



Otimização do posicionamento de PMUs numa rede elétrica

TIAGO RAFAEL PINTO MONTEIRO

novembro de 2017

ISEP

Instituto Superior de Engenharia do Porto

Otimização do posicionamento de PMUs numa rede elétrica

Dissertação de Mestrado

Tiago Rafael Pinto Monteiro

Engenharia Eletrotécnica e de Computadores
Sistemas e Planeamento Industrial
2017

Orientadora:

Prof. Dr. Ana Viana

Co-Orientadora:

Dr. Xenia Klimentova (INESC TEC)

Agradecimentos

Em primeiro lugar gostaria de agradecer às minhas orientadoras, Prof. Dr. Ana Viana e Dr. Xenia Klimentova, por o serem em todo o sentido da palavra. Por todo o incentivo, partilha de conhecimento e disponibilidade que foram essenciais para o desenvolvimento deste trabalho.

Aos meus pais e irmãs, por todos os ensinamentos que fazem de mim quem sou, por acreditarem sempre no meu valor e me apoiarem incondicionalmente nos meus objetivos.

À minha namorada, por toda a paciência e compreensão durante a elaboração da presente dissertação. Pelo companheirismo e por ter sempre umas palavras de incentivo e carinho quando mais precisava.

Um as palavras de reconhecimento também para com os meus amigos, colegas de mestrado e professores que me acompanharam ao longo deste percurso académico.

Resumo

Phasor Measurement Units (PMUs) são dispositivos de monitorização de redes elétricas que são utilizados na obtenção e envio instantâneo de informações acerca da tensão do nó em que estão instalados e das correntes nos ramos a si ligados. Para além dessas medições diretas, a sua utilização permite, através de um conjunto de regras de análise de circuitos elétricos, obter informação relativa a outros nós que não aquele em que são instalados. Assim, através da definição de um conjunto de regras de observabilidade, um posicionamento diferente dos PMUs vai permitir monitorizar uma maior ou menor percentagem da rede. Isto conduz a um problema de optimização do posicionamento de PMUs em que se pretende decidir quais os nós em que devem ser colocados com o objetivo de minimizar o número de PMUs usados garantindo que a rede elétrica é totalmente monitorizada.

Nesta dissertação, o problema a ser tratado terá como base o problema referido anteriormente, considerando agora que há um investimento máximo que pode ser feito na aquisição de PMUs. Neste contexto o problema a estudar é o da maximização do número de nós monitorizados, utilizando os PMUs que o orçamento permite adquirir. O problema foi resolvido usando uma meta-heurística (*Simulated Annealing*), sendo os seus resultados validados através de testes computacionais num conjunto de instâncias existentes na literatura.

Palavras-chave

PMUs, problema de optimização, meta-heurísticas, *Simulated Annealing*.

Abstract

Phasor Measurement Units (PMUs) are monitoring devices designed to obtain and instantaneously send information about the voltage of the buses in which they are installed and the currents in the branches. In addition to these direct measurements, its use allows to obtain information on other buses besides the ones in which they are placed on. This is obtained through a set of basic rules that are used to perform the analysis of electrical circuits. Thus, by defining a set of observability rules, a different PMU positioning will allow the monitoring of a higher or smaller percentage of a network. This leads to the PMU positioning problem that aims to minimize the number of PMUs necessary to fully monitored a power systems network, by conveniently selecting the nodes where the PMUs should be positioned.

In this thesis, the problem to be addressed will be based on the previously mentioned one, considering now that there is a limit on the investment that can be made to acquire PMUs. Under such assumption we have a problem of maximizing the number of buses monitored using the PMUs that the budget allows to acquire.

The problem was solved with a metaheuristic (Simulated Annealing), and its results were validated through computational tests using a set on instances present in the literature.

Keywords

PMUs, optimization problem, metaheuristics, Simulated Annealing.

Conteúdo

1	Introdução	1
1.1	Enquadramento e Motivação	1
1.2	Objetivos	2
1.3	Estrutura do Relatório	2
2	Problema de posicionamento de PMUs: estado da arte	5
2.1	Contextualização	5
2.2	Descrição do problema	9
2.3	Modelo Matemático	13
2.4	Técnicas de Resolução	15
2.4.1	Programação Matemática	15
2.4.2	Meta-heurísticas	17
3	Metodologias desenvolvidas	29
3.1	Construção da solução inicial	30
3.2	Avaliação da qualidade das soluções	31
3.3	Operações de Vizinhança	33
3.3.1	Troca de uma posição	33
3.3.2	Operação Probabilística	33
3.4	Pesquisa Aleatória	35
3.5	Pesquisa Local	35
3.6	<i>Simulated Annealing</i>	36
4	Testes Computacionais e Análise de Resultados	39
4.1	Pesquisa Aleatória	41
4.2	Pesquisa Local	41
4.3	<i>Simulated Annealing</i>	45
5	Conclusão e Desenvolvimentos Futuros	49

A Instâncias de teste

57

Lista de Figuras

2.1	Evolução da População Mundial [32]	6
2.2	Rede Convencional vs Rede Inteligente [48]	7
2.3	Arquitetura de WAMS [25]	8
2.4	Diagrama de blocos de PMU [21]	8
2.5	Output PMU vs Output SCADA [52]	9
2.6	Cenários de aplicação das regras 2 (esquerda) e 3 (direita) (nós <i>zero-injection</i> representados por quadrados, os outros por círculos; verde para nós observados e branco para os que não são)	11
2.7	Exemplo de propagação das regras.	12
2.8	Exemplo de posicionamento de PMUs	12
2.9	(Meta-)heurísticas desenvolvidas desde 1947 até a 1996 [47]	18
2.10	Intensificação vs Diversificação [47]	18
2.11	Ótimos Local e Ótimo Global (adaptado de [51])	20
2.12	Comportamento das partículas com o decréscimo da temperatura [42]	21
2.13	Esquema de Arrefecimento [38]	23
2.14	Exemplo de cruzamento e mutação [5]	25
3.1	Modelo para a resolução do problema	30

Lista de Tabelas

3.1	Nós e ligações	34
4.1	Cenários e resultados obtidos através de [6]	40
4.2	Pesquisa Aleatória (PA)	42
4.3	Troca uma posição (PL-TP)	43
4.4	Operação Probabilística (PL-OP)	44
4.5	SA - Troca uma posição	46
4.6	SA - Operação Probabilística	47
4.7	SA - Troca de uma posição e Operação Probabilística	48
A.1	Instâncias Utilizadas	57

Lista de Acrónimos

ISEP	Instituto Superior de Engenharia do Porto
INESC TEC	Instituto de Engenharia de Sistemas e Computadores - Tecnologia e Ciência
PMU	<i>Phasor Measurement Units</i>
OPP	<i>Optimal PMU Placement</i>
IEEE	Institute of Electrical and Electronics Engineers
EMS	<i>Energy Management System</i>
WAMS	<i>Wide Area Monitoring System</i>
SCADA	Sistema de Supervisão e Aquisição de Dados
GPS	Sistema de Posicionamento Global
PDC	<i>Phasor Data Controller</i>
PQI	Programação Quadrática Inteira
AG	Algoritmo Genético
SA	<i>Simulated Annealing</i>
TS	<i>Tabu Search</i>
AIS	<i>Artificial Immune Systems</i>
GRASP	<i>Greedy Adaptive Search Procedure</i>
ACO	<i>Ant Colony Optimization</i>
PSO	<i>Particle Swarm Optimization</i>
HS	<i>Harmony Search</i>
ABC	<i>Artificial Bee Colony</i>

Capítulo 1

Introdução

1.1 Enquadramento e Motivação

O sistema elétrico tem vindo a evoluir em vários sentidos. Um dos fatores é a maior diversidade de formas de produzir energia com a introdução das fontes renováveis, o que aumenta a sua complexidade. A esse fator, acresce o facto de os mercados terem sido liberalizados, levando os operadores a terem de aumentar a sua eficiência de forma a serem mais competitivos. Acresce a questão da operacionalidade em segurança da rede elétrica que exige uma monitorização detalhada para ter a informação da sua condição e para proceder a operações de controlo. Estes fatores, impulsionaram a integração de tecnologia nas redes elétricas, sendo que, o problema que se vai tratar ao longo deste projeto, parte dessa necessidade como base.

Phasor Measurement Units (PMUs) são dispositivos de monitorização que são utilizados na obtenção e envio instantâneo de informação acerca do módulo e fase de tensão e corrente, tal como da hora e data da medição desses valores. Um PMU instalado num nó consegue monitorizar a tensão desse nó e correntes de todos, ou alguns, ramos a si ligados (dependendo do número de entradas do PMU). O sistema seria completamente e diretamente monitorizado se todos os nós possuíssem um PMU instalado. Mas, economicamente tal seria inviável devido ao alto custo dos mesmos e por nem todos os nós se localizarem em locais com as infraestruturas de comunicação necessárias à transmissão de informação por si capturada. Esta questão é ultrapassada se tivermos em conta um conjunto de regras, baseadas em leis básicas dos circuitos elétricos, que permitem que um PMU capte informação relativa a outros nós que não aquele em que são colocados. No entanto, a mesma quantidade de PMUs se posicionados em diferentes nós, permite que uma maior ou menor percentagem da rede seja observada. Deste facto resulta um problema de otimização que tem como objetivo minimizar o número de PMUs necessários para monitorizar uma rede

integralmente.

O problema, que implica definir a localização dos PMUs, é referido na literatura como “Posicionamento ótimo de PMUs” (*Optimal PMU Placement* - OPP - em inglês).

O trabalho a ser desenvolvido nesta dissertação terá como base o problema referido anteriormente, tendo agora em conta que há um investimento máximo que pode ser feito na compra de PMUs. Daqui resulta um problema de otimização diferente: o problema passa a ser de maximização e tem como objetivo maximizar o número de nós monitorizados, utilizando os PMUs que o orçamento permita adquirir.

Na literatura são propostas várias abordagens à resolução do problema base descrito acima, sendo de destacar a utilização de programação matemática e meta-heurísticas. Neste documento vão ser apresentadas algumas dessas abordagens, dando especial enfoque às meta-heurísticas, pois será através do desenvolvimento de uma meta-heurística (*Simulated Annealing*) que será proposta uma solução para problema aqui tratado.

A metodologia desenvolvida será validada por testes computacionais usando sistemas standard de teste: WSCC 9-bus; IEEE: 14-bus, 24-bus, 30-bus, 39-bus, 57-bus, 96-RTS, 118-bus, 300-bus; UIUC 150-bus; Texas 2000 [1].

1.2 Objetivos

O objetivo central desta dissertação é o estudo do problema OPP considerando restrições orçamentais e o desenvolvimento de uma meta-heurística eficiente para a sua resolução. Objetivos intermédios para que seja atingido o objetivo final são:

- Revisão do estado de arte do problema OPP, analisando diferentes propostas de modelação e estudando várias abordagens propostas, em particular as que consideram um limite orçamental;
- Proposta de uma meta-heurística que seja eficiente na resolução do problema;
- Estudo computacional da meta-heurística desenvolvida, comparando-se os resultados com os obtidos através de técnicas exatas.

1.3 Estrutura do Relatório

A dissertação é composta por cinco capítulos. No primeiro capítulo é feita uma introdução ao tema a ser desenvolvido ao longo do trabalho e são delineados os objetivos a atingir.

No capítulo 2 é feita uma breve descrição do problema, é apresentado o seu modelo matemático e são expostas várias metodologias utilizadas na resolução do mesmo.

No terceiro capítulo é explicado detalhadamente o trabalho desenvolvido para a resolução do problema.

No capítulo 4 são apresentados os resultados computacionais obtidos e é feita a sua análise crítica.

Por fim, no capítulo 5, são apresentadas as conclusões finais e propostas de trabalho futuro.

Capítulo 2

Problema de posicionamento de PMUs: estado da arte

Neste capítulo, serão apresentados os conceitos fundamentais para a compreensão do problema. Começa-se por uma contextualização e descrição do mesmo e apresenta-se um modelo matemático. Referem-se várias técnicas de otimização que podem ser utilizadas na sua resolução e, por fim, apresenta-se um conjunto de conceitos gerais sobre meta-heurísticas, abordagem utilizada neste trabalho.

2.1 Contextualização

A energia elétrica é um ingrediente fundamental na sociedade moderna, tendo o seu fornecimento um impacto direto no desenvolvimento social e económico de uma nação. A sua importância estende-se à noção de qualidade de vida existente no quotidiano das pessoas atualmente. Seja em lazer ou profissionalmente, existe uma total dependência de um fornecimento contínuo e económico de energia elétrica [18]. Como é possível observar na figura 2.1, a população tem vindo a aumentar e as projeções das Nações Unidas são de que atinja os 9.77 mil milhões de habitantes já em 2050. Ou seja, mais de 2 mil milhões de pessoas do que a população atual [49]. Esta projeção representa mais um desafio para o sector energético que, devido ao contexto macroeconómico, está a ser fortemente pressionado para a mudança das redes elétricas e para a forma como interage com os consumidores. A sustentabilidade ambiental, impõe a necessidade de novas formas de produção de energia que representem uma alternativa aos recursos fósseis de forma a reduzir a dependência externa e cumprir metas cada mais ambiciosas relativamente à redução de emissões de CO_2 . É necessário também tornar as redes mais fiáveis e com uma melhor qua-

lidade de fornecimento, tal como, a integração do consumidor com um papel mais ativo, seja na tomada de decisões com vista à otimização do seu consumo, seja na sua capacidade de microprodução [30].

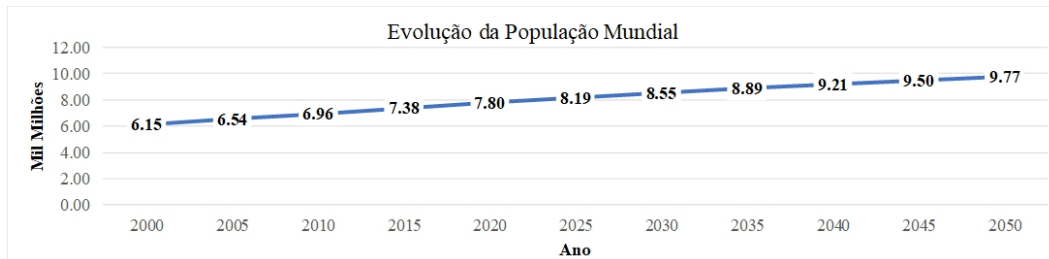


Figura 2.1: *Evolução da População Mundial [32]*

O passo a dar no sentido de apoiar o constante crescimento da economia global é a atualização da rede elétrica para uma rede inteligente (*smart grid*) [40]. A *smart grid* não possui uma definição universalmente aceite, podendo ser descrita através de termos mais simples ou de formas mais complexas.

A Plataforma Tecnológica Europeia define a *smart grid* como:

“A *smart grid* é uma rede elétrica que pode integrar de forma inteligente as ações de todos os utilizadores a si ligados (consumidores, produtores e os que combinam as duas) a fim de proporcionar de forma eficiente um fornecimento sustentável, económico e seguro.”[41]

Enquanto que o Departamento de Energia dos Estados Unidos define como:

“A *smart grid* regenera-se automaticamente, permite a participação ativa dos consumidores, opera de forma resiliente contra o ataque e desastres naturais, acomoda as operações de produção e armazenamento, possibilita a introdução de novos produtos, serviços e mercados, otimiza a utilização de recursos e opera de forma eficiente, fornece energia com qualidade para a economia digital.”[41]

Para uma melhor compreensão, apresenta-se a figura 2.2 que reflete dois tipos de rede. À esquerda pode-se observar um esquema de uma rede convencional, que possui produção centralizada e controlável, padrões de consumo estáveis e fluxos unidirecionais e previsíveis. Por sua vez, à direita, é possível observar uma rede inteligente, onde a produção é distribuída e intermitente, o balanço do consumo ou de produção é instável e os fluxos são bidirecionais e imprevisíveis [48].

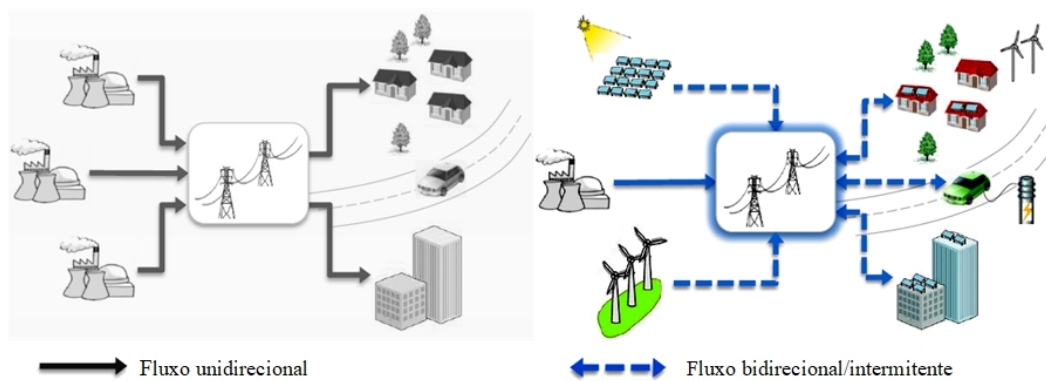


Figura 2.2: Rede Convencional vs Rede Inteligente [48]

A sua implementação significa um aumento na eficiência e fiabilidade da rede elétrica, pois torna-se possível fazer uma gestão informada e inteligente da produção, distribuição e consumo de eletricidade [40]. Apresenta, assim, uma solução para a manutenção da estabilidade da rede, que atualmente representa um enorme desafio devido às constantes oscilações entre oferta e procura de energia elétrica e, principalmente, devido à grande concentração da procura em períodos específicos (picos de procura) [11]. Facilita também a questão da introdução das fontes de energia renováveis na rede elétrica, seja por possuírem um custo mais baixo e não serem prejudiciais para o meio ambiente [40], ou por permitirem a integração da micro-produção, ou seja, a produção de energia pelo próprio consumidor utilizando equipamentos de pequena escala, como por exemplo painéis solares e micro-eólicas [10].

Para atingir os objetivos descritos nos parágrafos anteriores, tecnologias como as tecnologias de comunicação, de controlo de redes e de otimização, devem ser incorporadas no sistema de gestão da energia (EMS - *Energy Management System*). Uma parte fundamental desse sistema é um sistema de monitorização (WAMS- *Wide Area Monitoring System*). O WAMS é um sistema que permite monitorizar, operar e controlar redes elétricas numa área extensa, combinando a tecnologia dos aparelhos de medição com a tecnologia moderna de comunicação, possibilitando o acesso à informação de toda a rede através do posto de controlo [44]. A rede de comunicação, que é composta por uma rede de sensores, Wi-Fi, comunicação satélite, rede de telecomunicações, etc., é responsável pela aquisição de informação e transmissão de mensagens de controlo, especialmente se as fontes de energia renováveis estiverem ligadas à *smart grid*. O conhecimento da informação sobre o estado da rede permite que os operadores adaptem dinamicamente a sua produção e distribuição em tempo real [40].

Até recentemente, as medições disponibilizadas eram obtidas através da utilização

de sistemas SCADA (Sistemas de Supervisão e Aquisição de Dados), mas a utilização do sistema de posicionamento global (GPS) para a sincronização das medições efetuadas despoletou o desenvolvimento e utilização de PMUs [28].

Os PMUs foram introduzidos nos anos noventa, sendo utilizados num WAMS para a aquisição de dados. Através da sua utilização é possível obter, em tempo real, medições sincronizadas da tensão do nó em que se situa e das correntes dos ramos a ele ligados [28]. Na figura 2.3 é apresentada a arquitetura do WAMS. É possível observar a utilização de um sinal de GPS para a sincronização de medições efetuadas pelos PMUs e também que estes são agrupados num *Phasor Data Controller* (PDC), formando assim uma rede local de comunicação onde o PDC é o primeiro a receber as medições dos PMUs, procedendo de seguida ao envio dos dados para o posto de controlo da rede [25].

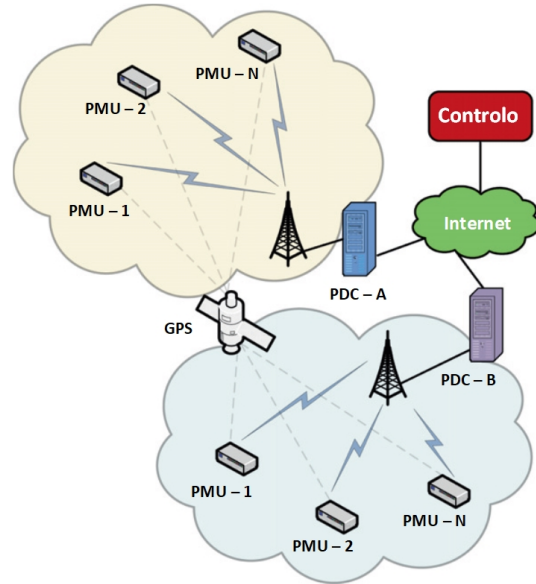


Figura 2.3: Arquitetura de WAMS [25]

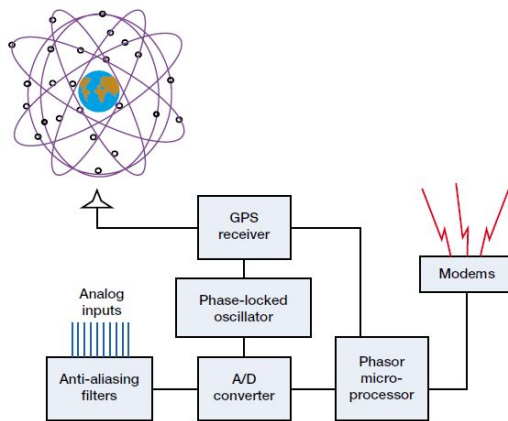


Figura 2.4: Diagrama de blocos de PMU [21]

acompanhando a evolução do *hardware*. Os primeiros PMUs tinham a capacidade de obter 12 amostras por ciclo da frequência nominal do sistema. Os mais modernos

Na figura 2.4 temos um diagrama de blocos baseado na configuração do primeiro PMU desenvolvido pela Virginia Tech. As entradas analógicas obtêm o sinal das tensões dos nós ou correntes dos ramos a si ligados para de seguida serem submetidos a filtros para evitar o efeito “*aliasing*”. Após esse pré-processamento o sinal é submetido a um conversor Analógico/Digital onde a amostragem é sincronizada com o sinal de um impulso por segundo obtido através do GPS. A taxa de amostragem tem vindo a evoluir ao longo dos anos,

já utilizam 96 e 128 amostras por ciclo. O microprocessador faz os cálculos finais de todas as correntes e tensões utilizando várias técnicas, como a Transformada Discreta de Fourier, para cumprir o formato definido pela IEEE standard 1344 [2]. Por fim, o output está pronto para ser transmitido [21, 15].

Na figura 2.5 é possível observar um exemplo da informação que é adquirida através da utilização de um PMU ou de um sistema SCADA. Nesta percebe-se a mais valia que os PMUs representam, pois recolhem mais informação (cerca de trinta amostragens por segundo) enquanto que o sistema SCADA (antigo) apenas recolhe uma amostragem a cada 4 segundos [52].

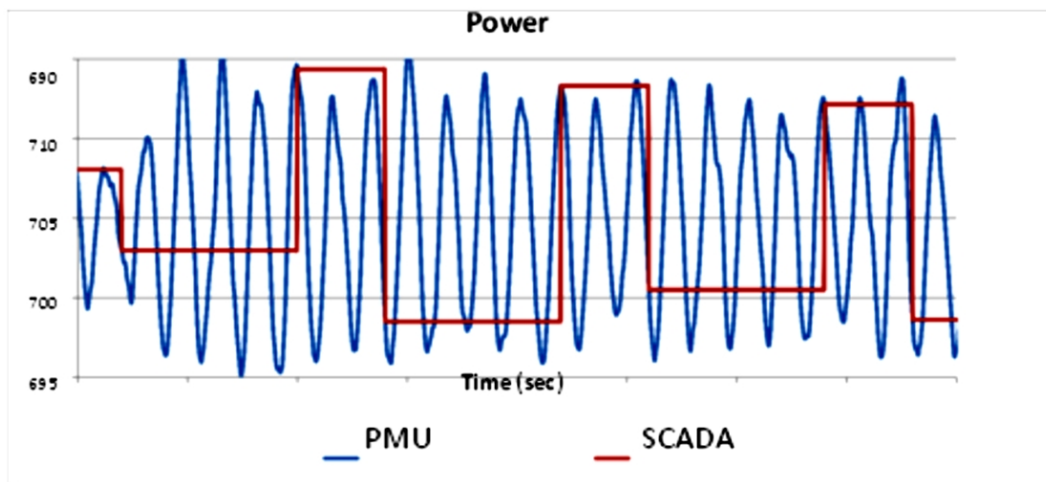


Figura 2.5: *Output PMU vs Output SCADA [52]*

Para que o aperfeiçoamento da monitorização, proteção e controlo das redes eléctricas possa ser alcançado, é necessário continuar com o aumento que se tem vindo a verificar na utilização destes dispositivos [28]. No entanto, a utilização de PMUs representa um grande custo, seja pelo seu valor de aquisição seja pelo custo da instalação de plataformas de comunicação, o que torna o problema de optimização de localização dos PMUs um importante desafio não só do ponto de vista de investigação mas também prático. Uma descrição breve do problema, bem como do modelo matemático associado são descritos nas secções seguintes.

2.2 Descrição do problema

Como referido anteriormente, de acordo com um conjunto de leis básicas de circuitos eléctricos, a mesma quantidade de equipamentos, posicionadas em diferentes nós, vai garantir uma maior ou menor percentagem de observação da rede [6]. O problema a ser investigado nesta dissertação, é uma variante de um OPP clássico que,

para além das suas propriedades, tem em conta que existe um orçamento disponível para a aquisição dos aparelhos. Ou seja, o objetivo não passa por obter a solução que minimize o número de PMUs necessários para a monitorização completa da rede, mas sim obter a solução que permita monitorizar a maior percentagem possível da rede com os PMUs que o orçamento permita adquirir, conhecendo à partida essa quantidade.

Para a resolução, importa compreender de que formas um nó pode ser observado. Diz-se que um nó é observado quando se conhece a sua tensão, enquanto que as linhas de transmissão (ramos), são observadas se as correntes forem conhecidas. Sendo que uma rede é totalmente observada se as tensões e correntes em todos os nós e ramos forem conhecidas [6].

Um nó pode ser observado diretamente se possuir um PMU ou, utilizando a lei de Ohm, se um nó a si diretamente ligado o possuir. Para nós que não possuam uma carga associada nem geração de energia (nós *zero-injection*) é possível obter informação adicional sobre a rede por aplicação da lei das correntes de Kirchhoff [6].

- **Lei de Ohm** permite calcular a corrente entre dois nós. Por exemplo, para obter a corrente I_{ij} que é a corrente entre os nós (i, j) recorre-se à fórmula $I_{ij} = \frac{V_i - V_j}{R_{ij}}$, onde R_{ij} representa a resistência da linha e V_i, V_j são as tensões (potenciais) nos nós i e j , respetivamente.
- **Lei das correntes de Kirchhoff** para um nó *zero-injection* i , a soma algébrica das correntes dos ramos ligados a esse nó é zero. Ou seja, $\sum_{j \in N(i)} I_{ij} = 0$, em que $N(i)$ representa os nós ligados ao nó i .

Com estas leis pode-se definir um conjunto de regras de observabilidade. Na nossa análise assumiremos que os PMUs possuem uma capacidade ilimitada, isto é, podem observar um número ilimitado de nós da rede. Ao todo serão consideradas três regras [6]:

Regra 1 um nó i é observado se o nó i ou um dos seus vizinhos tem um PMU;

Regra 2 um nó *zero-injection* i é observado se todos os seus vizinhos são observados;

Regra 3 se um nó *zero-injection* j e todos os seus vizinhos, exceto um (i) , são observados então o nó i é observado.

Na figura 2.6 é exemplificada a aplicação das regras 2 e 3, respetivamente. Para uma melhor compreensão da figura utilizou-se para representar os nós *zero-injection* quadrados, usando a cor verde para sinalizar os nós observados e branco para sinalizar os que não são.

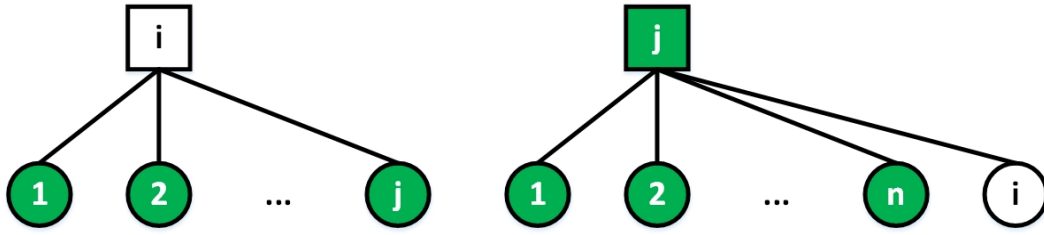


Figura 2.6: Cenários de aplicação das regras 2 (esquerda) e 3 (direita) (nós *zero-injection* representados por quadrados, os outros por círculos; verde para nós observados e branco para os que não são)

A compreensão das regras revela-se como um fator determinante para o sucesso deste projeto. Assim, será apresentado um exemplo de aplicação das regras de observabilidade. Para a análise será usada como base a rede IEEE 14-bus [1]. Esta possui 14 nós e 20 ramos, tendo um tamanho que permite uma análise simples e didática.

Como duas das regras incidem nas propriedades dos nós *zero-injection*, foi feita a adição de mais dois nas posições 3 e 10, sendo que a mesma já continha um na posição 7 (ver figura 2.7). Para facilitar a compreensão utilizaram-se várias cores: azul para os nós com PMUs, branco para representar os nós que não são monitorizados e verde para os nós que o são. Para além das cores, utilizam-se também formas, sendo que os quadrados representam os nós *zero-injection* e os círculos os restantes.

A rede resultante pode ser observada na figura 2.7, onde se exemplifica a utilização das várias regras na deteção dos nós que são observados. As regras tem natureza de propagação em cascata. Considere-se que os PMUs são instalados nos nós 4 e 6. Num primeiro passo preenche-se a verde os nós que são observados através da regra 1 (figura 2.7 (a)), ou seja, os nós na vizinhança dos nós que contêm PMU. Assim, o nó 4 é capaz de monitorizar os nós 2, 3, 5, 7 e 9 e o nó 6 os nós 5, 11, 12 e 13. Num segundo passo, através da regra 2, o nó 10 também é observado, pois a sua vizinhança é observada (nós 9 e 11), aplicando-se assim a regra de um nó *zero-injection* passar a ser observado se toda a sua vizinhança estiver observada (figura 2.7 (b)). Por último, a regra 3 refere que se um nó *zero-injection* e todos os seus vizinhos, exceto um, são observados, então esse nó passa a ser observado. Esse cenário acontece ao nó 7, que sendo um nó *zero-injection* tem na sua vizinhança um único nó não observado, o 8, que com a aplicação da regra passa a ser (figura 2.7 (c)). Estes nós, em conjunto, permitem monitorizar no total 12 dos 14 nós, sendo que os nós 1 e 14 não são observados através de nenhuma regra.

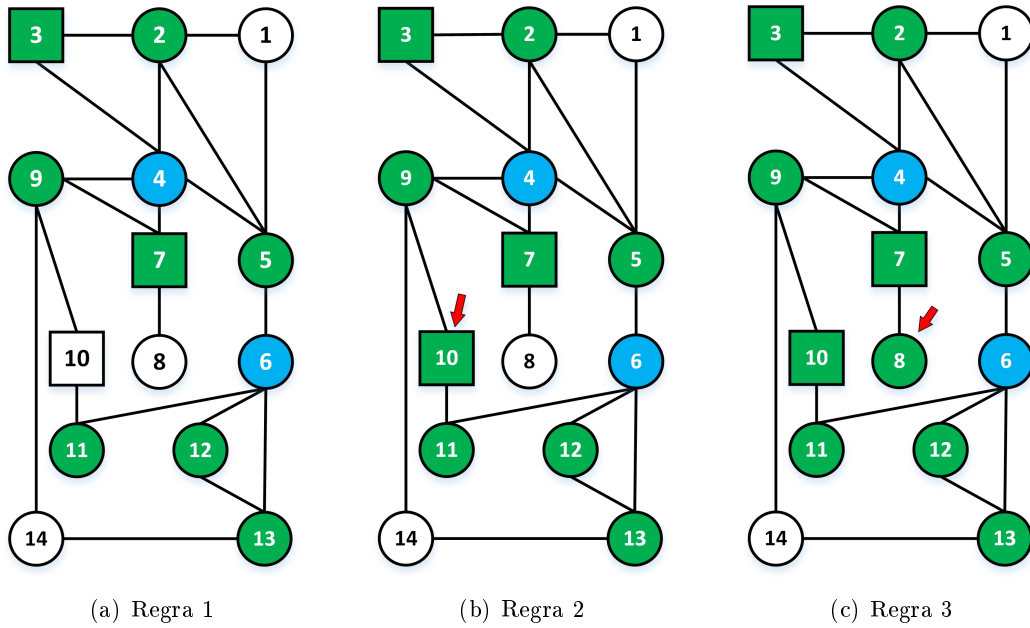


Figura 2.7: Exemplo de propagação das regras.

Com o intuito de mostrar que o posicionamento dos nós tem influência na quantidade de nós observados de uma rede, considere-se agora um cenário em que os PMUs se encontram instalados nos nós 4 e 5 utilizando a mesma rede que foi utilizada no exemplo anterior. Na figura 2.8 é apresentada essa hipótese e consequente resultado obtido. Para esta solução a rede é monitorizada em 9 dos seus 14 nós, menos três do que o cenário anterior.

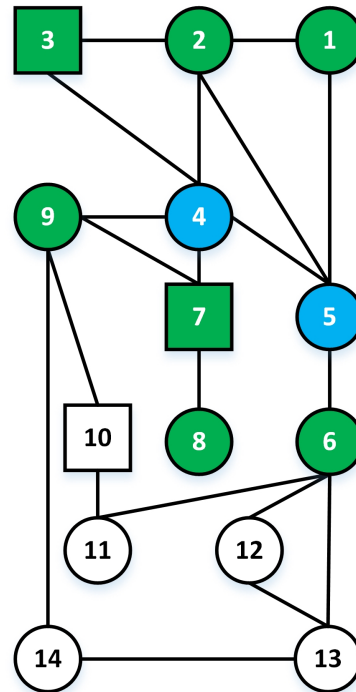


Figura 2.8: Exemplo de posicionamento de PMUs

2.3 Modelo Matemático

Nesta secção apresenta-se o modelo de Programação Matemática que reflete o problema em estudo. O modelo é adaptado de [6], de modo a considerar as restrições orçamentais. Para a sua definição foi considerada a notação seguinte:

Parâmetros

- $G = (V, E)$ - *graph* para representação da rede, onde V representa o conjunto de nós da rede e E o conjunto de ramos da rede. Assume-se que se $(i, j) \in E$ então $(j, i) \in E$;
- A vizinhança $N(i)$ para $i \in V$ é definida por nós adjacentes a i : $N(i) = \{j | (i, j) \in E\}$ e define $N[i] = N(i) \cup \{i\}$;
- $S \subseteq V$ conjunto de nós *zero-injection*;
- η - número máximo de iterações de propagação das regras 2 e 3: utilizando o índice $d \in D = \{0, 1, \dots, \eta\}$ como referência para cada iteração de propagação e considerando $D' = D \setminus \{\eta\}$. É suficiente definir $\eta = S$ para garantir que as regras 2 e 3 não são restringidas em termos de propagação [6];
- k - número de PMUs disponível (estipulado pelo orçamento);

Variáveis de decisão

- x_i - variável binária que toma valor 1 se um PMU é instalado no nó $i \in V$, 0 se não;
- w_{id} - variável binária que toma valor 1 se o nó $i \in V$ é observado no passo $d \in \{0, 1, \dots, \eta\}$, senão $w_{id} = 0$; variáveis $w_{i,\eta} = 1$ representam nós observados para um cenário de localização de PMUs dado;
- s_{id} - variável binária que toma valor 1 se a regra 2 é utilizada no passo d para observar o nó $i \in S$, 0 se não;
- r_{ijd} - variável binária que toma valor 1 se a regra 3 é utilizada no passo d para observar o nó $i \in N(j)$ com $j \in S$, 0 se não.

Para simplificação de leitura, no texto que se segue os índices serão omitidos quando se pretender fazer referência ao vetor todo dessa variável. Isto é, usar-se-á x para referir $(x_1, \dots, x_{|V|})$.

Com base nesta notação, constrói-se o modelo matemático seguinte:

$$\text{Obs} = \max_{x,w,s,r} \sum_{i \in V} w_{i,\eta} \quad (2.1)$$

$$\text{s.t.} \quad \sum_{i \in V} x_i \leq K, \quad \forall i \in V \quad (2.2)$$

$$x_i + \sum_{j \in N(i)} x_j \geq w_{i,0}, \quad \forall i \in V \quad (2.3)$$

$$w_{i,d+1} \geq w_{i,d}, \quad \forall i \in V, d \in D' \quad (2.4)$$

$$w_{i,d+1} - w_{i,d} \leq s_{i,d} + \sum_{j \in N(i) \cap S} r_{i,j,d}, \quad \forall i \in S, d \in D' \quad (2.5)$$

$$w_{i,d+1} - w_{i,d} \leq \sum_{j \in N(i) \cap S} r_{i,j,d}, \quad \forall i \in V \setminus S, d \in D' \quad (2.6)$$

$$s_{i,d} \leq w_{j,d}, \quad \forall i \in S, \forall j \in N(i), d \in D' \quad (2.7)$$

$$r_{i,j,d} \leq w_{j,d}, \quad \forall i \in V, \forall j \in N(i) \cap S, d \in D' \quad (2.8)$$

$$r_{i,j,d} \leq w_{k,d}, \quad \forall i \in V, \forall j \in N(i) \cap S, \quad (2.9)$$

$$\forall k \in N(j) \setminus \{i\}, d \in D'$$

- A função objetivo (2.1) visa maximizar o número de nós observados;
- As restrições (2.2) garantem que o número de PMUs não ultrapassa o máximo estipulado, enquanto que as restrições(2.3) fazem a modelação da regra 1;
- As restrições (2.4) implicam que se o nó i é observado no passo de propagação d , continua a ser observado para $d < \bar{d} \leq \eta$;
- As restrições (2.5) aplicam as regras 2 e 3 aos nós *zero-injection*;
- As restrições (2.6) aplicam a regra 3 aos nós não *zero-injection*;
- As restrições (2.7) declaram que a regra 2 pode ser aplicada ao nó i na iteração d se todos os nós em $N(i)$ forem observados na iteração d ;
- As restrições (2.8) garantem que se a regra 3 é utilizada para observar o nó i , então um dos seus nós *zero-injection* adjacentes, nó j , tem de ter sido observado;
- As restrições (2.9) replicam as restrições (2.8) para os restantes nós vizinhos do nó j .

2.4 Técnicas de Resolução

O problema de otimização estudado nesta dissertação, e em particular o problema base que o origina, tem sido resolvido através de técnicas exatas ou técnicas aproximadas (heurísticas). As técnicas exatas são assim designadas pois obtêm soluções ótimas. No entanto, há problemas em que a obtenção de soluções ótimas é impossível num período de tempo razoável. Nestes casos recorre-se a técnicas aproximadas que, apesar de não garantirem a obtenção de uma solução ótima, permitem obter uma “boa” solução num curto espaço de tempo.

Nesta secção serão referidas várias abordagens de resolução do problema através da utilização de programação matemática e meta-heurísticas. Tendo em conta que a abordagem proposta neste trabalho é uma meta-heurística, a secção contém uma descrição geral deste tipo de técnicas.

2.4.1 Programação Matemática

O termo “programação” era utilizado em 1940 para descrever o planeamento e agendamento das atividades de uma organização. Os “programadores” descobriram que podiam representar problemas do mundo real utilizando modelos matemáticos. A descoberta levou à representação da quantidade ou nível de cada atividade como uma variável cujo valor podia ser determinado (variáveis de decisão). Assim, podiam descrever matematicamente as restrições inerentes ao problema como um conjunto de igualdades ou desigualdades em função das variáveis de decisão. A solução que respeitasse todas as restrições era considerada como uma solução admissível. A experiência cedo revelou a dificuldade em modelar uma operação complexa recorrendo apenas a restrições. Se eram poucas restrições, soluções melhores podiam não ser escolhidas, enquanto que, no caso de serem demasiadas, soluções desejáveis não eram escolhidas ou então, no pior caso, não tinha solução. O sucesso da programação dependia de um caminho para contornar essa dificuldade. Para isso, para além das restrições, foi adicionado um objetivo. A função objetivo avalia a qualidade de uma solução. Assim, não importa a quantidade de soluções que respeitam as restrições. É importante sim encontrar aquela que, dentro dessas, maximiza ou minimiza a função objetivo. O termo “programação matemática” é utilizado para referir problemas de minimização ou maximização de uma função objetivo com várias variáveis, estando estas sujeitas a restrições [14].

Programação Linear Inteira

O trabalho descrito em [39] apresenta uma abordagem ao problema OPP com a restrição à capacidade de os PMUs de apenas um canal, ou seja, cada PMU só terá a capacidade de monitorizar um nó para além de si, tal como a respetiva linha de transmissão. Nesta abordagem é apresentada uma modelação às regras de observabilidade, sendo propostas duas metodologias baseadas em programação matemática: programação linear inteira e uma abordagem a dois níveis (*bilivel programming*).

Outro estudo no seguimento do referido anteriormente, é o apresentado em [6]. Nesse estudo são consideradas as duas abordagens referidas, sendo feita uma extensão do trabalho elaborado em [39]: adaptação do modelo de programação inteira para o problema OPP clássico, implementação de regras de observabilidade para todos os tipos de nós e consideração de capacidade ilimitada de canais para um PMU. São testadas duas variantes do problema: uma em que a capacidade de canais dos PMUs varia e outra em que está associado um custo variável para os PMUs. Os resultados obtidos através da implementação da solução proposta nesta dissertação serão comparados com os resultados obtidos através do algoritmo desenvolvido em [6].

O estudo apresentado em [4] pretende minimizar o número de PMUs a utilizar, considerando na sua resolução três cenários diferentes. Num procura-se a melhor solução sem considerar as particularidades dos nós *zero-injection*, noutra estas particularidades são consideradas e, finalmente, num terceiro cenário é prevista a hipótese de um PMU falhar (*loss of single PMU*).

O trabalho descrito em [19] apresenta uma formulação do problema. Como no artigo anterior, este também estuda os cenários com e sem nós *zero-injection*. A abordagem mostra que é possível modelar linearmente o problema para completa e incompleta observabilidade da rede de forma a ser resolvido recorrendo a programação linear inteira.

Em [45] é apresentado um método para alocar os PMUs de forma a que a rede seja completamente observada. Primeiro é proposto um algoritmo para decompor a rede em duas ou mais sub-redes. Após a decomposição, a localização ótima dos PMUs é determinada para cada sub-rede. O objetivo mantém-se como sendo a minimização do custo da instalação de PMUs (quantidade a utilizar).

Outras abordagens

Para além da programação linear inteira, outros métodos são propostos na literatura. Um desses métodos é a programação quadrática inteira (PQI). Um problema quadrático é um problema de otimização com uma função objetivo quadrática e

restrições lineares [34]. Na programação quadrática inteira, todas as variáveis tomam valores inteiros. O trabalho descrito em [7] apresenta um modelo de PQI que pretende minimizar o número de PMUs necessários para alcançar uma completa observabilidade da rede e maximizar a redundância das medições dos ramos. O modelo foi aplicado a vários sistemas de teste IEEE, tendo também sido estudada a hipótese de avaria numa linha de transmissão. Ou seja, em caso de uma avaria numa linha de transmissão, a rede continuaria totalmente observável.

2.4.2 Meta-heurísticas

O termo meta-heurística surge da junção da palavra heurística, que advém da palavra grega *heuriskein*, cujo significado é a arte de descobrir novas estratégias para a resolução de problemas, com o prefixo *meta*, também uma palavra grega, com o significado de metodologia superior [47]. Este termo foi introduzido por Glover em 1986 [16], designando uma estratégia mestra que guia e modifica outras heurísticas para produzir soluções além das que são normalmente geradas numa pesquisa local [8]. Antes, por volta dos anos 70, já tinham sido introduzidas as primeiras heurísticas construtivas para problemas combinatórios em que a utilização de técnicas exatas não se mostrava eficiente. Apesar do termo meta-heurística ter sido apresentado em 1986 antes estas já existiam, sendo conhecidas por “heurísticas modernas” [43].

Desde aí um longo caminho já foi percorrido, tendo especial enfoque as últimas três décadas, onde o desenvolvimento de meta-heurísticas foi particularmente intenso. Os Algoritmos Genéticos (AG) foram desenvolvidos por Holland, inspirados na seleção natural e evolução das espécies [43, 22]. Nos anos 80 várias meta-heurísticas são propostas, sendo que na primeira metade da década (1982) é proposto por Kirkpatrick et al o *Simulated Annealing* (SA), inspirado em sistemas físicos capazes de evoluir para um estado de energia mínima [26, 38], e em 1986 é proposto o *Tabu Search* (TS), inspirado no conceito de memória [17, 43]. Em 1986 é proposto o *Artificial Immune Systems* (AIS) por Farmer et al. [12] e no final da década o *Greedy Adaptive Search Procedure* (GRASP) [13]. Nos anos 90 são propostas várias técnicas de otimização evolutiva, como a *Ant Colony Optimization* (ACO) em 1992 por Dorigo, inspirada no comportamento social das formigas [43, 9] e o *Particle Swarm Optimization* (PSO), inspirado na partilha de informação e colaboração entre indivíduos através de simulação do seu comportamento social [24]. Em 2001 é proposto o *Harmony Search* (HS), inspirado na improvisação musical, e em 2005 é proposto o *Artificial Bee Colony* (ABC), inspirado em colónias de abelhas [43]. É possível analisar na figura 2.9, as (meta-)heurísticas desenvolvidas de 1947 até 1996.

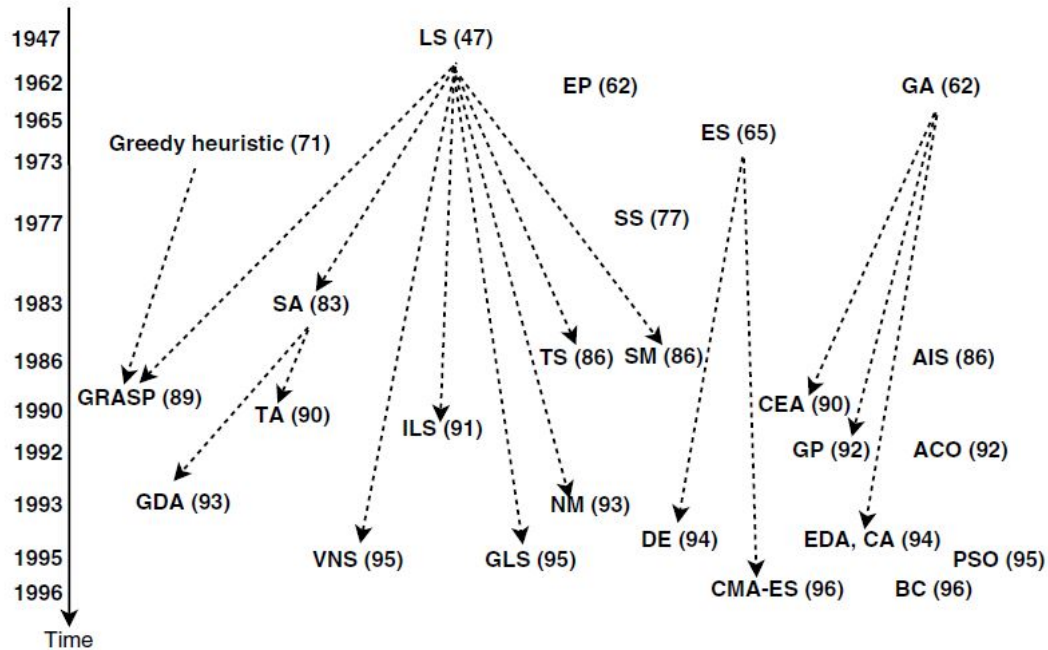


Figura 2.9: (Meta-)heurísticas desenvolvidas desde 1947 até a 1996 [47]

Importa compreender que na elaboração de uma meta-heurística, dois critérios devem ser tidos em conta: a exploração do espaço de soluções (diversificação) e a exploração das melhores soluções encontradas (intensificação) (figura 2.10). Na intensificação, as regiões do espaço de soluções promissoras são exploradas de forma mais metódica, com o objetivo de se encontrarem as melhores. Enquanto que, na diversificação as regiões não exploradas devem ser visitadas para garantir que todas as regiões do espaço de soluções estão uniformemente exploradas e que a procura não é limitada a um dado número de regiões. A pesquisa local utiliza sempre intensificação, enquanto que a pesquisa aleatória utiliza sempre diversificação. As meta-heurísticas tentam combinar estas duas, utilizando fases de intensificação e fases de diversificação [47].

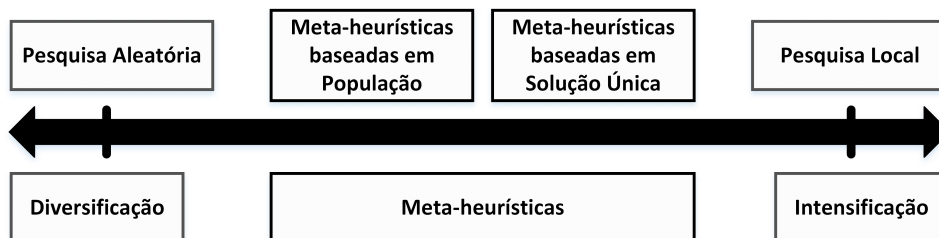


Figura 2.10: Intensificação vs Diversificação [47]

Pesquisa Local

Os métodos de pesquisa local são técnicas de otimização que se baseiam no método de otimização mais antigo - tentativa e erro - só que o fazem de uma forma sistemática. Para a sistematização da pesquisa são utilizadas estruturas de vizinhança. É importante perceber o conceito de vizinhança pois esta tem um papel fundamental na pesquisa local. A vizinhança de uma solução é um conjunto de soluções admissíveis, de algum modo idênticas com a solução em causa. Isto é, soluções separadas por um movimento elementar que pode ser manipulado para se transformarem umas nas outras.

No algoritmo 1 descrevem-se os passos principais da pesquisa local para um problema de minimização:

$$\min_s f(s), s \in S.$$

Começa por se obter uma solução inicial admissível do problema (s_0), geralmente de forma aleatória, para a qual vai ser feita uma pesquisa na sua vizinhança ($V(s_i)$) por soluções com melhor desempenho, com o intuito de substituir a atual por uma melhor. Se uma solução vizinha, s_j , for melhor do que a atual, substitui-a (passo 5). Caso contrário atualiza-se o espaço de vizinhança (passo 7) e repete-se o procedimento. Esse processo é repetido até não ser possível encontrar uma solução com melhor desempenho na atual vizinhança.

Algoritmo 1: Pesquisa Local [8]

Entrada: Informações do problema: $f(s)$, S
Saída: Melhor solução encontrada: s^*

- 1 Gerar uma solução inicial admissível $s_0 \in S$;
- 2 Solução corrente $s_i = s_0$;
- 3 Considerar um $s_j \in V(s_i)$;
- 4 **se** $f(s_j) < f(s_i)$ **então**
- 5 | $s_i = s_j$;
- 6 **senão**
- 7 | $V(s_i) = V(s_i) - s_j$;
- 8 | **se** $V(s_i) \neq \emptyset$ **então**
- 9 | Voltar a 3;
- 10 | **fim**
- 11 **fim**
- 12 Solução ótima local: $s^* = s_i$.

Ótimo Local e Ótimo Global

A técnicas de Pesquisa Local, tal como já foi referido anteriormente, procuram uma solução com melhor desempenho nas imediações da solução atual, não tendo uma visão de todo o espaço de soluções. Ora, esse fator pode conduzir o algoritmo para um ótimo local. Um ótimo local é uma solução que tem o melhor desempenho na sua vizinhança. Por outro lado, um ótimo global é a solução que tem o melhor desempenho em todo o espaço de soluções [43].

O cenário descrito é exemplificado na figura 2.11. O exemplo retrata um problema de maximização da função objetivo, tendo presente um ótimo global (ponto B) e dois ótimos locais (pontos A e C). Através das setas e o seu sentido é possível verificar que se uma solução estiver colocada na vizinhança de um ótimo local (simbolizado pelas setas vermelhas) vai ser sempre conduzida para o mesmo. Isto acontece pois a técnica de pesquisa local não “aceita” soluções piores, fazendo assim o percurso dado pelas setas. Assim, só se obteria o ótimo global caso a solução inicial se situasse na sua vizinhança (simbolizado pelas setas verdes).

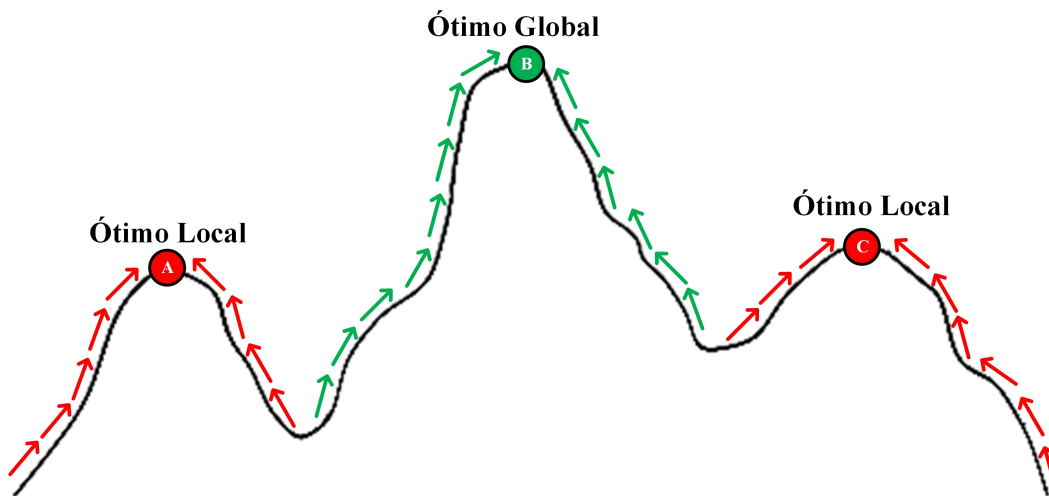


Figura 2.11: Ótimos Local e Ótimo Global (adaptado de [51])

A fim de tentar evitar esse cenário surgiram várias meta-heurísticas inspiradas na pesquisa local, mas com a inserção de estratégias para ultrapassar ótimos locais. Duas dessas meta-heurísticas são o *Simulated Annealing* e o *Tabu Search*, apresentadas a seguir.

Simulated Annealing

A meta-heurística *Simulated Annealing* (por vezes traduzida por Pesquisa por Arrefecimento Simulado) tem como inspiração a termodinâmica e a metalurgia. O termo “annealing” define uma estratégia que controla a temperatura de tal forma que pode ser aproximado um estado ótimo. Quando um metal em fusão é arrefecido de forma suficientemente lenta tende a solidificar num estado de energia mínima. O mesmo princípio resulta para o SA: no início quase todas as mudanças são aceites, permitindo explorar o espaço de soluções. Então, gradualmente, a temperatura é diminuída, o que significa que começa a ser mais seletivo na aceitação de novas soluções. Por fim, só mudanças que melhorem a solução é que são aceites [38].

Esta analogia pode ser observada através da figura 2.12, onde as partículas do metal tendem a ser mais seletivas quanto à sua posição quando se aproximam do seu estado de energia mínimo (temperatura diminui).

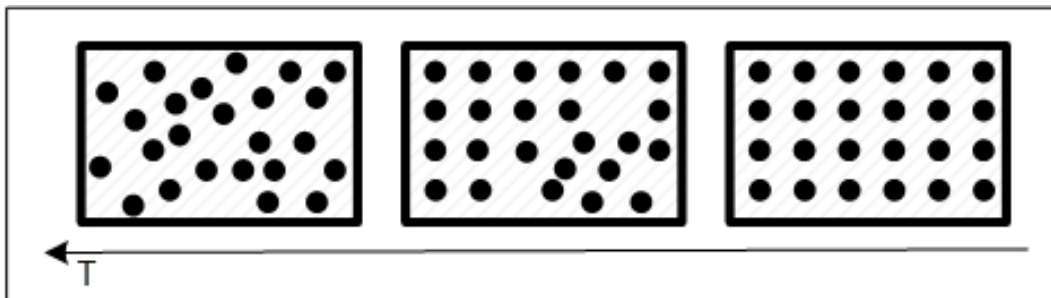


Figura 2.12: Comportamento das partículas com o decréscimo da temperatura [42]

No algoritmo 2 apresenta-se a estrutura básica do SA. O primeiro passo é a seleção de uma solução inicial, normalmente de forma aleatória. De seguida, e enquanto o critério de paragem não se verificar, os passos a dar consistem na escolha de uma solução na vizinhança da solução atual, $V(s)$ (passo 5), que será aceite como a solução atual caso seja melhor. Caso contrário, será aceite com uma probabilidade que será dada por uma função decrescente com a taxa de deterioração da solução, e crescente com o parâmetro designado temperatura (passo 14). Este último é gradualmente reduzido com o objetivo de o algoritmo ser mais seletivo na aceitação de novas soluções [8].

A estratégia que a meta-heurística *Simulated Annealing* apresenta para “escapar” aos ótimos locais é apresentada a partir da linha 11 até à linha 15. A estratégia, em caso de a solução de vizinhança ser pior, passa por escolher de forma aleatória um número entre 0 e 1 e, caso este seja menor que o valor obtido pela distribuição de probabilidade escolhida, então a solução é aceite.

Algoritmo 2: *Simulated Annealing* [8]

Entrada: Informações do problema: $f(s)$, S
Saída: Melhor solução encontrada: s^*

- 1 Calcular uma solução inicial admissível $s \in S$;
- 2 Inicializar a melhor solução encontrada $s^* = s$;
- 3 Inicializar a temperatura T ;
- 4 **enquanto** não se cumprir o critério de paragem **faça**
- 5 Selecionar aleatoriamente $s' \in V(s)$;
- 6 **se** $f(s') < f(s)$ **então**
- 7 $s = s'$;
- 8 **se** $f(s) < f(s^*)$ **então**
- 9 $s^* = s$;
- 10 **fim**
- 11 **senão**
- 12 Selecionar aleatoriamente um número $p \in [0, 1]$;
- 13 **se** $p \leq e^{-\frac{f(s')-f(s)}{T}}$ **então**
- 14 $s = s'$;
- 15 **fim**
- 16 **fim**
- 17 Atualizar a temperatura T ;
- 18 **fim**

De forma a tornar o algoritmo operacional para o problema em questão, é necessário tomar algumas decisões que terão influência na qualidade das soluções obtidas. Cada problema é diferente, e os parâmetros devem ser adaptados ao mesmo. Assim as decisões a tomar são [38]:

- **Escolha da probabilidade de aceitação $p(n)$.**

Por analogia com a termodinâmica é geralmente escolhida uma distribuição Boltzman:

$$p(n) = e^{-\frac{1}{T(n)}\Delta F_n}$$

Onde $\Delta F_n = f(s') - f(s_n)$ e T_n a temperatura no passo n definido a seguir.

- **Escolha de um esquema de arrefecimento.**

Um esquema de arrefecimento clássico (figura 2.13) consiste em partir de uma temperatura inicial T_0 , e decrescer o seu valor a cada número constante de passos L através do produto por um fator fixo α , $0 < \alpha < 1$, para diminuir ou

excluir a escolha de soluções piores na parte final da pesquisa. Assim, depois de kL passos, a temperatura é $T(kl) = T_k = \alpha^k T_0$.

Decisões fundamentais neste passo são a escolha de uma temperatura inicial T_0 , do valor de α e o número de passos L . Estes devem permitir uma maior permeabilidade à escolha de soluções piores no início, não comprometendo a evolução do algoritmo.

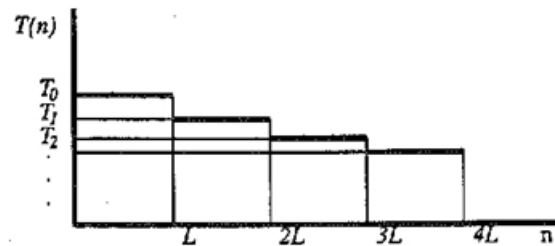


Figura 2.13: Esquema de Arrefecimento [38]

- **Escolha de critério de paragem.**

Existem duas variantes que são normalmente utilizadas. Uma corresponde à definição de uma temperatura mínima em que o algoritmo termina. Outra é o número de movimentos aceites num determinado número de iterações sucessivas ser inferior a um limiar estabelecido.

Tabu Search

Tabu Search (TS) é outra técnica baseada em pesquisa local, também desenvolvida para evitar ótimos locais sendo proposta por Glover em 1986 [17]. Ao contrário do SA, é uma meta-heurística dotada de memória. Ou seja, é uma meta-heurística que utiliza informação sobre as soluções obtidas anteriormente com o objetivo de orientar o processo de pesquisa futura. Esta é inspirada no processo de memória humana através de uma lista de soluções ou movimentos proibidos, que vão permitir diversificar a pesquisa, orientando-a para espaços não explorados, uma vez que não é possível voltar a uma solução que foi recentemente visitada [43].

A meta-heurística *Tabu Search* utiliza na sua configuração mais simples uma lista, *lista tabu*, de *atributos* dos últimos movimentos realizados para evitar os ótimos locais. O objetivo desta lista consiste em proibir que soluções avaliadas recentemente sejam novamente consideradas, forçando assim a diversificação da pesquisa, pois esta é orientada para zonas do espaço de soluções ainda não exploradas. Nas versões básicas de TS, em alternativa à pesquisa numa vizinhança pode ser utilizada antes uma pesquisa numa sub-vizinhança SV, isto é, num subconjunto da vizinhança, que terá

uma dimensão pré-definida e será constituída por elementos retirados aleatoriamente da vizinhança [8].

O algoritmo 3 descreve o TS. Em primeiro lugar, como num algoritmo de pesquisa local clássico, é necessário calcular uma solução admissível. O passo a ser dado de seguida é inicializar a lista tabu. Neste aspeto importa compreender que o tamanho da lista vai ter influência na forma como a pesquisa no espaço de soluções é feita. Caso o tamanho da lista seja maior esta vai fazer pesquisas mais diversificadas. Caso o seu tamanho seja mais reduzido a pesquisa será feita de forma mais intensa. Existe a possibilidade de utilizar um tamanho de lista dinâmico, alterando o seu comprimento para ir ao encontro do objetivo que se queira atingir. Feitas as inicializações, de seguida e enquanto não se verificar o critério de paragem, é feita a escolha de uma solução (s') entre as soluções (s'') pertencentes à sub-vizinhança de s que não pertença à lista tabu ou então que satisfaça o critério de aspiração (passos 5 e 6). O critério de aspiração define um patamar de desempenho para qual movimentos que estejam proibidos pela lista tabu possam ser considerados, evitando assim que soluções que, por exemplo, sejam melhores do que a solução atual, sejam proibidas. De seguida verifica-se se esta solução tem melhor desempenho que a atual. Caso tenha, é inserido o movimento na lista tabu. Geralmente caso esta esteja completa é removido o primeiro movimento [8].

Algoritmo 3: *Tabu Search* [8]

Entrada: Informações do problema: $f(s)$, S
Saída: Melhor solução encontrada: s^*

- 1 Calcular uma solução inicial admissível $s \in S$;
- 2 Inicializar a melhor solução encontrada $s^* = s$;
- 3 Inicializar a lista tabu $TL = \{\}$;
- 4 **enquanto** *não se cumprir o critério de paragem* **faça**
- 5 Determinar $s' \in SV(s)$ com $f(s') < f(s'')$, para todo o $s'' \in SV(s)$;
- 6 $((s, s')$ *não faz parte de lista tabu ou* s *satisfaz o critério de aspiração*);
- 7 **se** $f(s') < f(s)$ **então**
- 8 Inserir atributos de (s, s') em TL , removendo o primeiro elemento se TL estiver completa;
- 9 **se** $f(s') < f(s^*)$ **então**
- 10 $s^* = s'$;
- 11 **fim**
- 12 **fim**
- 13 $s = s'$;
- 14 **fim**

Algoritmos Genéticos

Após o estudo de duas meta-heurísticas baseadas em pesquisa local, apresenta-se a meta-heurística Algoritmos Genéticos baseada em populações/recombinação.

O termo Algoritmo Genético foi introduzido por Holland em 1972 [22], tendo por base a evolução genética e a seleção natural (Teoria de Evolução de Charles Darwin). Essa inspiração traduz-se na utilização de operadores de seleção, cruzamento e mutação com o objetivo de fazer evoluir a população. Os operadores de seleção representam a sobrevivência dos seres com melhor aptidão, o cruzamento permite a mistura, recombinação e variação das características hereditárias dos pais de modo a formar descendentes com novas potencialidades. A mutação permite diversificar as características das soluções [8].

Na figura 2.14 apresenta-se um exemplo de aplicação destes operadores a dois elementos da população (A e B). Numa primeira fase é feito o cruzamento dos elementos, ou seja, uma recombinação de características, optando-se neste caso por uma característica do elemento A e duas características do elemento B. De seguida é feita a operação de mutação, onde é feita a alteração de uma característica de forma aleatória.

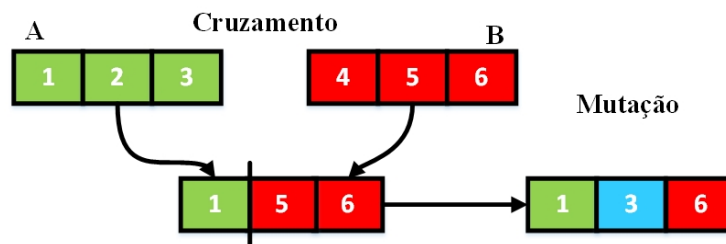


Figura 2.14: Exemplo de cruzamento e mutação [5]

O algoritmo 4 reflete a estrutura de um AG. Parte de uma população inicial de indivíduos (soluções) e aplica operadores genéticos em cada reprodução. Cada indivíduo tem um grau de aptidão (valor associado à função objetivo). Aos indivíduos mais aptos é dada a oportunidade de reprodução (geração de novas soluções) através da troca de informação genética (características das soluções), num processo de cruzamento (recombinação) com outros indivíduos altamente aptos. Desta forma, são produzidas novas gerações, que partilham características dos pais. Após o cruzamento, é frequentemente aplicada uma mutação, alterando aleatoriamente alguns genes. A nova geração pode substituir totalmente a população anterior (abordagem geracional) ou apenas os indivíduos menos aptos (abordagem estacionária). Este ciclo de avaliação-seleção-reprodução é feito até que se verifique o critério de paragem [8].

Algoritmo 4: *Algoritmo Genético [8]*

Entrada: Informações do problema: $f(s), S$
Saída: Melhor solução encontrada: s^*

- 1 Construir uma população inicial P , com soluções $s \in S$;
- 2 Determinar em P qual a melhor solução s^* ;
- 3 **enquanto** não se cumprir o critério de paragem **faça**
- 4 Selecionar, com reposição, M pares de indivíduos de P , sendo a probabilidade de seleção crescente com a qualidade do indivíduo;
- 5 **para** cada par **faça**
- 6 Selecionar aleatoriamente um número $p \in [0, 1]$;
- 7 **se** $p <$ probabilidade de cruzamento **então**
- 8 | Gerar um par de filhos usando o operador de cruzamento;
- 9 **senão**
- 10 | Gerar um par de filhos idênticos aos pais;
- 11 **fim**
- 12 **fim**
- 13 **para** cada filho s **faça**
- 14 Aplicar com probabilidade μ o operador de mutação;
- 15 **se** $f(s) < f(s^*)$ **então**
- 16 | $s^* = s$;
- 17 **fim**
- 18 **fim**
- 19 Selecionar, sem reposição, $2M$ pares de indivíduos de P , com probabilidade de seleção decrescente com a qualidade do indivíduo;
- 20 Substituir os $2M$ indivíduos pelos filhos;
- 21 **fim**

Concluindo, a abordagem sugerida por esta meta-heurística baseia-se em quatro fases principais, sendo estas [27]:

- **Seleção:** são selecionados os indivíduos que irão fazer parte da reprodução;
- **Reprodução:** é feita a aplicação de operadores de seleção de modo a que sejam criados novos indivíduos (cruzamento e mutação);
- **Avaliação da aptidão dos indivíduos:** é feito o cálculo do valor da função objetivo para cada indivíduo;
- **Substituição:** é feita a seleção dos novos indivíduos com maior aptidão, substituindo os que possuem pior aptidão.

Utilização de meta-heurísticas na resolução do problema OPP

Têm sido apresentadas na literatura várias propostas baseadas na utilização de meta-heurísticas para a resolução do problema OPP clássico (ou variantes do mesmo). Em [28] e [33] é apresentada uma revisão de literatura de abordagens utilizadas, com particular enfoque em estudos desenvolvidos recorrendo a meta-heurísticas.

O trabalho descrito em [23] apresenta uma abordagem que utiliza o SA para a minimização do número de PMUs necessários para monitorizar a rede na sua totalidade. A qualidade das medições efetuadas pelos PMUs é também tida em conta, através do cálculo de um parâmetro (sensibilidade) para cada nó. Este parâmetro é considerado durante a pesquisa de novas soluções (pretende-se posicionar os PMUs nos nós com valores de sensibilidade maiores). Nesta abordagem não são apresentados os valores utilizados na parametrização do SA, sendo referido apenas a operação de vizinhança utilizada: mudança da localização de um PMU a cada iteração. É referido também que os valores para a temperatura inicial e final são obtidos através de várias tentativas. Em [35] é apresentado um estudo que tem em conta a rede não ser totalmente observada (observabilidade incompleta). Baseia-se na utilização de árvores geradoras de custo mínimo e usa pesquisa em árvore para determinar o posicionamento dos PMUs. O SA é usado para resolver o problema OPP com restrições nas instalações de comunicação. Tal como no estudo referido anteriormente, a forma utilizada para obter soluções na vizinhança é alterar uma posição da solução a ter em conta. Quanto ao esquema de arrefecimento utilizado, não são especificados os valores para a temperatura inicial (refere que foi obtido através de experiências) e temperatura final, sendo proposto terminar quando atingir 50 iterações em que a temperatura é atualizada. É também definido o intervalo de iterações (L) em que a temperatura deve ser atualizada, utilizando para esse efeito a fórmula $L = k|S|$ onde $|S|$ representa a quantidade de PMUs e k uma constante que, normalmente toma o valor 5. Para a diminuição da temperatura é utilizado $\alpha = 0.8$.

O estudo apresentado em [37] utiliza como metodologia para a minimização da quantidade de PMUs o TS. Tal como nos estudos referidos anteriormente, a forma utilizada na procura de soluções na vizinhança da solução atual é a mudança de posição de um PMU. Para a parametrização do TS é definido um intervalo para o tamanho que a lista tabu pode tomar, sendo contido entre um valor mínimo e máximo variável de acordo com as soluções em si contidas e a vizinhança da solução atual.

O mesmo problema (minimização do número de PMUs necessários para monitorizar a rede na sua totalidade) foi resolvido em [29] e [31] usando um algoritmo genético. No primeiro, as soluções são selecionadas através de definição de probabili-

dades baseadas no valor da função objetivo para cada uma. Após a seleção é feito o cruzamento de duas soluções (o ponto de cruzamento é selecionado aleatoriamente) sendo aplicado o operador de mutação num valor de forma aleatória. Quanto ao segundo estudo, não são apresentadas as estratégias utilizadas para a seleção, cruzamento e mutação. Outra proposta de resolução com o mesmo objetivo é apresentada em [3] através da utilização de um *Immunity Genetic Algorithm*.

Outras meta-heurísticas foram propostas para a resolução do problema, sendo por exemplo utilizado o PSO em [20] e [46] e ACO em [50]. Mais abordagens podem ser consultadas nos artigos [28] e [33] referidos anteriormente.

Capítulo 3

Metodologias desenvolvidas

Neste capítulo serão apresentadas as metodologias propostas para a resolução do problema em estudo bem como algum trabalho preliminar. Numa primeira fase o problema foi resolvido por enumeração utilizando para o efeito um algoritmo programado na linguagem de programação Python. O objetivo da abordagem foi meramente pedagógico e permitiu perceber a real importância da utilização de técnicas de otimização neste tipo de problema: caso o processo de obtenção de todas as combinações não se revelasse demasiado moroso não seria necessário recorrer a técnicas de otimização para a sua resolução. No entanto, quando se testou o caso de uma rede com 57 nós e 3 PMUs ao fim de quarenta e cinco minutos terminou-se o processo, sem garantia de que a solução ótima era encontrada.

Na sequência desta experiência, a primeira metodologia proposta para o problema tem como base um algoritmo de pesquisa local, sendo esta elaborada essencialmente para melhor compreender as operações de vizinhança desenvolvidas e a importância de uma estratégia que procure evitar os ótimos locais. Por fim, a segunda metodologia é desenvolvida através da utilização de uma meta-heurística, tendo-se optado pelo *Simulated Annealing*.

Cada um dos algoritmos referidos anteriormente requer a especificação e implementação de outros algoritmos que permitam obter uma solução inicial admissível, que efetuem as operações de vizinhança e, por fim, que avalie a qualidade das soluções. Neste trabalho, como ilustrado na figura 3.1, foram desenvolvidos um algoritmo para a obtenção da solução inicial, dois para a realização de operações de vizinhança e um para avaliar o desempenho das soluções (quantidade de nós observados). Cada um dos algoritmos desenvolvidos é referido em mais detalhe nas secções seguintes.

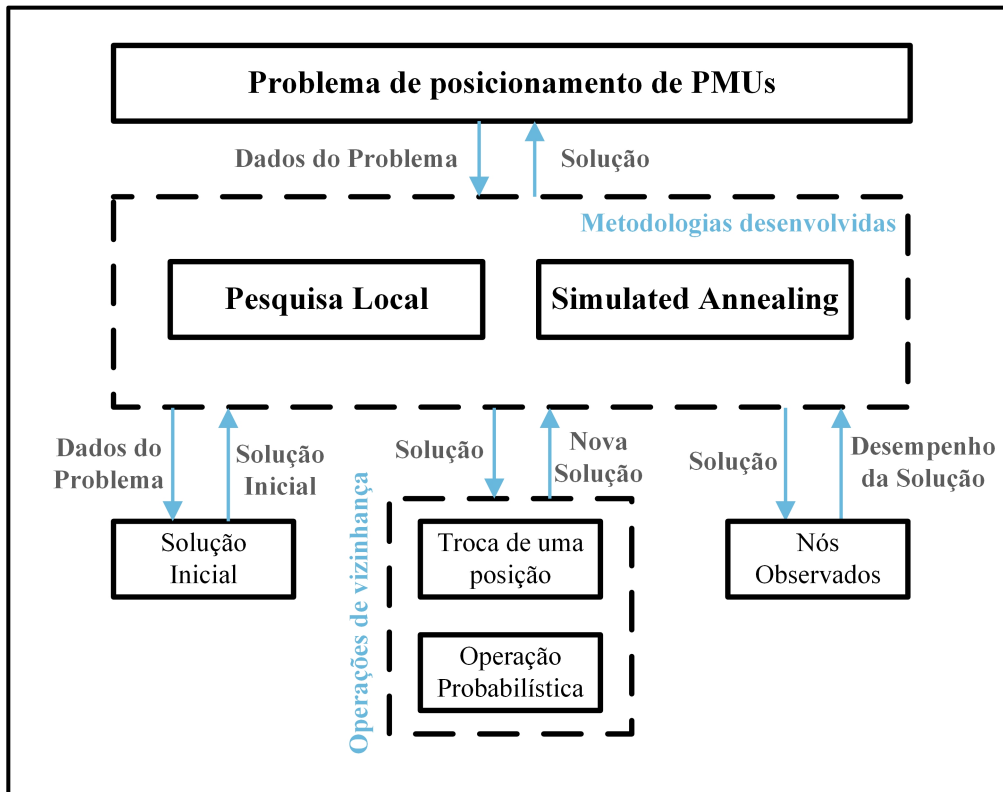


Figura 3.1: Modelo para a resolução do problema

A título de curiosidade e de forma a verificar a diferença nos resultados obtidos, para além dos algoritmos referidos anteriormente, foi desenvolvido um algoritmo para uma pesquisa feita de forma totalmente aleatória. Este funcionará nos mesmos moldes que a pesquisa local, sendo diferente na forma em que é feita a operação de vizinhança, ou seja, não realizará uma operação de vizinhança à solução atual, sendo proposta de raiz uma nova solução a cada iteração.

Para a elaboração das várias metodologias foi utilizada a linguagem de programação C recorrendo ao *software* Visual Studio 2017. Todos os algoritmos foram construídos de forma modular, permitindo assim a utilização dos mesmos nas duas metodologias propostas.

3.1 Construção da solução inicial

O primeiro passo a ter em consideração na conceção de um algoritmo, seja de pesquisa local ou de uma meta-heurística, é de construção de uma solução inicial (geralmente) admissível.

O algoritmo implementado neste estudo parte de uma solução inicial obtida de

forma totalmente aleatória. Isto é, os nós onde os PMUs são colocados são escolhidos de forma totalmente aleatória. Não é claro que fazer um esforço para obter uma “boa” solução inicial seja o ideal para o desempenho do algoritmo, procura-se antes que esta seja um ponto de partida para que o algoritmo utilizado evolua para uma solução ótima.

Como se pode observar no algoritmo 5, onde k corresponde ao número de PMUs disponível e $x_i = 1$ se um PMU é colocado no nó i , para a construção de uma solução inicial admissível para o problema é feita uma seleção aleatória dos nós onde os PMUs serão posicionados de forma a respeitar a quantidade imposta pelo orçamento (passo 2). Com a utilização dos passos 4 e 5 garante-se que um nó que já possua um PMU não é escolhido. Por fim, para indicar que o PMU está posicionado no nó i é colocado $x_i = 1$ (passo 8).

Algoritmo 5: *Solução Inicial*

Entrada: V - número de nós da rede; k - número de PMUs a serem posicionados
Saída: Solução inicial

- 1 $x \leftarrow (0, 0, \dots, 0)$;
- 2 **enquanto** $\sum_{i \in V} x_i \neq k$ **faça**
- 3 Selecionar, de forma aleatória, um nó $i \in V$;
- 4 **enquanto** $x_i = 1$ **faça**
- 5 | Selecionar, de forma aleatória um nó $i \in V$;
- 6 **fim**
- 7 $x_i = 1$;
- 8 **fim**

3.2 Avaliação da qualidade das soluções

Para se poder afirmar que uma solução é mais adequada que outra, é necessário um mecanismo que avalie a qualidade das soluções obtidas. No problema estudado nesta dissertação, como o objetivo passa por maximizar a quantidade de nós observados, o que vai distinguir as soluções é a quantidade de nós monitorizados.

Para isso teve de se implementar um algoritmo que proceda à aplicação das regras pelas quais um nó pode ser observado (definidas anteriormente). O algoritmo 6 apresenta as operações para calcular a quantidade de nós da rede que são monitorizados por um dado posicionamento de PMUs. Entre os passos 2 e 6 são descritos os procedimentos associados à regra 1. Ou seja, um nó i é observado se o nó i ou um dos seus vizinhos tem um PMU. Para a regra 2, que refere que um nó *zero-injection* i é observado se todos os seus vizinhos são observados, estão associados os passos

13 a 16. Por fim, a regra 3 que dita que se um nó *zero-injection* j e todos os seus vizinhos, exceto um (i), são observados então o nó i é observado está presente entre os passos 19 e 22.

Algoritmo 6: Nós observados

Entrada: gráfico $G = (V, E)$, S - posição dos nós *zero-injection*, vetor x com posição dos PMUs

Saída: vetor $w \in \{0, 1\}^{|V|}$ dos nós observados

```

1  $w \leftarrow (0, 0, \dots, 0)$ ;
  /* Aplicação da regra 1 */
2 para  $i \in V$  faça
3   se  $x_i = 1$  então
4      $w_i \leftarrow 1$ ;
5      $w_j \leftarrow 1$  para  $j \in N(i)$ ;
6   fim
7 fim
8  $end \leftarrow True$  e  $S' = \text{cópia}(S)$ ;
9 enquanto  $end$  faça
10   $end \leftarrow False$  e  $S'' = \text{cópia}(S')$ ;
11  para  $i \in S''$  faça
12    se  $w_i = 0$  então
13      se  $\sum_{j \in N(i)} w_j = |N(i)|$  então
14         $w_i \leftarrow 1$ ;
15        /* Aplicação da regra 2 */
16         $end \leftarrow True$ ;
17         $S' \leftarrow S' \setminus \{i\}$ ;
18      fim
19    senão
20      se  $\sum_{j \in N(i)} w_j = |N(i)| - 1$  então
21         $w_j = 1$  para  $j \in N(i)$ ;
22        /* Aplicação da regra 3 */
23         $end \leftarrow True$ ;
24         $S' \leftarrow S' \setminus \{i\}$ ;
25      fim
26    fim
27  fim
28  retorna  $w$ ;

```

3.3 Operações de Vizinhança

Como referido anteriormente, as operações de vizinhança têm um papel fundamental para o sucesso das técnicas de otimização utilizadas para a resolução do problema em estudo. Estas é que vão definir de que forma é feita a pesquisa de soluções na vizinhança de uma dada solução para tentar melhorar a qualidade da solução final.

Definida a forma de obtenção de uma solução inicial, o passo seguinte passa pela necessidade de definir um conjunto de operações que quando realizadas de forma sistemática permitam melhorar a qualidade da solução e idealmente atingir uma solução ótima. Ao longo deste subcapítulo, serão apresentadas as estruturas de vizinhança desenvolvidas neste trabalho.

3.3.1 Troca de uma posição

Tal como já foi referido anteriormente, o método mais utilizado para se obter uma solução de vizinhança ($s' \in V(s)$) da solução atual s consiste apenas numa mudança de uma posição de um PMU. Ou seja, um nó que possua um PMU deixa de o ter, passando esse PMU para um nó que não tenha. A escolha do nó de onde é retirado um PMU e do nó onde este é colocado é feita de forma totalmente aleatória. Este procedimento é descrito no algoritmo 7.

Algoritmo 7: Troca de uma posição

Entrada: solução Atual: x
Saída: Solução na vizinhança: x'

- 1 $x' \leftarrow x$;
- 2 Selecionar, de forma aleatória, um nó i em que $x_i = 1$;
- 3 $out \leftarrow i$;
 /* guardar posição para ser removido o PMU */
- 4 Selecionar, de forma aleatória, um nó i em que $x_i = 0$;
- 5 $in \leftarrow i$;
 /* guardar posição para ser inserido o PMU */
- 6 $x_{out} = 0$;
- 7 $x_{in} = 1$;

3.3.2 Operação Probabilística

No sentido de perceber se uma operação de vizinhança não totalmente aleatória trazia vantagens em relação às apresentadas anteriormente, foi desenvolvido uma operação de vizinhança mais complexa.

As operações anteriores para a conceção de uma solução vizinha escolhem de forma aleatória o nó de que o PMU é retirado e o nó que é selecionado para alocar um PMU. Na operação agora proposta a escolha do nó de onde vai ser retirado um PMU é feita com base numa probabilidade baseada no número de ligações que cada nó possui. Após o cálculo de probabilidades é feita a construção de um vetor com vários intervalos correspondentes aos nós e é feita a escolha de um número entre 0 e 1 de forma aleatória que ditará qual é a posição a ser alterada.

Para uma melhor compreensão apresenta-se um exemplo onde se considera um caso em que a rede possui três PMUs na sua solução. Na tabela 3.1 a cada nó i em que o PMU se situa indicam-se o número de ramos b_i ligados a esse nó.

Tabela 3.1: Nós e ligações

1	2	3
$b_1 = 4$	$b_2 = 3$	$b_3 = 1$

A probabilidade vai ser calculada em função da quantidade de nós ligados a cada nó i a tratar, sendo feita através da utilização da fórmula: $p_i = \frac{1}{b_i \cdot \sum_{k \in V} \frac{1}{b_k}}$ onde p_i é a probabilidade de i ser selecionado. Assim os cálculos necessários para este passo, para o exemplo considerado, são:

$$p_1 = \frac{1}{4 \cdot (\frac{1}{4} + \frac{1}{3} + 1)} \approx 0.1579$$

$$p_2 = \frac{1}{3 \cdot (\frac{1}{4} + \frac{1}{3} + 1)} \approx 0.2105$$

$$p_3 = \frac{1}{1 \cdot (\frac{1}{4} + \frac{1}{3} + 1)} \approx 0.6316$$

Se forem analisados os resultados obtidos, verifica-se que o valor obtido para nós com menos conexões é superior ao obtido para nós com mais conexões e que a sua soma é igual a 1. Partindo-se destes valores, constrói-se um vetor (*prob*) que contém valores pelos quais serão delimitados intervalos, sendo o seu formato dado por: $prob_0 = 0$; $prob_i = p_i + prob_{i-1}$, $i = 1, \dots, |V|$. Para o exemplo considerado, o vetor resultante seria: $[0; 0.1579; 0.3684; 1]$. De seguida é escolhido um número entre 0 e 1 de forma aleatória. Caso o número sorteado se situe no intervalo $[0; 0.1579[$ o PMU a ser retirado será o do nó 1. Se se situar no intervalo $[0.1579; 0.3684[$ o PMU a ser retirado é o do nó 2 e no intervalo $[0.3684; 1]$ o do nó 3.

Após obter o nó de que o PMU será retirado é feita a escolha de forma aleatória de um nó que não possua PMU para passar a ter. Estes passos são descritos no algoritmo 8.

Algoritmo 8: Operação Probabilística

Entrada: Solução atual: x
Saída: Solução na vizinhança: x'

- 1 Inicializar parâmetro para a quantidade de nós ligados ao nó i , b_i com $i \in V$;
- 2 **para** nós que possuam PMU **faça**
- 3 | Calcular a probabilidade de saída p_i utilizando a fórmula:
 |
$$p_i = \frac{1}{b_i \cdot \sum_{k \in x_k=1} \frac{1}{b_k}}$$
- 4 **fim**
- 5 $prob_0 = 0$; $prob_i = p_i + prob_{i-1}$, $i = 1, \dots, |V|$;
- 6 Selecionar, de forma aleatória, um número $t \in [0, 1]$;
- 7 **se** $t \in [prob_{i-1}, prob_i]$ **então**
- 8 | $out \leftarrow i$;
- 9 **fim**
- 10 Selecionar, de forma aleatória, um nó i em que $x_i = 0$: $in \leftarrow i$;
- 11 $x_{out} = 0$;
- 12 $x_{in} = 1$;

3.4 Pesquisa Aleatória

De modo a propor soluções alternativas à solução atual, uma das formas desenvolvidas foi a de selecionar de forma totalmente aleatória uma outra solução. Ou seja, é feita a construção de raiz de uma nova solução para posicionar os PMUs, não utilizando qualquer regra nem sendo feita qualquer perturbação à solução atual para ser obtida a solução de vizinhança. Assim, é feita basicamente a mesma operação já utilizada para a obtenção de uma solução inicial (algoritmo 5). Para a sua implementação foi utilizado o algoritmo 1, sendo que no passo 3 do mesmo não é considerada uma operação de vizinhança mas sim uma solução totalmente nova. Definiu-se para critério de paragem um número limite de iterações. Neste trabalho considerou-se que ao fim de 500000 iterações a execução é terminada.

3.5 Pesquisa Local

Uma vez definidas as formas de obter uma solução inicial admissível, as operações de vizinhança e forma para avaliar a qualidade das soluções, as mesmas podem ser integradas num algoritmo de Pesquisa Local, descrito no algoritmo 1 apresentado no capítulo anterior. O critério de paragem utilizado neste trabalho é o mesmo utilizado para a Pesquisa Aleatória (ao fim de 500000 iterações a execução é terminada).

3.6 *Simulated Annealing*

Para além das abordagens referidas em secções anteriores, foi também usado *Simulated Annealing* na resolução do problema. Sendo esta uma meta-heurística de fácil implementação, considerou-se que deveria ser a primeira a estudar. Caso se verificasse que o seu desempenho não era positivo, considerar-se-ia a possibilidade de implementar outra meta-heurística. O algoritmo 2, apresentado no capítulo anterior, foi utilizado como guia para a implementação da metodologia, sendo portanto necessário proceder à parametrização do SA.

Os parâmetros do SA têm um grande impacto no seu desempenho, sendo que uma boa parametrização à partida resultará numa pesquisa eficiente e com boas soluções, enquanto que uma má parametrização se poderá revelar desastrosa. Os parâmetros a ter em conta são: a temperatura inicial, o esquema de arrefecimento, o número de iterações à mesma temperatura e o critério de paragem.

- **Temperatura Inicial (T_0):** a temperatura inicial é geralmente determinada em função da probabilidade inicial para a aceitação de soluções piores, pretendendo-se que inicialmente esta probabilidade seja elevada (próxima de 1).

Para o problema em estudo, a temperatura inicial escolhida foi de $T_0 = 20$. Nestas condições, para o caso em que a solução de vizinhança seja pior em apenas uma unidade, ou seja, $\Delta F_n = 1$ temos que a probabilidade é $e^{-\frac{1}{20}} \approx 0.95$ (como pretendido);

- **Valor de α :** o valor de α foi testado e alterado conforme os resultados obtidos, tendo-se optado por utilizar $\alpha = 0.8$.
- **Número de iterações à mesma temperatura (L):** existem várias formas para a definição da quantidade de iterações para a mesma temperatura. Neste estudo foi implementado uma forma baseada na proposta em [38] que apresenta $L = 16 \cdot |V(s)|$, onde $|V(s)|$ corresponde ao tamanho da vizinhança de s .

A fórmula utilizada neste estudo é:

$$L = 10 \cdot (|V| - |k|).$$

- **Critério de Paragem:** no capítulo anterior são apresentadas duas formas para a escolha do critério de paragem geralmente utilizadas no SA. Neste estudo optou-se por definir um limite para o número de iterações sem que se verifique qualquer melhoria na solução. Após alguns testes e análise dos resultados

que iam sendo obtidos, optou-se por terminar a pesquisa se ao fim de 50000 iterações consecutivas a solução não melhorasse.

Capítulo 4

Testes Computacionais e Análise de Resultados

A validação e comparação das metodologias descritas no capítulo 3 foi feita usando um conjunto de instâncias disponíveis na literatura [1]. A dimensão das redes estudadas varia entre 9 e 2007 nós, sendo alguns deles *zero-injection* e outros não. As instâncias de maiores dimensão foram testadas só com o método que se apresentou mais efetivo - *Simulated Annealing*. A descrição de cada uma das instâncias consideradas é apresentada em apêndice.

Para cada uma das instâncias foram considerados vários cenários de teste, sendo que a cada cenário estava associado um número diferente de PMUs. A informação relativa aos cenários e os melhores resultados obtidos para cada cenário através de um modelo de programação inteira é apresentada na tabela 4.1. Para a obtenção destes valores foi usado o modelo apresentado na secção 2.3, inserido no “solver” de otimização comercial Gurobi [36]. Estes valores servirão de referência para a análise aos resultados obtidos através da utilização da solução proposta neste trabalho. A itálico, são apresentados valores para os quais não há garantia de otimalidade ao fim de um limite de tempo (lt) de 30 minutos. Para a construção das tabelas apresentadas ao longo deste capítulo foi utilizada a notação seguinte:

- $|V|$: Número de nós da rede;
- k : Número de PMUs utilizados;
- *Obs**: Número de nós observados usando o modelo de programação inteira proposto em [6];
- **Obs**: Número de nós observados usando a técnica em análise;

- **Sol:** Solução obtida para o problema (posição dos PMUs);
- **It:** Iteração em que a melhor solução foi obtida;
- **Time:** Tempo, em segundos, para o qual a melhor solução foi obtida;
- **Erro:** Diferença percentual entre a solução obtida neste estudo e a obtida por Obs^* . Calculada através de $Erro = \frac{Obs^* - Obs}{Obs^*} \cdot 100\%$.

Tabela 4.1: Cenários e resultados obtidos através de [6]

$ V $	$ k $	Obs^*	Time (s)
9	1	4	0.04
14	1	7	0.02
	2	11	0.02
24	2	12	0.21
	3	17	0.22
	4	20	0.12
30	3	22	0.22
	4	26	0.13
	6	29	0.10
39	3	20	38.93
	5	30	748.18
	7	37	2.6
57	5	37	77.8
	8	49	66.97
73	7	45	244.36
	10	58	268.57
	14	68	37.77
118	11	77	9.4
	17	98	3.47
	23	111	3.33
150	15	96	<i>tl</i>
	22	133	<i>tl</i>
	30	149	19.52
300	30	213	<i>tl</i>
	45	259	<i>tl</i>
	60	293	<i>tl</i>
2007	200	–	–
	300	–	–
	400	–	–

Como se pode observar na tabela 4.1, para $|V| \geq 150$ as técnicas exatas deixam de ser eficientes, não havendo garantia de otimalidade para a maioria das soluções

testadas. Para $|V| = 2007$ o solver não revelou capacidade suficiente para a resolução do problema.

4.1 Pesquisa Aleatória

Os resultados obtidos através da pesquisa aleatória são apresentados na tabela 4.2. A geração aleatória de soluções consegue atingir o valor ótimo para instâncias pequenas (até 39 nós e 7 PMUs). A partir deste, e com o aumento do tamanho das redes e da quantidade de PMUs, não foi obtida nenhuma solução que igualasse as obtidas através do modelo de programação inteira.

Uma explicação para este acontecimento é a ausência de uma fase de intensificação da procura de soluções numa vizinhança da solução atual.

4.2 Pesquisa Local

Os resultados obtidos usando pesquisa local encontram-se sumariados nas tabelas 4.3 e 4.4. A tabela 4.3 apresenta os resultados obtidos usando a operação de vizinhança “Troca de uma posição” e a tabela 4.4 os resultados obtidos usando a operação de vizinhança “Operação Probabilística”.

Uma análise geral dos resultados obtidos com a utilização de um algoritmo de pesquisa local clássico, sem nenhuma estratégia delineada para ultrapassar os ótimos locais, mostra que a pesquisa local encontra a mesma solução dos *Obs** para vários cenários. Apresentando inclusive, para o cenário da rede de tamanho 150 com a utilização de 15 PMUs, soluções melhores em comparação com *Obs**.

Comparativamente à abordagem discutida na secção anterior, a operação de vizinhança “Troca de uma posição” diminuiu consideravelmente o erro obtido para redes de maior tamanho, tendo apresentado uma solução melhor para $|V| = 150$ e $|k| = 15$. Por outro lado, para redes de menor tamanho o seu desempenho foi inferior, o que numa primeira instância levou a crer que esta estrutura e vizinhança não era suficientemente “ampla” para permitir uma correta exploração do espaço de pesquisa.

No entanto, os resultados obtidos com a utilização da operação de vizinhança “Operação Probabilística” inserida num algoritmo de pesquisa local foram idênticos aos obtidos através da utilização da operação de vizinhança anterior o que leva a crer que há uma necessidade generalizada de implementação de mecanismos que permitem escapar a ótimos locais. Destaque-se nesta abordagem a obtenção da melhor solução até ao momento para o cenário $|V| = 150$ e $|k| = 15$.

Tabela 4.2: Pesquisa Aleatória (PA)

$ V $	$ k $	Obs^*	Obs	Sol	It	Time(s)	Erro
9	1	4	4	4	1	0.00	0.00%
14	1	7	7	4	2	0.00	0.00%
	2	11	11	6,9	14	0.00	0.00%
24	2	12	12	9,16	109	0.00	0.00%
	3	17	17	9,10,16	1	0.00	0.00%
30	4	20	20	3,10,16,21	229	0.00	0.00%
	3	22	22	10,12,27	1,211	0.01	0.00%
39	4	26	26	1,10,12,27	890	0.00	0.00%
	6	29	29	2,10,12,18,24	8,613	0.05	0.00%
57	3	20	20	8,13,16	15,588	0.10	0.00%
	5	30	30	2,8,13,16,26	11,715	0.10	0.00%
73	7	37	36	2,7,12,16,22,26,34	270,715	2.71	2.70%
	5	37	36	6,13,24,35,38	214,432	2.85	2.70%
118	8	49	46	1,7,13,24,31,38,53,56	387,283	6.42	6.12%
	7	45	40	9,34,35,39,57,64,71	34,916	0.63	11.11%
150	10	58	52	9,14,27,34,39,45,58,63,66,71	160,894	3.14	10.34%
	14	68	62	1,8,13,21,23,25,2,34,45,47,51,58,64,70	335,933	7.65	8.82%
150	11	77	64	8,12,20,27,32,37,49,80,93,107,110	81,869	1.90	16.88%
	17	98	83	9,12,17,19,23,34,39,47,49,56,61,71,77,79,85,100,109	249,225	7.27	15.31%
150	23	111	93	1,9,12,9,26,29,32,40,45,48,51,52,55,56,66,71,77,80,86,89,94,105,110	52,830	1.91	16.22%
	15	96	88	10,22,23,33,40,63,70,73,80,121,137,142,144,146,150	177,468	14.19	8.33%
150	22	133	105	3,10,12,21,22,23,33,40,63,70,73,80,81,84,121,136,137,138,142,144,146,150	117,884	11.79	21.05%
	30	149	125	88,107,119,130,134,136,137,139,141,144,146,150	496,235	57.45	16.11%

Tabela 4.3: Troca uma posição (PL-TP)

$ V $	$ k $	Obs^*	PA	Obs	Sol	It	Time(s)	Erro
9	1	4	4	4	9	2	0.00	0.00%
14	1	7	7	7	4	1	0.00	0.00%
	2	11	11	11	6,9	6	0.00	0.00%
24	2	12	12	12	9,16	71	0.00	0.00%
	3	17	17	17	3,10,16	58	0.00	0.00%
30	4	20	20	19	1,12,16,21	14	0.00	5.00%
	3	22	22	22	10,12,27	24	0.01	0.00%
39	4	26	26	24	6,10,15,29	43	0.00	7.69%
	6	29	29	29	2,10,12,15,18,27	118	0.00	0.00%
57	3	20	20	19	3,8,12	105	0.00	5.00%
	5	30	30	30	3,8,10,16,26	128	0.00	0.00%
73	7	37	36	35	10,16,20,23,26,31,39	1,130	0.01	5.41%
	5	37	36	37	6,13,24,38,56	761	0.02	0.00%
118	8	49	46	44	4,9,15,20,25,36,49,52	105	0.00	10.20%
	7	45	40	42	10,15,23,27,47,60,64	521	0.01	6.67%
150	10	58	52	57	9,16,21,34,39,46,47,51,58,64	5,079	0.14	1.72%
	14	68	62	64	6,7,9,16,18,26,32,40,45,47,57,60,67,69	5,917	0.15	5.88%
150	11	77	64	77	8,12,32,37,49,59,70,80,85,92,105	2,896	0.10	0.00%
	17	98	83	97	4,10,12,15,23,32,37,49,56,61,71,77,80,85,92,105,110	4,162	0.16	1.02%
150	23	111	93	109	10,11,12,19,21,24,27,31,35,40,44,49,52,56,62,65,70,77,80,85,92,105,110	6,381	0.246	1.80%
	15	96	88	104	17,19,25,40,64,68,69,78,88,137,139,144,146,150	2,875	0.31	-8.33%
150	22	133	105	126	4,7,21,26,27,36,42,47,56,59,62,68,69,71,129,137,139,141,142,144,146,150	2,819	0.41	5.26%
	30	149	125	145	4,6,8,20,21,31,34,37,45,50,60,62,65,71,78,79,84,87,90,136,137,139,140,141,142,144,146,149,150	3,229	0.45	2.68%

Tabela 4.4: Operação Probabilística (PL-OP)

V	k	Obs*	PA	PL-TP	Obs	Sol	It	Time(s)	Erro
9	1	4	4	4	4	9	2	0.01	0.00%
14	1	7	7	7	7	4	1	0.03	0.00%
	2	11	11	11	11	6,9	2	0.00	0.00%
24	2	12	12	12	12	9,16	2	0.01	0.00%
	3	17	17	17	16	10,15,23	21	0.00	5.88%
30	4	20	20	19	20	6,9,16,18	11	0.00	0.00%
	3	22	22	22	22	10,12,27	7	0.01	0.00%
39	4	26	26	24	26	2,10,12,27	32	0.00	0.00%
	6	29	29	29	29	2,10,12,19,24,27	40	0.00	0.00%
57	3	20	20	19	18	6,10,16	36	0.05	10.00%
	5	30	30	30	30	8,13,16,26,30	236	0.01	0.00%
73	7	37	36	35	39	8,10,16,18,23,25,31	11	0.00	10.81%
	5	37	36	37	32	6,15,20,36,49	181	0.01	13.51%
118	8	49	46	44	49	1,4,13,24,29,31,38,56	410	0.01	0.00%
	7	45	40	42	43	9,21,27,34,40,57,64	344	0.01	4.44%
150	10	58	52	57	56	9,10,16,21,33,40,47,53,57,64	498	0.02	3.45%
	14	68	62	64	67	2,10,15,20,21,26,34,39,46,47,51,58,67,69	1,185	0.03	1.47%
150	11	77	64	77	77	5,12,17,23,37,49,59,80,85,92,105	926	0.04	0.00%
	17	98	83	97	97	10,11,12,15,27,32,37,49,56,61,72,75,80,85,92,105,110	7,286	0.27	1.02%
150	23	111	93	109	109	4,10,12,15,17,20,23,29,34,40,45,49,51,59,67,71,77,80,85,92,105,110,115	6,940	0.29	1.80%
	15	96	88	104	107	22,26,40,47,56,62,78,137,139,140,141,144,146,147,150	7,231	0.83	-11.46%
150	22	133	105	126	130	13,22,36,37,39,42,47,56,68,71,75,79,83,90,137,138,139,141,144,146,147,150	3,282	0.42	2.26%
	30	149	125	145	147	4,6,21,26,30,34,38,40,43,50,60,62,68,70,71,77,78,83,84,90,137,139,140,141,144,146,147,149,150	2,329	0.31	1.34%

Concluindo, a operação de vizinhança “Operação Probabilística” inserida num algoritmo de pesquisa local foi a que até este ponto apresentou melhores resultados, não tendo atingido a melhor solução para 11 cenários. As restantes abordagens não atingiram a melhor solução conhecida para 12 cenários.

4.3 *Simulated Annealing*

Os resultados obtidos através da implementação da meta-heurística SA podem ser observados nas tabelas 4.5 para a operação “Troca de uma posição” e na 4.6 para a “Operação Probabilística”.

Como se pode observar, as duas metodologias atingiram os valores de melhor solução conhecida para cada cenário, tendo inclusive apresentado um melhor resultado para o cenário $|V| = 150$ e $|k| = 15$.

Não se pode afirmar que um operador de vizinhança se destaque em relação ao outro, pois tanto a nível de número de iterações e de tempo necessário para obter as melhores soluções não apresentam diferenças significativas.

Para a abordagem baseada em SA foram consideradas um conjunto de redes com 300 e 2007 nós. Os resultados obtidos encontram-se na tabela 4.7.

Para os cenários propostos para a rede com 300 nós, os resultados obtidos foram iguais para as duas operações de vizinhança propostas. Sendo obtidos resultados que apresentam uma melhoria significativa na qualidade da solução para os dois primeiros cenários, que utilizam 30 e 45 PMUs.

Para os cenários de teste da rede com 2007 nós não existem resultados obtidos através do modelo de programação inteira para comparação, sendo que a comparação será feita entre os resultados obtidos para as duas operações de vizinhança. Ao contrário dos casos anteriores, para esta rede a “Operação Probabilística” apresenta melhores resultados na quantidade de nós observados, no número de iterações que necessita para o atingir e também no intervalo de tempo necessário.

De forma geral os resultados obtidos com a utilização de SA revelaram-se positivos, tendo atingido a meta proposta para esta dissertação. Ou seja, o desenvolvimento de uma metodologia eficiente para a resolução do problema proposto. Isto pode ser afirmado, uma vez que os resultados são obtidos num intervalo de tempo inferior ao que eram obtidos através do modelo de programação inteira garantindo a melhor solução conhecida para cada cenário. Outro fator em que a meta-heurística se revelou uma técnica de otimização eficiente, foi na obtenção de soluções para os cenários da rede com 2007 nós, pois a partir de técnicas exatas não foi possível obter nenhuma solução.

Tabela 4.5: SA - Troca uma posição

V	k	Obs*	Obs	Sol	It	Time(s)	Erro
9	1	4	4	7	2	0.00	0.00%
14	1	7	7	4	1	0.00	0.00%
	2	11	11	4,6	12	0.00	0.00%
24	2	12	12	9,16	193	0.00	0.00%
	3	17	17	6,9,16	385	0.00	0.00%
30	4	20	20	5,9,16,22	166	0.00	0.00%
	3	22	22	10,12,27	1,371	0.01	0.00%
39	4	26	26	2,10,12,27	734	0.01	0.00%
	6	29	29	2,10,12,19,24,27	2,727	0.02	0.00%
57	3	20	20	8,13,16	7,303	0.06	0.00%
	5	30	30	3,8,10,19,26	3,594	0.04	0.00%
73	7	37	37	6,13,24,38,56	7,655	0.12	0.00%
	5	37	37	6,13,24,38,56	761	0.02	0.00%
118	8	49	49	1,6,9,15,24,31,38,56	7,029	0.13	0.00%
	7	45	45	9,16,34,39,47,58,71	11,991	0.24	0.00%
150	10	58	58	9,19,21,34,39,41,47,54,57,64	37,546	0.86	0.00%
	14	68	68	3,10,19,21,26,27,34,40,45,50,53,57,67,69	27,925	0.69	0.00%
150	11	77	77	8,12,32,37,49,59,70,80,85,92,105	17,435	0.47	0.00%
	17	98	98	10,11,12,17,23,34,37,49,59,71,77,80,85,92,105,110,115	20,793	0.69	0.00%
150	23	111	111	10,11,12,17,21,23,28,34,37,45,49,52,56,61,72,75,77,80,85,92,105,110,114	24,598	0.92	0.00%
	15	96	108	16,23,37,47,56,65,70,78,88,137,139,141,144,146,150	30,147	3.12	-12.50%
150	22	133	133	4,13,29,36,38,47,54,56,62,65,71,75,84,87,90,137,139,140,141,144,146,150	30,575	3.43	0.00%
	30	149	149	3,6,21,25,26,30,36,38,40,44,47,54,56,59,62,65,71,75,77,79,85,87,135,137,139,142,143,144,146,150	44,570	5.30	0.00%

Tabela 4.6: SA - Operação Probabilística

$ V $	$ k $	Obs^*	Obs	Sol		It	Time(s)	Erro
9	1	4	4	7		2	0.01	0.00%
14	1	7	7	4		1	0.00	0.00%
	2	11	11	4,6		18	0.00	0.00%
24	2	12	12	9,16		2	0.00	0.00%
	3	17	17	6,9,16		194	0.00	0.00%
30	4	20	20	9,10,16,22		223	0.00	0.00%
	3	22	22	10,12,27		1,769	0.02	0.00%
39	4	26	26	2,10,12,27		330	0.00	0.00%
	6	29	29	2,10,12,19,24,27		2,313	0.02	0.00%
57	3	20	20	8,13,16		2,300	0.03	0.00%
	5	30	30	3,8,10,16,26		1,702	0.02	0.00%
73	7	37	37	2,6,12,16,20,23,26		4,527	0.06	0.00%
	5	37	37	6,13,24,38,56		7,177	0.13	0.00%
118	8	49	49	1,6,9,15,24,31,38,56		8,121	0.17	0.00%
	7	45	45	9,16,34,39,47,58,71		9,474	0.20	0.00%
150	10	58	58	9,10,21,27,34,41,51,58,66,67		10,172	0.24	0.00%
	14	68	68	1,10,19,21,27,34,39,41,44,53,54,57,67,69		11,464	0.31	0.00%
118	11	77	77	8,12,32,37,49,59,70,80,85,92,105		14,594	0.43	0.00%
	17	98	98	8,11,12,17,20,23,28,37,49,59,71,77,80,85,92,105,110		19,612	0.68	0.00%
150	23	111	111	8,11,12,17,21,27,32,34,37,40,45,49,53,56,61,72,75,77,80,85,92,105,110		27,302	1.04	0.00%
	15	96	108	13,23,37,47,56,65,70,78,88,137,139,141,144,146,150		29,101	3.15	-12.50%
150	22	133	133	13,19,21,25,38,39,42,47,56,62,65,71,75,84,87,137,139,142,144,146,150		34,411	3.98	0.00%
	30	149	149	4,16,19,21,24,25,28,34,38,39,41,47,54,56,59,62,68,69,71,78,79,81,83,88,137,139,142,144,146,150		25,051	3.15	0.00%

Tabela 4.7: SA - Troca de uma posição e Operação Probabilística

Troca de uma posição						
$ V $	$ k $	Obs^*	Obs	It	Time(s)	Erro
300	30	<i>213</i>	224	55,118	24.16	-5.16%
	45	<i>259</i>	269	78,838	37.97	-3.86%
	60	<i>293</i>	293	54,412	27.81	0.00%
2007	200	-	1087	71,010	1078.76	-
	300	-	1440	103,900	1790.14	-
	400	-	1717	180,437	3439.91	-
Operação Probabilística						
$ V $	$ k $	Obs^*	Obs	It	Time(s)	Erro
300	30	<i>213</i>	224	60,099	26.96	-5.16%
	45	<i>259</i>	269	51,969	26.16	-3.86%
	60	<i>293</i>	293	56,064	29.25	0.00%
2007	200	-	1094	68,458	1053.71	-
	300	-	1446	87,210	1540.11	-
	400	-	1722	118,827	2286.6	-

Capítulo 5

Conclusão e Desenvolvimentos Futuros

Nesta dissertação é proposta uma solução para o problema OPP, tendo em consideração a existência de um investimento máximo que pode ser feito na compra de PMUs. O problema insere-se na evolução que se tem vindo a verificar no sistema elétrico, tendo os PMUs um papel fundamental para que o aperfeiçoamento da monitorização, proteção e controlo das redes elétricas seja alcançado.

O primeiro passo para a sua resolução, de forma a compreender os seus conceitos fundamentais, foi o estudo acerca do problema OPP clássico, sendo apresentada uma contextualização e descrição do mesmo. De seguida foi proposto um modelo matemático de forma a refletir o problema em estudo (considerando as restrições orçamentais). Foram também referidas várias técnicas de otimização que podem ser utilizadas para a resolução do problema, tal como um conjunto de conceitos gerais sobre meta-heurísticas (abordagem utilizada neste trabalho).

Na fase de desenvolvimento de metodologias para a resolução do problema em estudo, antes de se elaborar uma solução baseada numa meta-heurística, foram testadas soluções em que se recorreu às técnicas de Pesquisa Local e Pesquisa Aleatória que se revelaram determinantes na compreensão da importância de o algoritmo proposto, realizar tanto a fase de diversificação e intensificação na pesquisa de soluções na vizinhança. Foram ainda propostas duas estruturas de vizinhança distintas, posteriormente usadas tanto na Pesquisa Local como no *Simulated Annealing*. A Pesquisa Aleatória não possui uma operação de vizinhança, pois a forma que utiliza para encontrar uma solução é gerando de raiz uma outra solução.

Em termos de resultados dos testes realizados, a meta-heurística *Simulated Annealing* revelou-se como o método mais eficiente para a resolução do problema aqui

apresentado, tendo obtido as melhores soluções para cada cenário. Na verdade, para todas as instâncias para que era conhecida a solução ótima, o método convergiu para essa solução. Em vários casos, encontrou uma solução melhor do que a conhecida até ao momento. Outro ponto a favor, foi o facto de o resultado obtido para a rede de 2007 nós ter ido ao encontro da génese dos métodos aproximados, ou seja, a procura de obter uma solução “boa” para o problema num curto espaço de tempo, para o qual os métodos exatos se revelam incapazes de resolver. Assim, pode-se afirmar que os objetivos delineados para esta dissertação foram atingidos com sucesso.

Como perspetiva de desenvolvimento futuro sugere-se a implementação de outros métodos de otimização, mais concretamente a utilização de outras meta-heurísticas (*Tabu Search*, Algoritmos Genéticos, etc) e fazer uma comparação com a metodologia já desenvolvida. Também poderia ser considerada outra variante ao problema OPP, podendo, por exemplo, restringir o número de nós que cada PMU pode observar ou considerar custos associados a cada nó.

Bibliografia

- [1] Illinois center for a smarter electric grid (icseg). <http://icseg.iti.illinois.edu/power-cases/>. Acedido a: 30/01/2017.
- [2] Ieee standard for synchrophasers for power systems. *IEEE Std 1344-1995(R2001)*, 1995.
- [3] F. Aminifar, C. Lucas, A. Khodaei, and M. Fotuhi-Firuzabad. Optimal placement of phasor measurement units using immunity genetic algorithm. *IEEE Transactions on Power Delivery*, 24(3):1014–1020, July 2009.
- [4] Xu Bei, Yeo Jun Yoon, and A Abur. Optimal placement and utilization of phasor measurements for state estimation. *PSERC Publication*, 01 2005.
- [5] Brandao. genetic algorithms in php code. <http://www.abrandao.com/2015/01/simple-php-genetic-algorithm/>. Acedido a: 12/10/2017.
- [6] Margarida Carvalho, Xenia Klimentova, and Ana Viana. Observability of power with optimal pmu placement. *Technical Report DS4DM-2017-010, CERC Data Science for Real-time Decision Making*, 2017 (submitted).
- [7] S. Chakrabarti, E. Kyriakides, and D. G. Eliades. Placement of synchronized measurements for power system observability. *IEEE Transactions on Power Delivery*, 24(1):12–19, Jan 2009.
- [8] João Claro. Uma abordagem orientada por objectos para meta-heurísticas multi-objectivo. <https://repositorio-aberto.up.pt/bitstream/10216/62513/1/Jo%C3%A3o%20Claro.PDF>, 2002. Acedido a: 02/10/2017.
- [9] M. DORIGO. Optimization, learning and natural algorithms. *Ph.D. Thesis, Politecnico di Milano, Italy*, 1992.
- [10] EDP. Microprodução. <https://www.edpsu.pt/pt/PRE/Microproducao/Pages/Microgeracao.aspx>. Acedido a: 02/10/2017.
- [11] EDP. Redes inteligentes. <https://www.edpdistribuicao.pt/pt/rede/InovGrid/Pages/RedesInteligentes.aspx>. Acedido a: 02/10/2017.
- [12] J.Doyne Farmer, Norman H Packard, and Alan S Perelson. The immune system, adaptation, and machine learning. *Physica D: Nonlinear Phenomena*, 22(1):187 – 204, 1986. Proceedings of the Fifth Annual International Conference.

- [13] Thomas A Feo and Mauricio G. C Resende. A probabilistic heuristic for a computationally difficult set covering problem. *Oper. Res. Lett.*, 8(2):67–71, April 1989.
- [14] Robert Fourer, David M. Gay, and Brian W. Kernighan. *AMPL: A Modeling Language for Mathematical Programming*, volume 36. 2003.
- [15] Arun G Phadke and J.s Thorp. *Synchronized Phasor Measurements and Their Applications*. Springer International Publishing, 01 2008.
- [16] Fred Glover. Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research*, 13(5):533 – 549, 1986. Applications of Integer Programming.
- [17] Fred Glover. Future paths for integer programming and links to artificial intelligence. *Comput. Oper. Res.*, 13(5):533–549, May 1986.
- [18] Antonio Gomez-Exposito, Antonio J. Conejo, and Claudio Cañizares. *Electric Energy Systems: Analysis and Operation*. CRC Press, 01 2008.
- [19] B. Gou. Generalized integer linear programming formulation for optimal pmu placement. *IEEE Transactions on Power Systems*, 23(3):1099–1104, Aug 2008.
- [20] M. Hajian, A. M. Ranjbar, T. Amraee, and A. R. Shirani. Optimal placement of phasor measurement units: Particle swarm optimization approach. In *2007 International Conference on Intelligent Systems Applications to Power Systems*, pages 1–6, Nov 2007.
- [21] David G. Hart, David Uy, Vasudev Gharpure, Damir Novosel, Daniel Karlsson, and Mehmet Kaba. Pmus – a new approach to power network monitoring. *ABB Review 1/2001*, pages 58–61, 2001.
- [22] John H. Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. Cambridge, Mass. MIT Press, 1994.
- [23] Zhao Hong-Shan, Li Ying, Mi Zeng-qiang, and Yu Lei. Sensitivity constrained pmu placement for complete observability of power systems. In *2005 IEEE/PES Transmission Distribution Conference Exposition: Asia and Pacific*, pages 1–5, 2005.
- [24] James Kennedy and Russell C. Eberhart. Particle swarm optimization. In *Proceedings of the 1995 IEEE International Conference on Neural Networks*, volume 4, pages 1942–1948. Perth, Australia, IEEE Service Center, Piscataway, NJ, 1995.
- [25] Reduan Khan and J.Y. Khan. A comprehensive review of the application characteristics and traffic requirements of a smart grid communications network. In *Computer Networks*, volume 57, page 825–845. 02 2013.

- [26] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.
- [27] Mariana Magalhães. Optimização de rotas. https://paginas.fe.up.pt/~ee05135/Relatorio%20Final%20PDI/ee05135_Relatorio%20Final%20PDI.pdf, 2011. Acedido a: 02/10/2017.
- [28] N. M. Manousakis, G. N. Korres, and P. S. Georgilakis. Optimal placement of phasor measurement units: A literature review. In *2011 16th International Conference on Intelligent System Applications to Power Systems*, pages 1–6, Sept 2011.
- [29] F. J. Marín, F. García-Lagos, G. Joya, and F. Sandoval. *Optimal Phasor Measurement Unit Placement using Genetic Algorithms*, pages 486–493. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.
- [30] António Aires Messias. Redes inteligentes de energia – smart grids. http://www.ordemengenheiros.pt/fotos/editor2/eventos/aires_messias.pdf, 19 de Junho de 2009.
- [31] B Mohammadi-Ivatloo. Optimal placement of pmus for power system observability using topology based formulated algorithms. *Journal of Applied Sciences*, 9(13):2463–2468, 2009.
- [32] United Nations. World population prospects - population division - united nations. <https://esa.un.org/unpd/wpp/DataQuery/>. Acedido a: 30/10/2017.
- [33] M. Nazari-Heris and B. Mohammadi-Ivatloo. Application of heuristic algorithms to optimal pmu placement in electric power systems: An updated review. *Renewable and Sustainable Energy Reviews*, 50(Supplement C):214 – 228, 2015.
- [34] J. Nocedal and S. Wright. *Quadratic Programming*, pages 448–492. Springer New York, New York, NY, 2006.
- [35] R. F. Nuqui and A. G. Phadke. Phasor measurement unit placement techniques for complete and incomplete observability. *IEEE Transactions on Power Delivery*, 20(4):2381–2388, Oct 2005.
- [36] GUROBI Optimization. <http://www.gurobi.com/>.
- [37] Jiangnan Peng, Yuanzhang Sun, and H.F. Wang. Optimal pmu placement for full network observability using tabu search algorithm. *International Journal of Electrical Power & Energy Systems*, 28(4):223 – 231, 2006.
- [38] Marc Pirlot. General local search heuristics in combinatorial optimization: a tutorial. *Belgian Journal of Operations Research, Statistics and Computer Science (JORBEL)*, 32, 01 1992.

- [39] Pierre-Louis Poirion, Sonia Toubaline, Claudia D'Ambrosio, and Leo Liberti. The power edge set problem. *Netw.*, 68(2):104–120, September 2016.
- [40] M. Qiu, H. Su, M. Chen, Z. Ming, and L. T. Yang. Balance of security strength and energy for a pmu monitoring system in smart grid. *IEEE Communications Magazine*, 50(5):142–149, May 2012.
- [41] Vaishali Rampurkar. Smart grid & wams. http://sites.ieee.org/northern-canada-pesias/files/2016/01/Smart_grid_January-26.pdf. Acedido a: 26/01/2016.
- [42] David Rutten. Simulated annealing, a brief introduction. <https://ieatbugsforbreakfast.wordpress.com/2011/10/14/simulated-annealing-a-brief-introduction/>, 2011. Acedido a: 10/10/2017.
- [43] André Santos. Análise do desempenho de técnicas de otimização no problema de escalonamento. https://recipp.ipp.pt/bitstream/10400.22/7114/1/Dis_SantosAndré_2015.pdf, 2015. Acedido a: 02/10/2017.
- [44] M. Shahraeini and M.H. Javidi. *Wide Area Measurement Systems*. INTECH Open Access Publisher, 2012.
- [45] Ranjana Sodhi and S C Srivastava. Optimal pmu placement to ensure observability of power system. 10 2017.
- [46] C. Su and Z. Chen. Optimal placement of phasor measurement units with new considerations. In *2010 Asia-Pacific Power and Energy Engineering Conference*, pages 1–4, March 2010.
- [47] El-Ghazali Talbi. *Metaheuristics: From Design to Implementation*. Wiley Publishing, 2009.
- [48] João Torres. Powerpoint presentation-joao_torres.pdf. http://www.apenergia.pt/uploads/docs/Joao_torres.pdf, 2014. Acedido a: 02/10/2017.
- [49] Department of Economic United Nations and Population Division (2017) Social Affairs. World population prospects: The 2017 revision. <https://esa.un.org/unpd/wpp/DataQuery/>, 24.09.2017.
- [50] B. Wang, D. Liu, and X. Li. An improved ant colony system in optimizing power system pmu placement problem. In *2009 Asia-Pacific Power and Energy Engineering Conference*, pages 1–3, March 2009.
- [51] Hector Wiggins. Presentation on theme: "evolutionary algorithms"— presentation transcript. <http://slideplayer.com/slide/6964904/>. Acedido a: 02/10/2017.

- [52] Dan Woodfin. Presentation on theme: "synchrophasor measurement implementation in ertot"— presentation transcript. <http://slideplayer.com/slide/6855120/>. Acedido a: 02/10/2017.

Apêndice A

Instâncias de teste

Em apêndice encontram-se as características das instâncias de teste utilizadas nesta dissertação. Sendo que podem ser consultadas em mais detalhe em [1].

Tabela A.1: *Instâncias Utilizadas*

WSCC 9-bus		
Quantidade de nós	Quantidade de ramos	nós <i>zero-injection</i>
9	9	3
Posição dos nós <i>zero-injection</i>		
4, 7, 9		
IEEE 14-bus		
Quantidade de nós	Quantidade de ramos	nós <i>zero-injection</i>
14	20	1
Posição dos nós <i>zero-injection</i>		
7		
IEEE 24-bus		
Quantidade de nós	Quantidade de ramos	nós <i>zero-injection</i>
24	34	4
Posição dos nós <i>zero-injection</i>		
11, 12, 17, 24		
IEEE 30-bus		
Quantidade de nós	Quantidade de ramos	nós <i>zero-injection</i>
30	41	6
Posição dos nós <i>zero-injection</i>		
6, 9, 22, 25, 27, 28		
IEEE 39-bus		
Quantidade de nós	Quantidade de ramos	nós <i>zero-injection</i>
39	46	12
Posição dos nós <i>zero-injection</i>		
1, 2, 5, 6, 9, 10, 11, 13, 14, 17, 19, 22		

IEEE 57-bus		
Quantidade de nós	Quantidade de ramos	nós <i>zero-injection</i>
57	78	15
Posição dos nós <i>zero-injection</i>		
4, 7, 11, 21, 22, 24, 26, 34, 36, 37, 39, 40, 45, 46, 48		
IEEE 96-RTS		
Quantidade de nós	Quantidade de ramos	nós <i>zero-injection</i>
73	108	13
Posição dos nós <i>zero-injection</i>		
11, 12, 17, 24, 35, 36, 41, 48, 59, 60, 65, 72, 73		
IEEE 118-bus		
Quantidade de nós	Quantidade de ramos	nós <i>zero-injection</i>
118	179	10
Posição dos nós <i>zero-injection</i>		
5, 9, 30, 37, 38, 63, 64, 68, 71, 81		
UIUC 150-bus		
Quantidade de nós	Quantidade de ramos	nós <i>zero-injection</i>
150	203	33
Posição dos nós <i>zero-injection</i>		
92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150		
IEEE 300-bus		
Quantidade de nós	Quantidade de ramos	nós <i>zero-injection</i>
300	409	68
Posição dos nós <i>zero-injection</i>		
4, 7, 12, 16, 23, 129, 130, 18, 28, 29, 30, 33, 36, 39, 40, 52, 54, 56, 57, 62, 65, 68, 70, 71, 72, 73, 82, 94, 95, 96, 99, 107, 108, 109, 110, 111, 112, 113, 123, 137, 139, 144, 145, 147, 148, 153, 172, 173, 142, 143, 174, 184, 189, 191, 198, 205, 216, 219, 223, 245, 246, 266, 270, 271, 272, 273, 276, 291		
Texas 2000		
Quantidade de nós	Quantidade de ramos	nós <i>zero-injection</i>
2007	5214	308
Posição dos nós <i>zero-injection</i>		
1419, 1420, 1421, 1422, 1423, 1424, 1425, 1426, 1427, 1428, 1429, 1430, 1431, 1432, 1433, 1434, 1435, 1436, 1437, 1438, 1439, 1440, 1441, 1442, 1443, 1444, 1445, 1446, 1447, 1448, 1449, 1450, 1451, 1452, 1453, 1454, 1455, 1456, 1457, 1458, 1459, 1460, 1461, 1462, 1463, 1464, 1465, 1466, 1467, 1468, 1469, 1470, 1471, 1472, 1473, 1474, 1475, 1476, 1477, 1478, 1479, 1480, 1481, 1482, 1483, 1484, 1485, 1486, 1487, 1488, 1489, 1490, 1491, 1492, 1493, 1494, 1495, 1496, 1497, 1498, 1499, 1500, 1501, 1783, 1784, 1785, 1786, 1787, 1788, 1789, 1790, 1791, 1792, 1793, 1794, 1795, 1796, 1797, 1798, 1799, 1800, 1801, 1802, 1803, 1804, 1805, 1806, 1807, 1808, 1809, 1810, 1811, 1812, 1813, 1814, 1815, 1816, 1817, 1818, 1819, 1820, 1821, 1822, 1823, 1824, 1825, 1826, 1827, 1828, 1829, 1830, 1831, 1832, 1833, 1834, 1835, 1836, 1837, 1838, 1839, 1840, 1841, 1842, 1843, 1844, 1845, 1846, 1847, 1848, 1849, 1850, 1851, 1852, 1853, 1854, 1855, 1856, 1857, 1858, 1859, 1860, 1861, 1862, 1863, 1864, 1865, 1866, 1867, 1868,		

1869, 1870, 1871, 1872, 1873, 1874, 1875, 1876, 1877, 1878, 1879, 1880, 1881,
1882, 1883, 1884, 1885, 1886, 1887, 1888, 1889, 1890, 1891, 1892, 1893, 1894,
1895, 1896, 1897, 1898, 1899, 1900, 1901, 1902, 1903, 1904, 1905, 1906, 1907,
1908, 1909, 1910, 1911, 1912, 1913, 1914, 1915, 1916, 1917, 1918, 1919, 1920,
1921, 1922, 1923, 1924, 1925, 1926, 1927, 1928, 1929, 1930, 1931, 1932, 1933,
1934, 1935, 1936, 1937, 1938, 1939, 1940, 1941, 1942, 1943, 1944, 1945, 1946,
1947, 1948, 1949, 1950, 1951, 1952, 1953, 1954, 1955, 1956, 1957, 1958, 1959,
1960, 1961, 1962, 1963, 1964, 1965, 1966, 1967, 1968, 1969, 1970, 1971, 1972,
1973, 1974, 1975, 1976, 1977, 1978, 1979, 1980, 1981, 1982, 1983, 1984, 1985,
1986, 1987, 1988, 1989, 1990, 1991, 1992, 1993, 1994, 1995, 1996, 1997, 1998,
1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007