

**iscte**

**INSTITUTO  
UNIVERSITÁRIO  
DE LISBOA**

Departamento de Ciências e Tecnologias da Informação

**Sistema para optimização da  
estimativa de relações de vizinhança  
em redes móveis baseado na  
cobertura geográfica**

Paulo Vítor Mateus de Pina

Dissertação submetida como requisito parcial para obtenção do grau  
de

**Mestre em Engenharia Informática**

**Orientador**

Prof. Dr. Pedro Sebastião, Professor Auxiliar

ISCTE-IUL

**Co-Orientador**

Prof. Dr. Lúcio Studer Ferreira, Professor Auxiliar

INESC-ID

Agosto 2020



*"Everything we need is already here."*

Porter Robinson



# Resumo

Com o crescimento dos sistemas de comunicação móveis celulares heterogêneos, a procura por largura de banda móvel aumenta e, conseqüentemente, um bom desempenho de *Handovers* (HOs) traz alguns desafios. Estes desafios são amplificados na presença de muitas células de características diferentes numa dada localidade, criando confusão no sistema uma vez que um dado dispositivo nem sempre se liga à célula ótima num contexto de otimização, degradando a qualidade do serviço e potencialmente aumentando custos operacionais. Neste sentido, para garantir a otimização da ligação dos dispositivos com as estações base, é necessário desenvolver modelos de suporte à decisão para as melhores vizinhas de cada célula em diferentes contextos para que os HOs sejam feitos da forma mais ótima possível.

Soluções como *Automated Neighbour Relations* (ANR) em redes 4G tomam em consideração estatísticas de HOs para escolher as melhores vizinhas, mas falham em tomar em consideração a cobertura de sinal e células planeadas, o que dificulta o planeamento de redes e esconde ineficiências no serviço. Assim, esta dissertação propõe um sistema que preenche estas duas lacunas através da estimativa de relações de vizinhança em redes móveis baseada na cobertura geográfica, suportando também coberturas simuladas de células ainda por instalar. O sistema proposto funciona num ambiente *cloud* automatizado e incorporado no ecossistema do *Metric Software as a Service* (SaaS), complementando o ANR com listas de vizinhas de elevada qualidade que podem ser ainda mais otimizadas por este, mitigando os problemas mencionados anteriormente.

**Palavras-chave:** redes auto-organizadas, relações de vizinhança, serviços *cloud*, planeamento de redes móveis.



# *Abstract*

With the growth of heterogeneous cellular mobile communications systems, the demand for mobile bandwidth increases and, consequently, good Handover (HO) performance brings some challenges. These challenges are amplified by the presence of several cells of different characteristics in certain locations, creating confusion in a system where a user device doesn't always connect to the optimal cell in an optimization context, degrading quality of service and potentially increasing operational costs. In that sense, to guarantee optimal device-base station connections, it's necessary to develop decision-support models to know what the best neighbours are for each cell in different contexts so that HOs are made in the most optimal ways.

Solutions such as Automated Neighbour Relations (ANR) in 4G networks take valuable HO statistics into consideration to decide which neighbours are best, but they fail to consider signal coverage and planned cells, which in turn makes network planning more difficult and hides inefficiencies in the service. Therefore, this dissertation proposes a system that fills these gaps through the estimation of neighbour relations based on geographical coverage, also supporting simulated coverages from inactive cells. The proposed system works in an automated cloud-based environment integrated in the Metric Software as a Service (SaaS) package, complementing the ANR with high-quality neighbour cell lists which can be further optimized by the latter, mitigating the aforementioned problems.

**Keywords:** self-organized networks, neighbour relations, cloud services, mobile network planning.





# *Agradecimentos*

Gostaria de agradecer à minha família por me suportarem durante toda a Licenciatura e o Mestrado, fazendo assim a minha participação neste projeto possível. Aos meus orientadores, Prof. Dr. Pedro Sebastião e Prof. Dr. Lúcio Studer Ferreira, ao líder técnico do projeto Nuno Valente, aos meus colegas André Godinho, Daniel Fernandes, Diogo Clemente e Rodrigo Cortesão e ao pessoal da Multivision pelo inigualável e excelente apoio prestado.

Em particular, sem nenhuma ordem específica: Miguel Rodrigues, Ricardo Silva, Miguel Nobre, Ângelo Xavier, Gonçalo Afonso, Nuno Lopes, David Taveira, Daniel Negreiros, Diogo Felix, Miguel Lourenço, Andreas W. Neitzel, Christian Phillipe Neitzel, Michelle Schmidt, Ariana Gonçalves, Tiago Almeida, Francisco Silva, Diogo Oliveira, Carson Downen, António Ferreira, Pedro Miranda, Erik Cruz, e aos meus outros colegas do curso e ao pessoal de Mortágua por me terem acompanhado, aturado, etc. de uma forma ou de outra durante todo este tempo.



# Conteúdo

<b>Resumo</b>	<b>v</b>
<b>Abstract</b>	<b>vii</b>
<b>Agradecimentos</b>	<b>ix</b>
<b>Lista de Figuras</b>	<b>xii</b>
<b>Lista de Algoritmos</b>	<b>xiv</b>
<b>Lista de Tabelas</b>	<b>xv</b>
<b>Abreviaturas</b>	<b>xvi</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Enquadramento . . . . .	1
1.2 Motivação . . . . .	1
1.3 Objetivos . . . . .	2
1.4 Contribuições . . . . .	3
1.5 Estrutura . . . . .	5
<b>2 Estado da Arte</b>	<b>7</b>
2.1 Sistemas para redes celulares . . . . .	7
2.2 Gestão de redes celulares . . . . .	9
2.3 Planeamento celular . . . . .	12
2.3.1 Cobertura celular . . . . .	12
2.3.2 Redes heterogéneas . . . . .	15
2.3.3 Vizinhança celular e <i>handover</i> . . . . .	17
2.4 Criação de listas de células vizinhas . . . . .	20
<b>3 Modelos e Algoritmos</b>	<b>23</b>
3.1 Análise de requisitos e casos de uso . . . . .	23

3.2	Visão geral do padrão de trabalho . . . . .	25
3.3	Algoritmos de identificação de interseção de grelhas de células . . .	28
3.3.1	Método iterativo . . . . .	29
3.3.2	Método recursivo com <i>Quad Tree</i> . . . . .	32
3.4	Algoritmos de identificação de vizinhas . . . . .	34
3.4.1	Algoritmo iterativo de agregação de células e as suas coberturas . . . . .	34
3.4.2	Algoritmo iterativo de agregação de pixels com células . . .	36
<b>4</b>	<b>Implementação e Integração do Sistema</b>	<b>39</b>
4.1	Tecnologias e ferramentas utilizadas . . . . .	39
4.1.1	Tecnologias . . . . .	40
4.1.2	Ferramentas . . . . .	41
4.2	Implementação do padrão de trabalho . . . . .	42
4.3	Integração com a ferramenta Metric SaaS . . . . .	46
4.4	Testes de aceitação e de integração no Metric . . . . .	46
4.4.1	Teste à identificação de interseção de grelhas . . . . .	47
4.4.2	Teste à identificação de vizinhas . . . . .	49
<b>5</b>	<b>Análise de Resultados</b>	<b>51</b>
5.1	Cenários . . . . .	51
5.1.1	Comparação de desempenho dos algoritmos de interseção . .	52
5.1.2	Cenário pequeno de vizinhanças . . . . .	52
5.1.3	Cenário grande de vizinhanças . . . . .	54
5.2	Resultados . . . . .	55
5.2.1	Comparação de desempenho dos algoritmos de interseção . .	56
5.2.2	Análise de desempenho no cenário pequeno de vizinhas . . .	59
5.2.3	Análise de desempenho no cenário grande de vizinhanças . .	62
<b>6</b>	<b>Conclusões</b>	<b>69</b>
	<b>Referências</b>	<b>71</b>

# Lista de Figuras

Figura 2.1: Visualização da arquitetura do Metric. . . . .	12
Figura 2.2: Os modelos principais de cobertura/planeamento celular. . . . .	13
Figura 2.3: Cobertura celular: (a) - Fictícia; (b) - Ideal; (c) - Real . . . . .	14
Figura 2.4: Exemplo de uma <i>hetnet</i> , em que células de maior dimensão co- existem com outras de menor dimensão [37]. . . . .	15
Figura 2.5: Representação estimada de quando um <i>handover</i> é feito, dado uma transição da célula A para a B. . . . .	18
Figura 3.1: Os casos de uso do sistema na perspectiva de um operador de rede.	25
Figura 3.2: A arquitetura do sistema integrado no Metric. . . . .	26
Figura 3.3: Um exemplo de uma grelha de pixels com uma estimacão da cobertura de uma célula. . . . .	27
Figura 3.4: Representação das coberturas das células como retângulos. . . . .	29
Figura 3.5: Diagrama que exemplifica o funcionamento do algoritmo iterativo.	29
Figura 3.6: Exemplo fictício de uma Quad Tree e as células inseridas nela como coberturas retangulares. . . . .	33
Figura 3.7: Diagrama do algoritmo de agregação de células e as suas cober- turas. . . . .	34
Figura 3.8: Diagrama do algoritmo de agregação de pixels com células. . . . .	36
Figura 4.1: Padrão de trabalho integrado com o AWS. . . . .	43
Figura 4.2: O sistema integrado no Metric. . . . .	47
Figura 5.1: Excerto do mapa do Metric que mostra a célula escolhida (mar- cada em amarelo) e outras células a testar. . . . .	53
Figura 5.2: Excerto do mapa do Metric, mostrando as duzentas e vinte e duas células. . . . .	55
Figura 5.3: Demonstração das curvas de desempenho de acordo com a Ta- bela 5.1 . . . . .	57

Figura 5.4: As vizinhas recomendadas pelo ANR são assinaladas por um contorno colorido e o seu rácio de HOs bem sucedidos está à direita. . . . .	60
Figura 5.5: Demonstrações da <i>cellA</i> , <i>cellC</i> and <i>cellH</i> no mapa do Metric. . . . .	62
Figura 5.6: Dispersão entre as vinte células de topo do mais alto ao mais baixo rácio de HOs para dentro bem sucedidos. . . . .	65
Figura 5.7: Dispersão entre as vinte células de topo do mais alto ao mais baixo rácio de HOs para fora bem sucedidos. . . . .	65
Figura 5.8: Dispersão entre as vinte células de topo do mais alto ao mais baixo rácio de cobertura partilhada a -120 dBm. . . . .	66
Figura 5.9: Dispersão entre as vinte células de topo do mais alto ao mais baixo rácio de cobertura partilhada a -100 dBm. . . . .	66
Figura 5.10: Dispersão entre as vinte células de topo do mais alto ao mais baixo rácio de cobertura partilhada a -80 dBm. . . . .	67
Figura 5.11: Dispersão entre as vinte células de topo do mais alto ao mais baixo rácio de cobertura partilhada a -60 dBm. . . . .	67

# Lista de Algoritmos

1	Cálculo de interseções à força bruta . . . . .	31
2	Cálculo de áreas de cobertura de cada célula . . . . .	35
3	Cálculo de vizinhos . . . . .	36
4	Cálculo de vizinhos de cada célula . . . . .	37

# Lista de Tabelas

Tabela 2.1:	Síntese das diferentes gerações tecnológicas. . . . .	9
Tabela 4.1:	Exemplo de uma instância das bordas da grelha (B2.2). . . . .	44
Tabela 4.2:	Exemplo de uma lista das interseções das células (B3.1). . . . .	44
Tabela 4.3:	Exemplo de uma instância do mapa-mundo (B4.1), em que cada "t" representa um <i>tab</i> . . . . .	45
Tabela 4.4:	Exemplo da lista de vizinhas (B5.1). . . . .	45
Tabela 4.5:	Testes e resultados previstos para a identificação de interseções.	48
Tabela 4.6:	Testes e resultados previstos para a identificação de vizinhas. . .	49
Tabela 5.1:	Demonstração dos resultados da contagem de iterações. . . . .	57
Tabela 5.2:	Comparação da contagem de vizinhas da <i>cellA</i> pelo sistema im- plementado e a distribuição de tráfego de HO de acordo com o ANR do operador. . . . .	61





# Abreviaturas

<b>ANACOM</b>	<b>A</b> utoridade <b>N</b> acional de <b>C</b> omunicações
<b>ANR</b>	<b>A</b> utomated <b>N</b> eighbour <b>R</b> elations
<b>AWS</b>	<b>A</b> mazons <b>W</b> eb <b>S</b> ervices
<b>CSV</b>	<b>C</b> omma <b>S</b> eparated <b>V</b> alue
<b>dBm</b>	<b>d</b> ecibel- <b>m</b> illiwatts
<b>EB</b>	<b>E</b> stação <b>B</b> ase
<b>EC2</b>	<b>E</b> lastic <b>C</b> omputing <b>C</b> loud
<b>EDGE</b>	<b>E</b> nhanced <b>D</b> ata <b>R</b> ates for <b>G</b> SM <b>E</b> volution
<b>eNB</b>	<b>L</b> TE <b>E</b> volved <b>N</b> ode <b>B</b>
<b>E-UTRAN</b>	<b>E</b> volved- <b>U</b> MTS <b>T</b> errestrial <b>R</b> adio <b>A</b> ccess <b>N</b> etwork
<b>GPRS</b>	<b>G</b> lobal <b>P</b> acket <b>R</b> adio <b>S</b> ervice
<b>GSM</b>	<b>G</b> lobal <b>S</b> ystem for <b>M</b> obile <b>C</b> ommunications
<b>HetNet</b>	<b>H</b> eterogeneous <b>N</b> etwork
<b>HO</b>	<b>H</b> andover
<b>HPSA</b>	<b>H</b> igh <b>S</b> peed <b>P</b> acket <b>A</b> ccess
<b>HSDPA</b>	<b>H</b> igh <b>S</b> peed <b>D</b> ownlink <b>P</b> acket <b>A</b> ccess
<b>HSUPA</b>	<b>H</b> igh <b>S</b> peed <b>U</b> plink <b>P</b> acket <b>A</b> ccess
<b>Intra-RAT</b>	<b>I</b> ntra- <b>R</b> adio <b>A</b> ccess <b>T</b> echnologies
<b>JSON</b>	<b>J</b> avascript <b>O</b> bject <b>N</b> otation
<b>KPI</b>	<b>K</b> ey <b>P</b> erformance <b>I</b> ndicator
<b>LTE</b>	<b>L</b> ong <b>T</b> erm <b>E</b> volution
<b>NAMTS</b>	<b>N</b> ippon <b>A</b> utomatic <b>M</b> obile <b>T</b> elephone <b>S</b> ystem
<b>NCL</b>	<b>N</b> eighbour <b>C</b> ell <b>L</b> ist
<b>NR</b>	<b>N</b> eighbour <b>R</b> elation

<b>NRT</b>	<b>N</b> eighbour <b>R</b> elation <b>T</b> able
<b>OPEX</b>	<b>O</b> perational <b>E</b> xpenditure
<b>OSS</b>	<b>O</b> perations <b>S</b> upport <b>S</b> ystems
<b>RAN</b>	<b>R</b> adio <b>A</b> ccess <b>N</b> etworks
<b>S3</b>	<b>S</b> imple <b>S</b> torage <b>S</b> ystem
<b>SaaS</b>	<b>S</b> oftware <b>a</b> s <b>a</b> <b>S</b> ervice
<b>SINR</b>	<b>S</b> ignal-to- <b>I</b> nterference-plus- <b>N</b> oise <b>R</b> atio
<b>SON</b>	<b>S</b> elf- <b>O</b> rganized <b>N</b> etwork
<b>SPM</b>	<b>S</b> tandard <b>P</b> ropagation <b>M</b> odel
<b>TSV</b>	<b>T</b> ab <b>S</b> eparated <b>V</b> alues
<b>UE</b>	<b>U</b> ser <b>E</b> quipment
<b>UMTS</b>	<b>U</b> niversal <b>M</b> obile <b>T</b> elecom <b>S</b> ystem
<b>W-CDMA</b>	<b>W</b> ideband <b>C</b> ode <b>D</b> ivision <b>M</b> ultiple <b>A</b> ccess
<b>XML</b>	<b>E</b> xtensible <b>M</b> arkup <b>L</b> anguage

# Capítulo 1

## Introdução

### 1.1 Enquadramento

O trabalho presente nesta dissertação está enquadrado no projeto *OptiNET-5G - Planning and Optimization Framework of 5G Heterogeneous Networks in a Cloud Environment* [25], um projeto da Multivision Lda. financiado pelo Centro 2020 e Portugal 2020 para o planeamento otimizado de redes de telecomunicação, e integrado na plataforma Metric SaaS da mesma organização.

### 1.2 Motivação

Cada rede móvel apresenta várias células interconectadas que servem vários tipos de *user equipments* (UEs). Estas células têm um alcance limitado, e assim que um UE esteja fora desse, elas não podem servir-los adequadamente. Nestes casos, um UE deve ser capaz de se ligar a outra célula para continuar a usufruir dos serviços da rede sem muita interrupção; esta operação designa-se por Handover (HO).

A questão então torna-se na seguinte: como é que um UE sabe para qual célula deve mudar? Enquanto que existem formas de procurar "cegamente" por qualquer sinal válido, essas não são muito eficientes em certos cenários, uma vez que se

podem ligar a células defeituosas. A situação complica-se no domínio das redes heterogêneas (HetNets), que misturam células de várias frequências, tecnologias e tamanhos nas mesmas redes para vários objetivos.

Uma forma de otimizar HOs é manter listas de vizinhas na base de dados de uma célula e informar os UEs ligados para que estes saibam a qual célula se devem ligar, e assim o problema será responder à questão de como é que essas listas devem ser feitas, que é a motivação presente neste trabalho.

Uma solução comum para o problema dos HOs ótimos é um chamado *Automatic Neighbour Relations* (ANR), que funciona através de um método que, para cada célula, compila uma lista de contagens de HOs bem-sucedidos contra todos os HOs entre essa e as outras células nas redondezas. As células vizinhas com o maior rácio de sucessos são consideradas as “melhores”.

O ANR não considera aspetos de cobertura topográfica (o sinal partilhado entre duas células pode apenas ser “bom” num espaço muito pequeno quando o espaço útil é muito maior, acabando por desperdiçar o HO) e células planeadas (ou seja, que ainda não foram instaladas), pelo que não é suficiente para garantir uma excelente configuração e, conseqüentemente, excelente desempenho em HOs.

### 1.3 Objetivos

Esta dissertação tem como objetivo propôr um modelo eficiente e simples para um sistema enquadrado na disciplina do planeamento de redes celulares, utilizando uma *pipeline* de funções *serverless* na *cloud* para fornecer um serviço de criação de listas de vizinhas a um grande número de clientes em simultâneo. Este sistema usa quatro módulos, porém esta dissertação focar-se-à em dois que foram desenvolvidos pelo autor:

- Identificação de interseções: Com células definidas como quadrados numa grelha bidimensional, identifica e guarda referências àquelas que se interse-tam.

- Identificação de células vizinhas: Dado um mundo em pixels com sinais de células e um filtro de sinal  $X$  dBm, identifica como vizinhas as células com sinal acima de  $X$  dBm, guardando o número de ocorrências dessas identificações para cada célula.

O primeiro módulo é dependente de um que lhe possa dar essas grelhas, e o seu *output* contribui para outro sistema que calcula o mapa mundo para o segundo módulo. O sistema completo enquadra-se no sistema Metric SaaS, uma plataforma de planeamento/monitorização de redes em cloud da Multivision, em que é usado em produção.

As principais questões de investigação que este trabalho pretende responder são as seguintes:

- Será que o sistema é capaz de planear a vizinhança de várias células ao mesmo tempo?
- Como é que um sistema de planeamento de vizinhanças beneficia do uso de serviços cloud?
- No contexto de redes auto-organizadas, como é que um sistema de planeamento de vizinhas contribui para os seus paradigmas?
- Será necessário o uso de variáveis dependentes de uma geração tecnológica (i.e., 2G, 3G, 4G, 5G) para fazer boas listas de vizinhas?

## 1.4 Contribuições

O trabalho apresentado nesta dissertação contribui principalmente para o projeto OptiNet-5G da Multivision Lda., desenvolvido entre outubro de 2018 e novembro de 2019. Entre as contribuições principais constam as seguintes publicações:

- Pina, P., Fernandes, D., Clemente, D., Soares, G., & Sebastião, P. (2019). Cloud-based Implementation of an Automatic Pixel-based Neighbour Identification System for Cellular Networks. *22nd International Symposium on Wireless Personal Multimedia Communications*.
- Pina, P.M, Godinho, A.F., Fernandes, D.F.S, Clemente, D., Sebastião, P., Soares, G, Ferreira, L.S., “Automatic Coverage Based Neighbour Estimation System: A Cloud-Based Implementation”; *in IEEE Access, Apr. 2020, Vol. 8: pp 69671-69682, DOI 10.1109/ACCESS.2020.2986755*
- Godinho, A.F., Fernandes, D., Clemente, D., Soares, G, Pina, P., Sebastião, P., Correia, A., Ferreira, L.S., “A novel way of automatically plan cellular networks supported by linear programming and cloud computing”; *in MDPI Applied Sciences: Electrical, Electronics and Communications Engineering, Vol. 10, 3072, Apr. 2020, DOI 10.3390/app10093072*

Os entregáveis relevantes estão listados a seguir:

- Multivision, *Módulos de optimização técnica, económica e ambiental*, entregável D4.1 Projeto OptiNet-5G, P2020, Lisboa Portugal, Fev. 2019 ([www.multivision.pt](http://www.multivision.pt)).
- Multivision, *Funcionamento dos módulos de optimização técnica, económica e ambiental*, entregável D4.2 Projeto OptiNet-5G, P2020, Lisboa Portugal, Fev. 2019 ([www.multivision.pt](http://www.multivision.pt)).
- Multivision, *Testes e avaliação dos módulos*, entregável D6.1 Projeto OptiNet-5G, P2020, Lisboa Portugal, Abr. 2019 ([www.multivision.pt](http://www.multivision.pt)).
- Multivision, *Testes e avaliação intermédia da integração dos módulos*, entregável D6.2 Projeto OptiNet-5G, P2020, Lisboa Portugal, Abr. 2019 ([www.multivision.pt](http://www.multivision.pt)).
- Multivision, *Teste e avaliação final e aceitação do protótipo produzido*, entregável D6.3, Projeto OptiNet-5G, P2020, Lisboa Portugal, Ago. 2019 ([www.multivision.pt](http://www.multivision.pt)).

Os protótipos desenvolvidos no âmbito do projeto, eventualmente integrados na plataforma Metric, foram os seguintes:

- Protótipo *standalone* para o cálculo de interseções entre coberturas de células;
- Protótipo *standalone* para a construção de listas de células vizinhas;
- *Pipeline* integrada na *Amazon Web Services*.

## 1.5 Estrutura

Esta dissertação está estruturada pelos Capítulos seguintes:

- **Capítulo 1: Introdução**  
Apresenta o enquadramento, a motivação e os objetivos da tese, bem como as perguntas de investigação principais e a sua estrutura.
- **Capítulo 2: Estado da Arte**  
Composta pela base teórica da tese e do projeto relacionado, introduzindo conceitos fundamentais de telecomunicações e de *Handovers*.
- **Capítulo 3: Modelos e Algoritmos**  
Elabora nos requisitos propostos pelo projeto, bem como os seus casos de uso, a visão geral do padrão de trabalho e as definições abstratas dos algoritmos propostos.
- **Capítulo 4: Implementação e Integração do Sistema**  
Descreve como é que o sistema foi feito na prática, com que ferramentas, com que padrão de trabalho e como é que se integrou com o Metric. Também inclui testes de aceitação para determinar o bom funcionamento do sistema desenvolvido.

- Capítulo 5: **Análise de Resultados**

Apresenta testes ao sistema com base em cenários realistas para avaliar a qualidade e utilidade dos *outputs*, bem como o desempenho do sistema.

- Capítulo 6: **Conclusões**

Inclui as deliberações finais mais importantes do projeto, as suas limitações e respostas às questões de investigação.



# Capítulo 2

## Estado da Arte

Neste Capítulo são descritas, de forma breve, as tecnologias existentes para redes móveis, iniciando com termos básicos e progressivamente evoluindo para o assunto de relações de vizinhança entre células na rede. Na Secção 2.1 abordam-se os básicos de sistemas para redes celulares; na Secção 2.2 discutem-se algumas das várias formas de gestão de redes celulares; na Secção 2.3 são elaborados os desafios e algumas soluções existentes para o planeamento celular; finalmente, na Secção 2.4 abordam-se a técnica de criação de listas de vizinhas e algumas das suas formas.

### 2.1 Sistemas para redes celulares

Os sistemas para redes celulares permitem a construção de redes de acesso radio (ou *radio access networks*, RANs), que fornecem ligações às redes de *backbone* (ou *Core Networks*) e coordenam a gestão de recursos pelos terminais de rádio. Estas têm composições dependendo da geração tecnológica de comunicação móvel, que são caracterizadas pelas suas velocidades e funcionalidades conceptualizadas.

Estes sistemas são, por defeito, limitados em capacidade pela interferência entre utilizadores a acederem à rede em simultâneo, por isso existem técnicas de acesso múltiplo que baseiam-se na reutilização de um canal, que pode ser a frequência, tempo ou código (ou *scrambling code*), dependendo da geração tecnológica;

ou seja, cada área (definida por uma célula) usa o seu próprio canal, mas áreas distantes umas das outras reutilizam esses canais [22]. Este método é necessário para conseguir servir o máximo de clientes possível e simultaneamente minimizar a interferência, tornando o fator limitativo a disponibilidade do espectro.

A primeira geração a aplicar redes celulares é normalmente denominada como 1G e apenas suportava chamadas de voz. Um exemplo conhecido de uma tecnologia desta geração é o *Nippon Automatic Mobile Telephone System* (NAMTS), criado no Japão [10]. As tecnologias 2G são caracterizadas, principalmente, pelo salto de sistemas analógicos para sistemas 100% digitais. O domínio de comunicações é limitado a uma comutação de circuitos.

O *Global System for Mobile Communications* (GSM) introduziu serviços por dados como mensagens por texto encriptadas através do SMS e o MMS. Como cada comunicação pode apenas ocupar 2 *timeslots* de cada vez, o débito máximo deste serviço é de 9.6 Kbps. O *Global Packet Radio Service* (GPRS) adicionou um domínio de comutação de pacotes através da junção de 2 *timeslots* e suporte para IPv4 com HOs. O débito máximo deste serviço é entre 40 a 50 Kbps [19]. O *Enhanced Data Rates for GSM Evolution* (EDGE) reduziu latência de comunicações ponta-a-ponta para menos de 150 ms. Empacotando 4 *timeslots*, o débito máximo pode chegar aos 236 Kbps, enquanto que com 8 chega aos 473 Kbps. No *Evolved EDGE*, uma evolução do EDGE, o débito máximo pode chegar aos 1 Mbps, rivalizando as primeiras tecnologias 3G [19].

As tecnologias 3G como o *Universal Mobile Telecom System* são caracterizadas pelo aumento do débito médio e máximo em *Downlink* e em *Uplink*, autenticação para utilizadores a quererem aceder à rede, suporte para IPv6 e para algumas de interfaces:

- *Wideband Code Division Multiple Access* (W-CDMA), com um débito máximo de 2 Mbps [1].

- *High Speed Packet Access* (HSPA), separado por *High Speed Downlink Packet Access* (HSDPA) e *High Speed Uplink Packet Access* (HSUPA), a 14.4 Mbps e a 5.7 Mbps em condições ideais [15].

O *Long Term Evolution* (LTE), baseado no UMTS/HSPA, é considerado uma tecnologia 3G. Não se deve confundir com o *LTE Advanced*, uma tecnologia 4G. As tecnologias 4G, por definição, devem suportar débitos máximos de 100 Mbps em condições de elevada mobilidade e 1 Gbps em baixa mobilidade. Perdeu-se o suporte para comutação de circuitos, baseando a comunicação puramente por comutação de pacotes IP, com suporte para Mobile IPv6 [31].

Prevê-se que as tecnologias 5G consigam obter débitos máximos de 20 Gbps a frequências muito elevadas e um aumento substancial em relação ao 4G em frequências mais baixas. Outros objetivos do 5G incluem maior eficiência energética, na performance da rede e conectividade entre vários utilizadores.

Com a largura de banda elevada típica destas gerações, serviços de voz, SMS, web, transferência de ficheiros, streaming e de geo-localização podem ser usados sem haver grandes problemas de desempenho. A Tabela 2.1 lista os serviços principais introduzidos com cada geração.

Geração	Sistema	Ano	Local	Duplex	Acesso Múltiplo	Banda [MHz]	Canal [kHz]	Modulações	Débito máximo [Mbps]	Serviços
1G	NAMTS	1978	JP	N/A	N/A	900	25	N/A	N/A	Voz
2G	GSM	1991	EU	FDD	TDMA	900, 1800	200	GMSK	0.05 (GPRS)	Voz, SMS, MMS
3G	UMTS	2002	EU, JP	FDD/TDD	CDMA	2000	5000	QPSK, 2-BPSK, 16QAM, 64QAM	Entre 384 e 84/10.8 (HSPA+)	Voz, Dados
4G	LTE	2010	SE, NO	FDD/TDD	OFDMA/SC-FDMA	800, 2600	1400 a 100000	QPSK, 16AQM, 64QAM	100 / 50	Dados

TABELA 2.1: Síntese das diferentes gerações tecnológicas.

## 2.2 Gestão de redes celulares

Consiste na análise da geografia da cobertura, densidade de potenciais utilizadores, acordos com operadores, custos da infraestrutura e apreciação dos reguladores para a criação e manutenção de redes móveis.

Gestão dos vários parâmetros e dados recolhidos da rede para efeitos de análise estatística da sua performance, eficiência, robustez e outros pontos para determinar potenciais mudanças a fazer para melhorar a rede. Devido ao elevado número de dados envolvidos no funcionamento da rede, a sua análise individual pode ser pouco conveniente, pelo que se usam *Key Performance Indicators* (KPIs) para generalizar vastos conjuntos de dados em atividades compreensíveis numa escala numérica.

Com uma monitorização compreensível, aplicam-se ajustes aos vários componentes da rede para melhorar a sua performance, concertar problemas, alterar a dimensão da cobertura, entre outros. Estes ajustes são quantificados por KPIs, que também devem ser analisados posteriormente.

As *Self-Organized Networks* (SONs) aplicam tecnologias de planeamento, monitorização e otimização de forma automática dependendo de uma série de parâmetros e *use cases*. Uma EB que se ligue a uma SON é auto-configurada de forma *plug-and-play*. A designação de uma SON pode depender da sua distribuição: [27]

- **SON Distribuída** - As funcionalidades da SON são distribuídas pelas fronteiras da rede.
- **SON Centralizada** - As funcionalidades da SON são distribuídas mais perto das estações mais importantes.
- **SON Híbrida** - Uma mistura dos dois casos anteriores.

As principais funções de uma SON são caracterizadas [27] como:

- **Auto-configurações** - Estabelecimento de ligações e atualizações do software, ajustes automáticos de parâmetros técnicos;
- **Auto-otimizações** - Adaptações para use cases diferentes através de ajustes de configurações;
- **Auto-reparações** - Detecção de estações com defeitos e ajuste de parâmetros e algoritmos para contornar o problema.

Um exemplo concreto de um sistema de auto-gestão de redes, em que esta dissertação também se insere, é o Metric [6] [24], que é uma ferramenta da Multivision Lda. baseada na *cloud*, inserida no âmbito de planeamento de redes e de apoio à decisão de otimização para redes 2G, 3G, 4G e, futuramente, 5G. Tem como principal objetivo guardar, compilar e demonstrar numa interface gráfica compreensível vários tipos de informação sobre a rede, a performance da mesma e do utilizador, com integração de KPIs, a partir da agregação de dados provenientes de *Operations Support Systems* (OSS) e monitorização de redes em que este projeto se baseia.

A motivação deste sistema vem do facto de que, dependendo do vendedor, os componentes das células de uma rede móvel podem apresentar diferenças semânticas, funcionais ou metodológicas ao nível dos seus parâmetros técnicos, o que pode dificultar o trabalho de otimização: por exemplo, numa célula 2G do vendedor X um parâmetro pode se chamar "**RxPower**", mas na célula 2G vendedor Y o parâmetro com funcionalidade equivalente chama-se "**Rx\_Power\_1**"; o Metric pretende remover estas ambiguidades providenciando uma interface comum a conceitos comuns.

O Metric também consegue construir *path loss grids* através da combinação de *Drive Tests* e modelos de propagação automaticamente afinados, tal como gerir eficientemente atividades de estações de base, que são essenciais para o planeamento automático de vizinhos; consegue separar as informações dependendo da geração das tecnologias das que foram adquiridas, *Placemarkers*, *Paths* e *Geofences*. Foi desenhado segundo o modelo SaaS e utiliza o serviço *Lambda* da *Amazon Web Services* (AWS) para o seu funcionamento. A Figura 2.1 sintetiza as funcionalidades e a arquitetura do Metric.

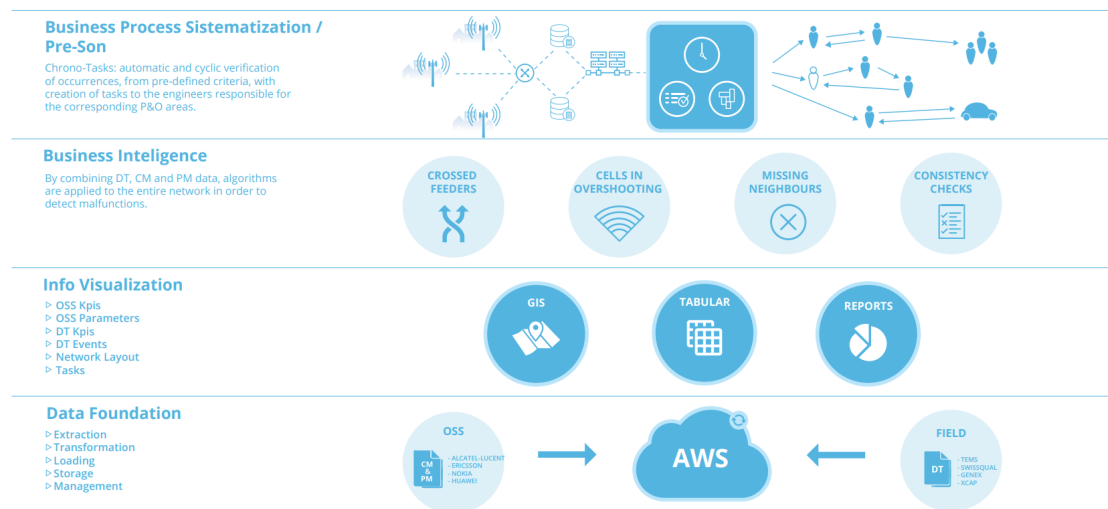


FIGURA 2.1: Visualização da arquitetura do Metric.

## 2.3 Planejamento celular

### 2.3.1 Cobertura celular

Todas as redes de comunicações móveis têm arquiteturas estritamente definidas dependendo da rede de acesso a ser usada. Em comum, estas redes usam estruturas de redes celulares.

Células são as componentes unitárias de uma rede celular e são definidas por uma Estação Base (EB), que serve a função de um transceptor entre os vários controladores que gerem as transmissões com os equipamentos dos utilizadores num raio definido e numa certa frequência. Duas células adjacentes não podem estar configuradas para transmitir na mesma frequência por questões de interferência, pelo que conjuntos de células (*Clusters*) não podem partilhar frequências entre si. Por regra, uma célula que cubra uma área maior terá menor suporte para vários utilizadores em paralelo, enquanto que uma que cubra uma área menor terá mais suporte para vários utilizadores.

O problema da estimação de cobertura tem três grandes vertentes:

- Cobertura fictícia;
- Cobertura real;

- Distribuição do tráfego.

A cobertura da área de serviço de uma célula é formada com uma Estação base no seu centro. Normalmente, na conceção de uma rede celular, usa-se um exemplar constante de um conjunto de polígonos regulares para estabelecer uma cobertura fictícia, exemplificados na Figura 2.2:

- Triângulos;
- Quadrados;
- Hexágonos.

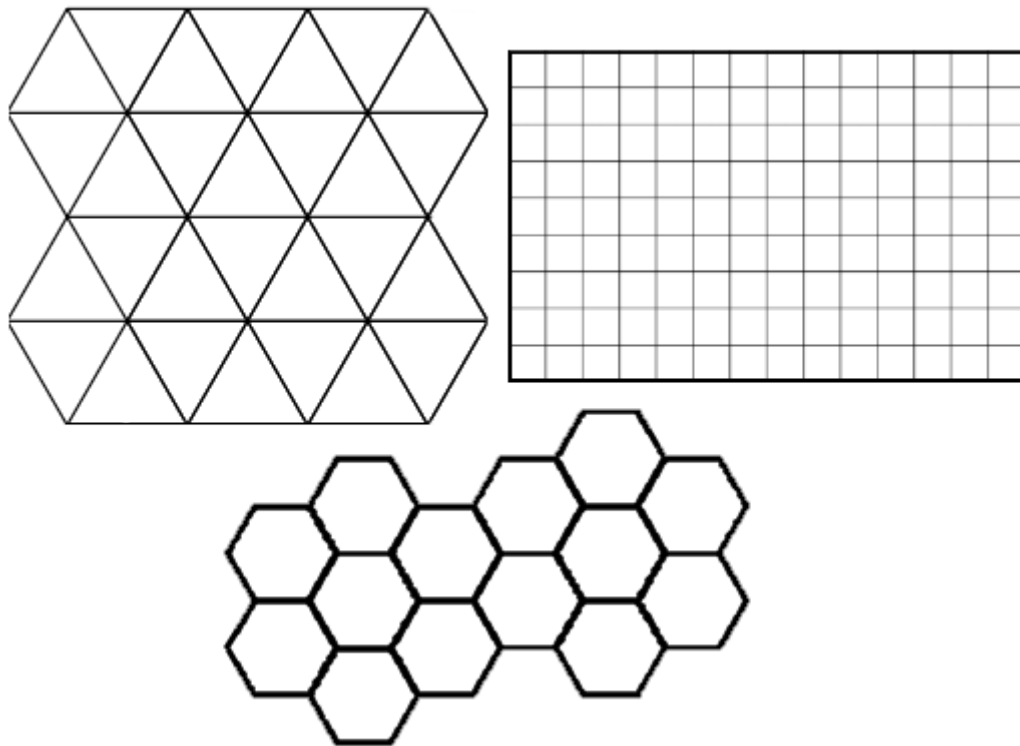


FIGURA 2.2: Os modelos principais de cobertura/planeamento celular.

Com uma cobertura fictícia, pretende-se modelar uma cobertura ideal, com o aspeto de um (ou mais) círculo(s), sem sobreposições ou lacunas. Assumindo que se quer uma cobertura completa, as sobreposições são pouco desejadas porque pode haver interferência entre as duas estações, e as lacunas também porque tratam-se de áreas que ficam sem sinal. Como o hexágono é o polígono mais próximo

a um círculo em termos de área coberta, é mais comum usar-lo sobre os outros polígonos. Na realidade, a cobertura de uma célula é afetada por vários aspetos de propagação, como a presença de obstáculos, condições atmosféricas e a curvatura da Terra, causando padrões irregulares. A Figura 2.3 demonstra um exemplo prático (com sobreposições) da cobertura ideal.

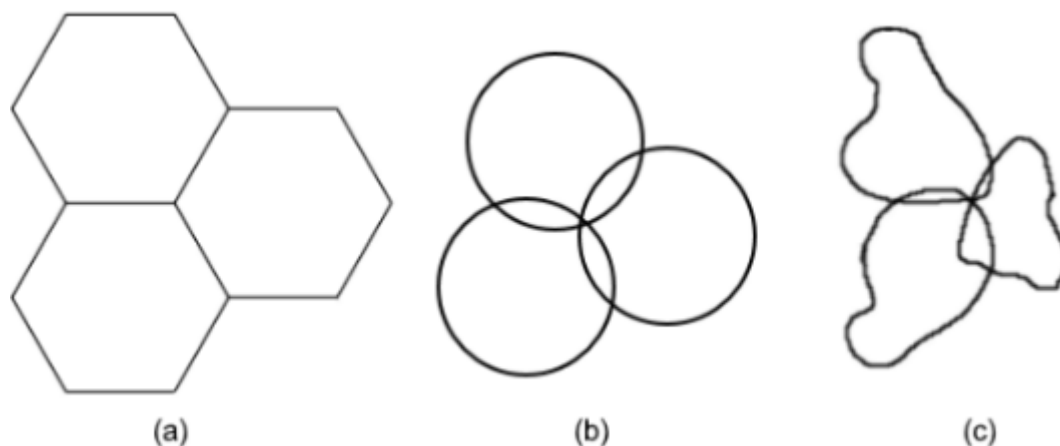


FIGURA 2.3: Cobertura celular: (a) - Fictícia; (b) - Ideal; (c) - Real

O tráfego de comunicações, na realidade, não é consistente na sua distribuição geograficamente. Como representado no início desta Secção, uma célula terá que estar comprometida entre cobrir uma área extensa ou servir o máximo de UEs que conseguir. Juntando vários *clusters* com células de dimensões diferentes pode aumentar a interferência entre as estações se as instalações não forem feitas tendo as zonas de transição entre as células em conta.

Existem vários tipos de células, definidas pelo seu raio:

- **Macro células** - Cobrem grandes áreas, maioritariamente zonas com menor densidade de utilizadores, como autoestradas e zonas rurais.
- **Micro células** - Cobrem pequenas áreas urbanas de alta densidade populacional, usando um sinal mais intenso.
- **Pico células** - Concebidas para melhorar o sinal de uma rede móvel em locais pequenos como escritórios, estações de comboio, etc.
- **Femto células** - Cobrem áreas muito pequenas como casas e lojas.



### 2.3.2 Redes heterogêneas

Todos os anos, há um aumento exponencial de tráfego nas redes móveis, e devido a esse fenômeno muitas células são instaladas em padrões densos e heterogêneos. Estas redes, também chamadas de *hetnets*, caracterizam-se pela sobreposição de estações, suportando a coexistência de sistemas (GSM, UMTS, LTE...) que apresentam diferenças significativas em alguma arquitetura, como a geração da tecnologia, o tipo de serviço que prestam, e os protocolos que usam. Um exemplo baseado nas diferenças de dimensões entre células está demonstrado na Figura 2.4.

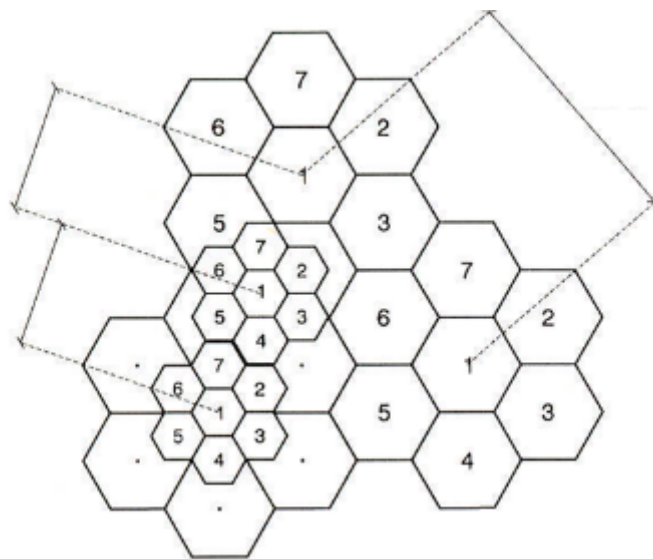


FIGURA 2.4: Exemplo de uma *hetnet*, em que células de maior dimensão coexistem com outras de menor dimensão [37].

Combinando macrocélulas numa rede multinível com outras células mais pequenas, mas mais fortes (como microcélulas, picocélulas e femtocélulas), a experiência de utilização das redes móveis pode ser melhorada até quatro vezes o desempenho comum [12]. Esta tecnologia permite:

- Maior robustez nas comunicações;
- Maior eficiência espectral;
- Maior cobertura da rede sem fios.

A cobertura de GSM e de UMTS é quase total, estas sendo garantidas por Macro células, enquanto que para o LTE só há cobertura em zonas específicas e mais habitadas, utilizando Micro células e Pico células. Todos estes sistemas coexistem entre si e, na prática, cada um foca-se em fornecer um serviço específico (por exemplo, o GSM foca-se no serviço de voz, mas o UMTS trata de *streaming*), para o utilizador a transição de um para o outro deve ser feita de forma suave e sem complicações.

A capacidade de redes heterogêneas pode ser difícil de medir, o HO de uma ligação é complexo devido à troca de protocolos a serem usados, e a interferência entre os pontos de acesso pode afetar a qualidade de serviço. A densidade natural deste tipo de redes pode causar uma densidade equivalente na superposição de áreas de cobertura, que irá exigir uma forma mais otimizada para concluir HOs, pois estes têm a tendência de serem atrasados com estas configurações, que vai negativamente afetar o desempenho prometido pelos padrões densos [4].

Estas redes são inerentemente complexas, pois o uso de *Intra-Radio Access Technologies* (Intra-RATs) e *cell splitting* (divisão de macro células para células mais pequenas de baixa energia para aumentar capacidade em áreas congestionadas) requerem métodos de alocação de frequências sofisticados para gerir capacidades altas de utilizadores e potenciais problemas na sobreposição de cobertura. Uma análise deste conceito colocou à prova três formas de alocação de frequências em redes heterogêneas [20]:

1. *Co-channel allocation*: Todas as células numa rede heterogênea partilham a mesma banda de frequência;
2. *Orthogonal frequency allocation*: Macro células e células locais são dadas bandas de frequência diferentes, não tendo nenhuma sobreposição umas com as outras;
3. *Overlapped frequency allocation*: Macro células e células locais têm uma sobreposição parcial de frequências; por exemplo, células locais que usam toda a frequência enquanto que macro células apenas usam uma parte.

Uma avaliação da capacidade do *downlink data channel system* fora então realizado, revelando que a terceira forma tinha os melhores resultados e a segunda os piores.

Como as redes heterogêneas são altamente suscetíveis a problemas com interferência, estas que fazem uso de muitas células com tamanhos diferentes precisam de alguma forma de a controlar. Algumas técnicas para o controlo dessa interferência foram propostas, que incluíam extensão de alcance de cobertura (expandir a cobertura de picocélulas para aumentar os ganhos oriundos do *cell splitting*), e duas formas diferentes de “Gestão avançada de interferência”, que eram protocolos de gestão de recursos [17]. Um deles, uma estratégia “intercelular”, coordenava interferência entre macro/pico estações base através da partição de recursos nos canais de controlo e canais de dados; no outro, uma estratégia “lentamente adaptável”, precisava de uma entidade central que negociava recursos independentemente em escalas de tempo. Estas técnicas deram melhoramentos no *throughput* do *downlink* que ajudaram redes heterogêneas a providenciar uma experiência melhor aos utilizadores da rede independentemente das suas localizações, solidificando a importância destas redes no crescimento.

### 2.3.3 Vizinhança celular e *handover*

Para manter uma ligação de um dispositivo a uma rede móvel sem haver quebras durante a transição de uma célula para a outra, deve ser possível realizar HOs. Quando um dispositivo se afasta do emissor de uma célula, o sinal que este recebe vai descer até a um certo limiar que depois despoleta a transferência de uma lista de vizinhas para o dispositivo, assim sabendo para quais as células deve fazer um HO. Um exemplo visual deste fenómeno e das suas margens de histerese encontra-se na Figura 2.5; numa mudança do equilíbrio entre potências de sinal deve ser realizada a troca para minimizar o *downtime*. Estratégias sobre como saltar cada HO ímpar podem melhorar o desempenho em média no longo prazo, mas não respondem ao problema de otimizar os próprios HOs através de, por exemplo, decidir qual EB passa a ligação para outra [4].

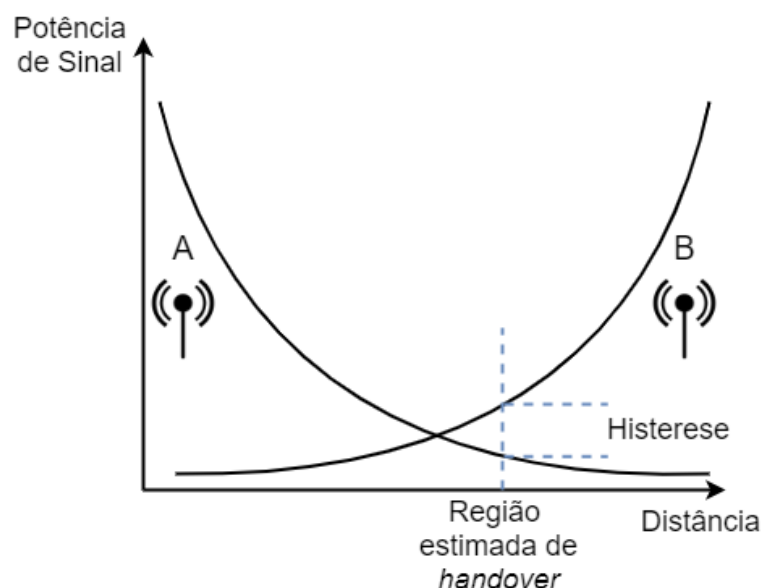


FIGURA 2.5: Representação estimada de quando um *handover* é feito, dado uma transição da célula A para a B.

Dados os conceitos anteriores, o planeamento mais importante a fazer na criação/expansão de uma rede celular é a interação entre as células vizinhas, sendo desejável alguma sobreposição para efetuar HOs. Para isso, cada tecnologia mantém listas das frequências das estações base na vizinhança de um UE, dependendo do tipo de acesso múltiplo do sistema.

Enquanto que as bandas usadas por cada tecnologia são variadas, as usáveis a um nível internacional são reguladas pela União Internacional de Telecomunicações e, a um nível nacional, pela Autoridade Nacional de Comunicações (ANACOM), causando um número escasso de frequências usáveis e obrigando uma excelente gestão de *Frequency Reuse* num grupo de *clusters*.

Não é sensata a calibração manual de inúmeras estações em conjuntos de vários *clusters* para uma boa otimização da interação entre células vizinhas, recorrendo-se assim a técnicas das SON referidas em 2.2 para efetuar o planeamento automático de vizinhos. No contexto de uma SON LTE, a configuração automática de relações de vizinhança entre células é tratada por uma função ANR, em que cada estação base tem a sua *Neighbour Relation Table* (NRT), e cada célula tem visibilidade para os seus vizinhos. A função ANR permite a deteção automática de *Neighbour Relations* (NRs), gestão da NRT, enviar atualizações à operadora

e aos gestores, etc.; tudo isto resulta em ligações, HOs e custos de manutenção significativamente melhores, pois as tarefas dispendiosas de configuração/afinação podem ser reduzidas [7].

A gestão de NRs pode ter variadas formas e muitas otimizações podem ser feitas. Esse fenómeno justifica-se devido à complexidade crescente das redes móveis, uma comunidade de utilizadores que não para de expandir e usar mais e mais serviços de terceiros, e a necessidade de reduzir custos operacionais (OPEX). Para LTE Evolved NodeBs (eNBs), que usam *Neighbour Cell Lists* (NCLs) um algoritmo fora proposto que fazia uso de distâncias físicas e interferências entre eNBs, media potências de sinal e comparava estatísticas de HOs para adicionar ou remover células de um NCL. Uma combinação boa de algoritmos é muito poderosa na criação de NCLs de alta qualidade, que são importantes para otimizar desempenho de HOs [8].

Para obter uma noção mais realista de como a mobilidade dos utilizadores afeta o desempenho de HOs em redes heterogêneas multinível, foram feitas análises a redes irregulares (com muitas células pequenas espalhadas) derivando a probabilidade de cobertura através da distribuição de SINR recebida e o rácio de HO para um dispositivo em dois casos: um em que o utilizador é imóvel (um HO não aconteceu) e outro em que o utilizador é móvel (um HO quase de certeza aconteceu). Uma simulação confirmou que quando o utilizador é móvel e o sistema é sensível a HOs, a associação de nível ótima e a probabilidade de cobertura dependem da velocidade do utilizador [30]. Isto sugere que um fator de enviesamento dependente na velocidade pode ajustar a associação de um utilizador a um nível para melhorar o desempenho do sistema.

A dimensão reduzida de uma femtocélula pode causar imprecisões, causando HOs desnecessários. Uma proposta para um algoritmo de previsão de HOs providenciou melhor desempenho do que outros algoritmos *standard* com um número consistente de HOs sem prever nenhum desnecessário [3].

## 2.4 Criação de listas de células vizinhas

Vários trabalhos sobre a criação automática e/ou a otimização de NCLs, especialmente no contexto de redes heterogêneas LTE, foram criados na década de 2010. Vários mostraram resultados positivos em muitas áreas, tais como melhoramentos no desempenho de HOs, na vida da bateria de dispositivos de utilizadores e na redução de *overhead* do sistema. Muitos destes foram alcançados cedo na década.

Uma proposta de um método de otimização automática de NCLs argumentava que o principal problema era sobre o *overhead* extra do sistema ao iterar NCLs excessivamente longas, causando falhas em ligações devido a atrasos nos HOs [26]. O trabalho envolvia medidas reais de células para autoconfigurar uma NCL inicial, que podia ser continuamente otimizada automaticamente durante as suas operações. O uso deste algoritmo precisava de algum investimento a longo prazo, mas esse iria valer a pena: cerca de 6000 medidas depois, o alvo de HO iria chegar a um rácio de sucesso de cerca de 99%. Este trabalho ajudou a provar que a automatização de otimizações de NCLs definitivamente tem uma utilidade no planeamento de redes e a sua manutenção no longo prazo, e comparado com uma solução manual pode potencialmente reduzir OPEX.

Outra proposta, mas desta vez apenas para a configuração inicial de um NCL, focou-se na análise de vários *Evolved-UMTS Terrestrial Radio Access Network* (E-UTRAN) *NodeBs* (eNBs) para gerar NCLs iniciais [18]. Operou sob o princípio de que um NCL ideal apenas deve conter relações de vizinhança em que todas as células relacionadas estão a sobrepor-se umas com as outras de alguma forma para formar regiões de HO sem interrupções de ligação. Células adjacentes deveriam reportar o rácio sinal-interferência (SINR) para comparar a um filtro definido e determinar se é ou não uma relação válida. A definição deste filtro dependia do tamanho da célula a analisar, sendo baixo se for uma macrocélula ou alto se for uma picocélula. O NCL gerado conseguia refletir o ambiente rádio verdadeiro, e o seu tamanho dependia no quão grande os filtros SINR eram: os altos geravam NCLs relativamente pequenos, que reduzia o *overhead* do sistema, mas aumentava a probabilidade de latências em HOs devido a falta de informação, enquanto que

os pequenos geravam NCLs mais extensos, reduzindo latências potenciais, mas aumentando o overhead. A simplicidade de tais protocolos de criação de NCLs eram, e ainda são pertinentes hoje para redes heterogêneas.

A criação e otimização de NCLs através de atualizações regulares não são limitados a dados provenientes de células. De facto, outra proposta utilizou medidas provenientes de utilizadores que seriam pedidas por estações base aos dispositivos [29]. Essas medidas então podiam ser feitas a uma lista específica de células ou para todas as células no NCL corrente do dispositivo. Este exemplo teve o cuidado de prevenir problemas com HOs, já que NCLs só podiam conter células cujo utilizador podia fazer HOs rapidamente, e também requeria os *scrambling codes* de cada célula pequena, pois a sua potência de sinal aumentava rapidamente nas suas redondezas. Uma simulação foi realizada numa rede 3G em Berlim, que relatou melhoramentos no rácio de sucesso em HOs e na vida da bateria dos dispositivos.

Tecnologias de gerações anteriores não foram ignoradas durante a década. Uma patente foi registada para uma funcionalidade envolvendo um sistema de NCLs tecnologicamente agnósticos, testado primariamente em redes GSM 2G e funcionava de uma maneira parecida a sistemas ANR modernos: duas estações base devem detetar e salvar o sinal de cada uma, devem verificar células para HOs dos dois lados e depois calcular um parâmetro de pesagem aglomerando esses dois componentes para determinar uma prioridade de HO para a relação de vizinhança de cada um [21].

Por outro lado, olhando para o futuro, tentativas na virtualização de funções para sistemas ANR do 5G foram feitas noutra proposta [33]. Como as frequências elevadíssimas usadas pela nova geração tecnológica não permitem que células cubram uma área grande, muitas estações são necessárias para alcançar coberturas desejadas, que aumenta o número de NCLs e o seu tamanho. HOs simultâneos foram considerados problemáticos, e medidas foram feitas nos recursos consumidos e nas latências oriundas dos NCLs grandes. A proposta centrava-se no uso de um computador central para alocar recursos enquanto corria instâncias de eNBs em

centrais de dados. Com tudo isto, o ANR era então capaz de responder eficientemente para adaptações na rede, com uma apropriada alocação de recursos e um desempenho em HOs que se adeque aos requisitos.



# Capítulo 3

## Modelos e Algoritmos

Este Capítulo aborda os requisitos propostos para o sistema, bem como os modelos e algoritmos que fazem parte. Na Secção 3.1 abordam-se os requisitos e casos de uso do sistema; na Secção 3.2 demonstra-se o padrão de trabalho a usar para o desenvolvimento do projeto; na Secção 3.3 são descritos os algoritmos de identificação de interseções de grelhas de células; na Secção 3.4 descrevem-se os algoritmos de identificação de vizinhas.

### 3.1 Análise de requisitos e casos de uso

Os requisitos funcionais para o sistema a desenvolver listam-se de seguida:

1. Dadas coberturas estimadas representadas por retângulos num plano bidimensional definidos por GeoHashes, o sistema deve ser capaz de identificar interseções entre células e listar-las para cada uma em listas em *Javascript Object Notation* (JSON);
2. Dado um limiar de potência em decibel-milliwatts (dBm) definido pelo utilizador e um mapa-mundo em *Tab Separated Values* (TSV) ou JSON que listam pares célula-potência (identificados pela abreviação do nome da célula e dBm) distribuídas por localidades definidas por GeoHashes, o sistema

deve conseguir identificar e contar pares com potências mais elevadas ou iguais ao limiar definido em cada GeoHash, que serão então considerados vizinhos. A partir daí será criado um JSON que lista cada célula com a sua própria lista de vizinhas que guarda o número de vezes que outras células foram encontradas nas suas vizinhanças;

3. Deve ser integrado com o sistema Metric SaaS, criado pela Multivision Lda.

Os requisitos não-funcionais para o sistema a desenvolver listam-se de seguida:

1. Os *inputs* e *outputs* devem ser legíveis por um humano e facilmente processáveis;
2. O sistema deve ser compatível com o sistema de cloud da *Amazon Web Services*;
3. O sistema deve ser tecnologicamente agnóstico, ou seja, não deve ser dependente de elementos de gerações tecnológicas de telecomunicação específicas, e deve funcionar tanto com o 1G ou, futuramente, o 5G.

O sistema tem três casos de uso principais, refletidos na Figura 3.1, primariamente relevantes a um operador de rede:

1. O planeamento automático de vizinhanças. O programa trivializa os cálculos necessários para planear relações de vizinhas numa rede celular, especialmente quando se adicionam novas células a uma rede existente;
2. Limpeza, otimização ou atualização de NCLs geradas por sistemas ANR. Essas otimizações podem ser realizadas criando *blacklists* para relações de vizinhança extremamente fracas de acordo com os resultados do programa;
3. Medir sobreposições de cobertura numa forma exclusivamente numérica. Os resultados podem ser úteis para medir o grau de sobreposição entre duas células.

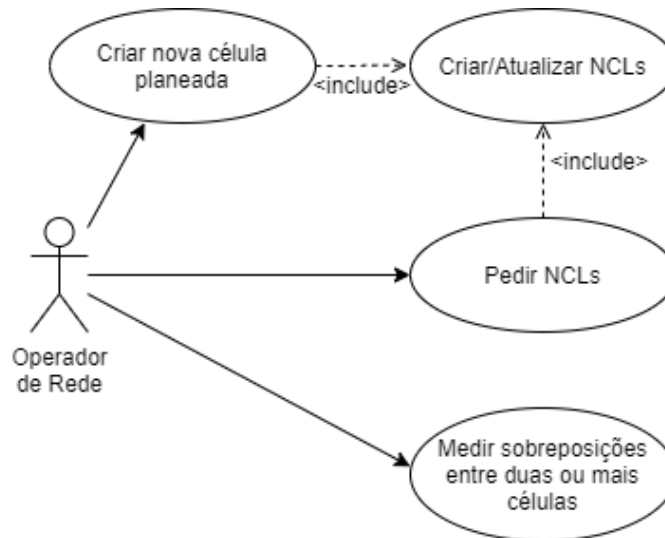


FIGURA 3.1: Os casos de uso do sistema na perspectiva de um operador de rede.

## 3.2 Visão geral do padrão de trabalho

A arquitetura que suporta este padrão é demonstrada na Figura 3.2, usando uma solução baseada na *cloud* apropriada para o processo moderno de planejamento de redes. Utilizadores móveis usam dispositivos que comunicam com células de uma RAN, que por sua vez são monitorizadas pelo OSS da rede. O OSS tem a responsabilidade de dar informação sobre a rede celular, tal como a localização das células, diagramas de antena, *drive tests* e dados do terreno para a *pipeline* para que possa receber NCLs para cada célula e atualizar os existentes de acordo com o que recebe. Os dispositivos podem, então, usar essas listas para determinar os melhores candidatos para HOs.

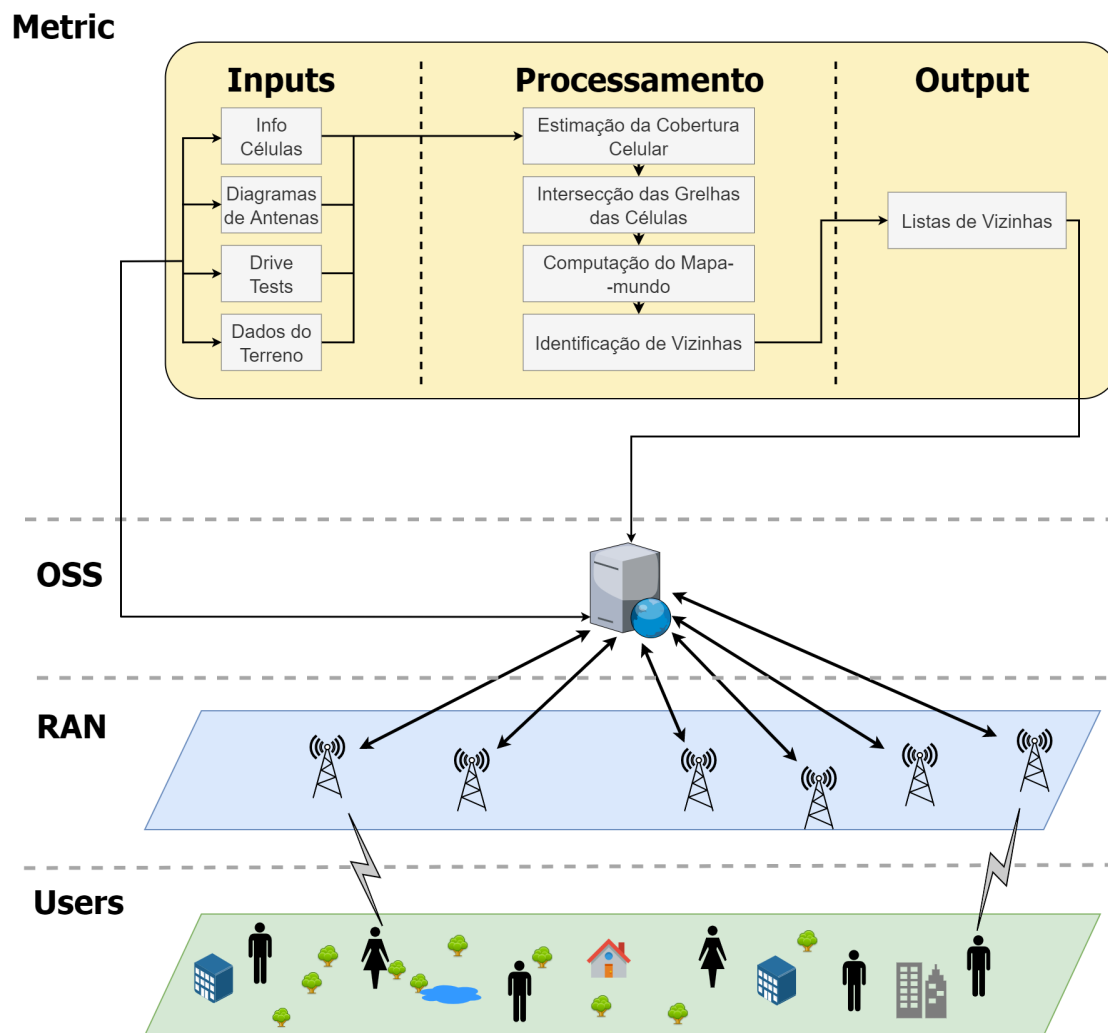


FIGURA 3.2: A arquitetura do sistema integrado no Metric.

Um operador de rede com acesso ao Metric pode pedir pelo NCL de uma ou mais células com recurso a este sistema. Este padrão de trabalho enquadra-se bem com os paradigmas de auto-configuração e auto-otimização de redes auto-organizadas, desde que o sistema seja capaz de estar operacional de forma completamente autónoma, que os *inputs* sejam apropriadamente enviados e que o *output* (os NCLs) sejam tratados por outro sistema automatizado; dado isto, este apenas pode actuar como um módulo de extensão para um sistema automatizado pre-existente, pois não faz qualquer forma de otimização sozinho.

Segue-se uma breve descrição de cada módulo demonstrado no passo "Processamento":

- **Módulo de Estimação da Cobertura Celular:** Cria uma grelha de pixeis geo-referenciados para cada antena que lhe é dado, combinando diferentes formas de informação como localização, altura, inclinação, diagramas verticais e horizontais, nível da potência de transmissão, dados de terreno e DTs utilizando o *Standard Propagation Model* (SPM) e o *Walfish-Ikegami propagation model* [9]. Uma grelha de cobertura criada pelo módulo é exemplificado na Figura 3.3;

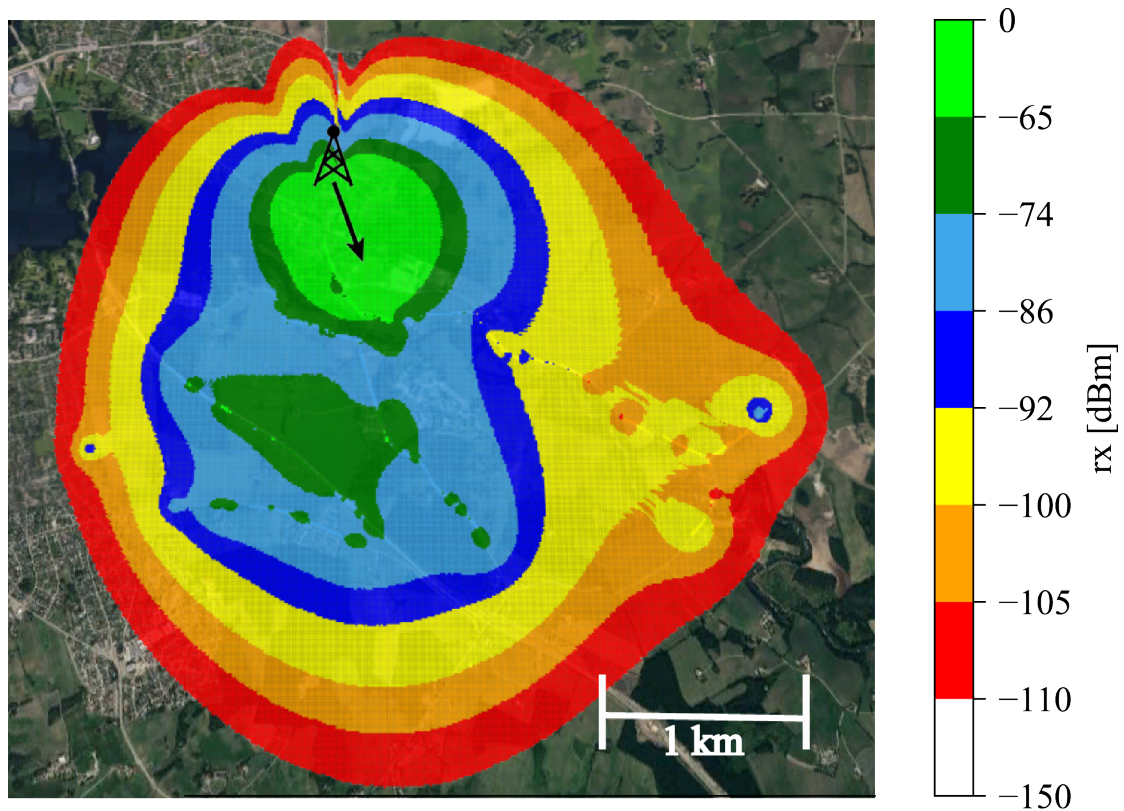


FIGURA 3.3: Um exemplo de uma grelha de pixeis com uma estimação da cobertura de uma célula.

- **Módulo de Interseção das Grelhas das Células:** Usando as coberturas estimadas do módulo anterior simplificadas para retângulos, este módulo descobre quais são as células que se intercetam umas com as outras. Para cada célula irá listar as outras com quem se interceta. Pode calcular as interseções de todas as células de uma dada lista, ou pode usar uma *Quad Tree* para assinalar uma área do mundo para descobrir as interseções de aí dentro;

- **Módulo de Computação do Mapa-mundo:** Com as interseções do módulo anterior e uma grelha do sinal percebido do primeiro, itera sobre todas as células que apresentam interseções e os pixels que estas cobrem para elaborar um mapa-mundo definido por uma lista de pixels, que por sua vez também listam pares célula-potência [13];
- **Módulo de Identificação de Vizinhas:** O módulo final produz NCLs através do mapa-mundo do anterior, iterando por cada pixel e declarando vizinhas as células que apresentam potências abaixo de um certo limiar parametrizado pelo utilizador. Cada NCL de cada célula é uma lista de pares nomes de células com o número de vezes que foi contada como vizinha.

Nas próximas Secções serão apresentadas propostas para os algoritmos de interseção de grelhas das células (um método iterativo e um método recursivo) e para a identificação de vizinhas (uma agregação de células e uma agregação de pixels).

### 3.3 Algoritmos de identificação de interseção de grelhas de células

Antes de se poder calcular a vizinhança das células, é preciso identificar as interseções das grelhas dessas. Estas grelhas são definidas por retângulos que delimitam, aproximadamente, a área de cobertura das células; foram usados retângulos especificamente pela sua simplicidade para que os cálculos não fossem muito complexos. Este fenómeno é exemplificado na Figura 3.4.

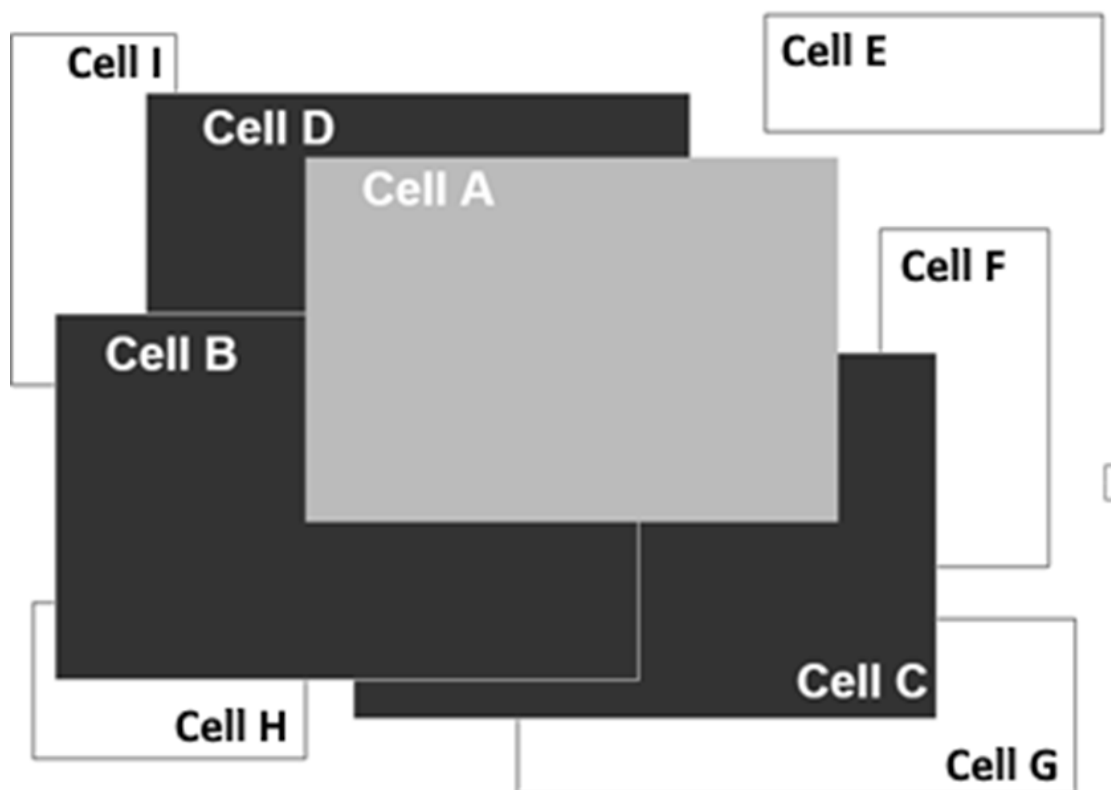


FIGURA 3.4: Representação das coberturas das células como retângulos.

### 3.3.1 Método iterativo

O primeiro algoritmo de cálculo de interseção geográfica que é proposto e exemplificado na Figura 3.5 tem como objetivo processar uma lista de células que contém coordenadas sobre a sua cobertura geográfica retangular; cada célula é analisada com o resto da lista para detectar potenciais interseções entre cada uma e, se existirem, são assinaladas.

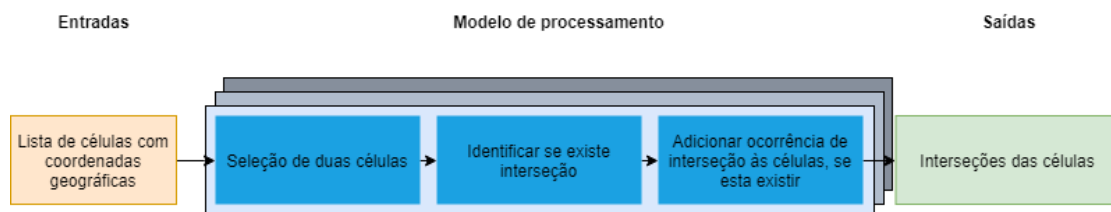


FIGURA 3.5: Diagrama que exemplifica o funcionamento do algoritmo iterativo.

O algoritmo necessita, como *input*, de uma tabela de células com as seguintes informações:

- A abreviação do nome da célula;
- O *GeoHash* correspondente ao canto inferior esquerdo da cobertura da célula;
- O *GeoHash* correspondente ao canto superior direito da cobertura da célula;
- O nome da localização da estação radio base.

Como *output* é retornada, para cada célula, uma lista de células com que se intersectou.

Cada célula é identificada por uma abreviação do seu nome, o nome do local da sua estação radio base, o canto inferior esquerdo e o canto superior direito do seu retângulo de cobertura em formato *GeoHash*. Para facilitar o cálculo de interseções, os *GeoHashes* são convertidos de volta para coordenadas geográficas. Segue-se, no Algoritmo 1, a estratégia a tomar em pseudocódigo para efetuar o cálculo de interseções de todas as células providenciadas por uma lista.



**Algorithm 1** Cálculo de interseções à força bruta

**Input:** Lista de células com coordenadas geográficas

**Output:** Uma lista de interseções para cada célula

```

1: procedure SETINTERSECTIONS(listOfCells)
2:    $i \leftarrow 1$ 
3:   while  $i < \text{listOfCells.length}$  do
4:      $cell1 \leftarrow \text{listOfCells}[i]$ 
5:      $j \leftarrow i + 1$ 
6:     while  $j < \text{listOfCells.length}$  do
7:        $cell2 \leftarrow \text{listOfCells}[j]$ 
8:       if  $cell1.upperLeftX > cell2.bottomRightX$  or  $cell2.upperLeftX >$ 
        $cell1.bottomRightX$  or  $cell1.upperLeftY < cell2.bottomRightY$  or  $cell2.upperLeftY <$ 
        $cell1.bottomRightY$  then
9:         continue ▷ Não existe interseção
10:      if  $cell2$  not in  $cell1.intersectionsList$  then
11:         $cell1.intersectionsList+ = cell2$ 
12:      if  $cell1$  not in  $cell2.intersectionsList$  then
13:         $cell2.intersectionsList+ = cell1$ 
14:       $j+ = 1$ 
15:     $i+ = 1$ 
return  $\text{listOfCells}$ 

```

---

### 3.3.2 Método recursivo com *Quad Tree*

Enquanto que o algoritmo descrito na Secção 3.3.1 calcula as interseções de todas as células e precisa de percorrer toda a lista de células, pode-se usar uma *Quad Tree* para calcular as interseções de uma lista específica de células, aproveitando-se as suas coordenadas geográficas para definir áreas de procura equivalentes aos limites de cobertura das células que se querem planear.

No contexto deste sistema, uma *Quad Tree* entende-se como uma estrutura de dados em árvore em que cada nó está associado a quatro nós filhos, que não estão inicializados no momento de criação do nó (ou seja, apenas são inicializados quando for útil). Cada nó é definido por um retângulo com coordenadas geográficas e o nó pai contém todos os outros. Um objeto apenas pode-se associar a um nó se este for contido totalmente pelos limites geométricos do nó, logo, se um objeto sobrepuser-se em vários nós, ele é considerado parte do nó pai desses. Cada nó pode conter um número parametrizável de objetos, e quando a quantidade máxima é atingida, os filhos são inicializados e os objetos são redistribuídos por eles, se for possível associar-los aos filhos.

Este algoritmo tem passos adicionais para a criação da *Quad Tree* e a seleção da área de procura. A API usada para a definição desta é uma versão modificada da de Ryan Pelletier [28], para suportar coordenadas geográficas. Em vez de conter coordenadas geográficas dos seus cantos como descrito na Secção 3.3.1, os limites da cobertura da célula são definidos pelas coordenadas do canto superior esquerdo, o comprimento e a largura do retângulo de cobertura. Esta diferença deve-se a um passo extra de compatibilidade com a API, que exige a definição de retângulos desta forma.

O algoritmo necessita, como *input*, de uma tabela de células com as mesmas informações descritas na Secção 3.3.1. Para além disso, também precisa de uma lista de células a que se querem planear as interseções ("células candidatas"). Como *output* é retornada, para cada célula especificada, uma lista de células com

que se intersectou. Um exemplo fictício do funcionamento da *Quad Tree* com as células apresenta-se na Figura 3.6.

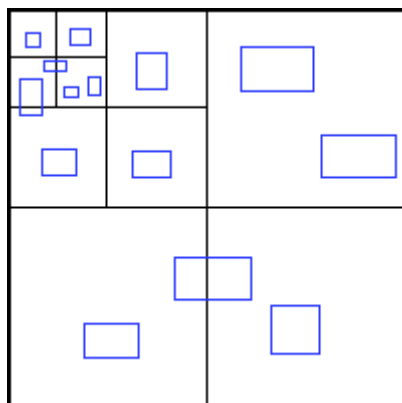


FIGURA 3.6: Exemplo fictício de uma *Quad Tree* e as células inseridas nela como coberturas retangulares.

Os passos para a execução desta estratégia são os seguintes:

1. Definição de todas as células e as "células candidatas";
2. Conversão das células geográficas para células retangulares;
3. Construção da *Quad Tree*, posicionando-a em longitude -180, latitude -90, comprimento 360 e largura 180, para que o nó raiz cubra o espaço de todo o globo;
4. Inserção de todas as células na *Quad Tree* construída anteriormente;
5. Para cada célula candidata:
  - (a) Criar uma "caixa delimitadora" com as dimensões e a posição da célula candidata;
  - (b) Extrair todas as células que estão contidas ou se intersectam com essa caixa;
  - (c) Inserir células extraídas numa lista de interseções;
6. Exportar lista de interseções de cada célula candidata.

## 3.4 Algoritmos de identificação de vizinhas

Para calcular as vizinhanças de cada célula, são propostos dois algoritmos. Estes baseiam-se no princípio de contagem de ocorrências de relações de vizinhança, ou seja, as células que forem contadas mais vezes como vizinhas de outra são consideradas as melhores vizinhas dessa.

### 3.4.1 Algoritmo iterativo de agregação de células e as suas coberturas

Será abordado o primeiro novo algoritmo de planeamento de vizinhos, exemplificado na Figura 3.7. Este tem como objetivo tratar cada célula como um indivíduo com visibilidade aos Pixeis onde tem cobertura e, a partir da lista de células desses Pixeis, descobrir a sua vizinhança.

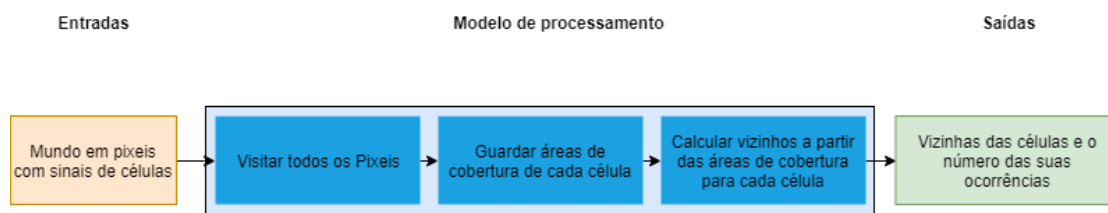


FIGURA 3.7: Diagrama do algoritmo de agregação de células e as suas coberturas.

O algoritmo necessita, como *input*, de um conjunto de Pixeis (quadrados/retângulos), identificado como um "Mapa mundo". Cada um contém um conjunto de células, e cada célula está associada a uma potência de sinal. Este mapa é limitado à localização das células, e não cobre todo o mundo. Como *output* é retornada uma lista de células, em que cada uma tem uma lista de células vizinhas e o número de ocorrências que foram assinaladas como vizinhas.

O mapa do Mundo é dividido numa grelha de Pixeis identificados por *GeoHashes*, códigos binários que identificam faixas de pares latitudes/longitudes. Cada Pixel guarda a intensidade de sinal de um conjunto de células; esse conjunto

corresponde a células que têm um nível de sinal recebido no Pixel superior a um determinado limiar. Cada célula é identificada por uma abreviação do seu nome e tem visibilidade para os Pixeis onde tem cobertura. Cada célula pode fornecer cobertura de sinal a vários Pixeis simultaneamente; a partir deste fenómeno, consegue-se descobrir a relação entre os Pixeis e as Células explicitamente.

Esta estratégia decompõe-se em dois passos implementados por algoritmos distintos. Segue-se, no Algoritmo 2, o pseudocódigo que descreve o cálculo de áreas de cobertura de cada célula, que corresponde identificar o conjunto de Pixeis em que cada célula tem cobertura. No Algoritmo 3, apresenta-se o pseudocódigo que descreve o cálculo de vizinhos de cada célula.

---

**Algorithm 2** Cálculo de áreas de cobertura de cada célula

---

**Input:** Mapa de Pixeis (identificados por GeoHashes), cada um contendo uma lista de pares (cellId,power)

**Output:** Lista de células, armazenando para cada uma o seu conjunto de pixeis cobertos

```

1: procedure SETCELLSCOVERAGE(mapOfPixelsWithCellsSignalLevel)
2:   listOfCellsCoverages  $\leftarrow$  list() ▷ Lista vazia
3:   for pixel in mapOfPixelsWithCellsSignalLevel do
4:     for cell, power in pixel do
5:       if cell in listOfCellsCoverages &
6:         power  $\geq$   $-120$  then
7:         listOfCellsCoverages[cell]+ = [pixel : power]
8:       else
9:         listOfCellsCoverages[cell]  $\leftarrow$  [pixel : power]
   return listOfCellsCoverages

```

---

**Algorithm 3** Cálculo de vizinhos

**Input:** Lista de áreas de coberturas das células

**Output:** Para cada célula, uma lista de pares (cellVizinha, ocorrências)

```

1: procedure SETCELLSNEIGHBOURS(listOfCellsCoverages)
2:   cellsNeighbours  $\leftarrow$  listOfCellsCoverages.keys       $\triangleright$  Lista de células,
   armazenando para cada uma o seu conjunto de células vizinhas
3:   for cell in listOfCellsCoverages do
4:     for cellNeighbour in listOfCellsCoverages do
5:       if cell  $\neq$  cellNeighbour then
6:         for pixel in cell do
7:           for pixelNeighbour, powerNeighbour in
8:             listOfCellsCoverages[cellNeighbour] do
9:               if pixel = pixelNeighbour &
10:                 powerNeighbour  $\geq$  -120 then
11:                 cellsNeighbours[cell][cellNeighbour]+ = 1

   return cellsNeighbours

```

### 3.4.2 Algoritmo iterativo de agregação de pixels com células

Será abordado o segundo novo algoritmo de planeamento de vizinhos, exemplificado na Figura 3.8. Este tem como objetivo processar cada Pixel individualmente e descobrir a vizinhança das suas células, em vez de ter todas as células carregadas de uma vez. O *Input/Output* é igual ao do Algoritmo 2, enquanto que a estrutura da Célula apenas contém o seu nome e a sua lista de vizinhos.

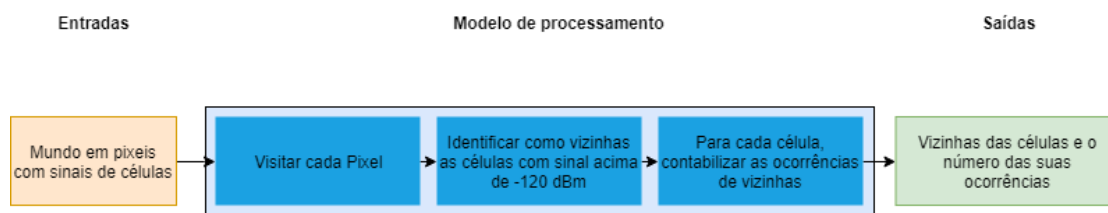


FIGURA 3.8: Diagrama do algoritmo de agregação de pixels com células.

No Algoritmo 4 apresenta-se a estratégia a tomar em pseudocódigo para efetuar o cálculo de vizinhos de cada célula através de um Mapa de Pixeis.

---

**Algorithm 4** Cálculo de vizinhos de cada célula

---

**Input:** Mapa de Pixeis (identificados por GeoHashes), cada um contendo uma lista de pares (cellId,power)

**Output:** Para cada célula, uma lista de pares (cellVizinha, ocorrências)

```

1: procedure SETNEIGHBOURS(mapOfPixelsWithCellsSignalLevel)
2:   cellsNeighbours  $\leftarrow$  list() ▷ Lista vazia
3:   for pixel in mapOfPixelsWithCellsSignalLevel do
4:     auxNeighboursList  $\leftarrow$  list() ▷ Lista vazia
5:     for cell, power in pixel do
6:       if power  $\geq$  -120 then
7:         auxNeighboursList+ = cell
8:       for cell in auxNeighboursList do
9:         for cellNeighbour in auxNeighboursList do
10:          if cell  $\neq$  cellNeighbour then
11:            if cellNeighbour not in cellsNeighbours[cell] then
12:              cellsNeighbours[cell][cellNeighbour]  $\leftarrow$  1
13:            else
14:              cellsNeighbours[cell][cellNeighbour] + = 1
return cellsNeighbours

```

---





# Capítulo 4

## Implementação e Integração do Sistema

Este Capítulo aborda a metodologia prática usada para o funcionamento real do sistema, comprovada por ferramentas e testes de aceitação. Na Secção 4.1 descrevem-se as tecnologias e ferramentas utilizadas para a implementação do sistema; na Secção 4.2 demonstra-se a realização do padrão de trabalho anteriormente descrito na prática; na Secção 4.3 descreve-se como é que o sistema interage com o Metric; finalmente, na Secção 4.4 são descritos testes de aceitação que comprovam o bom funcionamento do sistema implementado.

### 4.1 Tecnologias e ferramentas utilizadas

Para o desenvolvimento deste trabalho, foram usadas tecnologias e ferramentas complementares para acelerar o processo de desenvolvimento e alcançar os seus objetivos, que serão descritas neste capítulo.

### 4.1.1 Tecnologias

Este projeto foi desenvolvido maioritariamente em Java, uma linguagem de programação orientada por objetos que segue uma filosofia de virtualização: o código binário corre numa instância de uma máquina virtual, para que este não esteja dependente da arquitetura por baixo; ou seja, desde que uma plataforma suporte Java, um desenvolvedor de aplicações não precisa de se preocupar com a sua arquitetura e o seu código poderá ser executado em qualquer ambiente [5]. A escolha do Java foi devido à sua portabilidade, ao seu elevado suporte (em termos de disposição de APIs, popularidade, etc.) e à familiaridade do autor com a linguagem.

Os dados usados pelo sistema são partilhados por outros e esta transição é apoiada pelo JSON, um formato de intercâmbio interprocesso de dados fácil de ler e de utilizar [16]. No contexto do Java, uma API open-source da Google chamada GSON foi utilizada para processar dados provenientes de, ou escrever para ficheiros JSON [14].

O planeamento e a otimização de redes móveis são ambas muito exigentes computacionalmente, mas são independentes de tecnologia ou de vendedores: soluções SaaS baseadas na cloud são particularmente úteis para estas tarefas pois são relativamente rápidas de desenvolver e facilmente escaláveis para servir vários clientes. Esta velocidade e escalabilidade são importantes para a procura crescente, indo além das limitações de desenvolvimento tradicional de software e permitir a evolução de um produto num mercado de software liderado pela procura [36]. Soluções SaaS, ao longo dos anos, tem tido um impacto significativo no desenvolvimento de projetos de engenharia de software e de bases de dados, pois permite o serviço a quantidades enormes de utilizadores sem comprometer o desempenho ou segurança, ao ponto de ser capaz de providenciar boas capacidades de personalização para cada utilizador e armazenamento de dados fiável [35].

O potencial completo de soluções SaaS é realizado quando é combinado com um paradigma *serverless* na *cloud*. Uma arquitetura *serverless* permite aos engenheiros de software não terem que se preocupar com configurações ou manutenções

de redes de servidores para providenciar os serviços a serem desenvolvidos, pois essas tarefas intermédias são feitas por terceiros.

O GeoHash é um sistema de coordenadas geográficas que divide o espaço bidimensionalmente em retângulos numa hierarquia de precisão acumulativa; cada espaço individual é identificado por uma *string* que é tão extensa quanto o grau de *zoom* relativo ao espaço bidimensional. Duas propriedades importantes é que *strings* com prefixos semelhantes normalmente estarão muito próximas geograficamente, e é possível converter GeoHashes para intervalos de latitude/longitude e vice-versa [23].

Esta tecnologia é importante no contexto de guardar vários intervalos de coordenadas geográficas em massa. *Strings* de GeoHashes são tipicamente representados com muito menos caracteres do que intervalos de números decimais, uma vez que a precisão é personalizável à letra.

### 4.1.2 Ferramentas

Para adquirir dados sobre a cobertura, capacidade e qualidade de serviço de uma rede móvel, é necessário recorrer a testes especializados percorrendo as várias células a analisar. O Metric consegue providenciar a técnica dos *Drive Tests*, que consiste em fazer esse percurso com um veículo equipado com instrumentos de medição especificamente feitos para esse uso.

As informações providenciadas pelo Metric são indispensáveis para a realização deste projeto, pois são estas que fazem possível a determinação do espaço de cobertura das células diferentes. Deste conseguem-se grelhas de interseções e subsequentemente as relações de vizinhança.

Para implementar o sistema proposto utilizando o Metric, recorre-se a serviços *cloud* do AWS, que é uma plataforma da *Amazon* que oferece soluções de servidores virtuais em nuvem *on-demand* e pago à medida do uso dos seus serviços [2]. Os serviços oferecidos são vários e conseguem interagir uns com os outros, e este projeto utilizou os seguintes:

- AWS *Lambda* - *Hosting* de funções anônimas em instâncias virtuais que podem ser executadas *on-demand* e paralelamente;
- AWS *Elastic Computing Cloud* (EC2) - *Hosting* de máquinas virtuais que podem correr vários processos;
- AWS *Simple Storage System* (S3) - Sistema de armazenamento de objetos que funciona de maneira semelhante a um sistema de ficheiros.

Os APIs seguintes foram utilizados para usar estas funcionalidades: "aws-lambda-java-log4j2" para registos de eventos, "aws-lambda-java-core" para usar o AWS *Lambda*, e "aws-java-sdk-s3" para usar o S3. Para facilitar a translação do sistema para o AWS, foi usado o *Serverless Framework* [32].

O Apache Maven é uma ferramenta de gestão de dependências e automatização de compilação [11]. O projeto segue todas as convenções para manter a certeza que todas as suas dependências são incluídas na compilação, sendo a sua modificação e execução fáceis. O uso do Maven também foi um requisito para o funcionamento do sistema na cloud escolhida.

## 4.2 Implementação do padrão de trabalho

Na Figura 4.1 ilustra-se uma visão detalhada do padrão de trabalho do sistema, porém este projeto focar-se-à mais acentuadamente nos Lambdas 2 e 4. Dispositivos móveis comunicam com células a partir de uma RAN, que por sua vez são monitorizadas pelo OSS da rede. O OSS tem a responsabilidade de providenciar informação sobre a rede celular, como as localizações das células, diagramas das antenas, DTs e dados do terreno para a pipeline do sistema. Receberá então, depois, NCLs para cada célula para depois atualizar-las. Os dispositivos móveis então recebem listas destas para determinar a melhor vizinha com quem realizar um HO.

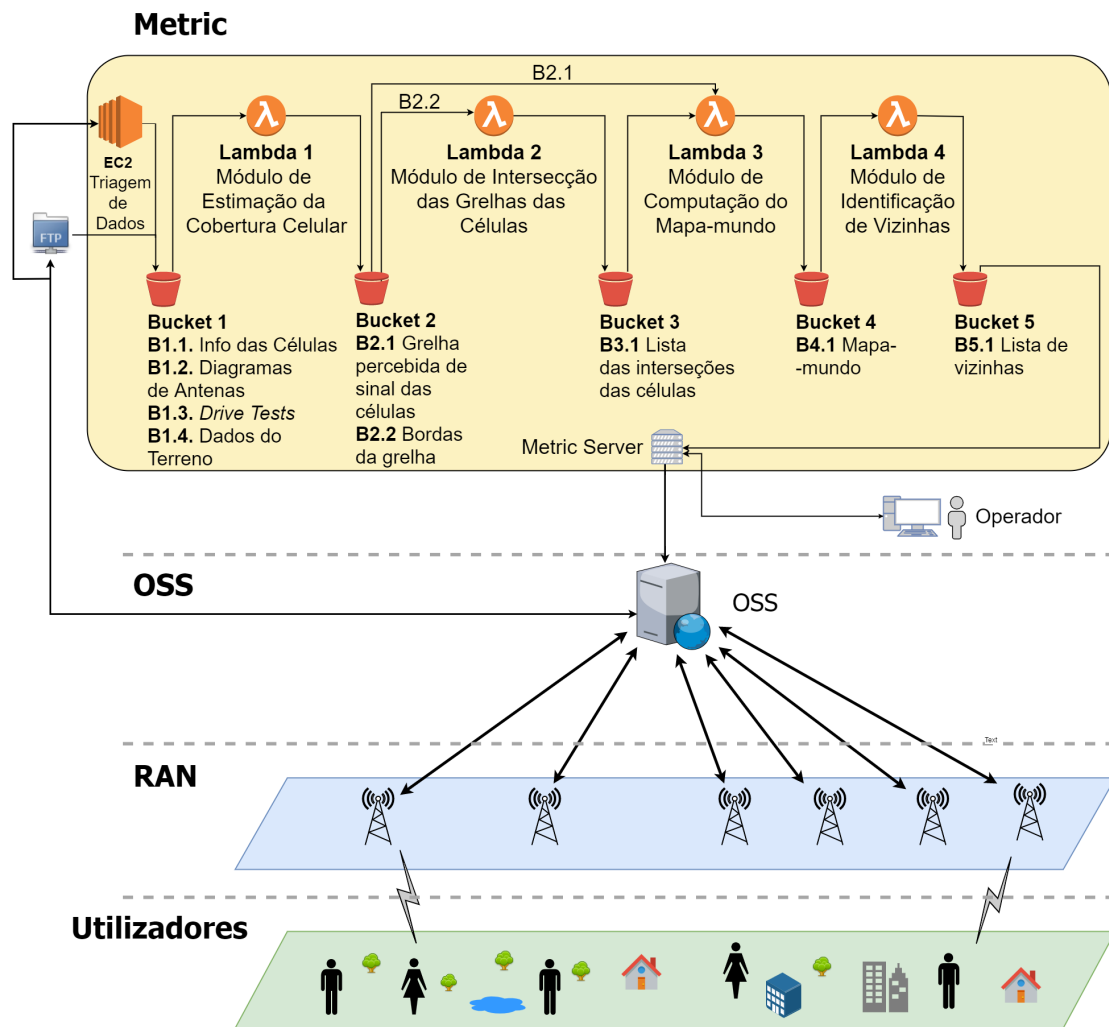


FIGURA 4.1: Padrão de trabalho integrado com o AWS.

Cada Lambda tem os seguintes propósitos:

- Lambda 1 - Módulo de Estimação da Cobertura Celular:** O módulo adquire diagramas de antenas, DTs, dados do terreno através do OSS num formato *Extensible Markup Language* (XML) e informações de células em ficheiros *Tab Separated Values* (TSV) através do *Bucket 1* para calcular um alcance estimado para cada célula pedida. No fim, faz o *output* de dois objetos para o *Bucket 2* do S3: a grelha percebida de sinal de cada célula num ficheiro JSON e bordas da grelha da cobertura de cada célula, assim como a localização das suas BSs numa lista *Comma Separated Value* (CSV) como exemplificado na Tabela 4.1.

cellAbbrev,	bottomLeftCorner,	topRightCorner,	btsLocation
nome1,	geohash00,	geohash01,	local1
nome2,	geohash00,	geohash02,	local2
nome3,	geohash10,	geohash02,	local3

TABELA 4.1: Exemplo de uma instância das bordas da grelha (B2.2).

- Lambda 2 - Módulo de Interseção das Grelhas das Células:** Usando a lista CSV de bordas das células do *Bucket 2*, este usa um algoritmo de interseções para encontrar sobreposições entre células. Cada sobreposição é registado numa lista JSON que contém células que por sua vez listam as outras com quem se intersetam; este resultado é também exportado para o *Bucket 3* e é exemplificado com a Tabela 4.2. Enquanto que este módulo apresenta apenas um output, este tem duas formas de operação como descritas nas Subsecções 3.3.1 e 3.3.2, Secção 3.3, Capítulo 3.

```
{
  'cellabbrev1':['cellabbrev3', 'cellabbrev4', 'cellabbrev7'],
  'cellabbrev3':['cellabbrev4', 'cellabbrev8'],
  'cellabbrev5':[]
}
```

TABELA 4.2: Exemplo de uma lista das interseções das células (B3.1).

- Lambda 3 - Módulo de Computação do Mapa-mundo:** Este módulo utiliza dois *inputs*, um ficheiro JSON que contém a grelha percebida de sinal das células vindo do *Bucket 2*, e outro que lista as interseções das células proveniente do *Bucket 3*, descartando aquelas que não apresentam sobreposições, para construir um Mapa-mundo que delimita a cobertura de cada célula por pixeis identificados por GeoHashes, criando assim uma lista de pixeis que contém células e a potência de sinal (em dBm) apresentada em cada [13]. O *output* é guardado como um JSON ou para um TSV orientado pela coluna para o *Bucket 4* e é exemplificado na Tabela 4.3.

```
geoHash1 \t cellabrev1 \t power1 \t cellabrev2 \t power2 \t ...
geoHash2 \t cellabrev5 \t power1 \t cellabrev9 \t power12 \t ...
```

TABELA 4.3: Exemplo de uma instância do mapa-mundo (B4.1), em que cada "\t" representa um *tab*.

- Lambda 4 - Módulo de Identificação de Vizinhas:** Finalmente, NCLs podem ser produzidos usando o Mapa-mundo em JSON ou TSV previamente produzido e guardado no *Bucket 4*, calculando relações de vizinhança entre as células apresentadas dentro de cada pixel comparando as suas forças de sinal a um limiar definido pelo utilizador; se o sinal for maior do que esse, então as células são consideradas vizinhas umas das outras e ganham 1 *hit* nas NCLs de cada uma. Os NCLs de cada célula funcionam como um *leaderboard* que classifica cada entrada pelas vezes que foi contada, ordenando do maior ao menor; teoricamente, o vizinho do topo deverá ser o melhor vizinho. Cada NCL de cada célula é exportada para ficheiros JSON para o *Bucket 5*, exemplificado pela Tabela 4.4.

```
{
  "cellD": {
    "celle": 1,
    "cellB": 1,
    "cellC": 1,
    "cellA": 2
  },
  "celle": {
    "cellD": 1,
    "cellA": 1
  }
}
```

TABELA 4.4: Exemplo da lista de vizinhas (B5.1).

### 4.3 Integração com a ferramenta Metric SaaS

Este sistema está completamente integrado com o *Metric*, permitindo um operador de rede lançar pedidos de NCLs de células específicas de acordo com um limiar de sinal especificado. Esta informação é útil para um número de aplicações, como a alocação de recursos rádio (por exemplo: frequências, códigos ou identificadores) para assistir na minimização de interferência entre células vizinhas, uma vez que o rácio de sobreposição de coberturas pode ser usado como uma heurística nesse aspeto; ou seja, duas células que não têm muita sobreposição entre elas provavelmente também não têm muita interferência. Este processo é, assim, tecnologicamente agnóstico.

Um caso real deste caso de uso é apresentado na Figura 4.2, com os nomes das células censurados para proteger a privacidade do operador de rede, sendo também possível correr este caso em paralelo por diferentes utilizadores graças à infraestrutura *serverless* baseada na *cloud* do sistema.

### 4.4 Testes de aceitação e de integração no Metric

Para avaliar se a implementação e integração do sistema estão corretas, foram elaborados testes com o objetivo de descobrir se a execução dos seus algoritmos dão os resultados pretendidos. Mais concretamente, dado um *input* específico esperou-se um *output* específico.

Foram feitos testes de formas semelhantes para os dois módulos relevantes à discussão deste projeto, pois ambos foram desenvolvidos em Java. Os testes foram feitos com auxílio da API JUnit 5 [34], instanciando o algoritmo diretamente com os inputs dados e analisando os resultados obtidos face ao resultado esperado.



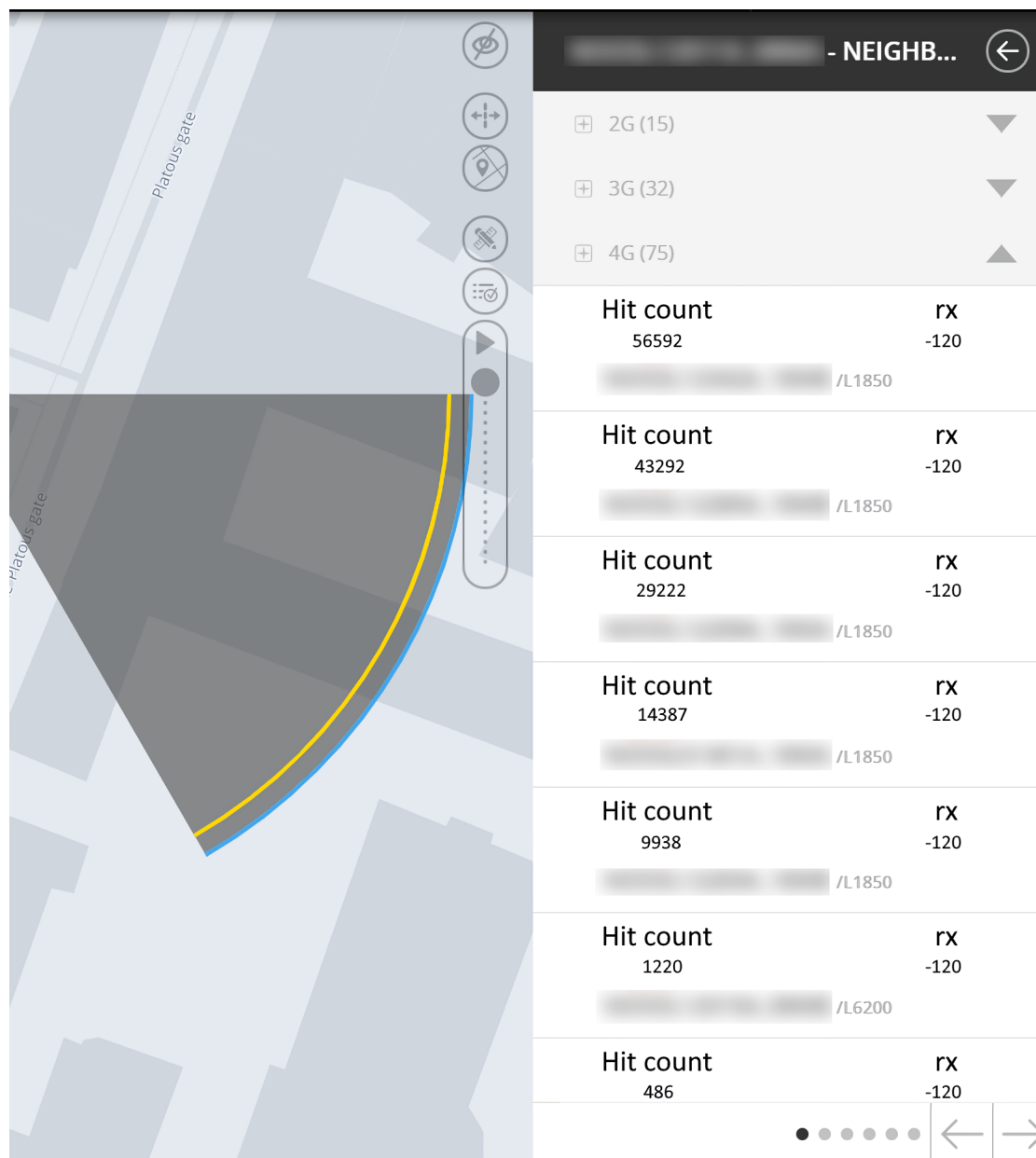


FIGURA 4.2: O sistema integrado no Metric.

#### 4.4.1 Teste à identificação de interseção de grelhas

Para validar a identificação de interseção de grelhas, usou-se uma grelha fictícia de coberturas de células de forma a verificar a conversão de um GeoHash para coordenadas geográficas e depois calcular a interseção das células a partir daí. Também foi validada a capacidade do sistema calcular e devolver as interseções de apenas uma célula.

O *input* do primeiro teste foi uma grelha de coberturas de células em formato

CSV, e o segundo teve um *input* adicional de um *array* de *strings* que especificava quais as células candidatas para a identificação de interseções. Os resultados dos dois testes apresentam-se na Tabela 4.5.

TABELA 4.5: Testes e resultados previstos para a identificação de interseções.

#	Teste	Resultado Previsto
1	<p>CSV com os seguintes conteúdos (atenção que é obrigatório apresentar o header): cellAbbrev,bottomLeftCorner,topRightCorner a,s02equ0,s144hk2 b,s09qv80,s0uk2w1 c,s04ven0,s0scp61 d,s07e3u0,s1hrbd3 e,s014f5h,s015urm</p>	<pre>{   "a": [     "b"   ],   "b": [     "a",     "d"   ],   "c": [     "d"   ],   "d": [     "b",     "c"   ],   "e": [] }</pre>
2	<p>CSV com os seguintes conteúdos (atenção que é obrigatório apresentar o header): cellAbbrev,bottomLeftCorner,topRightCorner a,s02equ0,s144hk2 b,s09qv80,s0uk2w1 c,s04ven0,s0scp61 d,s07e3u0,s1hrbd3 e,s014f5h,s015urm</p> <p>Array de Strings: { "a" }</p>	<pre>{   "a": [     "b"   ] }</pre>

Os resultados esperados foram alcançados.

### 4.4.2 Teste à identificação de vizinhas

Para validar a identificação de células vizinhas, recorreu-se a um cenário de teste a partir de um Mapa-mundo fictício em que se verifica todos os casos possíveis durante o processamento do algoritmo: casos em que nenhuma célula é vizinha, casos em que todas as células são vizinhas e casos em que algumas são vizinhas e outras não. O tratamento de exceções de nomes inválidos, forças de sinal inválidas e formatos de ficheiro inválidos são tomados como garantidos.

O *input* do primeiro teste foi um mapa mundo num formato TSV, e o segundo era igual mas com um formato JSON. Considerou-se que uma célula era vizinha de outras no mesmo pixel se apresentasse uma potência de sinal maior do que -120 dBm. O *output* foi escrito em formato JSON nos dois testes. Os resultados dos dois testes apresentam-se na Tabela 4.6.

TABELA 4.6: Testes e resultados previstos para a identificação de vizinhas.

#	Teste	Resultado Previsto
1	<p>TSV com os seguintes conteúdos (cada espaço representa um <i>tab</i>):</p> <pre> gbsuv cellA -78.0 cellB -50.0 cellC -25.1 cellD -2.12 gbsuy cellA -67.0 cellC -259 cellD -119.99999 cellE -20 gbsuu cellA -84.0 cellC -50 cellE -130 gbsut cellA -73.0 cellF -50                     </pre>	<pre> {   "cellD": {"cellE": 1, "cellB": 1,             "cellC": 1, "cellA": 2},   "cellE": {"cellD": 1, "cellA": 1},   "cellB": {"cellD": 1, "cellC": 1,             "cellA": 1},   "cellC": {"cellD": 1, "cellB": 1,             "cellA": 2},   "cellF": {"cellA": 1},   "cellA": {"cellD": 2, "cellE": 1,             "cellB": 1, "cellC": 2,             "cellF": 1} }                     </pre>

*Continua na próxima página*

Tabela 4.6 – Continua da página anterior

#	Teste	Resultado Previsto
2	JSON com os seguintes conteúdos: <pre> {   "Map": {     "gbsuv": {       "cellA": -78.0,       "cellB": -50.0,       "cellC": -25.1,       "cellD": -2.12     },     "gbsuy": {       "cellA": -67.0,       "cellC": -259,       "cellD": -119.99999,       "cellE": -20     },     "gbsuu": {       "cellA": -84.0,       "cellC": -50,       "cellE": -130     },     "gbsut": {       "cellA": -73.0,       "cellF": -50     }   } }                     </pre>	Equivalente ao resultado anterior.

Enquanto que os primeiros protótipos não alcançavam os resultados pretendidos, devido a *bugs* ou mudanças dos requisitos, com o decorrer do projeto eventualmente conseguiu-se chegar aos resultados esperados sem problemas. Estando o sistema funcional e capaz de produzir resultados corretos a partir de dados fictícios, foi então possível validar o sistema com dados reais.

# Capítulo 5

## Análise de Resultados

Neste Capítulo serão apresentados testes em contextos reais (i.e. com dados verdadeiros de operadoras anónimas), com descrições completas dos cenários escolhidos e os resultados dos testes. Na Secção 5.1 descrevem-se os cenários; na Secção 5.2 estão os resultados de cada teste.

### 5.1 Cenários

Foram desenvolvidos dois tipos de cenários, um para a comparação dos algoritmos de interseções (com um exemplo), e outro para o estudo da construção de NCLs (com dois exemplos, um pequeno e outro grande). Todos os cenários usaram configurações diferentes, mas em comum tinham o facto de que tratavam-se de células 4G LTE. Os nomes das células e das suas estações base foram censurados para proteger a privacidade da operadora de rede, e foram nomeados com designações genéricas.

### 5.1.1 Comparação de desempenho dos algoritmos de interseção

Para avaliar o desempenho do algoritmo iterativo comparado com o algoritmo com recursivo com uma *Quad Tree*, assumiu-se que o acesso à memória é o sub-processo dominante do algoritmo em termos de computação (ou seja, os outros passos podem ser ignorados), por isso contaram-se o número de iterações em três configurações diferentes baseados num cenário de teste com noventa e duas (92) células de Malanje, Angola:

1. Contagem de iterações para o processo de inserção de células na *Quad Tree*;
2. Comparação de iterações entre os dois algoritmos para o cálculo de interseções de uma só célula;
3. Comparação do número de iterações para o cálculo de interseções de 1, 2, 4, 8, 16 e todas as células para os dois algoritmos.

### 5.1.2 Cenário pequeno de vizinhanças

Este cenário realista foi baseado numa porção de uma rede celular em Vallerud, Noruega, analisando a NCL de uma célula escolhida criada pelo ANR do operador e comparando-a com os resultados do sistema. A célula escolhida foi designada genericamente por *cellA*.

Dezassete (17) células nas suas redondezas, incluindo algumas daquelas escolhidas pelo ANR, foram selecionadas para facilitar a comparação. Esta seleção foi feita considerando a distância (perto & longe) entre células e as suas relações de vizinhança definidas pelo ANR (se existiam). A análise também pretendia determinar se as duas implementações conseguiam gerar listas que faziam sentido: por exemplo, uma célula com um alcance muito longe para sul não deverá considerar vizinha uma que tenha um alcance muito longe para o norte oposto de si. As células escolhidas são apresentadas na Figura 5.1, mostrando todos os *sites* relevantes

que providenciam as células. Cada *site* é apresentado como um triângulo com um caráter que identifica o seu estado:

- Um 'P' significa que o *site* era planeado e não estava instalado durante o período de tempo escolhido;
- Um 'A' significa que todas as células de um *site* estavam ativas;
- Um 'H' significa que todas as células de um *site* estavam desativadas;
- Um '≠' de cor preta avisa que pelo menos uma das células difere com a maioria do *site*, por exemplo, se um tem duas ativas e uma desativada; um '≠' de cor vermelha avisa o contrário.

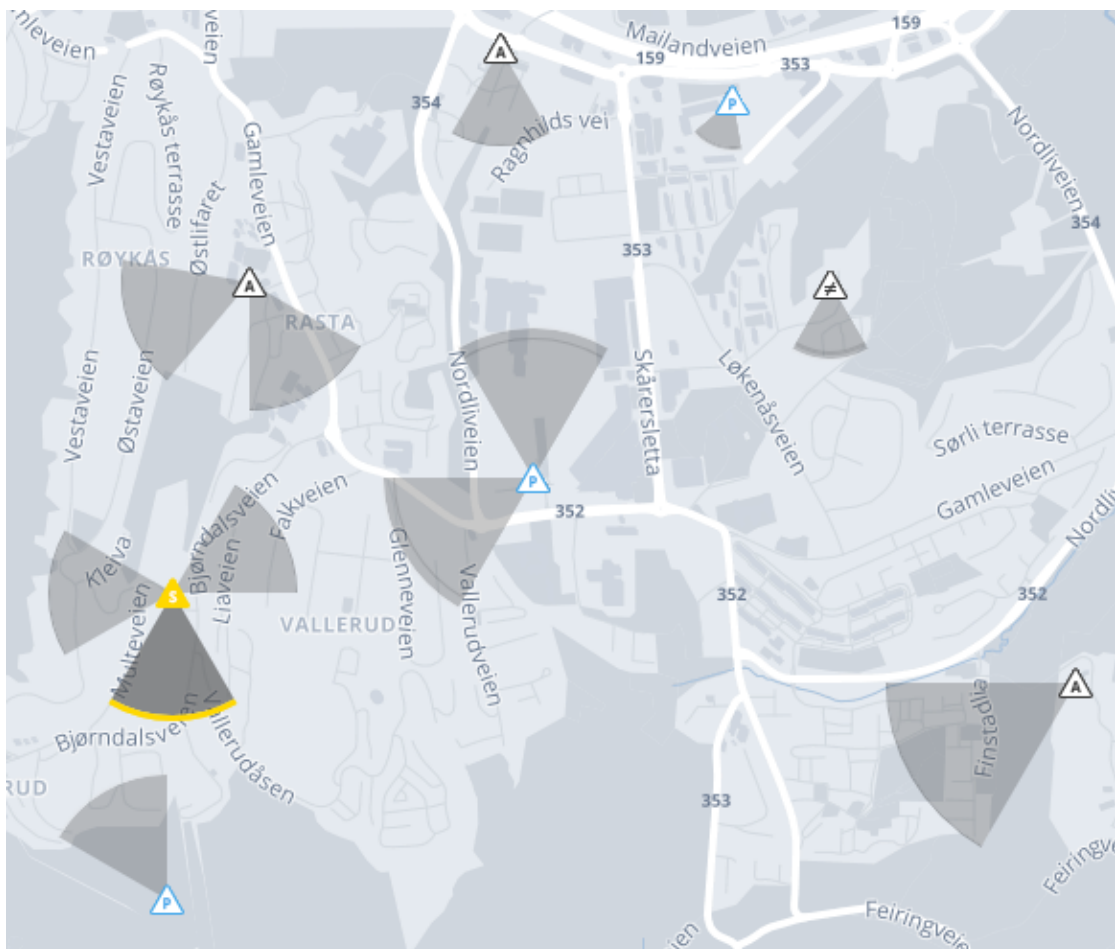


FIGURA 5.1: Excerto do mapa do Metric que mostra a célula escolhida (marcada em amarelo) e outras células a testar.

O sistema calculou a contagem de vizinhas para todas as células com um filtro de potência de sinal a -120 dBm. O ANR do operador utiliza uma métrica que relaciona o número de HOs para dentro/para fora que foram bem sucedidos num período de tempo para assistir na seleção dos melhores vizinhos; para esta análise, considerou-se o período de tempo entre junho de 2018 e junho de 2019.

### 5.1.3 Cenário grande de vizinhanças

Um cenário verdadeiro foi deliberadamente escolhido a partir de um grupo de duzentas e vinte e duas (222) células exteriores que partilham uma área urbana comum em Oslo, Noruega. Destas células, várias delas eram células planeadas, que não tinham sido instaladas ainda. Os critérios de seleção para estas células incluíram as suas distâncias entre cada uma, as suas áreas de cobertura e sobreposições entre cada uma, o rácio de células planeadas e instaladas e as suas tecnologias (todas as células usavam 4G). As células foram designadas genericamente entre *cell1* e *cell222*.

Deve-se notar que, fora desse grupo de células, haviam ainda outras sessenta e uma (61) células que não puderam ser usadas porque essas eram células interiores, que seguiam protocolos diferentes do que as exteriores. Células interiores são feitas especificamente para serem usadas em espaços fechados, e tipicamente não têm células exteriores nas suas NCLs; por exemplo, em vez de receber e averiguar uma NCL, um UE que entra num prédio com uma célula interior presente e começa a perder sinal a outra célula exterior começa a procurar cegamente por outra ligação, que vai quase sempre resultar numa ligação ao sinal mais forte disponível: a célula interior presente. Isto simplifica o cenário, mas considerando que existem utilizadores móveis que usam os seus equipamentos na rua ou enquanto viajam, continua a ser um cenário realista.

Deve ser notado que, no caso de células ativas, o ANR da operadora avaliou relações de vizinhança pelo número de HOs para/fora que aconteceram durante algum período de tempo, tal como no cenário pequeno, que foi especificado para



este teste como sendo entre o meio de setembro de 2019 e o meio de outubro de 2019. O mapa demonstrado na Figura 5.2 mostra uma perspectiva do cenário escolhido.

O sistema foi corrido quatro vezes com filtros de sinal diferentes, que foram (em dBm): -120, -100, -80 e -60. Estes filtros tiveram o propósito de ajudar a encontrar quais é que eram os vizinhos mais importantes através de relações de vizinhança que persistiram por vários filtros.

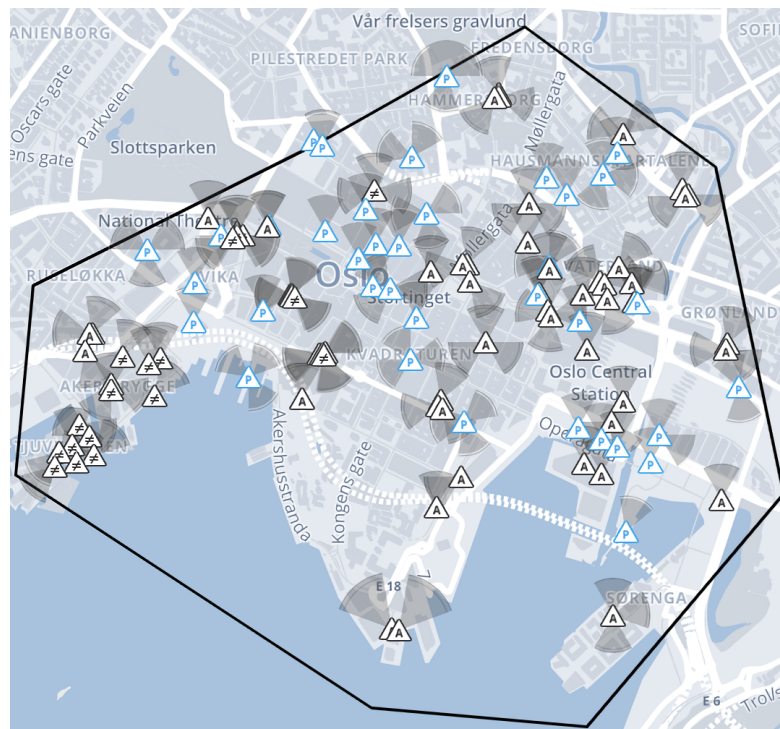


FIGURA 5.2: Excerto do mapa do Metric, mostrando as duzentas e vinte e duas células.

## 5.2 Resultados

Os algoritmos de interseção a comparar são o iterativo contra o uso de uma *Quad Tree*, enquanto que o estudo da construção de NCLs baseia-se não só na capacidade do algoritmo de agregação de pixels criar as ditas listas de vizinhas, mas também serão feitas comparações com os NCLs do sistema ANR existentes na rede.

O propósito da avaliação da construção de NCLs teve três objetivos fundamentais:

1. Avaliar o desempenho e resultados do sistema comparado ao ANR *standard* que o operador já utilizava e determinar que deliberações podem ser feitas entre eventos de HO e mobilidade dos utilizador;
2. Provar que o sistema consegue incorporar células planeadas em NCLs de células ativas;
3. Avaliar se consegue relações de vizinhança que sejam apropriadas (por exemplo, duas células que estejam muito longe de uma da outra não devem ter muitas contagens, se alguma de vizinhança nas suas NCLs).

Enquanto que o sistema calculou e salvou todas as relações de vizinhança entre todas as células, apenas um dos resultados para cada cenário será considerado, neste caso uma célula ativa para conseguir alcançar todos os objetivos.

### 5.2.1 Comparação de desempenho dos algoritmos de interseção

Foram contadas 1869 iterações para inserir as células do caso de teste descrito numa *Quad Tree*. Este número elevado deve à proximidade que cada célula tem com as outras, causando várias divisões na *Quad Tree* e reordenações das células. Para uma só célula, contaram-se:

- 98 iterações para o algoritmo de força bruta;
- 2027 iterações, contando com as 1869 iterações para a inserção das células, para o algoritmo com recurso a uma *Quad Tree*.

Enquanto que a diferença parece ser muito elevada, a procura da células da *Quad Tree* e conseqüente cálculo de interseções levou 158 iterações.

Na Tabela 5.1 e na Figura 5.3 demonstram-se comparações de iterações entre vários testes diferentes. Para fazer uma comparação mais concentrada na procura/interseção das células entre o algoritmo a força bruta e a *Quad Tree*, foi inserida uma coluna para mostrar o número de iterações para apenas pesquisas. Os valores dessa coluna foram obtidos subtraindo o valor equivalente de iterações da *Quad Tree* com 1869 iterações, proveniente do primeiro cenário.

Teste	Iterativo	<i>Quad Tree</i>	QT Só Pesquisas
1 Célula a planear	98	2027	158
2 Células a planear	195	2235	366
4 Células a planear	386	2642	773
8 Células a planear	756	3376	1507
16 Células a planear	1448	4738	2869
Todas as Células a planear	4851	23658	21789

TABELA 5.1: Demonstração dos resultados da contagem de iterações.

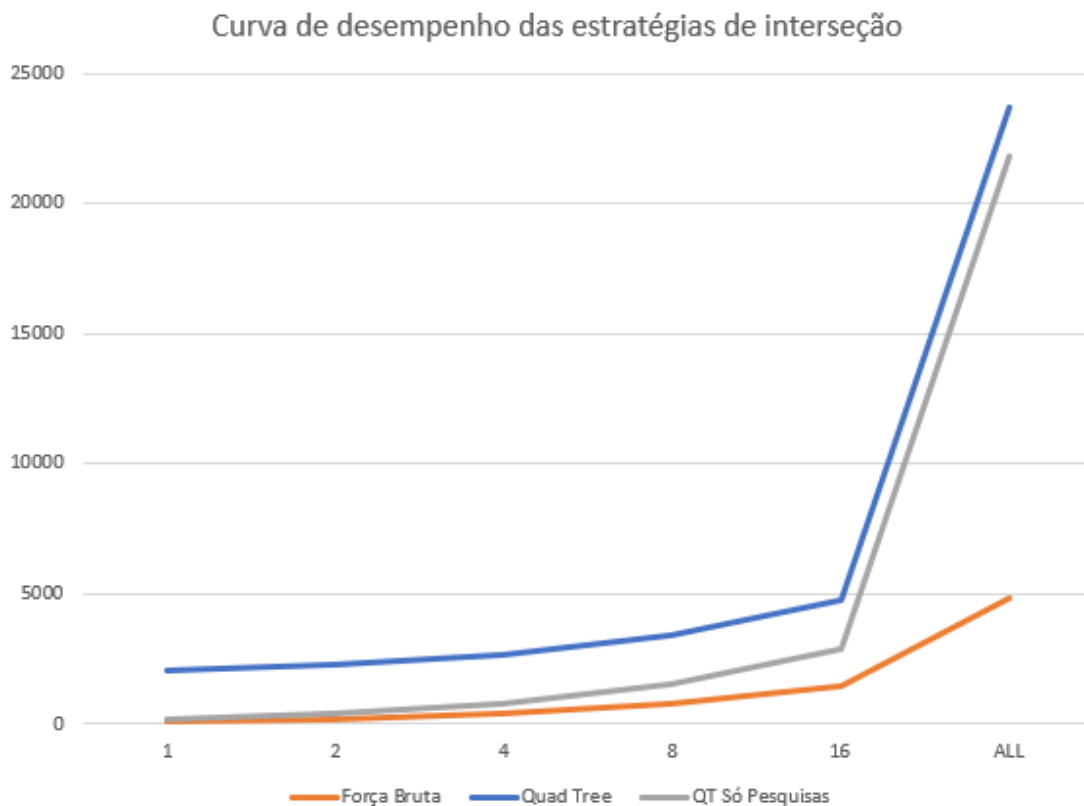


FIGURA 5.3: Demonstração das curvas de desempenho de acordo com a Tabela 5.1

Os resultados obtidos não foram os que eram esperados: enquanto que o algoritmo iterativo segue uma complexidade temporal, no seu melhor caso, aproximadamente de  $O(\frac{n^2}{2})$  (n sendo o tamanho da lista das células), o que era esperado do melhor caso da pesquisa da Quadtree seria um tempo próximo de  $O(\log n)$ , que apresentaria uma curva com declive muito menor que o algoritmo iterativo.

Esta discrepância no desempenho da Quadtree deve-se a um conjunto de factos que vêm da implementação utilizada:

- A natureza dos dados: O caso de teste usado não foi propriamente adequado para uma *Quad Tree*, pois os polígonos de cobertura das células estão todos muito próximos uns dos outros, causando quantidades excessivas de divisões e prejudicando o desempenho da *Quad Tree*;
- Cair repetidamente no pior caso: Em paralelo com a natureza dos dados, como o nível de profundidade ficou muito elevado, caiu-se consistentemente no pior caso de pesquisa na Quadtree,  $O(n \log n)$ , que é muito superior ao da força bruta;
- API pouco adequada: A API utilizada, mesmo tendo sido selecionada pela simplicidade da sua implementação, não foi a mais adequada para este caso de teste, pois foi desenhada primariamente para ser usada como um módulo para um detetor de colisões para videojogos 2D em Java. Infelizmente, durante a implementação da solução, não foi possível elaborar otimizações relevantes para o problema apresentado.

Durante a realização deste projeto, entendeu-se que não houve tempo ou recursos para resolver estes problemas, devido à complexidade apresentada. O uso de uma *Quad Tree* para este problema ainda pode ser explorado em outros casos, pois existem argumentos interessantes para o seu uso:

- Existência de um universo de células muito extenso: Num *use case* em que o mundo de células é muito grande (i.e. todo o globo), não faz sentido utilizar um algoritmo de força bruta porque a grande maioria das células não se

vão interseccionar umas com as outras. A *Quad Tree* possibilita a detecção de interseções de células num mundo com uma caixa de procura, limitando os cálculos a fazer apenas na área que se quer estudar;

- Universo de muitas células: O algoritmo de pesquisa na *Quad Tree*, em teoria, apresenta uma complexidade de  $O(n \log n)$  no seu pior caso, que é sempre melhor do que o algoritmo de força bruta ambos no seu melhor caso ( $O(\frac{n^2}{2})$ ) e no seu pior caso ( $O(n^2)$ ). Num ambiente de milhares de células, esta diferença torna-se relevante.

### 5.2.2 Análise de desempenho no cenário pequeno de vizinhas

Este teste explora dois conjuntos de resultados: um componente de comparação que utiliza dados do ANR e o componente testado que utiliza dados do sistema.

À esquerda na Figura 5.4, um mapa topográfico da área escolhida é representada. A célula escolhida tem um contorno amarelo, as suas vizinhas recomendadas pelo ANR têm um contorno colorido não-amarelo, e todas as outras células não têm contornos; à direita está representada a NCL 4G da célula escolhida, organizado por uma linha por relação de vizinhança com outra célula. Cada linha tem dois parâmetros que são representados por uma fração de HOs bem sucedidos contra todos os HOs tentados. O primeiro parâmetro representa essa fração de HOs entre a célula escolhida com a célula da linha (ou seja, HOs para fora), enquanto que o segundo parâmetro representa o contrário (HOs para dentro).

Na Tabela 5.2, as diferenças entre as relações de vizinhança são aparentes, considerando que o ANR reflete a mobilidade dos seus utilizadores e os HOs feitos, enquanto que o sistema implementado toma em atenção as forças de sinal ao longo de uma grelha topográfica. As melhores vizinhas para o sistema são determinadas por sobreposições de coberturas entre células para determinar qual é que pode providenciar serviços ao utilizador em que localizações, que é útil para escolher quais células são boas candidatas para HOs em situações específicas.

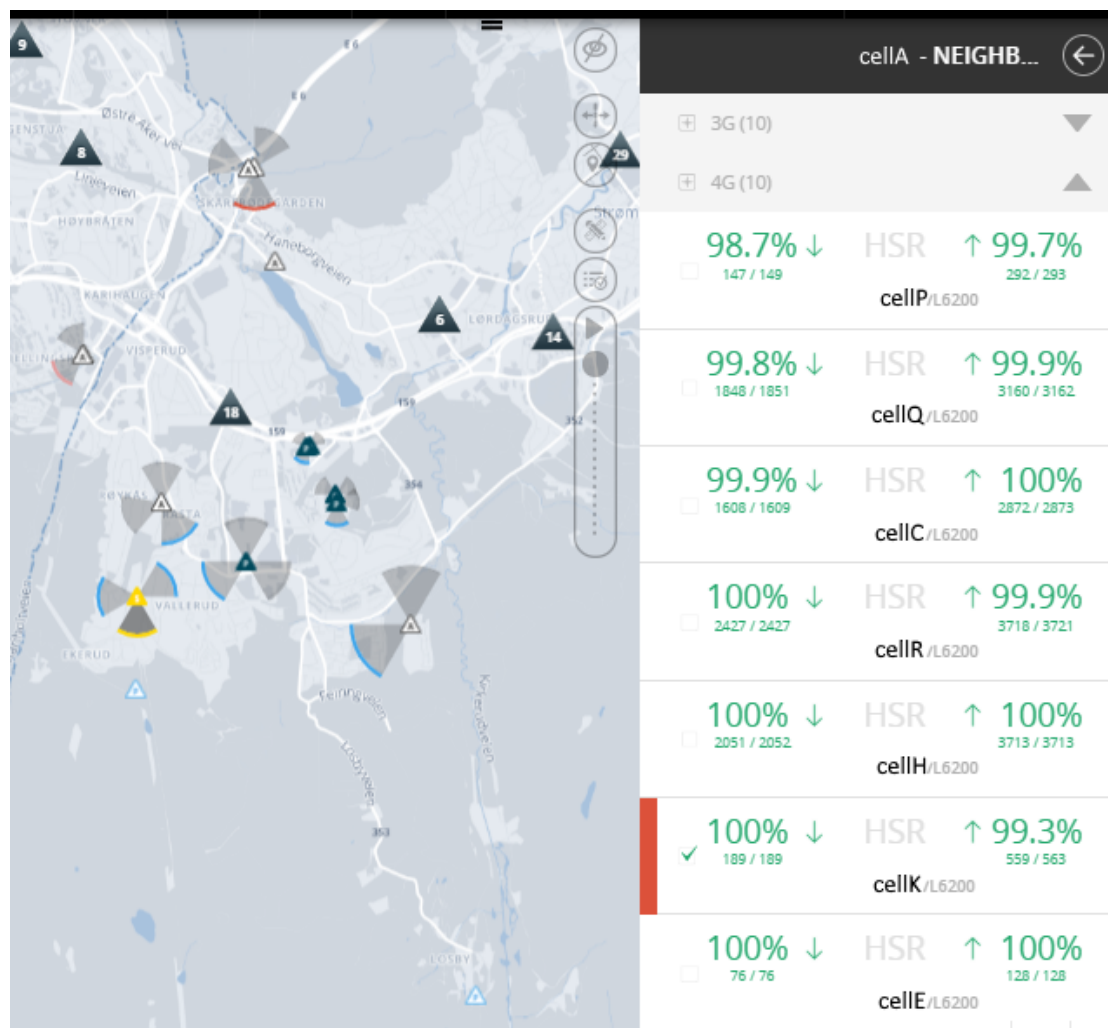


FIGURA 5.4: As vizinhas recomendadas pelo ANR são assinaladas por um contorno colorido e o seu rácio de HO's bem sucedidos está à direita.

Uma quantidade excessiva de células sobrepostas não é sempre preferível: por exemplo, em áreas de pouco tráfego, não há muita razão para haver demasiada sobreposição pois é um desperdício de energia providenciar serviços num lugar que é pouco usado. Este aparte também é importante por causa de mais dois factos:

1. Um site em particular que tem células apontadas para direções diferentes vai sistematicamente sobrepor-se consigo próprio numa maneira consistente, que é explicitado pela sobreposição demonstrada e alto tráfego de HO's entre a célula estudada e as suas "irmãs" (*cellQ* e *cellR*), de acordo com as diferenças mostradas na Tabela 5.2. Isto significa que estas configurações multi-células devem ser exploradas ao seu potencial máximo para cobrir uma área específica e não sobreponem-se com outras áreas sem necessidade;

2. A sobreposição pode não ser imediatamente previsível quando se concebe uma instalação em termos teóricos; ou seja, na prática haverá imprecisões por influências geográficas. O sistema implementado permite a um operador encontrar esses problemas e fazer decisões informadas a partir de tal.

TABELA 5.2: Comparação da contagem de vizinhas da *cellA* pelo sistema implementado e a distribuição de tráfego de HO de acordo com o ANR do operador.

Nome da vizinha	Rácio (-120 dBm) [%]	Tráfego HO [%]
cellH	84.78	25.29
cellG	83.32	0.00
cellO	68.64	0.00
cellE	68.00	0.90
cellN	65.49	0.00
cellJ	64.19	0.00
cellF	40.69	0.00
cellP	39.93	1.93
cellQ	39.63	21.98
cellK	33.11	3.28
cellM	32.24	0.00
cellB	27.45	0.00
cellR	27.23	26.97
cellC	19.64	19.66
cellI	7.08	0.00
cellD	5.15	0.00
cellL	4.73	0.00

A sinergia entre o ANR e o sistema implementado é evidente comparando o rácio de HOs e o rácio de relações de vizinhança detetadas, pois as "irmãs" da *cellA* (*cellQ* e *cellR*) partilham relações de vizinhança e rácios de HOs bem sucedidos elevadas. Isto sugere que as melhores vizinhas são aquelas que partilham uma localização muito próxima e comum.

Outros exemplos, como a da *cellC*, revela um rácio de HOs relativamente alto mas uma relação de vizinhança baixa, mesmo estando perpendicular e perto à *cellA*; comparando com a *cellH*, que está muito mais longe da *cellA*, mas apresenta valores muito maiores. Este tipo de caso pode indicar uma falha de cobertura, e esta em particular é causada por diferenças em elevação, como demonstrado pela representação topográfica na Figura 5.5.

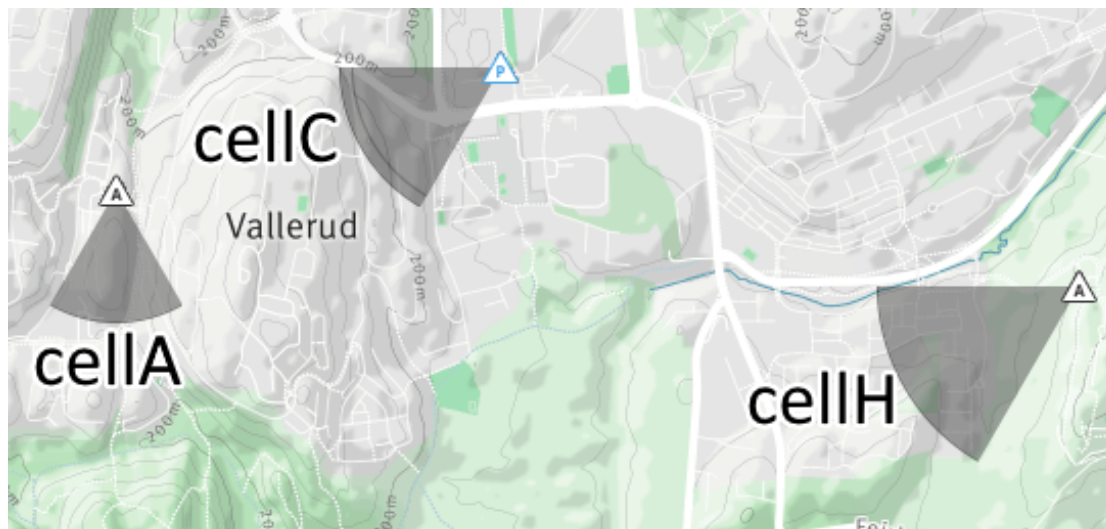


FIGURA 5.5: Demonstrações da *cellA*, *cellC* and *cellH* no mapa do Metric.

### 5.2.3 Análise de desempenho no cenário grande de vizinhanças

O teste juntou os resultados do sistema e do ANR para comparar o rácio de HOs bem sucedidos contra os quatro valores de cobertura de sinal de acordo com os filtros parametrizados. Como relações de vizinhança foram detetadas entre quase todas as células para o sistema no filtro mais sensível, resultando em mais de duzentas entradas, apenas os maiores vinte resultados destas categorias serão apresentados:

- Percentagem de HOs para dentro (Figura 5.6);
- Percentagem de HOs para fora (Figura 5.7);
- Percentagem de cobertura a -120 dBm (Figura 5.8);
- Percentagem de cobertura a -100 dBm (Figura 5.9);
- Percentagem de cobertura a -80 dBm (Figura 5.10);
- Percentagem de cobertura a -60 dBm (Figura 5.11).



Deve ser notado que a cobertura partilhada mostrada em cada Figura é exclusivamente entre a cobertura da *cell1* e as células mostradas nas Figuras; isto significa que, por exemplo, enquanto que a *cell1* partilha uma certa percentagem da sua cobertura com uma *cellX*, isso não significa necessariamente que essa *cellX* partilha o mesmo rácio de cobertura com a *cell1*, pois pode ocupar mais espaço na realidade do que esta.

Os resultados evidenciam que existe baixa correlação entre uma sobreposição de cobertura elevada e a taxa de sucessos de HOs, para além de algumas coincidências como a *cell30* e a *cell28* nas Figuras 5.6 e 5.8. Isto é devido ao ANR utilizar dados de mobilidade de UEs através de várias células, enquanto que este sistema utiliza dados de sobreposição de cobertura numa grelha bi-dimensional.

Nas Figuras 5.8 até 5.11 existem várias células que apresentam um rácio de HOs quantificado por "N/A", que significa que nenhum HO foi feito entre a *cell1* e essas células. Isto pode significar uma de três hipóteses:

1. Não existe nenhum caminho discernível ou conveniente para os utilizadores móveis atravessarem para justificar HOs entre essas duas células;
2. A célula está numa zona de pouco tráfego;
3. A célula é uma célula planeada e/ou ainda não foi ativada.

Dado que este cenário apresenta uma localidade urbana densamente habitada, a primeira opção é provável considerando que provavelmente há mais células no caminho que um UE considera ser o melhor candidato para um HO; um número excessivo de HOs para mudar de uma célula para a outra não é desejável, mesmo que implique saltar de uma célula para a outra que não partilhe tanta cobertura com a anterior. Mais, uma quantidade excessiva de sobreposição também não é desejável, especialmente se o rácio de HOs não o justificar (tal como em vários exemplos com maiores contagens nas Figuras 5.8-5.11), uma vez que constitui um desperdício de energia e o sinal podia ser utilizado melhor noutro lado. Assim, o

sistema é capaz de providenciar dados úteis para analisar *trends* do ANR e para encontrar sobreposições de cobertura excessivos.

Uma observação expectável feita clara pelos gráficos de dispersão das sobreposições de cobertura (Figuras 5.8-5.11) é que existe uma tendência para os rácios de sobreposição decrescerem mais depressa o quão menos sensível o filtro do sinal for. Por exemplo, as vinte células do topo na Figura 5.8 não descem de uma sobreposição a 80%, enquanto que as da Figura 5.11 seguem um decréscimo muito mais dramático até abaixo de 50%. Para além disso, os rácios aparentemente são distribuídos aleatoriamente, mas há uma pequena tendência para os filtros de sinal mais pequenos partilharem rácios semelhantes.

Não foi a norma, mas o sistema pôde reconhecer potenciais relações de vizinhança entre a *cell1* e algumas células planeadas, tal como a *cell54* nas Figuras 5.10 e 5.11, e a *cell83* na Figura 5.11. Muitos mais exemplos existem, mas esses não chegaram às vinte entradas de topo pois apresentam menores rácios de sobreposição de coberturas, que foi provavelmente consequência de falhas de cobertura ou obstáculos densos a interceptarem esses sinais e não um problema de distância.

Os resultados sugerem que o sistema pode servir como um ponto de partida para novas redes, já que consegue calcular relações de vizinhança entre células existentes e planeadas através de estimações das suas áreas de cobertura. Atua como um complemento ao ANR, que funciona melhor em redes existentes, enquanto que este sistema enche a lacuna entre redes novas ou planeadas; os resultados podem ajudar em outras formas para além de relações de vizinhança, como deteção de *overshooting* (ou seja, uma célula que apresenta uma cobertura bem além dos seus limites esperados. O sistema pode detetar sobreposições inadequadas consequentes desse efeito).

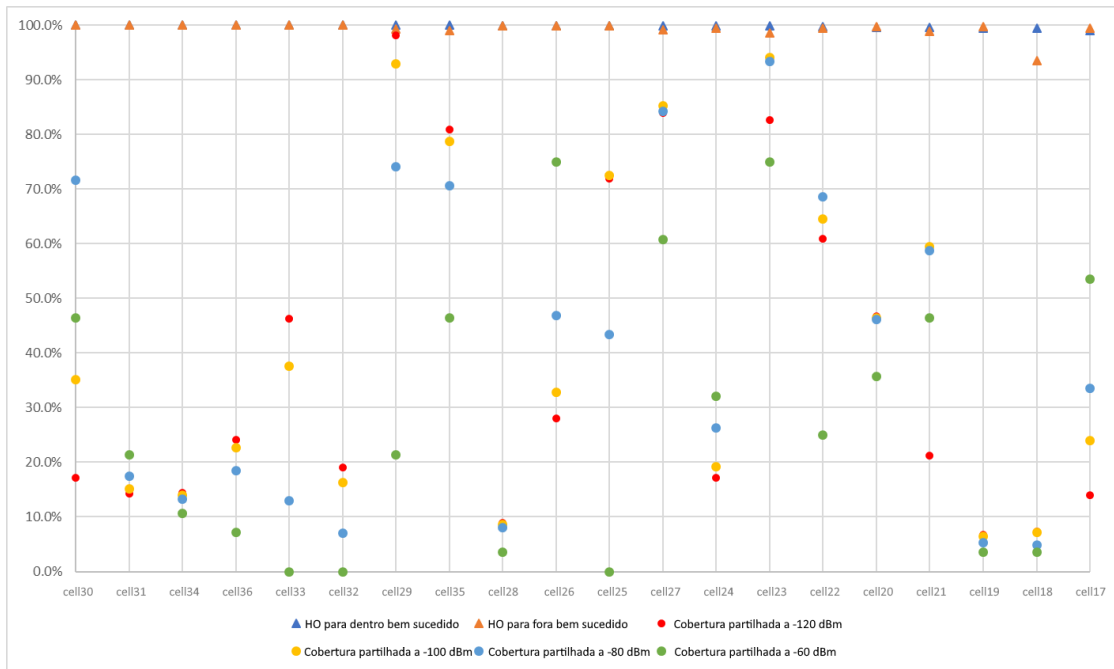


FIGURA 5.6: Dispersão entre as vinte células de topo do mais alto ao mais baixo rácio de HOs para dentro bem sucedidos.

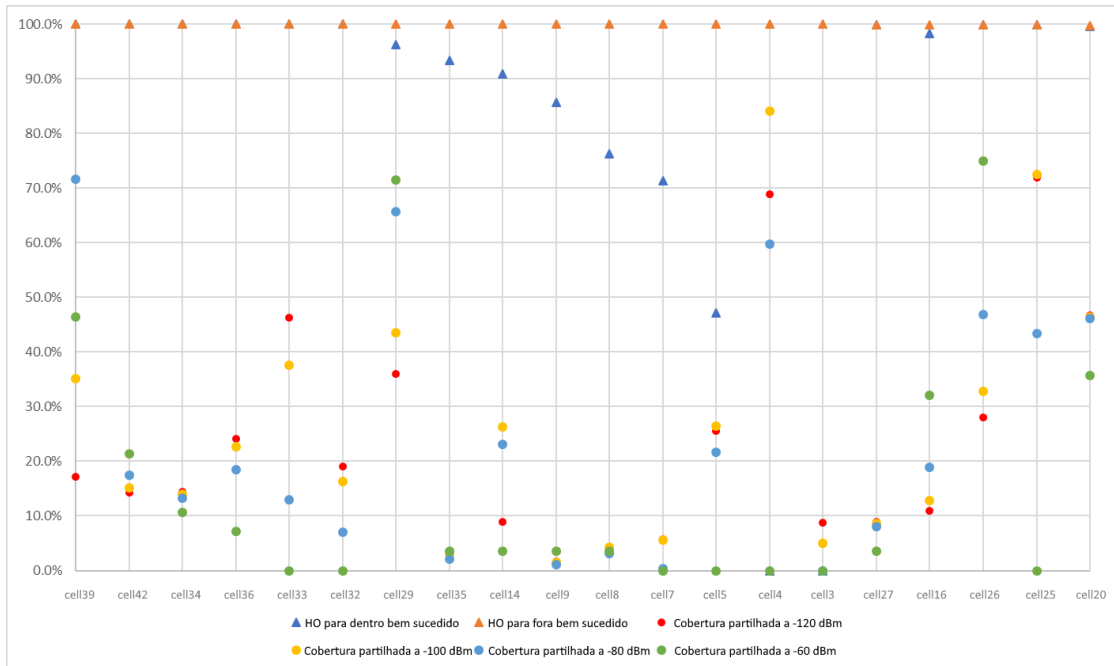


FIGURA 5.7: Dispersão entre as vinte células de topo do mais alto ao mais baixo rácio de HOs para fora bem sucedidos.

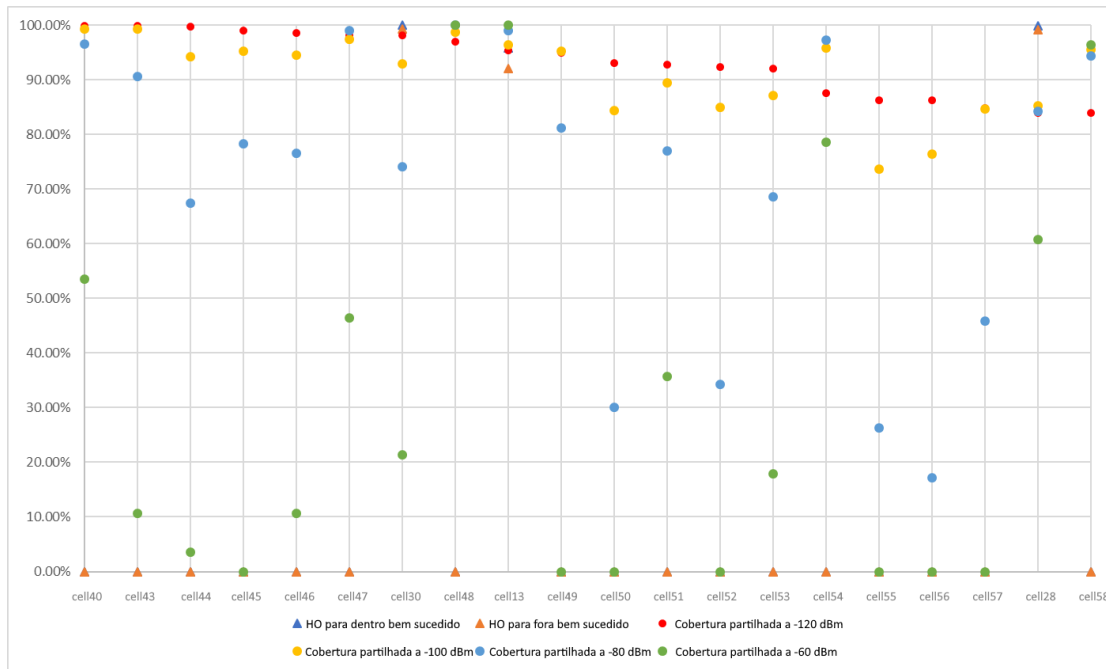


FIGURA 5.8: Dispersão entre as vinte células de topo do mais alto ao mais baixo rácio de cobertura partilhada a -120 dBm.

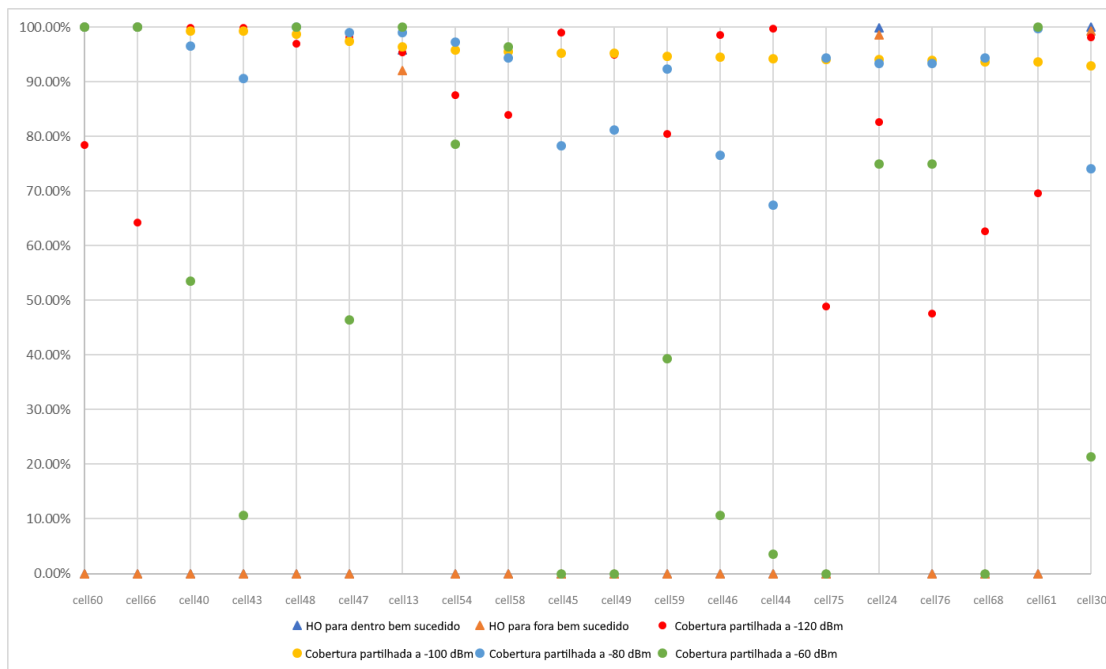


FIGURA 5.9: Dispersão entre as vinte células de topo do mais alto ao mais baixo rácio de cobertura partilhada a -100 dBm.

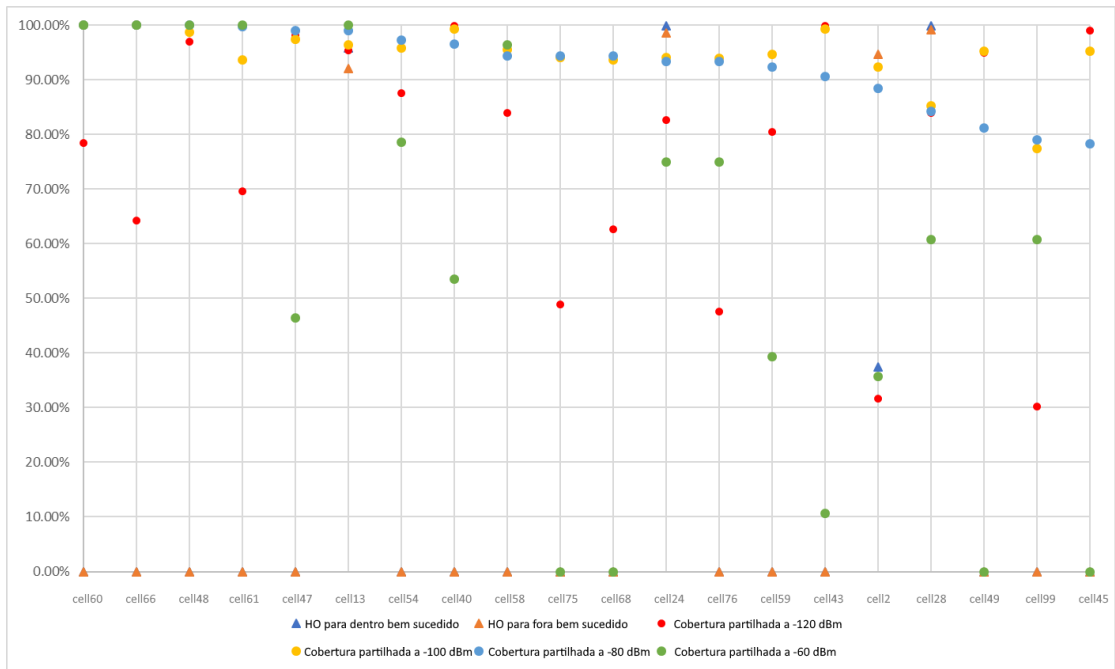


FIGURA 5.10: Dispersão entre as vinte células de topo do mais alto ao mais baixo rácio de cobertura partilhada a -80 dBm.

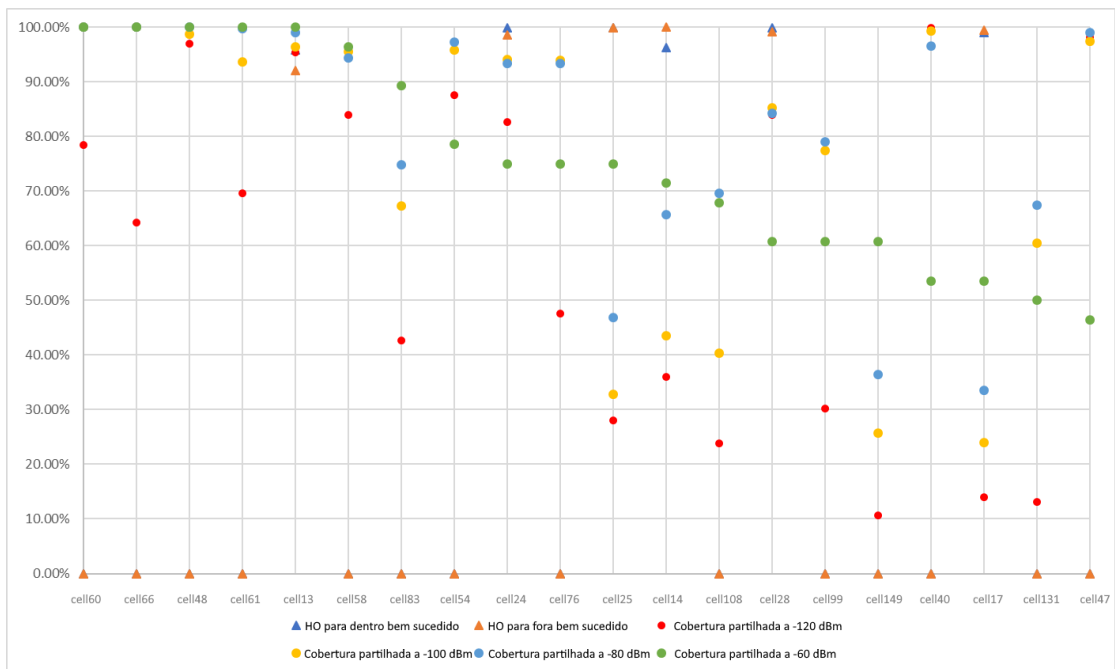


FIGURA 5.11: Dispersão entre as vinte células de topo do mais alto ao mais baixo rácio de cobertura partilhada a -60 dBm.



# Capítulo 6

## Conclusões

Com esta dissertação foi proposto, conceptualizado, e implementado um sistema otimizado para a estimativa de relações de vizinhança em redes móveis baseada na cobertura geográfica baseado na *cloud*, integrado no Metric SaaS para o uso no planeamento automático de vizinhos em redes celulares (sejam estas planeadas, existentes, ou uma mistura), limpeza e otimização de NCLs, e medições de sobreposição em grelhas de pixeis. O seu agnosticismo tecnológico permite a sua utilização em qualquer geração de redes celulares, e pode ainda sugerir boas relações de vizinhança entre células de tecnologias diferentes, se for esse o contexto do *input* do operador de rede.

Este sistema é composto por quatro módulos, mas o âmbito da dissertação concentrou-se em dois que foram desenvolvidos pelo autor. O primeiro, o módulo de interseção das grelhas de células, recebe bordas de grelhas de células com coordenadas geográficas codificadas por *GeoHashes* de um módulo de estimação da cobertura celular, para depois enviar uma lista de interseções das células a um módulo de computação de mapas-mundo; o segundo, o módulo de identificação de vizinhas, recebe um mapa-mundo que localiza as células e as suas potências de sinal apresentadas num mundo dividido por pixeis e estabelece relações de vizinhança dependendo de um filtro de sinal parametrizado pelo utilizador, assim criando NCLs para cada célula do mundo dado.

Para testar todas estas funcionalidades, foram feitos testes de aceitação com dados simulados, e para testar o seu desempenho isoladamente e comparado com sistemas ANR existentes, foram usados cenários urbanos realistas. A comparação principal feita foi entre as tendências de mobilidade dos utilizadores contra a sobreposição de coberturas entre células, que não sugeriu nenhuma correlação entre as duas. De facto, o sistema conseguiu detetar potenciais relações de vizinhança entre uma célula ativa e várias células planeadas/ativas, provando a sua capacidade de planear muitas células ao mesmo tempo. Esta capacidade é muito útil para o planeamento de redes, pois os cálculos manuais para determinar essas relações podem ser lentos. A sua facilidade de personalização faz com que este sistema seja apropriado para a estimação de relações de vizinhança e rácios de cobertura em qualquer configuração que um planeador de rede tenha necessidade.

Este sistema traz valor significativo aos paradigmas de auto-configuração e auto-otimização de redes auto-organizadas devido ao seu aspeto automatizado e aos dados que contribui, que seriam difíceis de descobrir de outras formas, reduzindo custos no planeamento, otimização e manutenção de redes. O sistema também beneficia do uso de serviços *cloud* devido a esse aspeto, conseguindo ser executado em paralelo por vários clientes entre várias configurações.

Apesar do sistema ter um impacto muito interessante para o planeamento, este carece de mais desenvolvimento para aumentar ainda mais a otimização na sua globalidade para planeamentos de muito larga escala. A versão corrente é dependente do AWS, e algum tempo deverá ser gasto para adaptar-lo para outros serviços *cloud*. Algumas extensões podem ser feitas para adicionar outros filtros para além da potência de sinal, e.g. localizações geográficas, fabricantes de redes e gerações tecnológicas.



# Referências

- [1] Tomi T Ahonen e Joe Barrett. *Services for UMTS: Creating killer applications in 3G*. John Wiley & Sons, 2002.
- [2] Amazon. *Amazon Web Services*. URL: <https://aws.amazon.com> (acedido em 24/06/2020).
- [3] Nurul'Ain Amirrudin et al. "Analysis of handover performance in LTE femto-cells network". Em: *Wireless Personal Communications* 97.2 (2017), pp. 1929–1946.
- [4] R Arshad et al. "Handover management in dense cellular networks: A stochastic geometry approach". Em: *2016 IEEE International Conference on Communications (ICC)*. Mai. de 2016, pp. 1–7. DOI: 10.1109/ICC.2016.7510709.
- [5] E Balagurusamy. *Programming with JAVA*. Vol. 6. McGraw-Hill Education, 2019.
- [6] D Clemente et al. "Implementation of a Cloud-based, Traffic Aware and Energy Efficient Management of Base Stations' Activity". Em: *2018 21st International Symposium on Wireless Personal Multimedia Communications (WPMC)*. Nov. de 2018, pp. 600–605. DOI: 10.1109/WPMC.2018.8713106.
- [7] A Dahlen et al. "Evaluations of LTE Automatic Neighbor Relations". Em: *2011 IEEE 73rd Vehicular Technology Conference (VTC Spring)*. Mai. de 2011, pp. 1–5. DOI: 10.1109/VETECS.2011.5956600.

- [8] D Duarte et al. “Neighbour list optimization for real LTE radio networks”. Em: *2014 IEEE Asia Pacific Conference on Wireless and Mobile*. Ago. de 2014, pp. 183–187. DOI: 10.1109/APWiMob.2014.6920278.
- [9] Daniel Fernandes et al. “Combining Measurements and Propagation Models for Estimation of Coverage in Wireless Networks”. Em: *2019 IEEE 90th Vehicular Technology Conference*. Honolulu, HI, USA: IEEE, set. de 2019.
- [10] John Edward Flood. *Telecommunication networks*. IET, 1997.
- [11] Apache Software Foundation. *Apache Maven*. URL: <https://maven.apache.org/> (acedido em 24/06/2020).
- [12] Amitabha Ghosh et al. “Heterogeneous cellular networks: From theory to practice”. Em: *IEEE communications magazine* 50.6 (2012), pp. 54–64.
- [13] André Godinho et al. “Cloud-based Cellular Network Planning System : Proof-of-Concept Implementation for GSM in AWS”. Em: *THE 22ND INTERNATIONAL SYMPOSIUM ON WIRELESS PERSONAL MULTIMEDIA COMMUNICATIONS* (2019).
- [14] Google. *Gson - A Java serialization/deserialization library to convert Java Objects into JSON and back*. <https://github.com/google/gson/>. 2019.
- [15] Harri Holma e Antti Toskala. *WCDMA for umts: hspa evolution and lte*. John Wiley & sons, 2007.
- [16] *JSON*. URL: <https://www.json.org/json-en.html> (acedido em 24/06/2020).
- [17] Aamod Khandekar et al. “LTE-Advanced: Heterogeneous networks”. Em: *2010 European Wireless Conference (EW)*. IEEE. 2010, pp. 978–982.
- [18] Dooyoung Kim et al. “Self-configuration of neighbor cell list utilizing E-UTRAN NodeB scanning in LTE systems”. Em: *2010 7th IEEE Consumer Communications and Networking Conference*. IEEE. 2010, pp. 1–5.
- [19] Alexander Kukushkin. *Introduction to Mobile Network Engineering: GSM, 3G-WCDMA, LTE and the Road to 5G*. John Wiley & Sons, 2018.

- [20] Wan Lei et al. “Heterogeneous network in LTE-advanced system”. Em: *2010 IEEE International Conference on Communication Systems*. IEEE. 2010, pp. 156–160.
- [21] Aminu Wada Maida, Alan James Auchmuty Carter e Housam Al Housami. *Creating neighbour cell lists*. 2013.
- [22] John D Matyjas, Fei Hu e Sunil Kumar. *Wireless Network Performance Enhancement via Directional Antennas: Models, Protocols, and Systems*. CRC Press, 2015.
- [23] R. Moussalli, M. Srivatsa e S. Asaad. “Fast and Flexible Conversion of Geohash Codes to and from Latitude/Longitude Coordinates”. Em: *2015 IEEE 23rd Annual International Symposium on Field-Programmable Custom Computing Machines*. 2015, pp. 179–186.
- [24] Multivision. *Metric*. 2019. URL: <https://www.metric.pt> (acedido em 15/07/2019).
- [25] Multivision. “OptiNET-5G - Planning and Optimization Framework of 5G Heterogeneous Networks in a Cloud Environment”. Em: *Project 023304, Funded by Centro2020, Portugal2020 and European Union* ().
- [26] Van Minh Nguyen e Holger Claussen. “Efficient self-optimization of neighbour cell lists in macrocellular networks”. Em: *21st Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*. IEEE. 2010, pp. 1923–1928.
- [27] Olav Østerbø e Ole Grøndalen. “Benefits of Self-Organizing Networks (SON) for Mobile Operators”. Em: *Journal of Computer Networks and Communications* 2012 (2012). Ed. por Guillaume de la Roche, p. 862527. ISSN: 2090-7141. DOI: 10.1155/2012/862527. URL: <https://doi.org/10.1155/2012/862527>.
- [28] Ryan Pelletier. *Java Simple QuadTree*. <https://github.com/ryanp102694/java-simple-quadtree/>. 2018.

- [29] Rouzbeh Razavi, David López-Pérez e Holger Claussen. “Neighbour cell list management in wireless heterogeneous networks”. Em: *2013 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE. 2013, pp. 1220–1225.
- [30] Sanam Sadr e Raviraj S Adve. “Handoff rate and coverage analysis in multi-tier heterogeneous networks”. Em: *IEEE Transactions on Wireless Communications* 14.5 (2015), pp. 2626–2638.
- [31] M. Sawahashi et al. “Broadband radio access: LTE and LTE-advanced”. Em: *2009 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*. 2009, pp. 224–227.
- [32] Serverless. *Serverless Framework*. URL: <https://serverless.com> (acedido em 24/06/2020).
- [33] Yoonsu Shin e Songkuk Kim. “Virtualized ANR to Manage Resources for Optimization of Neighbour Cell Lists in 5G Mobile Wireless Networks”. Em: *Mobile Information Systems 2017* (2017).
- [34] The JUnit Team. *JUnit 5*. URL: <https://junit.org/junit5/> (acedido em 24/06/2020).
- [35] WeiTek Tsai. “Software-as-a-service (SaaS) perspectives and challenges”. Em: (2014).
- [36] M Turner, D Budgen e P Brereton. “Turning software into a service”. Em: *Computer* 36.10 (out. de 2003), pp. 38–44. ISSN: 0018-9162. DOI: 10.1109/MC.2003.1236470.
- [37] M. D. Yacoub. *Foundations of Mobile Radio Engineering*. CRC Press (1768), 1993.