

Base Station Placement across a Simulated City using a Biased Random-key Genetic Algorithm

Lucas O. Amorim¹, Daniel H. C. Vassalo¹, Gabriel B. Pereira¹, Rian G. S. Pinheiro¹

¹Instituto de Computação – Universidade Federal de Alagoas (UFAL)
Maceió – AL – Brazil

{loa, dhcv, gbp, rian}@ic.ufal.br

Abstract. *As mobile communication technologies evolve, smart base station positioning has been getting increasingly important. The base station placement problem deals with the problem of efficiently positioning cell sites, in order to achieve balance between coverage and service cost. This paper proposes an implementation using the BRKGA meta-heuristic, which focus on achieving a weighted coverage/cost balance. Several tests have been conducted to prove the effectiveness of the proposed solution, and BRKGA showed values 1.4% apart from optimal coverage, on average. These results are better in both execution time and area coverage, when compared to methods introduced in literature.*

Resumo. *Conforme as tecnologias de comunicação móvel evoluem, o posicionamento inteligente de estações-base vem se tornando cada vez mais importante. O problema de posicionamento de estações-base trata de posicionar estações de forma eficiente, de forma a trazer um balanço entre cobertura e custo de serviço. Neste trabalho é proposta uma implementação utilizando a meta-heurística BRKGA, que visa alcançar esse balanço de forma ponderada. Foram realizados diversos testes para provar a eficácia da solução proposta, onde o BRKGA apresentou valores distantes 1,4%, em média, da cobertura ótima, resultados superiores em termos de tempo de execução e cobertura de área frente aos métodos adotados pela literatura.*

1. Introdução

O problema de posicionamento de estações-base vem se tornando cada vez mais importante nos últimos anos. A chegada da quinta geração de redes móveis, acompanhada do suporte nativo para *Multiple Input/Multiple Output* (MIMO) massivo [Gao et al. 2015], os dispositivos para internet das coisas (IoT) [Skouby and Lynggaard 2014] e as operações em faixas de frequência de ondas milimétricas (mmWave) [Gao et al. 2015], exigem cada vez mais capilaridade nas redes, o que se traduz na necessidade de mais ERBs (estações rádio-base) numa mesma área ou região, quando comparado a tecnologias anteriores, como o *Global System for Mobile* (GSM) ou o *Long Term Evolution* (LTE) [Ge et al. 2016].

Não somente isso, nos últimos anos temos visto um crescimento expressivo e veloz na cobertura 4G LTE pelo Brasil todo. Ao final de 2016, a operadora TIM Brasil cobria aproximadamente 900 municípios com tecnologia 4G LTE, durante o ano de 2017, a cobertura expandiu muito rapidamente, atingindo a marca de 3000 municípios ao fim do

ano, um aumento de mais de 300%¹. Um aumento por si só, por menor que seja, já requer um planejamento buscando entregar o melhor serviço, que envolve evitar interferência de células vizinhas, evitar áreas de sombra e prover uma cobertura superior.

Para uma adição de ERBs na escala que a TIM adotou no Brasil no ano de 2017, é necessário um conhecimento profundo do problema e trabalhar com soluções que permitem não somente chegar em um resultado ótimo para grandes instâncias, mas também chegar em tais resultados em tempo hábil, de forma a condizer com a propaganda e satisfazer o interesse de seus consumidores.

Tendo em vista que novas ERBs serão necessárias para a introdução de novas tecnologias e expansão das atuais, faz-se necessário o desenvolvimento de uma técnica de posicionamento inteligente. O aumento da faixa de frequência de operação é uma tendência mundial, uma vez que a largura de banda disponível nas frequências baixas está cada vez mais escassa [Boccardi et al. 2014]. Conforme cresce a frequência de operação, o raio de cobertura diminui (fenômeno abordado nos modelos de *path loss*), combina-se isso ao fato de que existe uma burocracia para o licenciamento de novas ERBs, e então forma-se um problema complexo, que deve considerar não somente a localização, mas também custo e cobertura.

Um outro fator a ser considerado nesse problema é a diversidade de frequências em que uma tecnologia pode operar. Nos últimos 4 anos, foi observado um crescimento da técnica de *refarming* no Brasil¹. No *refarming*, faixas de frequência utilizadas em tecnologia legada são realocadas para serem usadas em tecnologias atuais, disponibilizando assim mais espectro, que torna-se cada dia mais precioso [Vieira et al. 2010].

Devido à vantagem de transferir o espectro das tecnologias 2G e 3G para o LTE e 5G, as operadoras móveis realizaram o *refarming* das mais variadas faixas de frequência leiloadas ao passar dos anos. Por exemplo, a TIM, na cidade de Maceió/AL, realizou o *refarming* de toda a faixa de frequência dos 2100MHz² para o LTE. Pelo fato da interface aérea do LTE ser fundamentalmente diferente de uma rede 3G, os princípios de cobertura são totalmente alterados [Toskala et al. 2006], logo, é necessário realizar novamente um estudo de potência e frequência das ERBs em operações de *refarming*.

Vários métodos são utilizados na literatura para otimizar o posicionamento das ERBs, como por exemplo: programação linear inteira [Mathar and Niessen 2000], algoritmo evolucionário [Zimmermann et al. 2003] e algoritmos genéticos tais como o proposto por Jin Kyu Han et al. (2001). Abordagens que utilizam algoritmos genéticos são bastante utilizadas nesse tipo de problema em que se quer otimizar o posicionamento e o número de células em um local. Neste artigo, é proposta uma abordagem que faz uso de algoritmo genético utilizando a meta-heurística BRKGA (*biased random-key genetic algorithm*). Através do uso do BRKGA, foram atingidos resultados próximos dos ótimos nos mais variados casos testados, levando em consideração cobertura e/ou nível de potência da ERB, simulando assim uma ampliação ou um *refarming* do serviço, conforme proposto anteriormente.

¹Dados da consultoria Teleco entre 2016 e 2017 (https://www.teleco.com.br/4g_cobertura.asp)

²Mais detalhes em <https://ri.tim.com.br/ShowApresentacao/TIM-Day-2017-Leonardo-Capdeville-CTO- apenas-em-ingles-?=2erITddKyDxF/CTWnPnSjQ==> (em inglês)

1.1. Trabalhos relacionados

Por ser um assunto altamente relevante, vários estudos podem ser encontrados abordando o tema. Eles variam em relação à abordagem adotada, aos algoritmos utilizados, aos escopos e a outros detalhes relativos ao problema. Foram analisados alguns trabalhos relacionados ao tema, que são descritos a seguir.

Jin Kyu Han et al. (2001) criaram um algoritmo genético para posicionamento de ERBs, utilizando uma representação por números reais e novos operadores genéticos. A função de aptidão considera ambos custo e cobertura, em busca de eficiência no posicionamento. O trabalho leva em conta uma cidade plana e sem objetos que possam afetar a propagação do sinal, o que causa um sinal uniforme ao redor da ERB. A seleção dos indivíduos que seguirão para a próxima geração, com exceção do indivíduo mais adaptado, que é sempre selecionado, é feita com o método de seleção “Roleta Russa”, onde a chance de um indivíduo ser selecionado é diretamente proporcional ao seu valor de adaptação. A função de aptidão proposta por Jin Kyu Han et al. (2001) foi utilizada na solução proposta sem alterações, mais detalhes a respeito dela podem ser encontrados na Seção 1.2.

Posteriormente, Cardeiro and Correia (2006) implementaram um algoritmo que objetiva melhorar a cobertura de uma rede de rádio adicionando novas ERBs, em que o objetivo não é encontrar o melhor conjunto de ERBs dentre uma série de conjuntos candidatos, mas sim encontrar possíveis localizações para as ERBs que forneçam uma melhor cobertura. Este artigo se difere de outros estudos por tratar do UMTS (*Universal Mobile Telecommunications System*) e por alocar as ERBs de um modo não uniforme em um espaço contínuo, utilizando de diferentes heurísticas para os diferentes tipos de áreas não cobertas. Como resultado, o algoritmo em questão alocou mais ERBs em locais onde haviam uma maior quantidade de tráfego e menos ERBs onde não havia tanto tráfego aparentando ser efetivo para essa abordagem.

Mais recentemente, Ahmed et al. (2012) propuseram um algoritmo genético considerando demanda de tráfego, interrupções de serviço e consumo de energia, para o posicionamento de ERBs *indoor* em um aeroporto. Tradicionalmente, a otimização de posicionamento de ERBs em ambientes *indoor* leva em conta somente uma das considerações citadas acima. Levando os 3 pontos em consideração, é possível fazer um planejamento eficiente com relação a custo e QoS (*Quality of Service*). Devido ao crescimento contínuo de fluxo de passageiros em aeroportos, soluções como essa são bem-vindas.

Algumas publicações expandiram o problema para se adequar melhor aos dias atuais, por exemplo, Chen et al. (2012) implementam uma solução baseada nos algoritmos genéticos para o ajuste do nível de potência de cada antena, além da seleção do melhor conjunto de ERBs dentre os conjuntos candidatos, com o objetivo de reduzir ao máximo o custo. Aqui as ERBs tem sua posição fixa e iniciam todas ativas e com uma potência de mesmo valor. A medida que o algoritmo permanece em execução, as ERBs são desativadas ou permanecem ativas, e sua potência é ajustada, podendo ficar maior ou menor de acordo com o resultado da função de aptidão. O algoritmo genético que foi proposto obteve resultados satisfatórios, onde valores muito próximos aos ótimos foram obtidos de modo eficiente, o que se mostra muito importante na área das SONS (*Self-Organizing Networks*), porém que precisam ser melhor investigados.

A maioria das soluções anteriores utiliza algoritmos genéticos, já Li et al. (2017)

propuseram uma abordagem que se utiliza de AIAs (*Artificial Immune Algorithms*), selecionando um conjunto com número mínimo de ERBs dentre os *sites* candidatos, com objetivo de diminuir o custo de construção e maximizar a cobertura. Para realizar o objetivo proposto, os autores utilizaram-se de uma função de aptidão que realizava o balanço de ajuste entre o raio de cobertura para cada antena a ser utilizada, e a localização de cada antena em uma área dada. A proposta em questão obteve bons resultados com uma rápida velocidade de convergência.

1.2. Definição formal do problema

Visto que diferentes problemas de posicionamento de ERBs foram encontrados na literatura, nesta seção será apresentado o problema abordado neste trabalho. Tal problema tem o nome de PLEB (Problema de Localização de Estações-Base) e é abordado pelos trabalhos de Ahmed et al. (2012), Li et al. (2017) e Carneiro and Correia (2006).

Dadas uma matriz quadrada binária M com dimensão t_{max} representando o *layout* da cidade, um inteiro K representando a quantidade máxima de ERBs que poderão ser posicionadas em M e um raio R de cobertura de cada ERB. Uma solução para o PLEB é formada por um conjunto $\mathcal{S} = \{S_1, S_2, S_3, \dots, S_\ell\}$ composto de tuplas $S_i = (x_i, y_i)$ representando ℓ ERBs ativas (note que $\ell \leq K$). Os valores x_i e y_i representam a sua posição no espaço de posicionamento — limitados pelo valor de t_{max} .

Para atingir a solução do PLEB, são estabelecidas duas funções que levam em conta as restrições consideradas anteriormente, as funções de cobertura e economia. A função de cobertura estabelece a porcentagem de área coberta com relação ao espaço de posicionamento. A área coberta é representada pela união dos raios de cobertura das ERBs de cada grupo, esta função é dada por:

$$F_t(\mathcal{S}) = \frac{\text{área coberta}}{\text{área total}}.$$

Já a função de economia retorna a proporção de ERBs desativadas em uma dada solução, esta função é dada pela seguinte equação:

$$F_e(\mathcal{S}) = \frac{K - n(\mathcal{S})}{K},$$

em que $n(\mathcal{S}) = \ell$ representa o número de ERBs ativas na solução \mathcal{S} . Por fim, para encontrar a solução ótima, a seguinte função deve ser maximizada:

$$F(\mathcal{S}) = w_t \cdot F_t(\mathcal{S}) + w_e \cdot F_e(\mathcal{S}).$$

No cálculo da função *fitness*, a influência da cobertura e da economia são considerados através dos pesos w_t e w_e , que representam a importância da cobertura e da economia na resolução do problema, respectivamente. Note que, as funções de cobertura e economia são conflitantes, isto é, a medida que o número de ERBs ativas cresce, a função de economia decresce e a função de cobertura cresce. Dessa forma, basta maximizar a função F que obtemos o melhor resultado no geral, considerando os pesos w_t e w_e — cuja soma deve ser sempre igual a 1.

A representação da cidade foi inspirada no trabalho de Jin Kyu Han et al. (2001), onde a cidade representada é plana, ou seja, todos os seus pontos estão na mesma altura.

Não existem corpos que atrapalhem a propagação do sinal, tais como prédios, casas ou estruturas espelhadas, e as torres não sofrem interferência umas das outras. Foi escolhido esse tipo de abordagem por se tratar de uma abordagem simplificada com foco na otimização de cobertura e redução de custos.

A estrutura do trabalho é organizada da seguinte forma: na Seção 2, é explicado o conceito de BRKGA e apresentado o algoritmo proposto. A Seção 3 detalha os experimentos computacionais realizados em busca de comparar o algoritmo proposto com a literatura. Por fim, a Seção 4 apresenta as conclusões e trabalhos futuros.

2. Meta-heurística BRKGA

Os algoritmos genéticos são uma abordagem bio-inspirada que trata as soluções dos problemas em que são aplicados como indivíduos em um genoma (cromossomos), utilizando-se de operadores tais como a seleção, a mutação e o cruzamento para determinar os indivíduos que sobreviveriam a certos critérios formados para maximizar ou minimizar uma função objetivo dada. Essa função objetivo é formulada de acordo com o problema adotado. Segundo Whitley (2019), os algoritmos genéticos são diferentes da maioria das outras meta-heurísticas pois exploram três ideias-chave: (i) o uso de uma população de soluções para guiar a pesquisa; (ii) o uso do operador de cruzamento que recombina duas ou mais soluções para gerar novas soluções potencialmente melhores e (iii) o gerenciamento da diversidade das soluções para sustentar a exploração no espaço de busca.

No algoritmo genético clássico propostos por Holland (1992), os cromossomos são simplesmente uma estrutura de dados, normalmente uma cadeia (*array*) de bits representando uma possível solução a ser otimizada, para o problema em questão. Os cromossomos normalmente são os parâmetros para a função objetivo. O operador de seleção, inspirado na seleção natural dos seres vivos, é o responsável por selecionar os melhores cromossomos a cada geração que serão utilizados nos operadores de cruzamento e de mutação, existem vários métodos de seleção tais como o da “roleta”, “torneio” e “ranking”. O operador de cruzamento é aplicado a um par de cromossomos retirado da população, onde seus elementos serão combinados entre si para gerar descendentes (normalmente dois), já o operador de mutação através de uma probabilidade p_m de acontecer, modifica um elemento específico do cromossomo, no caso de um *array* de bits por exemplo, ele inverte o valor do bit (se o valor é zero, ele o substitui pelo valor um).

Um algoritmo genético de chave aleatória (BRKGA) é uma meta-heurística de busca geral para encontrar soluções ótimas ou quase ótimas para problemas de otimização combinatória [Gonçalves and Resende 2011], que possui três características principais: (i) um cromossomo fixo codificado na forma de um vetor de chaves aleatórias no intervalo de 0 a 1 (fechado em zero e aberto em um), (ii) um processo evolucionário bem definido que adota cruzamento uniforme parametrizado e (iii) a inserção de cromossomos mutantes na população no lugar do operador de mutação (utilizado na literatura clássica). Além disso, um decodificador se faz necessário para mapear o cromossomo (intervalo contínuo) em uma solução válida para o problema de otimização em questão, obtendo assim o valor da função objetivo.

O BRKGA é a evolução de outro algoritmo genético, o RKGA (*random-key genetic algorithm*), que é citado pela primeira vez na literatura por Bean (1994), resolvendo problemas de otimização com soluções que podiam ser representadas como permutações.

Nesta meta-heurística, diferentemente do algoritmo genético clássico proposto por Holland (1992), cada cromossomo do algoritmo é representado como um vetor de chaves aleatórias maiores ou iguais a zero e menores que um, ou seja, um conjunto de números reais gerados de forma aleatória no intervalo de $[0, 1)$. Esses vetores de chaves aleatórias são passados a decodificadores, que nada mais são do que algoritmos determinísticos que recebem essa sequência de valores aleatórios e as transformam em soluções do problema real de otimização, que são processadas pela função objetivo, retornando assim um valor de aptidão.

As gerações são as iterações que o RKGA utiliza para evoluir a população. Após a obtenção da população P_i na i -ésima iteração, calcula-se o valor de aptidão para cada um dos indivíduos através de uma função de aptidão (função objetivo), contida no decodificador. Com isso dois grupos de indivíduos são formados, um grupo de elite E_i , com os indivíduos que possuem os melhores valores de aptidão, e um grupo de não-elite $P_i - E_i$, com o restante da população, vale ressaltar que o grupo de elite é menor do que o grupo não-elite, ou seja, $|P_i - E_i| > |E_i|$. A evolução se dá através da criação de uma nova população P_{i+1} , onde todos os indivíduos de elite da população anterior P_i são copiados sem modificação.

A mutação do RGKA se dá através da inserção de mutantes na população e possuem a mesma função dos operadores de mutação nos algoritmos genéticos clássicos que é evitar a convergência para um ótimo local, não global. Esses mutantes nada mais são do que um vetor de chaves aleatórias gerado da mesma maneira que um elemento da população inicial. A cada geração, um pequeno número de mutantes obtidos, são inseridos na nova população P_{i+1} , complementando a população de elite da população P_i já adicionada previamente. O restante dos indivíduos da nova população P_{i+1} são adquiridos através do processo de cruzamento.

No RKGA, o cruzamento ocorre entre dois indivíduos a e b selecionados de toda a população de forma aleatória. A repetição na seleção dos indivíduos para o cruzamento é permitida, sendo assim, um indivíduo pode produzir mais de um descendente na mesma geração. O cruzamento uniforme parametrizado de Spears and De Jong (1991) é utilizado, onde o j -ésimo alelo de um filho f será o j -ésimo alelo de um dos seus pais, com uma probabilidade p_1 para o primeiro indivíduo a , e com uma probabilidade $p_2 = 1 - p_1$, para o segundo indivíduo b .

O que diferencia o BRKGA do RKGA é a forma com que os pais são selecionados para a operação de cruzamento, pois no BRKGA um dos pais é selecionado, de forma aleatória, da partição da população chamada de elite E , e o outro pai é selecionado do restante da população $P - E$ (partição não-elite). Além disso, no BRKGA, a seleção de cada gene do descendente (no cruzamento) é feita através de uma probabilidade $p_e > 0,5$ para o pai da elite e uma probabilidade complementar $1 - p_e$ para o pai não-elite, gerando um indivíduo descendente.

Na Figura 1, tem-se um fluxograma que demonstra o funcionamento da meta-heurística BRKGA adaptado de Toso and Resende (2015). No começo da execução do algoritmo, a população é inicializada, ou seja, vários vetores de chaves aleatórias são gerados, esses vetores são os cromossomos a serem utilizados. Após a geração da população inicial, esses vetores de chaves aleatórias são passados para o decodificador que retorna

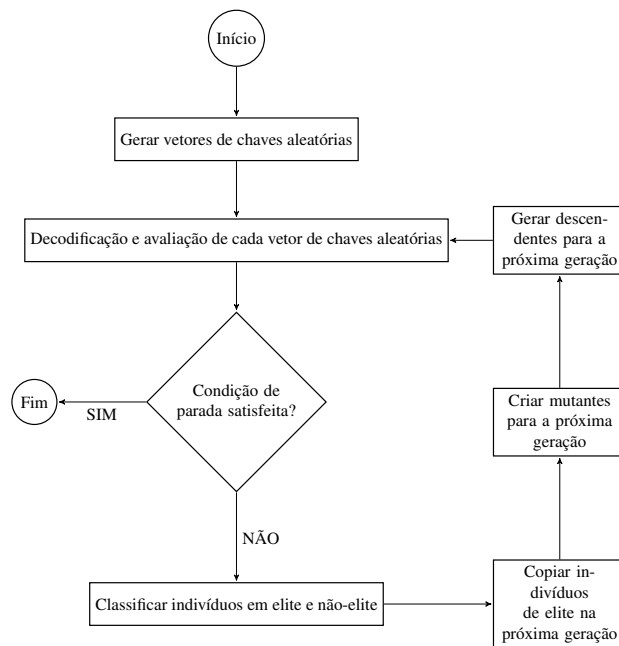


Figura 1. Fluxograma do funcionamento da meta-heurística BRKGA. Adaptado de Toso and Resende (2015)

seus respectivos valores de aptidão. Caso a condição de parada seja satisfeita, o algoritmo se encerra e o indivíduo com melhor valor de aptidão é retornado, caso isso não ocorra, os indivíduos são separados em uma partição de elite E e uma partição não-elite $P - E$, onde a população de elite é passada para a próxima geração. Os mutantes são gerados e adicionados a essa nova população, juntamente com os descendentes da operação de cruzamento. Após isso, os valores de aptidão são calculados para a nova geração e são verificadas novamente as condições de parada, encerrando ou não a execução.

2.1. Representação da solução e decodificação

As soluções são representadas por um vetor de genes C com dimensão $3K$. Como visto na definição do BRKGA, todos os genes são valores reais contidos no intervalo de 0 a 1. Para construir uma solução, o vetor C é dividido, semanticamente, em trios de genes e cada trio $c_{3i}, c_{3i+1}, c_{3i+2}$, com $i = 1, \dots, K$, representa a i -ésima ERB a ser posicionada. O primeiro gene c_{3i} indica se a i -ésima ERB está sendo utilizada ou não. Se seu valor for menor que 0,5, a ERB não está sendo utilizada. Já os genes c_{3i+1} e c_{3i+2} fornecem suas coordenadas x e y , respectivamente. Para obtê-las basta multiplicar os valores dos genes por t_{max} .

Como visto anteriormente, uma implementação de BRKGA necessita de um decodificador, de forma a converter o vetor de chaves aleatórias em uma solução do problema proposto. O decodificador é apresentado no Algoritmo 1. Inicialmente, a função $DECODIFICAR(c)$, que recebe como parâmetro o vetor C , contendo as chaves aleatórias. Na linha 4, um vetor de ERBs é criado a partir das chaves aleatórias, utilizando sempre 3 valores consecutivos de C para cada ERB. O primeiro valor diz se a ERB está ativa ou não e os seguintes referem-se às coordenadas da ERB.

Após a construção do vetor de ERBs, é criada a forma geométrica que define a união de todas as áreas cobertas. Uma ERB cobre uma área circular de raio R ao redor de

seu ponto central, que tenta aproximar a área hexagonal de cada célula da cidade descrita acima. Para calcular a área coberta utilizada na função F_t (veja Seção 1.2), é necessário considerar a união das áreas de todas as circunferências das ERBs e a união das áreas dos hexágonos que compõe a cidade simulada. Em seguida, a cobertura é determinada pela razão entre a interseção da cidade simulada com a forma geométrica criada a partir das circunferências das ERBs e a área total da cidade. Para facilitar o trabalho geométrico durante a implementação, foi utilizada a JTS (*Java Topology Suite*), uma biblioteca de código-aberto criada na linguagem Java que fornece um modelo de objeto para geometria linear plana euclidiana e funções essenciais, a versão utilizada foi a 1.16.0.

A função `areavazia()` retorna uma geometria com área 0, para propósitos de inicialização. Já, a função `ativa(ERB)` verifica se uma ERB está ativa usando seu valor α . Por fim, `criarArea(ERB)` retorna a forma geométrica equivalente às configurações da ERB passada. Ao final da função `DECODIFICAR(c)`, chama-se a função *fitness* F (ver Seção 1.2) para obter o valor de aptidão dessa configuração de ERBs.

Algorithm 1 Algoritmo de decodificação

```

1: procedure DECODIFICAR(vetor de chave aleatórias  $C$ )
2:   erbs[i]  $\leftarrow$  0,  $i = 0 \dots |C|$ 
3:   for  $i \leftarrow 0 \rightarrow |C|$  step 3 do
4:     erbs[i]  $\leftarrow$  ERB( $c[i] < 0.5, c[i + 1] * t_{max}, c[i + 2] * t_{max}$ )
5:   end for
6:   area  $\leftarrow$  areavazia() ▷ Cria uma área vazia.
7:   for  $i \leftarrow 0 \rightarrow |erbs|$  do
8:     if ativa(erbs[i]) then
9:       area = area  $\cup$  criarArea(ERB) ▷ Verifica a interseção.
10:    end if
11:  end for
12:  return  $F(erbs, area)$ 
13: end procedure

```

2.2. Parâmetros

A implementação original do BRKGA observada em Toso and Resende (2015) disponibiliza diversos parâmetros a serem ajustados em busca de adequá-lo a diferentes problemas. Os parâmetros obrigatórios são descritos a seguir:

- número n de genes em cada cromossomo: foi utilizado $n = K \cdot 3$ na solução proposta, uma vez que a representação é um vetor tupla de 3 valores, uma tupla para cada ERB;
- número de indivíduos na população P : foi escolhido de acordo com o cenário de entrada variando entre 15 e 30 de acordo com o *layout da cidade*;
- número de indivíduos na população elite E : foi utilizado $p_e = 0,2$ na solução proposta;
- número de indivíduos mutantes introduzidos na população a cada geração: foi determinado $p_m = 0,3$ na solução proposta;
- probabilidade ρ de que um filho herde o alelo de seu pai de elite: foi utilizado $\rho = 0.7$ na solução proposta.

Os valores acima foram determinados após a execução extensiva de testes de parametrização em busca do melhor resultado para o PLEB. Para outros problemas, ou mesmo variações deste, os valores podem variar.

3. Testes computacionais

Esta seção mostra a efetividade do método proposto frente ao mostrado pelo algoritmo genético (AG) proposto por Jin Kyu Han et al. (2001), para isso, testes computacionais foram realizados em um computador com processador Intel i5 4430 com 8GB de memória RAM e sistema operacional Windows 10 64-bits.

3.1. Instâncias

Para averiguar os desempenhos, os algoritmos foram executados em 3 cenários (cidades) diferentes, veja a Figura 2. O objetivo dos algoritmos é posicionar as ERBs de forma a cobrir o máximo possível da área cinza (cidade) sem consumir muitos recursos econômicos, por isso consideramos $w_e = w_t = 0.5$ em todas as instâncias testadas. A primeira cidade é composta por 7 hexágonos regulares e foi retirada de Jin Kyu Han et al. (2001), a segunda é composta por 20 hexágonos e a terceira é composta por 14 hexágonos — estas últimas são originais deste trabalho. Hexágonos são usados pois representam a unidade ideal de uma célula, circunferências vizinhas deixam espaços sobrando, e quadrados fogem muito de um modelo de propagação padrão. Os hexágonos são uma escolha ideal, pois não deixam espaços sobrando e apresentam uma diferença relativamente pequena de distância entre os pontos de suas arestas e o centro.

Tabela 1. Instâncias utilizadas.

Instância	Cidade	K	Raio (pixels)
1	1	10	62.5
2	1	7	62.5
3	2	25	62.5
4	2	23	62.5
5	3	20	62.5
6	3	21	62.5
7	1	7	70
8	2	20	70
9	3	14	70
10	1	7	50
11	2	20	50
12	3	14	50

Na Tabela 1 estão dispostas todas as instâncias utilizadas para produzir os testes de referência do AG de Jin Kyu Han et al. (2001) e da meta-heurística BRKGA. E em todas elas o valor $t_{max} = 625$. A coluna *Cidade* indica o cenário utilizado e a coluna *K* o número máximo de ERBs por *site*. Por fim, a coluna *Raio* indica o raio da ERB.

3.2. Comparação BRKGA x Algoritmo Genético

Como a implementação feita do AG não está disponível, foi realizada uma própria implementação, em Java na versão 8 (disponível em: <https://www.java.com/>), do algoritmo descrito para fazer as devidas comparações e testes. Vale ressaltar que os autores do AG afirmam que o algoritmo proposto consegue alcançar 99% de cobertura, no entanto, não se conseguiu reproduzir tais resultados.

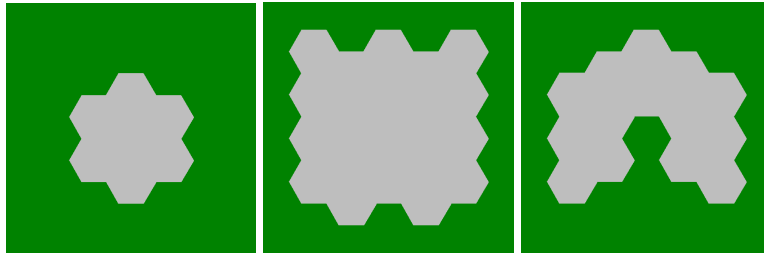


Figura 2. *Layouts* das cidades utilizadas (1, 2 e 3, respectivamente).

Tabela 2. Comparação entre BRKGA e AG.

Instância	Tempo limite	C_{max}	BRKGA		AG	
			C	Tempo (s)	C	Tempo (s)
1	80	1.0	0.99	15	0.78	6
2	80	1.0	0.99	29	0.58	32
3	250	1.0	0.97	184	0.72	12
4	250	1.0	0.98	249	0.68	10
5	150	1.0	0.99	72	0.69	189
6	150	1.0	0.99	65	0.71	15
7	80	1.0	0.99	13	0.66	93
8	250	1.0	0.99	234	0.70	3
9	150	1.0	0.99	113	0.64	67
10	80	0.77	0.76	26	0.46	70
11	250	0.77	0.74	243	0.48	22
12	150	0.77	0.75	139	0.43	18
Média		0.94	0.93	115	0.63	45

Para cada instância, após a calibração de parâmetros em ambos os algoritmos, AG e BRKGA, estes foram executados 10 vezes durante uma mesma quantidade de tempo. A calibração nos deu a seguinte configuração padrão dos parâmetros do AG, descritos em Jin Kyu Han et al. (2001): $P_{mut} = 0.3, P_n = 0.2, P_v = 0.8, \sigma_m = 1, \sigma_c = 1$. A Tabela 2 apresenta os resultados computacionais. As colunas, da esquerda para a direita representam: a instância, o tempo limite para ambos os algoritmos, a cobertura ótima para essa instância, a média da cobertura média encontrada pelo BRKGA (função *fitness F*), o tempo médio de convergência do BRKGA, a média da cobertura média encontrada pelo AG (função *fitness F*), o tempo médio de convergência do AG, respectivamente. Pode-se observar que, apesar do algoritmo genético convergir, em média, em 45 segundos, o valor de cobertura C que ele alcança, em média, é muito baixo, 63% de um máximo de 94%. Enquanto que o BRKGA alcança, em média, 93% em 115 segundos, ou seja, quase o valor máximo de cobertura média (94%). Além disso, em todas as instâncias o AG fica preso em ótimos locais e o BRKGA obtém uma melhor cobertura.

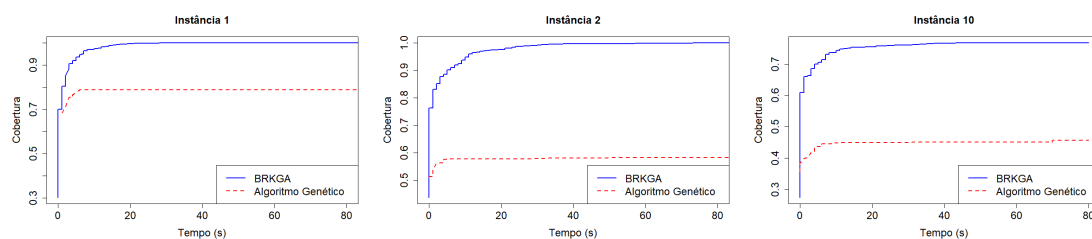


Figura 3. Gráficos de evolução da cobertura ao decorrer do tempo

Outros experimentos foram feitos para evidenciar a superioridade do BRKGA. A Figura 3 mostra a evolução da cobertura por ambos os algoritmos ao decorrer do tempo. É visível que o AG converge, em média, em poucos segundos. O BRKGA em todos os momentos apresentou uma cobertura superior a do AG e continuou evoluindo até o término de sua execução. Se o tempo limite fosse indefinido, o BRKGA atingiria a cobertura ótima C_{max} , que em algumas das instâncias vale 1, mas em outras, não.

4. Conclusões e trabalhos futuros

Neste trabalho foi apresentada uma solução para o PLEB em um cenário geográfico plano, utilizando-se da meta-heurística BRKGA, com o objetivo de maximizar a cobertura e minimizar o custo.

Foram realizados testes computacionais em um conjunto de instâncias da literatura para a avaliação da meta-heurística BRKGA onde constatou-se que o mesmo conseguia chegar sempre nos resultados ótimos ou próximos do ótimo em termos de cobertura e custo para as condições em que foi sujeito, sempre em um tempo hábil relativamente pequeno. Quando comparado a um AG proposto na literatura, no conjunto de experimentos realizados, a meta-heurística BRKGA mostrou-se superior, sempre alcançando os resultados do algoritmo clássico em um tempo menor e os superando em termos de otimalidade. Com isso notou-se que a meta-heurística BRKGA tem um desempenho melhor tanto em termos de qualidade da solução, quanto em termos de performance.

Como trabalhos futuros, serão buscadas novas formas de acelerar o método proposto para obtenção de uma melhor performance, além de se buscar novas variantes do PLEB, como por exemplo, levar em consideração espaços geográficos 3D ao invés do espaço geográfico plano considerado, interferência entre antenas, ou ainda novas variáveis de otimização, além do custo e cobertura.

Referências

- Ahmed, I. E., Qazi, B. R., and Elmirghani, J. M. H. (2012). Base stations locations optimisation in an airport environment using genetic algorithms. In *2012 8th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pages 24–29.
- Bean, J. C. (1994). Genetic algorithms and random keys for sequencing and optimization. *ORSA Journal on Computing*, 6(2):154–160.
- Boccardi, F., Heath, R. W., Lozano, A., Marzetta, T. L., and Popovski, P. (2014). Five disruptive technology directions for 5g. *IEEE Communications Magazine*, 52(2):74–80.

- Cardeiro, J. and Correia, L. M. (2006). Optimisation of base station location in umts-fdd for realistic traffic distributions. In *2006 IEEE 17th International Symposium on Personal, Indoor and Mobile Radio Communications*, pages 1–5.
- Chen, Q. B., Wang, Y., Liu, Z. J., and Liang, C. C. (2012). A new intelligent approach based on genetic algorithm for finding optimal base stations configuration in the dynamic scenario used in son. In *Frontiers of Manufacturing and Design Science II*, volume 121 of *Applied Mechanics and Materials*, pages 4325–4329. Trans Tech Publications Ltd.
- Gao, Z., Dai, L., Mi, D., Wang, Z., Imran, M. A., and Shakir, M. Z. (2015). Mmwave massive-mimo-based wireless backhaul for the 5g ultra-dense network. *IEEE Wireless Communications*, 22(5):13–21.
- Ge, X., Tu, S., Mao, G., Wang, C., and Han, T. (2016). 5g ultra-dense cellular networks. *IEEE Wireless Communications*, 23(1):72–79.
- Gonçalves, J. F. and Resende, M. G. C. (2011). Biased random-key genetic algorithms for combinatorial optimization. *Journal of Heuristics*, 17(5):487–525.
- Holland, J. H. (1992). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. MIT Press, Cambridge, MA, USA.
- Jin Kyu Han, Byoung Seong Park, Yong Seok Choi, and Han Kyu Park (2001). Genetic approach with a new representation for base station placement in mobile communications. In *IEEE 54th Vehicular Technology Conference. VTC Fall 2001. Proceedings (Cat. No.01CH37211)*, volume 4, pages 2703–2707 vol.4.
- Li, L., Ma, B., Jia, Z., and Si, Y. (2017). Base station locations optimization in lte using artificial immune algorithm. In *2017 10th International Symposium on Computational Intelligence and Design (ISCID)*, volume 1, pages 165–168.
- Mathar, R. and Niessen, T. (2000). Optimum positioning of base stations for cellular radio networks. *Wireless Networks*, 6(6):421–428.
- Skouby, K. E. and Lynggaard, P. (2014). Smart home and smart city solutions enabled by 5g, iot, aai and cot services. In *2014 International Conference on Contemporary Computing and Informatics (IC3I)*, pages 874–878.
- Spears, W. and De Jong, K. (1991). On the virtues of parametrized uniform crossover.
- Toskala, A., Holma, H., Pajukoski, K., and Tiitola, E. (2006). Utran long term evolution in 3gpp. In *2006 IEEE 17th International Symposium on Personal, Indoor and Mobile Radio Communications*, pages 1–5.
- Toso, R. and Resende, M. (2015). A c++application programming interface for biased random-key genetic algorithms. *Optimization Methods and Software*, 30.
- Vieira, R. D., Paiva, R. C. D., Hulkkonen, J., Jarvela, R., Iida, R. F., Saily, M., Tavares, F. M., and Niemela, K. (2010). Gsm evolution importance in re-farming 900 mhz band. In *2010 IEEE 72nd Vehicular Technology Conference - Fall*, pages 1–5.
- Whitley, D. (2019). *Next Generation Genetic Algorithms: A User's Guide and Tutorial*, pages 245–274. Springer International Publishing, Cham.
- Zimmermann, J., Höns, R., and Mühlenbein, H. (2003). Encon: An evolutionary algorithm for the antenna placement problem. *Computers Industrial Engineering*, 44:209–226.