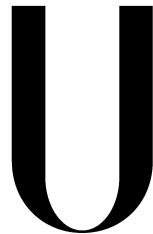


UNIVERSIDADE DE LISBOA
Faculdade de Ciências
Departamento de Estatística e Investigação Operacional



LISBOA

UNIVERSIDADE
DE LISBOA

**ALGORITMO GENÉTICO PARA O
PROBLEMA DE LOCALIZAÇÃO DE
SENSORES**

Ana Sofia Fonseca de Carvalho

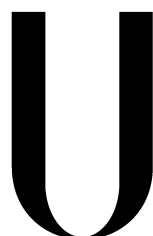
DISSERTAÇÃO

Mestrado em Estatística e Investigação Operacional

Especialização em Investigação Operacional

2013

UNIVERSIDADE DE LISBOA
Faculdade de Ciências
Departamento de Estatística e Investigação Operacional



LISBOA

UNIVERSIDADE
DE LISBOA

**ALGORITMO GENÉTICO PARA O
PROBLEMA DE LOCALIZAÇÃO DE
SENSORES**

Ana Sofia Fonseca de Carvalho

DISSERTAÇÃO

Mestrado em Estatística e Investigação Operacional

Especialização em Investigação Operacional

Orientadores:

Prof^a. Dr^a. Maria Eugénia Vasconcelos Captivo

Prof. Dr. António José Lopes Rodrigues

2013

Aos meus orientadores, a Professora Doutora Maria Eugénia Captivo e o Professor Doutor António José Rodrigues, o meu obrigada por todo o apoio, disponibilidade e colaboração ao longo da realização deste trabalho. O saber transmitido, as palavras de incentivo, as opiniões e as críticas foram fundamentais em todas as etapas desta dissertação.

Quero agradecer, também, à minha colega e amiga, Ângela Santos, pela troca de ideias, companheirismo e motivação constante.

Pelo apoio incondicional, agradeço à minha família e aos meus amigos, em particular ao meu namorado, a minha fonte de inspiração, pela paciência infinita e pela força que sempre me transmitiu.

A todos, o meu sincero obrigada.

“Ninguém escapa ao sonho de voar, de ultrapassar os limites do espaço onde nasceu, de ver novos lugares e novas gentes. Mas saber ver em cada coisa, em cada pessoa, aquele algo que a define como especial, um objeto singular, um amigo, é fundamental. Navegar é preciso, reconhecer o valor das coisas e das pessoas, é mais preciso ainda.”

Antoine de Saint-Exupéry in “O Príncipezinho”

Resumo

Ao longo dos anos, os problemas de redes de sensores têm-se estendido, progressivamente, a diversas áreas de aplicação. Nesta dissertação propõe-se estudar o problema de localização de sensores no âmbito da proteção de portos face a eventuais ataques terroristas numa área de interesse contínua. Mais especificamente, o principal objetivo é encontrar a melhor configuração de recursos, ou seja, o número e tipo de sensores a instalar, que maximize a proporção total de área coberta e, conseqüentemente, minimize o risco de uma ameaça não ser detetada – problema mono-objetivo. Posteriormente, inclui-se também na análise do problema a minimização dos custos totais que incluem custos fixos e custos de localização – problema biobjetivo.

A abordagem ao problema considera a introdução de obstáculos na área de interesse. Assim, para além do caso mais simples em que a área de interesse corresponde apenas à região marítima, a existência de obstáculos introduz dois casos de estudo: analisa-se a questão da admissibilidade das soluções mediante as restrições de localização para cada tipo de sensor e estuda-se o problema da linha de visão. Para os três casos em estudo foram obtidas soluções através da aplicação do *biased random-key genetic algorithm*, uma heurística baseada em algoritmos genéticos onde a codificação de cada solução é feita através de chaves aleatórias.

Adotando algumas simplificações, foram testados vários parâmetros de forma a obter os melhores resultados. O problema mono-objetivo, considerando a área de interesse mais simples, permitiu analisar a qualidade da heurística. Posteriormente, foram realizados testes para os restantes casos de estudo e para o problema biobjetivo.

Os resultados obtidos permitiram concluir que esta metodologia é eficaz para o problema em estudo, esperando-se que, futuramente, possa ser aplicada a casos reais, através de algumas adaptações e desenvolvimentos que se sugerem.

Palavras-chave:

Rede de Sensores, Proteção de Portos, Localização Contínua, Admissibilidade, Linha de Visão, Heurística, Chaves Aleatórias

Abstract

Over the years, sensor network issues have been considered for a wide range of applications. In this dissertation, the study of the problem of locating sensors in order to protect ports in a continuous area of interest is proposed. The main goal is to find the best configuration of resources, concerning the number and type of sensors to install so that the total proportion of covered area is maximized. Therefore, the risk of not detecting a threat is minimized – a mono-objective problem. Besides, it is also considered in the analysis of the problem the minimization of total costs which includes fixed costs and costs dependent on location – a bi-objective problem.

The effect of incorporating obstacles in the area of interest is studied. While the simplest case considers that the area of interest corresponds only to the maritime area, the existence of obstacles leads to two study cases: the feasibility of solutions is analysed by the location constraints for each type of sensor and the line of sight problem is also considered. Solutions to all three study cases have been obtained through a biased random-genetic algorithm which codifies each solution by random numbers.

Taking into account some simplifications, different parameters have been tested in order to get the best results. The mono-objective problem considering the simplest area of interest allows the analysis of the heuristic's quality. Various tests were performed considering the remaining study cases and the bi-objective problem.

The results obtained allow us to conclude that this is an effective methodology to the problem studied. In the future, we expect that it can be applied to real cases through some adaptations and developments that are suggested.

Keywords:

Sensor Network, Port Protection, Continuous Location, Feasibility, Line of Sight, Heuristic, Random-Key

Índice

Lista de Figuras	vii
Lista de Tabelas	ix
Lista de Algoritmos	xi
1 Introdução	1
2 Descrição do problema	7
2.1 Principais características do problema	7
2.1.1 Área de interesse	7
2.1.2 Tipos de sensores	8
2.1.3 Critérios a otimizar	10
2.2 Cobertura de um sensor	12
2.2.1 Cobertura Binária	12
2.2.2 Cobertura Probabilística	14
2.3 Custos associados a um sensor	17
2.4 Formulação	18
2.4.1 Formulação do problema mono-objetivo	18
2.4.2 Formulação do problema biobjetivo	18
3 Metodologia de resolução	21
3.1 Algoritmos genéticos e chaves aleatórias	21
3.1.1 <i>Biased random-key genetic algorithm</i>	24
3.2 Problema mono-objetivo	26
3.3 Problema biobjetivo	30
3.3.1 Avaliação das soluções em problemas multiobjetivo	30
3.3.2 Codificação das soluções	34
4 Casos de estudo	41
4.1 Problema sem restrições	41
4.2 Admissibilidade das soluções	42
4.2.1 Método de penalidade	43

4.2.2	Método de reparação	45
4.3	Linha de visão	47
5	Análise e Discussão	51
5.1	Problema mono-objetivo	51
5.1.1	Problema sem restrições	52
5.1.2	Admissibilidade das soluções	58
5.1.3	Linha de visão	64
5.1.4	Análise do comportamento dos <i>sensores_{SC}</i>	68
5.2	Problema biojetivo	71
5.2.1	Otimização dos parâmetros	72
5.2.2	Exemplo	73
6	Conclusões	79
6.1	Principais conclusões	79
6.2	Desenvolvimentos futuros	80
	Referências	85
A	Otimização dos parâmetros	89
A.1	Problema mono-objetivo	89
A.1.1	Parâmetros do <i>biased random-key genetic algorithm</i>	89
A.1.2	Dimensão da população e critério de paragem	93
B	Tabelas de resultados	95
B.1	Valores obtidos pelo BRKGA	95
B.2	Tempo de CPU	97
C	Soluções do problema biobjetivo	99
C.1	Espaço dos objetivos completo	100
C.2	Espaço dos objetivos reduzido	101
D	Funcionamento dos programas	103

Lista de Figuras

1.1	Exemplo do esquema geral de funcionamento das abordagens consideradas.	2
1.2	Dois exemplos de portos marítimos em Portugal.	4
2.1	Exemplos de áreas de interesse.	8
2.2	Algumas orientações possíveis para o mesmo sensor setor circular.	9
2.3	Diferentes tipos de sensores.	10
2.4	Aproximação da proporção total de área coberta.	13
2.5	Importância da precisão δ da grelha.	14
2.6	Cobertura probabilística de um $sensor_{SC}$ utilizando a definição de $Cob(u, s)$ proposta por Akbarzadeh <i>et al.</i> [1].	16
3.1	Funcionamento geral de um algoritmo genético	22
3.2	Descodificador de um algoritmo genético com chaves aleatórias	23
3.3	Funcionamento do RKGA	23
3.4	Cruzamento no BRKGA considerando $\rho_e = 0.7$.	24
3.5	Solução S associada ao vetor K^1 de chaves aleatórias.	29
3.6	Solução S associada ao vetor K^2 de chaves aleatórias.	39
4.1	Introdução de obstáculos - questão da admissibilidade.	42
4.2	Método de penalidade aplicado a uma solução.	44
4.3	Exemplo ilustrativo da desvantagem do método de penalidade.	45
4.4	Aplicação do método de reparação a uma solução do problema.	47
4.5	Redução do campo de visão com a introdução de obstáculos.	48
5.1	Soluções ótimas para dois subproblemas.	53
5.2	Soluções obtidas pelo BRKGA para dois subproblemas.	56
5.3	Erros associados ao valor obtido pelo BRKGA em cada subproblema.	57
5.4	Área de interesse utilizada nos testes realizados.	58
5.5	Análise comparativa dos métodos estudados para a admissibilidade.	59
5.6	Soluções obtidas pelo BRKGA aplicando o método de penalidade.	60
5.7	Evolução do número de soluções não admissíveis ao longo das gerações.	61
5.8	Soluções obtidas pelo BRKGA aplicando o método de reparação.	62
5.9	Análise comparativa dos tempos de CPU no estudo da admissibilidade.	63

5.10	Influência do estudo da linha de visão na análise comparativa dos métodos estudados para a admissibilidade.	65
5.11	Soluções obtidas pelo BRKGA incluindo a análise da linha de visão.	66
5.12	Influência do estudo da linha de visão nos tempos de CPU	67
5.13	Localização de 4 sensores do tipo III.	69
5.14	Localização de 8 sensores do tipo III.	69
5.15	Fronteira potencialmente de Pareto obtida para o exemplo em estudo.	74
5.16	Soluções apresentadas ao decisor.	75
5.17	Fronteira potencialmente de Pareto obtida reduzindo o espaço dos objetivos.	76
5.18	Soluções apresentadas ao decisor com a redução do espaço dos objetivos.	77
6.1	Exemplo de aplicação de uma fase de pesquisa local.	83
A.1	Soluções não dominadas para as combinações de parâmetros p e nG	94
D.1	Parâmetros necessários ao funcionamento do algoritmo.	104
D.2	Dados do problema.	105
D.3	Ficheiro <i>porto.dat</i>	106
D.4	Área de interesse correspondente ao ficheiro da figura D.3.	106

Lista de Tabelas

2.1	Principais características dos problemas em estudo.	11
3.1	Descodificação para a solução K^1	29
3.2	Descodificação para a solução K^2	38
5.1	Valores de referência - Proporção total máxima de área por cobrir.	53
5.2	Proporção total de área por cobrir - caso de estudo sem restrições.	56
5.3	Análise comparativa do tempo de CPU para os vários tipos de sensores. . .	70
A.1	Otimização dos parâmetros do algoritmo.	92
A.2	Otimização da dimensão da população e do número de gerações.	93
B.1	Proporção total de área por cobrir - estudo da admissibilidade das soluções aplicando o método de penalidade.	95
B.2	Proporção total de área por cobrir - estudo da admissibilidade das soluções aplicando o método de reparação.	96
B.3	Proporção total de área por cobrir - análise da linha de visão incluída no estudo da admissibilidade das soluções aplicando o método de penalidade. .	96
B.4	Proporção total de área por cobrir - análise da linha de visão incluída no estudo da admissibilidade das soluções aplicando o método de reparação. . .	96
B.5	Tempo médio de CPU (em segundos) - caso de estudo sem restrições.	97
B.6	Tempo médio de CPU (em segundos) - estudo da admissibilidade das soluções aplicando o método de penalidade.	97
B.7	Tempo médio de CPU (em segundos) - estudo da admissibilidade das soluções aplicando o método de reparação.	98
B.8	Tempo médio de CPU (em segundos) - análise da linha de visão incluída no estudo da admissibilidade das soluções aplicando o método de penalidade. 98	
B.9	Tempo médio de CPU (em segundos) - análise da linha de visão incluída no estudo da admissibilidade das soluções aplicando o método de reparação. 98	
C.1	Soluções potencialmente não dominadas - espaço dos objetivos completo. . .	100
C.2	Soluções potencialmente não dominadas - espaço dos objetivos reduzido. . .	101

Lista de Algoritmos

1	Determinação da cobertura binária de uma solução	14
2	Determinação da cobertura probabilística de uma solução	16
3	Cruzamento utilizado no BRKGA	25
4	<i>Biased random-key genetic algorithm</i>	25
5	Descodificador para o problema mono-objetivo	28
6	Determinar o <i>rank</i> das soluções de uma população	33
7	Reduzir o número de soluções potencialmente não dominadas	34
8	Descodificador para o problema biobjetivo	37

*“O mais importante de tudo é nunca deixar de se perguntar.
A curiosidade tem a sua própria razão de existir.”*

Albert Einstein

Capítulo 1

Introdução

A crescente necessidade de monitorizar e prever desastres de várias naturezas numa sociedade em constante crescimento, acompanhando a evolução tecnológica, estendeu, progressivamente, o desenvolvimento de redes de sensores estudado nas áreas militares e de defesa a outras áreas de aplicação. As primeiras aplicações de redes de sensores surgiram, essencialmente, a partir dos anos setenta no decorrer da Guerra Fria: um sistema de sensores acústicos foi instalado em localizações estratégicas no fundo do oceano para detetar submarinos soviéticos e radares de defesa aérea foram desenvolvidos para vigiar a zona continental dos Estados Unidos da América e do Canadá. No trabalho de Chong *et al.* [6] é feita uma síntese da investigação realizada ao longo dos anos para os problemas de redes de sensores e salienta-se a importância do estudo destes problemas em várias vertentes. Atualmente são diversas as áreas em que se aplicam redes de sensores, nomeadamente na saúde, vigilância ambiental, monitorização de trânsito e controlo de tráfego aéreo.

Numa lógica de defesa contra ameaças do tipo terrorista surge a necessidade de adotar medidas de vigilância e segurança para a proteção de portos com o objetivo de prevenir eventuais danos que podem ocorrer com consequentes prejuízos monetários, humanos, ambientais e sociais. É precisamente neste sentido que surge o projeto SAFEPOR, um projeto ambicioso e de grande dimensão promovido pela NATO e com a contribuição da Marinha Portuguesa, que pretende desenvolver um sistema de apoio à decisão para a defesa de portos. Este projeto pretende abordar o problema complexo de determinar a melhor configuração de recursos com o intuito de monitorizar e proteger uma determinada zona portuária. Mais especificamente, entre outros objetivos, pretende-se determinar o número e tipo de sensores e a respetiva localização de forma a garantir a máxima proteção da área de interesse [17]. Tendo como motivação o problema introduzido pelo projeto SAFEPOR, a abordagem que se apresenta neste trabalho procura desenvolver alguns dos aspetos fundamentais do problema de localização de sensores. O problema que se propõe estudar tem como objetivo localizar numa área de interesse contínua diferentes tipos de sensores, assegurando a proteção da área marítima em estudo e tendo, também, em consideração o custo total envolvido na localização dos recursos.

Na presente dissertação são consideradas algumas simplificações na abordagem do problema. A área de interesse contínua onde se pretende localizar os sensores corresponde ao quadrado unitário $[0, 1]^2$ e os tipos de sensores existentes são sonares representados pela forma da área que conseguem cobrir: existem sensores cuja área de cobertura é circular e sensores que cobrem uma área correspondente a um setor circular. Este último caso é um aspeto inovador no trabalho que se apresenta uma vez que, nos trabalhos anteriormente desenvolvidos no âmbito do projeto, apenas se considera a existência de sensores cuja área de cobertura corresponde a um círculo.

O problema em estudo é abordado em duas perspetivas: o problema mono-objetivo e o problema biobjetivo tal como ilustra a figura 1.1 que apresenta um exemplo do esquema

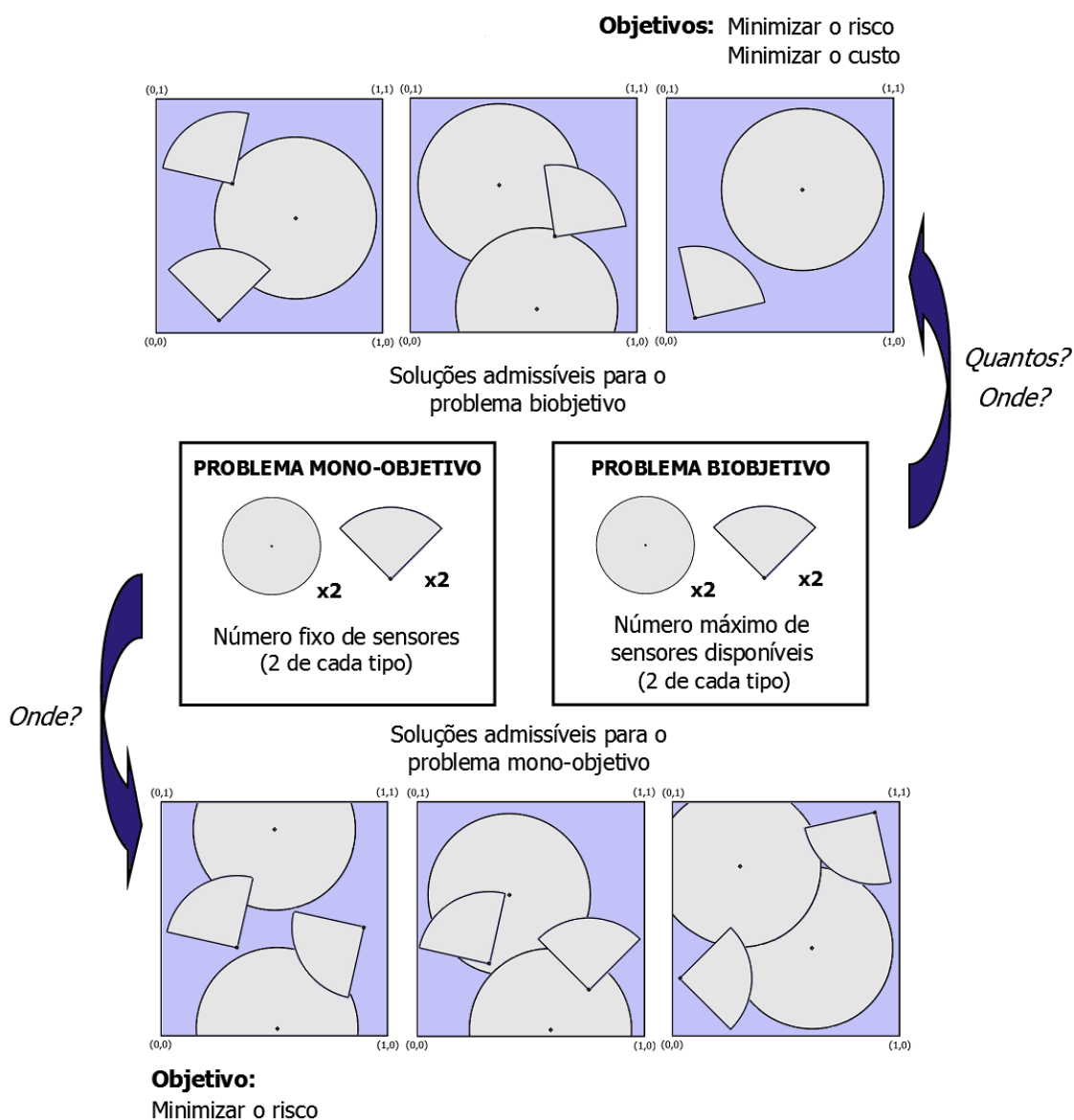


Figura 1.1: Exemplo do esquema geral de funcionamento das abordagens consideradas.

geral de funcionamento das respectivas abordagens. No primeiro caso, partindo de uma configuração que explicita quantas unidades são usadas de cada tipo de sensores apenas se pretende saber *onde* localizar os mesmos. O objetivo é minimizar o *risco* de uma solução, ou seja, a probabilidade de uma ameaça não ser detetada. Cada sensor consegue detetar a ocorrência de determinado evento de acordo com um alcance de deteção, estando a definição de *cobertura* intrinsecamente relacionada com a noção de risco. Neste trabalho considera-se que a definição de cobertura de uma solução S , $Cob(S)$, se relaciona com a definição de risco através da expressão: $Cob(S) = 1 - Risco(S)$. Por outro lado, no problema biobjetivo pretende-se, sem configurações pré-definidas, estudar o problema de localização de sensores tendo em conta, para além da minimização do risco, a minimização do custo total da solução que engloba custos fixos de aquisição e custos variáveis dependendo, por exemplo, da distância a que um determinado sensor se encontra de um gerador de energia. Estes objetivos são conflituosos entre si: à medida que o número de sensores aumenta, espera-se obter uma menor proporção de área marítima por cobrir mas, também, um aumento no custo total da solução. Este problema é mais complexo pois, para além de ter em consideração dois objetivos que se pretendem minimizar, existem, também, duas decisões a tomar: *quantos* sensores utilizar e *onde* fazer a respetiva localização. Neste caso, conhece-se apenas o número máximo de unidades disponíveis para cada tipo de sensor e, eventualmente, pode também existir interesse em exigir que um número mínimo de sensores com determinadas características tenha de ser instalado.

Os contributos de Moniz [19] e Silva [26] no âmbito do projeto SAFEPOR, consideram o caso mais simples em que a área de interesse é apenas constituída por mar e pretende-se monitorizar toda a região. Porém, o problema de localização de sensores torna-se mais realista se forem introduzidos obstáculos na área de interesse que, consoante o contexto do problema em causa, podem representar realidades bastante distintas. Tendo como motivação para o problema em estudo a proteção de portos, é bastante intuitivo pensar que a área de interesse engloba duas zonas: o porto propriamente dito onde existem diversas instalações e onde os barcos e navios atracam para cargas e descargas e a superfície marítima envolvente, conforme ilustra figura 1.2. Os obstáculos representam, assim, a costa ou zonas de terra que possam existir no interior da região marítima.

A abordagem apresentada nesta dissertação, ao considerar a introdução de obstáculos, pretende ser mais abrangente e mais próxima da realidade do problema em estudo. A área que se pretende monitorizar corresponde, agora, à zona marítima que exclui os obstáculos. Ao contrário do que acontece no caso mais simples em que a área de interesse corresponde apenas à região marítima, a existência de obstáculos introduz duas questões adicionais ao problema em estudo. Em primeiro lugar, assume-se que, mediante as características de cada sensor existem restrições em relação à respetiva localização. Alguns sensores só podem ser localizados no mar enquanto que outros estão preparados para ser colocados em terra. Sem perda de generalidade irá assumir-se que os sensores considerados apenas podem ser instalados na zona marítima. Neste caso, existem locais da área de interesse

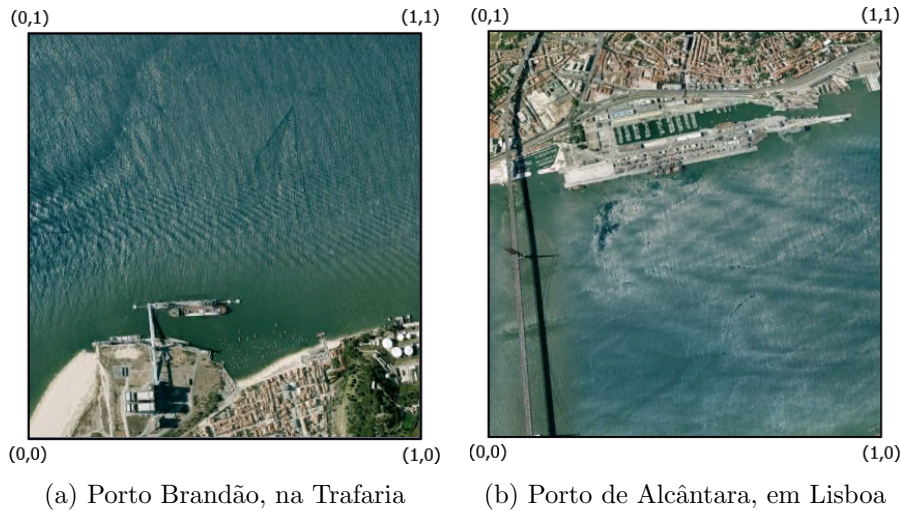


Figura 1.2: Dois exemplos de portos marítimos em Portugal.

em que não é permitida a colocação dos meios de vigilância, sendo necessário analisar a admissibilidade das soluções que se obtêm. Surge, também, a questão da linha de visão, ou seja, algumas zonas que anteriormente eram detetadas por um sensor por se encontrarem dentro da área que o mesmo consegue atingir podem agora não o ser, caso algum obstáculo obstrua a linha de visão que o sensor estabelece com esses pontos.

Tendo em conta as considerações feitas no parágrafo anterior, serão apresentados três casos de estudo distintos nesta dissertação: o caso mais simples, em que toda a área de interesse corresponde à região marítima e, com a introdução de obstáculos, analisam-se os casos de estudo que consideram a questão da admissibilidade e da linha de visão.

Devido à complexidade do problema enunciado, foi adotada uma técnica heurística com o objetivo de determinar uma solução admissível, de preferência com valor próximo do valor da solução ótima. Mais especificamente, recorreu-se à aplicação de um algoritmo genético cuja codificação das soluções é feita através de chaves aleatórias, revelando-se uma abordagem eficaz na resolução do problema de localização de sensores no espaço contínuo. A metodologia que se apresenta, designada por *biased random-key genetic algorithm* [12], tem uma característica muito interessante que resulta da clara divisão existente entre uma componente independente do problema em estudo e a componente específica. Assim, apenas é necessário focar a atenção na segunda componente o que permite um estudo mais direcionado para o problema em análise. Tirando partido desta versatilidade e da forma como as soluções são codificadas, os três casos de estudo anteriormente apresentados são analisados aplicando esta mesma metodologia que é facilmente adaptada tendo em conta as características específicas de cada situação.

A adaptabilidade da metodologia anteriormente referida face a diferentes vertentes do problema de localização de sensores foi um dos fatores decisivos para a sua escolha. Existem, porém, muitas outras abordagens possíveis que poderiam ter sido desenvolvidas.

Na literatura encontram-se diferentes metodologias utilizadas para o problema de localização de sensores, como por exemplo o método de *simulated annealing* que é utilizado por Akbarzadeh *et al.* [1, 2] para uma variante do problema que se estuda neste trabalho considerando igualmente a introdução de obstáculos na área de interesse. O método de *particle swarm* é utilizado por Ngatchou *et al.* [21] numa abordagem multiobjetivo do problema e por Silva [26] para comparar os resultados obtidos aplicando este método com os resultados do algoritmo de atração que desenvolveu.

O problema mono-objetivo surge como motivação para o problema biobjetivo mais complexo, com o intuito de introduzir a metodologia adotada para o problema e de avaliar a qualidade do procedimento desenvolvido comparando os resultados obtidos com os resultados de outros trabalhos para o problema mono-objetivo no caso mais simples (em que a área de interesse corresponde apenas à zona marítima).

Existem várias abordagens possíveis para resolver o problema biobjetivo. Moniz [19] considera, na sua dissertação, que o problema pode ser visto como um problema binível em que num “nível superior” se pretende otimizar a configuração dos recursos e num “nível inferior” otimiza-se a localização dos recursos dessa configuração. A metodologia desenvolvida no presente trabalho pretende ser inovadora ao responder simultaneamente às questões de *quantos* sensores utilizar e *onde* os localizar o que é possível pela codificação das soluções através de chaves aleatórias. Assim, com uma única execução do algoritmo genético proposto é possível obter um conjunto de *soluções potencialmente não dominadas* tendo em conta os dois objetivos em estudo. Para uma breve explicação destes conceitos considerem-se duas soluções A e B para o problema de localização de sensores tais que:

$$\text{Solução A} : \text{Risco}(A) = 0.06, \text{Custo}(A) = 6.5$$

$$\text{Solução B} : \text{Risco}(B) = 0.01, \text{Custo}(B) = 4.5$$

A solução B domina a solução A uma vez que B apresenta valores não piores em ambos os objetivos e existe pelo menos um objetivo em que o valor de B é melhor que o respetivo valor de A (neste caso, B é melhor em ambos os objetivos). Para um conjunto de soluções, as *soluções não dominadas* são aquelas para as quais não existe nenhuma solução que as domine. O conceito de soluções potencialmente não dominadas é introduzido uma vez que as soluções são obtidas através de uma heurística, não se conhecendo todas as soluções possíveis para o problema. Na seção 3.3.1 estes conceitos serão introduzidos com mais detalhe e fazem-se algumas considerações sobre a aplicação de algoritmos genéticos a problemas em que se pretende otimizar mais do que um critério.

Os capítulos que se seguem descrevem, detalhadamente, os aspetos anteriormente mencionados. Apesar das considerações que serão feitas terem como motivação a proteção de portos, podem ser facilmente adaptadas para outras áreas de aplicação. O problema em estudo é descrito no capítulo 2, seguindo-se a apresentação da metodologia adotada para a

resolução do mesmo, no capítulo 3. No capítulo 4 apresentam-se os casos e estudo e os resultados dos testes realizados analisam-se e discutem-se no capítulo 5. No último capítulo são feitas as considerações finais, onde se incluem as principais conclusões e são focados alguns aspetos que podem ser desenvolvidos futuramente para melhorar ou complementar o trabalho apresentado nesta dissertação.

Capítulo 2

Descrição do problema

O problema de localização de sensores pode ser estudado em diferentes áreas de aplicação, conforme foi referido no capítulo introdutório. No presente capítulo, será feita a descrição do problema em estudo e, apesar de grande parte das considerações que se apresentarão não se restringir à proteção de portos e ser válida noutras vertentes do problema, esta é a principal motivação para o trabalho que se apresenta e serão feitas algumas referências específicas ao tema ao longo da presente dissertação. Depois de apresentadas as principais características do problema geral, introduzem-se as particularidades do problema mono-objetivo e do problema biobjetivo que se propõe estudar.

2.1 Principais características do problema

Existem muitas questões relacionadas com o problema de localização de sensores e diversas variantes que podem ser consideradas. Assim, é essencial definir as principais características que definem o problema em estudo nesta dissertação. Serão adotadas algumas simplificações que, não comprometendo a qualidade dos resultados obtidos, permitem que os métodos adotados sejam executados em menor tempo computacional.

2.1.1 Área de interesse

Na abordagem que se irá adotar para o problema de localização de sensores pretende-se que sejam detetadas eventuais ameaças que surjam na área marítima acima do nível da água do mar como, por exemplo, embarcações que se aproximem da zona portuária. A aproximação de mergulhadores ou outro tipo de ameaças que possam surgir em profundidade não são tidas em consideração.

Considera-se que a área de interesse A , ou seja, a área na qual reside a preocupação na deteção de ameaças, corresponde ao quadrado unitário $[0, 1]^2$. Esta simplificação não restringe o estudo do problema a determinadas zonas portuárias, uma vez que qualquer zona que se pretenda vigiar pode ser ajustada à forma da área de interesse que se considera, tal como ilustram as figuras 1.2a e 1.2b anteriormente apresentadas no capítulo 1, onde o quadrado $[0, 1]^2$ se ajustou ao porto Brandão e ao porto de Alcântara. O objetivo

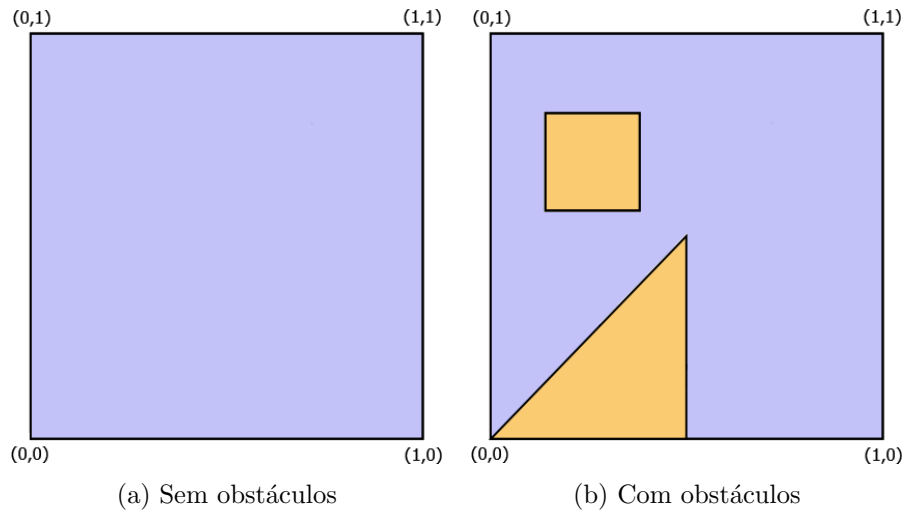


Figura 2.1: Exemplos de áreas de interesse.

é concentrar o estudo numa área mais restrita e tornar as metodologias adotadas independentes do local particular que se pretende monitorizar uma vez que as coordenadas dos pontos que constituem a área de interesse são convertidas para valores entre 0 e 1.

A área de interesse pode corresponder simplesmente à área marítima onde se pretendem detetar eventuais ameaças ou pode ser constituída, para além das zonas de mar, por obstáculos que definem zonas terrestres. Nesta dissertação, assume-se que os obstáculos são modelados como polígonos de dimensão finita no plano e não têm falhas no seu interior. Supondo que existem n obstáculos em A e que cada obstáculo i ocupa a área Obs_i , a área total ocupada por obstáculos é dada por $O = \bigcup_{i \in \{1..n\}} Obs_i$. A área que se pretende proteger continua a ser apenas o mar, não incluindo os obstáculos, ou seja, pretende-se monitorizar a área correspondente a $A - O$. Na figura 2.1, em que a região marítima se encontra representada a azul e os obstáculos correspondem aos polígonos coloridos a bege, são apresentadas duas áreas de interesse que se distinguem pela introdução de obstáculos numa delas.

2.1.2 Tipos de sensores

Tratando-se de um problema de localização de sensores é fundamental definir quais os tipos de sensores que se consideram disponíveis para monitorizar a área de interesse anteriormente definida. Existem sensores com características muito variadas e uma das formas de proceder à sua caracterização é através do tipo de energia que detetam. Apesar de estar fora do âmbito deste trabalho caracterizar detalhadamente o tipo de sensores que podem ser instalados, considera-se que os sensores utilizados são sensores de distância acústicos, mais precisamente sonares, do inglês *Sound Navigation And Ranging*. Estes dispositivos detetam e localizam objetos através das ondas sonoras que os alvos refletem ou produzem e, apesar do seu funcionamento ser análogo ao funcionamento de um radar, distinguem-se

destes uma vez que os radares utilizam ondas de rádio (que se propagam apenas alguns poucos metros sob a água). Para efeitos da presente dissertação, os tipos de sensores utilizados serão apenas distinguidos tendo em conta a área de deteção que conseguem abranger. Assim, serão utilizados sensores cuja área de deteção corresponde a um círculo e sensores que definem uma área correspondente a um setor circular. Os dois tipos de sensores considerados serão referidos, daqui em diante, respetivamente por sensores circulares ($sensores_C$) e sensores setor circular ($sensores_{SC}$). Considera-se, também, que os sensores são localizados numa posição fixa.

A grande maioria dos trabalhos realizados para problemas de localização de sensores, nomeadamente os trabalhos desenvolvidos anteriormente no âmbito do projeto SAFE-PORT consideram apenas a existência de sensores circulares. Estes sensores distinguem-se entre si através do raio, r_s , de cada sensor s , podendo existir sensores circulares com maior ou menor área de deteção. A introdução dos $sensores_{SC}$ no problema em estudo surgiu com o intuito de tirar partido da versatilidade da forma de um setor circular, o que é bastante interessante para a maximização da área coberta conforme se irá referir posteriormente. Para além do raio, r_s , associado a cada $sensor_{SC}$ s , existe também um ângulo de visão, a_s (em radianos), que o caracteriza. Estes sensores podem estar orientados em diferentes direções. Esta propriedade, intrínseca ao funcionamento de um sensor deste tipo, não é previamente conhecida ao contrário do raio e do ângulo de visão: é adaptada consoante for mais conveniente, em cada situação em particular, tendo em consideração os critérios que se pretendem otimizar. Osais *et al.* [22] utilizam este tipo de sensores num problema de minimização do custo de localização destes meios de vigilância e, consoante a amplitude de visão de cada sensor, as orientações que podem ser consideradas encontram-se pré definidas. Neste trabalho, pretende-se que cada sensor possa ser orientado em qualquer direção sem pré definições. Assim, a orientação, o_s , corresponde à amplitude (em radianos) do ângulo definido a partir do eixo horizontal no sentido contrário ao sentido dos ponteiros do relógio, conforme ilustra a figura 2.2.

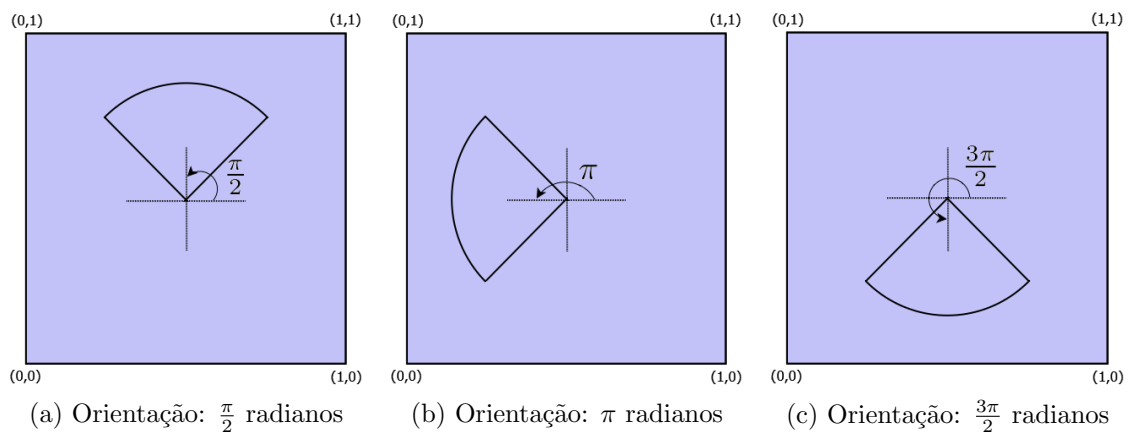


Figura 2.2: Algumas orientações possíveis para o mesmo sensor setor circular.

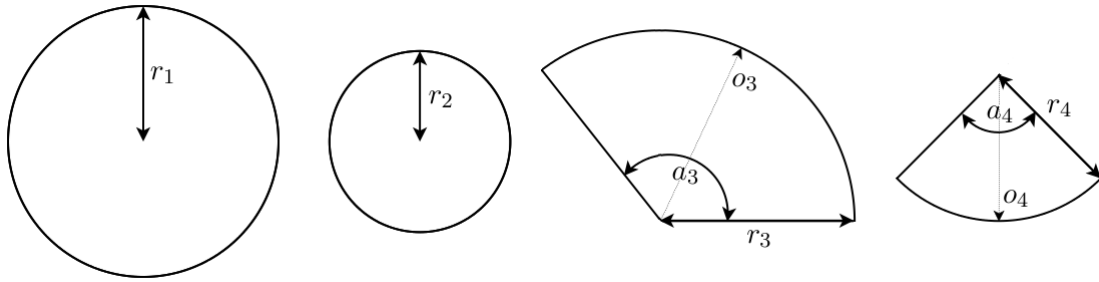


Figura 2.3: Diferentes tipos de sensores.

Tanto os *sensores_C* como os *sensores_{SC}* são fixos nas coordenadas dos respectivos centros, não existindo rotação sobre esse ponto. Esta questão é essencialmente relevante para o caso dos *sensores_{SC}*, uma vez que se fosse permitida esta rotação, diferentes zonas da área de interesse seriam monitorizadas em diferentes intervalos de tempo. Apesar de ser uma questão muito interessante é, também, bastante complexa pelo que não será considerada, sugerindo-se como desenvolvimento futuro.

A figura 2.3 apresenta diversos tipos de sensores que variam nas características anteriormente mencionadas.

2.1.3 Critérios a otimizar

Existem, essencialmente, duas questões a que o problema de localização de sensores pretende responder. Em primeiro lugar é necessário determinar uma configuração, ou seja, explicitar *quantos* sensores são utilizados de cada tipo e, em segundo lugar, indicar *onde* localizar esses recursos na área de interesse. Uma solução para o problema de localização de sensores consiste, assim, numa configuração para a qual se indica as coordenadas correspondentes aos centros dos sensores que se pretendem instalar. Numa solução em que existam *sensores_{SC}* indica-se, também, a orientação de cada um.

O problema em estudo é um *problema de localização contínua* uma vez que a instalação dos sensores não está restrita a pontos específicos da área de interesse. A inclusão de obstáculos restringe a localização dos sensores à zona que os exclui, porém, a abordagem é a mesma sendo adotados mecanismos especiais para garantir a admissibilidade das soluções, conforme se irá referir no capítulo 4.

São dois os principais critérios a ter em consideração no problema de localização de sensores. A avaliação do *risco* de uma solução, ou seja, a probabilidade de uma ameaça não ser detetada, é um critério fundamental uma vez que permite analisar a qualidade da monitorização da área de interesse. As soluções do problema devem igualmente ser avaliadas no que diz respeito ao custo total envolvido na aquisição e instalação dos meios de vigilância. Nesta dissertação, os critérios mencionados são estudados em dois problemas distintos: o problema mono-objetivo em que apenas se tem em consideração a avaliação do risco das soluções e o problema biobjetivo em que se avalia também o custo. Este

último problema, apesar da sua complexidade acrescida, é a abordagem de maior interesse e proximidade com a realidade.

O problema mono-objetivo é apresentado neste trabalho uma vez que a metodologia adotada para a resolução do problema, apresentada no capítulo 3, é mais intuitiva quando, partindo de um número fixo de sensores, se pretende apenas minimizar o risco. Assim, depois de compreendido o procedimento para o caso mais simples, torna-se mais perceptível a adaptação feita para o caso biobjetivo. Além disso, o estudo da minimização do risco tem, por si só, especial interesse neste problema e, muitas das abordagens anteriores preocupam-se apenas em otimizar este critério. Neste sentido, o problema mono-objetivo torna-se essencial para avaliar a qualidade da metodologia adotada uma vez que existem resultados para o mesmo problema provenientes de outros trabalhos com os quais pode ser feita uma análise comparativa. O problema biobjetivo é posteriormente introduzido e, conhecendo as limitações no número de cada tipo de sensor existente, foi adotada uma metodologia inovadora para este tipo de problema em que se pretende responder simultaneamente às duas questões inicialmente formuladas.

A tabela 2.1 sintetiza as principais características de cada problema.

	Problema mono-objetivo	Problema biobjetivo
Decisões a tomar	<i>Onde?</i>	<i>Quantos? Onde?</i>
Área de interesse	Quadrado $[0, 1]^2$	Quadrado $[0, 1]^2$
Tipos de sensores	O número de sensores de cada tipo que se pretende instalar é conhecido <i>a priori</i> .	Conhece-se o número máximo de unidades disponíveis e o número mínimo de recursos a instalar para cada tipo de sensores. Pode ser instalada qualquer quantidade dentro dos respetivos limites.
Critérios a otimizar	<ul style="list-style-type: none"> • Minimização do risco 	<ul style="list-style-type: none"> • Minimização do risco • Minimização do custo

Tabela 2.1: Principais características dos problemas em estudo.

Na seção 2.2 são apresentadas as principais considerações a ter em conta para a determinação do risco de uma solução: o conceito de cobertura, estando intrinsecamente relacionado com a definição de risco, é fundamental nesta abordagem. Em relação ao critério do custo de uma solução, a seção 2.3 descreve as definições adotadas. No final do presente capítulo, apresentam-se as formulações formais dos problemas em estudo que englobam toda a informação mencionada.

2.2 Cobertura de um sensor

Um aspeto fundamental que surge no âmbito dos problemas de redes de sensores é a determinação da cobertura. A cobertura de um sensor pode ser considerada uma medida da qualidade do seu funcionamento, uma vez que um nível elevado de cobertura da área de interesse significa que a região se encontra bem monitorizada. Existem, essencialmente, três classes para classificar os problemas de cobertura: cobertura de barreira, cobertura de pontos e cobertura de área. A cobertura de barreira envolve a colocação de sensores ao longo de zonas de fronteira de uma região com o intuito de detetar a presença de intrusos. A cobertura de pontos consiste na vigilância de um conjunto de pontos relevantes e, por fim, a cobertura de área refere-se à localização de sensores de forma a monitorizar toda a área de interesse. É, precisamente, sobre esta última classe que o presente trabalho se irá focar com o objetivo de minimizar a proporção total de área por cobrir.

Para resolver o problema de cobertura de um sensor adotou-se uma estratégia bastante comum na literatura: ajustar uma grelha de pontos à área de interesse [3]. Num problema de redes de sensores, esta grelha pode ser utilizada de duas formas: para calcular a cobertura e para determinar a posição dos sensores. Neste caso, estamos apenas interessados em obter o valor da cobertura, não restringindo a localização destes meios de vigilância aos pontos da grelha.

Tendo em conta a definição de risco adotada neste trabalho, a cobertura de uma solução S , $Cob(S)$, para o problema de localização de sensores relaciona-se com o valor do risco através da expressão 2.1.

$$\begin{aligned} Cob(S) &= 1 - P(\text{Uma ameaça não ser detetada}) \\ &= 1 - Risco(S) \end{aligned} \tag{2.1}$$

Apresentam-se, de seguida, dois modelos distintos para determinar a cobertura de um sensor: o modelo binário e o modelo probabilístico que, ao ser mais realista, é mais complexo que o anterior.

2.2.1 Cobertura Binária

A determinação da cobertura binária de um sensor consiste simplesmente em determinar a proporção de área coberta, ou seja, a razão entre a área coberta e a área total de interesse. Porém, determinar a área exata coberta pelos sensores torna-se uma tarefa difícil, especialmente se existirem formas irregulares que surgem quando existe sobreposição de sensores. O método da grelha de pontos simplifica este problema uma vez que a cobertura passa a ser avaliada apenas nos pontos da grelha, permitindo obter uma aproximação da proporção de área coberta de acordo com a expressão 2.2. Recorde-se que, para o problema em estudo, apenas estamos interessados em determinar a cobertura dos sensores no mar, pelo que apenas os pontos da grelha localizados na região marítima são considerados caso existam obstáculos na área de interesse.

$$Cob(S) = \frac{\text{N}^\circ \text{ de pontos da grelha cobertos no mar}}{\text{N}^\circ \text{ total de pontos da grelha localizados no mar}} \quad (2.2)$$

A figura 2.4 ilustra a utilização da expressão anterior para duas soluções do problema de localização de sensores considerando, respetivamente, a área de interesse simples e com introdução de obstáculos. Na figura 2.4b, os pontos assinalados a vermelho, por se encontrarem no interior dos obstáculos não são considerados no cálculo da cobertura.

Mais especificamente, a cobertura binária de um ponto u da grelha de pontos através de um sensor $s \in S$, $Cob(u, s)$, determina-se através do algoritmo 1 que se divide em dois casos, consoante a área de cobertura do sensor s representa um círculo ou um setor circular. No caso de se tratar de um $sensor_{SC}$ é necessário determinar, para cada ponto u , o ângulo formado pelo segmento $[u, centro_s]$ e o eixo horizontal, θ_u , de forma análoga à determinação da orientação do sensor.

É essencial ter em conta que a precisão, δ , da grelha utilizada influencia a qualidade do valor obtido: quanto menor o tamanho de cada célula da grelha, melhor a aproximação da proporção de área total que se obtém, conforme ilustra a figura 2.5. Na figura 2.5a, em que a precisão da grelha é de $\delta = 0.25$, a proporção total de área coberta é 1 uma vez que todos os pontos da grelha se encontram cobertos. Porém, como facilmente se pode verificar, a área de interesse não se encontra totalmente coberta e ao refinar a grelha utilizando uma precisão de $\delta = 0.1$ obtém-se uma proporção de área coberta de 0.917 que corresponde a uma aproximação mais rigorosa.

Como se verá no capítulo 5, é necessário definir um valor ponderado para δ tendo em conta a precisão do valor obtido e o tempo de processamento, que aumenta para valores menores de δ .

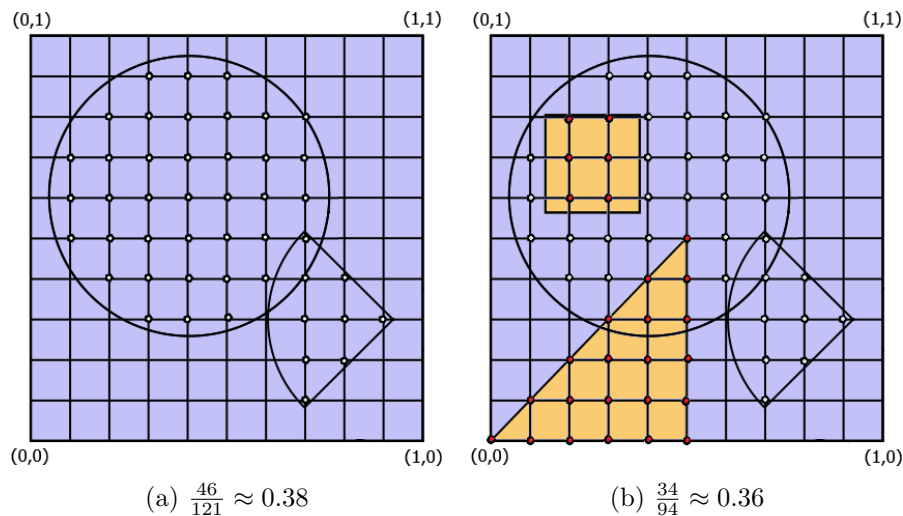


Figura 2.4: Aproximação da proporção total de área coberta.

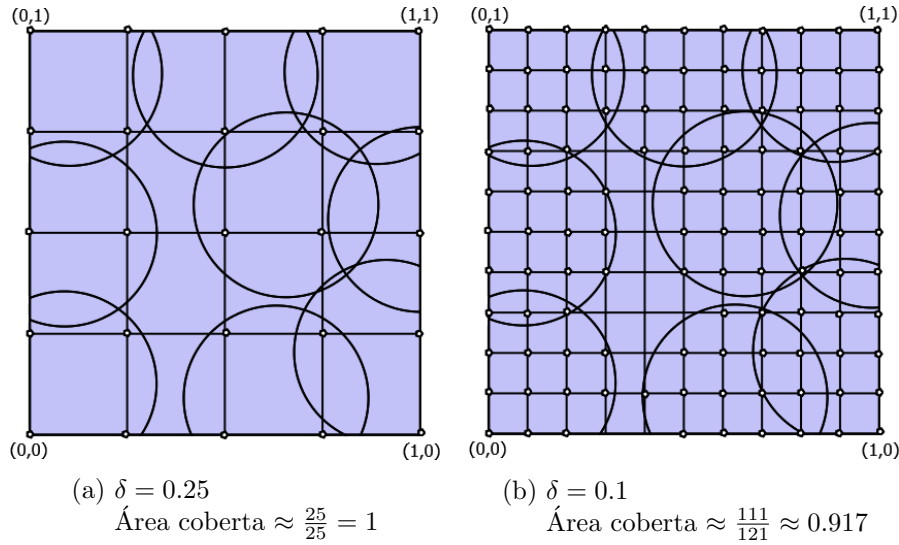


Figura 2.5: Importância da precisão δ da grelha.

Algoritmo 1 Determinação da cobertura binária de uma solução

Input: Solução S

$Cob(S) = coberturaBinaria(S)$

```

1:  $pontosMar \leftarrow 0, pontosCobertos \leftarrow 0$ 
2: for cada ponto  $u$  da grelha de pontos do
3:   if  $u$  está localizado no mar then
4:      $pontosMar \leftarrow pontosMar + 1$ 
5:     for cada sensor  $s \in S$  do
6:        $Cob(u, s) = 0$ 
7:       if  $s$  é  $sensor_C$  and  $\|u - centro_s\| \leq r_s$  then
8:          $Cob(u, s) = 1$ 
9:       end if
10:      if  $s$  é  $sensor_{SC}$  and  $\|u - centro_s\| \leq r_s$  and  $o_s - \frac{a_s}{2} \leq \theta_u \leq o_s + \frac{a_s}{2}$  then
11:         $Cob(u, s) = 1$ 
12:      end if
13:    end for
14:  end if
15: end for
16: for cada ponto  $u$  da grelha de pontos do
17:  if  $\exists s \in S : Cob(u, s) = 1$  then
18:     $pontosCobertos \leftarrow pontosCobertos + 1$ 
19:  end if
20: end for

```

Output: $Cob(S) = \frac{pontosCobertos}{pontosMar}$

2.2.2 Cobertura Probabilística

Em muitos trabalhos já desenvolvidos para problemas de localização de sensores é adotado o modelo de cobertura binária para avaliar o risco de uma solução. Assim, nas zonas em que os sensores conseguem detetar ameaças, a capacidade de deteção dos mesmos é uni-

forme. Fora da área que conseguem cobrir a probabilidade de uma ocorrência ser detetada é nula. Apesar de ser um modelo pouco complexo e bastante intuitivo, na prática não é razoável considerar que a probabilidade associada ao sinal de deteção de um sensor é reduzida abruptamente de 1 para 0. É mais realista considerar que a redução na capacidade de detetar ameaças é gradual e que existe alguma probabilidade de detetar ocorrências para além da área pré-definida (circular ou setor circular) anteriormente considerada. Ao ignorar esta capacidade, o modelo binário pode estar a considerar a instalação de mais sensores do que na realidade é necessário o que se reflete num aumento desnecessário do custo total da solução para o problema biobjetivo.

Hefeeda *et al.* [15] apresentam uma breve revisão bibliográfica de alguns modelos de cobertura probabilística e salientam o facto de que estes modelos capturam, de melhor forma, o comportamento de deteção de um sensor e são mais realistas na medida em que os fenómenos que se pretendem detetar, as condições ambientais e a conceção dos próprios sensores são de natureza estocástica. Existem, essencialmente, duas principais desvantagens nesta abordagem para a cobertura dos sensores. Em primeiro lugar, a capacidade de deteção deixa de ter um comportamento uniforme e, neste sentido passa a existir uma preocupação adicional em definir a noção de sobreposição dos sensores. Em segundo lugar, a definição tradicional de cobertura tal que cada ponto da área de interesse deve estar dentro da capacidade de alcance de pelo menos um sensor deixa de ser válida devido à natureza probabilística da deteção que agora se considera. Assim, é necessário adotar uma nova definição de cobertura a partir do momento que se recorre ao modelo probabilístico para fazer a avaliação do risco num ponto e, em geral, de uma solução do problema.

O valor da cobertura de um ponto u da grelha através de um sensor s da solução S em análise, $Cob(u, s)$, é determinado pela função que define o modelo probabilístico adotado, existindo uma grande variedade de funções que pretendem refletir o comportamento de deteção de um sensor. Akbarzadeh *et al.* [1, 2] apresentam uma abordagem muito interessante uma vez que, ao contrário da maioria dos trabalhos existentes na literatura, a função que sugerem está adaptada para *sensores_{SC}* (figura 2.6) e considera também a visibilidade dos pontos sob a presença de obstáculos. Chung *et al.* [7], num trabalho sobre a deteção de ameaças numa região marítima através de patrulheiros que se movimentam ao longo da área de interesse, consideram o modelo de *Poisson Scan* para caracterizar o comportamento de deteção dos sensores.

A partir do valor de $Cob(u, s)$ determina-se a cobertura do ponto u , conforme se apresenta na expressão 2.3. Tal como acontecia para o caso binário, a cobertura de um ponto é apenas determinada se o mesmo se localizar no mar, sendo nula caso contrário.

O valor da cobertura da solução S , $Cob(S)$ determina-se através da média aritmética dos valores $Cob(u)$ para os quais u está localizado no mar.

$$Cob(u) = 1 - \prod_{\forall s \in S} (1 - Cob(u, s)) \quad (2.3)$$

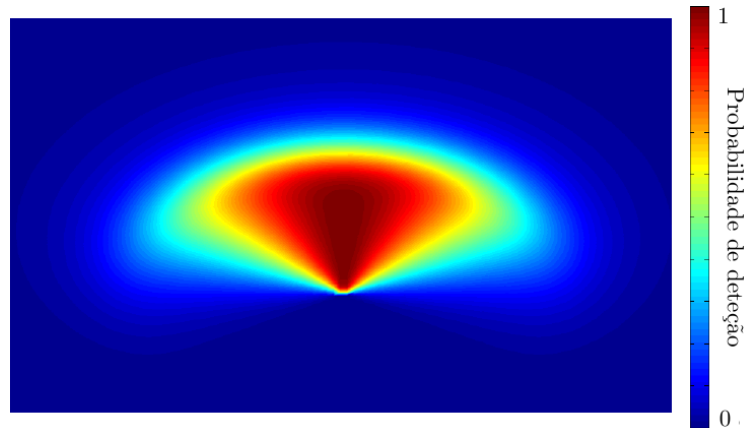


Figura 2.6: Cobertura probabilística de um $sensor_{SC}$ utilizando a definição de $Cob(u, s)$ proposta por Akbarzadeh *et al.* [1].

Os passos principais da determinação da cobertura probabilística anteriormente mencionados, encontram-se sintetizados no algoritmo 2. Note-se que o valor escolhido para a precisão da grelha de pontos continua a ser muito importante para garantir o rigor dos resultados obtidos.

Algoritmo 2 Determinação da cobertura probabilística de uma solução

Input: Solução S

$Cob(S) = coberturaProbabilistica(S)$

1: $pontosMar \leftarrow 0$

2: **for** cada ponto u da grelha de pontos **do**

3: **if** u está localizado no mar **then**

4: $pontosMar \leftarrow pontosMar + 1$

5: **for** cada sensor $s \in S$ **do**

6: Determinar $Cob(u, s)$ de acordo com o modelo probabilístico adotado

7: **end for**

8: **else**

9: $Cob(u, s) = 0, \forall s \in S$

10: **end if**

11: $Cob(u) = 1 - \prod_{\forall s \in S} (1 - Cob(u, s))$

12: **end for**

Output: $Cob(S) = \frac{\sum_{u \in grelha} Cob(u)}{pontosMar}$

Nesta dissertação, não se pretende explorar a aplicação de um modelo probabilístico. Assim, os exemplos ilustrativos dos casos de estudo e os testes experimentais efetuados dizem, apenas, respeito à aplicação do modelo binário. Porém, o código que foi desenvolvido e implementado encontra-se preparado para que o modelo probabilístico seja utilizado bastando apenas definir a função que caracteriza o comportamento dos sensores.

2.3 Custos associados a um sensor

A cobertura de um sensor, discutida na seção anterior é, sem dúvida, um aspecto fulcral no problema de localização de sensores. O principal objetivo deste problema é monitorizar da melhor forma a área de interesse, cobrindo a maior proporção possível de área através destes meios de vigilância. Porém, é necessário ter consciência que a instalação dos sensores acarreta custos e interessa, o mais possível, obter uma solução que, para além de minimizar o risco, minimize o custo total.

Para além dos custos de aquisição que variam consoante as características de cada sensor, é de esperar que exista um custo de localização que aumenta com a distância de cada sensor a um gerador de energia. Neste trabalho consideram-se, assim, duas componentes para a função que define o custo de uma solução S :

- **Custo de aquisição:** $C_A(S)$

Consiste na soma dos custos de aquisição de cada sensor da solução, $C_A(s)$, $s \in S$. Estes custos variam consoante as propriedades específicas de cada tipo de sensor, nomeadamente o alcance de deteção que possuem. Os sensores com uma maior capacidade de deteção têm associado um custo mais elevado. Estes custos deverão ser introduzidos pelo utilizador antes da execução do algoritmo.

- **Custo de localização:** $C_L(S)$

Sem perda de generalidade, considera-se que, neste trabalho, existe um único gerador de energia localizado no ponto de coordenadas $(0, 0)$. Assim, o custo de localização de cada sensor $s \in S$ cujo centro se encontra localizado em (x_s, y_s) é definido por $C_L(s) = C_A(s) \times \sqrt{x_s^2 + y_s^2}$. O custo de localização tem em consideração a distância ao gerador e é também proporcional ao custo de aquisição de forma a refletir que sensores com determinadas características tornam mais dispendiosa a sua localização.

O custo de uma solução S é, então, definido através da expressão 2.4.

$$\begin{aligned} \text{Custo}(S) &= C_A(S) + C_L(S) \\ &= \sum_{s \in S} C_A(s) + \sum_{s \in S} C_L(s) \end{aligned} \tag{2.4}$$

Se existir um orçamento máximo disponível para a localização dos sensores, o problema pode continuar a ser visto como um problema mono-objetivo em que se pretende minimizar o risco, excluindo soluções que ultrapassem o valor de custo pretendido, sendo apresentada ao agente de decisão a solução final obtida. No entanto, este valor é muitas vezes alvo de incerteza e o decisor pode ser flexível quando confrontado com soluções que ultrapassem ligeiramente o valor de custo pretendido mas garantam uma melhor proteção da área de interesse. Assim, a introdução do custo surge no âmbito do problema biobjetivo, apresentando-se ao decisor um conjunto de soluções potencialmente não dominadas de forma a poder ponderar a que melhor se enquadra nos seus objetivos.

2.4 Formulação

2.4.1 Formulação do problema mono-objetivo

Nesta seção apresenta-se a formulação do problema mono-objetivo em que o número e tipo de sensores a instalar é conhecido *a priori*. As únicas variáveis de decisão são as coordenadas do centro de cada sensor e a orientação dos *sensores* $_{SC}$. A função objetivo corresponde à minimização do risco.

Dados:

A : Área de interesse

O : Área ocupada por obstáculos

N_C : N° de diferentes tipos de sensores circulares existentes.

N_{SC} : N° de diferentes tipos de sensores setor circular existentes.

C_i : N° de sensores circulares do tipo i a instalar, $i = 1, \dots, N_C$.

SC_j : N° de sensores setor circular do tipo j a instalar, $j = 1, \dots, N_{SC}$.

Variáveis:

(x_s, y_s) : Coordenadas do centro do sensor s , $s \in S$

o_s : Orientação do sensor setor circular s , $s \in SC$

$$S = C \cup SC$$

$$S = \left\{ 1, \dots, \sum_{i=1}^{N_C} C_i \right\} \cup \left\{ \sum_{i=1}^{N_C} C_i + 1, \dots, \sum_{i=1}^{N_C} C_i + \sum_{j=1}^{N_{SC}} SC_j \right\}$$

Minimizar $Risco(S)$ s.a: $(x_s, y_s) \in A - O, \quad \forall s \in S$ $o_s \in [0, 2\pi], \quad \forall s \in SC$

2.4.2 Formulação do problema biobjetivo

Ao contrário da decisão adotada para o problema mono-objetivo estudado nesta dissertação, no problema biobjetivo considera-se que o número e tipo de sensores a instalar não é conhecido *a priori*. Neste caso, passa a existir uma restrição adicional que limita o número de sensores a utilizar mediante a disponibilidade de cada recurso. Para além da existência de um limite superior, pode existir um interesse adicional em instalar no mínimo um determinado número de sensores com características particulares. O número de sensores a instalar deve respeitar, assim, os limites impostos consoante o respetivo tipo.

Para além da minimização do risco, a minimização do custo é também considerada, apresentando-se uma nova função objetivo que engloba estes dois critérios.

Dados:

A : Área de interesse

O : Área ocupada por obstáculos

N_C : N° de diferentes tipos de sensores circulares existentes.

N_{SC} : N° de diferentes tipos de sensores setor circular existentes.

C_i^- : N° mínimo de sensores circulares do tipo i a instalar, $i = 1, \dots, N_C$.

SC_j^- : N° mínimo de sensores setor circular tipo j a instalar, $j = 1, \dots, N_{SC}$.

C_i^* : N° de sensores circulares disponíveis do tipo i , $i = 1, \dots, N_C$.

SC_j^* : N° de sensores setor circular disponíveis do tipo j , $j = 1, \dots, N_{SC}$.

Variáveis:

C_i : N° de sensores circulares do tipo i a instalar, $i = 1, \dots, N_C$.

SC_j : N° de sensores setor circular do tipo j a instalar, $j = 1, \dots, N_{SC}$.

(x_s, y_s) : Coordenadas do centro do sensor s , $s \in S$

o_s : Orientação do sensor setor circular s , $s \in SC$

$$S = C \cup SC$$

$$S = \left\{ 1, \dots, \sum_{i=1}^{N_C} C_i \right\} \cup \left\{ \sum_{i=1}^{N_C} C_i + 1, \dots, \sum_{i=1}^{N_C} C_i + \sum_{j=1}^{N_{SC}} SC_j \right\}$$

Minimizar	$\{Risco(S), Custo(S)\}$
s.a:	$(x_s, y_s) \in A - O, \quad \forall s \in S$
	$o_s \in [0, 2\pi], \quad \forall s \in SC$
	$C_i^- \leq C_i \leq C_i^* \quad \forall i = 1, \dots, N_C$
	$SC_j^- \leq SC_j \leq SC_j^* \quad \forall j = 1, \dots, N_{SC}$
	$C_i, SC_j \in \mathbb{N}_0 \quad \forall i = 1, \dots, N_C; \forall j = 1, \dots, N_{SC}$

Os problemas anteriormente formulados e que se pretendem estudar nesta dissertação são *NP-difíceis* e, por esse motivo, aplicar algoritmos determinísticos para os resolver é computacionalmente ineficiente. A evolução tecnológica, permitindo o aumento da capacidade dos computadores, tornou as técnicas heurísticas baseadas em populações abordagens muito práticas e comuns para lidar com este tipo de problemas, obtendo-se resultados de boa qualidade num espaço de tempo razoável. Neste sentido, o capítulo que se segue apresenta a metodologia de resolução adotada para resolver o problema descrito.

Capítulo 3

Metodologia de resolução

Tendo em conta as características do problema apresentado no capítulo anterior, introduz-se, de seguida, uma metodologia de resolução baseada em técnicas heurísticas, mais precisamente, trata-se de um algoritmo genético onde a codificação das soluções é feita através de chaves aleatórias. À semelhança de qualquer técnica heurística, uma heurística baseada no conceito de população, como é o caso de um algoritmo genético, procura determinar uma solução admissível para o problema, de preferência com valor próximo do valor da solução ótima.

3.1 Algoritmos genéticos e chaves aleatórias

Os algoritmos genéticos foram introduzidos por Holland nos primeiros anos da década de 70, tendo como inspiração a teoria de Darwin. Essencialmente, um algoritmo genético é um algoritmo de pesquisa que recorre aos princípios básicos da genética e da hereditariedade e inspira-se na teoria darwinista da seleção natural na medida em que de uma geração para a seguinte sobrevivem os indivíduos mais aptos.

Num algoritmo genético, a população consiste num conjunto de soluções ou indivíduos que são representados pelos seus cromossomas. O processo de seleção e reprodução é feito a partir do momento em que os indivíduos mais aptos da geração atual são selecionados e se cruzam para dar origem à geração seguinte. O processo de seleção seleciona um certo número de soluções parentais, as quais, através do operador cruzamento, são utilizadas para gerar novas soluções (descendência). Um outro operador utilizado é o operador mutação que tem como objetivo fazer modificações aleatórias nas soluções. A avaliação de cada indivíduo (solução) é feita recorrendo à chamada função aptidão que representa o seu mérito relativo quando comparado com os restantes elementos da população em que se encontra inserido. A figura 3.1 sintetiza os principais aspetos do funcionamento de um algoritmo genético, anteriormente mencionados.

Existem, essencialmente, dois aspetos fundamentais que é necessário definir quando se aplica um algoritmo genético a um dado problema: o modo de funcionamento dos operadores adotados e a forma como as soluções são codificadas. Para um mesmo problema podem

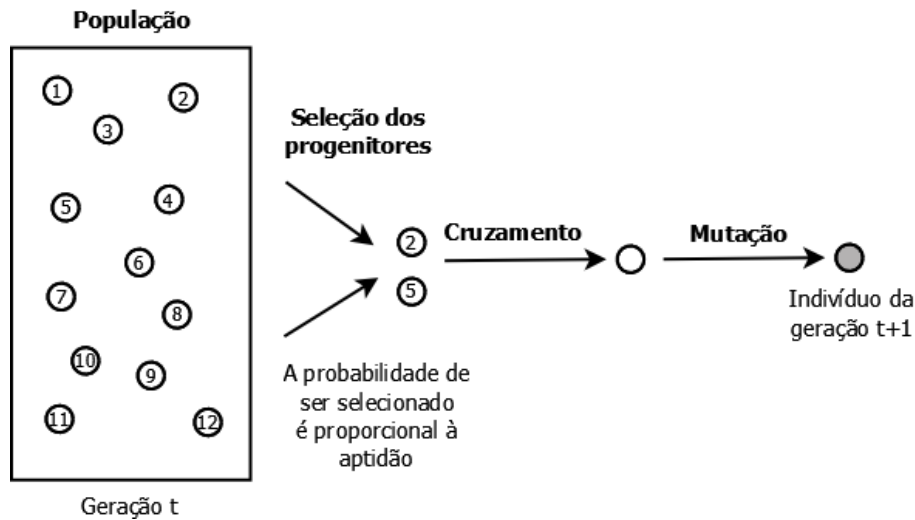


Figura 3.1: Funcionamento geral de um algoritmo genético

existir várias formas de codificar as soluções e para cada uma delas, podem existir diversos operadores genéticos válidos. A codificação de soluções através de chaves aleatórias, conforme se apresenta de seguida, é uma das formas possíveis de codificação e será utilizada para representar soluções para o problema de localização de sensores num espaço contínuo.

A abordagem dos algoritmos genéticos utilizando chaves aleatórias (do inglês *random-key genetic algorithms* - RKGA) foi introduzida por Bean para problemas de sequenciamento [4]. Esta codificação assume que os indivíduos são vetores ($1 \times N$) e que cada elemento (gene) corresponde a um valor aleatório entre 0 e 1. Um algoritmo determinístico que se irá referir, daqui em diante, como descodificador associa a cada cromossoma uma solução do problema. O espaço de soluções é, assim, analisado indiretamente, uma vez que as soluções são percorridas no espaço das chaves aleatórias recorrendo ao descodificador para avaliar a aptidão da solução correspondente à respetiva chave (figura 3.2). Note-se que o descodificador é um algoritmo determinístico, ou seja, para um dado vetor de chaves aleatórias a respetiva solução do problema é sempre a mesma. O descodificador tem um papel fundamental para o sucesso deste procedimento, como se irá referir mais à frente.

O RKGA envolve uma população inicial de p indivíduos que correspondem a p vetores de chaves aleatórias que evoluem ao longo de várias gerações. Em cada geração, as soluções são divididas em dois subconjuntos de acordo com o respetivo valor de aptidão: um grupo mais pequeno de $p \times p_e$ indivíduos constitui a elite, ou seja, os elementos da população com melhor aptidão, sendo p_e a proporção de indivíduos da elite e os $p - (p \times p_e)$ indivíduos restantes constituem o segundo grupo. De uma geração para a seguinte, as soluções de elite mantêm-se, sendo as restantes geradas através de mutação e cruzamento, conforme ilustra a figura 3.3. De acordo com o procedimento sugerido por Bean, um indivíduo gerado através de mutação consiste simplesmente num novo vetor de chaves aleatórias que é introduzido na população. Em cada geração são introduzidos $p \times p_m$ indivíduos

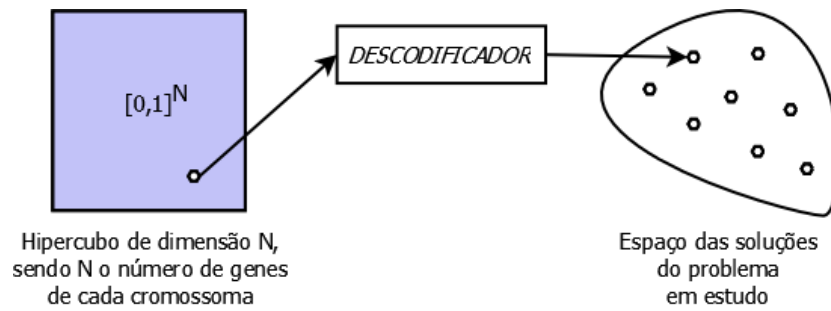


Figura 3.2: Descodificador de um algoritmo genético com chaves aleatórias. Considere-se, como exemplo, o cromossoma (0.23, 0.14, 0.72, 0.51) para um problema de sequenciamento. Este cromossoma codifica a sequência 2-1-4-3 uma vez que 0.23 corresponde ao 2º menor elemento, 0.14 ao 1º menor elemento e assim sucessivamente.

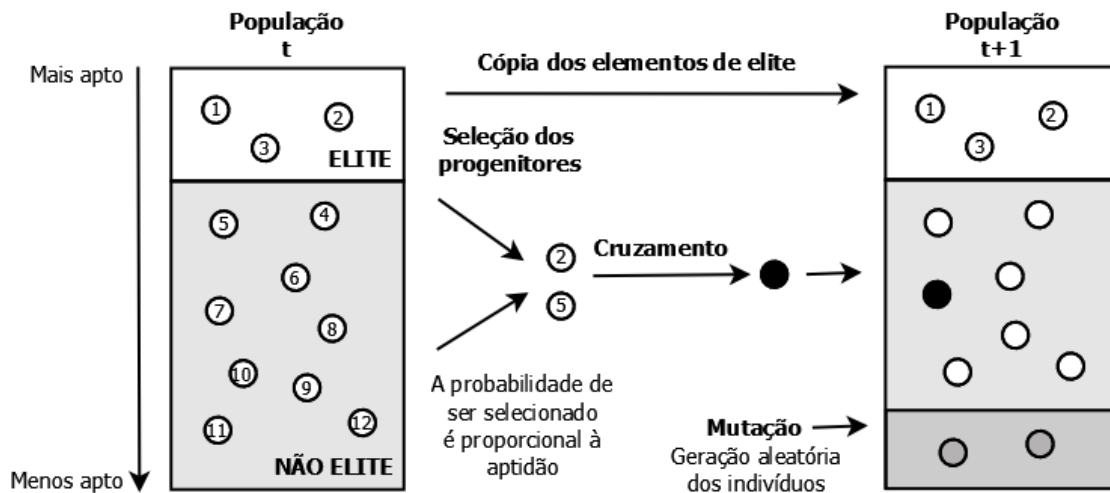


Figura 3.3: Funcionamento do RKGA

mutados na população de acordo com a proporção de soluções originadas por mutação, p_m , previamente estabelecida. De forma a manter o número p de indivíduos em cada geração, os restantes $p - (p \times p_e) - (p \times p_m)$ indivíduos são gerados por cruzamento. Bean sugere que sejam selecionados aleatoriamente dois progenitores de toda a população e que, de acordo com uma probabilidade ρ_e , se determine se cada gene do descendente provém do primeiro ou do segundo progenitor. Uma vez que se assume que qualquer vetor de chaves consegue ser decodificado dando origem a uma solução, os descendentes resultantes de cruzamento são sempre válidos, podendo ser decodificados numa solução do problema.

Salienta-se o facto de o operador mutação não ser utilizado sobre o operador cruzamento. Assim, os descendentes gerados por cruzamento não sofrem mutação como é usual nos algoritmos genéticos mas o intuito deste operador continua a ser o mesmo: obter diversidade e evitar prematuramente a estagnação em ótimos locais.

3.1.1 Biased random-key genetic algorithm

A partir do trabalho de Bean, têm surgido várias extensões e adaptações do algoritmo genético proposto para resolver uma grande variedade de problemas como é o caso do *biased random-key genetic algorithm* (BRKGA) proposto por Gonçalves e Resende [12] para a otimização de problemas combinatórios. Este algoritmo difere do algoritmo proposto por Bean, essencialmente, na forma como os progenitores são selecionados para gerar a descendência através de cruzamento. Assim, o funcionamento geral do BRKGA segue o esquema apresentado na figura 3.3: as soluções de elite mantêm-se de uma geração para a seguinte, uma determinada proporção dos elementos é gerada através de mutação e os restantes indivíduos geram-se através de cruzamento. O BRKGA introduz as novas características ao RKGA que se apresentam de seguida.

- Os progenitores continuam a ser escolhidos aleatoriamente mas um deles é escolhido do conjunto de soluções pertencentes à elite. Pode-se optar por escolher o segundo progenitor apenas no conjunto de soluções que não pertencem à elite ou em toda a população, excluindo o indivíduo já selecionado.
- Seja $\rho_e \geq 0.5$ um parâmetro definido *a priori* que define a probabilidade do descendente herdar cada gene do progenitor proveniente da elite. Assim, para cada alelo do cromossoma, existe uma probabilidade ρ_e do mesmo ser herdado do progenitor da elite e uma probabilidade $1 - \rho_e$ do mesmo ser herdado do outro progenitor. Assim, o descendente torna-se mais propício a herdar características do progenitor da elite. O procedimento de Bean não se preocupava em determinar a qual dos progenitores estava associada uma maior probabilidade de herdar características: era simplesmente o primeiro progenitor selecionado, podendo ou não ser o progenitor mais apto. Na figura 3.4 apresenta-se um exemplo para o cruzamento utilizado no BRKGA e no algoritmo 3 encontra-se sintetizado o respetivo funcionamento geral.

O algoritmo 4 apresenta os passos principais do *biased random-key genetic algorithm*.

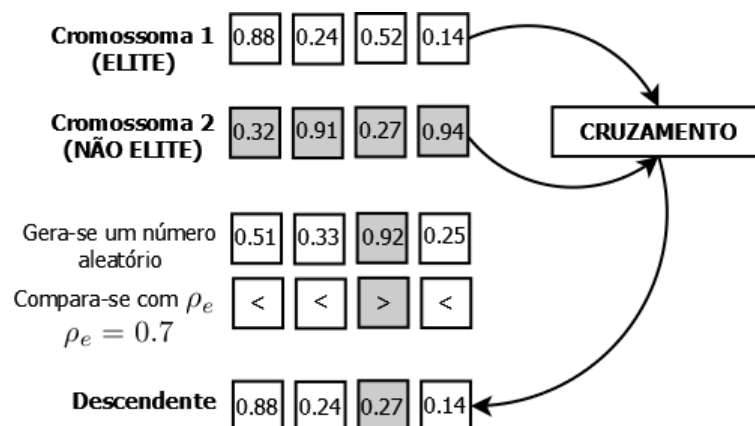


Figura 3.4: Cruzamento no BRKGA considerando $\rho_e = 0.7$.

Algoritmo 3 Cruzamento utilizado no BRKGA

Input: ρ_e, P_1 proveniente da elite, P_2 proveniente da não-elite $D = \text{cruzamento}(\rho_e, P_1, P_2)$

```

1: for cada gene  $i = 1 \dots n$  do
2:    $decisao \leftarrow$  Gerar um número aleatório entre 0 e 1
3:   if  $decisao < \rho_e$  then
4:      $D(i) = P_1(i)$ 
5:   else
6:      $D(i) = P_2(i)$ 
7:   end if
8: end for

```

Output: Descendente $D = (D(1), \dots, D(n))$

Algoritmo 4 *Biased random-key genetic algorithm*

Input: p, p_e, p_m, ρ_e, N (*) $\text{brkga}(p, p_e, p_m, \rho_e, N)$

```

1:  $t \leftarrow 0$ 
2:  $pop(t) = 0_{\{p \times N\}}$ 
3:  $aptidao = 0_{\{1 \times p\}}$ 
4: for  $i = 1 \dots p$  do
5:   Gerar o indivíduo  $pop^i(t)$  aleatoriamente
6: end for
7:  $aptidao \leftarrow \text{descodificador}(pop(t))$ 
8: Ordenar por ordem decrescente  $aptidao$ 
9: Ordenar  $pop(t)$  de acordo com a ordenação de  $aptidao$ 
10:  $t \leftarrow 1$ 
11: while não se verificar o critério de paragem do
12:   for  $j = 1 \dots p_e \times p$  do
13:      $pop^j(t) \leftarrow pop^j(t - 1)$ 
14:   end for
15:   for  $j = p_e \times p \dots p - p_e \times p - p_m \times p$  do
16:     Escolher aleatoriamente  $P_1 = pop^i(t - 1): i \in \{1 \dots p \times p_e\}$ 
17:     Escolher aleatoriamente  $P_2 = pop^i(t - 1): i \in \{p \times p_e + 1 \dots p\}$ 
18:      $pop^j(t) \leftarrow \text{cruzamento}(\rho_e, P_1, P_2)$  (algoritmo 3)
19:   end for
20:   for  $j = p - p_e \times p - p_m \times p + 1 \dots p$  do
21:     Gerar aleatoriamente  $pop^j(t)$ 
22:   end for
23:    $aptidao \leftarrow \text{descodificador}(pop(t))$ 
24:   Ordenar por ordem decrescente  $aptidao$ 
25:   Ordenar  $pop(t)$  de acordo com a ordenação de  $aptidao$ 
26:    $t \leftarrow t + 1$ 
27: end while

```

Output: (*) Como será explicado nas seções seguintes, o *output* varia consoante o problema é mono-objetivo ou biobjetivo. Para o primeiro caso, o algoritmo devolve o indivíduo mais apto da última geração, enquanto que, no segundo caso, obtém-se um conjunto de soluções potencialmente não dominadas e cabe ao decisor escolher a solução final.

Uma das características mais interessantes do *biased random-key genetic algorithm* é a nítida divisão existente entre a estrutura geral do algoritmo e o decodificador das soluções. O funcionamento geral do algoritmo não necessita de nenhum conhecimento *a priori* do problema em estudo, limitando-se à procura de soluções no hipercubo conforme já ilustrado na figura 3.2. A única ligação com o problema é feita através do decodificador que, a partir dos vetores de chaves aleatórias, determina a correspondente solução para o problema em causa e o valor da respetiva aptidão. Assim, para cada problema em particular, apenas é necessário definir a representação dos cromossomas e o funcionamento do decodificador.

Existem vários problemas em que a aplicação do algoritmo anteriormente descrito produz resultados competitivos em comparação com a aplicação de outras técnicas heurísticas, como é o caso do problema de alocação de unidades geradoras descrito por Roque *et al.* [23]. Outras aplicações deste algoritmo são apresentadas no trabalho de Gonçalves *et al.* [12], nomeadamente o problema de empacotamento de retângulos [13], o problema de minimizar o congestionamento de tráfego nas estradas, otimizando a localização de portagens [5] e o problema de sequenciamento de atividades de um projeto [18].

Nas seções que se seguem apresentam-se as considerações necessárias para aplicar esta metodologia ao problema de localização de sensores, nomeadamente a codificação das soluções e o algoritmo *descodificador* necessário para obter a respetiva aptidão para os problemas mono-objetivo e biobjetivo. Enquanto que para o problema mono-objetivo o cálculo da aptidão é obtido diretamente através da determinação da cobertura conforme explicado na seção 2.2, para o problema biobjetivo serão feitas algumas referências adicionais sobre a utilização de algoritmos genéticos em problemas multiobjetivo.

3.2 Problema mono-objetivo

Nesta seção discute-se a aplicação do BRKGA ao problema de localização de sensores numa área de interesse contínua em que, dado um número pré-definido de sensores de diferentes tipos, se pretende determinar a melhor localização para os recursos disponíveis de forma a minimizar o risco, ou equivalentemente a proporção total de área por cobrir. Uma vez que estamos perante um problema com um único critério a otimizar, este problema designa-se por problema mono-objetivo.

Conforme foi mencionado aquando da descrição do problema, existem sensores cuja área de cobertura é representada por um círculo (sensores circulares) e sensores que cobrem uma área correspondente a um setor circular (sensores setor circular). Para cada um destes sensores existem vários tipos disponíveis que variam quanto ao raio (cada sensor s tem um raio r_s) e, no caso dos sensores setor circular, variam também relativamente ao ângulo de visão (a_s).

Antes de se descrever a forma como cada solução, representada através de um vetor de chaves aleatórias, é decodificada tendo em conta o problema em estudo, apresenta-se a notação que será utilizada.

Notação utilizada no problema mono-objetivo:

N_C : N° de diferentes tipos de sensores circulares existentes.

N_{SC} : N° de diferentes tipos de sensores setor circular existentes.

C_i : N° de sensores circulares do tipo i a instalar, $i = 1, \dots, N_C$.

SC_j : N° de sensores setor circular do tipo j a instalar, $j = 1, \dots, N_{SC}$.

$N'_C = \{C_1 + \dots + C_{N_C}\}$: N° total de sensores circulares a instalar.

$N'_{SC} = \{SC_1 + \dots + SC_{N_{SC}}\}$: N° total de sensores setor circular a instalar.

$N' = N'_C + N'_{SC}$: N° total de sensores a instalar.

Sendo $k[n]$ a n -ésima chave aleatória do vetor que representa cada solução, a representação de cada indivíduo é feita através de um vetor de chaves aleatórias, K , conforme se apresenta de seguida.

$$K = \underbrace{(k[1], \dots, k[N'])}_{\text{Sequenciamento}} \mid \underbrace{(k[N'+1], \dots, k[2N'])}_{\text{Coordenadas } x \text{ do centro}} \mid \underbrace{(k[2N'+1], \dots, k[3N'])}_{\text{Coordenadas } y \text{ do centro}} \mid \underbrace{(k[3N'+1], \dots, k[3N'+N'_{SC}])}_{\text{Orientação}} \\ \text{(só para sensores setor circular)}$$

Para proceder à localização dos sensores, é utilizada uma heurística muito simples para decodificar as soluções e que se sintetiza nos passos que se apresentam de seguida.

1. Determinar a ordem de instalação dos N' sensores através da ordenação das primeiras N' chaves aleatórias. Cada chave aleatória $k[1], \dots, k[N']$ diz respeito a um tipo de sensor pela ordem em que os tipos de sensores são definidos no problema. Este aspeto será melhor explicado no exemplo que se apresenta no final desta seção.
2. Pela ordem definida no passo anterior localizar os sensores na área de interesse nas coordenadas do respetivo centro. No caso em que a área de interesse corresponde ao quadrado unitário, as coordenadas do centro são diretamente determinadas através dos valores das chaves aleatórias. Os casos de estudo que incluem restrições na localização de sensores requerem mecanismos especiais para garantir a admissibilidade das soluções como se irá referir na seção 4.2 do capítulo 4.
3. Se o sensor s a localizar for um sensor setor circular é, ainda, necessário ter em consideração a orientação do mesmo, o_s . Como foi referido no capítulo 2 quando foram descritos os vários tipos de sensores, a orientação não é uma característica pré-definida destes sensores mas sim uma propriedade que se altera no decorrer do algoritmo. A cada chave aleatória $k[3N'+1], \dots, k[3N'+N'_{SC}]$ relativa à orientação dos sensores, corresponde uma amplitude entre 0 e 2π (em radianos) que é obtida multiplicando a respetiva chave por 2π conforme refere Silva *et al.* [27] para aplicações do BRKGA a problemas de otimização global contínua com restrições de domínio.
4. Determinar a aptidão da solução que corresponde à proporção total de área marítima por cobrir, considerando que os sensores se encontram localizados de acordo com os passos 1-3.

A determinação da cobertura de uma solução S , $Cob(S)$, é feita tendo em consideração os aspetos mencionados na seção 2.2. O valor da aptidão de uma solução é, então, determinada através de $1 - Cob(S)$. Note-se que como o objetivo do problema é minimizar o risco, os indivíduos mais aptos são aqueles aos quais está associado um menor valor de $1 - Cob(S)$ (proporção de área por cobrir).

Aplicando esta heurística a todas as p soluções de uma população $pop(t)$ da geração t define-se o *descodificador* referido no algoritmo 4. Este procedimento encontra-se sintetizado no algoritmo 5. Assim, estão encontradas todas as componentes necessárias à aplicação do BRKGA para o problema mono-objetivo de localização de sensores. A solução para o problema corresponde ao indivíduo mais apto da geração atual, com uma proporção total de área por cobrir correspondente ao valor da sua aptidão.

Algoritmo 5 Descodificador para o problema mono-objetivo

Input: $pop(t)$, p , $raiosSensores$, $amplitudeSC$

$aptidao = descodificador(pop(t))$

```

1:  $sequenciamento = 0_{\{1 \times N'\}}$ ,  $ordem = 0_{\{1 \times N'\}}$ ,  $aptidao = 0_{\{1 \times p\}}$ 
2: for  $k = 1 \dots p$  do
3:    $K \leftarrow pop^k(t)$ 
4:    $S \leftarrow \{\}$ 
5:   for  $i = 1 \dots N'$  do
6:      $sequenciamento(i) \leftarrow i$ -ésimo menor elemento de  $K$ 
7:      $ordem(i) \leftarrow$  posição de  $sequenciamento(i)$  no vetor  $K$ 
8:   end for
9:   for  $s = 1 \dots N'$  do
10:     $centros_s = (k[N' + ordem(s)], k[2N' + ordem(s)])$ 
11:     $r_s = raiosSensores(ordem(s))$ 
12:    if  $ordem(s)$  é  $sensor_{SC}$  then
13:       $o_s = 2\pi \times k[3N' + ordem(s) - N'_C]$ 
14:    end if
15:     $S \leftarrow S \cup \{localização\ do\ sensor\ s\ em\ centros_s\ com\ o_s\ (para\ os\ sensores_{SC})\}$ 
16:  end for
17:   $aptidao(k) = 1 - Cob(S)$ 
18: end for

```

Output: $aptidao$

Para uma melhor compreensão do funcionamento do descodificador anteriormente apresentado, considere-se o seguinte exemplo em que existem três tipos de sensores:

- **Tipo I**

Sensor circular de raio $r_I = \frac{\sqrt{2}}{4}$.

Pretendem-se instalar 2 unidades.

- **Tipo II**

Sensor circular de raio $r_{II} = \frac{\sqrt{2}}{6}$.

Pretende-se instalar 1 unidade.

- **Tipo III**

Sensor setor circular de raio $r_{III} = \frac{\sqrt{2}}{6}$ e amplitude $a_{III} = \frac{\pi}{2}$.

Pretende-se instalar 1 unidade.

Tem-se $N' = N'_C + N'_{SC} = 3 + 1 = 4$ e, portanto, uma solução admissível para este problema é dada pelo vetor K^1 de chaves aleatórias que se apresenta de seguida.

$$K^1 = (\underbrace{0.03, 0.65}_{\text{Tipo I}}, \underbrace{0.07}_{\text{Tipo II}}, \underbrace{0.54}_{\text{Tipo III}} | 0.09, 0.51, 0.54, 0.93 | 0.56, 0.72, 0.17, 0.14 | 0.25)$$

A descodificação da solução K^1 encontra-se apresentada na tabela 3.1.

Ordem	Sequenciamento	Sensor	Coordenada do centro	Orientação
1º	0.03	I	(0.09,0.56)	-
2º	0.07	II	(0.54,0.17)	-
3º	0.54	III	(0.93,0.14)	$0.25 \times 2\pi = \frac{\pi}{2} \text{ rad}$
4º	0.65	I	(0.51,0.72)	-

Tabela 3.1: Descodificação para a solução K^1 .

Considerando o modelo de cobertura binária e uma grelha de pontos de precisão $\delta = 0.1$ para a determinação da aptidão da solução correspondente ao vetor K^1 , obtém-se um valor de 0.36 ($1 - \frac{77}{121} \simeq 1 - 0.64 = 0.36$) para a proporção de área marítima por cobrir.

A figura 3.5 apresenta a solução obtida com a grelha de pontos ajustada para o cálculo da respetiva aptidão.

Note-se que, neste exemplo, se considera o caso mais simples em que a área de interesse corresponde apenas à região marítima. No capítulo 4 serão referidos os mecanismos necessários para adaptar a metodologia apresentada nesta seção para áreas de interesse que incluem obstáculos.

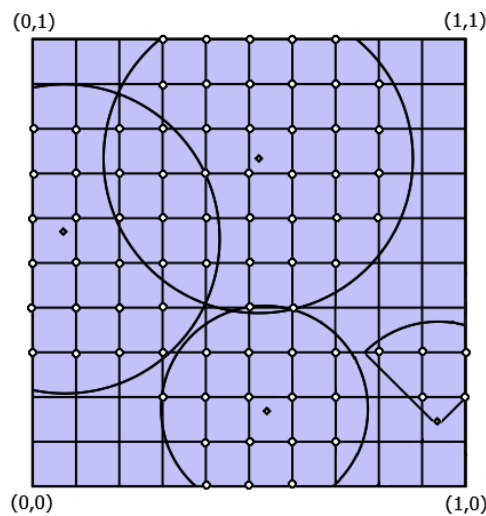


Figura 3.5: Solução S associada ao vetor K^1 de chaves aleatórias.

3.3 Problema biobjetivo

O problema biobjetivo para a localização de sensores que se propõe estudar na presente dissertação, considera que existem dois critérios a otimizar: se por um lado, continua a ser essencial a minimização do risco, por outro lado, não deixa de ser igualmente importante a minimização do custo.

A metodologia apresentada para o problema mono-objetivo considera que o número de sensores a instalar é conhecido *a priori*. Porém, na prática esta situação não é muito realista e torna-se mais interessante supor que o número de sensores pode variar dentro de um número limitado de recursos disponíveis, podendo existir, também, um limite inferior para o número de sensores a localizar. Assim, o problema biobjetivo analisa duas questões distintas: *quantos* sensores utilizar e *onde* fazer a respetiva localização na área de interesse em estudo. Apesar da complexidade acrescida do problema biobjetivo, este problema é muito mais completo que o problema mono-objetivo anteriormente descrito e, por esse motivo, torna-se muito mais útil e próximo da realidade. A abordagem que se apresenta pretende ser inovadora na medida em que as duas decisões a tomar são consideradas em simultâneo, como se irá explicar.

Em termos estruturais, a abordagem adotada para o problema biobjetivo difere da abordagem considerada para o problema mono-objetivo, essencialmente, em termos da codificação das soluções e da determinação da respetiva aptidão. Apesar do decodificador funcionar de modo análogo é importante ter em conta que, neste caso existem dois objetivos em estudo e, por essa razão, tem de existir um critério para ordenar as soluções de forma a continuar a existir uma divisão entre os indivíduos de elite e os restantes. Neste caso, não existe uma única solução final para o problema mas sim um conjunto de soluções potencialmente não dominadas que definem a fronteira potencialmente de Pareto. Estes conceitos serão explicados em 3.3.1.

Antes de se proceder à explicação do modo como as soluções são codificadas, evidenciando as duas decisões a tomar relativas à quantidade de sensores a instalar e à respetiva localização, apresentam-se algumas considerações sobre a utilização de algoritmos genéticos em problemas onde existe mais que um objetivo para otimizar. Serão descritos alguns conceitos essenciais e será apresentado o procedimento adotado, neste trabalho, para avaliar as soluções em termos da respetiva aptidão.

3.3.1 Avaliação das soluções em problemas multiobjetivo

A maioria dos problemas reais com que nos deparamos requer a otimização simultânea de múltiplos objetivos que, na maior parte dos casos, são conflituosos entre si. Para o problema biobjetivo de localização de sensores, a minimização do risco entra, também, em conflito com a minimização do custo uma vez que é de esperar que ao introduzir um maior número de sensores na solução o risco diminua mas haja um aumento no custo. Para obter a solução para problemas com mais que um objetivo recorre-se, por vezes, a uma

função de utilidade que permite combinar os critérios num só, tornando o problema mono-objetivo. Porém, nem sempre é conhecida *a priori* uma função de utilidade adequada ao processo e, nesse caso, o problema deve ser tratado como um *problema multiobjetivo* (PMO). Problemas em que se pretende otimizar dois critérios designam-se por *problemas biobjetivo*.

Nos problemas multiobjetivo, um conjunto de soluções é providenciado ao decisor para que este possa ir tomando decisões de forma a escolher a solução final do problema em estudo. Apesar do agente de decisão poder ter vários tipos de intervenção, irá considerar-se que, neste problema, o mesmo só intervém ativamente *a posteriori*. As soluções apresentadas ao decisor depois de se ter aplicado a metodologia de resolução designam-se por *soluções não dominadas* e fazem parte da *fronteira de Pareto*. Esta situação difere de um problema mono-objetivo em que se apresenta uma única solução final com melhor valor para o objetivo a otimizar.

Assumindo-se, sem perda de generalidade, um problema multiobjetivo em que se pretende minimizar cada um dos critérios, apresenta-se, de seguida, a definição formal de uma *solução não dominada*.

Considere-se a seguinte formulação geral para um problema multiobjetivo.

$$\begin{aligned} \text{(PMO)} \quad & \text{Min } f(S) = \{f_1(S), \dots, f_i(S), \dots, f_m(S)\} \\ & \text{s.a: } S \in D \\ & D = \{S : g_j(S) \leq 0, j = 1, \dots, J, h_k(S) = 0, k = 1, \dots, K\} \end{aligned}$$

Uma solução S^* , no espaço dos objetivos, diz-se *não dominada* se e só se não existe uma outra solução $S \in D$ tal que $f_i(S) \leq f_i(S^*) \forall i = 1, \dots, m$ e $f_i(S) < f_i(S^*)$ para pelo menos um i .

Uma vez que para resolver o problema em estudo se recorre a uma técnica heurística não existe a garantia de que todas as soluções do problema foram analisadas. Neste caso, os conceitos anteriormente definidos dizem apenas respeito às soluções conhecidas até ao momento e, portanto, ao aplicar o *biased random-key genetic algorithm* ao problema de localização de sensores obtém-se uma fronteira potencialmente de Pareto constituída por soluções potencialmente não dominadas.

Para utilizar o *biased random-key genetic algorithm* no problema biobjetivo tem de existir um critério para caracterizar as soluções de modo a que as melhores constituam a elite. Uma das opções que poderia ter sido adotada seria fazer corresponder as soluções potencialmente não dominadas à elite e as restantes ao grupo não elite. Esta abordagem apresenta, porém, algumas desvantagens. Em primeiro lugar, o número de soluções potencialmente não dominadas poderia ultrapassar o número estabelecido para a proporção de indivíduos da elite. Uma alternativa seria ajustar, em cada geração, o valor de p_e de modo a que o número de soluções pertencentes à elite correspondesse ao número de soluções

potencialmente não dominadas nessa geração. Porém, caso o número de soluções potencialmente não dominadas fosse superior à dimensão pré-estabelecida para a população, o número total de indivíduos seria variável de geração para geração. Um outro problema que surgiria, nesse caso, seria a determinação do número de indivíduos resultantes de cruzamento e mutação. Uma possibilidade para contornar os problemas anteriormente mencionados seria guardar, paralelamente à execução do algoritmo, as soluções potencialmente não dominadas que se vão obtendo, atualizando, em cada geração, as soluções encontradas até ao momento que continuam potencialmente não dominadas. Apesar de não se perderem soluções potencialmente não dominadas, a evolução do algoritmo não teria em conta os princípios mencionados na seção 3.1.1 e evoluiria de modo praticamente aleatório. Se o número de soluções potencialmente não dominadas fosse inferior ao número de elementos que constituem a elite, algumas soluções da elite seriam dominadas e era-lhes atribuída a mesma importância que às soluções potencialmente não dominadas, o que não parece ser um critério razoável. Tendo em conta as considerações anteriormente mencionadas, sugere-se uma forma alternativa de ordenar as soluções.

Na literatura encontram-se vários trabalhos de otimização para problemas multiobjetivo através de algoritmos genéticos. Coello Coello *et al.* [9] apresentam uma síntese de diferentes abordagens possíveis para adaptar algoritmos evolutivos a problemas multiobjetivo. Mantendo uma população de soluções, os algoritmos genéticos permitem a procura simultânea de várias soluções não dominadas, sendo esta uma das características mais atrativas desta técnica heurística para os problemas com mais que um objetivo. Fonseca e Fleming [10] modificaram os algoritmos genéticos de forma a lidar com múltiplos objetivos de modo a que o problema em estudo não tenha de ser convertido numa série de problemas de otimização para cada um dos critérios em análise. Estas modificações tornaram estes algoritmos conhecidos como algoritmos genéticos multiobjetivo. Mais tarde, Narayanan e Azarm [20] propuseram alguns melhoramentos ao trabalho de Fonseca e Fleming. Nesta dissertação, à semelhança do que acontece com as abordagens que recorrem explicitamente à definição de fronteira de Pareto, considera-se que cada indivíduo é avaliado tendo em conta o respetivo *rank*. Adaptando a definição adotada por Fonseca e Fleming considera-se que o *rank* de uma solução corresponde ao número de soluções que, na população corrente, dominam essa solução. Às soluções não dominadas atribui-se, assim, *rank* nulo. O *rank* de uma solução, sendo um critério de comparação de diferentes soluções do problema, corresponde, neste trabalho, à respetiva aptidão.

Com o objetivo de reduzir o tempo computacional e de diminuir o número de soluções potencialmente não dominadas apresentadas ao agente de decisão, algumas destas soluções são eliminadas em cada geração. É importante que o decisor tenha um leque variado de opções para poder escolher a solução que melhor se enquadra com as suas pretensões mas, este número de soluções, não deverá ser demasiado elevado pois irá dificultar a decisão. É especialmente irrelevante ter muitas soluções numa mesma zona do espaço dos objetivos. Assim, o critério utilizado nesta dissertação para diminuir o número de soluções

potencialmente não dominadas é escolher, no máximo, uma única solução representativa de cada zona do espaço dos objetivos. Para tal, é utilizada uma grelha com uma determinada precisão δ^* , ajustada aos valores de risco e custo obtidos em cada geração. Em cada quadrícula da grelha em que exista mais do que uma solução potencialmente não dominada, é escolhida a solução mais equilibrada em ambos os objetivos, ou seja, aquela para a qual a soma ponderada dos valores dos objetivos, considerando que ambos têm igual importância, é melhor. De forma a penalizar as restantes soluções, é-lhes atribuído um *rank* M elevado. Apenas as soluções que não foram penalizadas continuam com um *rank* nulo atribuído, procurando-se de geração em geração soluções potencialmente não dominadas bem espalhadas ao longo do espaço dos objetivos, sem existirem zonas grandemente povoadas nas quais o decisor teria dificuldade em analisar as diferenças entre as soluções.

O método anteriormente apresentado, ao reduzir o número de soluções potencialmente não dominadas que são apresentadas ao agente de decisão, evita que sejam geradas mais soluções potencialmente não dominadas que o número de indivíduos de elite. Assim, estas soluções conservam-se naturalmente de geração em geração, sem ser necessário guardá-las em paralelo à execução do algoritmo. Note-se que o valor escolhido para a precisão da grelha tem de ser um valor equilibrado de forma a serem apresentadas várias soluções alternativas ao decisor mas evitando que este número seja demasiado elevado pois irá aumentar o tempo computacional e dificultar a tomada de uma decisão. Este aspeto será referido na seção 5.2.1.

O procedimento descrito para atribuir o *rank* a cada solução apresenta-se sintetizado no algoritmo 6, salientando-se, no algoritmo 7, o processo de redução do número de soluções potencialmente não dominadas. Note-se que a determinação do *rank* de uma solução assume que é previamente conhecido o valor do custo e da proporção de área por cobrir de todas as soluções da população de dimensão p , respetivamente guardados no vetor $custo_{\{1 \times p\}}$ e $risco_{\{1 \times p\}}$. Por este motivo, o algoritmo 6 determina simultaneamente o *rank* de todas as soluções da população.

Algoritmo 6 Determinar o *rank* das soluções de uma população

Input: p , $custo$, $risco$

$rank = determinarRank(p, custo, risco)$

1: $rank = 0_{\{1 \times p\}}$

2: **for** $i = 1 \dots p$ **do**

3: **for** $j = 1 \dots p$ **do**

4: **if** $(risco(j) \leq risco(i) \text{ and } custo(j) < custo(i))$ **or**

5: $(risco(j) < risco(i) \text{ and } custo(j) \leq custo(i))$ **then**

6: $rank(i) = rank(i) + 1$

7: **end if**

8: **end for**

9: **end for**

10: $rank = eliminarSolucoes(rank, risco, custo)$ (algoritmo 7)

Output: $rank$

No algoritmo 7, de forma a selecionar as soluções com o método da grelha, é utilizada uma matriz $Q_{\{\frac{1}{\delta^*}, \frac{1}{\delta^*}\}}$ que guarda em cada posição o número de soluções que pertencem a cada quadrícula da grelha e a matriz $Q^*_{\{\frac{1}{\delta^*}, \frac{1}{\delta^*}\}}$ que guarda as respetivas soluções.

Algoritmo 7 Reduzir o número de soluções potencialmente não dominadas

Input: $p, rank, risco, custo, maxRisco, maxCusto, \delta^*, M$

$rankAtualizado = eliminarSolucoes(rank, risco, custo)$

```

1:  $rankAtualizado \leftarrow rank$ 
2:  $solucaoEscolhida \leftarrow \{\}$ 
3:  $Q \leftarrow \{\}, Q^* \leftarrow \{\}$ 
4: for  $i = 1 \dots p$  do
5:   if  $rank(i) = 0$  then
6:      $Q(\lfloor \frac{risco(i)}{maxRisco \times \delta^*} \rfloor + 1, \lfloor \frac{custo(i)}{maxCusto \times \delta^*} \rfloor + 1) \leftarrow Q(\lfloor \frac{risco(i)}{maxRisco \times \delta^*} \rfloor + 1, \lfloor \frac{custo(i)}{maxCusto \times \delta^*} \rfloor + 1) + 1$ 
7:      $Q^*(\lfloor \frac{risco(i)}{maxRisco \times \delta^*} \rfloor + 1, \lfloor \frac{custo(i)}{maxCusto \times \delta^*} \rfloor + 1) \leftarrow Q^*(\lfloor \frac{risco(i)}{maxRisco \times \delta^*} \rfloor + 1, \lfloor \frac{custo(i)}{maxCusto \times \delta^*} \rfloor + 1) \cup \{i\}$ 
8:   end if
9: end for
10: for  $i = 1 \dots \frac{1}{\delta^*}$  do
11:   for  $j = 1 \dots \frac{1}{\delta^*}$  do
12:     if  $Q(i, j) > 1$  then
13:        $solucaoEscolhida(i, j) = Min_{k \in Q^*(i, j)}(0.5 \times risco(k) + 0.5 \times custo(k))$ 
14:     end if
15:     for  $k \in Q^*(i, j)$  do
16:       if  $k \neq solucaoEscolhida(i, j)$  then
17:          $rankAtualizado(k) = M$ 
18:       end if
19:     end for
20:   end for
21: end for

```

Output: $rankAtualizado$

Depois de apresentada a forma como as soluções são avaliadas e feitas as considerações necessárias para o cálculo da respetiva aptidão, segue-se a explicação da codificação adotada para as soluções no problema biobjetivo.

3.3.2 Codificação das soluções

A codificação apresentada para o problema mono-objetivo considera que o número de sensores é conhecido *a priori*. O problema biobjetivo poderia utilizar essa mesma codificação uma vez que o custo total de cada solução não depende só do número e tipo de sensores instalados, dependendo também da distância de cada sensor ao gerador de energia. Assim, mesmo conhecendo a configuração adotada teria interesse otimizar não só o risco mas também o custo. Porém, com o objetivo de aproximar o problema da realidade, optou-se por introduzir no problema biobjetivo uma codificação das soluções que não necessite de conhecer, à partida, o número e tipo de sensores a instalar. É de esperar que não haja um interesse exclusivo na instalação de um número pré-definido de sensores, mas sim na obtenção de uma solução que cumpra os requisitos pretendidos minimizando ambos os objetivos, independentemente da configuração que se considera.

A notação que será utilizada para a codificação das soluções no caso do problema biobjetivo apresenta-se de seguida.

Notação utilizada no problema biobjetivo:

N_C : N° de diferentes tipos de sensores circulares existentes.

N_{SC} : N° de diferentes tipos de sensores setor circular existentes.

$N^{total} = N_C + N_{SC}$: N° total de diferentes tipos de sensores existentes.

C_i : N° de sensores circulares do tipo i a instalar, $i = 1, \dots, N_C$.

SC_j : N° de sensores setor circular do tipo j a instalar, $j = 1, \dots, N_{SC}$.

C_i^- : N° mínimo de sensores circulares do tipo i a instalar, $i = 1, \dots, N_C$.

SC_j^- : N° mínimo de sensores setor circular tipo j a instalar, $j = 1, \dots, N_{SC}$.

C_i^* : N° de sensores circulares disponíveis do tipo i , $i = 1, \dots, N_C$.

SC_j^* : N° de sensores setor circular disponíveis do tipo j , $j = 1, \dots, N_{SC}$.

$N_C^* = \{C_1^* + \dots + C_{N_C}^*\}$: N° total de sensores circulares disponíveis.

$N_{SC}^* = \{SC_1^* + \dots + SC_{N_{SC}}^*\}$: N° total de sensores setor circular disponíveis.

$N'^* = N_C^* + N_{SC}^*$: N° total de sensores disponíveis.

Tendo em conta a notação apresentada e sendo, novamente, $k[n]$ a n -ésima chave aleatória do vetor que representa cada solução, a representação de cada indivíduo é feita através de um vetor de chaves aleatórias, K , como a seguir se apresenta.

$$K = \underbrace{(k[1], \dots, k[N'^*])}_{\text{Sequenciamento}} \mid \underbrace{(k[N'^* + 1], \dots, k[2N'^*])}_{\text{Coordenadas } x \text{ do centro}} \mid \underbrace{(k[2N'^* + 1], \dots, k[3N'^*])}_{\text{Coordenadas } y \text{ do centro}} \mid \underbrace{(k[3N'^* + 1], \dots, k[3N'^* + N_{SC}^*])}_{\text{Orientação (só para sensores setor circular)}} \mid \underbrace{(k[3N'^* + N_{SC}^* + 1], \dots, k[3N'^* + N_{SC}^* + N^{total}])}_{\text{Sensores a instalar}}$$

São essencialmente duas as principais diferenças entre a codificação anterior e a codificação em que se conhece *a priori* o número de sensores a instalar:

- Quando o número de sensores de cada tipo a instalar é conhecido, o conjunto de chaves aleatórias relativas ao sequenciamento, às coordenadas x do centro e às coordenadas y do centro corresponde ao número total de sensores que se pretende instalar. A orientação dos sensores setor circular é, igualmente, definida para o número total deste tipo de sensores que se pretende localizar. Por outro lado, quando não existem configurações pré-definidas, em vez de se considerar o número de sensores a instalar, é considerado o número total de sensores disponíveis, N'^* . Este valor é determinado a partir do número máximo de unidades disponíveis para cada tipo de sensor.
- A codificação apresentada inclui um conjunto de chaves aleatórias para definir o número de sensores a instalar. Consideram-se N^{total} chaves aleatórias relativas à determinação de *quantos* sensores instalar, uma para cada diferente tipo de sensor.

O procedimento adotado para localizar os sensores sofre algumas alterações relativamente à heurística utilizada para o problema mono-objetivo. Os passos principais que são, agora, necessários para descodificar um vetor de chaves aleatórias para o problema biobjetivo que se considera nesta dissertação são apresentados de seguida. Os passos 3 e 4 correspondem, respetivamente, aos passos 2 e 3 da codificação referida na seção anterior.

1. Determinar o número de sensores a instalar, recorrendo aos valores das chaves aleatórias para determinar um número aleatório inteiro entre os limites considerados. Para tal, utilizam-se as expressões abaixo indicadas e, de seguida, o valor obtido é arredondado para o inteiro mais próximo.

$$C_i = C_i^- + k[3N'^* + N_{SC}^* + i] \times (C_i^* - C_i^-), \quad i = 1, \dots, N_C$$

$$SC_j = SC_j^- + k[3N'^* + N_{SC}^* + N_C + j] \times (SC_j^* - SC_j^-), \quad j = 1, \dots, N_{SC}$$

2. Analogamente ao que acontecia na codificação em que o número de sensores a instalar era conhecido *a priori*, as chaves aleatórias correspondentes ao sequenciamento encontram-se apresentadas em K pela ordem em que os sensores são definidos no problema. Assim, cada conjunto de chaves aleatórias correspondente a um tipo de sensor é ordenado de forma crescente e são selecionadas as primeiras chaves aleatórias correspondentes ao número de sensores a instalar do respetivo tipo conforme determinado no passo anterior.
3. Depois de definidos os sensores a instalar de cada tipo, o processo de localização é análogo ao processo descrito na seção anterior: pela ordem definida no passo 2 os sensores são localizados na área de interesse nas coordenadas do respetivo centro. Caso a área de interesse inclua obstáculos, as considerações da seção 4.2 do capítulo 4 são igualmente válidas para o problema biobjetivo.

A introdução de restrições no problema biobjetivo não constitui um problema adicional uma vez que, como será explicado no próximo capítulo e tal como sugere Kurpati *et al.* [16], garantir em primeiro lugar a verificação das restrições e só posteriormente avaliar as soluções em termos dos objetivos em estudo constitui uma abordagem eficaz para incluir restrições nos problemas multiobjetivo.

4. Se o sensor s a localizar for um sensor setor circular é, ainda, necessário ter em consideração a orientação do mesmo, o_s , que se obtém multiplicando a respetiva chave aleatória do conjunto de chaves que definem a orientação por 2π .

Quando os passos 1-4 forem realizados para todos os indivíduos da população numa determinada geração, procede-se ao cálculo da respetiva aptidão através do procedimento descrito no algoritmo 6.

Os passos anteriormente descritos sintetizam-se no algoritmo 8 que representa o *descodificador* para o problema biobjetivo.

Algoritmo 8 Descodificador para o problema biobjetivo

Input: $pop(t)$, p , $raiosSensores$, $amplitudeSC$, N^{total}
 $aptidao = descodificador(pop(t))$

```

1:  $aptidao = 0_{\{1 \times p\}}$ 
2:  $risco \leftarrow \{\}$ 
3:  $custo \leftarrow \{\}$ 
4:  $maxTipo \leftarrow [C^* \ SC^*]$ ,  $minTipo \leftarrow [C^- \ SC^-]$ 
5: for  $k = 1 \dots p$  do
6:    $K \leftarrow pop^k(t)$ 
7:    $S^k \leftarrow \{\}$ 
8:   for  $j = 1 \dots N^{total}$  do
9:      $sequenciamento \leftarrow \{\}$ 
10:     $ordem \leftarrow \{\}$ 
11:     $nInstalar \leftarrow minTipo(j) + k[3N'^* + N'_{SC} + j] \times (maxTipo(j) - minTipo(j))$ 
12:     $K^{aux} \leftarrow \{\}$ 
13:     $soma \leftarrow \sum_{m=1}^{j-1} maxTipo(m)$ 
14:     $indice \leftarrow 1$ 
15:    for  $l = soma + 1 \dots soma + maxTipo(j)$  do
16:       $K^{aux}(indice) = K(l)$ 
17:       $indice = indice + 1$ 
18:    end for
19:    for  $i = 1 \dots maxTipo(j)$  do
20:       $sequenciamento(i) \leftarrow i$ -ésimo menor elemento de  $K^{aux}$ 
21:       $ordem(i) \leftarrow$  posição de  $sequenciamento(i)$  no vetor  $K^{aux}$ 
22:    end for
23:    for  $s = 1 \dots nInstalar$  do
24:       $centro_s = (k[N'^* + soma + ordem(s)], k[2N'^* + soma + ordem(s)])$ 
25:       $r_s = raioSensores(soma + ordem(s))$ 
26:      if  $ordem(s)$  é  $sensor_{SC}$  then
27:         $o_s = 2\pi \times k[3N'^* + soma - N'_C + ordem(s)]$ 
28:      end if
29:       $S^k \leftarrow S^k \cup \{localização \text{ do sensor } s \text{ em } centro_s \text{ com } o_s \text{ (para os } sensores_{SC})\}$ 
30:    end for
31:  end for
32:   $risco(k) = 1 - Cob(S^k)$ 
33:   $custo(k) = calcularCusto(S^k)$ 
34: end for
35:  $aptidao = determinarRank(p, custo, risco)$  (algoritmo 6)

```

Output: $aptidao$

Segue-se um exemplo que pretende ilustrar o funcionamento do descodificador anteriormente descrito. À semelhança do exemplo apresentado para o problema mono-objetivo, considera-se a existência de três tipos diferentes de sensores. Neste caso, é conhecido o limite inferior e superior para o número de sensores a instalar. A área de interesse contínua não inclui obstáculos.

- **Tipo I**

Sensor circular de raio $r_I = \frac{\sqrt{2}}{4}$ e custo fixo $c_I = \frac{\pi}{8} u.m.$

Nº mínimo de sensores a instalar: -

Nº máximo de sensores disponíveis: 3 unidades.

- **Tipo II**

Sensor circular de raio $r_{II} = \frac{\sqrt{2}}{6}$ e custo fixo $c_{II} = \frac{\pi}{18} u.m.$

Nº mínimo de sensores a instalar: -

Nº máximo de sensores disponíveis: 3 unidades.

- **Tipo III**

$Sensor_{SC}$ de raio $r_{III} = \frac{\sqrt{2}}{6}$, amplitude $a_{III} = \frac{\pi}{2}$ e custo fixo $c_{III} = \frac{\pi}{72} u.m.$

Nº mínimo de sensores a instalar: 1 unidade.

Nº máximo de sensores disponíveis: 2 unidades.

Tendo em conta os dados apresentados, K^2 corresponde a um vetor de chaves aleatórias que codifica uma solução admissível para o problema de localização de sensores, considerando simultaneamente a análise de *quantos* sensores utilizar e *onde* fazer a respetiva localização.

$$K^2 = \underbrace{(0.03, 0.65, 0.12)}_{\text{Tipo I}}, \underbrace{(0.07, 0.92, 0.36)}_{\text{Tipo II}}, \underbrace{(0.57, 0.12)}_{\text{Tipo III}} | 0.09, 0.51, 0.54, 0.93, 0.12, 0.67, 0.05, 0.64 |$$

$$0.56, 0.72, 0.27, 0.14, 0.28, 0.78, 0.26, 0.65 | 0.25, 0.75 | 0.77, 0.42, 0.88)$$

As últimas três chaves aleatórias de K^2 são utilizadas para determinar o número de sensores de cada tipo que serão instalados:

Tipo I: $0.77 \times 3 = 2.31 \simeq 2$

Tipo II: $0.42 \times 3 = 1.26 \simeq 1$

Tipo III: $1 + 0.88 \times (2 - 1) = 1.88 \simeq 2$

A descodificação da solução K^2 encontra-se apresentada na tabela 3.2 e na figura 3.6 encontra-se a representação gráfica da respetiva solução.

	Ordem	Sequenciamento	Coordenada do centro	Orientação
Tipo I	1º	0.03	(0.09,0.56)	-
	2º	0.12	(0.54,0.27)	-
Tipo II	1º	0.07	(0.93,0.14)	-
Tipo III	1º	0.12	(0.64,0.65)	$0.25 \times 2\pi = \frac{\pi}{2} rad$
	2º	0.57	(0.05,0.26)	$0.75 \times 2\pi = \frac{3\pi}{2} rad$

Tabela 3.2: Descodificação para a solução K^2 .

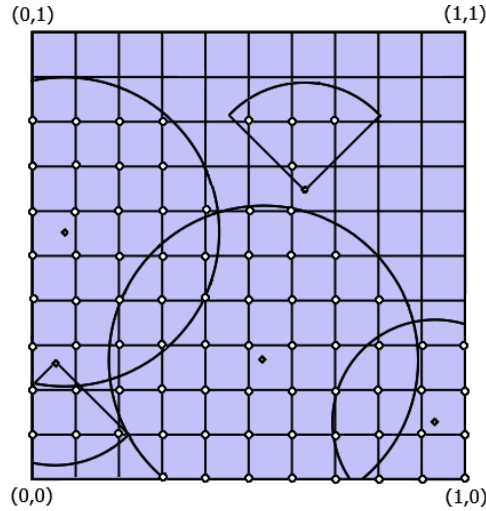


Figura 3.6: Solução S associada ao vetor K^2 de chaves aleatórias.

A solução decodificada apresenta uma proporção total de área por cobrir de 0.36 ($1 - \frac{78}{121} \simeq 1 - 0.64 = 0.36$), se se considerar o modelo binário para a determinação da cobertura e uma grelha de precisão $\delta = 0.1$ para a respetiva avaliação.

O custo total da solução, tendo em conta os custos fixos dos sensores e os custos variáveis, que dependem da distância ao gerador de energia localizado no ponto $(0,0)$, é de 1.7225 *u.m.*. A determinação deste valor, de acordo com a expressão enunciada na seção 2.3, é feita da seguinte forma:

$$\begin{aligned} \text{Custo}(S) &= C_A(S) + C_L(S) \\ &\simeq 1.0472 + 0.6753 = 1.7225 \end{aligned}$$

$$C_A(S) = 2 \times \frac{\pi}{8} + \frac{\pi}{18} + 2 \times \frac{\pi}{72} \simeq 1.0472$$

$$C_L(S) = \frac{\pi}{8} \times \sqrt{0.09^2 + 0.56^2} + \frac{\pi}{8} \times \sqrt{0.54^2 + 0.27^2} + \frac{\pi}{18} \times \sqrt{0.93^2 + 0.14^2} + \frac{\pi}{72} \times \sqrt{0.64^2 + 0.65^2} + \frac{\pi}{72} \times \sqrt{0.05^2 + 0.26^2} \simeq 0.6753$$

Sendo conhecidas todas as soluções da população onde a solução anterior se encontra inserida, com base nos valores calculados para os dois objetivos, determina-se o *rank* da solução S que corresponde, no problema biobjetivo, ao valor da respetiva aptidão.

O capítulo que se segue apresenta os casos de estudo que foram considerados e que correspondem a diferentes vertentes do problema de localização de sensores. O estudo destes casos será feito através da metodologia descrita no presente capítulo.

Capítulo 4

Casos de estudo

Os algoritmos genéticos, por serem bastante simples e eficientes, são uma técnica heurística amplamente utilizada em várias áreas de aplicação e, como tal, existem inúmeros trabalhos na literatura que estudam variações ao seu funcionamento básico. A metodologia apresentada no capítulo 3 pode, também, ser facilmente adaptada perante diferentes situações que surjam no âmbito do problema em estudo. É, precisamente, tirando partido desta característica que, nesta seção, se apresentam os diferentes casos do problema de localização de sensores que foram considerados na presente dissertação e os respetivos processos adotados para os resolver. O caso mais simples consiste na localização dos sensores em qualquer zona da área de interesse. Posteriormente considera-se a introdução de obstáculos que representam zonas de terra e restringe-se a localização destes meios de vigilância à região marítima. Com a introdução dos obstáculos surge, para além da questão da admissibilidade das soluções, a questão da linha de visão: alguns pontos deixam de ser visíveis, apesar de se encontrarem ao alcance dos sensores.

A descrição dos três casos de estudo e os métodos utilizados em cada um deles para adaptar o *biased random-key genetic algorithm* apresentam-se, em detalhe, de seguida. Note-se que todas as considerações que serão feitas no presente capítulo são válidas tanto para o problema mono-objetivo como para o problema biobjetivo.

4.1 Problema sem restrições

O primeiro caso de estudo admite que a região admissível para a localização dos sensores corresponde a qualquer zona do quadrado unitário que define a área de interesse. Este caso é bastante simples uma vez que a codificação das soluções utilizada no *biased random-key genetic algorithm* permite que as coordenadas de localização dos sensores sejam obtidas diretamente através dos valores das respetivas chaves aleatórias. Assim, para este caso de estudo, o algoritmo proposto não necessita de nenhum mecanismo adicional para ser executado. Esta simplicidade, que resulta da uniformidade da área de interesse que se resume à zona marítima, já não se verifica com a introdução de obstáculos, como se descreve nos casos de estudo que se seguem.

O problema de localização de sensores em estudo, tendo como motivação a proteção de portos, torna-se mais realista se forem incluídos obstáculos na área de interesse. Estes obstáculos pretendem representar a costa ou zonas de terra que possam existir no interior da região marítima. Por simplicidade, na presente dissertação, assume-se que os obstáculos são definidos através de polígonos sem falhas no seu interior.

As considerações que serão feitas daqui em diante podem ser adotadas quer se utilize o modelo binário ou o modelo probabilístico para o cálculo da proporção total de área por cobrir. Para facilitar a representação das soluções, as imagens que se apresentam neste capítulo consideram o modelo binário, encontrando-se a cinzento a área coberta por cada sensor. Note-se que, como apenas se pretende cobrir a área marítima, algumas zonas abrangidas pelos sensores que coincidem com obstáculos não são contabilizadas como área coberta pelos mesmos.

4.2 Admissibilidade das soluções

Consoante as características de cada sensor, a respetiva localização pode ser mais adequada na terra ou no mar. Sem perda de generalidade, nas explicações que se seguem, considera-se que todos os sensores têm de ser localizados na zona marítima.

Com a introdução de obstáculos e tendo em conta a restrição de localização definida no parágrafo anterior é facilmente perceptível que a codificação adotada para o algoritmo genético em estudo não garante a admissibilidade de todas as soluções. O aparecimento de soluções não admissíveis ocorre quando as coordenadas do centro de pelo menos um dos sensores da solução coincide com o interior de um obstáculo, conforme ilustra a figura 4.1. Perante este segundo caso de estudo em que os obstáculos introduzem a questão da admissibilidade, é necessário incluir um mecanismo que controle esta situação, mantendo a eficácia e simplicidade do BRKGA que se verificou para o primeiro caso de estudo e garantindo que na solução final obtida todos os sensores se encontram localizados no mar.

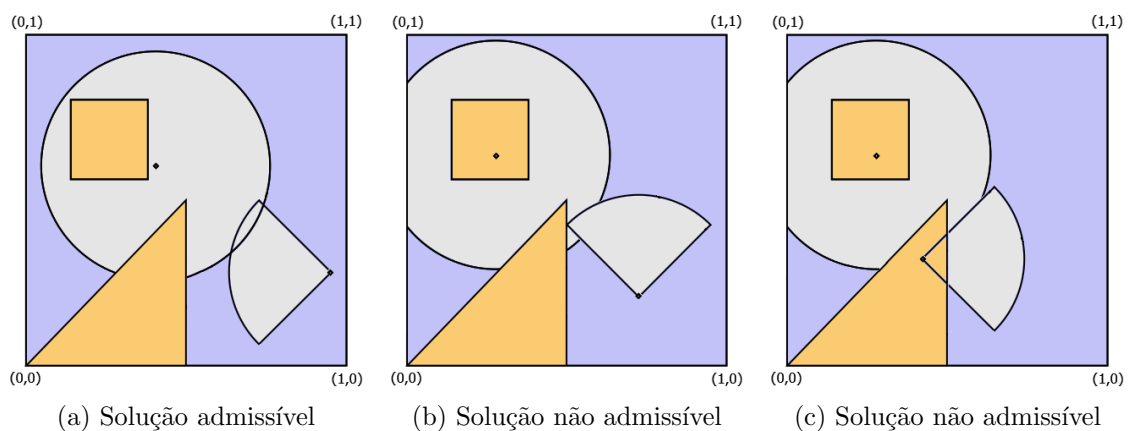


Figura 4.1: Introdução de obstáculos - questão da admissibilidade.

Os algoritmos genéticos, originalmente criados para lidar com problemas sem limitações, necessitam de mecanismos específicos para incorporar restrições. É essencial estudar a melhor forma de lidar com a questão da admissibilidade e, muitos autores, apresentam nos seus trabalhos diferentes abordagens para este problema.

De acordo com Gaspar-Cunha *et al.* [11], existem, essencialmente, cinco categorias para englobar as técnicas de tratamento de restrições: métodos baseados em funções de penalidade, métodos baseados na preservação da admissibilidade das soluções, métodos baseados na distinção entre soluções admissíveis e não admissíveis, métodos baseados em representações e operadores e métodos híbridos. Coello Coello [8] apresenta uma compilação dos métodos mais comuns para lidar com restrições em algoritmos evolutivos desde procedimentos mais simples a metodologias mais sofisticadas que variam consoante o problema em estudo e as restrições a considerar. De entre muitas alternativas possíveis, tendo em conta a codificação das soluções utilizada pelo BRKGA e a restrição na localização dos sensores, optou-se por aplicar, ao problema em estudo, um método de penalidade e um método de reparação das soluções. Estes métodos serão descritos, em detalhe, de seguida.

4.2.1 Método de penalidade

O método de penalidade é um dos métodos mais comuns para incorporar restrições nos algoritmos evolutivos e foi um dos primeiros métodos a ser estudado em particular em algoritmos genéticos. Neste método é adicionado um termo à função objetivo que penaliza a violação das restrições de modo a que as soluções admissíveis para o problema em estudo sejam favorecidas durante o processo de seleção de indivíduos. Este mecanismo deve ter em conta três princípios essenciais: i) a penalização deve ser proporcional ao grau de violação da restrição em causa, ii) nunca deve compensar violar a restrição que se está a considerar e iii) a penalização de uma solução válida é nula.

Existem vários métodos que podem ser englobados nesta classe de métodos de tratamento de restrições. Porém, é essencial garantir que a metodologia adotada se adequa ao problema em estudo tendo em conta a informação que se pode extrair de cada solução. Mesmo que um determinado método possa ser facilmente aplicado ao problema de localização de sensores, é necessário analisar rigorosamente se o critério de penalidade utilizado consegue guiar corretamente o algoritmo para o espaço de soluções admissíveis.

O método de penalidade por morte que consiste em rejeitar soluções não admissíveis, atribuindo o valor zero à aptidão de indivíduos que violam alguma das restrições em estudo, é, talvez, a técnica mais simples de implementar e apresenta bons resultados quando o espaço de soluções admissíveis corresponde a uma grande porção do espaço de procura [8]. Este método poderia ter sido facilmente aplicado ao problema em estudo mas existem essencialmente duas razões para se ter rejeitado a sua aplicação. Em primeiro lugar, caso os obstáculos ocupem uma grande parte da área de interesse, o espaço de soluções admissíveis corresponde a uma pequena porção do espaço de procura, o que faria com que este método apenas atingisse bons resultados para algumas instâncias do problema.

Além disso, caso a população inicial fosse exclusivamente formada por indivíduos inválidos (visto que é gerada aleatoriamente) o processo evolutivo iria “estagnar” uma vez que a todos os indivíduos seria atribuída a mesma aptidão.

Com o objetivo de tirar partido da simplicidade do método de penalidade por morte mas com o intuito de diferenciar o grau de violação das restrições de localização das soluções não admissíveis foi adotado um método que considera inativos os sensores localizados numa zona não permitida. Isto significa que um sensor localizado numa região não admissível não cobre nenhum ponto mesmo que esse ponto se encontre ao alcance do sensor – o sensor é, assim, inativo uma vez que a sua presença é completamente irrelevante a partir do momento em que se encontra instalado numa zona proibida. A área que, teoricamente, esse sensor cobre não é contabilizada para o cálculo da proporção total de área coberta. A figura 4.2 mostra um exemplo de uma solução para o problema de localização de sensores com um sensor circular e um sensor setor circular. Enquanto que em 4.2a a área coberta pelo sensor circular é contabilizada apesar de o mesmo se encontrar localizado numa zona não permitida, em 4.2b a área coberta por este sensor não é considerada uma vez que se está a aplicar o método de penalidade referido. Assim, ao penalizar a solução, apenas a área coberta pelo setor circular contribui para o cálculo da proporção total de área coberta e consequentemente para o cálculo do risco.

Recorde-se que o objetivo é proteger a zona marítima pelo que apenas se considera a área que os sensores cobrem no mar. Assim, caso existam zonas de obstáculos ao alcance dos sensores, esta mesma área não é contabilizada para a área que os sensores conseguem cobrir. Esta técnica de penalidade tem, assim, a desvantagem de apenas considerar o grau de violação das restrições no que diz respeito ao número e tipo de sensores localizados numa zona não admissível. Conforme ilustra a figura 4.3 existem soluções não admissíveis com a mesma proporção total de área coberta em que uma delas se encontra “mais perto” de se tornar admissível, no sentido em que a localização dos sensores inválidos se encontra

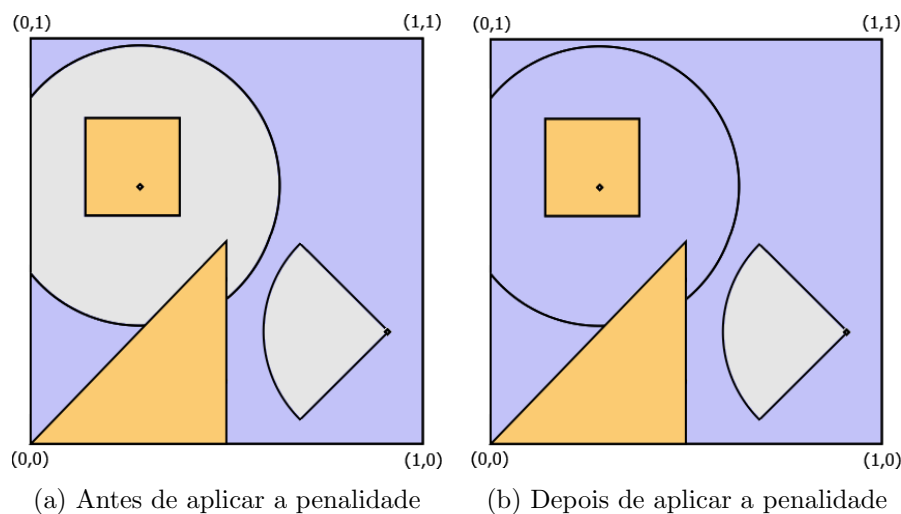


Figura 4.2: Método de penalidade aplicado a uma solução.

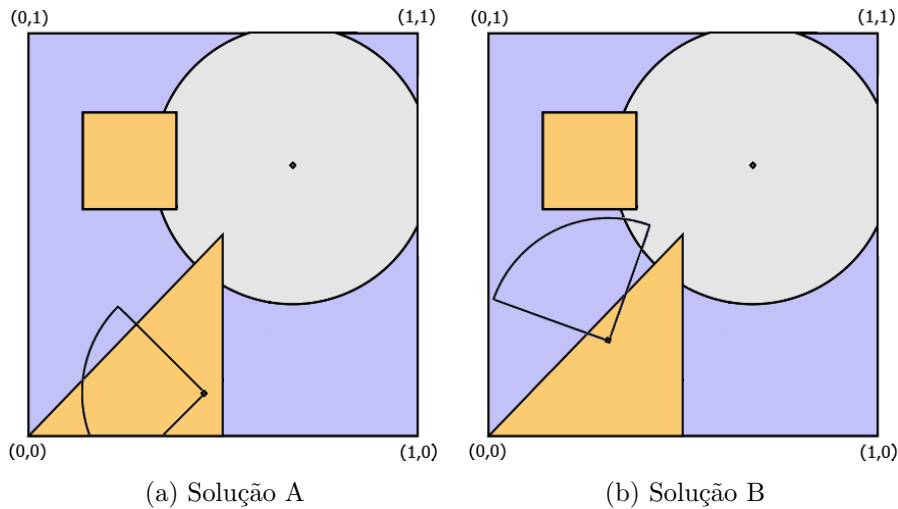


Figura 4.3: Exemplo ilustrativo da desvantagem do método de penalidade.

mais próxima da zona de fronteira com o mar. É precisamente o que sucede com a solução B uma vez que o centro do setor circular está bastante mais próximo da fronteira com o mar do que na solução A.

Apesar da introdução deste método no algoritmo genético proposto não garantir a admissibilidade de todas as soluções, o algoritmo consegue estabilizar o número de indivíduos inválidos ao longo das gerações. Consegue garantir também que, em geral, todas as soluções da elite são admissíveis na última geração o que garante a obtenção de uma solução final admissível para o problema em estudo. Estas observações serão referidas detalhadamente no capítulo 5 onde se descrevem os testes realizados e os respetivos resultados obtidos através da aplicação do BRKGA com o método de penalidade para o problema de localização de sensores numa área de interesse com obstáculos.

4.2.2 Método de reparação

O método de penalidade anteriormente referido, ao reduzir o valor da proporção total de área coberta de cada solução não admissível, consegue guiar o algoritmo para zonas do espaço de soluções que são válidas. Porém, não consegue evitar que indivíduos inválidos prossigam de geração em geração. Com o objetivo de tornar todas as soluções admissíveis, sugere-se um método alternativo para lidar com a questão da admissibilidade: o método de reparação. A técnica de reparação é uma boa opção quando uma solução não admissível pode ser facilmente transformada numa solução admissível e esta alteração não torna demasiado aleatório o processo evolutivo.

Alguns autores como é o caso de Guo *et al.* [14] e Xiao *et al.* [30] recorrem a algoritmos de reparação para melhorar o desempenho dos algoritmos genéticos que propõem evitando simultaneamente a existência de indivíduos que violem as restrições em estudo. Estes autores utilizam um conjunto de operadores genéticos cuidadosamente adaptados ao respetivo problema que fazem uso do conhecimento acerca do domínio que permite passar

da região não admissível para a região admissível, de forma eficiente. Em [14] o algoritmo de reparação desenvolvido para o problema é comparado com um método de penalidade e com um método de mutação forçada. Este último converte indivíduos inválidos em soluções que respeitam as restrições através de mutação. Porém, esta mutação “forçada” acaba por aumentar a taxa de probabilidade de mutação aproximando-se dos métodos de pesquisa aleatórios. Os autores concluem que a velocidade de convergência e os resultados de otimização obtidos através do algoritmo genético que recorre a métodos de reparação das soluções são superiores aos que se obtêm com o método de penalidade e o método de mutação forçada.

Não existe um procedimento *standard* para construir os algoritmos de reparação, existindo várias possibilidades para guiar o processo no espaço de soluções. Contudo, o sucesso desta abordagem no tratamento de restrições em algoritmos genéticos deve-se, essencialmente, à capacidade de se conseguir desenvolver uma heurística adequada ao problema em causa. Um outro aspeto a salientar que diverge de autor para autor é o facto de se poder utilizar um algoritmo de reparação apenas para determinar a aptidão da solução não admissível “reparada” ou, para além disso, fazer a substituição desta solução pela solução admissível obtida com uma determinada probabilidade de inserção na população. No primeiro caso, acredita-se que o indivíduo inválido melhora ao longo das gerações de forma natural, acabando por se tornar válido de acordo com a teoria evolutiva de *Lamarck*.

Tendo em conta as características específicas do problema em estudo apresenta-se, de seguida, o método de reparação de soluções não admissíveis adotado. Na presente dissertação optou-se por substituir sempre a solução não admissível pela correspondente solução “reparada”.

Dada uma solução S do problema, para cada sensor s cujo centro de coordenadas $centro_s = (x_s, y_s)$ se encontre numa região não admissível, o processo de reparação resume-se, essencialmente, aos dois passos que se apresentam de seguida.

- **Passo 1**

Determinar o ponto (x^*, y^*) para o qual é mínima a distância do ponto (x_s, y_s) a uma aresta que delimita o obstáculo no qual o sensor está localizado e não coincidente com as arestas que definem a área de interesse.

- **Passo 2**

O ponto (x^*, y^*) pertence à fronteira que delimita a zona de terra. De forma a garantir que o sensor passa a estar localizado na zona admissível, introduz-se um fator ϵ pequeno às coordenadas determinadas de forma a que o ponto (x^*, y^*) seja deslocado no sentido da região marítima.

As coordenadas do centro do respetivo sensor são atualizadas, correspondendo às coordenadas do ponto assim determinado, ou seja, $(x_s, y_s) = (x^* + \epsilon, y^* + \epsilon)$, podendo-se considerar, por exemplo, $\epsilon = 0.0001$ ou $\epsilon = -0.0001$ consoante o sentido do deslocamento.

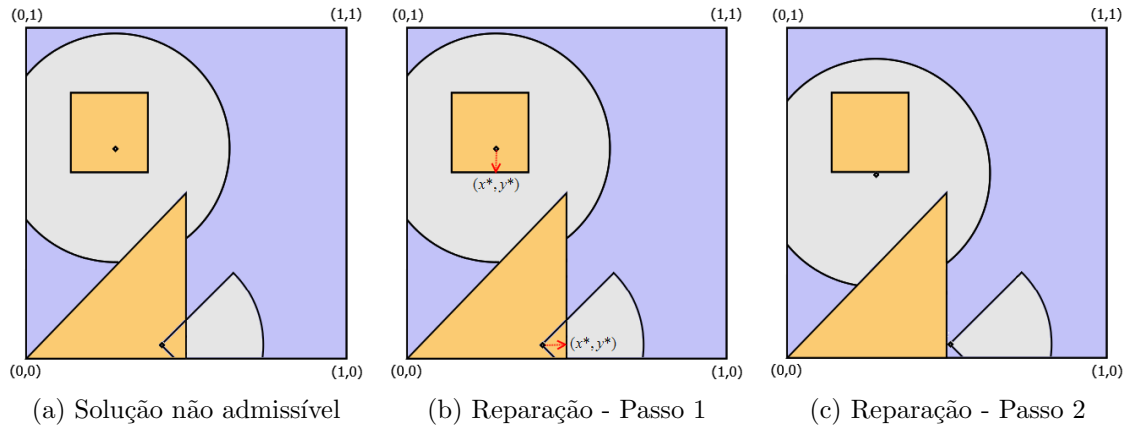


Figura 4.4: Aplicação do método de reparação a uma solução do problema.

Ao alterar as coordenadas do centro dos sensores que se encontravam localizados numa região não permitida, é necessário atualizar a cobertura e, no caso do problema biobjetivo, o custo da nova solução obtida.

A figura 4.4a apresenta uma solução não admissível onde ambos os sensores se encontram localizados em regiões não permitidas. Ao aplicar o mecanismo de reparação, determina-se, no passo 1, o ponto (x^*, y^*) que, para cada sensor se encontra mais próximo da fronteira que separa os obstáculos do mar (figura 4.4b). Note-se que, no caso do sensor setor circular, a fronteira que coincide com uma das arestas do quadrado unitário que define a zona de interesse não é considerada para a determinação deste ponto. Os sensores são deslocados, conforme ilustra a figura 4.4c, para o ponto determinado ajustado de um fator ϵ pequeno, de modo a que os respetivos centros não fiquem exatamente na fronteira mas sim no região marítima conforme se pretende.

No capítulo 5 analisa-se e discute-se a aplicação do método de reparação ao *biased random-key genetic algorithm* de forma a garantir a obtenção de soluções admissíveis para o problema de localização de sensores. Este método será comparado com o método de penalidade no que diz respeito aos valores obtidos para as soluções do problema e em relação aos tempos de computação. As principais conclusões serão mencionadas no mesmo capítulo.

4.3 Linha de visão

Igualmente motivado pela introdução de obstáculos na área de interesse surge, para além da questão da admissibilidade das soluções anteriormente referida, o terceiro caso de estudo em que se pretende analisar a redução do campo de visão de um sensor ao encontrar zonas não marítimas. Enquanto que para a questão da admissibilidade é difícil encontrar na literatura trabalhos sobre problemas de redes de sensores que abordem diretamente o assunto, em relação à obstrução da linha de visão com a introdução de obstáculos existem

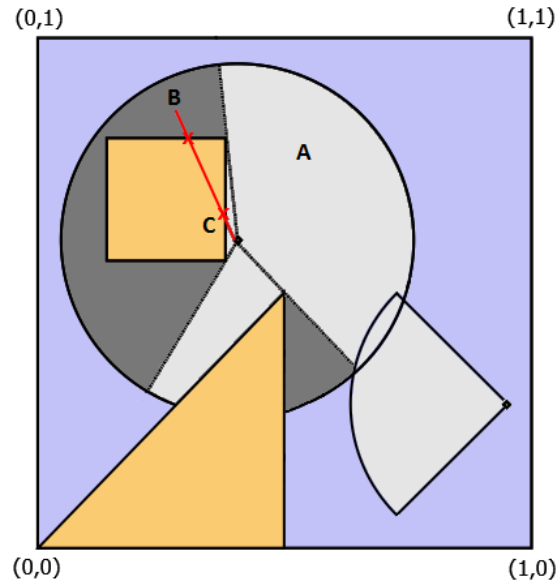


Figura 4.5: Redução do campo de visão com a introdução de obstáculos.

algumas contribuições [2, 25, 28, 29]. Akbarzadeh *et al.* [2] apresentam, no seu trabalho, um contributo bastante relevante para este assunto, pretendendo otimizar a localização de sensores tendo em conta a topografia do terreno e um modelo de cobertura probabilística. Os autores salientam a importância do estudo da linha de visão na presença de obstáculos e realçam o facto de que a maioria das abordagens existentes simplifica demasiado o problema em causa, comprometendo, assim, a qualidade dos resultados obtidos.

A inclusão de obstáculos pode levar à obstrução da linha de visão de determinados pontos que, apesar de se encontrarem dentro da área de deteção dos sensores, não são cobertos pelos mesmos. Na figura 4.5, onde a região colorida a cinzento-escuro representa o conjunto de pontos que perde a visibilidade, ao contrário do ponto A que é coberto pelo sensor, o ponto B encontra-se numa zona de sombra e não é coberto pelo mesmo uma vez que, nessa direção, o sensor perde a visibilidade ao atingir o ponto C. A questão da determinação da linha de visão encontra-se, assim, diretamente relacionada com a definição de cobertura de um sensor, como se irá explicar de seguida.

Sendo $A - O$ a zona da área de interesse que exclui os obstáculos, a existência de linha de visão entre um ponto $u = (x_u, y_u)$ e um sensor s de centro $centro_s = (x_s, y_s)$ é definida através da variável binária $\vartheta(u, s)$ tal que:

$$\vartheta(u, s) = \begin{cases} 1, & \text{se existe visibilidade entre } u \text{ e } s \\ 0, & \text{caso contrário} \end{cases}$$

Este terceiro caso de estudo assume que é previamente adotado um dos mecanismos anteriormente mencionados para garantir a obtenção de soluções admissíveis. Se for ado-

tado o método de penalidade não faz sentido determinar a visibilidade dos pontos a partir de sensores inativos e se for utilizado o mecanismo de reparação existe a garantia de que todos os sensores da solução se encontram localizados na região marítima. Tendo em conta os pressupostos referidos, assume-se que s está localizado no mar e o procedimento adotado para determinar o valor a atribuir a $\vartheta(u, s)$ consiste em verificar se o segmento que une o $centro_s$ ao ponto u intersesta algum obstáculo, ou seja, $\vartheta(u, s)=1$ se $ku + (1 - k)centro_s \in A - O, \forall k \in [0, 1]$ e $\vartheta(u, s)=0$ caso contrário. Para tal, verifica-se se o segmento $[centro_s, u]$ intersesta alguma das arestas que definem a fronteira entre os obstáculos e a zona marítima. No caso em que pelo menos uma das arestas é intersestada não existe visibilidade. Retomando o exemplo da figura 4.5, a obstrução da linha de visão entre o centro do sensor e o ponto B é garantida pela interseção do segmento definido por estes pontos com duas das arestas que definem um dos obstáculos da área de interesse.

Independentemente de se adotar o modelo de cobertura binário ou o modelo de cobertura probabilístico, a existência de visibilidade é uma condição necessária para a cobertura de um ponto por um sensor. Isto significa que, neste terceiro caso de estudo, a cobertura de um ponto u através de um sensor s passa a ser definida como: $Cob'(u, s) = Cob(u, s) \times \vartheta(u, s)$, sendo que $Cob(u, s)$ representa a cobertura do ponto u através do sensor s tal como se definiu no capítulo 2. Em relação ao problema biobjetivo, o custo total associado à solução em estudo não sofre qualquer alteração.

Partindo da metodologia do *biased random-key genetic algorithm*, o problema de localização de sensores foi estudado tendo em consideração os casos apresentados no presente capítulo. Para os três casos de estudo que refletem diferentes características do problema em causa foram realizados testes experimentais cujos resultados se apresentam, comparam e discutem no capítulo que se segue.

Capítulo 5

Análise e Discussão

Neste capítulo analisa-se e discute-se a aplicação do *biased random-key genetic algorithm* ao problema de localização de sensores. Para tal, foram realizados diversos testes que pretendem ser elucidativos do comportamento desta heurística para cada um dos três casos de estudo apresentados. O algoritmo foi implementado em MATLAB na versão R2009a e, apesar de ter sido desenvolvido para o problema de localização de sensores no contexto da proteção de portos, as características desta metodologia permitem que o *software* desenvolvido seja facilmente utilizado noutras aplicações, através da adaptação do decodificador. Os resultados que se apresentam neste capítulo foram obtidos através da execução da heurística num PC com um processador *Intel®Core™ 2DUO CPU E7500 @ 2.93GHz*.

As imagens que se irão apresentar correspondentes às soluções do problema para cada situação em particular foram obtidas recorrendo às capacidades gráficas do MATLAB. Para não sobrecarregar a apresentação gráfica das soluções não se apresentam as coordenadas dos centros dos sensores. Porém, sendo fundamental identificar as coordenadas onde o centro de cada sensor será instalado, esta informação é apresentada num ficheiro anexo criado automaticamente pelo programa aquando da sua execução. A análise gráfica permite ter uma boa perceção do comportamento do algoritmo pelo que, para efeitos de discussão e análise na presente dissertação, estes valores não serão apresentados.

5.1 Problema mono-objetivo

Como foi referido nos capítulos anteriores, o estudo do problema mono-objetivo deve-se à relevância da minimização do risco para problemas de redes de sensores. Em particular, para o problema em estudo, é essencial garantir que a área de interesse é bem monitorizada para minimizar a probabilidade de uma ameaça não ser detetada. Tendo em conta o especial interesse da análise deste critério, o problema mono-objetivo foi estudado para os vários casos apresentados, considerando que a configuração no número e tipo de sensores é conhecida *a priori*. Todos os testes que se apresentam consideram o modelo de cobertura binária para o cálculo da proporção de área por cobrir.

5.1.1 Problema sem restrições

Uma boa heurística deve reunir, essencialmente duas qualidades: eficiência e eficácia. A heurística deve obter em tempo útil uma boa solução para as instâncias do problema que se pretendem testar. Considera-se que a solução obtida é “boa” quando o valor da função objetivo obtido para a solução é próximo do valor ótimo do problema. É precisamente com o objetivo de testar a qualidade da metodologia de resolução adotada que se consideram os subproblemas que se apresentam nesta seção.

Para os testes realizados, considera-se a existência de dois tipos de sensores:

- **Tipo I**

Sensor circular de raio $r_I = \frac{\sqrt{2}}{4}$.

Número de sensores que podem ser instalados: $0 \leq n_I \leq 4$.

- **Tipo II**

Sensor circular de raio $r_{II} = \frac{\sqrt{2}}{6}$.

Número de sensores que podem ser instalados: $0 \leq n_{II} \leq 9$.

Daqui em diante, para o estudo do problema mono-objetivo serão considerados 50 subproblemas correspondentes às diferentes combinações possíveis no número de sensores a instalar do tipo I e II. A escolha destes tipos de sensores para a realização dos testes que se irão apresentar deve-se ao facto de serem conhecidos alguns dos valores ótimos para os subproblemas considerando apenas a minimização do risco no caso em que a área de interesse corresponde apenas à zona marítima. Além disso, a minimização do risco no quadrado unitário foi estudada através de outras heurísticas para este conjunto de subproblemas pelo que, mesmo não sendo conhecido o valor ótimo, conhece-se o melhor valor obtido até ao momento. Estes valores encontram-se apresentados na tabela 5.1 e designam-se por valores de referência.

A figura 5.1 apresenta a solução ótima para dois dos subproblemas cujo valor ótimo corresponde à cobertura da totalidade da área de interesse, ou seja, o risco associado a estas soluções é nulo.

A simplicidade do problema mono-objetivo para o problema sem restrições irá permitir a otimização dos parâmetros que será posteriormente generalizada para áreas de interesse que considerem obstáculos e para o problema biobjetivo.

Apesar de serem apenas considerados sensores circulares para a análise que se apresenta nesta seção e para a comparação dos resultados obtidos no estudo da admissibilidade das soluções e da influência da linha de visão, apresentam-se na seção 5.1.4 alguns exemplos que ilustram o comportamento dos sensores cuja área de cobertura corresponde a um setor circular. Nessa seção, serão feitas algumas considerações sobre o funcionamento destes sensores nos três casos em estudo.

		Tipo I				
		0	1	2	3	4
Tipo II	0	1	0,6073	0,3095	0,1216	0
	1	0,8255	0,4328	0,1670	0,0117	0
	2	0,6509	0,2889	0,0400	0,0010	0
	3	0,4764	0,1546	0,0130	0	0
	4	0,3019	0,0751	0,0032	0	0
	5	0,1973	0,0197	0	0	0
	6	0,1208	0,0064	0	0	0
	7	0,0584	0	0	0	0
	8	0,0195	0	0	0	0
	9	0	0	0	0	0

Tabela 5.1: Valores de referência - Proporção total máxima de área por cobrir.

Antes de se apresentarem os resultados obtidos para o problema sem restrições, são referidos os valores dos parâmetros que foram adotados para a realização dos respectivos testes.

Otimização dos parâmetros

A escolha dos parâmetros utilizados é de extrema importância uma vez que influencia a qualidade dos resultados obtidos. Assim, de forma a serem obtidos os melhores resultados, os parâmetros necessários à execução da metodologia adotada foram previamente otimizados.

Segue-se uma breve descrição da importância dos parâmetros e apresenta-se o valor que será considerado, para cada um deles, nos testes que se apresentam daqui para a frente.

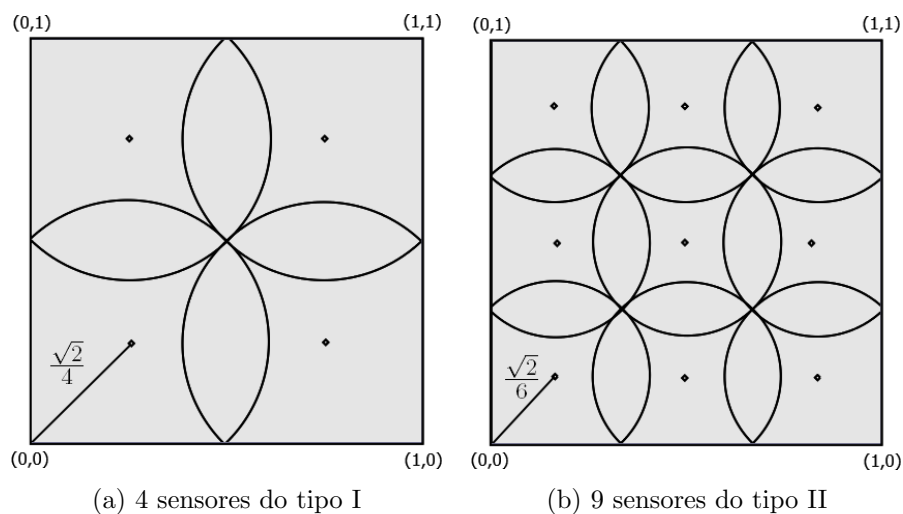


Figura 5.1: Soluções ótimas para dois subproblemas.

- **Precisão da grelha de pontos para a avaliação da cobertura - δ**

O valor de δ está diretamente relacionado com a precisão dos resultados obtidos. Como se ilustrou na figura 2.5, uma menor precisão da grelha de pontos implica maior probabilidade de o método parar em soluções subótimas. Por outro lado, uma maior precisão leva a um aumento no tempo de computação para obter a solução do problema. Assim, o valor considerado para δ tem de refletir um compromisso entre os dois fatores anteriormente mencionados.

O valor adotado para δ nos testes realizados foi de $\delta = 0.01$.

- **Parâmetros do *biased random-key genetic algorithm***

Existem alguns parâmetros no funcionamento do *biased random-key genetic algorithm* que precisam de ser definidos, nomeadamente a proporção de indivíduos que pertencem à elite (p_e), a proporção de indivíduos gerados através de mutação (p_m) e a probabilidade de um descendente herdar cada alelo do progenitor de elite (ρ_e).

Com o objetivo de determinar os melhores valores para cada um dos parâmetros, os mesmos foram testados em simultâneo fixando o número de indivíduos da população e o número de gerações correspondentes à execução do algoritmo. Assim, sob as mesmas condições, é possível determinar qual a combinação para os valores dos três parâmetros para a qual se obtêm os melhores resultados. Para tal, tendo em conta o intervalo de valores recomendados para cada parâmetro [12] e os resultados dos testes realizados que se encontram no anexo A.1.1, os valores adotados foram os seguintes:

$$\diamond p_e = 0.10$$

$$\diamond p_m = 0.30$$

$$\diamond \rho_e = 0.70$$

- **Dimensão da população**

A dimensão da população, p , é um parâmetro de grande importância, afetando diretamente o desempenho global e a eficiência de um algoritmo genético. Uma vez que o funcionamento de um algoritmo genético é extremamente sensível ao tamanho da população, é importante que o valor adotado seja cuidadosamente otimizado. Caso o número de indivíduos na população seja demasiado reduzido, não existirá variedade genética e existe a possibilidade de não serem obtidas boas soluções. Uma grande população fornece, geralmente, uma cobertura representativa do domínio do problema, prevenindo convergências prematuras para ótimos locais mas leva a um aumento no tempo de computação, podendo aproximar o algoritmo de um processo de busca exaustiva o que não é pretendido.

Assim, deverá ser escolhido um valor sensato para p que procure obter soluções diversificadas num tempo de CPU razoável. De forma a ponderar os dois critérios

anteriormente mencionados, a otimização deste parâmetro foi feita em simultâneo com o número de iterações em que o algoritmo é executado. Os resultados dos testes realizados, conforme se apresenta no anexo A.1.2, mostram que $p = 50$ é um bom valor de compromisso entre os resultados obtidos e o tempo de CPU.

- **Critério de Paragem**

Existem vários critérios de paragem que podem ser adotados para determinar quando é que se deve terminar a execução de um algoritmo genético. Estes critérios podem ter em consideração, por exemplo, um determinado número de gerações sem existir melhoria na aptidão do melhor indivíduo, a falta de diversidade entre as soluções da população ou um determinado número pré-definido de iterações. Neste trabalho, optou-se por considerar este último critério, correspondendo cada iteração a uma geração do processo evolutivo. O número de gerações, nG , em que o algoritmo é executado deve ser um parâmetro cuidadosamente escolhido uma vez que este não é um critério tão específico como aqueles que se baseiam na qualidade das soluções.

A otimização do número de gerações a considerar é feita simultaneamente com a otimização da dimensão da população. Desta forma, pretende-se equilibrar a existência de diversidade entre os indivíduos da população com o tempo de computação necessário para obter soluções de boa qualidade. De acordo com os resultados obtidos que se encontram no anexo A.1.2, os testes que se apresentam executam o algoritmo durante $nG = 500$ gerações.

Salienta-se o facto de, apesar da otimização dos parâmetros ser feita para o problema sem restrições, se adotarem os valores obtidos para os restantes casos de estudo que incluem obstáculos na área de interesse. Não seria possível otimizar os parâmetros para uma área de interesse com obstáculos sem ter conhecimento *a priori* da mesma. Como se pretende que os parâmetros estejam definidos independentemente da informação adicional que é específica de cada instância do problema, as considerações feitas para os parâmetros do problema sem restrições são aplicadas ao problema que estuda a admissibilidade das soluções e ao problema da linha de visão.

Avaliação da qualidade da heurística

Aplicando o *biased random-key genetic algorithm* (BRKGA) ao problema mono-objetivo de localização de sensores em que a área de interesse corresponde apenas à zona marítima, considerando os parâmetros anteriormente otimizados, é possível analisar a qualidade desta heurística, comparando os resultados obtidos com os valores de referência apresentados na tabela 5.1. Para tal, cada um dos 50 subproblemas foi executado 5 vezes, sendo retido o melhor valor para a proporção de área por cobrir. Estes resultados encontram-se na tabela 5.2, assinalando-se, a vermelho, os subproblemas para os quais o valor obtido coincide com o valor de referência.

		Tipo I				
		0	1	2	3	4
Tipo II	0	1	0.6142	0.3143	0.1254	0.0004
	1	0.8279	0.4428	0.1714	0.0133	0.0001
	2	0.6557	0.2952	0.0434	0.0029	0
	3	0.4840	0.1615	0.0149	0	0
	4	0.3133	0.0807	0.0069	0	0
	5	0.2023	0.0225	0.0005	0	0
	6	0.1262	0.0106	0	0	0
	7	0.0632	0.0002	0	0	0
	8	0.0232	0.0001	0	0	0
	9	0.0002	0	0	0	0

Tabela 5.2: Proporção total de área por cobrir - caso de estudo sem restrições.

Na figura 5.2 apresentam-se as melhores soluções encontradas pelo *biased random-key genetic algorithm* para o problema que considera a instalação de 4 sensores do tipo I e para o problema que considera que são localizados 9 sensores do tipo II. Comparando estas soluções com as respectivas soluções ótimas apresentadas na figura 5.1 conclui-se que o algoritmo consegue obter soluções muito semelhantes. Apesar destas soluções não corresponderem à cobertura total da área de interesse, a proporção de área por cobrir é relativamente baixa correspondendo, respetivamente, a 0.0004 e 0.0002.

Para uma melhor análise comparativa dos valores obtidos em cada subproblema $i = 1, \dots, 50$ pelo BRKGA, v_i , com os valores de referência, v_i^* , referidos na tabela 5.1, o gráfico da figura 5.3 apresenta os valores dos erros $e_i = v_i - v_i^*$. Uma vez que o problema em estudo é um problema de minimização e os valores de referência podem ser considerados, para o efeito, como os verdadeiros valores ótimos de cada subproblema, os valores dos erros são sempre não negativos.

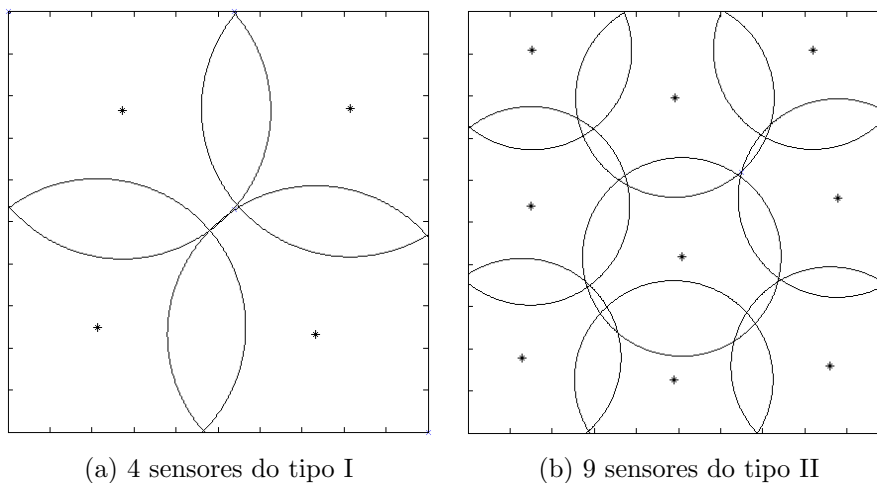


Figura 5.2: Soluções obtidas pelo BRKGA para dois subproblemas.

A numeração dos subproblemas que se apresenta no gráfico corresponde às várias posições da tabela 5.2 iniciando a contagem da esquerda para a direita, sequencialmente pelas linhas apresentadas. O subproblema 1 corresponde a 0 sensores do tipo I e 0 sensores do tipo II, o subproblema 2 corresponde a 1 sensor do tipo I e 0 sensores do tipo II e assim sucessivamente.

A tabela 5.2 conjuntamente com o gráfico da figura 5.3 permite concluir que:

- Em 21 dos 50 subproblemas o valor obtido pelo algoritmo coincide com o valor de referência.

Note-se que a técnica de ajustar uma grelha de pontos para determinar a área coberta pelos sensores permite apenas uma aproximação ao verdadeiro valor de área que os sensores conseguem cobrir. Esta situação é facilmente visível no caso em que se considera um único sensor do tipo I e no caso em que se pretende instalar um único sensor do tipo II. A solução obtida em ambos os casos pelo BRKGA consegue incluir toda a área de cobertura dos sensores dentro da área de interesse. Porém, como o valor obtido é uma aproximação ao verdadeiro valor, o mesmo não coincide com o valor de referência, apesar de ser óbvio que esta é a solução ótima.

- Em todos os subproblemas os desvios aos valores de referência são inferiores a 0.015.
- O algoritmo consegue obter resultados mais próximos do valor de referência perante a existência de um maior número de sensores. Os erros obtidos são mais significativos quando existem sensores do tipo I em menor quantidade.

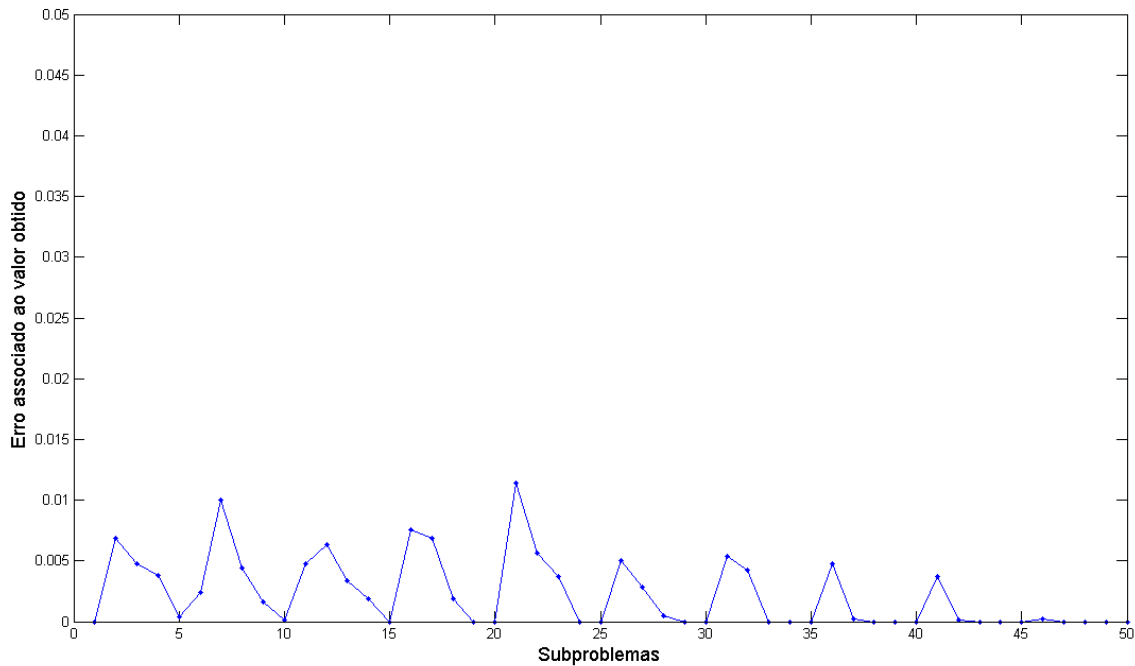


Figura 5.3: Erros associados ao valor obtido pelo BRKGA em cada subproblema.

O tempo total médio de CPU para a execução dos 50 subproblemas é de 42.45 minutos. Na tabela B.5 do anexo B.2 apresentam-se os tempos médios de CPU para cada subproblema, tendo em conta as 5 execuções realizadas.

Tendo em conta as observações feitas nesta subsecção, pode-se concluir que, em geral, os resultados obtidos pelo *biased random-key genetic algorithm* são bastante satisfatórios. Nas subsecções 5.1.2 e 5.1.3 analisa-se o comportamento do BRKGA para o problema de localização de sensores em que a área de interesse inclui obstáculos.

5.1.2 Admissibilidade das soluções

Ao contrário do que acontece no caso mais simples em estudo, com a introdução de obstáculos os valores ótimos de cada subproblema passam a depender de cada área de interesse em particular. Assim, serão utilizados os parâmetros anteriormente otimizados nos testes que se seguem de forma a generalizar a utilização do algoritmo a qualquer área de interesse sem existir uma preocupação adicional na escolha dos melhores parâmetros.

Para o estudo das variantes do problema em que são introduzidos obstáculos, os testes realizados utilizam a área de interesse que se apresenta na figura 5.4. O algoritmo poderia ter sido testado numa grande variedade de áreas de interesse com diferentes características. A escolha desta área de interesse pretende, sem perda de generalidade, ilustrar o funcionamento do algoritmo nestes casos de estudo, de forma a que se possa compreender o seu comportamento para que, de futuro, possa ser aplicado a dados reais da linha de costa. De modo a facilitar a interpretação e análise dos resultados que se apresentam, as imagens que se apresentam, resultantes do programa desenvolvido, mantêm as cores apresentadas nos exemplos dos capítulos anteriores: as zonas que correspondem a obstáculos encontram-se coloridas a bege e a região marítima corresponde à tonalidade azul.

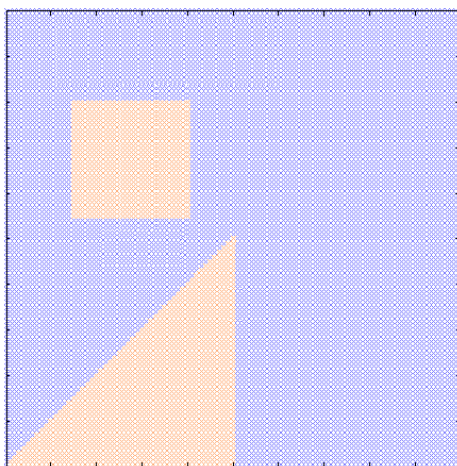


Figura 5.4: Área de interesse utilizada nos testes realizados.

Apesar do programa se encontrar preparado para definir qual a região admissível para a localização de cada tipo de sensor, considera-se, nos testes realizados, que tanto os sensores do tipo I como os sensores do tipo II só podem ser localizados na região marítima com o intuito de detetar ameaças com origem no mar.

Nesta seção apresentam-se os resultados obtidos através da aplicação do *biased random-key genetic algorithm* ao problema que estuda a admissibilidade das soluções. Os dois métodos propostos (método que considera os sensores inválidos inativos e método de reparação) serão analisados e discutidos de forma a perceber qual o que melhor se adequa ao problema em estudo. Cada um dos 50 subproblemas foi executado 5 vezes sendo retido o melhor valor obtido para a proporção total de área por cobrir para cada um dos métodos em análise. Note-se que esta proporção é determinada tendo apenas em consideração a área que exclui os obstáculos. Assim, a cobertura total da área de interesse diz respeito à cobertura da totalidade da área marítima.

O gráfico da figura 5.5 apresenta a diferença entre o melhor valor obtido para a proporção total de área por cobrir através do método de penalidade que considera inativos sensores inválidos, V_P , e o melhor valor obtido através do método de reparação, V_R .

A análise do gráfico da figura 5.5 permite concluir os seguintes aspetos:

- Os valores positivos apresentados no gráfico permitem concluir que em 13 dos 50 subproblemas o método de penalidade obteve melhores resultados que o método de reparação. Por outro lado, em 7 dos 50 subproblemas foi o método de reparação

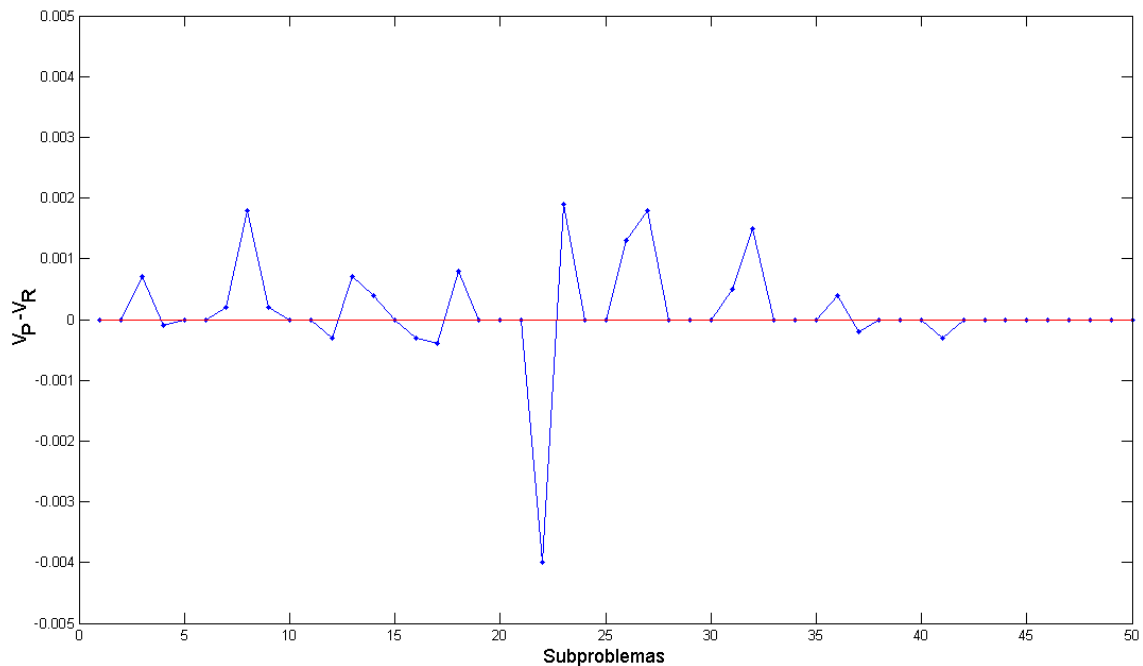


Figura 5.5: Análise comparativa dos métodos estudados para a admissibilidade.

que conseguiu obter melhores valores para a proporção total de área por cobrir. Nos restantes 30 subproblemas ambos os métodos obtiveram o mesmo resultado.

- Apesar do método de penalidade produzir, em geral, melhores resultados, o método de reparação obtém um valor significativamente melhor para o problema que considera um sensor do tipo I e 4 sensores do tipo II.
- Parece não existir uma relação evidente entre o método utilizado e as características dos subproblemas em que se obtém melhores resultados com um ou outro método. Os dois métodos produzem, em geral, resultados muito semelhantes, não existindo evidência para preferir um deles.
- Apesar de não serem conhecidos os valores ótimos de cada subproblema uma vez que se introduziram obstáculos na área de interesse, espera-se que os valores obtidos não se afastem muito desse valor, principalmente nos casos em que os dois métodos obtêm os mesmos resultados.

Considerando, como exemplo, o subproblema em que são instalados 4 sensores do tipo I e o subproblema em que são considerados 9 sensores do tipo II, as soluções apresentadas na figura 5.6 relativas à aplicação do método de penalidade correspondem a uma proporção de área por cobrir de, respetivamente, 0.001 e 0.

Tal como foi referido na seção 4.2.1 do capítulo 4, o método de penalidade não garante a admissibilidade de todos os indivíduos da população. No entanto, a utilização deste método pretende, precisamente, guiar o algoritmo para o espaço de soluções admissíveis, garantindo a obtenção de uma solução final em que todos os sensores se encontrem localizados no mar.

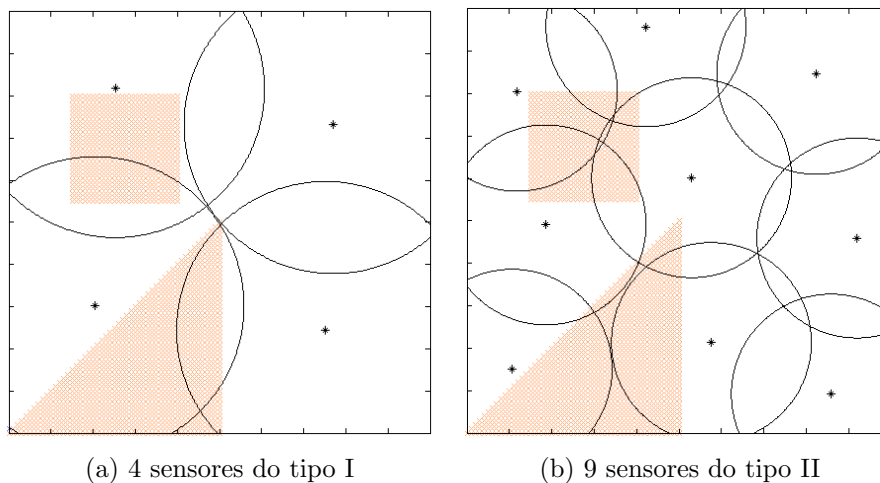
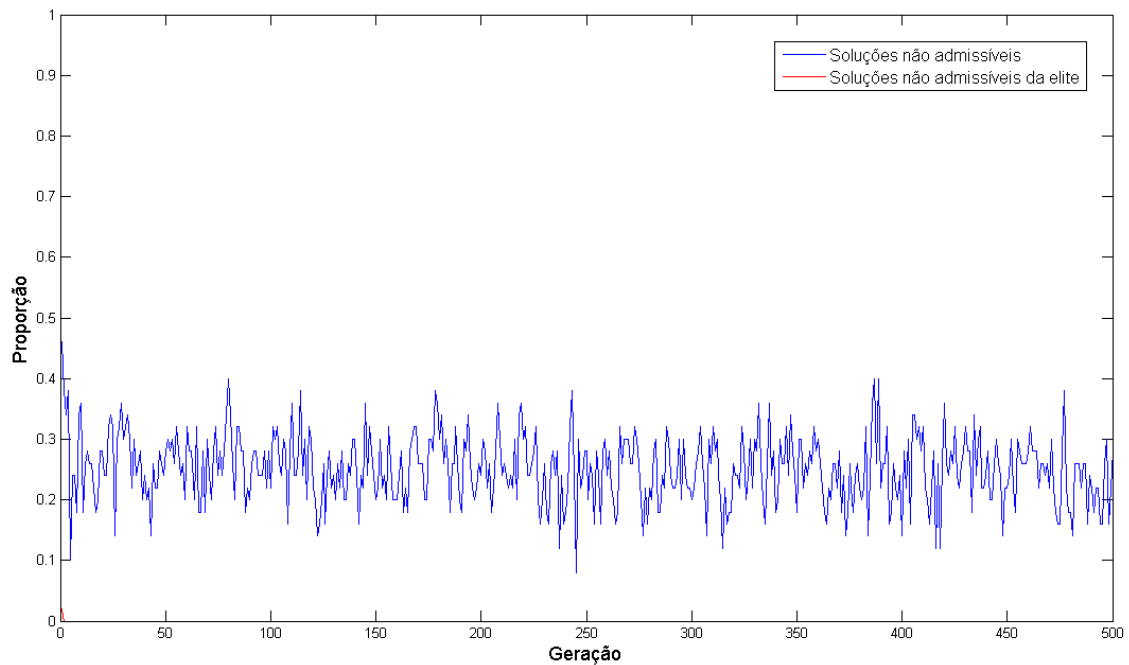
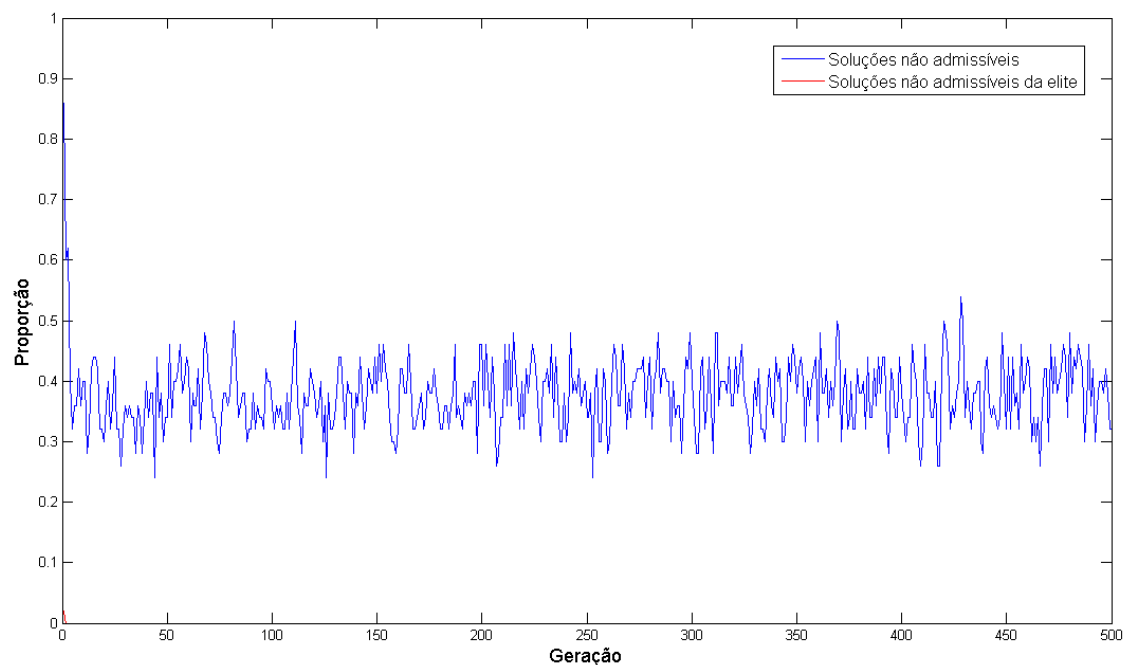


Figura 5.6: Soluções obtidas pelo BRKGA aplicando o método de penalidade.

Os gráficos que se apresentam na figura 5.7 mostram o comportamento da heurística no que diz respeito à evolução do número de soluções não admissíveis ao longo das 500 gerações em que é executada para obter as soluções apresentadas na figura 5.6.



(a) 4 sensores do tipo I



(b) 9 sensores do tipo II

Figura 5.7: Evolução do número de soluções não admissíveis ao longo das gerações.

A análise dos gráficos da figura 5.7 permite concluir que o número de soluções não admissíveis tem tendência a estabilizar, oscilando, ao longo das gerações, em torno de um valor que depende das características do subproblema em estudo. Na primeira geração o número de soluções não admissíveis é significativamente mais elevado uma vez que a população inicial é gerada aleatoriamente.

A obtenção de soluções admissíveis torna-se mais difícil quanto maior for o número de sensores a localizar uma vez que é maior a probabilidade de o centro de pelo menos um sensor coincidir com um obstáculo. Este aspeto é evidenciado na análise comparativa dos dois gráficos da figura 5.7 uma vez que o número de soluções não admissíveis é, em geral, mais elevado em cada geração para o subproblema que considera 9 sensores do tipo II.

Um outro aspeto importante a salientar é o facto dos indivíduos da elite corresponderem todos a soluções admissíveis na última geração, o que garante que a solução obtida é admissível uma vez que corresponde ao indivíduo mais apto da população. Verifica-se, em ambos os subproblemas, que inicialmente uma pequena proporção dos indivíduos da elite corresponde a soluções não admissíveis. Porém, rapidamente esta proporção se torna nula, mantendo-se com este valor até à geração final.

Apesar de se obter o mesmo valor para ambos os subproblemas anteriormente mencionados ao aplicar o método de reparação (a proporção de área por cobrir é de, respetivamente, 0.0001 e 0), as soluções obtidas são ligeiramente diferentes e encontram-se apresentadas na figura 5.8.

Note-se que uma solução ótima conhecida para o problema sem restrições só deixa de o ser caso os obstáculos intersectem as coordenadas de localização ótima dos sensores nessa solução. É precisamente esta situação que se verifica nos dois subproblemas referidos, sendo necessário adotar os métodos estudados para a obtenção de uma solução admissível.

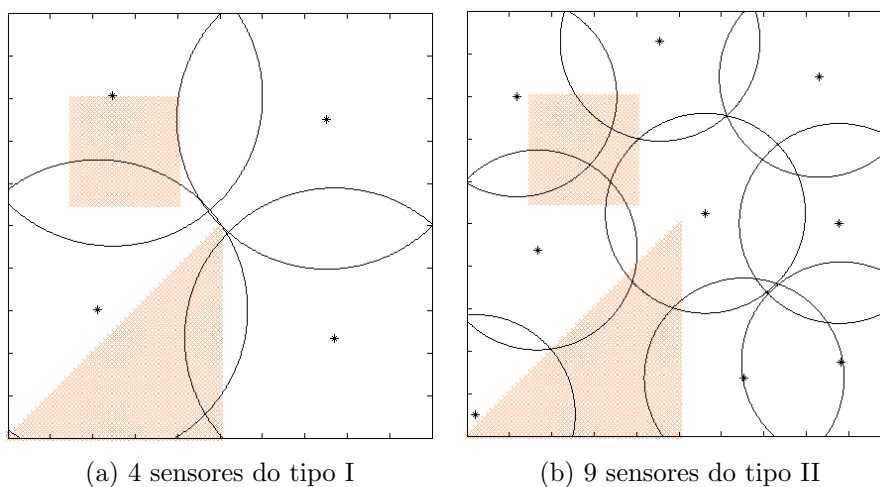


Figura 5.8: Soluções obtidas pelo BRKGA aplicando o método de reparação.

Em relação aos tempos de CPU, o gráfico da figura 5.9 compara os resultados obtidos. Apresenta-se, para cada subproblema, a média dos tempos de CPU obtidos considerando as 5 execuções realizadas.

Em relação aos tempos de computação salientam-se, de seguida, algumas das conclusões mais relevantes.

- Enquanto que o método de penalidade, que considera inativos os sensores inválidos, não exige um grande esforço computacional uma vez que apenas deixa de ter em consideração para o cálculo da cobertura os sensores localizados numa região não admissível, o método de reparação requer um cálculo mais exigente quando se determina o ponto de interseção com a zona de terra para o qual o sensor deverá ser deslocado. Porém, o gráfico da figura 5.9 permite concluir que nos primeiros 35 subproblemas o tempo de CPU é muito semelhante para os dois métodos utilizados. Existem, principalmente, dois motivos para não se evidenciar o aumento de tempo de computação que seria de esperar ao utilizar o método de reparação:

1. Recorrendo às capacidades do MATLAB, a reparação de um sensor inválido é feita de forma muito eficiente através de cálculos matriciais. O tempo de CPU para realizar a reparação de um sensor é, em geral, inferior a 10^{-4} segundos.
2. A reparação de sensores localizados numa região não admissível leva à cobertura de uma maior área que não precisa de ser analisada posteriormente.

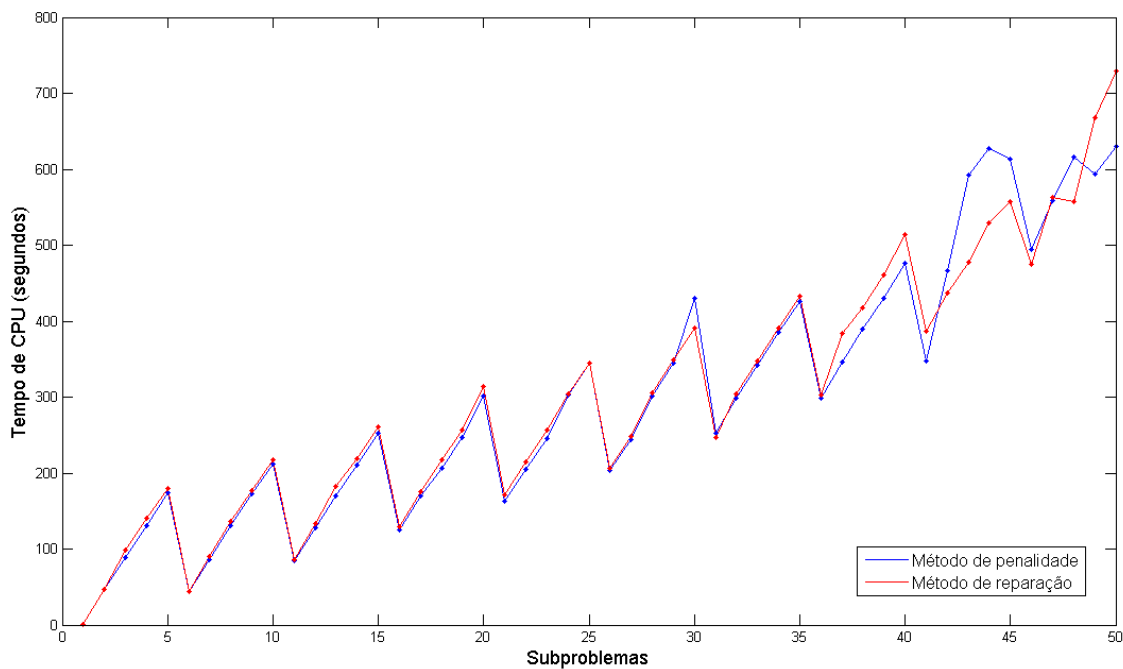


Figura 5.9: Análise comparativa dos tempos de CPU no estudo da admissibilidade.

- Para os últimos subproblemas, a existência de um maior número de sensores faz com que o comportamento do gráfico se altere, tornando-se imprevisível qual dos métodos obtém os resultados em menor tempo de CPU. Parece não existir uma relação evidente entre os tempos de computação obtidos por cada um dos métodos e os subproblemas em estudo.

Os resultados obtidos devem-se, provavelmente, às soluções que vão sendo obtidas ao longo das gerações. O método de reparação consegue, nalguns casos, fazer uso das duas características anteriormente mencionadas para reduzir o tempo de CPU em relação ao método de penalidade. Noutros casos, as soluções que se vão obtendo poderão não ter sensores inválidos suficientes para compensar a reparação das soluções, tornando-se mais eficiente o método de penalidade.

À semelhança do que acontece com os resultados obtidos para a proporção de área por cobrir, os tempos de CPU não são muito conclusivos para decidir qual o melhor método a adotar. A obtenção de melhores resultados por parte de um dos métodos varia de subproblema para subproblema mediante as considerações feitas anteriormente.

Os resultados obtidos para cada subproblema através de cada um dos métodos em estudo podem ser consultados no anexo B.1 e os respetivos tempos de CPU encontram-se, em detalhe, no anexo B.2.

5.1.3 Linha de visão

Ao introduzir obstáculos e considerando que os mesmos são suficientemente elevados para quebrar a linha de visão entre determinados pontos e o centro de um sensor, torna-se essencial incluir este aspeto na abordagem do problema.

A análise do comportamento do algoritmo ao considerar que os obstáculos podem obstruir a linha de visão vai ser feita conjuntamente com a aplicação dos métodos anteriormente estudados que garantem a admissibilidade das soluções. É essencial garantir a obtenção de soluções admissíveis sempre que a área de interesse inclui obstáculos, independentemente de se aplicar ou não o estudo da linha de visão.

Nos testes realizados foi utilizada a área de interesse anteriormente apresentada na figura 5.4 o que permite a comparação dos resultados que serão obtidos nesta seção com os resultados anteriormente descritos na seção 5.1.2 em que apenas se estuda a admissibilidade das soluções.

Os 50 subproblemas foram executados 5 vezes, sendo retido o melhor valor obtido para cada um deles. Tal como se procedeu anteriormente, o gráfico da figura 5.10 apresenta a diferença entre o melhor valor obtido para a proporção total de área por cobrir através do método de penalidade, V_P , e o melhor valor obtido através do método de reparação, V_R . Para cada subproblema, apresenta-se o valor de $V_P - V_R$ considerando a análise da linha de visão e estudando apenas a admissibilidade das soluções (valores anteriormente apresentados no gráfico da figura 5.5).

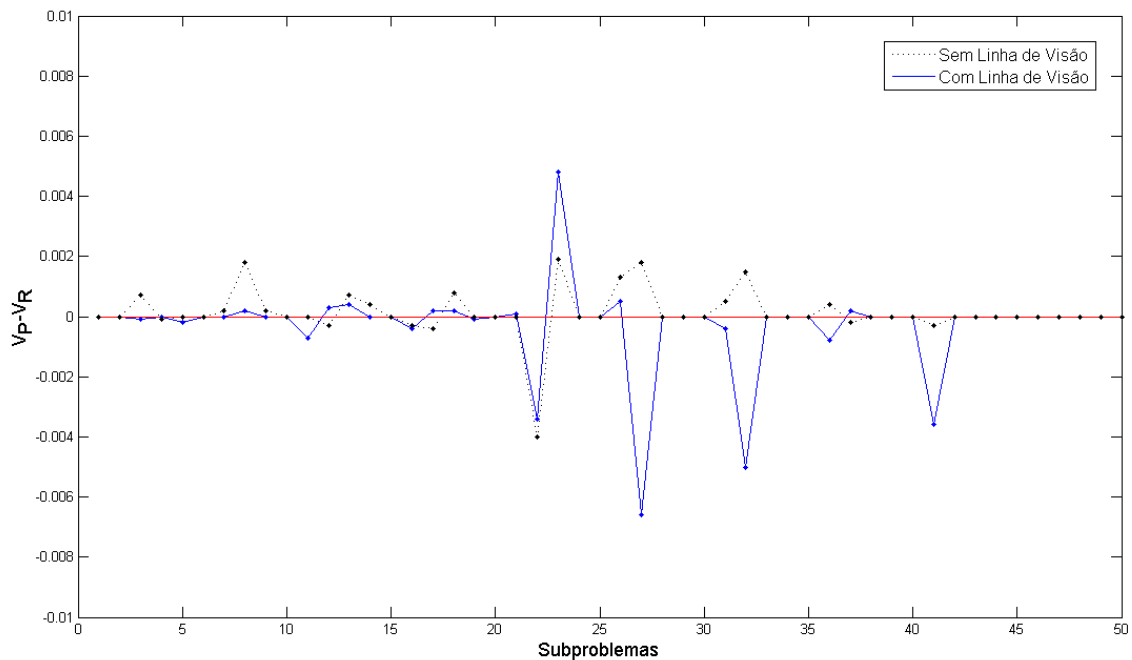


Figura 5.10: Influência do estudo da linha de visão na análise comparativa dos métodos estudados para a admissibilidade.

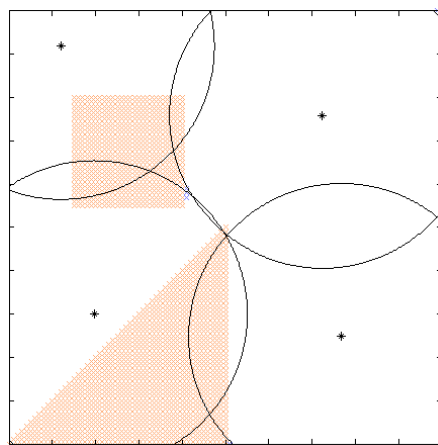
Apresentam-se, de seguida, alguns dos principais aspetos que o gráfico da figura 5.10 permite salientar.

- Ao incluir a análise da linha de visão, para 9 dos 50 subproblemas o método de penalidade obteve melhores resultados. Por outro lado, o método de reparação foi mais eficaz em 11 dos 50 subproblemas e, para os restantes casos, o valor obtido pelos dois métodos foi precisamente o mesmo.
- Para os subproblemas em que os dois métodos obtêm resultados diferentes verifica-se, em geral, uma diferença mais significativa no valor de $|V_P - V_R|$ quando se inclui a análise da linha de visão.
- Verifica-se que, em alguns subproblemas, o método utilizado para obter o melhor resultado ao incluir a análise da linha de visão não coincide com o método que obtém o melhor resultado quando apenas se analisa a admissibilidade das soluções. No seguimento das conclusões feitas na seção 5.1.2, parece não existir evidência para preferir um dos métodos uma vez que os mesmos têm comportamentos diferentes para cada subproblema em particular. Tratando-se de uma heurística, o algoritmo proposto obtém melhores resultados através de um ou de outro método mediante as soluções que vão surgindo ao longo do processo evolutivo.

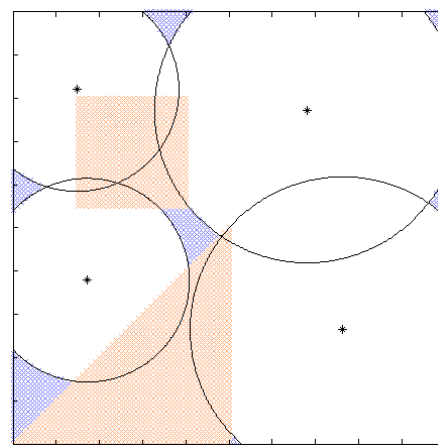
A análise da linha de visão corresponde a uma restrição adicional no cálculo da cobertura de um sensor uma vez que a existência de visibilidade é uma condição necessária

para um ponto se encontrar coberto. Assim, caso se partisse da solução obtida, para cada subproblema, no estudo da admissibilidade das soluções, poderia obter-se uma solução admissível para o problema que inclui a análise da linha de visão verificando quais os pontos que deixam de se encontrar cobertos. Neste caso, às soluções obtidas estaria, necessariamente, associada uma maior ou igual proporção total de área por cobrir. No entanto, uma vez que os resultados apresentados correspondem a uma nova execução da heurística aplicando a análise da linha de visão poderão, eventualmente, surgir novas soluções ao longo das gerações que permitam a obtenção de soluções finais com uma menor proporção total de área por cobrir quando comparadas com as respectivas soluções obtidas quando não se tem em consideração a linha de visão.

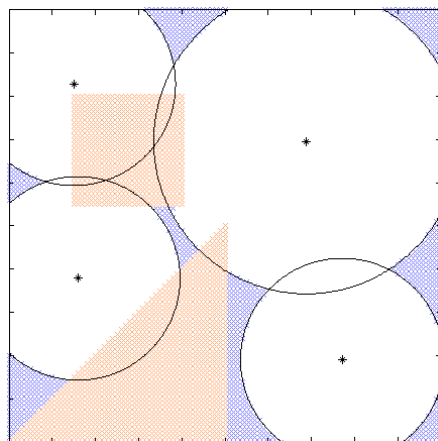
Mediante os resultados obtidos através do método de penalidade e do método de reparação para garantir a admissibilidade das soluções, na figura 5.11 apresentam-se as melhores soluções obtidas para alguns dos subproblemas quando se inclui a análise da linha de visão.



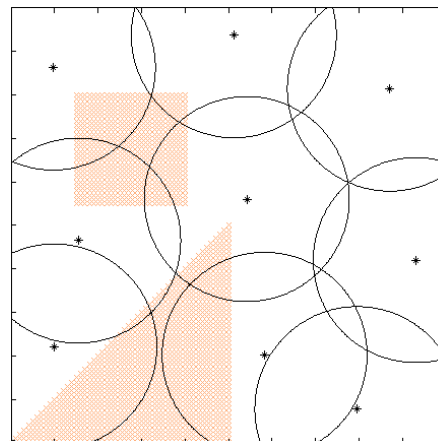
(a) 4 sensores do tipo I



(b) 2 sensores tipo I, 2 sensores tipo II



(c) 1 sensor tipo I, 3 sensores tipo II



(d) 9 sensores do tipo II

Figura 5.11: Soluções obtidas pelo BRKGA incluindo a análise da linha de visão.

As soluções apresentadas na figura 5.11 mostram que o algoritmo consegue adaptar-se de forma bastante satisfatória às restrições impostas na análise da linha de visão, localizando os sensores de forma a cobrir uma grande proporção da área marítima a proteger.

Em relação ao tempo de CPU foi feita a média das execuções realizadas tal como se procedeu nos testes anteriores e os resultados obtidos podem ser observados no gráfico da figura 5.12. Conclui-se que os tempos de CPU são significativamente superiores em cada subproblema quando se inclui a análise da linha de visão o que se deve a um maior esforço computacional que é exigido para verificar quais os pontos que, para além de se encontrarem dentro do alcance de um sensor, não têm a respetiva linha de visão obstruída.

Para além do aspeto anteriormente mencionado e, ao contrário do que acontece quando apenas se estuda a admissibilidade das soluções em que não existe evidência para decidir qual dos métodos é mais eficiente, ao incluir a análise da linha de visão, o método de penalidade obteve, em todos os subproblemas, os resultados num menor tempo de CPU. Assim, este método mostra-se mais eficiente e, uma vez que não é conclusivo qual dos métodos obtém melhores resultados para a proporção de área por cobrir, o método de penalidade deverá ser adotado para obter a solução de qualquer problema de localização de sensores. O programa desenvolvido deve ser independente das características do problema que se pretende estudar e, portanto, os resultados obtidos para os subproblemas em estudo pretendem apenas elucidar para conclusões mais gerais. Apesar de não ser tão eficiente, uma alternativa será executar os dois métodos e escolher a melhor das duas soluções.

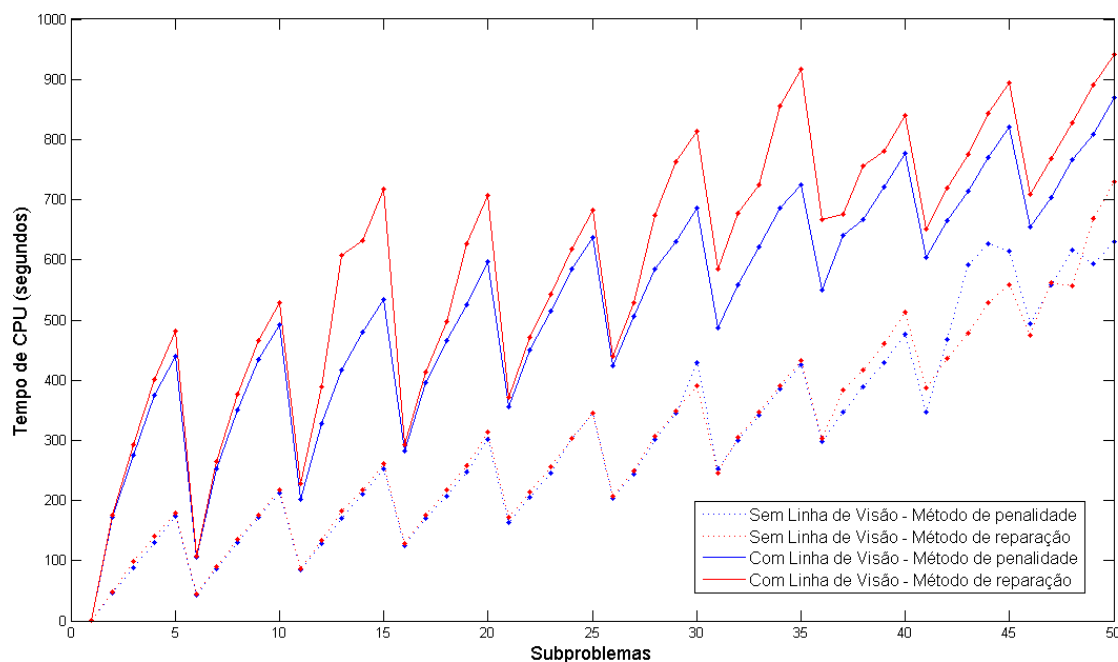


Figura 5.12: Influência do estudo da linha de visão nos tempos de CPU .

Os resultados obtidos para a proporção total de área por cobrir e para os tempos de CPU encontram-se, em detalhe, respetivamente no anexo B.1 e B.2.

5.1.4 Análise do comportamento dos *sensores_{SC}*

A análise feita anteriormente considera apenas a utilização de sensores cuja área de cobertura é circular (*sensores_{SC}*). Porém, a abordagem apresentada nesta dissertação para o problema de localização de sensores, considera que, para além deste tipo de meios de vigilância, existem também sensores cuja área de cobertura corresponde a um setor circular (*sensores_{SC}*). Não existindo resultados provenientes da aplicação de outros algoritmos para comparar o comportamento da heurística adotada quando se pretendem instalar *sensores_{SC}*, esta seção apresenta alguns exemplos de execução do *biased random-key genetic algorithm* considerando a utilização destes sensores em cada um dos três casos em estudo.

Para os exemplos que se apresentam nesta seção, serão considerados *sensores_{SC}* do tipo III com as seguintes características:

- **Tipo III**

Área de cobertura: Setor circular

Ângulo de visão: $a_{III} = \frac{\pi}{2}$ radianos

Raio: $r_{III} = \frac{\sqrt{2}}{6}$

As figuras 5.13 e 5.14 apresentam dois exemplos ilustrativos do comportamento dos *sensores_{SC}*. Para cada caso em estudo, são apresentadas as melhores soluções obtidas em 5 execuções do algoritmo utilizando, respetivamente, 4 sensores do tipo III e 8 sensores do mesmo tipo. Em cada solução apresenta-se a proporção total de área por cobrir, v , e o tempo médio das 5 execuções, \bar{t} , em minutos. Para a análise da admissibilidade e da linha de visão foram utilizados os dois métodos em estudo: método de penalidade e método de reparação, referindo-se em cada solução qual o método que a permitiu obter.

Apesar da análise feita nesta seção não ser exaustiva e as figuras 5.13 e 5.14 pretenderem apenas ilustrar a forma como o *biased random-key genetic algorithm* se comporta perante a introdução de *sensores_{SC}* nos vários casos em estudo para os exemplos concretos referidos, existem algumas considerações que podem ser feitas que se apresentam de seguida.

- Em primeiro lugar salienta-se a importância da orientação dos sensores não se encontrar pré-definida, adaptando-se às características da área de interesse de forma a serem obtidas soluções de melhor qualidade. Esta característica do algoritmo para os *sensores_{SC}* é especialmente importante para as áreas de interesse que contêm obstáculos, tornando-se ainda mais evidente quando se pretendem localizar sensores deste tipo em maior quantidade. No exemplo que considera 8 sensores do tipo III é visível uma boa adaptação dos sensores à área de interesse nos dois casos em que se incluem obstáculos. Os sensores são orientados de forma a que, em conjunto,

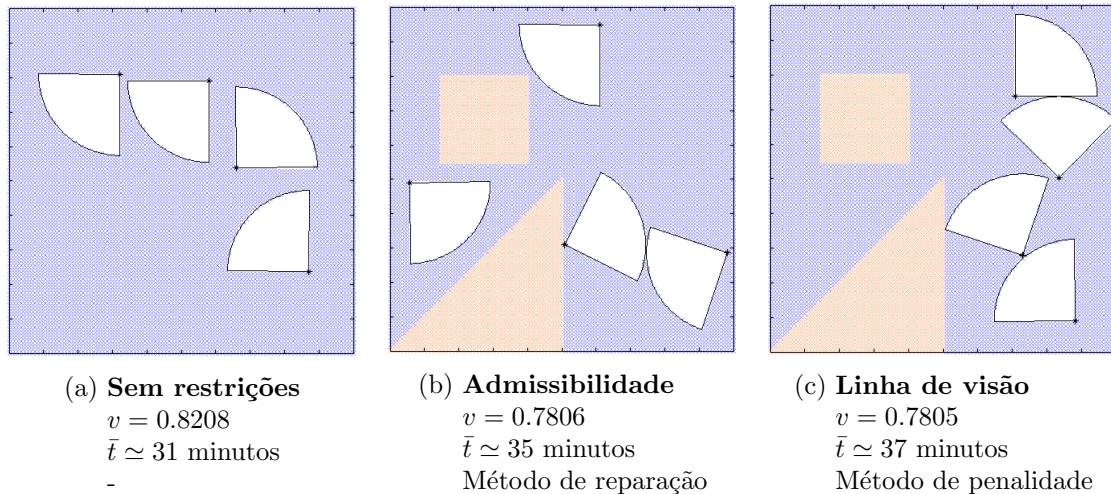


Figura 5.13: Localização de 4 sensores do tipo III.

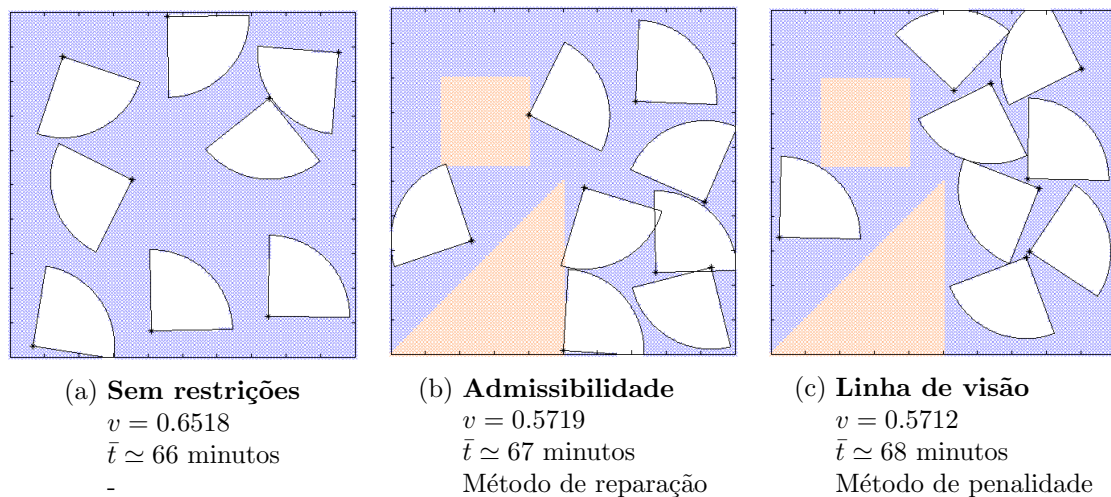


Figura 5.14: Localização de 8 sensores do tipo III.

consigam detetar ameaças numa grande parte da área de interesse. Apesar de poderem existir sobreposições nas áreas de cobertura dos sensores, os dois exemplos apresentados mostram que o algoritmo consegue aproveitar ao máximo a área de cobertura dos mesmos evitando, em geral, sobreposições.

- A introdução da análise da linha de visão faz com que seja tida em consideração uma restrição adicional pelo que as soluções obtidas deverão ser não melhores que as soluções que se obtêm quando apenas se considera a admissibilidade. Porém, em ambos os exemplos apresentados, existe uma ligeira melhoria no valor de v obtido quando se considera a linha de visão. Esta situação ocorre uma vez que ao incluir a análise da linha de visão continua a ser possível localizar os sensores na zona marítima sem existir interferência dos obstáculos e, tratando-se de uma heurística, determinadas soluções que surgiram ao longo do algoritmo deram origem a soluções

ligeiramente melhoradas.

- Uma vez que os sensores do tipo III têm um ângulo de visão de $\frac{\pi}{2}$ radianos e um raio de $\frac{\sqrt{2}}{6}$, 4 sensores deste tipo correspondem à mesma área de cobertura de um sensor circular do tipo II e 8 sensores do tipo III têm uma área de cobertura equivalente a 2 sensores do tipo II. Os valores obtidos para a proporção total de área por cobrir não podem ser comparados uma vez que, ao dividir em setores circulares a área de cobertura de um sensor circular do tipo II, passa a existir uma maior facilidade em localizar os *sensores_{SC}* pois a menor área de cobertura que lhes está associada é mais facilmente adaptada às características da área de interesse.
- Em relação aos tempos de computação, há um aumento bastante significativo na obtenção das soluções quando se introduzem *sensores_{SC}*. A tabela 5.3 apresenta a comparação dos tempos médios de CPU utilizando sensores com área de cobertura circular e *sensores_{SC}* que, em conjunto, têm a mesma área total de cobertura.

O grande aumento no tempo de CPU com a introdução dos *sensores_{SC}* deve-se ao facto de existir um critério adicional para verificar se um ponto da área de interesse se encontra ou não dentro da área de cobertura de um sensor deste tipo. Enquanto que para um sensor circular s de centro $centro_s$ e raio r_s basta verificar a condição $\|u - centro_s\| \leq r_s$, sendo u o ponto que se pretende analisar, para um *sensor_{SC}* é também necessário verificar se o ângulo de visão engloba esse ponto mediante a orientação do sensor. Poderão existir formas mais eficientes de implementar esta verificação pelo que, de futuro, este deve ser um dos aspetos a melhorar de forma a obter as soluções num menor intervalo de tempo. Esta situação não influencia a eficácia do algoritmo desenvolvido.

- Tal como acontecia com a localização de *sensores_C*, com a instalação de *sensores_{SC}* existe um aumento no tempo de CPU com a introdução de obstáculos e com a análise da linha de visão. Neste caso, este aumento não é tão significativo tendo em conta que os tempos de CPU são mais elevados pelas razões descritas no tópico anterior.

A introdução deste tipo de sensores na abordagem feita nesta dissertação não pretende comparar a sua utilização com a utilização de sensores circulares. Ambos os tipos de sensores, variando na forma da área que conseguem cobrir, podem ser localizados na área de interesse, apresentando vantagens e desvantagens. A escolha de quais e quantos os

	Sem restrições	Admissibilidade	Linha de visão
1 sensor do tipo II	0.1 min	0.7 min	1.8 min
4 sensores do tipo III	31 min	35 min	37 min
2 sensores do tipo II	0.2 min	1.4 min	3.4 min
8 sensores do tipo III	66 min	67 min	68 min

Tabela 5.3: Análise comparativa do tempo de CPU para os vários tipos de sensores.

sensores de cada tipo que devem ser instalados depende, para além da disponibilidade dos recursos, de vários fatores nomeadamente das características da área de interesse que poderão ser mais propícias à instalação de um determinado tipo. Dependendo do custo que lhes está associado a solução apresentada pode ser mais ou menos variada no que diz respeito aos tipos de sensores utilizados. Estes e outros aspetos serão referidos com mais detalhe na seção seguinte em que se discute a aplicação do *biased random-key genetic algorithm* ao problema biobjetivo.

5.2 Problema biobjetivo

O problema mono-objetivo anteriormente analisado e discutido motivou o estudo do problema biobjetivo que, para além de se aproximar mais da realidade uma vez que tem, também, em consideração os custos que estão associados à localização dos meios de vigilância, pretende ser uma ferramenta bastante útil como Sistema de Apoio à Decisão para o problema de localização de sensores. Uma vez que o número e tipo de sensores não é fixo tal como acontecia no problema mono-objetivo existe uma maior proximidade com as características de um problema real de localização contínua. É de esperar que o decisor não tenha um interesse particular em definir, à partida, os recursos que pretende localizar mas sim, que possa tomar uma decisão mediante várias alternativas que vão de encontro às suas pretensões. Assim, apenas se conhecem as limitações no número de sensores de cada tipo, podendo existir, também, um número mínimo de sensores com determinadas características que se pretendem utilizar na solução.

Tratando-se de um Sistema de Apoio à Decisão, a obtenção de uma solução final para o problema em estudo deve ser reflexo de uma participação ativa por parte do agente de decisão. Ao executar o programa com os dados pretendidos será apresentado ao decisor um conjunto de soluções potencialmente não dominadas bem espalhadas no espaço dos objetivos de forma a que o decisor possa ter variedade de escolha. Apesar de qualquer uma dessas soluções poder ser escolhida, encontrando-se num ficheiro anexo a descrição pormenorizada de cada uma das soluções potencialmente não dominadas, o programa apresenta ao decisor a representação gráfica de 6 dessas soluções que pretendem ser variadas no que diz respeito à configuração que apresentam. Caso o decisor pretenda incluir uma restrição de risco e custo máximo nas soluções que lhe são apresentadas poderá indicar um valor $Risco_{MAX}$ e/ou $Custo_{MAX}$ tal que para cada solução S apresentada se verifique respetivamente a condição 5.1 e 5.2.

$$Risco(S) \leq Risco_{MAX} \quad (5.1)$$

$$Custo(S) \leq Custo_{MAX} \quad (5.2)$$

As restrições 5.1 e 5.2 tornam a pesquisa de soluções potencialmente não dominadas mais específica no espaço dos objetivos. Apesar de se continuarem a procurar soluções bem espalhadas, a inclusão das condições anteriores reduz o espaço dos objetivos e, por esse

motivo, o decisor é confrontado com mais soluções com as características que pretende. Se a pesquisa de soluções for feita na totalidade do espaço dos objetivos, a probabilidade de se encontrarem soluções variadas com determinadas características torna-se mais reduzida.

5.2.1 Otimização dos parâmetros

À semelhança do que foi referido para o problema mono-objetivo, a escolha dos parâmetros a utilizar deve ser feita cuidadosamente de forma a obter melhores resultados. Nesta dissertação, os parâmetros não foram especificamente otimizados para o problema biobjetivo, mantendo-se ou adaptando-se os resultados obtidos no problema mono-objetivo para os parâmetros comuns aos dois problemas. As considerações feitas na seção 5.1.1 continuam, assim, válidas salvo referência em contrário.

- **Precisão da grelha de pontos para a avaliação da cobertura - δ**

Assumindo um compromisso entre a precisão das soluções obtidas e o tempo de CPU, o valor adotado para δ nos testes realizados foi de $\delta = 0.01$.

- **Parâmetros do *biased random-key genetic algorithm***

São utilizados os mesmos parâmetros para o BRKGA: $p_e = 0.10$, $p_m = 0.30$ e $\rho_e = 0.70$.

- **Dimensão da população**

A dimensão da população, p , utilizada para o problema mono-objetivo é de 50 indivíduos. Porém, uma vez que o problema biobjetivo considera que o número de sensores não é fixo, variando dentro das limitações estabelecidas, não faz sentido continuar a considerar o mesmo tamanho para a dimensão da população. O algoritmo necessita, neste caso, de uma população com uma dimensão superior de forma a poder obter soluções diversificadas para cada combinação possível no número de parâmetros. Assim, de forma a adaptar a otimização feita para o problema mono-objetivo, a dimensão da população deveria ser um múltiplo de 50 consoante o número de combinações possíveis entre os vários tipos de sensores, correspondendo cada combinação a uma configuração fixa. Porém, mesmo para exemplos com um menor número de sensores, a dimensão da população seria demasiado elevada, tornando o algoritmo pouco eficiente a nível computacional.

Tendo em conta as considerações feitas no parágrafo anterior e conhecendo as limitações no número de unidades a instalar de cada sensor s da solução S , ou seja, $\underline{\lim}_s \leq n_s \leq \overline{\lim}_s$, apresenta-se na expressão 5.3 a dimensão da população adotada.

$$p = 50 \times \sum_{s \in S} \left(\overline{\lim}_s - \underline{\lim}_s + 1 \right) \quad (5.3)$$

A dimensão da população é, assim, menor do que substituindo o somatório pelo produto das parcelas, procurando um equilíbrio entre o tempo de computação e a variedade de soluções obtidas ao longo do processo evolutivo.

- **Critério de Paragem**

Uma vez que a dimensão da população foi adaptada para o problema biobjetivo de acordo com a otimização feita para o problema mono-objetivo, o critério de paragem mantém-se, executando-se o algoritmo ao longo de $nG = 500$ gerações.

Para além dos parâmetros anteriormente mencionados, comuns ao problema mono-objetivo, é necessário definir o valor de δ^* que representa a precisão da grelha ajustada ao espaço dos objetivos. O valor de δ^* deve refletir um compromisso entre a obtenção de um leque vasto de soluções potencialmente não dominadas para que o decisor possa tomar uma decisão e o tempo de CPU necessário para obter a fronteira potencialmente de Pareto. Além disso, deve-se ter em conta que, apesar de se pretender mostrar ao decisor uma variedade razoável de soluções, o mesmo não deve ser confrontado com demasiadas soluções que tornem o processo de análise e seleção exaustivo. Para efeitos da presente dissertação, considera-se o valor de $\delta^* = 0.05$ como um valor equilibrado nos aspetos referidos.

Para garantir a admissibilidade das soluções irá utilizar-se o **método de reparação**. Ao introduzir um novo objetivo no problema de localização de sensores, as soluções passam a ser analisadas segundo os dois critérios: minimização da área por cobrir e minimização do custo. Uma vez que se obtém um conjunto de soluções potencialmente não dominadas que são apresentadas ao agente de decisão, o método de penalidade não consegue garantir que todas as soluções apresentadas são admissíveis, apesar de conseguir estabilizar o número de soluções que não respeitam as restrições de localização. Para evitar que o decisor tenha contacto com soluções não admissíveis este método não é adotado no estudo do problema biobjetivo.

5.2.2 Exemplo

O exemplo para o problema biobjetivo que se apresenta de seguida considera a mesma área de interesse utilizada no problema mono-objetivo (figura 5.4) e considera a análise da admissibilidade e da linha de visão. Este é precisamente o problema completo que tem interesse estudar. A análise detalhada feita para o problema mono-objetivo para os três casos em estudo pretende mostrar a forma como a inclusão de novas restrições afeta a eficácia e eficiência da heurística adotada, esperando-se um comportamento semelhante para o problema biobjetivo.

No exemplo que se apresenta considera-se a existência dos três tipos de sensores anteriormente referidos com as seguintes características:

- **Tipo I**

Sensor circular de raio $r_I = \frac{\sqrt{2}}{4}$.

Número de sensores a instalar: $0 \leq n_I \leq 4$.

Custo fixo de instalação: $custo_I = \frac{\pi}{8}$.

- **Tipo II**

Sensor circular de raio $r_{II} = \frac{\sqrt{2}}{6}$.

Número de sensores que podem ser instalados: $0 \leq n_{II} \leq 9$.

Custo fixo de instalação: $custo_{II} = \frac{\pi}{18}$.

- **Tipo III**

Sensor setor circular de raio $r_{III} = \frac{\sqrt{2}}{6}$ e ângulo de visão: $a_{III} = \frac{\pi}{2}$ radianos.

Número de sensores que podem ser instalados: $2 \leq n_{III} \leq 4$.

Custo fixo de instalação: $custo_{III} = \frac{\pi}{72}$.

Recorde-se que o custo total da solução tem em conta não só o custo fixo de cada sensor mas também o custo variável consoante a distância ao gerador de energia que se considera localizado no ponto de coordenadas (0,0). Apesar de se poder alterar facilmente a fórmula que calcula o custo de uma solução consoante as características do problema em estudo, nesta dissertação utiliza-se, sem perda de generalidade, a fórmula apresentada na equação 2.4 da seção 2.3. O custo fixo que se considera para cada sensor corresponde à respetiva área de cobertura.

Em relação às restrições no número de sensores que podem ser utilizados considera-se que a disponibilidade de recursos é de, respetivamente, 4, 9 e 4 para cada um dos três tipos. Exige-se que pelo menos duas unidades de sensores do tipo III sejam incluídas na solução do problema.

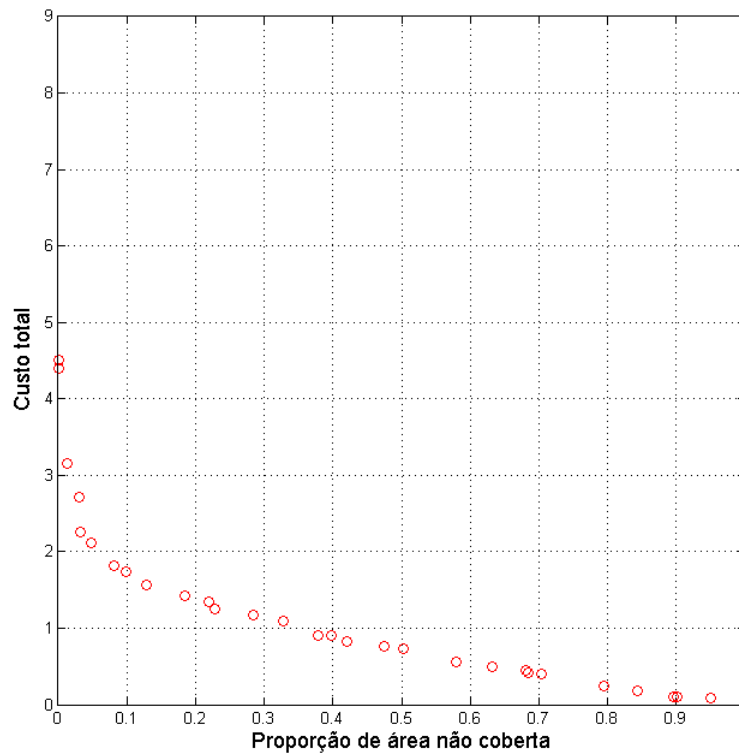


Figura 5.15: Fronteira potencialmente de Pareto obtida para o exemplo em estudo.

Na figura 5.15 apresenta-se a fronteira potencialmente de Pareto obtida para o problema biobjetivo com as características anteriormente referidas. Não existindo restrições adicionais para o valor máximo de risco e custo que o decisor está disposto a aceitar, a pesquisa de soluções potencialmente não dominadas é feita em todo o espaço dos objetivos. O valor de risco máximo é 1 e, neste caso, nunca é atingido uma vez que é obrigatório a instalação de dois sensores do tipo III. Não existindo limitações inferiores no número de sensores, o valor de risco 1 corresponderia à solução de não instalar nenhum sensor. Em relação ao custo, o algoritmo determina o valor máximo de custo que se pode obter consoante os dados considerados. Esta situação corresponde à instalação de todos os sensores disponíveis na posição mais afastada do gerador.

A figura 5.16 apresenta as 6 soluções potencialmente não dominadas que são apresentadas ao decisor para que possa ter uma visualização gráfica do tipo de soluções existentes no espaço dos objetivos. Porém, o decisor pode optar por qualquer solução que pertença à fronteira potencialmente de Pareto, encontrando-se a descrição detalhada de cada solução no anexo C.1. As soluções apresentadas ao decisor pretendem ser variadas no que diz respeito ao número e tipo de sensores utilizado de forma a que o decisor seja confrontado com alternativas diversas que possam ir de encontro às suas expectativas. A partir do momento em que se exige que um número mínimo de sensores de determinado tipo seja instalado, aumenta a probabilidade de se encontrarem soluções como a terceira solução da figura 5.16

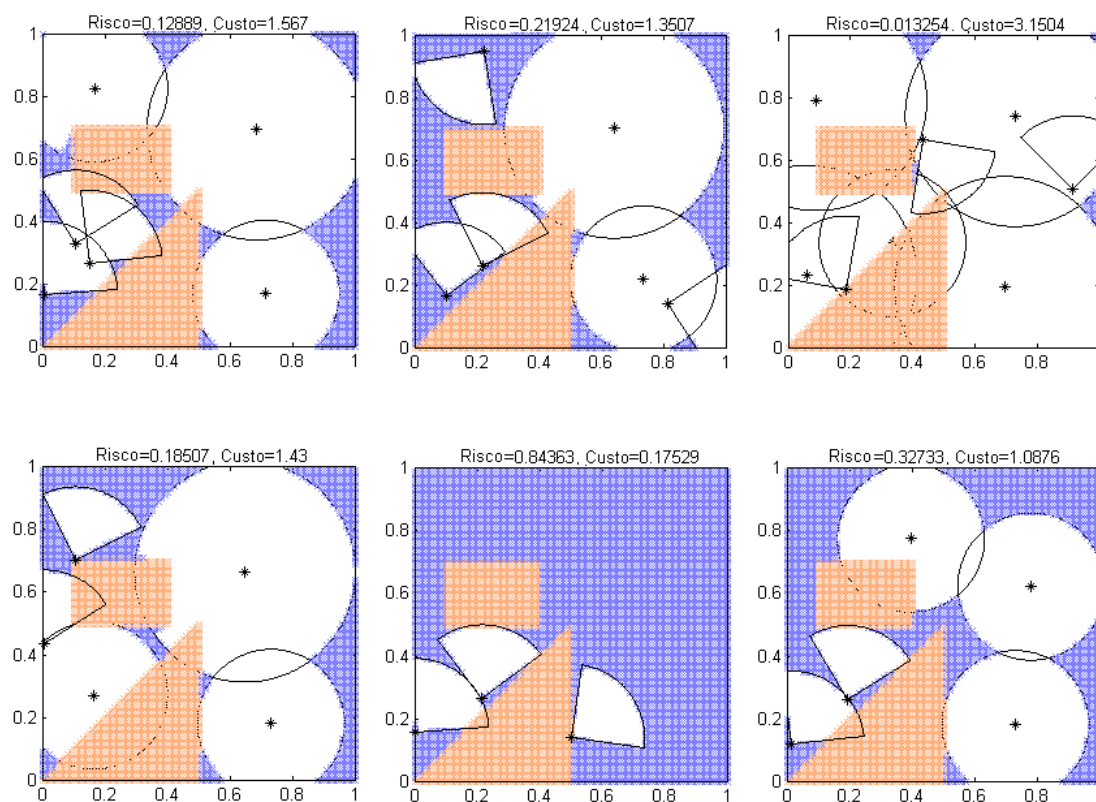


Figura 5.16: Soluções apresentadas ao decisor.

em que a utilização dos sensores de tipo III é completamente redundante pois a área que cobrem já se encontra coberta por outros sensores. Neste caso, o algoritmo deveria deslocar estes sensores para as pequenas áreas que ainda se encontram por cobrir. Este aspeto pode ser melhorado conforme se sugere na seção 6.2 através de um mecanismo de pesquisa local.

Ao analisar a fronteira potencialmente de Pareto, pode existir uma determinada zona de especial interesse para o decisor. Assim, o agente de decisão pode querer refinar a procura de soluções nessa zona definindo o limite máximo para o risco e para o custo que está disposto a aceitar. Como exemplo, supõe-se que o decisor pretende incluir na pesquisa as seguintes restrições:

$$Risco(S) \leq 0.3$$

$$Custo(S) \leq 2$$

O algoritmo é novamente executado incluindo as restrições anteriores, obtendo-se a fronteira potencialmente de Pareto apresentada na figura 5.17. Tal como foi anteriormente referido, ao incidir a pesquisa sobre uma determinada zona do espaço dos objetivos são apresentadas mais soluções com as características pretendidas. Esta situação verifica-se uma vez que, numa análise mais geral, algumas destas soluções não eram consideradas suficientemente distintas em relação aos valores dos objetivos que apresentam.

Mais uma vez são apresentadas 6 soluções ao decisor, conforme ilustra a figura 5.18

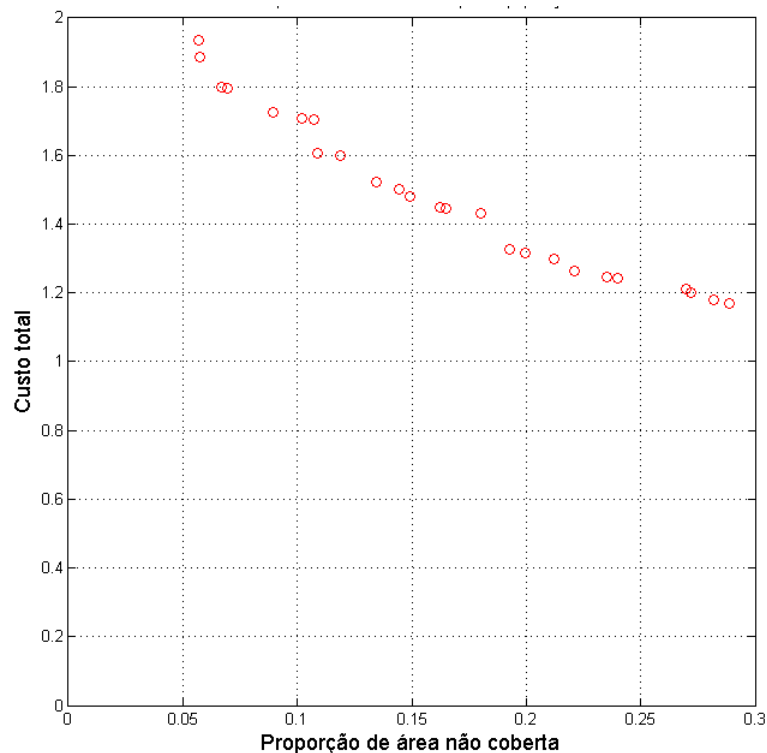


Figura 5.17: Fronteira potencialmente de Pareto obtida reduzindo o espaço dos objetivos.

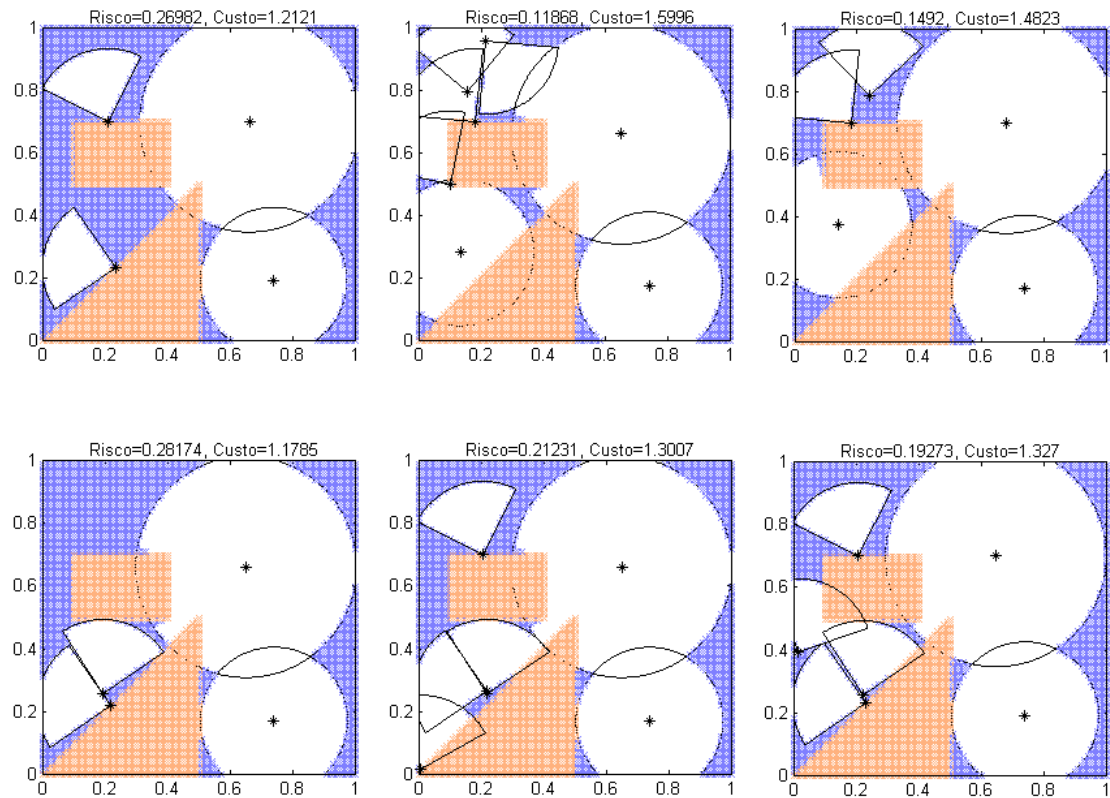


Figura 5.18: Soluções apresentadas ao decisor com a redução do espaço dos objetivos.

que pretendem ser elucidativas do tipo de soluções existentes no espaço dos objetivos analisado. A descrição detalhada de todas as soluções apresentadas na figura 5.17 pode ser encontrada no anexo C.2.

O processo iterativo de escolha da solução para o problema termina quando o agente de decisão encontrar uma solução que satisfaça os seus objetivos. Assim, quando confrontado com um conjunto de soluções potencialmente não dominadas, deve decidir terminar o processo com a escolha de uma delas ou continuar a reduzir o espaço dos objetivos de forma a poder aumentar a variedade de soluções com os valores máximos de risco e custo que está interessado em obter.

Em relação ao tempo de CPU, a fronteira potencialmente de Pareto da figura 5.15 foi obtida em cerca de 531 minutos e a da figura 5.17 em cerca de 526 minutos. A utilização de sensores do tipo III contribui para a obtenção de tempos de CPU bastante mais elevados do que se pretende. Este é um aspeto que deve ser tido em conta conforme se referiu na seção 5.1.4 e que, futuramente, deve ser melhorado de forma a tornar o algoritmo mais eficiente na obtenção de soluções para o problema de localização de sensores.

No anexo D é feita uma breve descrição do modo de funcionamento dos programas utilizados para obter os resultados apresentados na presente seção.

Capítulo 6

Conclusões

Neste capítulo apresentam-se as principais conclusões da aplicação do *biased random-key genetic algorithm* aos três casos em estudo que correspondem a diferentes vertentes do problema de localização de sensores: problema sem restrições, admissibilidade das soluções face a restrições de localização e análise da linha de visão. Apesar da metodologia apresentada ser bastante completa para a resolução do problema, mostrando uma grande adaptabilidade face a diferentes situações, é essencial focar os aspetos que poderão ser melhorados e/ou introduzidos na abordagem que esta dissertação apresenta. Neste sentido, serão referidos, também, os principais desenvolvimentos futuros a ter em consideração.

6.1 Principais conclusões

A heurística proposta como metodologia de resolução é um algoritmo genético que codifica as soluções através de chaves aleatórias e mostrou ser uma abordagem bastante eficaz para o problema de localização de sensores. Esta forma de codificar as soluções é uma das principais vantagens do algoritmo proposto, permitindo uma grande adaptabilidade face a diferentes situações conforme se verificou na análise dos diferentes casos de estudo.

Apesar da motivação central da metodologia apresentada ser a proteção de portos, as considerações feitas são facilmente generalizadas para outras áreas de aplicação. Como exemplo, pode-se referir a proteção de florestas face a incêndios. Esta é uma situação bastante comum no nosso país e, apostando na monitorização, podem ser prevenidos muitos danos económicos, humanos e sociais. Este exemplo ilustra a importância da minimização do *risco* por si só: muitas vezes, mesmo que a instalação de meios de vigilância acarrete custos elevados, estes custos poderão ser compensados se forem evitados os danos anteriormente referidos. Por este motivo, começou por se estudar o problema mono-objetivo e, posteriormente, motivou-se o estudo do problema biobjetivo que pretende ser uma ferramenta útil para integrar, de futuro, um Sistema de Apoio à Decisão no âmbito do projeto SAFEPORT. A metodologia que se considera para o problema biobjetivo, indicando simultaneamente o número e tipo de sensores a utilizar e a respetiva localização, permitiu obter um conjunto de soluções potencialmente não dominadas bem espalhadas no espaço

dos objetivos, deixando ao critério do decisor a escolha, entre elas, da solução a implementar.

De acordo com os testes realizados para o problema mono-objetivo pode-se concluir que o *biased random-key genetic algorithm* apresenta bons resultados para o problema em estudo. O problema sem restrições foi estudado em primeiro lugar e, posteriormente, com a introdução de obstáculos na área de interesse incluiu-se a análise da admissibilidade das soluções e da linha de visão.

Comparando os valores obtidos para o problema sem restrições com os valores de referência dos 50 subproblemas estudados conclui-se que o algoritmo consegue obter resultados com um erro associado muito baixo, o que significa que a heurística adotada é eficaz na obtenção das soluções para o problema de localização de sensores. Salienta-se o facto dos parâmetros do algoritmo terem sido cuidadosamente otimizados, nomeadamente a dimensão da população e o critério de paragem que, otimizados conjuntamente, permitem a geração de soluções variadas ao longo do processo evolutivo, levando a uma solução final de boa qualidade num tempo razoável de computação.

Com a introdução de obstáculos deixam de ser conhecidos os valores de referência que permitem avaliar a qualidade dos resultados obtidos. Porém, foram obtidas proporções de área por cobrir bastante reduzidas concluindo-se que a heurística continua a apresentar um bom comportamento. Para o estudo da admissibilidade das soluções foi aplicado o método de penalidade e o método de reparação. Os dois métodos apresentam um comportamento semelhante não existindo razões para preferir um deles. Porém, ao introduzir a análise da linha de visão, apesar das proporções totais de área por cobrir obtidas pelos dois métodos serem bastante semelhantes, o método de penalidade é mais eficiente, obtendo os resultados em menor tempo de computação.

A heurística apresentada adapta-se facilmente ao problema biobjetivo que considera, para além da minimização do risco, a minimização do custo, permitindo ao decisor escolher a solução que melhor se enquadra nos seus objetivos. O agente de decisão é confrontado com um conjunto de soluções potencialmente não dominadas bem espalhadas no espaço dos objetivos, podendo restringir a pesquisa a um valor máximo de risco e/ou custo que está disposto a aceitar.

6.2 Desenvolvimentos futuros

Ao longo da dissertação foram referidas algumas simplificações que foram adotadas de forma a facilitar a análise do problema em estudo. Porém, para que o procedimento desenvolvido possa ser aplicado a problemas concretos nomeadamente no âmbito do projeto SAFEPOR para a proteção de portos marítimos, muitas dessas simplificações têm de se tornar mais realistas.

É de esperar que o comportamento das funções de risco e custo adotadas não seja assim tão simples. Em relação à avaliação do risco, tal como foi referido na seção 2.2.2,

o modelo de cobertura probabilístico representa de forma mais realista a capacidade de detecção de um sensor e deverá ser utilizado em vez do modelo de cobertura binária. Assim, mediante as características de cada sensor deverá ajustar-se uma função representativa do seu comportamento com a qual deverá ser calculada a cobertura do mesmo em cada ponto da área de interesse. Para a função de custo que se considera no problema biobjetivo, poderão existir outros aspetos a ter em conta para além do custo fixo e da distância ao gerador de energia como, por exemplo, a topografia do terreno: determinados locais poderão exigir um maior esforço de instalação que se reflete no custo da solução. Além disso, considerou-se apenas a existência de um gerador de energia localizado no ponto (0,0) mas poderão existir vários geradores e restrições na associação dos diferentes tipos de sensores aos mesmos.

Neste trabalho, considera-se que todas as zonas da área de interesse têm igual relevância para serem monitorizadas. Desta forma, a definição de risco adotada representa a probabilidade de uma ameaça não ser detetada pelos sensores instalados. Porém, é possível que determinadas zonas da área de interesse possuam características particulares que as tornem especialmente importantes para serem protegidas. Ao longo dos anos têm sido propostas diferentes definições de *risco* tendo, precisamente, em consideração o aspeto anteriormente mencionado. Rodrigues [24] apresenta uma possível definição de risco que engloba três fatores: ameaça, vulnerabilidade e consequência, conforme se apresenta na expressão 6.1.

$$\text{Risco} = \text{Ameaça} \times \text{Vulnerabilidade} \times \text{Consequência} \quad (6.1)$$

Os componentes que constituem o risco podem ser interpretados da seguinte forma:

$$\begin{aligned} \text{Ameaça} &= P(\text{ocorrer um ataque}) \\ \text{Vulnerabilidade} &= P(\text{ataque resultar em danos} \mid \text{ocorrer um ataque}) \\ \text{Consequência} &= E(\text{estragos} \mid \text{ocorrer um ataque e resultar em danos}) \end{aligned}$$

Uma outra questão que foi alvo de simplificações foi a área de interesse. O estudo do problema foi feito a duas dimensões (no quadrado unitário $[0, 1]^2$) mas o problema torna-se mais realista se este estudo considerar uma terceira dimensão associada à profundidade. Desta forma, a detecção de ameaças é feita não só à superfície mas, também, sob o mar onde podem existir mergulhadores que sejam potenciais ameaças. Além disso, em relação aos obstáculos incluídos na área de interesse, os mesmos foram representados através de polígonos, o que não corresponde à realidade. Porém, não se prevê que seja muito difícil adaptar a implementação feita de modo a incluir dados reais da linha de costa.

Note-se que os aspetos anteriormente mencionados para tornar mais realista a abordagem ao problema não interferem com o funcionamento da metodologia de resolução proposta, podendo ser facilmente implementados. Devido à maior complexidade que lhes está associada espera-se um aumento significativo no tempo de computação.

A utilização de sensores cuja área de cobertura corresponde a um setor circular mostrou-se uma mais-valia uma vez que os mesmos permitem uma maior versatilidade comparativamente aos sensores que cobrem uma área correspondente a um círculo. O facto da orientação dos mesmos não se encontrar pré-definida, permite que estes sensores se adaptem, mais facilmente, a zonas da área de interesse que se encontram por cobrir. Um aspeto bastante interessante que poderá ser desenvolvido futuramente para este tipo de sensores tem, precisamente, a ver com a importância da orientação. Neste trabalho, assume-se que os sensores são fixos e que, a partir do momento em que o algoritmo determina a respetiva orientação, o sensor monitoriza sempre a mesma área. Porém, poderá considerar-se que os sensores rodam sobre um eixo fixo. Apesar desta situação não interferir com a área coberta por um sensor circular, vai influenciar o comportamento dos sensores que cobrem uma área correspondente a um setor circular. Esta rotação requer, assim, uma análise no espaço temporal de forma a avaliar a zona coberta em cada intervalo de tempo. Este aspeto torna a abordagem mais complexa mas poderá ser bastante útil na minimização do número de sensores utilizados para monitorizar a área de interesse se os sensores instalados forem corretamente sincronizados.

A desvantagem que estes sensores apresentam relativamente ao maior tempo de computação que introduzem na execução do algoritmo é inevitável uma vez que a função que determina se um ponto se encontra ou não no interior de um setor circular não é tão simples como a expressão que indica quais os pontos que se encontram no interior de um círculo.

Existem algumas considerações sobre o funcionamento do *biased random-key genetic algorithm* que parecem, à partida, contribuir para melhorar o seu funcionamento no que diz respeito à obtenção de soluções de melhor qualidade para o problema de localização de sensores que se estuda nesta dissertação.

Um procedimento comum adotado para esta metodologia é a utilização de múltiplas populações que evoluem de forma independente e que, periodicamente, trocam soluções entre si. Este mecanismo procura acelerar a convergência do algoritmo e pode ser uma boa forma de reduzir o aumento do tempo computacional que se espera quando forem introduzidos alguns dos melhoramentos que se sugerem. É essencial otimizar o número de indivíduos que trocam de população e de quantas em quantas gerações é feita essa passagem. Trocar demasiadas soluções entre as populações perturba o conceito de evolução e trocar constantemente informação faz com que as populações não tenham tempo suficiente de produzir boas novas soluções. As soluções que deverão ser substituídas quando se procede à troca de informação, serão aquelas com pior aptidão no contexto do problema. Apesar desta ferramenta, sugerida por Gonçalves e Resende [12] ser bastante poderosa, é essencial que os parâmetros anteriores sejam rigorosamente otimizados.

Algumas soluções obtidas através da aplicação do *biased random-key genetic algorithm* podem visivelmente ser melhoradas com um pequeno deslocamento de um ou mais sensores, conforme ilustra a figura 6.1. Na figura 6.1a apresenta-se a solução obtida pelo

algoritmo para o problema mono-objetivo no caso mais simples, considerando a localização de um sensor do tipo I e seis sensores do tipo II. Um deslocamento do centro dos sensores do tipo II localizados no canto superior esquerdo na direção definida pelas setas permite a cobertura de uma maior proporção da área de interesse uma vez que as zonas assinaladas passam a estar melhor monitorizadas conforme se observa na figura 6.1b. Apesar da qualidade das soluções obtidas pelo algoritmo ser bastante boa, a partir do momento em que se obtém uma solução muito próxima da solução ótima, o algoritmo não consegue, em muitos casos, refinar a procura de modo a melhorar ainda mais a solução obtida. Aplicando uma fase de pesquisa local à solução obtida através do algoritmo (ou soluções no caso do problema biobjetivo), esta situação poderia ser contornada. Neste caso, o algoritmo podia ser executado num menor número de gerações, procedendo-se depois à pesquisa local de soluções vizinhas da solução encontrada, sob o critério de deslocar na direção de zonas não cobertas um ou mais sensores até que o valor da solução deixe de melhorar. Espera-se que com este procedimento o algoritmo possa obter soluções melhores em menos tempo, evitando o aparecimento de soluções para as quais se percebe facilmente que podem ser melhoradas.

Sendo fundamental a escolha dos parâmetros na obtenção de bons resultados, os parâmetros utilizados no problema biobjetivo deverão ser sujeitos a otimização independente da otimização feita para o problema mono-objetivo. Esta otimização é especialmente importante para determinar o número de indivíduos na população e o número de gerações em que o algoritmo deve ser executado uma vez que o problema biobjetivo apresenta características diferentes do problema mono-objetivo existindo, possivelmente, outros valores para os parâmetros que melhor se adequam ao problema, melhorando a qualidade das soluções obtidas. Com a adaptação dos parâmetros utilizada pretende-se mostrar o funcionamento do algoritmo para o problema com dois objetivos, servindo de base a um Sistema de Apoio à Decisão que deverá ter em conta as considerações feitas nesta seção de

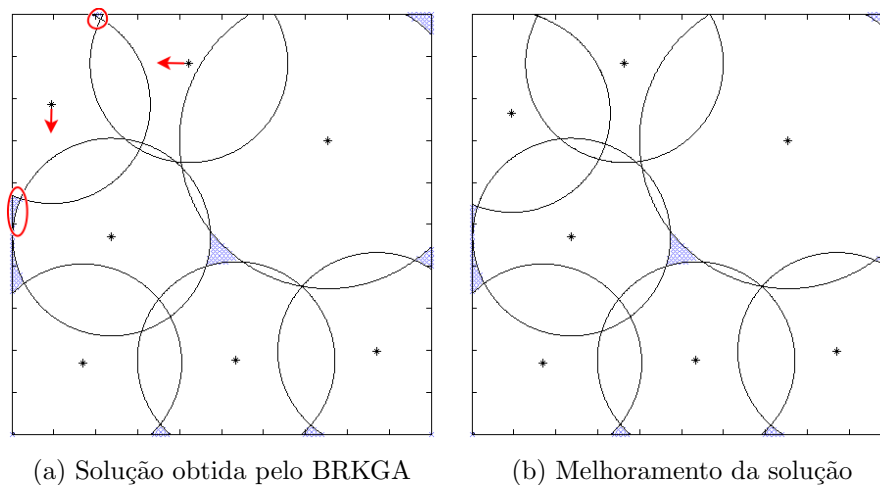


Figura 6.1: Exemplo de aplicação de uma fase de pesquisa local.

forma a poder ser melhorado, aproximando-se mais das necessidades reais que se esperam para a resolução do problema de localização contínua de sensores.

Ao introduzir obstáculos na área de interesse e com o objetivo de garantir a admissibilidade das soluções, não foi conclusivo qual dos métodos obtém melhores resultados. Porém, o método de penalidade mostrou ser mais eficiente, principalmente quando se inclui a análise da linha de visão. Um dos aspetos que poderá ser melhorado de forma a tornar o método de penalidade mais eficaz (e aliando este facto à obtenção de resultados num menor tempo computacional seria o melhor método) é incluir um mecanismo que consiga quantificar o desrespeito pelas restrições de localização. Isto significa que duas soluções com o mesmo número e tipo de sensores localizados na região não admissível, conforme ilustra a figura 4.3 anteriormente apresentada para referir a desvantagem do método de penalidade, deverão ser penalizadas de forma distinta consoante o afastamento do centro dos sensores à zona marítima. Para soluções em que os centros dos sensores localizados numa região não admissível se encontrem perto da fronteira da zona de terra deverá existir uma menor penalização. Desta forma, o algoritmo consegue perceber que uma simples deslocação desses sensores levará a uma solução admissível. Pelo contrário, caso os sensores estejam localizados muito longe da fronteira deverá existir uma maior penalização. Uma possível sugestão é introduzir uma parcela adicional que tenha em consideração as distâncias dos centros dos sensores não admissíveis à fronteira entre a terra e o mar. Esta parcela deverá ser subtraída ao valor da cobertura da solução, já considerando inativos os sensores não admissíveis.

Os desenvolvimentos futuros que foram referidos pretendem ser uma mais-valia para tornar o algoritmo proposto mais consistente e melhor adaptado a características dos problemas reais que terão de ser enfrentadas mais cedo ou mais tarde. Apesar dos melhoramentos que se sugerem, considera-se que o trabalho apresentado nesta dissertação constitui uma base sólida para a resolução de problemas de localização de sensores, em particular no âmbito da proteção de portos.

Referências

- [1] Akbarzadeh, V., Gagné, C., Parizeau, M., Mostafavi, M. A. (2011). Black-box optimization of sensor placement with elevation maps and probabilistic sensing models. In *Robotic and Sensors Environments (ROSE), 2011 IEEE International Symposium on*, pp. 89-94.
- [2] Akbarzadeh, V., Gagné, C., Parizeau, M., Argany, M., Mostafavi, M. A. (2012). Probabilistic Sensing Model for Sensor Placement Optimization Based on Line-of-Sight Coverage. In *Instrumentation and Measurement, IEEE Transactions on*, vol. 62(2), pp. 293-303.
- [3] Aziz, N. A. A., Aziz, K. A., Ismail, W. Z. W. (2009). Coverage strategies for wireless sensor networks. *World Academy of Science, Engineering and Technology*, vol. 50, pp. 145-150.
- [4] Bean, J. C. (1994). Genetic algorithms and random keys for sequencing and optimization. *ORSA journal on computing*, vol. 6(2), pp. 154-160.
- [5] Buriol, L. S., Hirsch, M. J., Pardalos, P. M., Querido, T., Resende, M. G., Ritt, M. (2010). A biased random-key genetic algorithm for road congestion minimization. *Optimization Letters*, vol. 4(4), pp. 619-633.
- [6] Chong, C. Y., Kumar, S. P. (2003). Sensor networks: evolution, opportunities, and challenges. *Proceedings of the IEEE*, vol. 91(8), pp. 1247-1256.
- [7] Chung, H., Polak, E., Royset, J. O., Sastry, S. (2011). On the optimal detection of an underwater intruder in a channel using unmanned underwater vehicles. *Naval Research Logistics (NRL)*, vol. 58(8), pp. 804-820.
- [8] Coello Coello, C. A. (2002). Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. *Computer methods in applied mechanics and engineering*, vol. 191(11), pp. 1245-1287.
- [9] Coello Coello, C. A., Lamont, G. B., Van Veldhuisen, D. A. (2007). *Evolutionary algorithms for solving multi-objective problems*. Springer.
- [10] Fonseca, C. M., Fleming, P. J. (1993). Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In *ICGA*, vol. 93, pp. 416-423.

-
- [11] Gaspar-Cunha, A., Takahashi, R., Antunes, C. (2012). *Manual de Computação Evolutiva e Meta Heurísticas*. Imprensa da Universidade de Coimbra.
- [12] Gonçalves, J. F., Resende, M. G. (2011). Biased random-key genetic algorithms for combinatorial optimization. *Journal of Heuristics*, vol. 17(5), pp. 487-525.
- [13] Gonçalves, J. F., Resende, M. G. (2011). A parallel multi-population genetic algorithm for a constrained two-dimensional orthogonal packing problem. *Journal of combinatorial optimization*, vol. 22(2), pp. 180-201.
- [14] Guo, H. Y., Zhang, L., Zhang, L. L., Zhou, J. X. (2004). Optimal placement of sensors for structural health monitoring using improved genetic algorithms. *Smart Materials and Structures*, vol. 13(3), pp. 528-534.
- [15] Hefeeda, M., Ahmadi, H. (2007). A probabilistic coverage protocol for wireless sensor networks. In *Network Protocols, 2007. ICNP 2007. IEEE International Conference on*, pp. 41-50.
- [16] Kurpati, A., Azarm, S., Wu, J. (2002). Constraint handling improvements for multi-objective genetic algorithms. *Structural and Multidisciplinary Optimization*, vol. 23(3), pp. 204-213.
- [17] Martins, M., Casimiro, R., Gonçalves, G., Sousa, J., Calado, J., Manso, M., Lopes, J., Rodrigues, A., Captivo, M.E., Freitas, J., Abreu, A., Bezzeghoud, M., Salgado, R. (2010). The SAFE-PORT project: An approach to port surveillance and protection. In *Waterside Security Conference (WSS), 2010 International*, pp. 1-9.
- [18] Mendes, J. J. D. M., Gonçalves, J. F., Resende, M. G. (2009). A random key based genetic algorithm for the resource constrained project scheduling problem. *Computers & Operations Research*, vol. 36(1), pp. 92-109.
- [19] Moniz, M. (2012). Modelo de otimização binível e biobjetivo para a escolha e localização de sensores. Dissertação de Mestrado, DEIO, FCUL.
- [20] Narayanan, S., Azarm, S. (1999). On improving multiobjective genetic algorithms for design optimization. *Structural Optimization*, vol. 18(2-3), pp. 146-155.
- [21] Ngatchou, P. N., Fox, W. L., El-Sharkawi, M. A. (2006). Multiobjective multistatic sonar sensor placement. In *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*, pp. 2713-2719.
- [22] Osais, Y., St-Hilaire, M., Yu, F. R. (2008). The minimum cost sensor placement problem for directional wireless sensor networks. In *Vehicular Technology Conference, 2008. VTC 2008-Fall. IEEE 68th*, pp. 1-5.

-
- [23] Roque, L. A., Fontes, D. B., Fontes, F. A. (2011). A biased random key genetic algorithm approach for unit commitment problem. In *Experimental Algorithms* (pp. 327-339). Springer Berlin Heidelberg.
- [24] Rodrigues, A.J. (2012). Minimizing Port Security Risk. In *Proceedings of the NATO RTO SCI-247 Symposium on Port and Regional Maritime Security*, Lerici, Italy (2012)
- [25] Shyam, M., Kumar, A. (2010). Obstacle Constrained Total Area Coverage in Wireless Sensor Networks, *arXiv preprint arXiv:1001.4753*.
- [26] Silva, D. (2013). Heurísticas para a localização de sensores. Dissertação de Mestrado, DEIO, FCUL.
- [27] Silva, R. M. A., Resende, M. G. C., Pardalos, P. M., Gonçalves, J. F. (2012). Biased random-key genetic algorithm for bound-constrained global optimization. In *Proceedings of the Global Optimization Workshop*, pp. 133-136.
- [28] Tan, H., Wang, Y., Hao, X., Hua, Q. S., Lau, F. C. (2010). Arbitrary obstacles constrained full coverage in wireless sensor networks. In *Wireless Algorithms, Systems, and Applications* (pp. 1-10). Springer Berlin Heidelberg.
- [29] Wang, Y. C., Hu, C. C., Tseng, Y. C. (2008). Efficient placement and dispatch of sensors in a wireless sensor network. *Mobile Computing, IEEE Transactions on*, vol. 7(2), pp. 262-274.
- [30] Xiao, J., Michalewicz, Z., Zhang, L. (1996). Evolutionary planner/navigator: Operator performance and self-tuning. In *Evolutionary Computation, 1996., Proceedings of IEEE International Conference on*, pp. 366-371.

Anexo A

Otimização dos parâmetros

A.1 Problema mono-objetivo

A.1.1 Parâmetros do *biased random-key genetic algorithm*

Os parâmetros do *biased random-key genetic algorithm*, mais especificamente, a proporção de indivíduos pertencentes à elite (p_e), a proporção de indivíduos resultantes de mutação e a probabilidade de um descendente herdar um alelo do progenitor de elite (ρ_e) foram otimizados tendo em conta os pressupostos que se apresentam de seguida. Os valores a testar têm em conta o intervalo de valores recomendados para estes parâmetros [12].

- A dimensão da população foi fixada a 50 indivíduos.
- O critério de paragem utilizado foi a execução do algoritmo durante 100 gerações.
- $p_e \in \{0.10, 0.15, 0.2, 0.25\}$
- $p_m \in \{0.10, 0.15, 0.2, 0.25, 0.30\}$
- $\rho_e \in \{0.5, 0.6, 0.7, 0.8\}$

Para as diferentes combinações possíveis dos parâmetros p_e , p_m e ρ_e foram executados os 50 subproblemas de teste correspondentes à variação no número de sensores do tipo I e do tipo II, procedendo-se da seguinte forma para cada combinação de valores:

1. Executar 5 vezes cada subproblema $i = 1, \dots, 50$ e reter o melhor resultado das 5 execuções, v_i , correspondente à proporção de área por cobrir. Tendo em conta o valor de referência de cada subproblema i , v_i^* , determinou-se $e_i = v_i - v_i^*$. Os valores de e_i são sempre não negativos uma vez que o problema é de minimização e os valores de referência podem ser visto como valores ótimos.

Em relação ao tempo de CPU é feita a média das 5 execuções, \bar{t}_i .

2. Com base nos valores obtidos para os 50 subproblemas são determinadas as seguintes medidas de qualidade:

- **Erro quadrático médio - EQM**

O erro quadrático médio pretende medir a discrepância existente entre o valor ótimo e o valor obtido através do algoritmo aplicado. Esta medida de precisão atribui maior peso aos resultados que se afastam mais do valor ótimo.

O EQM pode ser calculado através da expressão: $\sum_{i=1}^{50} \frac{(e_i)^2}{50}$.

- **Média dos desvios quadrados dos erros à média dos erros - $MDQEM$**

A média dos desvios quadrados dos erros à média dos erros é uma medida da variação dos erros e_i para cada subproblema i em relação à média dos erros obtidos no conjunto dos 50 subproblemas.

A expressão que permite calcular $MDQEM$ é dada por $\sum_{i=1}^{50} \frac{(e_i - \bar{e})^2}{50}$, sendo o valor de \bar{e} obtido através da expressão $\bar{e} = \sum_{i=1}^{50} \frac{e_i}{50}$.

Para a análise do tempo de computação determina-se o tempo total de execução média de cada um dos 50 subproblemas, $T = \sum_{i=1}^{50} \bar{t}_i$.

De acordo com os resultados que se apresentam na tabela A.1, pode-se concluir que o tempo total de CPU para a execução dos 50 subproblemas não apresenta uma grande variação nos vários testes realizados, uma vez que o número de indivíduos na população e o critério de paragem estão fixos (sendo estes os principais parâmetros responsáveis por alterações no tempo de CPU). As diferenças verificadas não correspondem a valores significativos tendo em conta que se estão a analisar 50 subproblemas.

Apesar dos valores obtidos terem, em geral, a mesma ordem de grandeza, é importante analisar as pequenas variações existentes pois, uma vez que as proporções totais de área para cobrir correspondem a valores entre 0 e 1, uma pequena melhoria no EQM significa a obtenção de resultados de melhor qualidade. Os valores apresentados permitem concluir que a combinação de parâmetros para a qual é menor o erro quadrático médio é ($p_e = 0.10$, $p_m = 0.3$, $\rho_e = 0.7$). Assim, esta será a combinação de parâmetros adotada para a realização dos testes que se apresentam nesta dissertação.

p_e	p_m	ρ_e	EQM ($\times 10^{-5}$)	$MDQEM$ ($\times 10^{-5}$)	T (segundos)
0.1	0.1	0.5	6.0656	3.4462	503.84
		0.6	6.5319	3.7504	509.77
		0.7	8.0521	5.0578	525.17
		0.8	10.125	6.0110	481.31
	0.15	0.5	5.2532	2.9817	508.93
		0.6	6.0809	3.7654	509.83
		0.7	5.7477	3.6243	509.76
		0.8	4.8557	2.7964	510.26
	0.2	0.5	5.2424	3.0969	526.31
		0.6	4.5146	2.6742	526.59
		0.7	3.7432	2.0688	528.34
		0.8	4.8457	2.7645	528.51
	0.25	0.5	4.2575	2.3479	525.48
		0.6	3.7211	2.0695	525.56
		0.7	3.4205	1.8221	525.98
		0.8	3.1065	1.7728	523.62
0.3	0.5	4.5982	2.5714	518.03	
	0.6	5.3485	3.2029	519.99	
	0.7	3.0160	1.6721	522.02	
	0.8	3.1860	1.7873	522.87	
0.15	0.1	0.5	5.8897	3.1919	454.15
		0.6	6.1254	3.4318	479.97
		0.7	7.1656	4.3098	486.40
		0.8	7.7425	4.3085	433.12
	0.15	0.5	4.9245	2.9282	518.14
		0.6	3.7953	2.0813	492.35
		0.7	7.5339	4.7482	485.67
		0.8	5.8087	3.3127	479.37
	0.2	0.5	5.1804	3.1083	466.31
		0.6	4.2964	2.5223	469.70
		0.7	3.5286	1.9684	472.20
		0.8	3.9129	2.1824	473.44
	0.25	0.5	5.0587	2.9775	467.04
		0.6	4.3189	2.5109	467.13
		0.7	3.7736	2.0860	467.43
		0.8	3.8783	2.1973	467.54
0.3	0.5	5.2425	2.9786	487.91	
	0.6	4.1835	2.4010	489.22	
	0.7	4.1568	2.3198	489.63	
	0.8	4.0114	2.3024	489.47	

p_e	p_m	ρ_e	EQM ($\times 10^{-5}$)	$MDQEM$ ($\times 10^{-5}$)	T (segundos)
0.2	0.1	0.5	5.3377	2.8993	496.56
		0.6	6.1827	3.5242	490.63
		0.7	6.3445	3.4220	472.79
		0.8	8.2915	4.6963	483.80
	0.15	0.5	4.1877	2.3370	499.16
		0.6	4.9756	2.8707	506.65
		0.7	5.6927	3.5453	483.93
		0.8	5.1911	2.9689	510.65
	0.2	0.5	4.5611	2.5791	443.76
		0.6	3.2355	1.7176	443.33
		0.7	3.8685	2.1346	444.03
		0.8	5.2391	2.9063	444.12
	0.25	0.5	5.0865	2.7536	457.73
		0.6	4.7789	2.8464	458.52
		0.7	4.5902	2.5634	458.37
		0.8	3.5644	2.0042	455.77
	0.3	0.5	5.5826	3.1045	452.94
		0.6	5.2552	3.1134	452.58
		0.7	3.6184	2.0708	453.13
		0.8	4.5048	2.6299	455.34
0.25	0.1	0.5	6.0088	3.2964	435.90
		0.6	5.5213	3.1690	435.32
		0.7	6.6317	3.9048	432.84
		0.8	7.2178	4.1597	434.25
	0.15	0.5	5.1922	2.9073	434.04
		0.6	5.1725	3.0638	434.19
		0.7	5.5729	3.3147	434.11
		0.8	5.7396	3.3072	435.39
	0.2	0.5	4.9997	2.8132	412.41
		0.6	4.6108	2.6038	412.19
		0.7	5.7556	3.3192	417.45
		0.8	4.5773	2.5864	416.56
	0.25	0.5	6.8506	4.1111	417.03
		0.6	5.2835	2.9215	417.32
		0.7	4.2039	2.3514	416.60
		0.8	4.7276	2.6227	418.71
	0.3	0.5	9.0851	5.4538	418.36
		0.6	7.6639	4.7500	418.36
		0.7	4.2090	2.3358	418.74
		0.8	4.2353	2.3361	419.37

Tabela A.1: Otimização dos parâmetros do algoritmo.

A.1.2 Dimensão da população e critério de paragem

A otimização do número de indivíduos na população e do número de gerações em que o algoritmo é executado foi feita em simultâneo. Estes dois critérios encontram-se inter-relacionados e a sua otimização conjunta permite a obtenção de melhores resultados.

Utilizando como parâmetros do *biased random-key genetic algorithm* os parâmetros anteriormente otimizados, foram testados os seguintes valores para a dimensão da população, p , e para o número de gerações, nG , (definido como critério de paragem):

- $p \in \{20, 30, 40, 50, 60, 100\}$
- $nG \in \{100, 300, 500, 1000\}$

Os 50 subproblemas foram executados para as diferentes combinações possíveis dos parâmetros p e nG e procedeu-se de modo análogo à otimização dos parâmetros do algoritmo BRKGA para determinar o valor de EQM , $MDQEM$ e T .

A tabela A.2 apresenta os resultados obtidos.

p	nG	EQM ($\times 10^{-5}$)	$MDQEM$ ($\times 10^{-5}$)	T (minutos)
20	100	12.258	7.1116	3.41
	300	3.1006	1.7582	10.17
	500	2.0848	1.1896	16.96
	1000	1.4476	0.8888	33.93
30	100	6.1443	3.6042	5.08
	300	1.9664	1.1416	15.56
	500	1.5387	0.9384	25.65
	1000	1.3138	0.8472	50.59
40	100	3.7234	2.0538	6.94
	300	1.8619	1.1232	21.29
	500	1.5150	0.9495	34.26
	1000	1.3097	0.8449	67.50
50	100	3.0160	1.6721	8.44
	300	1.6078	0.9596	25.44
	500	1.3831	0.8559	42.45
	1000	1.1821	7.5776	111.55
60	100	3.4816	2.0889	11.08
	300	1.5247	0.9400	30.87
	500	1.3226	0.8412	54.22
	1000	1.159	7.6647	129.97
100	100	1.9708	1.1902	16.74
	300	1.3196	0.8514	50.17
	500	1.2182	0.7996	83.47
	1000	1.1088	0.7401	167.16

Tabela A.2: Otimização da dimensão da população e do número de gerações.

Os resultados obtidos na tabela A.2 permitem concluir que, à medida que o número de gerações aumenta, fixando o número de indivíduos, o EQM diminui. A esta melhoria na qualidade das soluções está associado um aumento do tempo de CPU. Em geral, existe, também, uma tendência para obter resultados melhores num maior tempo computacional quando, fixando um determinado número de gerações, se aumenta o número de indivíduos.

No processo de otimização destes parâmetros, uma vez que os tempos de computação são bastante variáveis, interessa obter uma combinação que tenha em consideração a minimização dos tempos de CPU e a minimização do erro médio quadrático. Para tal, apresentam-se no gráfico da figura A.1 as combinações que correspondem a soluções não dominadas, ou seja, soluções para as quais não existe nenhuma outra combinação de parâmetros que as domine (neste caso, uma solução domina a outra quando não apresenta valores piores no tempo de CPU e no EQM e existe pelo menos um destes objetivos em que é melhor). Para facilitar a análise das soluções apresentadas, a cada dimensão da população encontra-se associada uma cor e a cada número de gerações corresponde um símbolo. A legenda do gráfico indica a dimensão da população e o número de gerações através da notação (p, nG) .

Esta análise biobjetivo permite escolher a dimensão da população e o número de gerações em que o algoritmo será executado consoante os pesos que se pretende atribuir a cada um dos critérios. Os testes realizados nesta dissertação pretendem obter soluções de boa qualidade num tempo razoável de computação. Assim, optou-se por escolher a combinação $(p = 50, nG = 500)$ que apresenta um valor de compromisso para estes critérios.

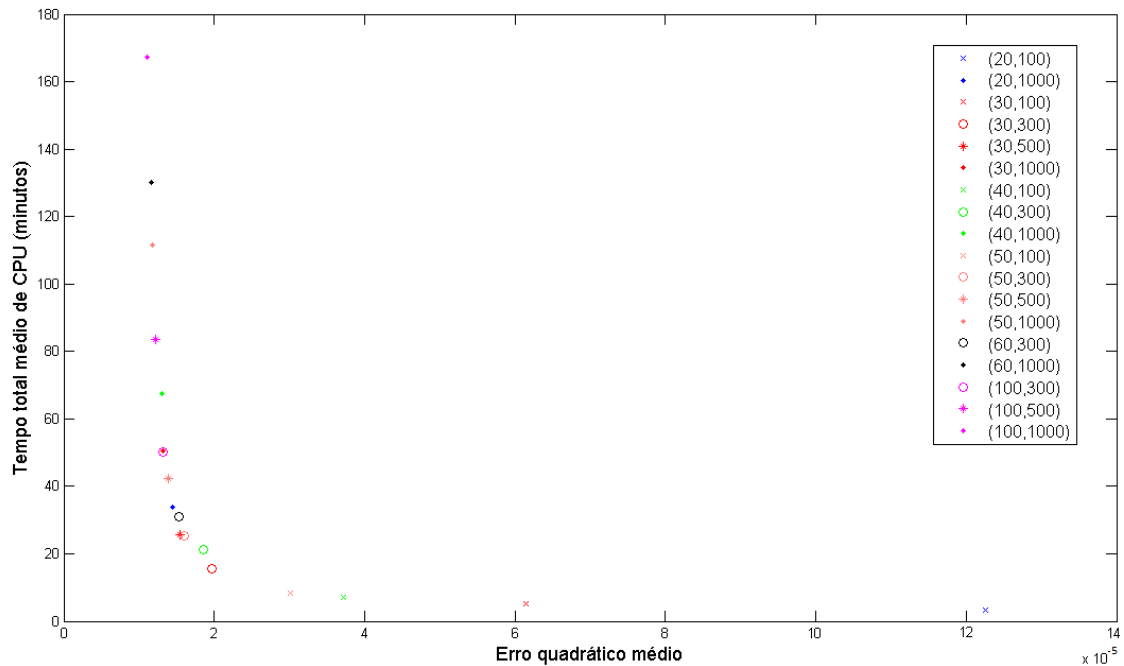


Figura A.1: Soluções não dominadas para as combinações de parâmetros p e nG .

Anexo B

Tabelas de resultados

B.1 Valores obtidos pelo BRKGA

Nesta seção apresentam-se os valores de proporção total de área por cobrir obtidos através da aplicação do *biased random-key genetic algorithm* aos casos em estudo que incluem obstáculos na área de interesse. O algoritmo foi executado 5 vezes para cada caso em estudo e para cada subproblema, encontrando-se o melhor valor obtido nas tabelas que se apresentam de seguida.

		Tipo I				
		0	1	2	3	4
Tipo II	0	1	0.5578	0.3016	0.0733	0.0001
	1	0.7858	0.3802	0.1366	0.0056	0
	2	0.5721	0.2220	0.0312	0.0017	0
	3	0.4144	0.0996	0.0093	0	0
	4	0.2800	0.0467	0.0028	0	0
	5	0.1642	0.0150	0	0	0
	6	0.0838	0.0048	0	0	0
	7	0.0283	0	0	0	0
	8	0.0093	0	0	0	0
	9	0	0	0	0	0

Tabela B.1: Proporção total de área por cobrir - estudo da admissibilidade das soluções aplicando o método de penalidade.

		Tipo I				
		0	1	2	3	4
Tipo II	0	1	0.5578	0.3009	0.0734	0.0001
	1	0.7858	0.3800	0.1348	0.0054	0
	2	0.5721	0.2223	0.0305	0.0013	0
	3	0.4147	0.1000	0.0085	0	0
	4	0.2800	0.0507	0.0009	0	0
	5	0.1629	0.0132	0	0	0
	6	0.0833	0.0033	0	0	0
	7	0.0279	0.0002	0	0	0
	8	0.0096	0	0	0	0
	9	0	0	0	0	0

Tabela B.2: Proporção total de área por cobrir - estudo da admissibilidade das soluções aplicando o método de reparação.

		Tipo I				
		0	1	2	3	4
Tipo II	0	1	0.5586	0.3160	0.0970	0.0011
	1	0.7858	0.3807	0.1583	0.0066	0
	2	0.5721	0.2231	0.0431	0.0013	0
	3	0.4142	0.1020	0.0131	0	0
	4	0.2804	0.0494	0.0060	0	0
	5	0.1745	0.0179	0	0	0
	6	0.0860	0.0043	0	0	0
	7	0.0268	0.0002	0	0	0
	8	0.0104	0	0	0	0
	9	0	0	0	0	0

Tabela B.3: Proporção total de área por cobrir - análise da linha de visão incluída no estudo da admissibilidade das soluções aplicando o método de penalidade.

		Tipo I				
		0	1	2	3	4
Tipo II	0	1	0.5586	0.3161	0.0970	0.0013
	1	0.7858	0.3807	0.1581	0.0066	0.0013
	2	0.5728	0.2228	0.0427	0.0013	0
	3	0.4146	0.1018	0.0129	0.0001	0
	4	0.2803	0.0528	0.0012	0	0
	5	0.1740	0.0245	0	0	0
	6	0.0864	0.0093	0	0	0
	7	0.0276	0	0	0	0
	8	0.0140	0	0	0	0
	9	0	0	0	0	0

Tabela B.4: Proporção total de área por cobrir - análise da linha de visão incluída no estudo da admissibilidade das soluções aplicando o método de reparação.

B.2 Tempo de CPU

Os resultados apresentados nesta seção têm por base a realização de 5 execuções do *biased random-key genetic algorithm* de acordo com os parâmetros previamente otimizados. Cada tabela apresenta, em segundos, o tempo médio de execução de cada subproblema para cada caso em estudo.

		Tipo I				
		0	1	2	3	4
Tipo II	0	1.17	7.82	13.97	19.92	25.79
	1	7.35	13.61	19.69	25.61	31.45
	2	13.16	19.37	25.38	31.23	36.91
	3	18.96	25.11	31.02	36.88	60.58
	4	24.72	30.76	36.64	60.11	68.40
	5	30.45	36.43	59.95	68.31	76.61
	6	36.13	59.71	68.00	76.36	84.90
	7	59.47	67.82	76.08	84.61	92.71
	8	67.60	75.93	84.60	92.53	100.89
	9	75.79	84.32	92.22	100.63	109.62

Tabela B.5: Tempo médio de CPU (em segundos) - caso de estudo sem restrições.

		Tipo I				
		0	1	2	3	4
Tipo II	0	0.97	46.94	89.02	131.00	173.75
	1	43.46	86.29	130.32	172.86	212.19
	2	84.19	128.01	169.62	210.37	252.03
	3	125.74	169.60	206.40	247.17	302.03
	4	163.56	204.72	245.78	302.62	345.10
	5	203.42	244.33	301.84	344.98	429.55
	6	251.85	299.10	342.46	385.15	426.20
	7	298.06	346.23	389.86	430.06	476.42
	8	347.74	467.06	591.76	627.61	613.84
	9	493.90	558.66	616.58	593.93	630.15

Tabela B.6: Tempo médio de CPU (em segundos) - estudo da admissibilidade das soluções aplicando o método de penalidade.

		Tipo I				
		0	1	2	3	4
Tipo II	0	0.99	47.26	99.30	140.62	179.72
	1	44.10	89.92	136.19	176.35	217.82
	2	85.74	133.90	182.27	218.25	261.82
	3	128.82	175.95	217.49	257.01	313.72
	4	171.70	214.75	256.53	304.17	345.19
	5	206.55	248.84	306.04	348.30	391.43
	6	246.41	304.70	347.10	390.86	432.62
	7	303.04	383.77	417.22	460.58	513.58
	8	387.16	436.79	477.92	528.70	557.84
	9	475.14	562.73	557.48	668.22	729.32

Tabela B.7: Tempo médio de CPU (em segundos) - estudo da admissibilidade das soluções aplicando o método de reparação.

		Tipo I				
		0	1	2	3	4
Tipo II	0	1.13	172.94	275.27	375.65	439.74
	1	106.41	251.80	349.90	433.83	492.75
	2	201.20	328.31	417.17	480.04	534.45
	3	283.17	396.82	465.28	524.72	596.54
	4	355.35	450.29	514.12	583.95	637.15
	5	423.72	505.59	584.81	630.83	687.03
	6	487.36	557.90	621.23	686.51	725.24
	7	549.45	640.15	666.74	721.89	776.95
	8	604.01	664.68	713.56	769.47	821.22
	9	654.18	704.32	766.01	807.92	869.55

Tabela B.8: Tempo médio de CPU (em segundos) - análise da linha de visão incluída no estudo da admissibilidade das soluções aplicando o método de penalidade.

		Tipo I				
		0	1	2	3	4
Tipo II	0	1.28	1.75.82	293.67	401.49	481.19
	1	107.87	264.68	376.43	466.78	529.28
	2	228.70	388.57	606.68	632.31	718.06
	3	292.89	413.48	496.54	626.15	706.78
	4	370.75	471.97	542.44	617.94	682.44
	5	440.19	529.35	674.61	763.55	813.94
	6	584.96	677.17	725.12	854.95	916.97
	7	666.82	675.36	756.30	781.17	840.59
	8	650.60	719.32	774.65	843.16	893.58
	9	708.98	768.49	827.35	891.53	941.01

Tabela B.9: Tempo médio de CPU (em segundos) - análise da linha de visão incluída no estudo da admissibilidade das soluções aplicando o método de reparação.

Anexo C

Soluções do problema biobjetivo

Neste anexo encontra-se a descrição detalhada de todas as soluções potencialmente não dominadas obtidas através da execução do *biased random-key genetic algorithm* para os exemplos do problema biobjetivo apresentados na seção 5.2.

As tabelas seguintes apresentam o valor de risco e custo associado a cada solução, o número de sensores de cada tipo a instalar (I, II e III) e as respectivas coordenadas do centro indicadas por ordem de tipo de sensor.

C.1 Espaço dos objetivos completo

Risco	Custo	I	II	III	Coordenadas
0.1347	1.5239	1	2	3	(0.6457,0.7001) (0.7391,0.1904) (0.1331,0.2814) (0.0999,0.4999) (0.2393,0.7871) (0.1815,0.7001)
0.0322	2.2507	2	2	4	(0.6812,0.6947) (0.7348,0.2311) (0.1632,0.8271) (0.1963,0.3147) (0.0066,0.2645) (0.0200,0.1833) (0.0275,0.1946) (0.0824,0.3828)
0.3273	1.0876	0	3	2	(0.3952,0.7758) (0.7296,0.1819) (0.7804,0.6223) (0.0106,0.1169) (0.1927,0.2620)
0.0478	2.1225	2	2	2	(0.6812,0.6947) (0.7348,0.2311) (0.1632,0.8333) (0.1562,0.3147) (0.0613,0.0951) (0.0275,0.0277)
0.2277	1.2529	1	1	3	(0.6385,0.6947) (0.7296,0.1818) (0.2166,0.2644) (0.0033,0.1658) (0.1030,0.7001)
0.0992	1.7408	1	3	2	(0.6615,0.6947) (0.1632,0.8272) (0.1200,0.3834) (0.7392,0.1705) (0.0999,0.1169) (0.0613,0.0735)
0.7033	0.4068	0	1	2	(0.6876,0.2335) (0.1667,0.2644) (0.0033,0.1060)
0.8968	0.1090	0	0	2	(0.2166,0.2644) (0.0022,0.1566)
0.1851	1.4300	1	2	2	(0.6469,0.6662) (0.1632,0.2700) (0.7296,0.1818) (0.0033,0.4356) (0.1030,0.7019)
0.2834	1.1741	1	1	2	(0.6541,0.6662) (0.7296,0.1818) (0.2166,0.2644) (0.0033,0.1566)
0.3981	0.8977	0	2	4	(0.5699,0.6248) (0.7296,0.2184) (0.2166,0.2645) (0.7859,0.4954) (0.0033,0.1658) (0.1030,0.7001)
0.0133	3.1504	4	1	3	(0.0917,0.7930) (0.7296,0.7417) (0.6957,0.1956) (0.0600,0.2311) (0.3350,0.3352) (0.4322,0.6653) (0.9153,0.5086) (0.1865,0.1867)
0.6815	0.4501	0	1	2	(0.7296,0.4091) (0.1465,0.1969) (0.1843,0.7001)
0.5801	0.5657	0	1	4	(0.7296,0.2184) (0.2166,0.2644) (0.5001,0.4954) (0.0022,0.1658) (0.1030,0.7001)
0.3786	0.9056	0	2	4	(0.5699,0.7001) (0.7296,0.2184) (0.2166,0.2645) (0.7308,0.4954) (0.0033,0.1658) (0.1030,0.7001)
0.8436	0.1753	0	0	3	(0.2166,0.2644) (0.5001,0.1385) (0.0022,0.1566)
0.9008	0.1064	0	0	2	(0.0106,0.1169) (0.1865,0.2621)
0.0007	4.5101	4	6	2	(0.0723,0.2878) (0.4557,0.7928) (0.7427,0.2648) (0.7893,0.7659) (0.0644,0.8272) (0.2294,0.2296) (0.5001,0.2267) (0.0056,0.1524) (0.0999,0.6568) (0.2314,0.4525) (0.1005,0.8497) (0.1030,0.4868)
0.7944	0.2447	0	0	4	(0.0106,0.1169) (0.5001,0.1931) (0.5523,0.3101) (0.1865,0.2611)
0.4748	0.7554	0	2	2	(0.6182,0.7001) (0.7296,0.2385) (0.2166,0.2644) (0.0022,0.1658)
0.6853	0.4175	0	1	2	(0.7296,0.2385) (0.2166,0.2644) (0.0022,0.1566)
0.6318	0.4915	0	1	3	(0.7296,0.2288) (0.2166,0.2644) (0.0022,0.1566) (0.1030,0.7001)
0.4204	0.8316	0	2	3	(0.6408,0.7001) (0.7296,0.2184) (0.2166,0.2644) (0.0033,0.1658) (0.1030,0.7001)
0.2192	1.3507	1	1	4	(0.6395,0.7045) (0.7334,0.2184) (0.0999,0.1669) (0.8106,0.1385) (0.2208,0.9514) (0.2156,0.2620)
0.0310	2.7085	3	1	4	(0.0917,0.7930) (0.7296,0.7417) (0.6957,0.1553) (0.1970,0.3334) (0.6016,0.7927) (0.4322,0.6653) (0.0067,0.5086) (0.1865,0.1867)
0.9501	0.0936	0	0	2	(0.0066,0.1169) (0.0075,0.0277)
0.5018	0.7390	0	2	2	(0.5699,0.6247) (0.7296,0.2189) (0.2166,0.2644) (0.0033,0.1658)
0.1289	1.5670	1	2	3	(0.6859,0.6947) (0.1632,0.8272) (0.7152,0.1693) (0.1465,0.2644) (0.0022,0.1658) (0.1030,0.3297)
0.0011	4.3905	4	5	3	(0.0494,0.3325) (0.6859,0.8281) (0.7427,0.1262) (0.0761,0.7659) (0.4298,0.4952) (0.5001,0.4045) (0.7296,0.2385) (0.8201,0.4999) (0.1150,0.1524) (0.1465,0.8832) (0.0033,0.2542) (0.9130,0.7662)

Tabela C.1: Soluções potencialmente não dominadas - espaço dos objetivos completo.

C.2 Espaço dos objetivos reduzido

Risco	Custo	I	II	III	Coordenadas
0.1347	1.5239	1	2	3	(0.6457,0.7001) (0.7391,0.1904) (0.1331,0.2814) (0.0999,0.4999) (0.2393,0.7871) (0.1815,0.7001)
0.2350	1.2453	1	1	3	(0.6503,0.6614) (0.7391,0.1693) (0.1904,0.2580) (0.0203,0.1070) (0.2057,0.7001)
0.1927	1.3270	1	1	4	(0.6457,0.7001) (0.7391,0.1904) (0.2203,0.2580) (0.0130,0.3924) (0.2304,0.2306) (0.2057,0.7001)
0.2208	1.2646	1	1	3	(0.6457,0.7001) (0.7391,0.1904) (0.1904,0.2579) (0.2176,0.2185) (0.2057,0.7001)
0.1801	1.4333	1	2	2	(0.6503,0.6614) (0.7391,0.1579) (0.1331,0.2814) (0.0999,0.4999) (0.1815,0.7001)
0.1019	1.7076	1	3	2	(0.6503,0.6614) (0.0559,0.1693) (0.7391,0.1693) (0.1708,0.8093) (0.2203,0.2600) (0.0111,0.3925)
0.2887	1.1697	1	1	2	(0.6503,0.6614) (0.7391,0.1693) (0.1904,0.2579) (0.0203,0.1053)
0.2817	1.1785	1	1	2	(0.6503,0.6614) (0.7391,0.1693) (0.1904,0.2580) (0.2183,0.2187)
0.1625	1.4479	1	2	2	(0.6503,0.6614) (0.7391,0.1693) (0.1331,0.2814) (0.2781,0.7871) (0.1815,0.7001)
0.1492	1.4823	1	2	2	(0.6815,0.7001) (0.7391,0.1693) (0.1413,0.3744) (0.2393,0.7871) (0.1815,0.7001)
0.0893	1.7247	1	3	2	(0.6815,0.6614) (0.0564,0.1905) (0.7567,0.1912) (0.1708,0.8093) (0.2292,0.2660) (0.0111,0.4028)
0.0567	1.9358	1	3	4	(0.6937,0.7001) (0.1813,0.8717) (0.8027,0.1816) (0.1332,0.3744) (0.7009,0.2580) (0.1396,0.1398) (0.8898,0.2490) (0.1301,0.1303)
0.2123	1.3007	1	1	4	(0.6503,0.6614) (0.7391,0.1693) (0.2203,0.2580) (0.0075,0.0158) (0.2183,0.2642) (0.2057,0.7001)
0.1447	1.5006	1	2	2	(0.6617,0.6614) (0.7391,0.1693) (0.1413,0.8093) (0.2028,0.2580) (0.2280,0.2282)
0.2400	1.2432	1	1	3	(0.6457,0.6614) (0.7391,0.1579) (0.1904,0.2580) (0.0130,0.1053) (0.1815,0.7001)
0.1998	1.3164	1	1	4	(0.6503,0.6614) (0.7391,0.1693) (0.2203,0.2570) (0.0111,0.3925) (0.2314,0.2318) (0.2057,0.7001)
0.0696	1.7942	1	3	2	(0.6937,0.7001) (0.7730,0.1693) (0.1813,0.8717) (0.1332,0.3744) (0.7009,0.2291) (0.1232,0.1530)
0.1088	1.6083	1	2	4	(0.6815,0.6614) (0.7391,0.1755) (0.1331,0.2814) (0.0999,0.4999) (0.2116,0.9599) (0.1528,0.7983) (0.1815,0.7001)
0.1072	1.7036	1	3	2	(0.6503,0.6614) (0.0559,0.1693) (0.7391,0.1693) (0.1708,0.7861) (0.2203,0.2600) (0.0111,0.3925)
0.0575	1.8848	1	3	3	(0.6937,0.7001) (0.1813,0.8717) (0.8027,0.1816) (0.1332,0.3744) (0.7009,0.2580) (0.1396,0.1398) (0.8898,0.2985)
0.1187	1.5996	1	2	4	(0.6503,0.6614) (0.7391,0.1755) (0.1331,0.2814) (0.0999,0.4999) (0.2116,0.9599) (0.1528,0.7983) (0.1815,0.7001)
0.0666	1.8007	1	3	2	(0.6937,0.7001) (0.1813,0.8717) (0.8027,0.1816) (0.1413,0.3744) (0.7009,0.2580) (0.1396,0.1398)
0.1651	1.4467	1	2	2	(0.6457,0.6614) (0.7391,0.1693) (0.1331,0.2814) (0.2781,0.7871) (0.1815,0.7001)
0.2698	1.2121	1	1	2	(0.6617,0.7001) (0.7391,0.1904) (0.2312,0.2314) (0.2057,0.7001)
0.2720	1.2005	1	1	2	(0.6815,0.7001) (0.7391,0.1904) (0.2203,0.2580) (0.2314,0.2316)

Tabela C.2: Soluções potencialmente não dominadas - espaço dos objetivos reduzido.

Anexo D

Funcionamento dos programas

Os exemplos apresentados no capítulo 5 pretendem ilustrar o funcionamento do *biased random key genetic algorithm*, focando alguns aspetos relevantes que podem ser concluídos através das instâncias testadas. No entanto, os programas desenvolvidos para o problema mono-objetivo e para o problema biobjetivo permitem que sejam obtidas soluções para quaisquer outras instâncias do problema de localização de sensores.

A estrutura dos programas corresponde, essencialmente, à implementação do algoritmo 4 considerando como decodificador o algoritmo 5 e 8 respetivamente para o problema mono-objetivo e biobjetivo. Neste anexo, é feita uma breve descrição ao modo de funcionamento dos programas no que diz respeito à introdução dos dados específicos de cada instância que se pretende testar.

Para o problema mono-objetivo, o programa desenvolvido pode ser encontrado no ficheiro MATLAB denominado *BRKGA_MO.m*. Apesar de terem sido implementadas outras funções necessárias ao funcionamento do algoritmo, este é o programa principal e ao longo do código são apresentados vários comentários que ajudam a identificar melhor as várias etapas da heurística. Existem três aspetos que podem ser definidos pelo utilizador:

- Parâmetros necessários ao funcionamento do algoritmo
- Dados do problema
- Caracterização da área de interesse

Cada um dos aspetos anteriormente referidos encontra-se num ficheiro independente do programa de forma a permitir ao utilizador a alteração dos parâmetros, dos dados e da área de interesse sem interferir com o código implementado.

A figura D.1 apresenta o ficheiro de dados *parametros.dat* onde se encontram os parâmetros necessários ao funcionamento do algoritmo. Uma vez que estes parâmetros foram previamente otimizados, os mesmos não devem ser alterados pelo utilizador. Apenas devem ser introduzidos os dados específicos do problema em estudo assim como a caracterização da área de interesse, conforme se apresenta nas figuras D.2 e D.3.

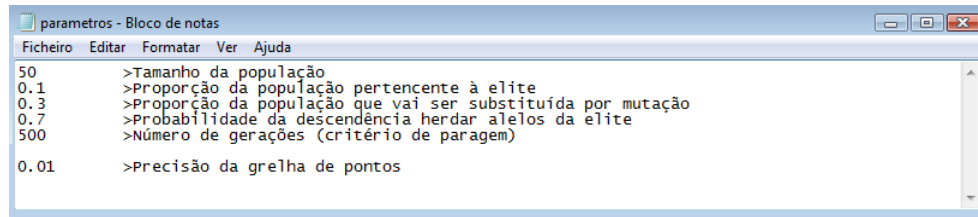


Figura D.1: Parâmetros necessários ao funcionamento do algoritmo.

Os dados do problema devem ser introduzidos no ficheiro *dados.dat* pela ordem ilustrada na figura D.2. Seguem-se algumas considerações a ter em conta.

- **Diferentes tipos de sensores existentes**

Os dois primeiros valores do ficheiro indicam o número de diferentes tipos de sensores existentes, respetivamente, para os sensores circulares e para os sensores setor circular. De seguida, deve-se indicar o número de recursos a instalar do respetivo tipo. No exemplo da figura D.2 existem 2 tipos de sensores circulares, pretendendo-se instalar 1 unidade de cada tipo. À semelhança dos exemplos apresentados no capítulo 5, estes tipos de sensores serão referidos daqui em diante como tipo I e tipo II. Em relação aos sensores setor circular, existe apenas um tipo de sensores que se designará por tipo III pretendendo-se instalar 1 unidade deste recurso.

- **Características dos sensores**

Um sensor circular define-se através do respetivo raio de deteção. Assim, deve-se indicar para cada tipo de sensor o respetivo raio. Os sensores de tipo I têm um raio de $\frac{\sqrt{2}}{4}$ e os sensores de tipo II têm um raio de $\frac{\sqrt{2}}{6}$.

Para os sensores setor circular deve-se definir para além do raio, o respetivo ângulo de visão. Neste caso, os sensores de tipo III têm um raio de $\frac{\sqrt{2}}{6}$ e um ângulo de visão de $\frac{\pi}{2}$ radianos.

- **Restrições de localização**

Consoante o tipo de sensor, pode existir uma restrição de localização que lhe está associada. Existem três possibilidades: localização em terra, localização no mar, ou não existe restrição de localização, sendo indiferente a zona onde o sensor é instalado. No exemplo apresentado na figura D.2 todos os sensores devem ser localizados no mar.

- **Opções de execução**

Devem ainda ser definidas algumas opções de execução. Deve-se indicar se se pretende ou não incluir a análise da linha de visão e qual o método de admissibilidade a considerar. Quando a área de interesse não inclui obstáculos deve-se selecionar a opção “sem qualquer método”. O método de penalidade por morte foi também implementado mas não foi referido no capítulo 5 uma vez que o método de penalidade

```

dados - Bloco de notas
Ficheiro Editar Formatar Ver Ajuda
2 >Número de diferentes tipos de sensores circulares
1 >Número de diferentes tipos de sensores setor circular
1 1 >Número de sensores circulares de cada tipo
1 >Número de sensores setor circular de cada tipo
sqrt(2)/4 sqrt(2)/6 >Raio dos sensores circulares de cada tipo
sqrt(2)/6 >Raio dos sensores SC de cada tipo
pi/2 >Ângulo de visão para cada sensor SC
0 0 >Localização permitida para cada tipo de sensor
(-1 terra, 0 mar, 1 indiferente)
1 >Obstrução da linha de visão pelos obstáculos (1-Sim, 0-Não)
1 >Método utilizado para lidar com a admissibilidade
(0 - Sem qualquer método)
1 >Reparação das soluções
2 >Penalidade por morte
3 >Um sensor não admissível é inativo

```

Figura D.2: Dados do problema.

que considera inativos os sensores inválidos é mais adequado para o problema em estudo.

A área de interesse deve ser definida no ficheiro *porto.dat*, independente do programa, no qual se encontram as indicações necessárias para a respetiva caracterização. O programa faz a leitura do mesmo permitindo, assim, que o algoritmo seja aplicado a qualquer região marítima com ou sem obstáculos.

Na figura D.3 apresenta-se o ficheiro *porto.dat* com um exemplo que corresponde à área de interesse da figura D.4. Neste ficheiro devem ser definidos os seguintes aspetos pela ordem indicada:

1. Número de obstáculos

No exemplo da figura D.3 existem 2 obstáculos que correspondem aos dois polígonos da figura D.4.

2. Número máximo de coordenadas de um obstáculo + 1

Este valor pretende dimensionar as matrizes das coordenadas.

Neste caso, um dos obstáculos tem 3 coordenadas e o outro tem 4. Assim, o valor pretendido é $5 = \max\{3, 4\} + 1$.

3. Matriz das abcissas

Há alguns aspetos importantes a salientar na definição das matrizes que indicam as coordenadas dos obstáculos. Em primeiro lugar, assume-se que diferentes obstáculos não se intersejam. Por outro lado, por uma questão de implementação, as coordenadas dos obstáculos devem ser indicadas por ordem, no sentido contrário aos ponteiros do relógio.

Na matriz das abcissas, cada linha corresponde a um obstáculo distinto e cada elemento corresponde à abcissa de um ponto do respetivo obstáculo. Depois de indicadas todas as abcissas deve-se repetir a abcissa inicial de forma a “fechar” o polígono.

```

porto - Bloco de notas
Ficheiro  Editar  Formatar  Ver  Ajuda
2
5
0 0.5 0.5 0 -1
0.15 0.4 0.4 0.15 0.15
0 0 0.5 0 -1
0.55 0.55 0.8 0.8 0.55
-----
Definir:
*Número de obstáculos
*Número máximo de coordenadas de um obstáculo + 1
*matrizPortoX
*matrizPortoY

```

Figura D.3: Ficheiro *porto.dat*.

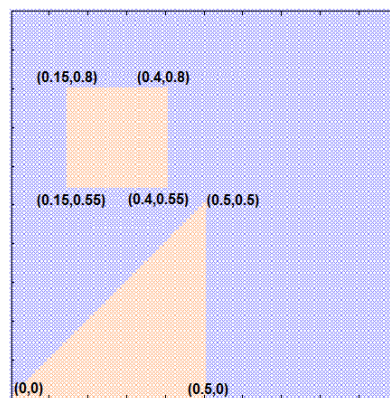


Figura D.4: Área de interesse correspondente ao ficheiro da figura D.3.

As posições indicadas a -1 servem apenas para efeitos de dimensionamento da matriz, de forma a que todas as linhas tenham a mesma dimensão. Assim, apenas a linha correspondente ao obstáculo com mais coordenadas não tem posições com o valor -1 atribuído.

4. Matriz das ordenadas

Depois de definida a matriz das abcissas, define-se, por fim, a matriz das ordenadas. Esta matriz é definida de forma semelhante à matriz das abcissas, conforme explicado no tópico anterior.

O programa *BRKGA_BIO.m* desenvolvido para o problema biobjetivo funciona de modo análogo. As modificações existentes nos dados a introduzir foram referidas ao longo da dissertação e são facilmente identificadas.