



Instituto Universitário de Lisboa

Departamento de Ciências e Tecnologias da Informação

**Algoritmo eficiente para
planeamento de frequências de redes
móveis celulares, suportado por
programação linear**

André Frias Godinho

Dissertação submetida como requisito parcial para obtenção do grau
de

Mestre em Engenharia Informática

Orientador

Professor Doutor Pedro Sebastião, Professor Auxiliar
ISCTE-IUL

Co-Orientador

Professor Doutor Lúcio Studer Ferreira, Professor Auxiliar
INESC-ID

Agosto 2020

"The difference between the novice and the master is that the master has failed more times than the novice has tried."

Yuusei Matsui

Resumo

O aumento do número de subscritores móveis em todo o mundo causou um crescimento da necessidade de novos algoritmos de planeamento/otimização de redes móveis, rápidos e fiáveis, especialmente porque, de forma a manter a qualidade de serviço de uma rede móvel, uma operadora terá que colocar mais equipamento no terreno. Este trabalho apresenta uma forma rápida e fiável de, automaticamente, planear as frequências em redes móveis celulares, usando tecnologias *cloud* e programação linear. O algoritmo proposto foi testado num cenário realístico de uma rede GSM (sendo generalizável e de utilização em outras gerações móveis), e os resultados foram comparados com outra ferramenta de planeamento comercial. Os resultados mostram que, apesar do desempenho no planeamento de um site ser similar, o novo algoritmo foi doze vezes mais rápido e usou quatro vezes menos memória de acesso aleatório. O algoritmo desenvolvido, conseguiu fazer a otimização do planeamento numa rede móvel celular de setenta células em menos de 3 minutos (contra mais de 5 horas com a utilização de uma ferramenta comercial), permitindo uma eficiência superior a 100 vezes, i.e., com um consumo de tempo inferior a 1% do tempo necessário com a ferramenta comercial. O algoritmo foi integrado na ferramenta comercial Metric e está atualmente a ser utilizado para planeamento celular.

Palavras-chave: Comunicações Móveis Celulares, Monitorização, Otimização, Planeamento de Frequências, Programação Linear Inteira, Serviços Cloud.

Abstract

With the increasing number of mobile subscribers worldwide, there is a need for fast and reliable algorithms for planning/optimization of mobile networks, especially because, in order to maintain a network's quality of service, an operator might need to deploy more equipment. This work presents a quick and reliable way to automatically plan frequencies in cellular networks, using both cloud technologies and linear programming. We evaluate our pattern in a realistic scenario of a GSM network (it can also be used in other mobile generation) and compare the results to another already implemented commercial tool. Results show that even though network quality was similar, our algorithm was twelve times faster and used four times less memory. It was also able to frequency plan seventy cells simultaneously in less than three minutes (against the 5 hours needed by a commercial tool). This mechanism was successfully integrated in the professional tool Metric, and is currently being used for cellular planning.

Keywords: Cellular mobile communications, Cloud-services, Frequency, Planning, Integer Linear-programming, Monitoring, Optimization.

Agradecimentos

Aos meus orientadores, Professor Doutor Pedro Sebastião e Professor Doutor Lúcio Ferreira, não só pelas oportunidades pedagógicas fornecidas, como também todo o apoio prestado no desenvolvimento e escrita da dissertação.

À Multivision, em especial ao Nuno Valente, Sameer Ibraimo, Ricardo Costa e Gabriela Soares, pelas suas sugestões, partilha de conhecimento e apoio oferecido.

Ao Paulo Pina, Daniel Fernandes, Diogo Clemente e Rodrigo Cortesão, participantes do mesmo projeto, com os quais partilhei dificuldades e soluções.

Um agradecimento especial á minha família, pelo máximo de apoio que podiam ter oferecido nos momentos mais difíceis, e em particular, á minha namorada, pelas palavras corretas no momento certo.

Por fim, gostaria de agradecer aos meus colegas de curso com que partilhei todas estas experiências, em especial, aos meus amigos, Ricardo Silva, Miguel Rodrigues, Miguel Nobre e Duarte Vital.

Conteúdo

Resumo	v
Abstract	vii
Agradecimentos	ix
Lista de Figuras	xii
Abreviaturas	xvi
1 Introdução	1
1.1 Motivação	1
1.2 Enquadramento e Objetivos	1
1.3 Abordagem metodológica e questões de investigação	2
1.4 Contribuições	3
1.5 Estrutura	4
2 Estado de Arte	7
2.1 Sistemas de Comunicação Móvel Celular para o GSM	7
2.2 Gestão de Redes Móveis Celulares	9
2.2.1 Planeamento, monitorização e otimização	9
2.2.2 Redes Auto-Organizadas	10
2.2.3 Serviços Cloud para Gestão e Planeamento da Rede	11
2.2.4 Metric	13
2.3 Planeamento de Sistemas de Comunicação Móveis Celulares	14
2.3.1 Planeamento Celular	14
2.3.2 Planeamento de frequências	17
2.3.3 Algoritmo de Planeamento de Frequências Metric	19
2.4 Algoritmos de Apoio ao Planeamento	24
2.4.1 Z-order	24
2.4.2 Geohash	24

2.4.3	Programação Linear	26
3	Modelo e Algoritmos de Planeamento de Frequências	29
3.1	Requisitos	29
3.2	Visão Geral do Modelo	30
3.3	Módulo de Pré-processamento	31
3.4	Módulo de congregação das células	33
3.5	Módulo de planeamento de frequências	34
3.5.1	Cálculo de interferência	34
3.5.2	Planeamento de frequências	36
4	Implementação	39
4.1	Tecnologias e ferramentas utilizadas	39
4.1.1	Tecnologias	39
4.1.2	Ferramentas	40
4.2	Implementação	41
4.3	Integração no Metric	45
4.4	Testes	45
5	Resultados	51
5.1	Cenários de Referência	51
5.2	Análise de resultados para planeamento de um único site	52
5.3	Análise de resultados para planeamento de múltiplos sites	55
6	Conclusão	59
6.1	Principais conclusões	59
6.2	Propostas de investigação futura	61

Lista de Figuras

Figura 2.1: Ciclo de vida das SON.	11
Figura 2.2: Estrutura de serviços <i>cloud</i> num modelo de negócio.	13
Figura 2.3: Interface do mapa do Metric SaaS.	14
Figura 2.4: Cobertura celular teórica e realista.	15
Figura 2.5: Esquerda: Setorização teórica em 3 partes. Direita: Setorização real com <i>Overlap</i>	17
Figura 2.6: <i>Workflow</i> do algoritmo de planeamento de frequências Metric. . .	19
Figura 2.7: Zona de visibilidade.	21
Figura 2.8: Área definida pelo ' <i>Boundaries</i> '.	21
Figura 2.9: Demonstração da divisão das áreas.	23
Figura 2.10: O uso da <i>Z-order curve</i> para interpolar a representação binário dos inteiros de 0 a 7.	25
Figura 2.11: Divisão do mundo usando o algoritmo Geohash.	25
Figura 3.1: Modelo para realizar o planeamento de frequências.	31
Figura 3.2: Exemplo de cobertura das células onde se pretende planear a célula A.	32
Figura 3.3: Representação da estimativa de sinal recebido de uma antena calibrada com <i>drive tests</i>	32
Figura 3.4: Representação do mapa mundo e exemplo de escrita de ficheiro numa célula.	33
Figura 3.5: Exemplo de uma área de interferência tendo em conta o decaimento de potência ao longo de espaço assumindo que A e B tem a mesma frequência.	35
Figura 4.1: Mapa de tecnologias e ferramentas usadas.	40
Figura 4.2: Integração dos serviços AWS no novo algoritmo.	42
Figura 4.3: Pedido de planeamento de frequências no Metric.	45
Figura 4.4: Exemplo de ficheiro de <i>output</i> do sistema.	46

Figura 5.1: Cenário de avaliação utilizado, extraído de uma existente rede GSM.	52
Figura 5.2: C/I da área coberta pelo <i>site</i> planejado.	53
Figura 5.3: Função de distribuição do C/I dentro da área coberta pelo <i>site</i> planejado.	54
Figura 5.4: Comparação do novo algoritmo com o velho algoritmo, tanto em tempo de execução como espaço ocupado.	55
Figura 5.5: C/I na cidade depois do planejamento de todas as células.	56
Figura 5.6: Tempo de execução com o incremento de células a planear.	57
Figura 5.7: Quantidade de espaço ocupado com o incremento de células a planear	57
Figura 5.8: Comparação de desempenho dos algoritmos entre este ser executado em <i>cloud</i> e numa máquina física.	58

Lista de Tabelas

Tabela 2.1: Tabela de ARFCN's a atribuir dependendo da banda	9
Tabela 2.2: Exemplo da representação binária da latitude 37° usando o Geohash.	26
Tabela 2.3: Conversão de base decimal para Geohash base 32.	26
Tabela 3.1: Exemplo de solução devolvida pelo otimizador.	37
Tabela 4.1: Exemplo de tabela de interferência	44
Tabela 4.2: Testes unitários e previsões	47
Tabela 4.3: Resultados dos testes unitários.	48
Tabela 4.4: Exemplo 1 de uma tabela de poluição	48
Tabela 4.5: Exemplo 2 de uma tabela de poluição	49
Tabela 4.6: Tabela de indicação de sites.	50
Tabela 4.7: Tabela de indicação de interferência caso células tenham a mesma frequência.	50
Tabela 4.8: Tabela de interferência para teste de aferição 2	50

Abreviaturas

8-PSK	Phase Shift Keying
API	Application Programming Systems
AWS	Amazon Web Services
ARFCN	Absolute Radio-Frequency Channel Number
BCCH	Broadcast Control Channel
CCH	Control Channel
CLP	COIN-OR Linear Programming kit
DIMACS	Center for Discrete Mathematics and Theoretical Computer Science
EC2	Elastic Computer Cloud
FaaS	Function as a Service
GLPK	Gnu Linear Programming Kit
GPRS	General Packet Radio Service
GSM	Global System for Mobile communications
IaaS	Interface as a Service
ITU	Institute Telecommunication Union
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
LAI	Location Area Identity
LTE	Long Term Evolution
MMS	Multimedia Messaging Service
NIST	National Institute Standard Tecnology
OSS	Operations Support System
PaaS	Platform as a Service
QoS	Quality of Service

Abreviaturas

SaaS	S oftware a s a S ervice
SMS	S hort M essage S ervice
SON	S elf O rganizing N etwork
SPM	S tandard P ropagation M odel
TCH	T raffic C hannel
UMTS	U niversal M obile T elecommunications S ystem

Capítulo 1

Introdução

1.1 Motivação

Em Setembro de 2019, o número de dispositivos moveis no mundo eram perto de oito mil milhões, com previsão para 8.9 biliões em 2025 (Ericsson, 2019). Como consequência, existe um aumento considerável de tráfego móvel, o que implica a adição de mais equipamento, incluindo antenas e células, de forma a obtermos uma qualidade de serviço satisfatório para todos os utilizadores. No entanto, a adição deste novo equipamento diminui a uniformidade da cobertura de rede. Isto normalmente origina problemas de interferência devido a coberturas com a mesma frequência que se intersejam. De forma a evitar este problema, um dos passos mais importantes no planeamento de uma rede é a atribuição de frequências. No entanto, devido à dimensão do problema, e a aspetos ligados à densidade urbana e topografia do terreno, este processo é manual e com um custo temporal elevado.

1.2 Enquadramento e Objetivos

Esta dissertação foi desenvolvido no âmbito do projeto Planeamento e Otimização de Redes Heterogéneas 5G num ambiente Cloud (OptiNet-5G), um projeto de investigação da empresa Multivision, financiado pelo Centro2020 e Portugal2020,

a decorrer entre outubro 2017 e setembro 2020. O objetivo é desenvolver um algoritmo de planeamento de frequências para uma rede celular, que deverá ser implementado recorrendo a recursos *cloud* e deverá ser integrado na ferramenta comercial Metric.

1.3 Abordagem metodológica e questões de investigação

O método de investigação usado envolve o uso de cenários fictícios (ou seja, menos complexos) e reais. O uso de dados fictícios é importante, não só para ter a certeza que o sistema cumpre os seus requisitos, mas também para testar situações em que a entrada de dados inadequados possa inviabilizar o comportamento do sistema, isto inclui dados com formatos não válidos, dados fora do intervalo teórico/realístico, e até mesmo a ausência dos mesmos. O sistema será testado usando uma rede móvel colocada em Malanje, Angola. Os dados serão retirados do software Metric, enviados como entrada para o sistema que devolverá um planeamento. Este planeamento será comparado com o planeamento neste momento implementado, calculando a quantidade de interferência na cidade e arredores.

As questões de investigação que se propõem responder, com este trabalho, são as seguintes:

1. Como é que o planeamento de frequências beneficia de sistemas *cloud*?
2. Será possível o planeamento de várias células em simultâneo?
3. Quais os limites de uma rede em que o algoritmo de atribuição de frequências original (mais complexo) deixa de ser viável?

1.4 Contribuições

O trabalho desenvolvido no âmbito desta dissertação, resultou numa série de contribuições. Foram escritas as seguintes publicações em duas revistas internacionais com fatores de impacto Q1 e Q2 e numa conferência internacional, com revisão por partes:

1. Godinho, A., Fernandes, D., Clemente, D., Soares, G., Sebastiao, P., Pina, P., & Ferreira, L. S. (2019). *Cloud-based Cellular Network Planning System: Proof-of-Concept Implementation for GSM in AWS. International Symposium on Wireless Personal Multimedia Communications, WPMC*, 2019-Nov. <https://doi.org/10.1109/WPMC48795.2019.9096082>;
2. Godinho, A., Fernandes, D., Soares, G., Pina, P., Sebastião, P., Correia, A., & Ferreira, L. S. (2020). *A novel way to automatically plan cellular networks supported by linear programming and cloud computing*. *Applied Sciences (Switzerland)*, 10(9), 3072. <https://doi.org/10.3390/app10093072> (factor de impacto Q2);
3. Pina, P. M., Godinho, A. F., Fernandes, D. F., Clemente, D. J., Sebastião, P., Soares, G. E., & Ferreira, L. S. (2020). *Automatic Coverage Based Neighbour Estimation System: A Cloud-Based Implementation*. *IEEE Access*, 8, 69671–69682. doi: 10.1109/ACCESS.2020.2986755 (factor de impacto Q1).

Em relação ao projeto OptiNet-5G, foram feitas contribuições para os seguintes entregáveis do projeto, que foram avaliados e aprovados por uma comissão independente:

1. Multivision, *Módulos de optimização técnica, económica e ambiental*, entregável D4.1 Projeto OptiNet-5G, P2020, Lisboa Portugal, Fev. 2019 (www.multivision.pt);

2. Multivision, *Funcionamento dos módulos de otimização técnica*, económica e ambiental, entregável D4.2 Projeto OptiNet-5G, P2020, Lisboa Portugal, Fev. 2019 (www.multivision.pt);
3. Multivision, *Testes e avaliação dos módulos*, entregável D6.1 Projeto OptiNet-5G, P2020, Lisboa Portugal, Abr. 2019 (www.multivision.pt);
4. Multivision, *Testes e avaliação intermédia da integração dos módulos*, entregável D6.2 Projeto OptiNet-5G, P2020, Lisboa Portugal, Abr. 2019 (www.multivision.pt);
5. Multivision, *Relatório de testes e avaliação de integração dos módulos*, entregável D6.3, Projeto OptiNet-5G, P2020, Lisboa Portugal, Ago. 2019 (www.multivision.pt).

Foi também desenvolvido um protótipo posteriormente integrado na ferramenta comercial Metric:

1. Protótipo para planeamento de recursos rádio de GSM, integrado na ferramenta Metric, Set. 2019.

1.5 Estrutura

A presente dissertação está dividida em seis capítulos que se sumarizam de seguida:

- O primeiro capítulo - **Introdução** - apresenta a estrutura da dissertação, incluindo enquadramento, objetivos, abordagem metodológica, questões de investigação e contributos;
- O segundo capítulo - **Estado da arte** - descreve o enquadramento teórico para o projeto;
- No capítulo três - **Modelo e Algoritmos de Planeamento de Frequências** - definem-se os requisitos funcionais/não funcionais, bem como o modelo, módulos e algoritmos desenvolvido;

- O capítulo quatro - **Implementação** - apresenta as ferramentas e tecnologias usadas, e descreve a implementação, recorrendo a serviços *cloud*, e a sua integração com o Metric, assim como testes de aferição dos módulos e algoritmos desenvolvidos;
- No capítulo cinco - **Resultados** - é apresentado um cenário de referência para o qual é avaliado o desempenho do algoritmo proposto. É também realizado uma análise comparativa com outro algoritmo comercial;
- Finalmente, no capítulo seis estão as principais conclusões, limitações do estudo e propostas de investigação.

Capítulo 2

Estado de Arte

Neste capítulo são referenciados e esclarecidos os temas análogos a este trabalho, sendo detalhado o enquadramento teórico para o projeto. O capítulo começa por explicar na Secção 2.1, o conceito de sistemas de comunicação móvel e o protocolo Global System for Mobile communications (GSM). Nas Secções 2.2 e 2.3, é esclarecido conceitos no domínio da gestão de redes móveis e no domínio do planeamento sistemas de comunicação móvel celular, respectivamente. Por fim, na Secção 2.4, são expostos vários algoritmos usados como apoio ao planeamento.

2.1 Sistemas de Comunicação Móvel Celular para o GSM

Os sistemas de comunicações móveis permitem a comunicação entre vários dispositivos móveis, como por exemplo, telemóveis, *tablets*, computadores portáteis, usando uma rede de comunicação de grande alcance. Esta permite, dependendo da geração, a utilização de serviços como voz, SMS, Internet, jogos *online* e outros.

A rede de comunicações moveis é composta por vários componentes que comunicam entre si. Estes componentes dependem da geração de sistema usada, mas em todas as gerações é possível dividi-los em dois tipos de redes:

- **Core Network** - Camada da rede que fornece vários serviços aos utilizadores;
- **Radio Access Network** - Camada de ligação entre *Users equipment* e a *Core Network*, usando tecnologia de acesso rádio.

A rede telefónica celular é agregada por várias Estações Rádio Base (ERB). Estas facultarão a cobertura de rede necessária para comunicar com os equipamentos dos utilizadores. Estes sistemas celulares tiveram uma grande evolução, existindo várias gerações. Substituindo o 1G, o 2G foi lançado com a tecnologia *Global System for Mobile communications* (GSM). Esta melhoria, não só permitiu o uso de serviços como o SMS e MMS, como as chamadas telefónicas passaram a ser digitalmente encriptadas. Isto foi possível devido ao sinal de rádio passar a ser digital em vez de analógico, como era na primeira geração (Huurdemán, 2003). Mais tarde, foram usadas outras tecnologias no 2G, como por exemplo, o *General Packet Radio Service* (GPRS) que permitiu a comunicação a partir de pacotes, e o EDGE, que introduziu o *Octal Phase Shift Keying* (8-PSK), aumentando o débito binário e reduzindo a latência.

Para a implementação do GSM, são usadas várias bandas de frequência, que vão desde os 380 MHz aos 1900 MHz. No entanto, dependendo da localização, só é usado um par de frequências. Em África, Ásia, Europa e Meio-Oriente é normalmente usado tanto o GSM-900 como o GSM-1800, enquanto que na maior parte da América teremos o GSM-850 e o GSM-1900 (Worldtimezone.com, 2016). Como consequência, terão de ser usados móveis *multi-band* de forma a evitar problemas de compatibilidade de frequências e facilitar o Roaming. Por exemplo, ao usarem-se telemóveis *tri-band* com cobertura sobre o GSM-900, GSM-1800 e GSM-1900, estaremos a obter uma boa cobertura na Europa e uma cobertura suficiente na América. Também existem os chamados World-Phones ou Quad-band Phone, que usam as quatro frequências. Existem exceções para esta regra. Países como o Brasil e o Uruguai, usam as 4 bandas de frequência. Existem também países que nunca implementam o 2G (Coreia do Sul e Japão) e outros que implementaram, mas, entretanto, desligaram (Austrália e Singapura).

Cada banda tem um conjunto de *Absolute Radio-Frequency Channel Numbers* (ARFCN), que especifica cada um par de frequências usado, no *uplink* como no *downlink*. A fórmula usada para especificar um ARFCN depende da banda a ser utilizada. Isto é demonstrado na Tabela 2.1, onde f_{UL} e f_{DL} é a frequência do *uplink* e *downlink*, respetivamente.

TABELA 2.1: Tabela de ARFCN's a atribuir dependendo da banda

Band	Designation	ARFCN	f_{UL}	f_{DL}
GSM 500	GSM 450	259-293	$450.6 + 0.2(n-259)$	$f_{UL}(n) + 10$
	GSM 480	306-340	$479.0 + 0.2(n-306)$	$f_{UL}(n) + 10$
GSM 700	GSM 750	438-511	$747.2 + 0.2(n-438)$	$f_{UL}(n) + 30$
GSM 850	GSM 850	128-251	$824.2 + 0.2(n-128)$	$f_{UL}(n) + 45$
GSM 900	P-GSM	1-124	$890.0 + 0.2n$	$f_{UL}(n) + 45$
	E-GSM	0-124	$890.0 + 0.2n$	$f_{UL}(n) + 45$
		975-1023	$890.0 + 0.2(n-1024)$	
	GSM-R	0-124	$890.0 + 0.2n$	$f_{UL}(n) + 45$
955-1023		$890.0 + 0.2(n-1024)$		
GSM 1800	DCS 1800	512-885	$1710.2 + 0.2(n-512)$	$f_{UL}(n) + 95$
GSM 1900	PCS 1900	512-810	$1850.2 + 0.2(n-512)$	$f_{UL}(n) + 80$

O GSM é composto por vários canais lógicos, divididos em canais de tráfego (TCH), que carregam uma chamada telefónica codificada ou então dados de utilizador, e canais de controlo (CCH). O *Broadcast Control Channel* (BCCH), é uma das componentes dos CCHs, e este é responsável pela transmissão, em *downlink*, de informações como a *Location Area Identity* (LAI), a lista de frequências usada na célula, a identidade da própria célula, o controlo de acesso, etc.

2.2 Gestão de Redes Móveis Celulares

2.2.1 Planeamento, monitorização e otimização

Geralmente, a criação e gestão das Redes de Comunicações Moveis requerem várias fases:

1. **Planeamento** - Existe um conjunto de requisitos a ter em conta na implementação de uma rede móvel. Como por exemplo, verificar se a cobertura é

suficiente, garantir QoS apropriada e minimizar a probabilidade de erro de bit. Isto com o objetivo de minimizar a infraestrutura e os custos da mesma. Um dos atributos mais importantes a ser planeado são as frequências;

2. **Monitorização** - Monitorizar a rede usando *Key Performance Indicators* (KPIs) de forma a obter uma visão geral do estado da rede;
3. **Optimização** - As KPIs retiradas da monitorização deverão ser analisadas de forma a encontrar soluções para melhorar/reparar a rede. Nestes casos é usual aplicar alterações como:
 - Alterar configurações da estação base;
 - Mudar a direção e intensidade da antena;
 - Colocar as antenas mais altas/baixas;
 - Alterar a localização da antena;
 - Alterar a localização das estações-base;
 - Modificar parâmetros rádio (por exemplo, a potência de emissão ou o ganho de emissão).

É importante ter em conta que os KPI's devem ser bem analisados de forma que se perceba realmente o que é que eles representam.

2.2.2 Redes Auto-Organizadas

O problema de gestão de redes móveis é que esse processo normalmente é manual, dispendioso e demorado. Uma solução criada para este problema é o conceito de *Self Organized Networks* (SONs) (Ramiro & Hamied, 2012), que automatizam este esforço em três áreas, ilustradas na Figura 2.1:

- *Self-Configuration* - Na criação de uma nova estação-base, os seus vizinhos automaticamente trocarão informações entre si, ajustando automaticamente os seus parâmetros técnicos;

- *Self-Optimization* - Permite a alteração de parâmetros dependendo da necessidade. Um exemplo seria: Durante a noite, uma estação-base se desligar, e as vizinhas alterarem ângulos e aumentarem intensidades de forma a recobrir a área. Ou então, no caso de um crescimento súbito de tráfego, estas estações "acordavam";
- *Self-Healing* - Permite a redução do impacto em caso de falhas ou a execução de certas ações como atualizações de *software/hardware*.

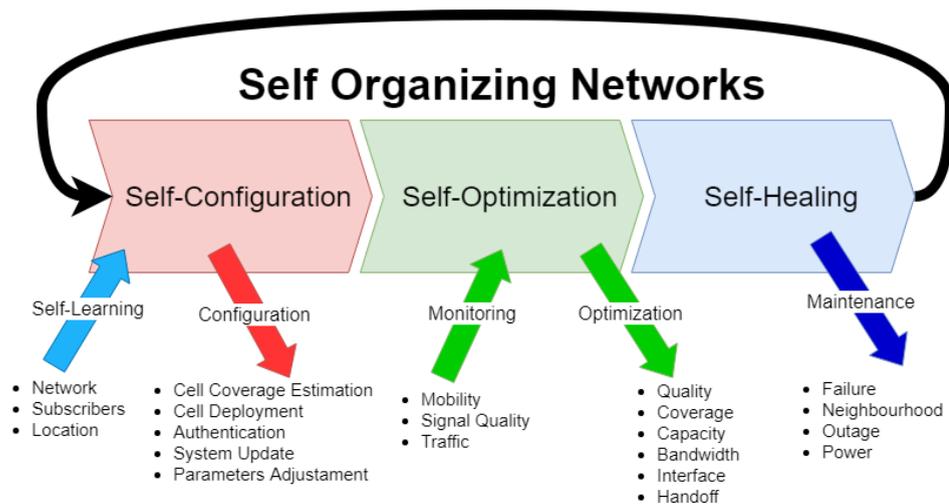


FIGURA 2.1: Ciclo de vida das SON.

2.2.3 Serviços Cloud para Gestão e Planeamento da Rede

Os serviços *cloud* permitem que utilizadores se conectem e usem serviços em locais remotos ao de onde o serviço está alojado. Avram (2014) explica várias vantagens no uso de sistemas *cloud*, tanto para os utilizadores como empresas. Por exemplo, os utilizadores podem executar e usar aplicações, artefactos ou algoritmos, sem a necessidade de *hardware/software*, especializado. Outro exemplo seria o facto de como os utilizadores podem aceder a estes dados independentemente da sua localização, desde que estes tenham acesso a rede, todos os dados serão iguais independentemente do dispositivo usado. Por fim, as aplicações *cloud* são usualmente “*pay per use*”. Normalmente, o valor pago pela gestão e manutenção do

hardware é maior que o do custo de uso destes serviços, com a vantagem de que se evitam despesas em recursos não usados.

Um exemplo de provedor de serviços *cloud* é a “Amazon Web Services”, que contém serviços e aplicações em áreas como as base de dados, computação, *machine learning*, *gaming*, inteligência artificial, *internet of things*, máquinas virtuais, e outras (Amazon, 2019). O Instituto Nacional de Padrões e Tecnologia (NIST), define três tipos de serviço padrão (Mell & Grance, 2011), representados também na Figura 2.2:

- **Infraestrutura como serviço (IaaS)** - “Onde o consumidor pode inserir e correr *software* arbitrário, que pode incluir sistemas operativos e aplicações. O consumidor não gerência ou controla a infraestrutura *cloud*, mas tem controlo sobre os sistemas operativos, armazenamento e aplicações inseridas; e possivelmente controlo limitado de algumas componentes de rede (Ex. Firewalls).”;
- **Plataforma como serviço (PaaS)** - “Dar a capacidade ao consumidor de inserir na infraestrutura *cloud*, aplicações adquiridas ou criadas pelo consumidor, usando linguagem de programação, livrarias, serviços, e ferramentas suportadas pelo provedor. O consumidor não gerência ou controla a infraestrutura *cloud*, incluindo a rede, servidores, sistemas operativos e armazenamento, mas tem controlo sobre as aplicações inseridas e possivelmente a um conjunto de definições de configuração para o ambiente de hospedagem de aplicações.”;
- **Software como serviço (SaaS)** - “Dar a capacidade ao consumidor de usar aplicações do provedor, executadas numa infraestrutura *cloud*. As aplicações são acessíveis a partir de vários dispositivos do cliente, seja a partir de uma *interface* “*thin cliente*”, como um navegador de *web* (Ex. e-mail baseado em *web*), ou uma interface de um programa. O consumidor não gerência ou controla a infraestrutura *cloud* incluindo a rede, servidores, sistemas operativos, armazenamento, ou até mesmo a capacidade de aplicações individuais, com a possível exceção de definições de configuração específicas para o utilizador.”.

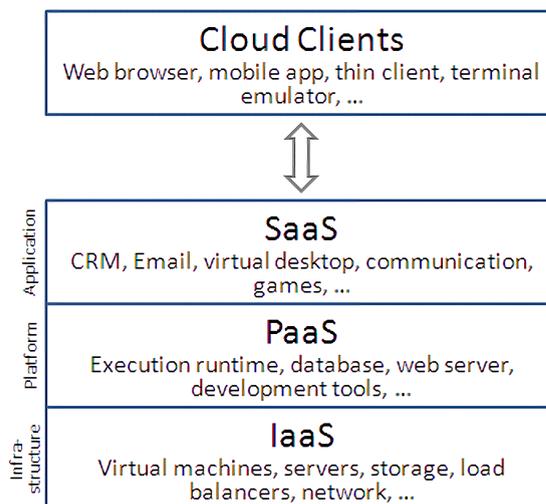


FIGURA 2.2: Estrutura de serviços *cloud* num modelo de negócio.

Outro tipo de serviço é referenciado por Roberts (2016), chamado “Função como serviço (FaaS)”. FaaS cria a capacidade de inserir funções em ambiente *cloud*. Estas apenas serão executadas depois de um “gatilho” ou evento, tal como acontece com os AWS lambdas (AWS, 2018). Esta alternativa é preferível para executar código cujo objetivo não seja correr continuamente ou executá-la múltiplas vezes num curto espaço de tempo.

2.2.4 Metric

O Metric (Multivision, 2019) é uma ferramenta de planeamento e gestão de redes, faz parte da categoria SaaS e foi criada e comercializada pela Multivision. Permite obter e organizar informação de diferentes fontes e gerações, de forma a haver uma maior rapidez e fluidez na análise de dados, reduzindo assim os custos para os seus clientes. Esta ferramenta, além de permitir a deteção de certas falhas na rede usando algoritmos, também permite a visualização e importação de dados como a força de sinal, a interferência, frequência das células, entre outros. O *software* já tem integração para as SON nas componentes de monitorização e manutenção, estando a desenvolver ferramentas para componentes da configuração e otimização. O Metric integra um algoritmo de planeamento de frequências. A

presente dissertação propõe um algoritmo alternativo mais eficiente. Um exemplo da interface deste sistema pode ser visualizada na Figura 2.3.



FIGURA 2.3: Interface do mapa do Metric SaaS.

2.3 Planeamento de Sistemas de Comunicação Móveis Celulares

2.3.1 Planeamento Celular

Devido ao curto espectro de frequências, e ao cada vez maior número de dispositivos móveis ativos, é de extrema importância que haja um planeamento celular de uma área, antes da implementação da mesma, de forma a aumentar a capacidade da rede e evitar interferências que, como consequência, diminui os custos e aumenta o QoS (Assunção, Estevinho, & Correia, 2001).

Idealmente, a cobertura deve ser composta por células iguais e não deve haver nem sobreposições nem vazios. Havendo só três polígonos regulares que são soluções para este problema (triângulo, quadrado e hexágono), foi escolhido o hexágono, por este se aproximar mais da cobertura teórica de uma antena omnidirecional, tal como ilustrado na Figura 2.4. No entanto, é preciso ter em conta que esta cobertura é apenas teórica. Apesar desta ser importante para, por exemplo,

localizar as estações base no centro dos hexágonos, na realidade, a mesma, tem uma forma irregular, devido a aspetos como a topografia do terreno, densidade urbana, etc, resultando bastante diferente da cobertura teórica, tal como se ilustra na Figura 2.4.

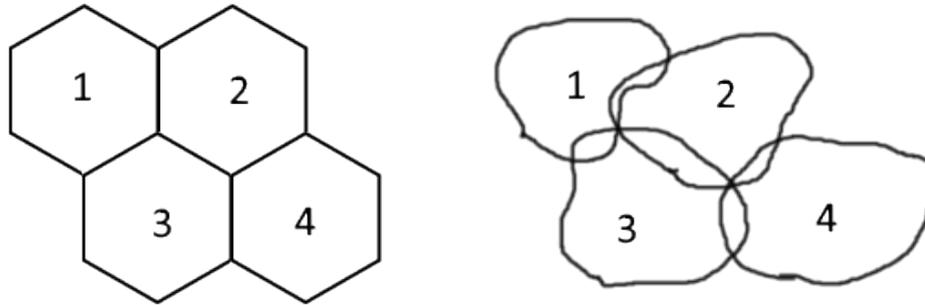


FIGURA 2.4: Cobertura celular teórica e realista.

É possível estimar a cobertura de uma célula usando modelos de propagação. Existem modelos empíricos e modelos determinísticos. O primeiro é baseado em medidas e observações enquanto que o outro faz uso das leis de propagação de ondas eletromagnéticas. Os determinísticos têm a desvantagem de precisar de informação específica do ambiente em que as células se encontram, incluindo a topografia do terreno. Exemplos de modelos empíricos usados, como o Okumura-hata e Walfish-Ikegami (Mishra, 2007). Existe outro modelo empírico de propagação denominado de *Standard propagation model* (SPM) baseado no Okumura-hata (Popoola et al., 2017), que é usado em ferramentas de planeamento e otimização como a ATOLL (Forsk, 1997).

Fernandes et al. (2019) criaram um modelo de propagação calibrado a partir do SPM usando *drive tests*. *Drive tests* são testes de desempenho à rede feitos por operadores, normalmente em veículos terrestres, de onde se retiram vários KPIs junto com coordenadas geográficas. Estes testes, além de serem feito entre intervalos de tempo definidos pela operadora, são normalmente realizados depois de alterações nas configurações das células para verificar se estes tiveram o efeito desejado. Este modelo permite realizar uma estimativa realística da cobertura, sendo o mesmo usado no presente trabalho.

Existem vários tipos de células:

- **Macro** - A maior de todas as células, permite uma cobertura entre 1 e 30 quilómetros. As antenas devem ser colocadas em localizações altas, como os telhados de edifícios, de forma a que se tenha uma visão desimpedida;
- **Micro** - Normalmente colocada em locais de elevada densidade urbana, tem mais intensidade que a Macro, mas a sua cobertura só chega a 2 km;
- **Pico** - Em locais em que o sinal de rede não é suficiente, como o interior de edifícios, e áreas em que possam existir uma grande quantidade de tráfego, como estádios e centros comerciais;
- **Femto** - Criadas para funcionar em ambientes domésticos e de pequenas empresas, permite uma cobertura de 10 metros, em localizações fechadas e pessoais.

De forma a que não exista interferência na comunicação com os vários dispositivos, cada célula terá uma frequência diferente das adjacentes. A cada conjunto da mesma frequências, chamamos de *cluster*.

Outro problema a ter em conta, em que a distribuição do tráfego também não é regular. Serão necessárias células de tamanhos diferentes dependendo da densidade do tráfego, ou seja, um maior número de células mais pequenas para zonas de muito tráfego e vice-versa. Por isto, é preciso ter atenção, tanto ao possível aumento de interferência, visto que a cobertura deixou de ser uniforme, como na localização das células nas zonas de transição de frequência.

Uma das abordagens é o recurso à setorização de uma célula, quando uma célula não tem a capacidade suficiente para o tráfego oferecido na área circundante. Freeman (2005) também explica modelos teóricos como a setorização e o *frequency hop*. A setorização, como referido anteriormente, é uma das soluções encontradas para atenuar a interferência co-canal, dividindo a célula em setores diferentes. Tipicamente, é usado os ângulos de 60° e 120°, dividindo a célula em seis e três

setores respetivamente. No entanto, simulações feitas por Assunção et al. (2001), mostram que é possível haver melhores resultados com divisões a 90° . A Figura 2.5, ilustra um dos problemas com a setorização em que seja qual for o ângulo de divisão, a radiação não está limitada a esse ângulo, havendo um *overlap* entre setores e originando interferência adjacente.

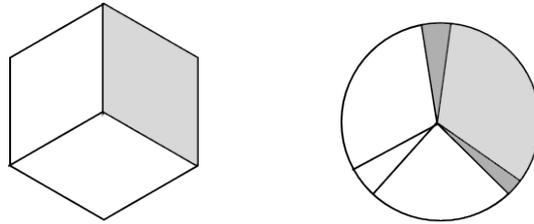


FIGURA 2.5: Esquerda: Setorização teórica em 3 partes. Direita: Setorização real com *Overlap*

2.3.2 Planeamento de frequências

Existem várias maneiras de atribuir frequências, mas as duas mais usadas são a atribuição de frequências fixa, onde se dá prioridade ao respeito máximo das condições de planeamento, e a atribuição de mínima amplitude, onde se dá prioridade ao uso do menor número possível de frequências, respeitando ainda as condições. Existem também outras soluções. Em 2010, Su, Lan, Yu, Gou, e Zhang (2010) propuseram usar as coordenadas geográficas de uma célula e o seu azimute e compara-la com as suas vizinhas, prevendo o risco das células interferirem umas com as outras. No entanto, este processo requer intervenção humana.

É necessário considerar 3 tipos de interferência (Freeman, 2005):

- **Interferência Co-canal** - Interferência criada por outras células com a mesma frequência. Normalmente evitada distanciando células com a mesma frequência e dividindo células em sectores (Sectorização);

- **Interferência Adjacente** - Interferência criada por células adjacentes com ARFCN's adjacentes. Facilmente resolvível usando filtros de alta qualidade e diferenciando minimamente as frequências;
- **Interferência Própria** - Consequência da reflexão do sinal de uma célula em obstáculos, como edifícios, montanhas, etc. Resolvido dividindo as frequências de *downlink* e *uplink*.

Este trabalho foca-se na interferência co-canal e canal adjacente.

O *frequency hop* é um método de transmissão onde a frequência de transmissão é alterada várias vezes durante uma conexão, “saltando” de uma frequência para outra. A ordem destes saltos é decidida por um código conhecido pelo transmissor e pelo recetor. Isto permite evitar interferência, evitar que uma conexão seja ouvida por terceiros, e a aplicação de *Multiple reuse patterns*.

Multiple reuse patterns é um método que, usando o *frequency hop*, se prioriza o reúso das frequências com o objetivo de aumentar a sua capacidade (Engstrom et al., 1998). Isto é aplicado ao GSM, atribuindo múltiplas frequências aos canais de tráfego de uma célula.

Para o planeamento ótimo de frequências do BCCH, é necessário ter em conta:

- Design local;
- Topografia;
- Distribuição dos subscritores;
- Regularidade no planeamento da célula.

Para o planeamento do TCH, é necessário ter em conta as mesmas considerações que no BCCH, mas neste caso é possível usar o *Frequency Hopping* e o *Multiple Re-use Patterns*. Isto significa que durante um planeamento, deverão ser atribuídas múltiplas frequências aos TCH's.

2.3.3 Algoritmo de Planejamento de Frequências Metric

O algoritmo de planejamento de frequências Metric (Multivision, 2019) é um algoritmo desenvolvido pela Multivision e integrado na ferramenta Metric. Este algoritmo é usado como referência comparativa ao proposto na presente dissertação, e tem como objetivo fazer o planejamento de frequências 2G usando, como base, a localização geográfica e os azimutes das células, e o modelo de propagação Okumura-Hata, de forma a calcular-se a matriz de interferência. O *workflow* do algoritmo está representado na Figura 2.6.

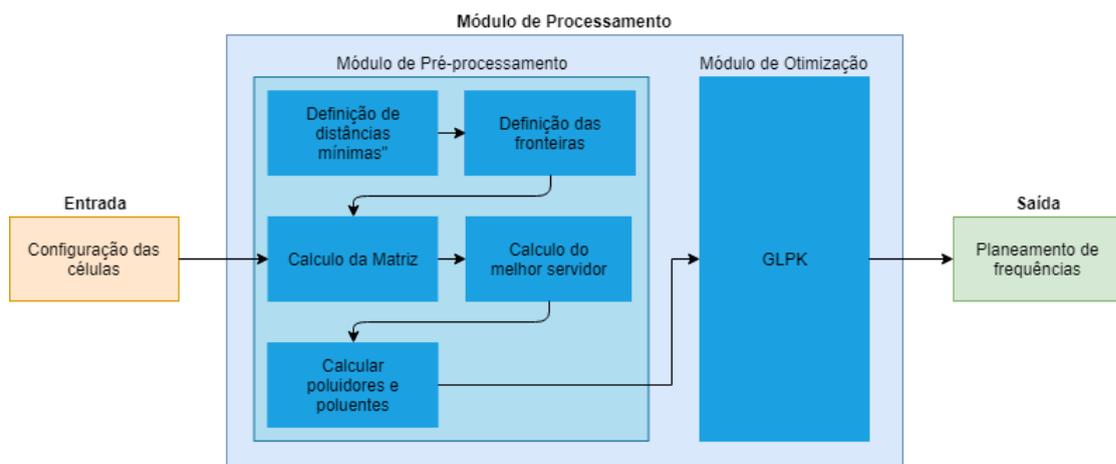


FIGURA 2.6: *Workflow* do algoritmo de planejamento de frequências Metric.

O algoritmo planeia apenas as células cujo campo BCCH esteja vazio e que o estado seja "PLANNED", recebendo como *inputs*:

- Ficheiro de Células 2G;
- Frequências: Bandas BCCH e TCHs do cliente.

Estes valores são depois usados no módulo de processamento, constituído por:

1. **Definição de distância mínima** - Descobrir a célula mais próxima na direcção do azimute;
2. **Definição das fronteiras** - Definição de uma fronteira para limitar os cálculos;

3. **Calculo da matriz** - Calculo de uma matriz de pontos bin que representam a potência usando o modelo *Okumura-Hata*;
4. **Cálculo do melhor servidor** - Descobrir qual é a célula com maior valor de sinal;
5. **Calcular poluidores e poluentes** - Calcular o número de poluidores/poluentes por célula.

O algoritmo depois gera um ficheiro SQL (execute.sql) que irá atualizar a tabela `cells_2g`, nos campos `BCCH`, `TCH1`, `TCH2` e `TCH3`. Existem condições no *output* deste algoritmo:

- Se existirem células sem coordenadas, estas não são consideradas;
- Se existirem células sem o campo 'BAND' preenchido, estas não são consideradas;
- Se o campo 'site_type' for inserido no ficheiro de células, só são consideradas as células com 'site_type = OUTDOOR';
- São planeadas sempre quatro frequências por célula: 1 BCCH e 3 TCH.

O ângulo de filtragem indica se a célula tem visibilidade para a célula vizinha. Tal como ilustrado na Figura 2.7, se "filter_angle" for igual a 1, então esta está visível a 60°.

A *Min Distance* é a distância da célula mais próxima, que não seja *sites* com a mesma 'btslocation/localização', e que tenha, obrigatoriamente, 'filter_angle=1'. O valor por defeito é 35 km.

De forma a limitar os cálculos, é definido um quadrado em torno do *site*, como é visível na Figura 2.8, onde $r = MinDistance \times FactorRadius$. Dentro desta área serão feitas análises de interferência. O valor máximo é de 35 km.

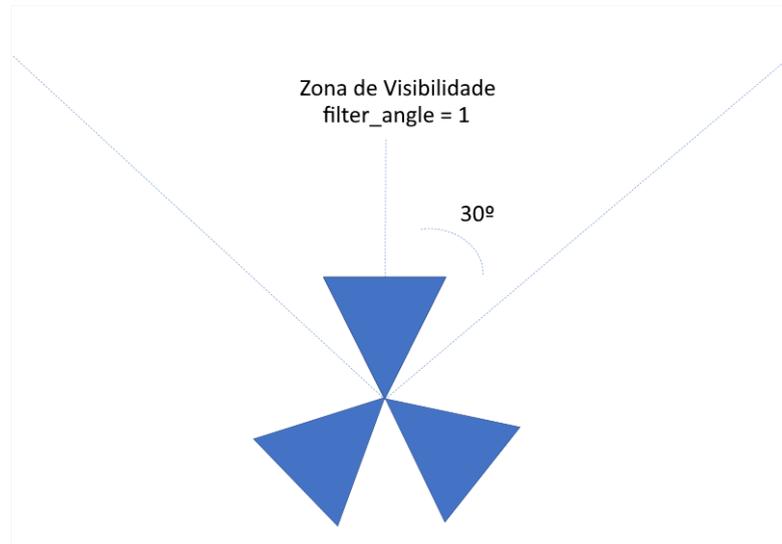


FIGURA 2.7: Zona de visibilidade.

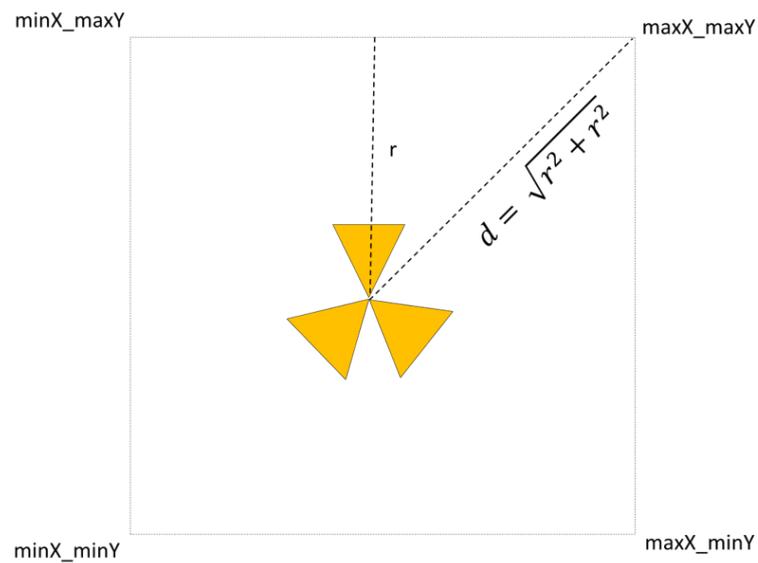


FIGURA 2.8: Área definida pelo 'Boundaries'.

Modelo de propagação, aplicado às *micro cells*, escolhido para prever a perda de sinal em ambientes exteriores com frequências entre 150 e 1500 MHz. Essa perda é calculada pela equação 2.1.

$$L[\text{dB}] = A + B \log(d) + C \quad (2.1)$$

onde:

$$A = 69.55 + 26.16 \log(f) - 13.82 \log(h_{BTS}) - a(h_{MS}) \quad (2.2)$$

$$B = 44.9 + 6.55 \log(h_{BTS}) \quad (2.3)$$

em que:

- d é a distancia em km;
- f é a frequência em MHz;
- h_{BTS} e h_{MS} é dada em m .

e, C e $a(h_{MS})$ são constantes dependentes do ambiente em estudo para:

- Pequenas e médias localidades;

$$a(h_{MS}) = [1.1 \log(f) - 0.7]h_{MS} - [1.56 \log(f) - 0.8] \quad (2.4)$$

$$C = 0 \quad (2.5)$$

- Áreas metropolitanas;

$$f(n) = \begin{cases} 8.29[\log(1.54h_{MS})^2] - 1.1, & f \leq 200 \text{ MHz} \\ 3.2[\log(11.75h_{MS})^2] - 4.97, & f \geq 400 \text{ MHz} \end{cases}$$

$$C = 0 \quad (2.6)$$

- Ambientes Suburbanos;

$a(h_{MS})$ tal como definido na Equação 2.4;

$$C = -2\left[\log\left(\frac{f}{28}\right)\right]^2 - 5.4 \quad (2.7)$$

- Ambientes Rurais. $a(h_{MS})$ tal como definido na Equação 2.4;

$$C = -4.78[\log(f)]^2 + 18.33 \log f - 40.98 \quad (2.8)$$

No entanto, o GSM pode ter frequências na ordem dos 1800 MHz, impedindo o uso deste modelo. Então, é importante aumentar o intervalo de frequências

em que este modelo pode ser utilizado. Assim surge o COST 231 - Hata model, redefinindo as variáveis anteriormente faladas para:

$$A = 46.3 + 33.9 \log(f) - 13.82 \log(h_{BTS}) - a(h_{MS}) \quad (2.9)$$

$$B = 44.9 + 6.55 \log(h_{BTS}) \quad (2.10)$$

$a(h_{MS})$ é igual à Expressão 2.4 e o C é 0 nas pequenas e médias localidades e 3 nas Áreas metropolitanas.

De seguida, é calculada uma matriz de pontos *bin*. Um *bin* é uma estrutura de dados usados no contexto da geometria computacional. Estes são calculados para a própria célula e para as células interseçadas pela *boundarie*, quatro casas decimais (cerca de 10 metros), com uma abertura de 120° e uma distância limite igual a $Min_distance * factor_radius$. De forma a manter o tamanho das divisões reduzido, são feitas divisões até 300m a cada 2.4° , e após 300m, a cada 0.6° . Na Figura 2.9 está uma representação da divisão.

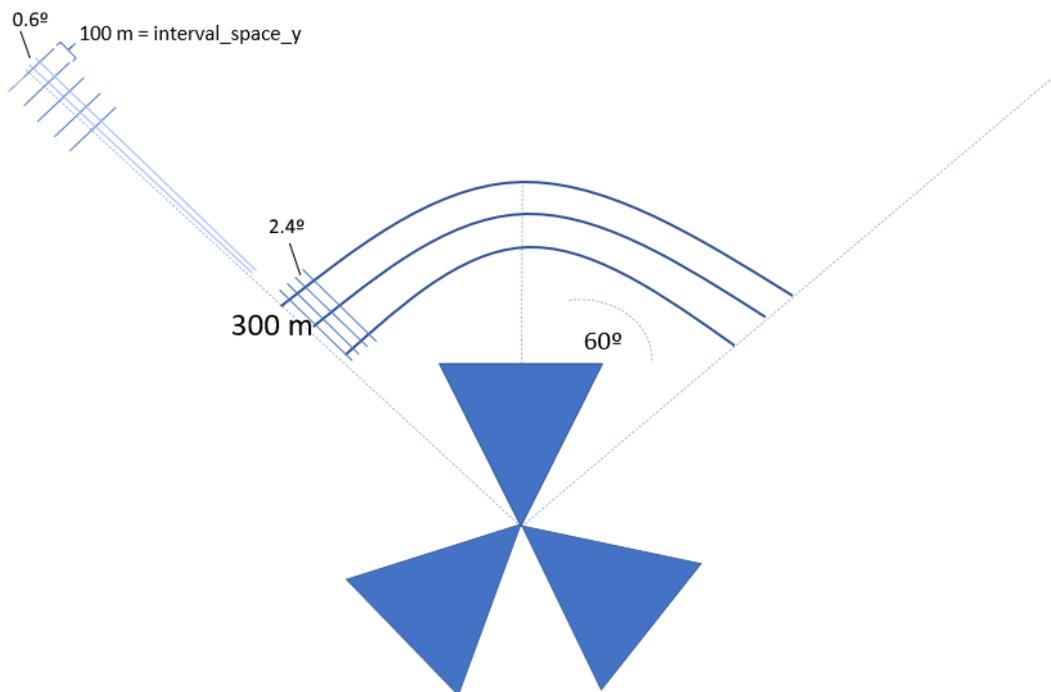


FIGURA 2.9: Demonstração da divisão das áreas.

Para cada pixel, usando as suas Matrizes, descobre-se qual a célula com melhor valor de sinal (*Best value*). Esta célula será o *Best Server*. Para cada pixel será contabilizada a interferência com o *Best Server*, que será registada e enviada para um otimizador linear. No entanto, as regras enviadas para o otimizador são simples e só permitem o planeamento de células que estejam no mesmo *site*. Além disso, a divisão feita na matriz de pontos *bin* não só não é uniforme, como é realizada para todas as células que tenham algum impacto sobre as que estamos a planear, o que significa um custo computacional elevado.

2.4 Algoritmos de Apoio ao Planeamento

Nesta secção são apresentados alguns algoritmos aplicados no desenvolvimento do algoritmo proposto na presente dissertação.

2.4.1 Z-order

O *Z-order curve* é um algoritmo que permite a conversão de dados multidimensionais em unidimensionais sem perder os dados originais. Foi criado e desenvolvido por Morton (1966), para o sequenciamento de ficheiros. O algoritmo funciona, interpolando representações binárias das coordenadas, como exemplificado na Figura 2.10. Existem múltiplas aplicações deste algoritmo, incluindo álgebra linear (Gilbert, Leiserson, & Street, 2009; Valsalam & Skjellum, 2002), mapeamento de texturas, codificação de imagem e vídeo, e geração de *quadtrees* (Bern, Eppslein, & Teng, 1993).

2.4.2 Geohash

O Geohash é uma das aplicações do *Z-order curve*, criado e desenvolvido por Niemeyer (2008). O Geohash transforma coordenadas geográficas numa sequência

	x: 0 000	1 001	2 010	3 011	4 100	5 101	6 110	7 111
y: 0 000	000000	000001	000100	000101	010000	010001	010100	010101
1 001	000010	000011	000110	000111	010010	010011	010110	010111
2 010	001000	001001	001100	001101	011000	011001	011100	011101
3 011	001010	001011	001110	001111	011010	011011	011110	011111
4 100	100000	100001	100100	100101	110000	110001	110100	110101
5 101	100010	100011	100110	100111	110010	110011	110110	110111
6 110	101000	101001	101100	101101	111000	111001	111100	111101
7 111	101010	101011	101110	101111	111010	111011	111110	111111

FIGURA 2.10: O uso da *Z-order curve* para interpolar a representação binário dos inteiros de 0 a 7.

binária que é depois codificada usando uma Tabela Base32 específica. Praticamente, tal como ilustrado na Figura 2.11, teremos uma divisão do mundo em vários retângulos, cuja quantidade e tamanho depende da precisão escolhida.

0000	0001	0100	0101
0010	0011	0110	0111
1000	1001	1100	1101
1010	1011	1110	1111

FIGURA 2.11: Divisão do mundo usando o algoritmo Geohash.

O algoritmo começa por transformar as componentes das coordenadas em binário. Cada *bit* representa uma série de divisão por 2. Este *bit* vai ser 1, se esta estiver no maior intervalo, ou 0 se vice-versa. Por exemplo, a transformação da latitude 37.000 em *bits*, será "1011", tal como demonstrado na Tabela 2.2. Esta sequência binária será depois intervalada com a sequência binária da longitude e finalmente codificada usando a Tabela 2.3.

Existem dois problemas com o Geohash. Primeiro: usualmente é possível usar os prefixos de um Geohash para encontrar pontos em proximidade, no entanto, nas

TABELA 2.2: Exemplo da representação binária da latitude 37° usando o Geohash.

Posição do Bit	Valor do Bit	Min	Mid	Max
1	1	-90.000	0.000	90.000
2	0	0.000	45.000	90.000
3	1	0.000	22.500	45.000
4	1	22.500	33.750	45.000

TABELA 2.3: Conversão de base decimal para Geohash base 32.

Base10	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Base32	0	1	2	3	4	5	6	7	8	9	b	c	d	e	f	g
Base10	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Base32	h	j	k	m	n	p	q	r	s	t	u	v	w	x	y	z

fronteiras, estes prefixos podem não ser semelhantes. Por exemplo, os Geohashes “s0000” e “kpbpb” são adjacentes, no entanto, estes estão em diferentes metades do globo (um no norte do equador e outro a sul do equador, respetivamente); Segundo, a inexistência de linearidade na distância que um Geohash representa no nosso planeta, visto que um grau na latitude e longitude representam distâncias diferentes. No equador existe um erro de 0.67% que tende a ir para infinito quando nos aproximamos dos pólos. A causa do problema não é o algoritmo, mas sim a dificuldade de converter coordenadas de uma esfera num valor unidimensional (Broadbent, Gauss, Hildebeitel, & Morehead, 1967). Apesar destes problemas, o algoritmo continua a ser aplicado em outros contextos como identificadores únicos e representações de dados em bases de dados com o MongoDB (MongoDB, 2019).

2.4.3 Programação Linear

Apesar de existirem muitas ferramentas que resolvem problemas de programação linear, a maioria ou têm custos ou têm licença proprietária. No entanto, existem alguns “CopyLeft” *softwares* como o CLP, GLPK, Cassowary constraint solver, Lp_Solve e Qoca. Apesar do GLPK ter sido considerados um dos menos eficientes em vários estudos (Gearhart et al., 2013; Meindl & Templ, 2012), este tem a vantagem de já existir uma camada de implementação para o AWS Lambda (Tempusenergy, 2019). Para o GLPK existem vários formatos de entrada:

- **CPLEX-LP** – Usado num *software* comercial chamado “IBM ILOG CPLEX Optimization Studio”. Destaque para a sua facilidade de escrita e leitura;
- **MPS** – Chamado de *Mathematical Programming System*, é um formato usado, tanto em programação linear, como em problemas de programação inteira. Este está limitado a campos numéricos com 12 caracteres. No entanto, versões mais modernas sem limitações também são aceites pelo GLPK;
- **GLPK** – Formato personalizado, baseado no DIMACS (Center for Discrete Mathematics and Theoretical Computer Science), criado pelo centro de matemática discreta e ciência da computação;
- **MathProg** – Linguagem de alto nível a que o GLPK tem dificuldades quando usado em modelos grandes, fazendo com que este seja demorado. A causa deve-se ao tradutor do GLPK não ser capaz de reconhecer atalhos na otimização. No entanto, não só é possível seguir certas heurísticas para reduzir o tempo de otimização, como também é possível converter o formato *MathProg* noutra formato (existem casos onde esta opção acaba por reduzir o tempo de execução).

Capítulo 3

Modelo e Algoritmos de Planeamento de Frequências

O Capítulo 3 apresenta um modelo para planeamento de frequências e algoritmos associados. Na Secção 3.1 apresenta os requisitos do modelo a desenvolver. Na Secção 3.2 é apresentada uma visão geral do modelo, sendo detalhado as três componentes propostas desde a Secção 3.3 à 3.5.

3.1 Requisitos

Para o desenvolvimento do modelo e sistema, são identificados um conjunto de requisitos a ter em conta. Requisitos funcionais (RFs):

- **RF1:** O algoritmo deverá ser capaz de planear conjuntos de frequências para células do mesmo *site*;
- **RF2:** O algoritmo deverá ser capaz de planear conjuntos de frequências para células de diferentes *sites*;
- **RF3:** O algoritmo deverá respeitar as regras de planeamento que evitam interferências de canal adjacente;

- **RF4:** O *software* deverá ser integrado com serviço AWS;
- **RF5:** O *software* deverá ser integrado com serviço Metric SaaS da Multivision.

Requisitos não-funcionais (RNFs):

- Consistência:
 - **RNF1:** *Input* dos mesmos dados em iterações diferentes deverá devolver o mesmo *output*;
 - **RNF2:** O algoritmo corrido em diferentes máquinas deverá devolver os mesmos resultados.
- Disponibilidade:
 - **RNF3:** O sistema deve estar disponível 24 horas por dia, 7 dias por semana.

3.2 Visão Geral do Modelo

Este modelo tem o objetivo de fazer o planeamento de frequência dos BCCH e TCH do sistema GSM, tal como se ilustra na Figura 3.1. A ideia geral é estimar a potência nas zonas cobertas pelas células a planear e obter o conjunto de frequências a atribuir que origina a menor quantidade de interferência possível. O modelo recebe como entrada, um pedido de planeamento de frequências e uma lista de configuração das células.

- **Pedido de planeamento de frequências** - Ficheiro com a informação de quais as células a planear.
- **Lista de configurações de cada célula** - Tabela com as características de todas as células da rede, incluindo o *site* a que pertencem, frequências, coordenadas geográficas e azimutes.

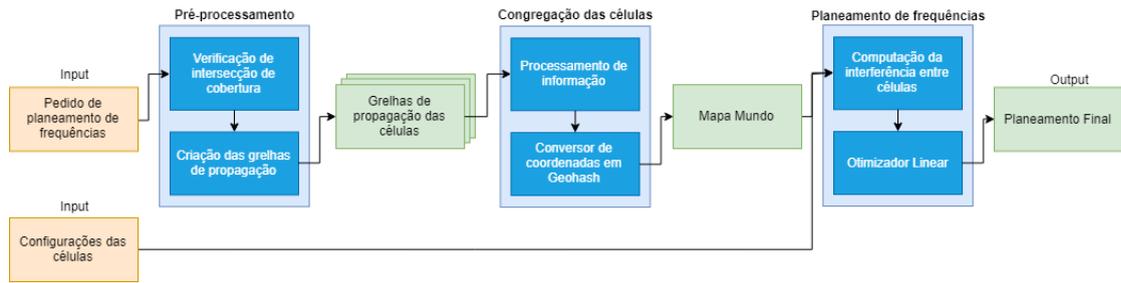


FIGURA 3.1: Modelo para realizar o planeamento de frequências.

O modelo divide-se em três módulos:

- **Pré-processamento** - Geração das grelhas de propagação usando modelos de propagação, para cada célula. Cada grelha terá uma coordenada geográfica de referência associada. As grelhas geradas serão as que foram pedidas para planear, e aquelas cuja cobertura intersesta com as que estamos a planear;
- **Congregação das células** - Processamento das grelhas de propagação das células e conversão das coordenadas em Geohash, agrupando-as para a criação do mapa mundo. Como resultado, tem como *output* um "Mapa Mundo", ficheiro que contém estimativas de potências de várias células na área coberta pelas mesmas (identificadas pelo Geohash);
- **Planeamento de frequências** - Módulo que realiza o planeamento de frequências, que consiste na computação da interferência entre células e posterior atribuição das frequências que optimizam o problema. Como resultado, apresenta um planeamento de frequências para as células escolhidas.

3.3 Módulo de Pré-processamento

Uma vez recebido o pedido de planeamento, o módulo começa por identificar as células por planear, seguido das células que partilhem cobertura com as mesmas, visto que as últimas podem interferir com as primeiras e vice-versa. O procedimento é simples, apenas assumindo de forma pessimista que as coberturas são

quadrados (toda a cobertura real está dentro deste quadrado) e verificando se as pontas intersectam com as coberturas das células a planear. Um exemplo é ilustrado na Figura 3.2, onde as células com coberturas a preto serão usadas para o mapa mundo ao contrário das células com cobertura a branco.

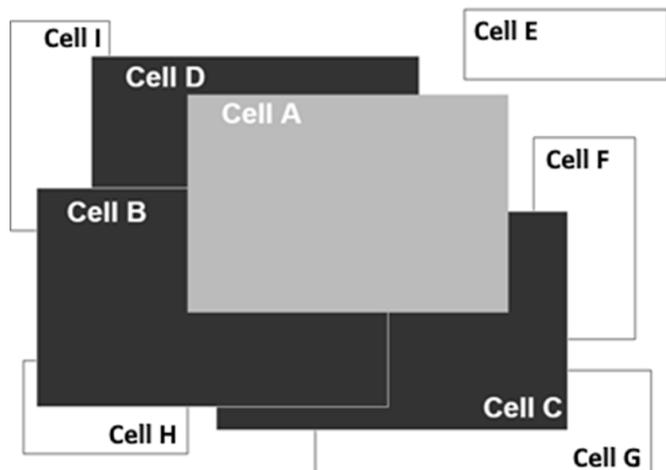


FIGURA 3.2: Exemplo de cobertura das células onde se pretende planejar a célula A.

Depois de identificadas as células que fazem parte do problema, são importadas as grelhas de propagação dessas células. Estas grelhas terão sido pré-geradas, visto que sem elas não é possível perceber se as coberturas se intersectam. As mesmas são calculadas usando o modelo referido na Secção 2.3.1. Um exemplo destas grelhas pode ser visualizada na Figura 3.3. Obtidas todas as grelhas, estas são enviada para o próximo módulo.

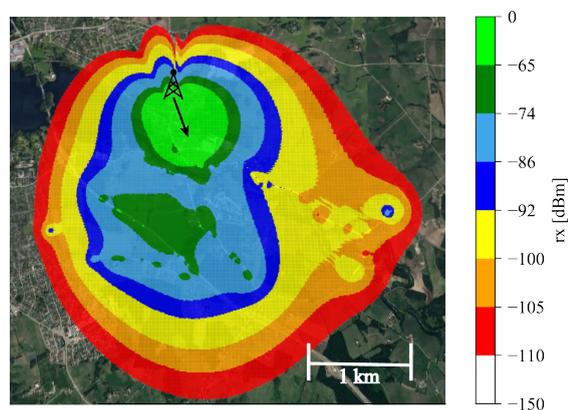


FIGURA 3.3: Representação da estimativa de sinal recebido de uma antena calibrada com *drive tests*

3.4 Módulo de congregação das células

É considerado senso comum que um computador trabalha melhor com dados discretos que contínuos. Sabendo isto, é importante encontrar uma forma de dividir o mundo em pequenas áreas, coerentemente e uniformemente. Isto é necessário de forma a comparar áreas comuns a células diferentes e verificar interferência, tal como ilustrado na Figura 3.4. O Geohash cobre estes requisitos. Apesar deste ter uma margem de erro que cresce quanto mais perto dos pólos o aplicarmos, este problema não é importante, visto que apenas é necessário que as áreas entre as diferentes células sejam na mesma posição e estas tenham a mesmas dimensões. Para cada grelha, há uma conversão das coordenadas em Geohash, tal como explicado na secção 2.4.2. A sua implementação é demonstrada em pseudo-código no Algoritmo 1.

Cada uma destas grelhas é iterada processando toda a informação num mapa cuja chave é o "Geohash " e valor é outro mapa com chave "nome da célula" e valor "potência". Este mapa é depois escrito num ficheiro tal como exemplificado na Figura 3.4.

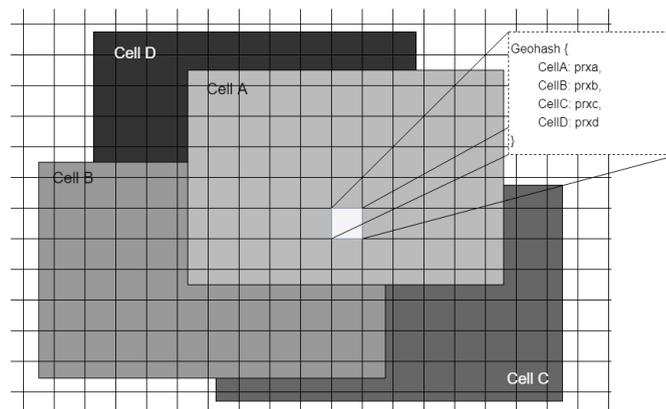


FIGURA 3.4: Representação do mapa mundo e exemplo de escrita de ficheiro numa célula.

Algorithm 1 GeoHash

```

procedure GEOHASH(lat, lng)    ▷ Transforma coordenadas decimais num
GeoHash
    latBits ← getLatBits()        ▷ Numero de bits representantes da latitude
    lngBits ← getLngBits()        ▷ Numero de bits representantes da longitude
    bitArray ← new BitArray(latBits + lngBits)
    min ← -180
    max ← 180
    mid ← 0
    for i < lngBits do
        if lng > mid then
            bitArray.setBit(i * 2)
            min ← mid
        else
            max ← mid
            mid ← (max + min)/2
    min ← -90
    max ← 90
    mid ← 0
    for i < latBits do
        if lat > mid then
            bitArray.setBit(i * 2 + 1)
            min ← mid
        else
            max ← mid
            mid ← (max + min)/2
    return encodeToGeohashBase32(bitArray)

```

3.5 Módulo de planeamento de frequências

3.5.1 Cálculo de interferência

O módulo começa por contabilizar a área de interferência numa tabela cujas linhas são as células por planear e colunas as frequências possíveis de atribuir.

O planeamento de frequências BCCH e TCH requer a escolha de frequências de forma a minimizar a interferência co-canal. Esta interferência ocorre quando, numa determinada área, a potência das células interferentes é suficientemente elevada para resultar num valor abaixo de C/I_{min} , tal como ilustrado na Figura 3.5. São contabilizados todos os pixeis em que tal situação ocorre.

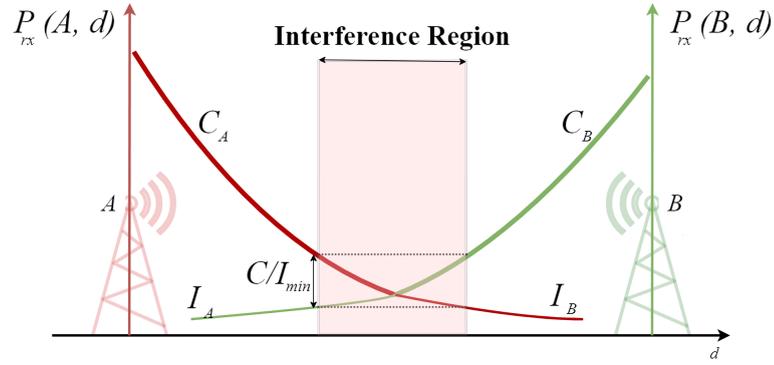


FIGURA 3.5: Exemplo de uma área de interferência tendo em conta o decaimento de potência ao longo de espaço assumindo que A e B tem a mesma frequência.

Por cada célula não planeada, c , é calculada a interferência de cada frequência f . Este procedimento é feito iterando todos os pixels do mapa mundo verificando interferências entre as células presentes nessa área \mathbf{p} . Em cada área \mathbf{p} , para uma frequência f , o *best server*, $c_{f,b}$ será a célula atribuída à frequência f com a maior potência, P_{rx} . As restantes células na mesma área \mathbf{p} e frequência f , será o conjunto de células não *best server*, $\mathbf{c}_{f,nb}$. A interferência é definida pela equação:

$$\frac{P_{rx}(\mathbf{p}, c_{f,b})}{\sum_{c_{f,n} \in \mathbf{c}_{f,nb}} P_{rx}(\mathbf{p}, c_{f,nb})} < (C/I)_{min} \quad (3.1)$$

onde:

- $P_{rx}(\mathbf{p}, c_{f,b})$ [mW]: nível de sinal recebido no pixel \mathbf{p} , da célula *best server* $c_{f,b}$;
- $P_{rx}(\mathbf{p}, c_{f,nb})$ [mW]: nível de sinal recebido no pixel \mathbf{p} , de células não *best server* $c_{f,nb}$, pertencente ao conjunto $\mathbf{c}_{f,nb}$;
- $(C/I)_{min}$: Mínimo C/I que garante comunicações sem interferência. Em GSM é 9 dB.

Assim, obtendo a tabela de interferência de $dim(\mathbf{c}) \times dim(\mathbf{f})$, considerado um conjunto de células \mathbf{c} e uma lista ordenada de frequências \mathbf{f} , cujos valores serão o número de vezes que a equação 3.1 se verificou.

No caso específico em que as células não planeadas pertençam a sites diferentes, é também calculada a interferência entre as mesmas. Este procedimento ocorre,

assumindo que as células acabam com a mesma frequência, obtendo um segunda tabela de interferência de $dim(\mathbf{c}) \times dim(\mathbf{c})$, cujos valores serão 1, se existir interferência entre estas células, ou 0, caso contrário. Estes dados serão também tidos em conta durante a fase de optimização.

3.5.2 Planeamento de frequências

Uma vez obtida a tabela de interferência das células a planear, o planeamento de frequências consiste em escolher a melhor frequência para cada célula. Para tal é usado um otimizador linear. Em geral, os otimizadores lineares precisam de três tipos de informação para obterem uma solução:

- **Objetivo** - Equação que indica ao otimizador qual a melhor solução;
- **Condições** - Condições a que a solução devolvida tem de cumprir de forma a ser válida;
- **Dados** - Dados usados na optimização, sejam estes simples como inteiros, *doubles*, *booleans* ou *strings*, como conjuntos ou até mesmo matrizes.

O nosso objetivo é minimizar a interferência. Seguindo a mesma nomenclatura, o objetivo será representado pela Equação 3.2.

$$\min \sum_{c \in \mathbf{c}} \sum_{f \in \mathbf{f}} \mathbf{D}_{cf} \times \mathbf{X}_{cf}^T \quad (3.2)$$

onde \mathbf{X} é uma matriz de atribuição de frequências (com valores Booleanos) com dimensões iguais à da matriz \mathbf{D} (que constitui a tabela de interferências, cujos elementos representam a área em pixels). A matriz \mathbf{X} é importante porque não só é a matriz onde as condições vão ser aplicadas, como a mesma nos comunicará qual a nossa solução, sendo que é atribuída à célula c a frequência f caso o valor correspondente esteja a "1". Um exemplo é dado na tabela 3.1, solução

TABELA 3.1: Exemplo de solução devolvida pelo otimizador.

Cell\BCCH	63	64	65	66	67	68	69	70	71	72	73	74
Cell 1	0	0	1	0	0	0	0	0	0	0	0	0
Cell 2	0	0	0	0	0	1	0	0	0	0	0	0
Cell 3	0	0	0	0	0	0	0	0	1	0	0	0

da tabela 4.1, onde foram atribuídas as frequências 65, 68 e 71 às células 1, 2 e 3, respectivamente.

As condições enumeradas a seguir, têm objectivo não só de evitar soluções impossíveis como evitar outros tipos de interferência, nomeadamente, a interferência adjacente.

Primeiro, apenas um BCCH pode ser atribuído a cada célula:

$$\forall c \in \mathbf{c} : \sum_{f \in \mathbf{f}} \mathbf{X}_{cf} = 1 \quad (3.3)$$

Segundo, células do mesmo *sites* s , sejam planeadas ou não planeadas, não podem ter os mesmos BCCH's:

$$dado \quad s, \quad \forall f \in \mathbf{f} : \sum_{c \in \mathbf{s}} \mathbf{X}_{cf} \leq 1 \quad (3.4)$$

Terceiro, células do mesmo s , não podem ter frequências adjacentes, a lista ordenada de frequências \mathbf{f} é indexada, a começar no index 0 e a acabar no index $dim(\mathbf{f})$. Considerando isto, a equação pode ser representada com:

$$\forall f_i, i \in [1, dim(\mathbf{f})] : \sum_{c \in \mathbf{s}} (\mathbf{X}_{cf_{i-1}} - \mathbf{X}_{cf_i}) \leq 1 \quad (3.5)$$

Finalmente, caso estejamos a planear células de *sites* diferentes, colocamos a restrição de que células não planeadas que interfiram entre si não podem ter a mesma frequência.

$$\forall f \in \mathbf{f} \text{ and } \forall c_1, c_2 \in \mathbf{c} : (\mathbf{X}_{c_1 f} + \mathbf{X}_{c_2 f}) \times \mathbf{U}_{c_1 c_2} \leq 1 \quad (3.6)$$

Onde $\mathbf{U}_{c_1 c_2}$ é "1" caso c_1 e c_2 se interferem, ou "0" caso contrário. Considera-se que as células interferem-se caso exista pixels com interferência se as células tiverem a mesma frequência. No entanto, esta restrição deveria ser relaxada visto que cria a possibilidade de não existir uma solução. Idealmente, $\mathbf{U}_{c_1 c_2}$ seria um *double* representando a percentagem de área que c_2 interfere c_1 .

Para o planeamento de TCHs existem algumas regras extra a cumprir. Primeiro, visto que cada célula pode ter múltiplos canais de tráfego, de forma a implementar *multiple reuse patterns*, em vez de termos apenas "Cell A", teremos vários "Cell A_ x " onde x serão inteiros diferentes dependendo do número de TCH's. Por exemplo, se "Cell A" tem três canais de tráfego, então teremos "Cell A_1", "Cell A_2" e "Cell A_3". Segundo, entre os TCHs tem de existir, no mínimo, a diferença de duas frequências, logo teremos de adicionar a seguinte condição:

$$\forall f_i, i \in [1, \dim(\mathbf{f})] : \sum_{c \in \mathbf{s}} \sum_{t \in \mathbf{t}_c} (\mathbf{X}_{t f_{i-2}} - \mathbf{X}_{t f_{i-1}} - \mathbf{X}_{t f_i}) \leq 1 \quad (3.7)$$

onde \mathbf{t}_c é o conjunto de canais de tráfego que a célula c tem.

Capítulo 4

Implementação

O Capítulo 4 começa por descrever as tecnologias e ferramentas usadas. De seguida, na Secção 4.2, é explicada a implementação do sistema e certas decisões tomadas. Depois, na Secção 4.3, é esclarecida a forma como o sistema foi integrado com o Metric. Finalmente, na Secção 4.4, são demonstrados vários testes de aferição de implementação do sistema e seus resultados.

4.1 Tecnologias e ferramentas utilizadas

4.1.1 Tecnologias

Nos próximos parágrafos são descritas as tecnologias usadas e as suas finalidades, tal como ilustrado na Figura 4.1.

O Java 8 (Java, 2019) é uma linguagem de programação orientada a objetos, desenvolvida com o objectivo do código ser compilado uma única vez e executado em qualquer máquina que tenha o Java instalado. Todo o código do projecto foi desenvolvido usando esta linguagem.

O Maven (The Apache Software Foundation, 2016) é uma ferramenta de automação de principal uso em projetos Java, no entanto, pode ser usado em projetos

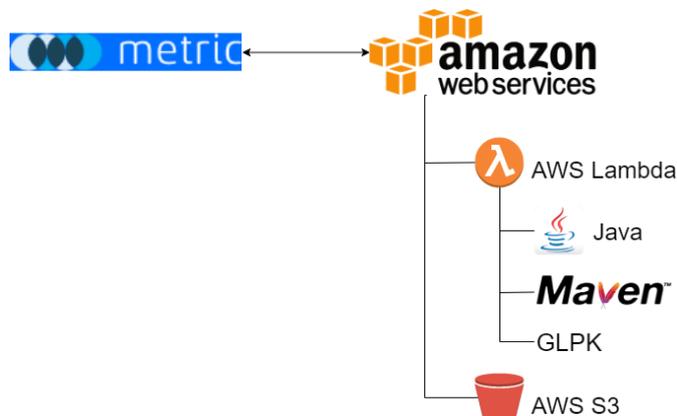


FIGURA 4.1: Mapa de tecnologias e ferramentas usadas.

de C#, Scala, Ruby, entre outras. Bibliotecas e *plug-ins* são guardados num repositório Maven e automaticamente compilado por projetos que os invoquem. Esta tecnologia é principalmente utilizada para a compilação da AWS API no projeto.

O Python 3.7.0 é uma linguagem de programação de alto nível com vários usos em diferentes áreas da programação e investigação. No entanto, para o desenvolvimento deste trabalho, foi apenas usado para observação e análise de resultados.

4.1.2 Ferramentas

Amazon Web Services (Amazon, 2019) é uma subsidiária da Amazon que desenvolveu um conjunto de serviços *cloud*, com base em "pagar por uso". Para este projeto, foi utilizado o "AWS Lambda" e um IaaS onde o código é implantado (neste caso, código Java) e que pode ser executado a partir de um pedido ou *trigger*, como o Amazon Simple Storage Service (Amazon S3), onde é possível guardar dados em formato de objetos. Todos os *inputs/outputs* dos AWS lambdas, são guardados no serviço S3 (PaaS).

Foi discutida a possibilidade de usar outro serviço chamado de "*Elastic Computer Cloud (EC2)*", no entanto, chegou-se à conclusão que não era suposto os módulos executarem-se continuamente. Na realidade, estes eram supostos executarem apenas no momento em que fossem necessários, seja devido a um *trigger* ou por invocação da função. No caso do EC2, teria de se pagar pelo serviço, mesmo

quando este não estava a ser usado, ao contrário do *pay-per-use* do AWS Lambda. O uso dos AWS *cloud services* fez com que fosse possível integrar o novo algoritmo na interface Metric, sendo que agora é possível, um utilizador fazer um pedido de planeamento de células numa rede GSM, na interface Metric SaaS.

GNU Linear Programming Kit (GLPK), é um pacote de apoio a resolução de problemas lineares de grande escala. A sua implementação permite integrar-se em sistemas operativos como Windows e Linux, ou directamente em programas de várias linguagens como Java. Existe uma camada que pode ser aplicada na AWS de forma a invocar ferramentas do pacote de GLPK (Tempusenergy, 2019).

4.2 Implementação

Na figura 4.2 está em detalhe a implementação proposta usando os serviços AWS. Este, recebe informação do OSS e devolve um conjunto de frequências para o planeamento pedido. É possível perceber através da Figura 4.2, que entre cada lambda, todos os dados processados são guardados em S3 *buckets*, cujos ID's depende de informações do cliente. Os dados guardados incluem, mas não exclusivamente: grelhas de propagação, ficheiros de intersecção, mapas mundo, matrizes de interferência, configurações das células, e outros dados do cliente.

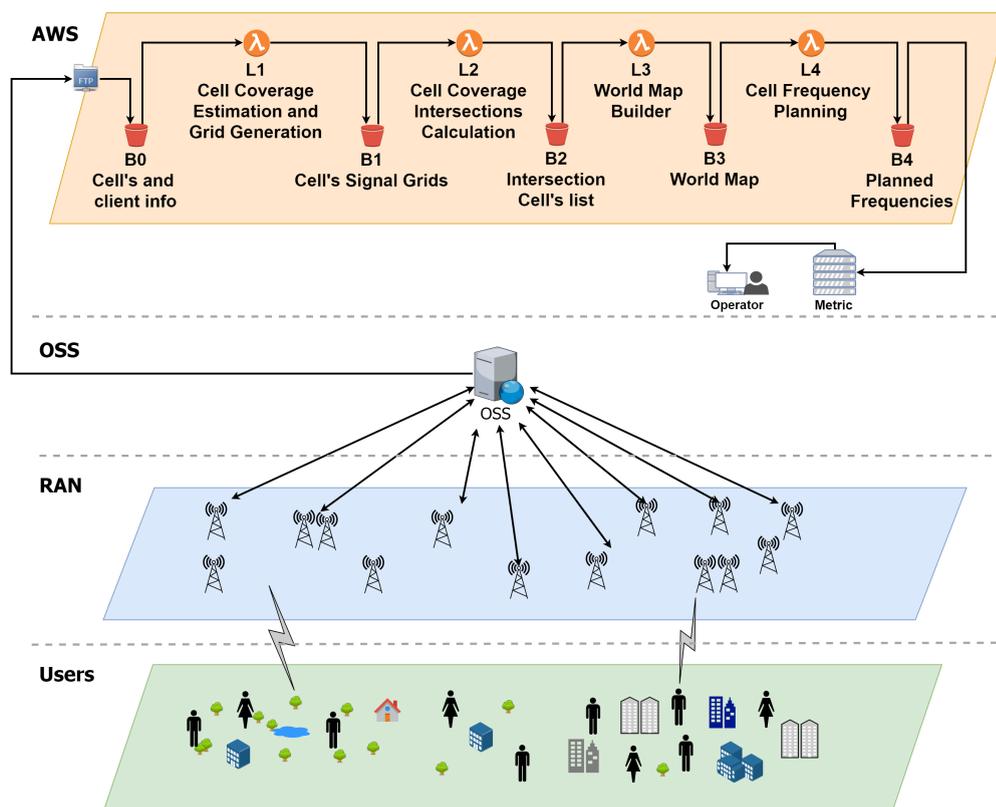


FIGURA 4.2: Integração dos serviços AWS no novo algoritmo.

Cada lambda recebe um ficheiro JSON como *input*. Este ficheiro tem que descrever os nomes das células a planear, a frequência de banda para a qual esta tem de ser planeada (tal como os padrões da International Telecommunication Union (ITU), e informação do cliente (automaticamente escrita baseada na chave de autenticação).

Existem algumas restrições a ter em conta em relação aos lambdas:

- Máximo de memória alocada é 3008 MB;
- Tempo máximo de execução é 900 segundos (15 min);
- 500 MB de memória virtual.

Estas informações são importantes uma vez que o mapa mundo pode ocupar uma quantidade considerável de espaço, logo, é necessário a implementação de *streaming* na leitura do mapa mundo. O mesmo se aplica no *upload* do mapa para

um S3 *bucket*, visto que o ficheiro tem primeiro de ser criado em memória virtual. É possível que constrangimentos ocorram caso se planeie um problema demasiado grande, onde o processo de otimização ultrapasse o limite dos 900 segundos.

Outro possível problema seria o facto dos S3 *buckets* terem um limite de 5 TB por objeto, mas este não é comparável com o limite de 500 MB do lambda. Existe também um limite de 100 *buckets* por conta (1000 caso seja *premium*), o que mais uma vez, não é problema, visto que no máximo seriam precisos cinco *buckets*.

Na Figura 4.2 temos os seguintes lambdas:

- **Cell coverage estimation and grid generation AWS Lambda (L1):** Através da informação do cliente, importada do S3 *bucket* (B0), o lambda cria uma grelha de *pixels* com informação de cobertura nos arredores da área. Esta cobertura é calculada usando modelos de propagação, *drive tests*, e outros parâmetros como a elevação, azimute, e inclinação (Fernandes et al., 2019). Estas grelhas serão salvas no S3 *bucket* (B1);
- **Intersections AWS Lambda (L2):** Através de um ficheiro de metadados importado de B1, L2 compila esses dados verificando se cada uma das coberturas de células interseam. O módulo funciona como explicado na Secção 3.4. Finalmente, os dados são exportados num ficheiro CSV para B2;
- **World Map AWS Lambda (L3):** Este lambda é responsável por criar um ficheiro TSV (mapa mundo) em que cada linha é composta por um Geohash e pares de células/potência. Tanto as grelhas de propagação como o mapa mundo têm de ter a mesma precisão de Geohash (tamanho do Bitset) de forma a evitar problemas de incoerência de escala/localização. Primeiro, são usados os ficheiro de intersecção de B2 para verificar todas as células cuja cobertura interseam com as que estão a ser planeadas. As grelhas de propagação destas células serão também importadas de B1, de forma a se compilar e gerar o mapa mundo para o nosso problema. Inicialmente, o mapa mundo era um ficheiro JSON, até se perceber que este ocupava uma enorme quantidade de espaço, o suficiente para que o uso de lambdas não

fossem viáveis, seja por passar do limite de espaço como o tempo perdido na leitura do ficheiro;

- **Planning AWS Lambda (L4)**: Primeiro, é compilada tanto a quantidade de interferência existente no mapa mundo (Tabela 4.1), como a possível interferência no caso de ser atribuída às células não planeadas a mesma frequência. Interferência é definida pela Equação (3.1). Estes dados são escritos num ficheiro "mathprog" que é depois enviado para o módulo GLPSOL da livreria GLPK (Makhorin, 2012), o otimizador linear. Dentro deste ficheiro é enviado também o objetivo do otimizador (Equação (3.2)) e todas as regras de interferência (Equações (3.2)–(3.7)). A solução exportada pelo otimizador será um documento com múltiplos $x[c, f]$, onde x é 1 ou 0. No primeiro caso, significa que a frequência f será atribuída á célula c . Estes dados são exportados do documento usando o regex—"`\[(\d+),(\w*)\]\s?\W*1`"— processados e devolvidos ao utilizador.

Todos os ficheiros importados dos S3 *buckets* são lidos em *stream*, sem existir a necessidade de salvar a informação em ficheiros temporários e evitar a possibilidade de ultrapassar o limite de 500 MB.

As tabelas de interferência (Tabela 4.1) terão como linhas os canais que estamos a tentar planear, como colunas as possíveis frequências a atribuir e como valores a quantidade de áreas no mundo (Geohashes) que teriam interferência se fosse aplicada essa combinação frequência canal.

TABELA 4.1: Exemplo de tabela de interferência

Cell\BCCH	63	64	65	66	67	68	69	70	71	72	73	74
Cell 1	2	3	0	374	374	223	6	6	0	0	1	63
Cell 2	32	32	7	85	13	4	14	14	0	0	12	22
Cell 3	279	279	279	258	258	128	232	232	0	0	237	237

4.3 Integração no Metric

A implementação foi integrada com o Metric SaaS, e está disponível na versão mais recente do mesmo. O utilizador começa por seleccionar uma área do mapa que contém as células a planear, seguido pelo pedido de planeamento (Figura 4.3). O sistema envia então um pedido aos serviços AWS, que, depois de processar os dados, devolve o melhor planeamento para a área.

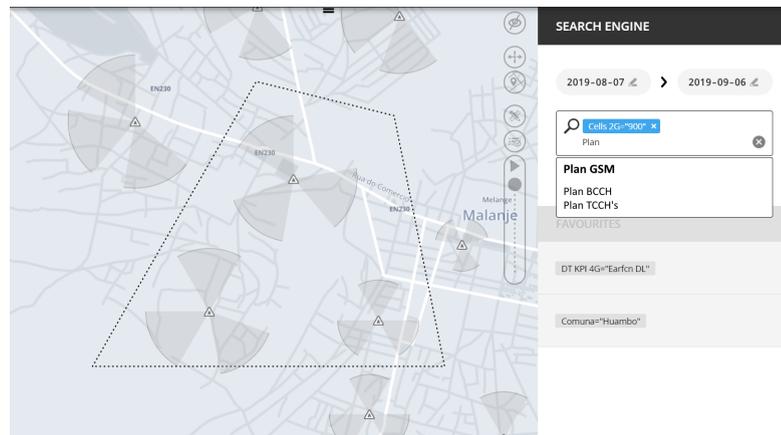


FIGURA 4.3: Pedido de planeamento de frequências no Metric.

O *output* será um ficheiro JSON semelhante ao da Figura 4.4. O campo "code", depende do sucesso da operação, sendo "200" sucesso, "400" falta de dados de *input*, e "500" um problema interno do sistema (Por exemplo, falta de ficheiros necessários para a operação e problemas internos com o GLPK). Uma descrição do problema é adicionada ao JSON caso o processo não tenha sucesso.

Este ficheiro poderá ser encaminhado de forma automática para o OSS que implementa a configuração proposta nas respectivas células, seguindo o paradigma SON de auto-configuração.

4.4 Testes

Nesta secção são descritos uma série de testes e seus resultados para aferir o correto funcionamento da implementação.

```
1 {
2   "code": 200,
3   "result": {
4     "CELLA": "70",
5     "CELLB": "63",
6     "CELLC": "63",
7     "CELLD": "72",
8     "CELLE": "70",
9     "CELLF": "72"
10  }
11 }
```

FIGURA 4.4: Exemplo de ficheiro de *output* do sistema.

Como primeiro cenário de teste, foi feito uma pequena simulação usando *Excel*, com duas células planeadas com o mesmo BCCH e outra célula por planear. As coberturas das três células são sobrepostas em quatro *pixels* com potências diferentes, cobrindo os vários casos possíveis. As descrições dos testes realizados e os seus resultados, estão descritos nas tabelas 4.2 e 4.3, respectivamente.

TABELA 4.2: Testes unitários e previsões

Nº	Teste	Resultado Previsto
1	Pequeno cenário em excel: Célula A: -120 dBm Célula B: -120 dBm Célula C: -110 dBm	Célula C é poluidora. Sendo que a soma de potências A+B é -116.99 dBm.
2	Pequeno cenário em excel: Célula A: -100 dBm Célula B: -110 dBm Célula C: -120 dBm	Célula C é poluente. Sendo que a soma de potências A+B é -99.5861 dBm.
3	Pequeno cenário em excel: Célula A: -90 dBm Célula B: -9 dBm Célula C: -120 dBm	Célula C é poluente. Sendo que a soma de potências A+B é -9 dBm.
4	Pequeno cenário em excel: Célula A: -90 dBm Célula B: -120 dBm Célula C: 0 dBm	Célula C não é poluidora nem poluente. Sendo que a soma de potências A+B é -89.9957 dBm.
5	Cenário de otimização de uma antena com uma matriz de poluentes gerada de um caso real (MALANGE)	Encontrada solução determinística reduzindo a quantidade de ruído.
6	Cenário de otimização de várias antenas com matriz de poluentes vazia.	Encontrada solução não determinística, obtendo resultados diferentes em diferentes execuções.
7	Cenário de otimização com menos BCCH's que células para planejar	Encontrada solução determinística.

TABELA 4.3: Resultados dos testes unitários.

Nº	Resultado alcançado
1	Comprovou-se.
2	Comprovou-se.
3	Comprovou-se.
4	Comprovou-se.
5	Encontrada solução determinística reduzindo a quantidade de ruído.
6	Encontrada solução, sendo esta igual em todas as execuções.
7	Encontrada solução, caso exista frequências suficientes para respeitar as regras de vizinhança.

Para um segundo cenário de teste, foram realizadas várias otimizações, usando o GLPSOL, dando especial importância a casos excepcionais que possam devolver erros, como por exemplo, uma matriz de poluidores completamente vazia, ou casos em que não é possível haver otimizações que respeitem todas as regras.

O otimizador também foi testado, foi lançado como entrada ao otimizador de dados (Tabela 4.4), de forma a verificar se este está a cumprir as regras.

Seguindo as regras de planeamento, uma atribuição correta seria: Cell C (64), CELL B (70/71) e CELL A (66). Isto porque as células não podem ter frequências adjacentes, logo seria impossível atribuir 70 e 71 a duas células diferentes. Sendo assim, este procurará pelos dois valores menores, neste caso 17880 para CELL A e 0 para CELL B. De facto, o glpsol atribui as frequências 64, 70 e 66 a CELL C, CELL B e CELL A respectivamente.

TABELA 4.4: Exemplo 1 de uma tabela de poluição

Frequência	CELL A	CELL B	CELL C
63	72204	64916	66520
64	26296	21225	0
65	61214	53657	8888
66	1788	23016	68637
67	54307	32297	35483
68	48794	26681	29626
69	55904	62417	61137
70	0	0	0
71	0	0	0
72	42868	40517	17115
73	66202	52587	20545

Foi necessário verificar o comportamento do otimizador nos casos em que a obtenção de uma solução seria impossível, por exemplo, se o número de frequências para planeamento de um site fosse igual ou menor ao número de células. Neste caso, a atribuição de frequência é impossível visto que não podem existir frequências iguais ou adjacentes. O otimizador detecta isto devolvendo um ficheiro sem soluções, junto com uma mensagem de “LP HAS NO PRIMAL FEASIBLE SOLUTION”.

TABELA 4.5: Exemplo 2 de uma tabela de poluição

Frequency	CELL A	CELL B	CELL C
63	72204	64916	66520
64	26296	21225	0
65	61214	53657	8888

Finalmente, verificou-se o planeamento de células de sites diferentes. Nestes casos, são necessárias duas tabelas extra: Uma que identifica o site de cada célula (Tabela 4.6), e outra que indica que células não podem ter a mesma frequência visto que a cobertura das duas se interseta (Tabela 4.7). O intervalo de frequências usado está entre 63 e 69, inclusive. O que quer dizer que, de forma a respeitar as regras de canal-adjacente, as células do *site 1* ("S1C1" significa "*site 1*" e "célula 1") têm de ficar com as frequências 63, 65, 67 e 69. A dúvida está na atribuição de frequências às células do *site 2*, visto que, de acordo com a Tabela 4.8, a melhor opção é a frequência 69, no entanto, a célula S2C1 tem também a opção da frequência 66. Correctamente, o otimizador devolve a seguinte atribuição:

- S1C1 - 63;
- S1C2 - 65;
- S1C3 - 67;
- S1C4 - 69;
- S2C1 - 66;
- S2C2 - 69.

Assim, podemos concluir que todos os requisitos funcionais foram satisfeitos.

TABELA 4.6: Tabela de indicação de sites.

Célula	Site 1	Site 2
S1C1	1	0
S1C2	1	0
S1C3	1	0
S1C4	1	0
S2C1	0	1
S2C2	0	1

TABELA 4.7: Tabela de indicação de interferência caso células tenham a mesma frequência.

Célula	S1C1	S1C2	S1C3	S1C4	S2C1	S2C2
S1C1	0	1	1	0	1	0
S1C2	0	0	0	0	0	0
S1C3	1	0	0	1	1	0
S1C4	1	0	0	0	0	0
S2C1	1	0	0	0	0	0
S2C2	0	0	1	0	0	0

TABELA 4.8: Tabela de interferência para teste de aferição 2

Frequency	S1C4	S1C3	S2C1	S2C2	S1C2	S1C1
63	98577	16720	48919	72529	81004	20312
64	66355	40520	14933	5873	21464	41453
65	97927	8068	94315	98535	9013	68674
66	12218	4542	0	24273	88195	91260
67	56383	12784	54163	88636	97191	22376
68	23112	77550	30996	79953	74103	26412
69	0	0	0	0	0	0

Capítulo 5

Resultados

Neste capítulo são definidos os cenários de referência para avaliação do algoritmo (Secção 5.1), seguido pelas Secções 5.2 e 5.3, onde são discutidos resultados em aspetos como a qualidade das soluções e os recursos utilizados.

5.1 Cenários de Referência

Nesta secção são apresentados os cenários de referência utilizados para avaliar o desempenho dos algoritmos propostos.

Depois de alguns testes de avaliação, o modelo foi aplicado a um cenário de avaliação. Este consiste em dados realistas de uma rede GSM que cobre 2422 km de área, com 570,000 habitantes. A Figura 5.1, representa este cenário de referência com a localização das células. Este tem 23 *sites* com um total de 70 células. O centro da cidade está intensamente coberto, enquanto que nas zonas rurais, apenas estão cobertas as estradas. Os BCCH's utilizados pelo operador na banda do 900 MHz são os canais entre 63 e 88, e na banda dos 1800 MHz, estão entre 733 e 769. Os canais de TCH para a banda dos 900 MHz estão entre o 89 e 124, enquanto que para os 1800 MHz, estão entre os 770 e 824.

O cenário foi dividido em dois cenários de referência:

1. **Planeamento de um único *site*:** Otimização de um único *site* com três células que foi manualmente planeado, neste caso, o *site* colorido a azul na figura 5.1;
2. **Planeamento de múltiplos *sites* de uma rede,** Planeamento de frequências do cenário ilustrado na Figura 5.1, para um total de 70 células.

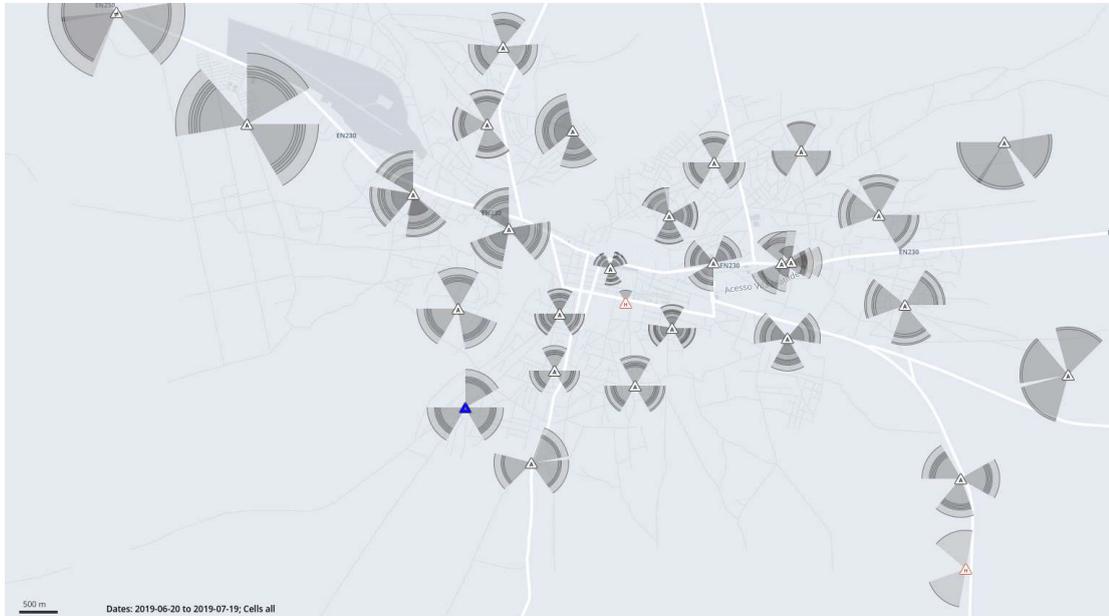


FIGURA 5.1: Cenário de avaliação utilizado, extraído de uma existente rede GSM.

5.2 Análise de resultados para planeamento de um único *site*

Para o planeamento de um único *site* com três células, foram gerados vários mapas de interferência com o C/I de cada pixel coberto por esse *site*: Um com o planeamento de frequências original (Figura 5.2a) e outro usando o novo algoritmo (Figura 5.2b). Estes resultados foram também comparados com o planeamento obtido pela implementação do Metric (descrita na secção 2.3.3).

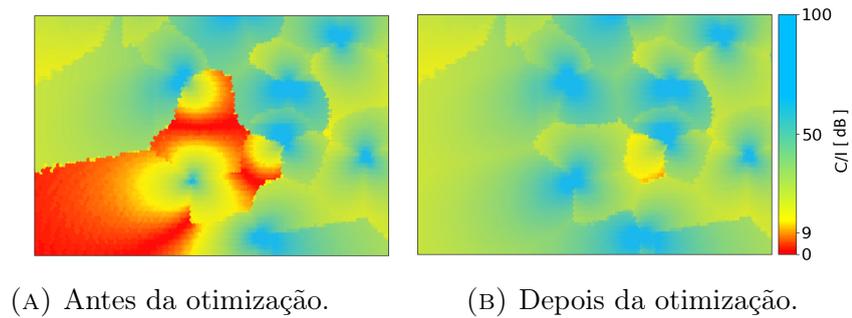


FIGURA 5.2: C/I da área coberta pelo *site* planejado.

Foi gerada uma função empírica de distribuição de forma a comparar a qualidade da rede (Figura 5.3). Esta distribuição não contabiliza áreas em que a célula mais potente não é suficiente para habilitar a conectividade, uma vez que a qualidade do planejamento não tem influencia sobre as mesmas. Como se pode ver na Figura 5.3, o planejamento do novo algoritmo comparado com o algoritmo Metric, obtiveram qualidade de rede semelhantes, mesmo tendo em conta que as frequências alocadas foram diferentes.

Antes da otimização, o planejamento manual, visualizável na Figura 5.2a, tinha 83.7% da área coberta com interferência menor que $(C/I)_{min}$ (9dB). É possível perceber a baixa qualidade da área pela média de C/I de 7 dB. Depois do planejamento, é observável na Figura 5.3 que, 99.9% da área coberta está com C/I superior a 15 dB, estabelecendo comunicações com excelente qualidade.

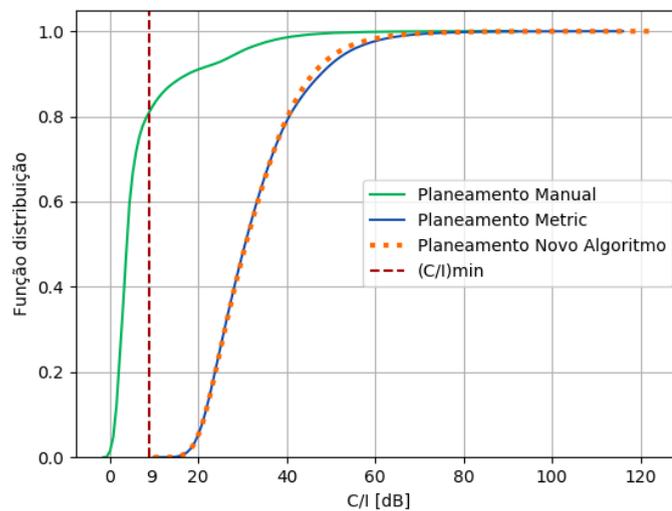


FIGURA 5.3: Função de distribuição do C/I dentro da área coberta pelo *site* planejado.

Apesar dos dois algoritmos terem resultados semelhantes, o seu desempenho (tempo de execução e uso de memória) é diferente. Esta diferença pode ser visualizada na Figura 5.4. Para o planejamento de três células, o novo algoritmo usou 985 MB de memória e demorou 27.7 segundos de execução, uma melhoria quando comparado com os 4096 MB utilizados e 329.2 segundos do algoritmo Metric, ou seja, o novo algoritmo foi 12 vezes mais rápido, e usou 4 vezes menos memória que o algoritmo Metric. Estes resultados excluem o tempo e a memória utilizados para calcular as grelhas de propagação, que são pré-processadas e disponíveis no Metric SaaS. Estas melhorias são justificadas por, não só as divisões do mundo serem mais uniformes, como também o mapa mundo é gerado uma única vez ao contrário de múltiplas vezes (uma por célula).

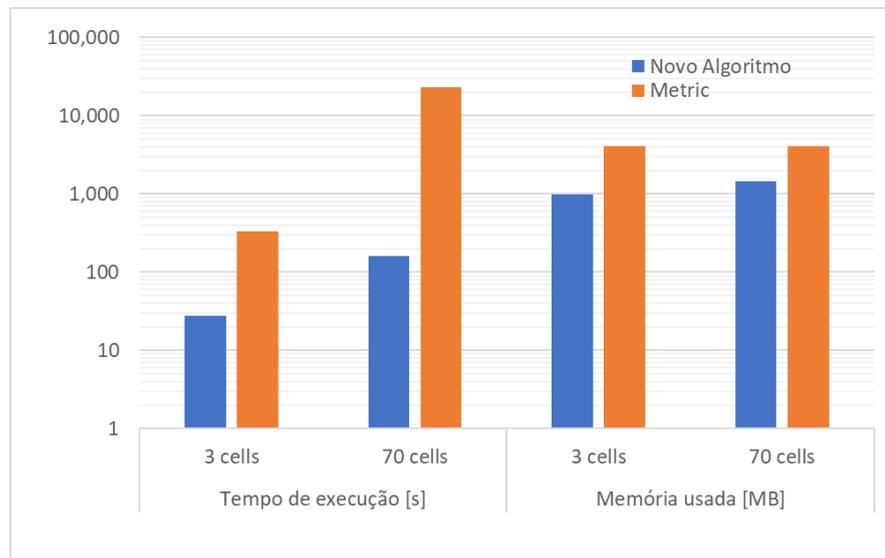


FIGURA 5.4: Comparação do novo algoritmo com o velho algoritmo, tanto em tempo de execução como espaço ocupado.

5.3 Análise de resultados para planeamento de múltiplos sites

De forma a avaliar o comportamento do algoritmo no planeamento de múltiplos *sites*, o mesmo foi aplicado às 70 células do cenário de avaliação. Este demorou 161.6 segundos e usou 1439 MB de memória (Figura 5.5). Não é possível comparar estes resultados com o algoritmo Metric, uma vez que este só consegue planear um *site* de cada vez. No entanto, pode considerar-se que este é capaz de planear as 70 células, uma de cada vez, estimando uma demora de 70 vezes 329 segundos (23.044 segundos). No entanto, existe outro problema, em que a solução obtida pode não ser a melhor devido à ordem de planeamento das células. Esta é uma grande limitação do algoritmo Metric, em que as condições de optimização usadas só se atribuem a células no mesmo *site*. Por consequência, o planeamento de várias células tem de ser sequencial.

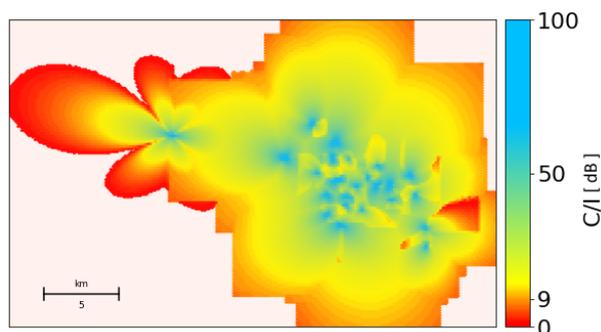


FIGURA 5.5: C/I na cidade depois do planeamento de todas as células.

O algoritmo proposto é composto por pesquisa no mapa mundo (*hashtable*) e a optimização linear. Para os próximos resultados, "World map" cobre tudo desde a leitura dos ficheiros necessários para a geração do mapa mundo, até ao processo de enviar o problema para optimização. A complexidade de pesquisa numa *hashtable* é $O(n)$, e a biblioteca de optimização linear usada, tem uma complexidade espacial de $poly(n)$ (Jain & Yao, 2011). Logo, o novo algoritmo também tem uma complexidade temporal de $poly(n)$. É possível confirmar esta complexidade na figura 5.6, em que, apesar de à primeira vista a curva parecer linear, é na verdade polinomial. A razão para esta ilusão está no facto de que a curva polinomial da optimização não ter muito peso ao planear um baixo número de células. As pequenas variações no tempo de execução são justificadas pelo tempo consumido na leitura e escrita na memória.

Em relação à complexidade espacial, ilustrada na Figura 5.7 com uma escala logarítmica, apesar de a optimização linear mostrar uma complexidade espacial de $poly(n)$, o espaço ocupado pelo mesmo não é significativo quando comparado com a quantidade de memória usada pelo módulo de criação do mapa mundo. Aliás, não é possível calcular a complexidade espacial para esse módulo, uma vez que esta depende mais das localizações das células que do mapa. Por exemplo, o espaço ocupado por três células perto umas das outras vai ser menor que o espaço ocupado por três células longe umas das outras, uma vez que a área coberta é maior. Esta observação pode ser constatada na Figura 5.7, onde há uma maior diferença de memória usada entre as 12 e 17 células.

Finalmente, procuramos conclusões em relação ao impacto dos serviços *cloud* no algoritmo, comparando tempos de execução do novo algoritmo numa máquina física com a implementação *cloud* (Figura 5.8). A execução na máquina física foi ligeiramente mais rápida. Esta discrepância é explicável pelo facto que a implementação *cloud* perde algum tempo nos processo de *downloading/uploading* ficheiros, incluído o abrir/fechar conexões com o serviço S3.

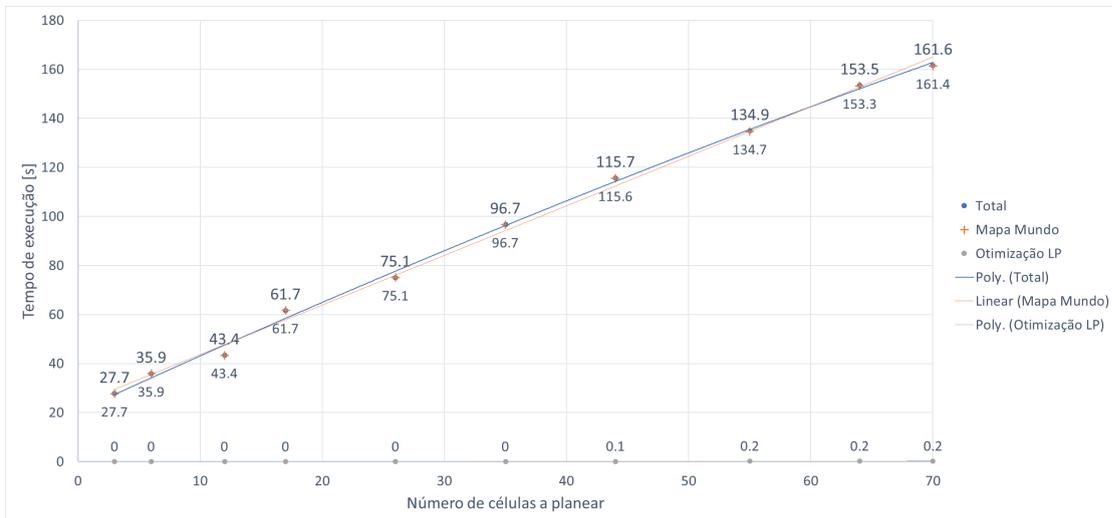


FIGURA 5.6: Tempo de execução com o incremento de células a planear.

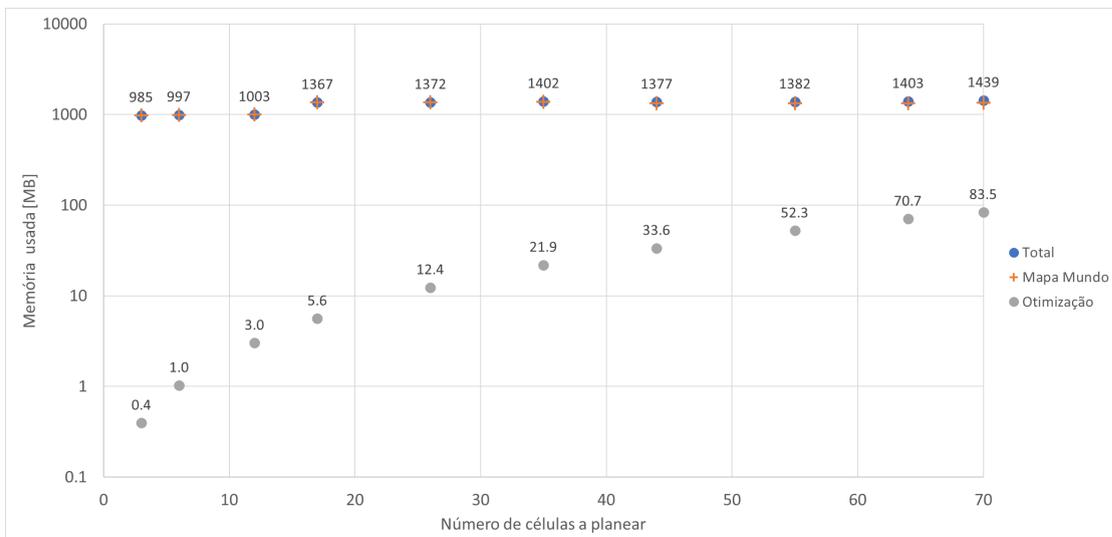


FIGURA 5.7: Quantidade de espaço ocupado com o incremento de células a planear

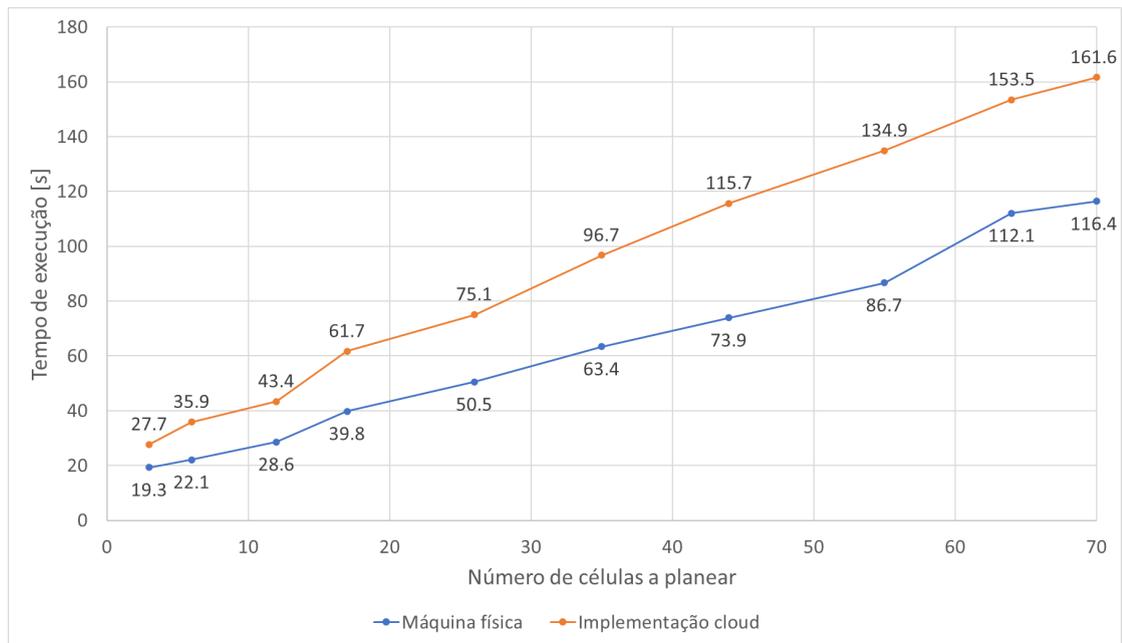


FIGURA 5.8: Comparação de desempenho dos algoritmos entre este ser executado em *cloud* e numa máquina física.

Capítulo 6

Conclusão

6.1 Principais conclusões

Nesta dissertação foi proposta uma nova forma de, automaticamente, planejar redes móveis celulares utilizando otimização linear e sistemas *cloud*. Enquanto que o sistemas *cloud* permitem a modularização do sistema, a otimização linear permite encontrar a melhor solução para o nosso problema. O sistema é baseado num algoritmo anteriormente implementado numa ferramenta comercial de planeamento de redes, onde foram aplicadas múltiplas melhorias, sejam estas em estimativas de cobertura, uniformização do mundo, e desempenho do algoritmo.

O modelo começa por estimar a área coberta pelas células a planejar e as células cuja cobertura intersecta com as primeiras. De seguida, é criado um mapa, dividido em pequenas áreas, com todas as estimativas de cobertura, de forma a ser possível calcular interferência das mesmas. Finalmente, calculada e criada a tabela de interferência, esta é enviada com um conjunto de regras para um otimizador linear, que nos devolve a melhor solução, neste caso, o melhor conjunto de frequências a ser atribuído.

A implementação foi feita num conjunto de módulos, ligados linearmente, em serviços *cloud*, neste caso, utilizando *AWS lambdas* e *S3 buckets*. Todo o código foi escrito em Java e o otimizador usado faz parte da livreria GLPK.

Para um cenário realista, com um *site* problemático, onde 83.7% da área de cobertura tinha interferência, o algoritmo proposto foi capaz de remover quase toda a interferência presente na área, garantindo excelentes condições de comunicação nessa área. Este processo foi 12 vezes mais rápido e usou um quarto da memória, quando comparado com uma ferramenta de planeamento executada numa máquina física (que só era capaz de planear um sites individualmente). Foi também discutida a complexidade temporal e espacial no planeamento de uma rede de 70 células, percebendo que otimizar grandes redes possa demorar demasiado tempo para ser considerado razoável.

Os resultados mostram que a implementação *cloud* é ligeiramente mais lenta que a implementação física (cerca de 50% mais lenta). No entanto, existem vantagens dos sistemas *cloud* a ter em conta. Por exemplo, o facto destes serviços serem *pay-per-use* permite rapidamente obter *hardware* necessário ao sistema. Outro exemplo seria o facto de cada módulo do sistema ser usável em outros sistemas por uma simples chamada.

Respondendo às questões de investigação:

- **Como é que o planeamento de frequências beneficia de sistemas *cloud*?** - Apesar de o algoritmo ter um tempo de execução ligeiramente superior quando executado num sistema *cloud*, este tem a vantagem de não existir a necessidade de compra de *hardware* próprio e de cada módulo poder ser usado por outros sistemas. Por exemplo, o módulo L3 descrito na secção 4.2, é usado noutro sistema também implementado no Metric, neste caso, para o cálculo de vizinhas (Pina et al., 2020);
- **Será possível o planeamento de várias células em simultâneo?** - Sim, desde que o número de células a planear seja o suficiente para respeitar as regras de interferência de canal-adjacente.

6.2 Propostas de investigação futura

Para trabalho futuro, propõe-se a extensão do modelo de forma a funcionar em outras tecnologias como UMTS, LTE, e até mesmo o moderno 5G. Propõe-se também uma avaliação do sistema utilizando outros otimizadores, como o Atoll (Forsk, n.d.) e o WinProp (Altair, 2019), de forma a, não só verificar se é possível tempos de execução mais baixo, como se as soluções devolvidas para o mesmo problema são determinísticas. Seria também importante, experimentar outros otimizadores lineares mais eficientes que o GLPK, tal como explicado na Secção 2.4.3. Finalmente, seria importante perceber quais os limites de uma rede em que esta solução deixa de ser viável. Estas limitações vêm provavelmente do limites do lambda AWS, uma vez que existe tanto um limite de espaço como de tempo. Tendo em conta essa limitação, seria importante investigar uma forma de dividir o mundo em problemas mais pequenos, mantendo a fiabilidade da solução.

Referências

- Altair. (2019). *WinProp Software for Wave Propagation and Radio Planning*. Retrieved 2020-04-04, from <https://altairhyperworks.com/product/Feko/WinProp-Propagation-Modeling>
- Amazon. (2019). *Amazon Web Services*. Recuperado de <https://aws.amazon.com>
- Assunção, P., Estevinho, R., & Correia, L. M. (2001). Assessment of cellular planning methods for GSM. In *Ieee international symposium on personal, indoor and mobile radio communications, pimrc* (Vol. 1). doi: 10.1109/pimrc.2001.965475
- Avram, M. (2014). Advantages and Challenges of Adopting Cloud Computing from an Enterprise Perspective. *Procedia Technology*, 12, 529–534. Recuperado de <https://linkinghub.elsevier.com/retrieve/pii/S221201731300710X> doi: 10.1016/j.protcy.2013.12.525
- AWS. (2018). *AWS Lambda – Serverless Compute - Amazon Web Services*. Retrieved 2020-01-12, from <https://aws.amazon.com/lambda/>
- Bern, M., Eppstein, D., & Teng, S. H. (1993). Parallel construction of quadtrees and quality triangulations. In *Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics)* (Vol. 709 LNCS, pp. 188–199).
- Broadbent, T. A. A., Gauss, K. F., Hildebeitel, A., & Morehead, J. (1967). General Investigations of Curved Surfaces. *The Mathematical Gazette*, 51(375), 69. doi: 10.2307/3613647
- Engstrom, S., Johansson, T., Kronstedt, F., Larsson, M., Lidbrink, S., & Olofsson, H. (1998). Multiple Reuse Patterns for frequency planning in GSM

- networks. In *Ieee vehicular technology conference* (Vol. 3, pp. 2004–2008). IEEE. doi: 10.1109/vetec.1998.686107
- Ericsson. (2019). *Ericsson Mobility Report November 2019* (Tech. Rep.). Author. Recuperado de www.ericsson.com/mobility-report
- Fernandes, D., Ferreira, S., Soares, G., Sebasti, P., Cercas, F., Dinis, R., ... Cortes, R. (2019). Combining Measurements and Propagation Models for Estimation of Coverage in Wireless Networks. In *2019 ieee 90th vehicular technology conference (vtc-fall)*. IEEE.
- Forsk. (n.d.). *Atoll Radio Planning Software Overview | Forsk*. Retrieved 2020-04-04, from <https://www.forsk.com/atoll-overview><http://www.forsk.com/atoll-overview>
- Forsk. (1997). *Atoll 3.3.0 Technical Reference Guide for Radio Networks* (Tech. Rep.). Author. Recuperado de <http://solutionsproj.net/software/atoll.pdf>
- Freeman, R. L. (2005). *Fundamentals of Telecommunications, Second Edition*. John Wiley & Sons, Inc.
- Gearhart, J. L., Adair, K. L., Detry, R. J., Durfee, J. D., Katherine, A., & Martin, N. (2013). *Comparison of Open-Source Programming Solvers Linear* (Tech. Rep. No. October). Sandia National Laboratories.
- Gilbert, J. R., Leiserson, C. E., & Street, C. (2009). Parallel Sparse Matrix-Vector and Matrix-Transpose-Vector Multiplication Using Compressed Sparse Blocks Categories and Subject Descriptors. *ACM Symp. on Parallelism in Algorithms and Architectures..* Recuperado de <https://people.eecs.berkeley.edu/~aydin/csb2009.pdf>
- Huurdeman, A. A. (2003). *The Worldwide History of Telecommunications*. Hoboken, NJ, USA: John Wiley & Sons, Inc. Recuperado de <http://doi.wiley.com/10.1002/0471722243> doi: 10.1002/0471722243
- Jain, R., & Yao, P. (2011). A parallel approximation algorithm for positive semidefinite programming. In *Proceedings - annual ieee symposium on foundations of computer science, focs* (pp. 463–471). doi: 10.1109/FOCS.2011.25
- Java. (2019). *Java | Oracle*. Retrieved 2019-09-16, from <https://www.java.com/>

en/

- Makhorin, A. M. A. I. (2012). *GLPK - GNU Project - Free Software Foundation (FSF)*. [Online] Available: <https://www.gnu.org/software/glpk/>. Retrieved 2019-07-19, from <https://www.gnu.org/software/glpk/>
- Meindl, B., & Templ, M. (2012). *Analysis of commercial and free and open source solvers for linear optimization problems* (Vol. 1; Tech. Rep. No. 1). Institut f. Statistik u. Wahrscheinlichkeitstheorie. Recuperado de <http://www.statistik.tuwien.ac.athttp://www.statistik.tuwien.ac.at/forschung/CS/CS-2012-1complete.pdf>
- Mell, P. M., & Grance, T. (2011). *The NIST Definition of Cloud Computing. Recommendations of the National Institute of Standards and Technology*. (Tech. Rep.). Gaithersburg, MD: National Institute of Standards and Technology. Recuperado de <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf> doi: 10.6028/NIST.SP.800-145
- Mishra, A. R. (2007). *Advanced cellular network planning and optimisation : 2G/2.5G/3G - evolution to 4G*. John Wiley.
- MongoDB. (2019). *MongoDB Manual - 2d Index Internals*. Retrieved 2019-09-03, from <https://docs.mongodb.com/manual/core/geospatial-indexes/>
- Morton, G. M. (1966). *A computer oriented geodetic data base and a new technique in file sequencing* (Tech. Rep.). Ottawa, Canada: IBM Ltd.
- Multivision. (2019). *Metric Software-as-a-Service*. Recuperado de <https://metric.pt>
- Niemeyer, G. (2008). *Geohash/ Labix Blog*. Retrieved 2019-06-21, from <https://web.archive.org/web/20080305223755/http://blog.labix.org:80/{#}post-85>
- Pina, P. M., Godinho, A. F., Fernandes, D. F., Clemente, D. J., Sebastião, P., Soares, G. E., & Ferreira, L. S. (2020). Automatic Coverage Based Neighbour Estimation System: A Cloud-Based Implementation. *IEEE Access*, 8, 69671–69682. doi: 10.1109/ACCESS.2020.2986755
- Popoola, S. I., Atayero, A. A., Faruk, N., Calafate, C. T., Olawoyin, L. A., & Matthews, V. O. (2017). Standard propagation model tuning for path loss

- predictions in built-up environments. In *Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics)* (Vol. 10409 LNCS, pp. 363–375). Springer Verlag. doi: 10.1007/978-3-319-62407-5_26
- Ramiro, J., & Hamied, K. (2012). *Self-organizing networks : self-planning, self-optimization and self-healing for GSM, UMTS and LTE*. Wiley.
- Roberts, M. (2016). Serverless Architectures. *Martinfowler.Com*, 1–36. Recuperado de <http://martinfowler.com/articles/serverless.html>
- Su, C., Lan, L., Yu, C., Gou, X., & Zhang, X. (2010). A new method of frequency planning for new cells in GSM. *2010 2nd Conference on Environmental Science and Information Application Technology, ESIAT 2010*, 3, 420–423. doi: 10.1109/ESIAT.2010.5568326
- Tempusenergy. (2019). *GitHub - tempusenergy/glpk-lambda-layer: Use an LP Solver in your AWS Lambda functions*. Retrieved 2020-01-12, from <https://github.com/tempusenergy/glpk-lambda-layer>
- The Apache Software Foundation. (2016). *Maven – Welcome to Apache Maven*. Retrieved 2019-09-16, from <https://maven.apache.org/>
- Valsalam, V., & Skjellum, A. (2002). A framework for high-performance matrix multiplication based on hierarchical abstractions, algorithms and optimized low-level kernels. *Concurrency Computation Practice and Experience*, 14(10), 805–839. Recuperado de http://people.cs.vt.edu/~asandu/Public/Qual2005/Q2005_{_}skjellum.pdf doi: 10.1002/cpe.630
- Worldtimezone.com. (2016). *GSM World Coverage Map- GSM Country List by frequency bands*. Retrieved 2020-01-11, from <https://www.worldtimezone.com/gsm.html>