$

$$LLR(c_z, p_z) = LLR(c_z^*, p_z) > LLR(c_z^*, p_z^*)$$

Por outro lado, se $c_z > c_z^*$ e $p_z < p_z^*$

$$LLR(c_z, p_z) > LLR(c_z, p_z^*) > LLR(c_z^*, p_z^*)$$

Em todos os casos chegamos à conclusão de que $LLR(c_z, p_z) > LLR(c_z^*, p_z^*)$, o que é um absurdo, já que a suposição é de que $\max LLR(c_z, p_z) = LLR(c_z^*, p_z^*)$. Logo, $(c_z^*, p_z^*)$ tem que estar no conjunto $\mathscr{S}$. ■

Suponha que queiramos resolver o problema bi-objetivo ($\max_z c_z$, $\min_z p_z$). A proposição 2 nos garante que o conjunto de soluções não dominadas $\mathscr{S}$ contém a solução que maximiza a função $LLR$ irrestrita.

Podemos chegar a um resultado análogo para o caso da compacidade, considerando a compacidade $K(z) = 4\pi a_z/h_z^2$ de uma zona $z$, onde $a_z$ é a área e $h_z$ é o perímetro de $z$.

**Proposição 3** *Seja $\mathscr{T}$ o conjunto formado por todos os pares $(\hat{a_z}, \hat{h_z})$ tais que $\nexists (a_z, h_z)$ tal que $a_z > \hat{a_z}$ e $h_z < \hat{h_z}$. Se o máximo de $K$ ocorre no ponto $(a_z^*, h_z^*)$ então $(a_z^*, h_z^*) \in \mathscr{T}$.*

Demonstração: A demonstração é análoga à da proposição 2, uma vez que a função $K$ é estritamente crescente em relação à variável $a_z$ e estritamente decrescente em relação à variável $h_z$. ■

Vamos considerar os seguintes problemas:

Problema $A$: $max(LLR(z), K(z))$

Problema $B$: $max(c_z), min(p_z), max(a_z), min(h_z)$

**Proposição 4** *Seja $\mathscr{P}$ o conjunto de todas as soluções não dominadas do problema $A$ e $\mathscr{P}'$ o conjunto de todas as soluções não dominadas do problema $B$. Se $z^* \in \mathscr{P}$ então $z^* \in \mathscr{P}'$.*

Demonstração: A proposição é equivalente a dizer que se $z^* \notin \mathscr{P}'$ então $z^* \notin \mathscr{P}$. De fato, se fossem $c_z > c_{z^*}$ e $p_z < p_{z^*}$, então $LLR(z) > LLR(z^*)$; e se $a_z > a_{z^*}$ e $h_z < h_{z^*}$ então $K(z) > K(z^*)$. Logo, se $z^* \notin \mathscr{P}'$ então $z^* \notin \mathscr{P}$. ■

A proposição 4 garante que $\mathscr{P} \subseteq \mathscr{P}'$, isto é, se encontrarmos o conjunto de soluções não dominadas para o problema $B$, então teremos encontrado todas as soluções não dominadas para o problema $A$.

## 6.2. Formulação Knapsack

Para um mapa com $n$ regiões, considere as variáveis binárias $x_1$, ...,$x_n$, onde $x_i = 0$ se a $i$-ésima região não está presente no cluster e $x_i = 1$ se a $i$-ésima região está presente no cluster. Considerando ainda:

- $c_i$: casos na região $i$
- $p_i$: população na região $i$
- $a_i$: área da região $i$
- $h_i$: perímetro da região $i$
- $h_{ij}$: comprimento da fronteira entre as regiões $i$ e $j$

Para o caso em que se deseja apenas maximizar a função $LLR$ a proposição 2 nos permite derivar um algoritmo para o cálculo do máximo da função $LLR$ resolvendo a família de problemas

$$\begin{aligned} \max \quad & \sum_i c_i x_i \\ \text{sujeito a} \quad & \sum_{i=1}^{n} p_i x_i \leq P_k \end{aligned} \tag{6.7}$$

obtidos variando-se a restrição no número de casos da seguinte forma: no primeiro passo fazemos $C_0 = 0$ e obtemos uma solução $S_0$ cuja população é $p_0$ e cujo número de casos é $c_0$. A partir daí basta ir resolvendo os problemas obtidos fazendo-se $C_k = c_{k-1} + 1$, até que se atinja $C_k = C$. De posse desse conjunto de soluções, basta percorrê-lo avaliando o valor de $LLR$ de cada solução. O número de problemas que deve ser resolvido é $O(C)$ e, embora o problema *knapsack* seja NP-hard (ainda que um dos mais fáceis deles), cada problema pode ser resolvido de forma relativamente eficiente.

Para o caso bi-objetivo, as funções problema $B$ podem ser escritas como:

$$
\begin{aligned}
c &= \sum_{i=1}^{n} c_i x_i \\
p &= \sum_{i=1}^{n} p_i x_i \\
a &= \sum_{i=1}^{n} a_i x_i \\
h &= \sum_{i} h_i x_i - \sum_{i} \sum_{j \neq i} h_{ij} x_i x_j
\end{aligned}
$$

onde $c$ e $p$ são as mesmas funções do problema mono-objetivo, $a$ é a área da solução, dada pela soma das áreas das regiões qu estão na solução, e $h$ é o perímetro. Note que o perímetro de uma solução será dado pela soma dos perímetros das regiões que estão na solução (primeiro somatório), menos duas vezes cada uma das fronteiras comuns entre regiões vizinhas presentes na solução (o que é obtido pelo segundo somatório), resultando em uma função quadrática. Desse modo, a menos da restrição de conexidade, o problema de detecção de clusters espaciais poderia ser resolvido através da resolução da família dos seguintes problemas *knapsack* com três objetivos:

$$
\begin{aligned}
\max \quad & (-H(x), C(x), A(x)) \\
\text{s.a.} \quad & P(x) \leq P_k
\end{aligned}
\tag{6.8}
$$

obtidos variando a capacidade $P_k$ da mochila. O problema 6.8 pode ser reescrito em termos de uma abordagem $P_\epsilon$ (ver seção A.2, página 104) da seguinte maneira:

$$
\begin{aligned}
\min \quad & \sum_{i} h_i x_i - \sum_{i} \sum_{j \neq i} h_{ij} x_i x_j \\
\text{sujeito a} \quad & \sum_{i=1}^{n} a_i x_i \geq A_{k_1} \\
& \sum_{i=1}^{n} c_i x_i \geq C_{k_2} \\
& \sum_{i=1}^{n} p_i x_i \leq P_{k_3}
\end{aligned}
\tag{6.9}
$$

obtidos variando as capacidades da mochila ($A_{k_1}$, $C_{k_2}$ e $P_{k_3}$).

Reescrevendo a função-objetivo:

$$h = \sum_i h_i x_i + \sum_i \sum_{j \neq i} -h_{ij} x_i x_j \tag{6.10}$$

o problema (6.9) é claramente um problema da mochila quadrático com três restrições lineares de capacidade. Apesar de haver métodos de aproximação eficientes para o problema da mochila quadrático, a maioria não se aplica quando a matriz **h** possui entradas negativas, como no nosso caso. No entanto há aproximação por relaxação lagrangeana para o problema da mochila quadrático supermodular (Gallo & Simeone, 1988). Seja $d$ um número inteiro positivo, $D = \{1, ..., d\}$ e $S$ a coleção de todos os subconjuntos de $D$. Uma função $f : S \to \mathbb{R}$ é supermodular se

$$f(x) + f(y) \leq f(x \cap y) + f(x \cup y)$$

onde $x, y \in S$. Sabe-se (Gallo & Simeone, 1988; Nemhauser *et al.*, 1978) que uma função $f$ quadrática é supermodular se, e somente se, todos os coeficientes dos termos quadráticos são não-negativos. Portanto, um problema *knapsack* quadrático na forma

$$\begin{aligned} \max \quad & \sum_i v_i x_i + \sum_i \sum_{j \neq i} v_{ij} x_i x_j \\ \text{sujeito a} \quad & \sum_i w_i x_i \leq W \end{aligned} \tag{6.11}$$

é supermodular se $v_{ij} \geq 0 \ \forall i \neq j$. Note que a FO do problema (6.9) reescrita na forma (6.10) tem todos os coeficientes quadráticos não-positivos. Podemos então transformar o probblema (6.9) no problema de maximização

$$\max \quad \sum_i \sum_{j \neq i} h_{ij} x_i x_j - \sum_i h_i x_i \tag{6.12}$$

$$\text{sujeito a} \quad \sum_{i=1}^{n} a_i x_i \geq A_{k_1} \tag{6.13}$$

$$\sum_{i=1}^{n} c_i x_i \geq C_{k_2} \tag{6.14}$$

$$\sum_{i=1}^{n} p_i x_i \leq P_{k_3} \tag{6.15}$$

obtendo então a forma de um problema da mochila quadrático supermodular com múltiplas restrições. A solução de cada problema quadrático supermodular (obtido variando-se os índices $k_1$, $k_2$ e $k_3$) pode ser obtida resolvendo-se $O(n)$ problemas de fluxo máximo (Pisinger, 2007; Chaillou *et al.*, 1989). Ainda assim temos uma dificuldade extra que é o fato de que o problema (6.12)-(6.15) possui múltiplas restrições, enquanto que os apresentados na literatura (Gallo & Simeone, 1988; Pisinger, 2007) têm a forma (6.11), com apenas uma restrição e cujo sinal é $\leq$.

Vamos então tentar colocar o problema (6.12)-(6.15) na forma (6.11). Podemos somar as restrições (6.13) e (6.14) à FO, obtendo uma nova FO com funções ponderadas por pesos. A FO ficaria

$$\lambda_1 \left( \sum_i \sum_{j \neq i} h_{ij} x_i x_j - \sum_i h_i x_i \right) + \lambda_2 \sum_i a_i x_i + \lambda_3 \sum_i c_i x_i$$

e agrupando os termos lineares

$$\lambda_1 \sum_i \sum_{j \neq i} h_{ij} x_i x_j + \sum_i (\lambda_2 a_i + \lambda_3 c_i - \lambda_1 h_i) x_i$$

com $\lambda_1 + \lambda_2 + \lambda_3 = 1$ e $\lambda_i \geq 0$, $i = 1, 2, 3$. Chegamos então à forma

$$\max \quad \lambda_1 \sum_i \sum_{j \neq i} h_{ij} x_i x_j + \sum_i (\lambda_2 a_i + \lambda_3 c_i - \lambda_1 h_i) x_i \tag{6.16}$$

$$\text{sujeito a} \quad \sum_{i=1}^{n} p_i x_i \leq P_k \tag{6.17}$$

Para resolver este problema podemos recorrer à abordagem $P_\lambda$ (ver Anexo A, seção A.1). No entanto, o tratamento de um problema multiobjetivo através da abordagem $P_\lambda$ não é uma técnica conveniente quando o problema em questão não é convexo, uma vez que nos permite apenas encontrar as soluções suportadas do problema. Às vezes encontrar as soluções suportadas do problema é suficiente para se ter uma noção do conjunto Pareto-ótimo. Mas no nosso caso sabemos que existem soluções que estão no conjunto Pareto-ótimo do problema *knapsack* mas que não são soluções eficientes no espaço $LLR$ vs. $K$. Encontrando apenas as soluções suportadas correríamos o risco de perder uma solução não-suportada $s_{ns}$ que fosse eficiente no espaço de objetivos original e que, eventualmente, dominasse uma solução suportada $s_s$ que, no universo das soluções suportadas apenas, fosse não-dominada. Ou seja, passaríamos do problema de não ter todas as soluções para o problema de incluir no conjunto Pareto-ótimo soluções sub-ótimas. Como queremos encontrar uma cota superior para o problema, soluções sub-ótimas não são do nosso interesse.

Uma alternativa para contornar esse problema e gerar todas as soluções eficientes é transformar o problema de forma a obtermos uma função objetivo linear e partir para uma abordagem através da formulação de um problema $P_\epsilon$ (ver Anexo A, seção A.2). Podemos transformar nosso problema em um problema de programação linear, simplesmente fazendo a mudança de variáveis $y_{ij} = x_i x_j$, $i \neq j$. A princípio essa mudança faz com que o número de varíaveis do problema cresça de forma quadrática e o problema linear obtido pode ter um número inviável de variáveis. Mas na prática esse número não deve ser tão grande. A matriz $\mathbf{h}$ é, em geral, bastante esparsa porque sabe-se que, mesmo para mapas com muitas regiões, o número médio de vizinhos de cada região dificilmente será maior que 6 (Simon, 2002). De modo que a matriz $\mathbf{h}$ terá poucos elementos não-nulos fora da diagonal principal. Como não precisamos incluir no modelo as variáveis $y_{ij}$ cujo coeficiente $h_{ij}$ seja nulo, espera-se que o aumento no número de variáveis seja linear, e não quadrático. A família de problemas lineares fica

$$\begin{aligned} \min \quad & \sum_i h_i x_i - \sum_i \sum_{j \neq i} h_{ij} y_{ij} \\ \text{sujeito a} \quad & \sum_{i=1}^{n} a_i x_i \geq A_{k_1} \\ & \sum_{i=1}^{n} c_i x_i \geq C_{k_2} \\ & \sum_{i=1}^{n} p_i x_i \leq P_{k_3} \end{aligned} \tag{6.18}$$

obtida variando-se os índices $k_1$, $k_2$ e $k_3$. Note-se que aqui, além de termos problemas maiores (com mais variáveis), o número de problemas a ser resolvido é bem maior que no caso mono-objetivo, visto que deve-se variar, não uma, mas três restrições de "capacidade" do problema.

## 6.3. Resultados experimentais

Apresentaremos a seguir os resultados obtidos pela abordagem *knapsack*.

### 6.3.1. Caso mono-objetivo

A Figura 6.1 ilustra os resultados obtidos através da formulação *knapsack* e o desempenho do GA. Para cada método foram resolvidos mil problemas gerados a partir da distribuição aleatória de casos sob a hipótese nula, mantendo-se fixo o número total de casos, sendo que os problemas resolvidos por cada método foram os mesmos. Além dos dois métodos usamos também um AG irrestrito (AGI). O AGI comporta-se exatamente como o AG, mas sem restrição de conexidade. Isso significa que as soluções obtidas pelo AGI devem aproximar as soluções obtidas na abordagem *knapsack*. O gráfico mostra a distribuição de Weibull ajustada para o conjunto de soluções encontrado por cada método. Nesse experimento as soluções encontradas pelo AGI foram, em média, 5% piores que as soluções exatas obtidas na formulação *knapsack* em tempo muito menor (cerca de 0,5% do tempo), o que consideramos ser um bom desempenho. Apesar de o problema restrito possuir uma estrutura diferente, cremos que este seja um indicativo de que o AG está tendo também um bom desempenho.

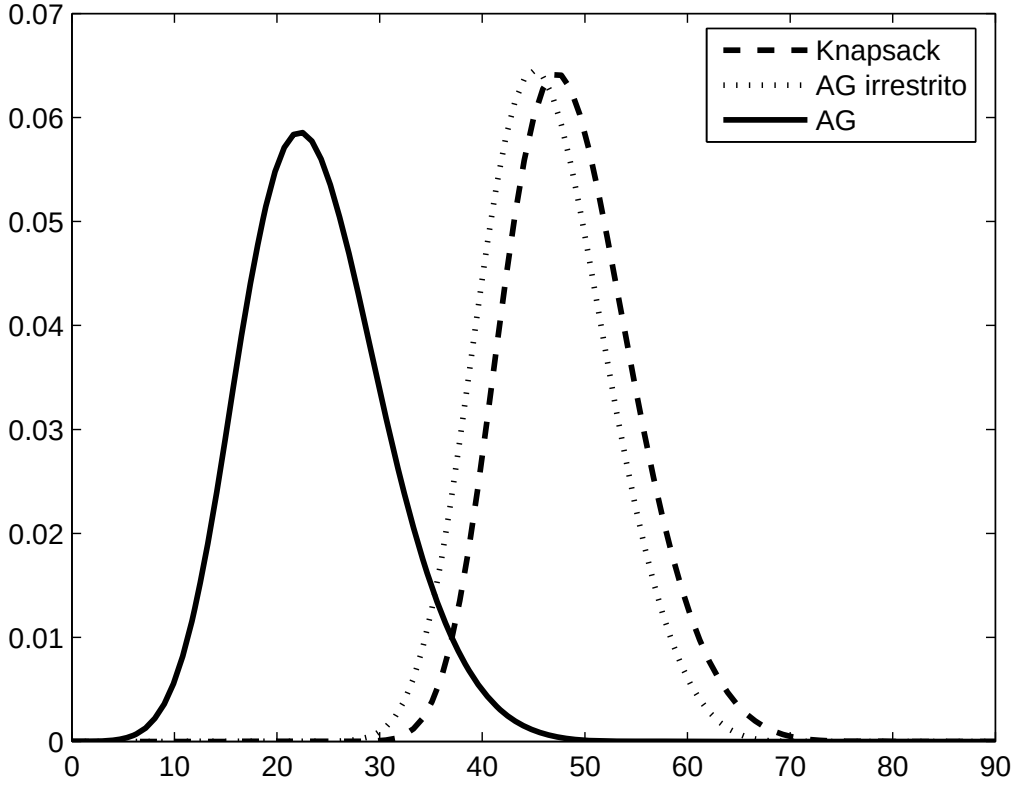



**Figura 6.1.:** Comparação entre as distribuições obtidas pelo AG e pela abordagem *knapsack* exata. A distribuição obtida pelo AGI sobre a formulação *knapsack* também é mostrada.

Assim sabemos que se $p_l$ é o $p$-valor calculado segundo a distribuição obtida pelo AG e $p_u$ é o $p$-valor obtido segundo a distribuição obtida pela abordagem *knapsack*, o verdadeiro $p$-valor certamente se encontra no intervalo $[p_l, p_u]$.

### 6.3.2. Caso bi-objetivo

Para o caso multiobjetivo consideramos apenas a solução obtida para os casos observados, já que a solução desse problema é bem mais difícil e demorada. Utilizamos um mapa do Estado de Minas Gerais dividido em 66 microrregiões, com população total $P$ = 17.751.651 e número de casos de câncer de mama $C$ = 3.543. O gráfico da Figura 6.2 mostra o conjunto Pareto-ótimo obtido pelo algoritmo *knapsack* quadrático e pelo AG. As soluções encontradas na abordagem *knapsack* foram divididas em conexas (■)

e desconexas (□). Um aspecto muito interessante desse resultado é que a compacidade "controla" a conexidade das soluções. Imagine uma solução formada por duas regiões $r_1$ e $r_2$ desconexas com perímetros $h_1$ e $h_2$ e áreas $a_1$ e $a_2$, respectivamente. Então a área total dessa solução será $a_1 + a_2$ e seu perímetro total será $h_1 + h_2$. Por outro lado, se $r_1$ e $r_2$ tem uma fronteira em comum de comprimento $f_{12}$, então a área da solução continuaria a mesma, mas o perímetro seria $h_1 + h_2 - 2f_{12}$. A segunda solução certamente teria compacidade mais alta que a primeira. Embora a restrição de conexidade não esteja presente na formulação do modelo há uma tendência a se buscar soluções conexas pois essas, em geral, minimizam o perímetro. Por esse motivo, as soluções de mais alta compacidade tendem a ser conexas.

Note ainda que o AG se comporta muito bem, apresentando um conjunto Pareto-ótimo muito próximo ao da solução exata, se considerarmos apenas as soluções conexas, que são as que nos interessam. Obviamente o AG não poderia se aproximar das soluções desconexas, já que ele considera a restrição de conexidade. Isso nos faz acreditar que o AG está fazendo um bom trabalho, e que os $p$-valores obtidos por essa via são razoáveis.

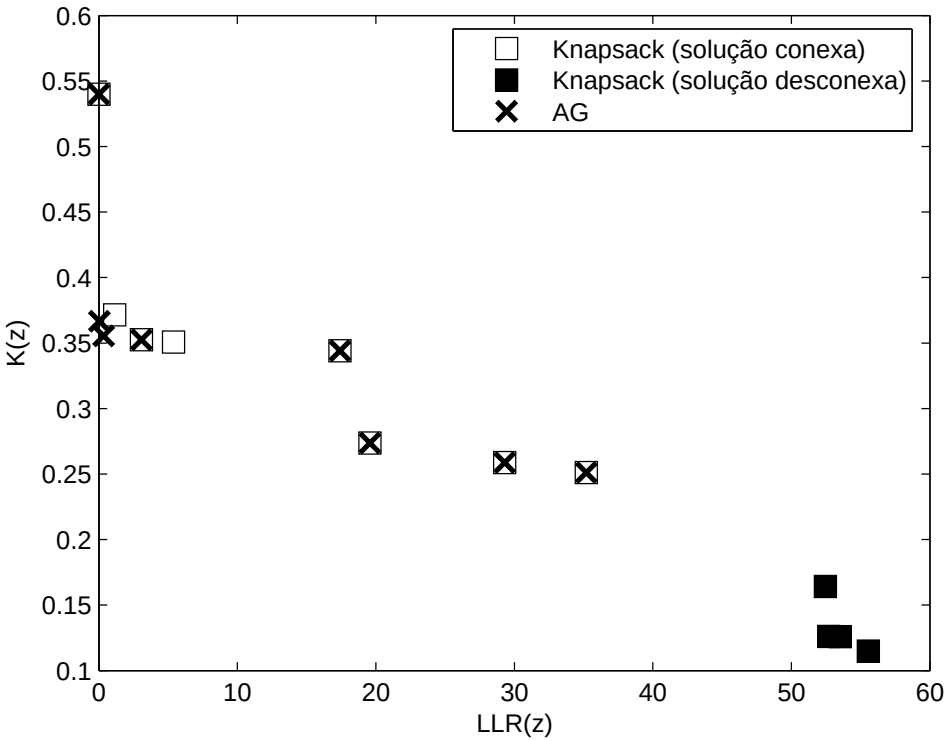



**Figura 6.2.:** Conjunto Pareto-ótimo encontrado pela abordagem *knapsack* e pelo AG.

As Figuras 6.3 e 6.4 mostram as soluções encontradas pelo *knapsack* e pelo AG, respectivamente.

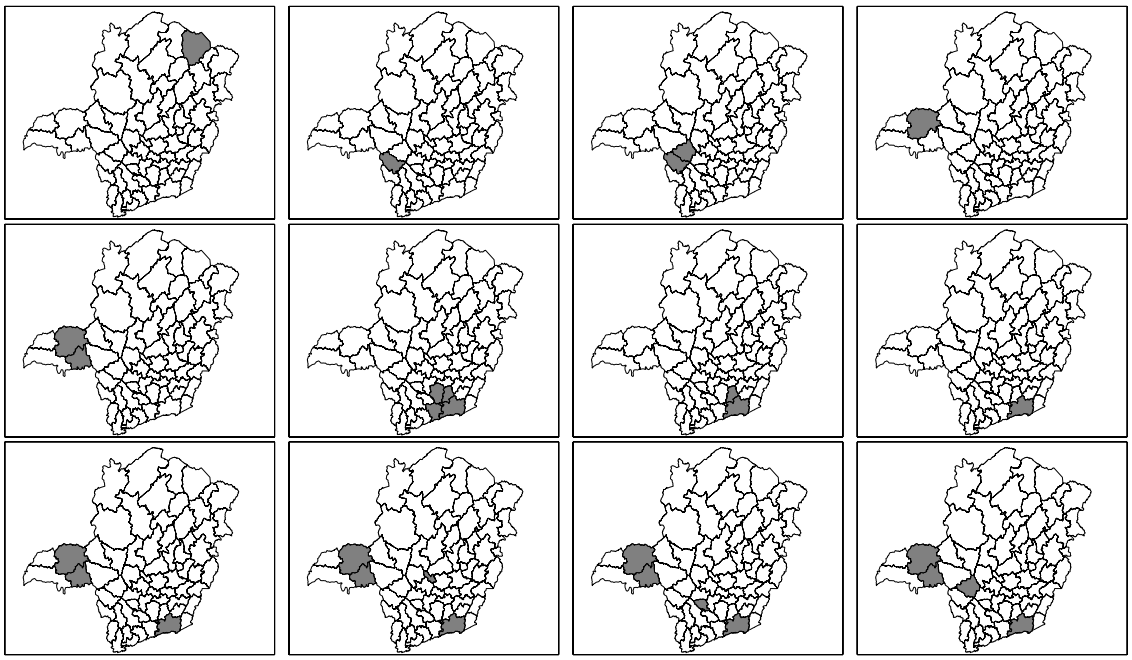


**Figura 6.3.:** Soluções dadas pela abordagem *knapsack*.

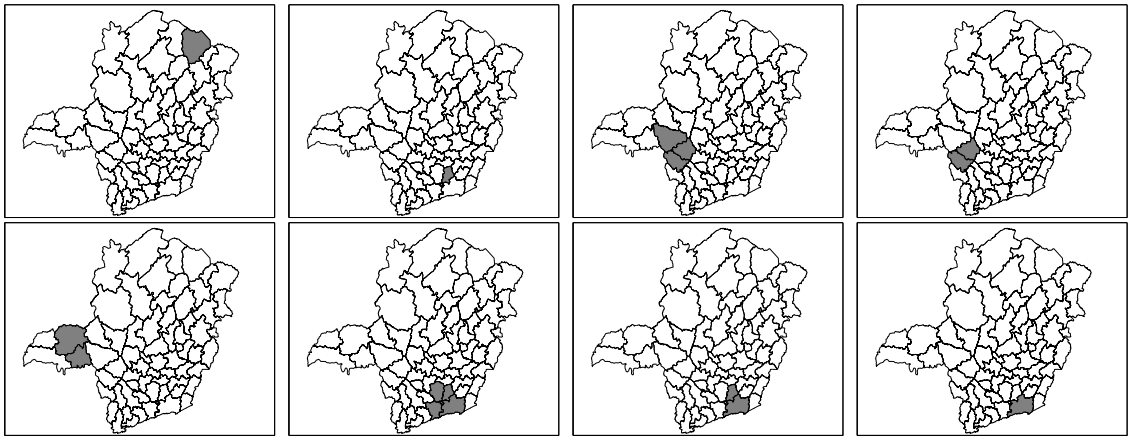


**Figura 6.4.:** Soluções dadas pelo AG.

## 6.4. Discussão

Para o problema multiobjetivo $(LLR, K)$ a variação das capacidades $A_k, C_k, P_k$ de forma ingênua nos leva a uma quantidade enorme de problemas que devem ser resolvidos para

se gerar todas as soluções não dominadas do espaço $(h, a, c, p)$. Essa variação é feita em três níveis de iteração, de acordo com o algoritmo 3. Nesse algoritmo, $S_a$ e $S_c$ são conjuntos que armazenam a área e o número de casos das soluções obtidas ao longo dos respectivos laços; $A$, $C$ e $P$ são, respectivamente, a área total do mapa, o número total de casos e a população total; $\mathcal{A}$, $\mathcal{C}$ e $\mathcal{P}$ são funções que recuperam a área, os casos e a população de uma solução $x$; e $H$ representa a solução do problema 6.18 para um conjunto de capacidades $A_k$, $C_k$ e $P_k$.

Este algoritmo inicia com o problema sem restrições, isto é, com $A_k = 0$, $C_k = 0$ e $P_k = P$. A partir daí a população vai diminuindo, sempre fazendo a capacidade $P_k$ da restrição ser menor em uma unidade que a última solução encontrada, até que se atinja $P_k = 0$ ou o problema não tenha mais solução factível. Nesse momento, soltamos novamente a restrição de população e repetimos o processo, mas com a restrição de casos apertada em uma unidade a mais do que a solução de menor número de casos obtida na última variação de população. E assim sucessivamente, até que a capacidade de casos atinja $C_k = C$ ou o problema não tenha mais soluções factíveis. Agora, de forma análoga à restrição de casos, apertamos a solução de área e repetimos o processo, até cobrirmos todo o espaço $(a, c, p)$.

Esse algoritmo, embora nos leve a todas as soluções eficientes do problema, gera uma grande quantidade de soluções repetidas. Isto porque, ao se resolver um problema não sabemos de antemão se sua melhor solução não atende também às restrições de algum outro problema resolvido anteriormente, de modo que podemos, por exemplo, ter que resolver milhares de problemas para, no fim, descobrirmos que temos apenas algumas dezenas de soluções distintas.

$A_k \leftarrow 0$;
**while** $A_k < A$ **do**
  $S_a \leftarrow \varnothing$;
  $C_k \leftarrow 0$;
  **while** $C_k < C$ **do**
    $S_c \leftarrow \varnothing$;
    $P_k \leftarrow P$;
    **while** $P_k > 0$ **do**
      $x \leftarrow H(A_k, C_k, P_k)$;
      $P_k \leftarrow \mathcal{P}(x) - 1$;
      $S_c \leftarrow S_c \cup \{\mathcal{C}(x)\}$;
      $S_a \leftarrow S_a \cup \{\mathcal{A}(x)\}$;
    **end**
    $c \leftarrow \min S_c$;
    $C_k \leftarrow c + 1$;
  **end**
  $a \leftarrow \min S_a$;
  $A_k \leftarrow a + \delta$;
**end**

**Algoritmo 3**: Geração de soluções eficientes via abordagem $P_\epsilon$.

# Capítulo 7.

# Considerações finais e trabalhos futuros

Nesta tese desenvolvemos um critério quantitativo para o problema de delineamento geográfico de *clusters* espaciais de geometria arbitrária, através da implementação de uma estratégia multiobjetivo para a detecção e inferência de *clusters* irregulares. Ao invés de executar um algoritmo de detecção de *clusters* sequencialmente com diferentes graus de penalização, um conjunto representativo de soluções é encontrado em paralelo. Desenvolvemos um algoritmo genético baseado na estrutura do NSGA-II que encontra o conjunto de soluções eficientes através da maximização de dois objetivos, a estatística *scan* e a regularidade da forma, ou compacidade. Este algoritmo multiobjetivo disponibiliza um conjunto de soluções que são ordenadas pelo critério de significância estatística. Dado um conjunto de soluções ótimas obtidas pelo algoritmo de busca de *clusters*, o problema foi reduzido à escolha da solução mais significante entre elas. O algoritmo desenvolvido neste trabalho se mostrou uma ferramenta apropriada e altamente eficiente, obtendo resultados satisfatórios. A aplicação de nossos métodos no mapa de casos de câncer de mama no nordeste dos Estados Unidos reitera a eficiência do algoritmo, delineando bastante bem os *clusters* presentes naquele mapa, que é amplamente conhecido e utilizado na literatura.

A utilização do perímetro das regiões para o cálculo da compacidade gerou bons resultados e mostrou que, se os dados estiverem em uma resolução adequada, o problema da explosão fractal do perímetro não ocorre. Dessa forma evitamos o uso de estimativas menos confiáveis para o perímetro, além de acelerar seu cálculo. O perímetro é obtido simplesmente através da avaliação de uma função quadrática.

Estendemos o conceito usual de significância de forma natural para o problema multiobjetivo através do conceito de isolinhas de $p$-valor. Essas isolinhas são calculadas através da função de aproveitamento, que se mostrou a ferramenta mais adequada para tratar esse problema. A função de aproveitamento é intuitiva e não compartilha dos vícios apresentados pelas outras tentativas de lidar com o problema. A distribuição obtida pela simulação de Monte Carlo é extrapolada com ajuste de modelos paramétricos de distribuições de probabilidade a partir de distribuições empíricas, permitindo maior precisão e velocidade computacional na estimação da significância. Assim, a análise permite que comparemos a posição relativa entre o conjunto de soluções não-dominadas e as isolinhas. A inclinação de um em relação ao outro define se as soluções mais regulares ou irregulares são mais significativas. Os modelos paramétricos foram testados e comparados, e consideramos que os resultados indicam a viabilidade de se usar esse tipo de abordagem.

Acreditamos que a possibilidade de identificar a "melhor solução" através de um critério quantitativo abre uma nova porta no problema de *clusters* irregulares. As tentativas anteriores de lidar com o problema de simultaneamente levar em conta a maximização da $LLR$ e a escolha de uma geometria adequada para o *cluster*, utilizando otimização mono-objetivo, necessariamente levavam a alguma escolha arbitrária de parâmetros de balanço entre os dois objetivos que não podia ser justificada satisfatoriamente. A introdução do conceito de conjunto de Pareto nesse problema, seguido da escolha da solução mais significativa, permite que a escolha da melhor solução seja rigorosa, mas sem a necessidade de nenhum parâmetro arbitrário. O processo de seleção automática e a velocidade do método removem duas barreiras à utilização de métodos de detecção de *clusters* irregulares. Testes numéricos mostram que o poder de detecção do algoritmo multiobjetivo é compatível com o poder do algoritmo genético mono-objetivo, sem comprometimento no que diz respeito ao esforço computacional.

Mostramos que o erro relativo à estimação do $p$-valor das soluções através do AG pode ser controlado através da resolução de famílias de problemas *knapsack* binários de onde conseguimos recuperar a solução do problema original relaxado. Essa solução para o problema relaxado nos dá uma cota superior, em contraponto com a cota inferior dada pelo AG, permitindo que encontremos um intervalo onde sabemos que a solução verdadeira se encontra. Embora os problemas *knapsack* constituam uma classe de problemas combinatórios e de alta complexidade computacional, mostramos que é possível

a aplicação dessa abordagem para problemas de dimensões razoavelmente grandes para nosso tipo de problema. Para o caso mono-objetivo os problemas podem ser resolvidos de forma a se obter uma distribuição que, em geral, superestima a distribuição da estatística de teste sob a hipótese nula, já que, em geral, as soluções encontradas são desconexas. Para o caso multiobjetivo o problema *knapsack* é bem mais complicado. Mesmo assim mostramos, para um exemplo pequeno, que essa abordagem pode ser viável, uma vez que, apesar de estarmos trabalhando com um problema relaxado, o uso da compacidade como um dos objetivos pode nos conduzir a soluções que satisfazem a condição de conexidade do problema original e, portanto, constituem soluções legítimas do ponto de vista de detecção de *clusters* espaciais.

## 7.1. Trabalhos futuros

O problema de detecção de *clusters* espaciais, como um problema de otimização, pode ser resolvido através do uso de várias ferramentas de otimização. Nesse trabalho utilizamos um algoritmo genético que se mostrou uma escolha adequada. No entanto outros algoritmos poderiam ser utilizados. Em particular, algoritmos evolucionários que sejam capazes de lidar com a restrição de conexidade imposta pelo problema podem, em princípio, ser utilizados. Temos interesse em aplicar outras técnicas e comparar o desempenho com o que tem sido feito até então. Além disso, cremos que a utilização de ferramentas de aprendizado como redes neurais (Moreira *et al.*, 2007) e métodos de inteligência baseados em estatística bayesiana (Neill *et al.*, 2005a) têm grande potencial no delineamento de *clusters* espaciais.

A modelagem do problema como um problema *knapsack* se mostrou bastante promissora, principalmente no caso multiobjetivo para o qual essa abordagem parece ter um potencial para resolver o problema de maneira exata. Nesse sentido, os principais pontos que merecem ser objeto de pesquisa são:

- Desenvolvimento de uma maneira mais eficiente de se variar as restrições de capacidade. O algoritmo implementado gera uma numerosa família de problemas que devem ser resolvidos para que se obtenha apenas algumas poucas soluções distintas.

- Uma outra maneira de se acelerar a resolução do problema seria a adoção de outra medida de compacidade, de forma que obtivéssemos uma função linear e o número original de variáveis fosse mantido. Uma alternativa possivelmente viável é a substituição do perímetro e do fecho convexo por uma medida linear, como distância, por exemplo. Selkirk (1982) descreve algumas medidas alternativas para a medida de compacidade.

## 7.2. Produção bibliográfica

Apresentamos as publicações que resultaram da nossa pesquisa durante o doutorado.

Publicações diretamente decorrentes do trabalho desenvolvido nessa tese:

**Artigos publicados em periódicos:**

- Duczmal, L., Cançado, A. L. F., & Takahashi, R. H. C. 2008. Delineation of irregularly shaped disease clusters through multiobjective optimization. *Journal of Computational and Graphical Statistics*, **17**(2), 243–262.

**Trabalhos em conferências**

- Duczmal, L. H. ; Cançado, A. L. F. ; Takahashi, R. H. C. . What is the true shape of a disease cluster? The multi-objective genetic scan. In: Syndromic Surveillance Conference, 2006, Baltimore, MD. *Advances in Disease Surveillance*.

Publicações decorrentes de aplicações do trabalho apresentado nessa tese:

**Capítulos de livros publicados:**

- Duczmal, L. H. ; Cançado, A. L. F. ; Takahashi, R. H. C. ; Bessegato, L. F. . A Comparison of Simulated Annealing, Elliptic and Genetic Algorithms for Finding Irregularly Shaped Spatial Clusters. In: Vedran Kordic. (Org.). *Simulated Annealing*. Viena: I-Techonline, 2008, v. 1, p. 1-18.

**Trabalhos em conferências:**

- Duarte, A. R. ; Duczmal, L. H. ; Ferreira Neto, S. J. ; Cançado, A. L. F. . Optimizing simultaneously the geometry and the internal cohesion of clusters. In:

International Society for Disease Surveillance Seventh Annual Conference, 2008, Raleigh. *Advances in Disease Surveillance*, 2008.

- Duczmal, L. H. ; Ferreira Neto, S. J. ; Duarte, A. R. ; Soares, M. V. ; Gontijo, E. D. ; Cançado, A. L. F. ; Takahashi, R. H. C. . Geographically meaningful cluster scanning through weak link correction. In: International Society for Disease Surveillance Seventh Annual Conference, 2008, Raleigh. *Advances in Disease Surveillance*, 2008.

- Duczmal, L. H. ; Cançado, A. L. F. ; Takahashi, R. H. C. ; Ferreira Neto, S. J.; Moura, F. R. ; Duarte, A. R. ; Tavares, R. . Multi-Objective Spatial Scans for Disease Cluster Detection. In: International Workshop in Applied Probability, 2008, Compiegne. *Proceedings of the International Workshop in Applied Probability 2008*.

- Patil, G. P. ; Duczmal, L. H. ; Tavares, R. ; Cançado, A. L. F. . Detection of spatial clusters in maps equipped with environmentally defined structures. In: 7th Annual International Conference on Digital Government Research, 2006, San Diego. *ACM Proceedings - Conference on Digital Government Research*, 2006.

**Trabalhos aceitos em conferências:**

- Duczmal, L. H. ; Tavares, R. ; Cançado, A. L. F. ; Patil, G. P. . Finding Spatial Clusters in Maps Equipped with Environmentally Defined Structures with Disease Policy Case Studies. In: Joint Statistical Meeting, Washington. *Proceedings of the 2009 Joint Statistical Meeting*, 2009.

**Artigos submetidos para publicação em periódicos:**

- Duczmal, L. H. ; Tavares, R. ; Cançado, A. L. F. . Finding spatial clusters in maps equipped with environmentally defined structures.

- Duarte, A. R., Duczmal, L. H., Ferreira Neto, S. J., Cançado, A. L. F. . Weak link correction for a graph based spatial scan with an application to Chagas' disease clusters.

- Duarte, A. R., Duczmal, L. H., Ferreira Neto, S. J., Cançado, A. L. F. . Optimizing simultaneously the geometric shape and internal cohesion of spatial clusters.

# Apêndice A.

# Técnicas de geração de soluções eficientes

Apresentaremos a seguir dois métodos de obtenção de soluções eficientes para o problema de otimização multiobjetivo dado por:

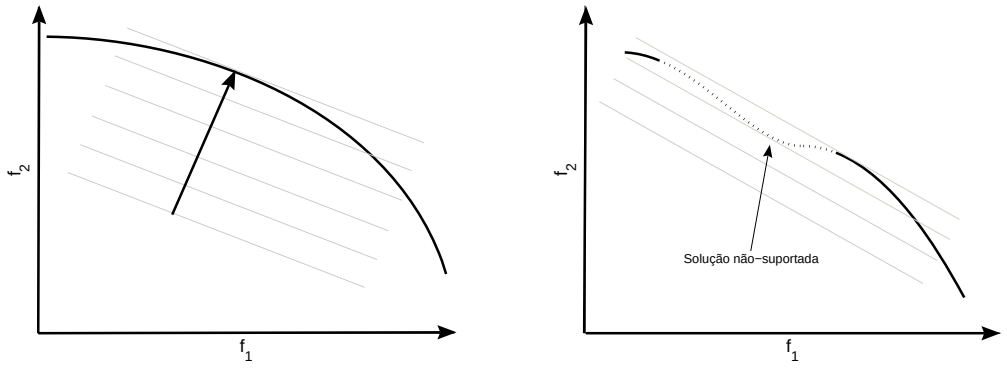
$$\max_x f(x) = (f_1(x), ..., f_n(x)) \tag{A.1}$$

## A.1. Problema Ponderado - $P_\lambda$

A abordagem $P_\lambda$ do problema de otimização da equação A.1 é feita a partir da substituição das funções-objetivo pela soma destas, ponderada pelo vetor de pesos $\lambda = \{\lambda_1, ..., \lambda_n\}$, com $\lambda_1 + ... + \lambda_n = 1$. Assim, o problema fica

$$\max_x \lambda f = \sum_{i=1}^{n} \lambda_i f_i(x)$$

que é um problema de otimização onde o único objetivo é dado pelo produto escalar $\lambda f$. Para um dado conjunto de pesos $\lambda$ o objetivo é encontrar $x^*$ tal que $y^* = \lambda f(x^*)$ seja o máximo de $\lambda f$. Note que a equação $y^* = \lambda f(x^*)$ é a equação de um hiperplano. Esse hiperplano é denominado *hiperplano suporte* do conjunto de soluções. Para exemplificar essa interpretação geométrica, a Figura A.1(a) ilustra o conjunto Pareto-ótimo de um problema de maximização de dois objetivos, $f_1$ e $f_2$. Observe que uma dada combinação




(a) (b)

**Figura A.1.:** (a) A combinação de pesos define uma direção de busca e uma família de hiperplanos. Nesse exemplo, o hiperplano que tangencia o conjunto pareto ótimo minimiza a função ponderada e é denominado hiperplano suporte. (b) O conjunto Pareto-ótimo possui um trecho côncavo. No trecho pontilhado as soluções não possuem hiperplano suporte e, portanto, não são encontradas pela abordagem de soma ponderada dos objetivos.

de pesos define uma família de hiperplanos paralelos (nesse exemplo, uma família de retas paralelas). O hiperplano correpondente ao menor valor de $\lambda f$, nesse caso a reta que tangencia o conjunto Pareto-ótimo, será o hiperplano suporte do conjunto das soluções no ponto $y^*$.

Agora vamos observar a Figura A.1(b). Nesse caso o conjunto Pareto-ótimo possui um trecho côncavo. Isso faz com que no trecho pontilhado nenhuma solução possua hiperplano suporte, porque sempre é possível encontrar uma solução, avançando-se na mesma direção definida pelo conjunto de pesos $\lambda$, melhor do que qualquer outra nesse trecho. Por isso essas soluções são denominadas soluções não-suportadas. A abordagem de soma ponderada de objetivos nos permite, portanto, encontrar apenas as soluções suportadas do problema.

## A.2. Problema $\epsilon$-restrito - $P_\epsilon$

Na abordagem $P_\epsilon$ o problema original A.1 é transformado de forma que apenas um dos objetivos é considerado, enquanto os outros são transformados em restrições, da seguinte forma

$$\begin{aligned} \max_{x} \quad & f_i(x) \\ \text{sujeito a:} \quad & f_j(x) \geq \epsilon_j \ , \ j \neq i \end{aligned}$$

sendo que as diferentes soluções são obtidas variando-se o vetor $\epsilon$ que limita as restrições. Geometricamente, para um problema originalmente de maximização de dois objetivos $f_1$ e $f_2$, suponha que o problema tenha sido transformado em

$$\begin{aligned} \max_{x} \quad & f_1(x) \\ \text{sujeito a:} \quad & f_2(x) \geq \epsilon \end{aligned}$$

Nesse caso, devemos variar $\epsilon$ de forma a gerar o conjunto de soluções eficientes. A princípio pode-se resolver o problema irrestrito $\max_x f_1(x)$, que dará uma solução ótima $x_0^*$ (veja Figura A.2) que maximiza a função $f_1$. O próximo passo pode então ser dado considerando-se a restrição $f_2(x) \geq \epsilon$, onde $\epsilon = f_2(x_0^*) + \delta$, com $\delta > 0$, e obtendo-se a solução $x_1^*$, e assim sucessivamente.

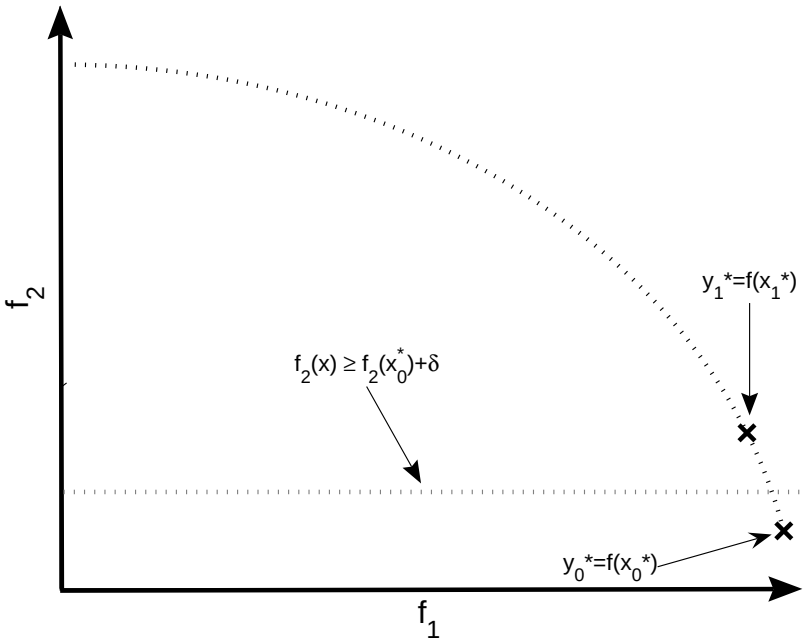



**Figura A.2.:** Abordagem $P_\epsilon$.

# Apêndice B.

# Teste de Kolmogorov-Smirnov

O teste de Kolmogorov-Smirnov é usado para decidir se uma amostra é proveniente de uma população com uma determinada distribuição $F$. Considerando uma amostra de variáveis independentes e identicamente distribuídas $\{X_1, X_2, ..., X_n\}$, o teste avalia a qualidade de ajuste da distribuição $F$ à distribuição das observações de $X_i$. A distribuição empírica das observações $X_i$ é dada por

$$F_n(x) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{I}(X_i < x)$$

onde $\mathbf{I}(X_i \leq x)$ é a função-indicador (igual a 1 se $X_i \leq x$, e zero caso contrário). O teste é formulado de acordo com as seguintes hipóteses:

$$\begin{cases} h_0: & \text{a distribuição da amostra é } F \\ h_a: & \text{a distribuição da amostra não é } F \end{cases}$$

A estatística de teste de Komogorov-Smirnov é baseada na diferença máxima $D_n$ entre as funções de distribuição empírica e ajustada, isto é

$$D_n = \max_x |F_n - F|$$

Sabe-se que, se a distribuição de $X_i$ é contínua, então sob a hipótese nula (isto é, se $F$ é de fato a distribuição de $X_i$) a distribuição de $\sqrt{n}D_n$ converge para a distribuição de Kolmogorov-Smirnov, dada por:

$$F_K(k) = P(K \leq k) = \frac{2\pi}{k} \sum_{i=1}^{\infty} e^{-(2i-1)^2\pi^2/(8x^2)}$$

seja qual for a distribuição $F$. Assim, uma vez computada a estatística $D_n$, basta compará-la à distribuição de Komogorov-Smirnov para decidir se a hipótese nula é ou não rejeitada. Para um nível de significância $\alpha$, a hipótese nula é rejeitada se $\sqrt{n}D_n > k_\alpha$, onde $k_\alpha$ é o ponto que satisfaz $F_K(k_\alpha) = 1 - \alpha$.

O teste de Kolmogorov-Smirnov só é válido se a distribuição hipotética $F$ está completamente especificada, isto é, se os parâmetros de $F$ não forem estimados a partir dos dados. Caso os parâmetros da distribuição $F$ sejam estimados a partir dos dados, a distribuição de $D_n$ passa a ser dependente de $F$ e a distribuição de Kolmogorov-Smirnov não pode mais ser utilizada para se calcular o $p$-valor do teste (Lilliefors, 1967; Durbin, 1975; Keutelian, 1991).

# Referências Bibliográficas


Abrams, A., Kulldorff, M., & Kleinman, K. 2006. Empirical/Assymptotic P-values for Monte Carlo-Based Hypothesis Testing: an Application to Cluster Detection Using the Scan Statistic. *Advances in Disease Surveillance*, **1**, 1.

Assunção, R., Tavares, A., Costa, M., & Ferreira, S. 2006. Fast detection of arbitrarily shaped disease clusters. *Statistics in Medicine*, **25**, 723–742.

Besag, J., & Newell, J. 1991. The detection of clusters in rare diseases. *Journal of the Royal Statistical Society*, **154**, 143–155.

Brent, R. P. 1973. *Algorithms for Minimization Without Derivatives*. Englewood Cliffs, New Jersey: Prentice-Hall.

Carrano, E. G., Soares, L. A. E., Takahashi, R. H. C., Saldanha, R. R., & Neto, O. M. 2006. Electric distribution network multiobjective design using a problem-specific genetic algorithm. *IEEE Transactions on Power Delivery*, **2**(21), 995–1005.

Chaillou, P., Hansen, P., & Mahieu, Y. 1989. *Best network flow bound for the quadratic knapsack problem*. Combinatorial Optimization, Lecture Notes in Mathematics. Springer. Pages 225–235.

Coles, S. 2001. *An Introduction to Statistical Modeling of Extreme Values*. Springer-Verlag London Limited.

Conley, J., Gahegan, M., & MacGill, J. 2005. A genetic approach to detecting clusters in point-data sets. *Geographical Analysis*, **37**, 286–314.

da Fonseca, V. G., Fonseca, C. M., & Hall, A. O. 2001. Inferential Performance Assessment of Stochastic Optimisers and the Attainment Function. *Pages 213–225 of: Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization, Lecture Notes In Computer Science*, vol. 1993. Berlin: Springer-Verlag.

Deb, K., Pratap, A., Agrawal, S., & Meyarivan, T. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, **6**(2), 182–197.

Dekker, T. 1969. Finding a zero by means of successive linear interpolation. *In:* Dejon, B., & Henrici, P. (eds), *Constructive Aspects of the Fundamental Theorem of Algebra*. London: Wiley.

Duczmal, L., & Assunção, R. 2004. A simulated annealing strategy for the detection of arbitrarily shaped spatial clusters. *Computational Statistics and Data Analysis*, **45**, 269–286.

Duczmal, L., & Buckeridge, D. L. 2005. Using Modified Spatial Scan Statistic to Improve Detection of Disease Outbreak When Exposure Occurs in Workplace. *Page 187 of: Morbidity and Mortality Weekly Report*, vol. 54.

Duczmal, L., & Buckeridge, D. L. 2006. A Workflow Spatial Scan Statistic. *Statistics in Medicine*, **25**, 743–754.

Duczmal, L., Kulldorff, M., & Huang, L. 2006. Evaluation of spatial scan statistics for irregularly shaped clusters. *Journal of Computational and Graphical Statistics*, **15**(2), 428–442.

Duczmal, L., Cançado, A. L. F., Takahashi, R. H. C., & Bessegato, L. F. 2007. A Genetic Algorithm for Irregularly Shaped Spatial Scan Statistics. *Computational Statistics and Data Analysis*, **52**, 43–52. DOI:10.1016/j.csda.2007.01.016.

Duczmal, L., Cançado, A. L. F., & Takahashi, R. H. C. 2008. Delineation of irregularly shaped disease clusters through multiobjective optimization. *Journal of Computational and Graphical Statistics*, **17**(2), 243–262.

Durbin, J. 1975. Kolmogorov-Smirnov tests when parameters are estimated with applications to tests of exponentiality and tests on spacings. *Biometrika*, **1**, 5–22.

Dwass, M. 1957. Modified randomization tests for nonparametric hypotheses. *Annals of Mathematical Statistics*, **28**(1), 181–187.

Fonseca, C. M., & Fleming, P. 1995. An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation*, **3**(1), 1–16.

Fonseca, C. M., da Fonseca, V. G., & Paquete, L. 2005. Exploring the Performance of Stochastic Multiobjective Optimisers with the Second-Order Attainment Function. *Pages 250–264 of: Proceedings of the Third International Conference on Evolutionary Multi-Criterion Optimization, Lecture Notes In Computer Science*, vol. 3410. Berlin: Springer-Verlag.

Gallo, G., & Simeone, B. 1988. On the supermodular knapsack problem. *Mathematical Programming*, **45**, 295–309.

Hüsler, J., Cruz, P., Hall, A., & Fonseca, C. M. 2002. On Optimization and Extreme Value Theory. *Methodology And Computing In Applied Probability*, **5**(2), 183–195.

Iyengar, V. S. 2004. Space-time Clusters with flexible shapes. *Pages 71–76 of: Morbidity and Mortality Weekly Report*, vol. 54.

Johnson, N. L., Kotz, S., & Balakrishnan, N. 1995. *Continuous Univariate Distributions*. 2nd edn. Wiley Series in Probability and Statistics, vol. 2. John Wiley & Sons.

Keutelian, H. 1991. The Kolmogorov-Smirnov test when parameters are estimated from data. *CDF Note 1285*. http://www-cdf.fnal.gov/publications/cdf1285_KS_test_after_fit.pdf.

Kotz, S., & Nadarajah, S. 2000. *Extreme Value Distributions. Theory and Applications*. Imperial College Press, London.

Kulldorff, M. 1997. A Spatial Scan Statistic. *Communications in Statistics: Theory and Methods*, **26**(6), 1481–1496.

Kulldorff, M., & Nagarwalla, N. 1995. Spatial disease clusters: detection and inference. *Statistics in Medicine*, **14**, 779–810.

Kulldorff, M., Feuer, E. J., Miller, B. A., & Freedman, L. S. 1997. Breast cancer clusters in the Northeast United States: a geographic analysis. *American Journal of Epidemiology*, **146**, 161–170.

Kulldorff, M., Tango, T., & Park, P. J. 2003. Power comparisons for disease clustering sets. *Computational Statistics and Data Analysis*, **42**, 665–684.

Kulldorff, M., Heffernan, R., Hartman, J., Assunção, R., & Mostashari, F. 2005. A Space-Time Permutation Scan Statistic for Disease Outbreak Detection. *PLoS Medicine*,

**2**(3), e59.

Kulldorff, M., Huang, L., Pickle, L., & Duczmal, L. 2006. An Elliptic Spatial Scan Statistic. *Statistics in Medicine*, **25**, 3929–3943.

Kulldorff, M., Mostashari, F., Duczmal, L., Yih, K., Kleinman, K., & Platt, R. 2007. Multivariate Scan Statistics for Disease Surveillance. *Statistics in Medicine*, **26**, 1824–1833.

Lawson, A., Biggeri, A., & Böhning, D. 1999. *Disease mapping and risk assessment for public health*. New York: John Wiley and Sons.

Lilliefors, H. W. 1967. On the Kolmogorov-Smirnov test for normality with mean and variance unknown. *American Statistical Association Journal*, **62**, 399–402.

Moreira, G. J. P., Takahashi, R. H. C., & Duczmal, L. H. 2007. Delineating Spatial Clusters with Artificial Neural Networks. *Page 104 of: Proceedings of the Syndromic Surveillance Conference*, vol. 4.

Neill, D. B., Moore, A. W., & Cooper, G. 2005a. A Bayesian scan statistic for spatial cluster detection. *Page 55 of: Proceedings of the National Syndromic Surveillance Conference*, vol. 1.

Neill, D. B., Moore, A. W., Maheshkumar, R. S., & Daniel, K. 2005b. An Expectation-Based Scan Statistic for Detection of Space-Time Clusters. *Page 56 of: Proceedings of the National Syndromic Surveillance Conference*, vol. 1.

Nemhauser, G. L., Wolsey, L. A., & Fisher, M. L. 1978. An analysis of approximations for maximizing submodular set functions - I. *Mathematical Programming*, **14**, 265–294.

Nepomuceno, E. G., Takahashi, R. H. C., Amaral, G. F. V., & Aguirre, L. A. 2003. Nonlinear identification using prior knowledge of fixed points: a multiobjective approach. *International Journal of Bifurcation and Chaos*, **13**(15), 1229–1246.

Openshaw, S., & Perrée, T. 1996. User-Centred Intelligent Spatial Analysis of Point Data. *Pages 119–134 of:* Parker, D. (ed), *Innovations in GIS*. Bristol: Taylor & Francis.

Parzen, E. 1962. On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, **33**(3), 1065–1076.

Patil, G. P., & Taillie, C. 2004. Upper level set scan statistic for detecting arbitrarily shaped hotspots. *Environmental and Ecological Statistics*, **11**, 183–197.

Patil, G. P., Duczmal, L., Tavares, R., & Cançado, A. L. F. 2006. Detection of spatial clusters in maps equipped with environmentally defined structures. *In: Proceedings of the 7th International Conference on Digital Government Research.*

Pisinger, D. 2007. The quadratic knapsack problem - a survey. *Discrete Applied Mathematics*, **155**, 623–648.

Qiao, H., & Tsokos, C. P. 1995. Estimation of the three parameter Weibull probability distribution. *Mathematics and Computers in Simulation*, **39**, 173–185.

Ramos, R. M., Saldanha, R. R., Takahashi, R. H. C., & Moreira, F. J. S. 2003. The real-biased multiobjective genetic algorithm and its application to the design of wire antennas. *IEEE Transactions on Magnetics*, **39**(3), 1329–1332.

Sahajpal, R., Ramaraju, G. V., & Bhatt, V. 2004. Applying niching genetic algorithms for multiple cluster discovery in spatial analysis. *In: International Conference on Intelligent Sensing and Information Processing.*

Selkirk, K. 1982. *Pattern and Place: An Introduction to the Mathematics of Geography.* New York: Cambridge University Press.

Simon, G. A. 2002. Map Neighbor Counts. *Geographical Analysis*, **34**(4), 363 – 375.

Takahashi, R. H. C., Vasconcelos, J. A., Ramirez, J. A., & Krahenbuhl, L. 2003. A multiobjective methodology for evaluating genetic operators. *IEEE Transactions on Magnetics*, **39**(3), 1321–1324.

Takahashi, R. H. C., Palhares, R. M., Dutra, D. A., & Gonçalves, L. P. S. 2004. Estimation of Pareto sets in the mixed H2/H-infinity control problems. *International Journal of Systems Science*, **35**(1), 55–67.

Tango, T., & Takahashi, K. 2005. A flexibly shaped spatial scan statistic for detecting clusters. *International Journal of Health Geographics*, **4**(11).

Teixeira, R. A., Braga, A. P., Takahashi, R. H. C., & Saldanha, R. R. 2000. Improving generalization of MLPs with multi-objetive optimization. *Neurocomputing*, **35**(4), 189–194.