

Pedro Miguel Carrasqueira

OTIMIZAÇÃO COM META-HEURÍSTICAS POPULACIONAIS EM MODELOS MULTIOBJETIVO E EM DOIS NÍVEIS

Tese de doutoramento em Gestão – Ciência Aplicada à Decisão, orientada por Prof. Doutora Maria João Teixeira Gomes Alves e por Prof. Doutor Carlos Alberto Henggeler Antunes e apresentada à Faculdade de Economia da Universidade de Coimbra

maio de 2017





FEUC FACULDADE DE ECONOMIA
UNIVERSIDADE DE COIMBRA

Pedro Miguel Carrasqueira

Otimização com Meta-Heurísticas Populacionais em Modelos Multiobjetivo e em Dois Níveis

Tese de Doutoramento em Gestão – Ciência Aplicada à Decisão,
apresentada à Faculdade de Economia da Universidade de
Coimbra para obtenção do grau de Doutor

Orientadores: Prof. Doutora Maria João Teixeira Gomes Alves e
Prof. Doutor Carlos Alberto Henggeler Antunes

Coimbra, 2017

Aos meus filhos, Joana e João

Agradecimentos

Aos meus orientadores Professora Maria João Teixeira Gomes Alves e Professor Carlos Alberto Henggeler de Carvalho Antunes, pelas suas valiosas contribuições, pela supervisão, compreensão, incentivo e empenho que sempre manifestaram ao longo de todo o trabalho de preparação desta tese

Ao Instituto de Engenharia de Sistemas e Computadores de Coimbra (INESC Coimbra) pelo apoio financeiro concedido para a minha participação em diversas conferências científicas, tanto a nível nacional como no estrangeiro

Ao DEEC da Faculdade de Ciências e Tecnologia da Universidade de Coimbra, pelos meios informáticos disponibilizados para realização de algumas experiências computacionais

À minha família que sempre me incentivou a percorrer este caminho

À minha mulher e aos meus filhos pelo apoio manifestado e pela sua compreensão para com a minha indisponibilidade em os acompanhar em muitas atividades ao longo deste percurso

Resumo

A otimização multiobjetivo permite tratar problemas considerando explicitamente múltiplos eixos de avaliação do mérito das soluções, possibilitando obter um conjunto de soluções não dominadas, as quais estabelecem um compromisso entre os valores alcançados para as funções objetivo. As meta-heurísticas populacionais têm-se revelado adequadas em problemas multiobjetivo fortemente não lineares e/ou combinatórios, pela capacidade de fazer evoluir a população de soluções para o conjunto de soluções não dominadas.

Neste trabalho começámos por analisar algumas das principais meta-heurísticas populacionais para problemas de otimização multiobjetivo. Propusemos modificações no algoritmo baseado no mecanismo eletromagnético, o qual tem sido pouco explorado em problemas multiobjetivo, resultantes da avaliação positiva do respetivo desempenho em comparação com outras meta-heurísticas populacionais adequadas para problemas com variáveis contínuas. Utilizámos as meta-heurísticas com melhor desempenho neste primeiro estudo para desenvolver algoritmos para problemas mais complexos, nos quais existem dois níveis de decisão e com múltiplas funções objetivo. Começámos por desenvolver um algoritmo para resolução de um problema em dois níveis (*bi-level*), multiobjetivo em ambos os níveis, sendo o problema de cada nível resolvido por uma meta-heurística populacional baseada em optimização por enxames de partículas.

Uma das áreas de aplicação em que os modelos de otimização em dois níveis são particularmente adequados diz respeito à interação entre as empresas comercializadoras de eletricidade (retalhista) e os consumidores. Neste trabalho desenvolvemos um modelo de otimização em dois níveis para um problema de definição de tarifas dinâmicas por parte de um comercializador de energia elétrica tendo em conta a resposta por parte do consumidor residencial, através da calendarização do funcionamento dos aparelhos que prestam diferentes serviços de energia. Num primeiro modelo considerámos um problema em dois níveis mono-objetivo, em que o objetivo do retalhista é a maximização do lucro e o objetivo do consumidor é a minimização da sua conta de eletricidade. Foram desenvolvidos algoritmos baseados em optimização por enxames de partículas e em algoritmos evolucionários especialmente adaptados para a resolução deste problema. O problema do consumidor foi ainda transformado num problema linear inteiro-misto e resolvido de forma exata por meio de um *solver*, integrado numa abordagem híbrida que utiliza operadores genéticos para resolver o problema do nível superior. Desenvolvemos

também um modelo com duas funções objetivo no nível inferior, considerando a minimização do possível desconforto associado à utilização dos equipamentos fora do período de funcionamento preferido como um objetivo adicional do consumidor. Adaptámos os algoritmos desenvolvidos anteriormente, baseados em otimização com enxames de partículas e em algoritmos evolucionários, para resolver diferentes instâncias do modelo, nas quais considerámos diferentes perfis de consumidores face à disponibilidade de alteração dos períodos de funcionamento dos equipamentos. Estes algoritmos determinam soluções para o problema de acordo com as perspetivas otimista e pessimista do retalhista sobre a reação do consumidor.

Ao longo deste trabalho verificámos que as abordagens baseadas em otimização por enxame de partículas e em algoritmos evolucionários obtiveram, em geral, bons resultados, quer para problemas *benchmark*, quer para o modelo de otimização em dois níveis para estudo da interação entre o retalhista e consumidores de energia elétrica. Estes algoritmos poderão ser adaptados para a resolução de outros problemas de otimização em dois níveis e/ou multiobjetivo.

Palavras-Chave: Otimização Multiobjetivo; Otimização em dois níveis (*Bi-level*); Otimização por Enxame de Partículas; Algoritmo baseado no mecanismo eletromagnético (*Electromagnetism-like Mechanism*); Algoritmos Evolucionários.

Abstract

Multiobjective optimization enables to deal with problems that consider multiple axes of evaluation of the merit of the solutions, providing a set of non-dominated solutions which establish a compromise between the objective functions. In non-linear and/or combinatorial problems, population-based metaheuristics have shown a good performance enabling populations to evolve to the non-dominated solution set.

In this work we started by analysing some of the main population-based metaheuristics to solve multiobjective optimization problems. We proposed modifications in the electromagnetism-like mechanism, which has not been much exploited in multiobjective problems. These modifications resulted from the assessment of this approach in comparison with other population-based metaheuristics for problems with continuous variables. We considered the best performing metaheuristics in this study to design algorithms for more complex problems with two decision levels (bi-level) and multiple objective functions. Accordingly, we have developed an algorithm based on particle swarm optimization to solve bi-level problems with multiple objective functions at both levels.

Bi-level optimization models are particularly adequate to study the interaction between electricity retailers and consumers. We have developed a bi-level optimization model to a problem of designing dynamic tariffs by electricity retailers, considering the residential consumer response by means of rescheduling the operation of his loads providing distinct energy services. In a first model, the retailer intends to maximize its profit and the consumer aims at minimizing his electricity bill. Algorithms based on particle swarm optimization and evolutionary algorithms adapted to the resolution of this problems have been developed. The consumer problem was reformulated as mixed integer linear problem, which was solved exactly integrated in a hybrid approach that uses genetic operators to solve the upper level problem. We have further developed another version of this model with a bi-objective problem at the lower level, which considers not only the minimization of the consumer's cost but also the consumer's dissatisfaction (comfort dimension) associated with rescheduling load operation. We have adapted the previously developed bi-level approaches based on particle swarm optimization and evolutionary algorithms to solve different instances of the model, considering different consumer's profiles regarding the willingness to accept changes in the load operation

periods. These approaches determine solutions to the problem according to the optimistic and the pessimistic perspectives of the retailer concerning the consumer's reaction.

Throughout this work we have observed that the approaches based on particle swarm optimization and evolutionary algorithms achieved, in general, good results, both in benchmark problems and the bi-level optimization problem to study the interaction between electricity retailers and consumers. These algorithms can be adapted to solve other bi-level and/or multiobjective optimization problems.

Keywords: Multiobjective Optimization; Bi-level Optimization; Particle Swarm Optimization; Electromagnetism-like Mechanism Algorithm; Evolutionary Algorithms.

Índice

1	Introdução	1
1.1	Motivação	1
1.2	Objetivos	5
1.3	Estrutura da tese	8
2	Meta-heurísticas populacionais multiobjetivo	11
2.1	Introdução	11
2.2	Otimização multiobjetivo	13
2.3	Descrição de algoritmos representativos do estado da arte MOEA, MOPSO, MODE e MOEM	14
2.3.1	Algoritmo NSGA-II	14
2.3.2	Algoritmo OMOPSO	16
2.3.3	Algoritmo MOSADE	18
2.3.4	Algoritmo MOEM	20
2.4	Algoritmo EMOEM	24
2.4.1	Descrição	24
2.5	Resultados experimentais	31
2.5.1	Medidas de desempenho	31
2.5.2	Problemas teste	32
2.5.3	Parametrização dos algoritmos	32
2.5.4	Comparação dos algoritmos EMOEM, MOEM, OMOPSO, MOSADE e NSGA-II	33
2.6	Considerações finais	37
3	Abordagens PSO e evolucionária para problemas em dois níveis multiobjetivo	39
3.1	Motivação	39
3.2	Formulação	42
3.3	Algoritmo BLEMO	44
3.4	Algoritmo MO-BLPSO	45
3.5	Análise comparativa entre os algoritmos MO-BLPSO e BLEMO	52
3.5.1	Problemas teste	52
3.5.2	Medidas de desempenho	53
3.5.3	Parametrização dos algoritmos	54
3.5.4	Resultados	54
3.6	Considerações finais	58
4	Um modelo BL para o estudo das interações retalhista – consumidor de energia elétrica	59
4.1	Introdução	59

4.2	Enquadramento do problema e abordagens	60
4.3	Modelo BL para o problema do mercado retalhista de eletricidade	65
4.4	Algoritmo híbrido evolucionário com <i>solver</i> MILP: H-BLEA	71
4.5	Algoritmos baseados em EA e PSO.....	75
4.5.1	Algoritmo BLEA	76
4.5.2	Algoritmo BL de otimização por enxame de partículas (BLPSO).....	80
4.6	Comparação dos algoritmos – Resultados experimentais e discussão.....	84
4.6.1	Dados do problema e definição de parâmetros	84
4.6.2	Resultados	89
4.7	Considerações finais	99
5	Um modelo BL semi-vetorial para o estudo da interação retalhista - consumidor de energia elétrica	101
5.1.	Motivação	101
5.2	Abordagens pessimista vs. otimista no problema BL semi-vetorial.....	102
5.3	Problema BL bi-objetivo no nível inferior para a interação retalhista-consumidor no mercado de eletricidade	103
5.4	Algoritmos populacionais para o problema BL semi-vetorial para a interação retalhista-consumidor no mercado de eletricidade	106
5.4.1	Algoritmo BL-MOPSO.....	107
5.4.2	Algoritmo BL-MOEA.....	109
5.5	Exemplo de aplicação dos algoritmos BL-MOPSO e BL-MOEA no problema BL semi-vetorial para a interação retalhista-consumidor de energia elétrica.....	111
5.5.1	Dados do problema	112
5.5.2	Resultados.....	114
5.6	Considerações finais	116
6	Conclusões e desenvolvimentos futuros	119
	Referências.....	123
	Apêndice.....	131

1 Introdução

1.1. Motivação

A utilização de modelos matemáticos para apoio à decisão em problemas de otimização, captando a complexidade do ambiente económico, social e tecnológico, requer algoritmos mais sofisticados e maior capacidade computacional. Estes problemas surgem em diversas áreas, nomeadamente nas ciências exatas, económicas e nos diversos campos de engenharia, podendo ser resolvidos através de métodos exatos ou de métodos aproximados como as meta-heurísticas. A resolução de problemas de otimização difíceis através de métodos exatos (programação matemática), torna-se muitas vezes impraticável devido à complexidade dos cálculos envolvidos e às características dos próprios problemas, designadamente as que derivam da dimensão, natureza combinatoria, restrições e não linearidades. A aplicação de métodos exatos exige, em geral, a verificação de condições de convergência, as quais, em alguns casos, podem ser demasiado exigentes. Neste contexto, para ultrapassar estas dificuldades têm sido propostas abordagens alternativas, as meta-heurísticas, para resolução de problemas de otimização.

As meta-heurísticas são métodos de pesquisa de soluções que combinam procedimentos heurísticos de pesquisa local (adaptados a cada tipo de problema) com estratégias de controlo da pesquisa de alto nível, de modo a criar procedimentos capazes de efetuar a exploração do espaço de soluções escapando a ótimos locais e não requerendo pressupostos quanto a diferenciabilidade ou convexidade das funções objetivo. As meta-heurísticas têm sido usadas para resolver problemas complexos, geralmente não tratáveis com algoritmos exatos de programação matemática. O desenvolvimento de meta-heurísticas tem recebido uma atenção crescente, quer do ponto de vista teórico e metodológico, quer do ponto de vista de implementação computacional, alicerçada no sucesso obtido na resolução de problemas complexos, tipicamente de natureza combinatoria e/ou não linear, em diversas áreas de aplicação.

As meta-heurísticas podem ser genericamente classificadas em abordagens de pesquisa local e baseadas em populações. As abordagens baseadas em pesquisa local

consideram, em geral, uma única solução candidata que procuram modificar e melhorar. As meta-heurísticas de pesquisa local incluem *simulated annealing* (Kirkpatrick et al., 1983), *tabu search* (Glover, 1989), *variable neighborhood search* (Mladenovic & Hansen, 1997), entre outras. As abordagens baseadas em populações exploram um conjunto de soluções candidatas (população de soluções). Neste tipo de meta-heurísticas incluem-se, para além dos algoritmos evolucionários (*evolutionary algorithms* - EA), a otimização por enxames de partículas (*particle swarm optimization* - PSO), o mecanismo inspirado em eletromagnetismo (*electromagnetism-like mechanism* - EM), a evolução diferencial (*differential evolution* - DE), entre outras. Estas abordagens têm sido desenvolvidas, em geral, a partir da analogia com processos físicos ou biológicos. Contudo, as operações usadas por estes algoritmos não estão limitadas à replicação computacional dos processos identificados nas analogias físicas ou biológicas.

Os problemas reais envolvem geralmente múltiplos aspetos de avaliação do mérito das soluções. A otimização multiobjetivo permite tratar este tipo de problemas possibilitando obter um conjunto de soluções de compromisso entre os múltiplos objetivos. Assim, os modelos matemáticos devem incluir explicitamente diferentes funções objetivo que reflitam as diferentes perspetivas de avaliação. Estas perspetivas podem representar aspetos de avaliação antagónicos conduzindo a funções conflituantes entre si, o que inviabiliza, em geral, a existência de uma solução ótima do problema.

Perante a existência de múltiplas funções objetivo, a noção de solução ótima (que conduz ao melhor valor admissível para uma única função objetivo) cede lugar à noção de solução ótima de Pareto, também designada por eficiente ou não dominada. Uma solução não dominada caracteriza-se por não existir outra solução admissível que melhore simultaneamente todos os objetivos: a melhoria num objetivo é alcançada à custa da degradação do valor de, pelo menos, um dos outros. Não existindo uma solução que otimize simultaneamente todas as funções objetivo, a simples comparação entre soluções não dominadas não fornece informação relevante na procura de uma solução não dominada que constitua uma possível solução final do problema multiobjetivo. Ou seja, a relação de dominância é uma relação “pobre”, que não permite distinguir entre as soluções não dominadas e que deve ser enriquecida por informação sobre as preferências do decisor. Num contexto de apoio à decisão, esta informação deve ser de fácil aceitação (ou seja, sem exigir a satisfação de requisitos de carácter matemático que dificilmente são verificados em problemas reais). Para além disso, em abordagens não baseadas em programação matemática não é possível garantir teoricamente a não dominância das

soluções. Não sendo também possível a comparação exaustiva entre todas as soluções, deve então falar-se em soluções potencialmente não dominadas (Antunes et al., 2016). Neste trabalho consideraremos a designação de solução ótima de Pareto ou eficiente para nos referirmos a uma solução do problema para a qual é possível garantir que não há nenhuma outra solução admissível que a domine, e utilizaremos a designação de solução não dominada para identificar uma solução que pode ser apenas potencialmente não dominada, i.e. o algoritmo que a determinou não conseguiu encontrar outra solução admissível que a domine.

As abordagens meta-heurísticas têm sido aplicadas a problemas multiobjetivo. Embora algumas destas abordagens resolvam o problema multiobjetivo como um problema mono-objetivo, agregando as múltiplas funções objetivo numa única função escalar, muitos algoritmos mantêm a estrutura multiobjetivo do problema, utilizando mecanismos adequados à sua resolução baseados na relação de não dominância. As meta-heurísticas populacionais são particularmente adequadas para a obtenção de uma representação de todo o conjunto das soluções não dominadas.

Em muitos problemas de otimização, a concretização dos objetivos do decisor está dependente da solução de outro problema de otimização controlado por outro decisor, que tem os seus próprios objetivos. O processo de decisão é sequencial havendo uma hierarquia entre os decisores e cada um dos decisores controla um conjunto diferente de variáveis. O decisor de nível superior é habitualmente designado por líder e o decisor de nível inferior por seguidor. Para representar este processo teremos de considerar dois problemas de otimização, um inserido no outro e cada um relativo a um decisor, sendo o problema de nível inferior uma restrição do problema de nível superior - problema de otimização em dois níveis (*bi-level* - BL). Os problemas de otimização do nível superior e do nível inferior podem ter diferentes características relativamente às funções objetivo e às restrições (linear/não linear) e ao número de funções objetivo. Os problemas BL são geralmente difíceis de resolver por serem inerentemente não convexos, mesmo no caso em que todas as funções são lineares. A não convexidade dos problemas em cada nível, a possível dependência das restrições do problema do nível superior de variáveis controladas pelo seguidor e a existência de múltiplas soluções ótimas no nível inferior são características que tornam um problema de otimização BL especialmente difícil de resolver. Contudo, encontramos com frequência problemas reais, por exemplo, problemas de gestão operacional/estratégica, de planeamento de tráfego ou de mercados de energia, onde uma ou mais destas características está presente. Procurando responder ao desafio de

encontrar soluções para estes problemas, muitos estudos têm sido encetados, em particular usando abordagens meta-heurísticas para determinar a solução ótima ou uma solução aproximada do problema.

Os problemas BL têm frequentemente natureza multiobjetivo, podendo conter um problema de otimização multiobjetivo num dos níveis ou em ambos. No caso de termos um problema multiobjetivo no nível inferior, a existência de múltiplas soluções ótimas de Pareto neste nível torna a análise do problema de otimização BL particularmente difícil. Note-se que uma solução para ser admissível do problema BL tem de ser ótima do problema de nível inferior (ótima de Pareto no caso multiobjetivo). Para além de o problema multiobjetivo de nível inferior ser mais difícil de resolver ao envolver a caracterização do conjunto de soluções ótimas de Pareto, a dificuldade por parte do líder em antecipar a escolha de uma destas soluções pelo seguidor torna o problema ainda mais complexo. Todos estes aspetos motivam o desenvolvimento de meta-heurísticas eficazes para a resolução de problemas BL, quer para o caso mono-objetivo, quer para os casos multiobjetivo.

Os problemas de otimização associados à interação entre as empresas comercializadoras de eletricidade (retalhista) e os consumidores constituem uma área emergente de aplicação em que os modelos BL são particularmente adequados para a representação do problema. No mercado retalhista de eletricidade, o retalhista e o consumidor têm uma relação hierárquica. O retalhista determina os preços da energia e o consumidor calendariza os seus consumos, ajustando-os de acordo com os preços de energia definidos pelo retalhista. Assim, o retalhista é o líder e o consumidor é o seguidor.

O abastecimento de energia elétrica baseado no modelo atual, onde a oferta segue a procura, enfrenta grandes desafios num ambiente de aumento do consumo global, escassez de recursos energéticos, necessidades de investimento avultado em capacidade de geração e infraestruturas de redes, e incerteza resultante da produção de energia por fontes renováveis cuja penetração na matriz de geração tem vindo a aumentar. Neste contexto, a utilização de tarifas dinâmicas (*time-of-use*) representa um mecanismo que pode ser importante na modificação dos padrões de consumo permitindo contribuir para adequar a procura à oferta e às condições da rede. Os recentes avanços tecnológicos permitem a construção de redes inteligentes que possibilitam uma comunicação bi-direcional entre retalhista e consumidor. O retalhista pode enviar para o consumidor os preços a praticar num determinado período futuro e, conhecendo os preços, o consumidor escolhe os períodos de funcionamento dos seus aparelhos de modo a minimizar a fatura

de eletricidade e/ou minimizar o desconforto associado à deslocação temporal do funcionamento dos aparelhos em relação aos períodos habituais. As escolhas do consumidor afetam também o lucro do retalhista pelo que este deve ter em conta a reação dos consumidores na definição das tarifas de eletricidade. A construção de novos modelos que estabeleçam a resposta do consumidor aos preços impostos pelo retalhista reveste-se da maior importância e existem ainda poucos estudos que modelem esta interação para obter uma caracterização realista do problema. Quer a construção dos modelos, quer o desenvolvimento de algoritmos eficientes para a sua resolução constituem um importante contributo para novas formas de funcionamento do mercado retalhista de energia elétrica. Neste sentido pretendemos incluir no nosso trabalho contribuições relativas ao desenvolvimento de modelos matemáticos BL para estudar a interação entre o retalhista e o consumidor de energia elétrica bem como as abordagens algorítmicas para tratar estes modelos.

1.2 Objetivos

Nas últimas décadas têm sido desenvolvidas diversas meta-heurísticas para resolução de problemas de otimização multiobjetivo. Na otimização multiobjetivo pretende-se, em geral, encontrar um conjunto de soluções que representem uma boa caracterização da frente ótima de Pareto do problema. Assim, um algoritmo para resolver este problema deve reunir várias características, conseguindo um equilíbrio entre a convergência para a frente ótima de Pareto e a dispersão das soluções encontradas sobre esta frente, de modo a oferecer soluções com diferentes compromissos (*trade-offs*) entre as funções objetivo. Notamos que diferentes características de um problema exigem capacidades específicas do algoritmo para a sua resolução, de forma a obter uma boa caracterização da frente ótima de Pareto do problema. Tendo em conta o bom comportamento da meta-heurística EM em problemas mono-objetivo e tendo sido muito pouco explorada em problemas multiobjetivo, pretendemos estudar e melhorar o desempenho da meta-heurística EM na resolução de problemas multiobjectivo com variáveis contínuas. Os resultados obtidos pelo novo algoritmo em problemas *benchmark* são comparados com os resultados de algoritmos representativos do estado da arte de outras meta-heurísticas populacionais, tais como PSO, DE e EA. Escolhemos estas meta-heurísticas para comparação por serem, em geral, adequadas para tratarem problemas com variáveis contínuas (total ou parcialmente). Posteriormente, são considerados os

algoritmos com melhor desempenho para desenvolvermos um algoritmo BL multiobjetivo.

Tendo em conta as características já anteriormente descritas dos problemas BL, existe um vasto campo de aplicação das meta-heurísticas, nomeadamente das abordagens populacionais. Recentemente, foram desenvolvidas algumas abordagens baseadas em EA para resolução de problemas BL multiobjectivo (Sinha & Deb, 2009) (Sinha et al., 2016). Tendo em conta o desempenho das meta-heurísticas populacionais multiobjectivo observado nos problemas *benchmark* acima referidos (com um único nível), em que os EA e abordagens PSO superaram as abordagens EM e DE, e existindo abordagens evolucionárias para problemas BL multiobjectivo, desenvolvemos um novo algoritmo BL multiobjetivo em ambos os níveis baseado em PSO.

As aplicações dos modelos de otimização BL são relevantes em diversas áreas e um dos campos de aplicação destes modelos é o setor da energia. Na gestão da procura da eletricidade têm sido propostos modelos utilizando esquemas de preços variáveis para induzir a deslocação dos picos de consumo e suavizar a distribuição temporal do funcionamento das cargas. Estes modelos limitam-se frequentemente a apresentar o consumo agregado total (Samadi et al., 2012) ou de cada equipamento (Mohsenian-Rad & Leon-Garcia, 2010), não caracterizando o perfil de consumo de cada equipamento, nomeadamente no que se refere aos consumidores residenciais. Por outro lado, apenas alguns estudos consideram a interação retalhista – consumidor, na qual há que ter em conta os objetivos de cada decisor e as restrições que devem ser observadas de forma a obter soluções realistas. Assim, desenvolvemos um modelo de otimização BL para representar a interação entre o retalhista e os consumidores associada à definição de tarifas dinâmicas. Estabelecemos um enquadramento realista para o problema, considerando os equipamentos residenciais típicos e os respetivos diagramas de consumo, bem como restrições relativas a ambos os intervenientes. No caso do retalhista, os preços da energia devem estar contidos em intervalos pré-definidos e consideramos que devem obedecer a um preço médio pré-definido num dado período de planeamento, para incorporar no modelo um elemento que represente de forma aproximada a concorrência no mercado liberalizado. Por outro lado, a calendarização do consumo dos equipamentos do consumidor tem de respeitar a potência disponível em cada período e os intervalos definidos pelo consumidor para a utilização dos equipamentos. Vamos considerar dois modelos BL deste problema: o modelo mono-objetivo em ambos os níveis, sendo o objetivo do retalhista a maximização do lucro e o objetivo do consumidor a minimização

da sua conta de eletricidade; o modelo mono-objetivo no nível superior e multiobjetivo no nível inferior, tendo o retalhista o objetivo de maximizar o lucro e o consumidor os objetivos de minimizar a sua fatura de eletricidade e minimizar o desconforto resultante da deslocação temporal do funcionamento dos seus equipamentos. Começamos por considerar o modelo mono-objetivo em ambos os níveis pelo facto de, neste caso, ser possível obter a solução exata do problema do nível inferior, transformando-o num problema linear inteiro misto (*mixed integer linear programming* - MILP). Assim, o problema BL mono-objetivo em ambos os níveis é resolvido de duas formas distintas: 1) recorrendo apenas a meta-heurísticas populacionais; 2) utilizando uma meta-heurística para resolver o problema do nível superior e resolvendo o problema do nível inferior de forma exata através de técnicas de programação matemática usando um *solver*. Deste modo, é possível avaliar a capacidade das meta-heurísticas populacionais para resolver o problema BL. Para resolver cada um dos modelos BL considerados recorrendo apenas a meta-heurísticas, são efetuadas as necessárias adaptações nos algoritmos BL desenvolvidos e apresentados anteriormente.

Apresentamos de seguida as principais contribuições desta tese:

- desenvolvimento de um algoritmo baseado em EM para resolução de problemas multiobjetivo – *enhanced multi-objective electromagnetism-like mechanism* (EMOEM);
- desenvolvimento de um algoritmo baseado em PSO para resolução de problemas de otimização em dois níveis com múltiplas funções objetivo em ambos os níveis - *multi-objective bi-level particle swarm optimization* (MO-BLPSO);
- construção de um modelo de otimização em dois níveis, mono-objetivo em ambos os níveis, para o problema da interação retalhista-consumidor no mercado de energia elétrica;
- desenvolvimento de um algoritmo híbrido que utiliza operadores genéticos para resolver o problema do nível superior e um *solver* para resolução do problema do nível inferior – *hybrid bi-level evolutionary algorithm* (H-BLEA);
- desenvolvimento de algoritmos populacionais baseados em PSO e em EA para resolução do problema da interação retalhista-consumidor no mercado de energia elétrica – *bi-level particle swarm optimization* e *bi-level evolutionary algorithm* (BLPSO e BLEA);

- construção de um modelo em dois níveis semi-vetorial (*bi-level semi-vectorial* - BLSV) obtido a partir do modelo mono-objetivo anterior, adicionando uma nova função objetivo no nível inferior relativa ao desconforto sentido pelo consumidor pela deslocação temporal do funcionamento dos seus equipamentos;
- desenvolvimento de dois algoritmos populacionais, um baseado em PSO (BL-MOPSO) e outro baseado em EA (BL-MOEA), para resolução do problema BLSV da interação retalhista-consumidor do mercado de energia elétrica e obtenção da solução do problema segundo duas perspectivas, otimista e pessimista.

1.3 Estrutura da tese

No capítulo 2 começamos por apresentar uma revisão da literatura relativa às meta-heurísticas populacionais multiobjetivo. Neste âmbito descrevemos as principais meta-heurísticas populacionais existentes na literatura. Descrevemos as principais características das abordagens EA, PSO, DE e EM, e analisamos os algoritmos multiobjetivo mais relevantes dentro de cada uma destas meta-heurísticas. Apresentamos de seguida o algoritmo EMOEM que desenvolvemos baseado na meta-heurística EM. O desempenho deste algoritmo é avaliado recorrendo a um conjunto de problemas *benchmark* existentes na literatura. Este algoritmo é também comparado com algoritmos de referência das meta-heurísticas descritas anteriormente.

No capítulo 3 apresentamos um novo algoritmo BL multiobjetivo em ambos os níveis, baseado na abordagem PSO, para resolução de problemas BL multiobjetivo. Este algoritmo é comparado com um algoritmo BL baseado em EA anteriormente proposto na literatura.

Na sequência desta investigação, no capítulo 4 desenvolvemos um modelo de otimização em dois níveis, captando a resposta do consumidor à política de preços estabelecida pelo retalhista, para o problema da interação retalhista-consumidor no mercado de eletricidade no setor residencial. Apresentamos também os novos algoritmos BL desenvolvidos para resolução deste problema e os respetivos resultados.

No capítulo 5 apresentamos uma extensão do modelo formulado no capítulo 4 considerando dois objetivos do consumidor: a minimização do custo e a minimização do possível desconforto associado à deslocação temporal dos seus consumos em relação aos períodos habituais de funcionamento das cargas. Assim, este modelo contém, no nível inferior, um problema de otimização com duas funções objetivo.

Por último apresentamos no capítulo 6 as conclusões da investigação efetuada e as principais direções de investigação futura.

2 Meta-heurísticas populacionais multiobjetivo

2.1 Introdução

A importância dos modelos multiobjetivo em aplicações práticas e as dificuldades que surgem na sua resolução têm reforçado a necessidade de investigação no domínio dos algoritmos multiobjetivo. Em geral, os problemas multiobjetivo têm um extenso conjunto de soluções não dominadas que podem ser difíceis de calcular. As meta-heurísticas populacionais são particularmente adequadas para a resolução de problemas multiobjetivo tendo em vista o cálculo de um conjunto de soluções não dominadas representativas de toda a frente ótima de Pareto. Para além dos algoritmos evolucionários multiobjetivo (MOEA), dos quais o mais utilizado talvez seja o algoritmo NSGA-II (Deb et al., 2002), têm sido propostas outras meta-heurísticas multiobjetivo baseadas em populações, tais como a otimização por enxame de partículas (MOPSO), a evolução diferencial (MODE) e o mecanismo eletromagnético (MOEM). Estas meta-heurísticas são representativas de diferentes estratégias e apropriadas para pesquisa em espaços contínuos; por esta razão foram selecionadas para uma análise comparativa dos respetivos desempenhos.

A Otimização por Enxame de Partículas (PSO) é inspirada no comportamento de algumas espécies, tal como o voo de pássaros quando procuram comida (Kennedy & Eberhart, 1995). Os membros da população movem-se no espaço de pesquisa de acordo com a sua própria experiência e a experiência dos seus vizinhos. Na abordagem PSO é usual referir os indivíduos como partículas. Muitos investigadores têm proposto modificações ao algoritmo original incluindo a extensão à resolução de problemas de otimização multiobjetivo. Muitas destas abordagens baseiam-se no conceito de dominância de Pareto. Entre estas abordagens, o algoritmo OMOPSO (Reyes-Sierra & Coello, 2005) tem sido considerado um dos mais competitivos (Durillo et al., 2009).

Na Evolução Diferencial (DE) (Storn & Price, 1995) os membros da população evoluem através de um mecanismo baseado em diferenças de vetores. Apesar da sua simplicidade, este mecanismo provou ser bastante eficaz (Price, 1997), (Swagatam &

Suganthan, 2011). Na última década, têm sido desenvolvidas diversas versões do algoritmo DE multiobjetivo (MODE). Alguns destes algoritmos adotam mecanismos dos algoritmos evolucionários multiobjetivo, designadamente ordenação por frentes de dominância (Deb et al., 2002) e técnicas de preservação de diversidade baseadas numa medida de concentração populacional (*crowding*). Em Mezura-Montes et al. (2008), é apresentada uma revisão do estado da arte dos algoritmos MODE. Em Wang et al. (2010) é apresentado o algoritmo MOSADE que utiliza um arquivo para guardar as soluções não dominadas, considera parâmetros adaptativos e utiliza também uma medida de *crowding*. Os autores comparam o algoritmo MOSADE com outros algoritmos MODE e evolucionários representativos do estado da arte, tendo o algoritmo MOSADE apresentado os melhores resultados nas experiências realizadas.

O mecanismo eletromagnético (EM) é uma meta-heurística introduzida por Birbil e Fang (2003), tendo sido utilizada na resolução de diversos problemas de otimização contínua mono-objetivo. Esta abordagem é inspirada no mecanismo de atração-repulsão da teoria do eletromagnetismo. Após inicialização, o algoritmo EM é composto por três passos principais: determinação do vetor força individual, movimentação dos indivíduos da população e procedimento de pesquisa local. O vetor força individual e o movimento das partículas da população são influenciados por todos os outros membros da população. No contexto da otimização mono-objetivo, têm sido propostas várias concretizações destes componentes. O procedimento de pesquisa local é considerado um dos componentes decisivos para o desempenho do algoritmo EM. Ao ser aplicado o procedimento de pesquisa local a uma partícula selecionada não é utilizada informação acerca das outras partículas. O desempenho do algoritmo EM está fortemente dependente da configuração dos seus componentes.

O algoritmo EM foi adaptado para resolver problemas multiobjetivo por Tsou e Kao (2006) e posteriormente foi também utilizado em Tsou e Kao (2008) e em Tsou et al. (2008) para resolver um problema de inventário. Tal como no caso mono-objetivo, o algoritmo MOEM baseia-se na determinação do vetor força individual, no movimento das partículas da população e no procedimento de pesquisa local. Com o objetivo de melhorar o desempenho do algoritmo MOEM, nesta tese desenvolvemos um algoritmo MOEM melhorado (EMOEM) (Carrasqueira et al., 2014, 2015a) no qual foram modificados o vetor força individual, a atualização da posição de cada partícula e o procedimento de pesquisa local, para ultrapassar algumas dificuldades do algoritmo MOEM relativas a convergência para a frente ótima de Pareto do problema. A estratégia de pesquisa local

melhorada é inspirada no algoritmo de Hooke e Jeeves (1961), que tem sido aplicado com muito sucesso tanto na otimização mono-objetivo (Rocha & Fernandes, 2009) como multiobjetivo (Mousa et al., 2012).

Após a introdução da notação e conceitos básicos de otimização multiobjetivo na secção 2.2, os algoritmos NSGA-II, OMOPSO, MOSADE e MOEM, representativos do estado da arte das respectivas classes de meta-heurísticas, são brevemente descritos na secção 2.3. O algoritmo EMOEM por nós desenvolvido é apresentado detalhadamente na secção 2.4. Na secção 2.5 apresentamos resultados computacionais e é feita uma comparação dos algoritmos.

2.2 Otimização multiobjetivo

Um problema de otimização multiobjetivo pode ser definido como

$$\begin{aligned} \text{Min } f(x) &= (f_1(x), f_2(x), \dots, f_m(x)) \\ \text{s.a } x &\in \Omega, \\ \Omega &= \{x \in \mathfrak{R}^d \mid g_i(x) \leq 0, i = 1, 2, \dots, p\} \end{aligned}$$

onde $f(x)$ é o vetor das funções objetivo a otimizar, $x = (x_1, x_2, \dots, x_d)$ é o vetor das variáveis de decisão e $g_i(x) \leq 0$ são as restrições. Alguns problemas consideram apenas restrições de limites inferior e superior nas variáveis. Nesse caso as restrições são dadas por $l_j \leq x_j \leq u_j, j = 1, \dots, d$, sendo l_j e u_j os limites inferior e superior da variável x_j , respetivamente.

Definição 1. Dominância - Um vetor $x \in \Omega$ domina um vetor $y \in \Omega$ e dizemos $x \prec y$, se e só se $f_k(x) \leq f_k(y), \forall k = 1, \dots, m$ e $\exists j \in \{1, \dots, m\}: f_j(x) < f_j(y)$.

Definição 2. Uma solução $x \in \Omega$ é ótima de Pareto (ou eficiente) se e só se $\nexists y \in \Omega: y \prec x$. O ponto correspondente no espaço das funções objetivo, $f(x)$, diz-se não dominado.

No caso de um vetor x não dominar um vetor y ($x \not\prec y$) e de o vetor y também não dominar o vetor x ($y \not\prec x$), os vetores x e y dizem-se mutuamente não dominados e representa-se esta relação por $x \not\prec y$.

O conjunto de todos os pontos no espaço das funções objetivo correspondentes às soluções ótimas de Pareto é designado por frente ótima de Pareto. O nosso propósito é obter uma representação da totalidade da frente ótima de Pareto do problema. As soluções obtidas pelas meta-heurísticas são soluções potencialmente ótimas de Pareto / eficientes / não dominadas, porque não é possível garantir teoricamente que sejam de facto soluções ótimas de Pareto, ao contrário do que acontece usando abordagens de programação matemática (algoritmos exatos) em determinados problemas. Ou seja, são consideradas soluções não dominadas desde que não sejam encontradas outras soluções que as dominem e, por simplicidade de linguagem, serão assim designadas ao longo do texto.

2.3 Descrição de algoritmos representativos do estado da arte MOEA, MOPSO, MODE e MOEM

2.3.1 Algoritmo NSGA-II

NSGA-II (Deb et al. 2002) é um algoritmo evolucionário multiobjetivo representativo do estado da arte da sua classe. Este algoritmo utiliza os operadores genéticos cruzamento, mutação e seleção para gerar sucessivas populações de soluções. Neste algoritmo é introduzido um mecanismo de ordenação para classificar os membros da população (indivíduos) por *frentes de não dominância*. Após cada iteração (geração) os elementos da população são ordenados por não dominância (Fig. 2.1). Os elementos não dominados relativamente a toda a população são selecionados primeiro e colocados na primeira frente ($ND = 1$). Entre os restantes elementos, identificam-se os não dominados os quais são colocados na segunda frente ($ND = 2$) e assim sucessivamente até serem selecionados todos os indivíduos da população. Cada membro da população é ainda classificado de acordo com uma medida de distância de *crowding* que será utilizada para ordenar os indivíduos em cada frente por ordem decrescente desta medida. A distância de *crowding* de um indivíduo representa a distância média deste relativamente aos seus vizinhos. A distância de *crowding* de um indivíduo é tanto maior quanto mais afastado estiver dos seus vizinhos.

Em cada geração o algoritmo começa por criar uma nova população de indivíduos utilizando os operadores genéticos. Juntam-se as populações antiga e nova, e a população resultante é classificada e ordenada por frentes de não dominância e, em cada frente, por

distância de *crowding*. De seguida é selecionada a população para a nova geração, começando pela primeira frente de não dominância e terminando quando estiverem selecionados todos os elementos da nova população. No caso de serem necessários apenas alguns indivíduos de uma frente de não dominância, a sua seleção é efetuada por ordem decrescente da sua medida de distância de *crowding*.

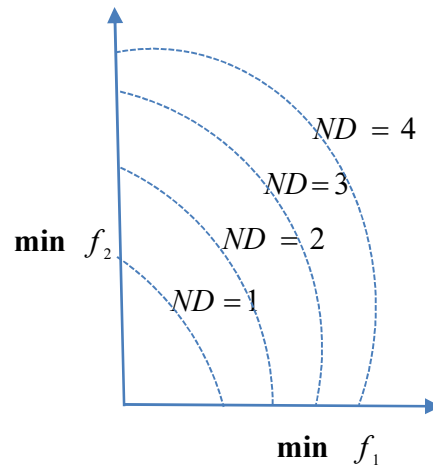


Fig. 2.1 Representação de frentes de não dominância

O algoritmo NSGA-II inclui um mecanismo para tratamento de restrições. Para cada solução, este processo começa por avaliar as funções relativas às restrições $g_i(x)$ e, nas que não forem satisfeitas, é calculado o valor da violação. Os valores correspondentes à violação de restrições de uma solução são agregados num único valor resultante da sua soma. As soluções não admissíveis são ordenadas por ordem crescente dos seus valores de violação das restrições. O pseudo-código do algoritmo NSGA-II é apresentação no Algoritmo 2.1.

Algoritmo 2.1.

Algoritmo NSGA-II – pseudo-código

Inicializa o contador de iterações, $k = 1$

Inicializa aleatoriamente cada indivíduo x_i^k da população Pop , $i = 1, \dots, n$

Avalia cada indivíduo x_i^k , calculando $f(x_i^k) = (f_1(x_i^k), f_2(x_i^k), \dots, f_m(x_i^k))$, $i = 1, \dots, n$

Inicializa $NewPop = \emptyset$

Enquanto o critério de paragem não é atingido

Para todos os indivíduos $i, i = 1, \dots, n$

 Seleciona aleatoriamente dois indivíduos p_1 e p_2 de Pop

$x_i^{k+1} \leftarrow$ cruzamento de p_1 e p_2

$x_i^{k+1} \leftarrow$ mutação do indivíduo x_i^{k+1}

 Avalia o indivíduo x_i^{k+1} , calculando $f(x_i^{k+1})$

$NewPop \leftarrow NewPop \cup x_i^{k+1}$

Termina Para

$Rpop \leftarrow Pop \cup NewPop$

 Determina uma ordem de não dominância (ND) e a distância *crowding* (CD) de cada indivíduo de $Rpop$

 Ordena $Rpop$ por ordem crescente de ND e decrescente de CD , lexicograficamente

$Pop \leftarrow$ primeiros n indivíduos de $Rpop$

Termina Enquanto

Devolve Pop

2.3.2 Algoritmo OMOPSO

O algoritmo PSO é geralmente inicializado gerando aleatoriamente uma população de soluções (partículas). Cada partícula tem duas componentes, a sua posição no espaço das soluções, e a sua velocidade, que indica a direção da movimentação da partícula, resultante da combinação das direções da melhor posição individual alcançada até ao momento pela partícula ($pbest_i$) e da melhor posição alcançada por toda a população ($gbest$). Em otimização multiobjetivo $pbest_i$ e $gbest$ não são únicos. Habitualmente, é escolhido um $gbest_i$ para cada partícula i de entre as soluções não dominadas. Cada partícula i move-se na iteração k de acordo com as expressões

$$v_i^{k+1} = wv_i^k + c_1 \times rand() \times (pbest_i - x_i^k) + c_2 \times rand() \times (gbest_i - x_i^k) \quad (2.1)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1}, i = 1, 2, \dots, n \quad (2.2)$$

onde w , c_1 e c_2 são parâmetros do algoritmo que representam o fator de inércia (w), a influência cognitiva (c_1) e a influência social (c_2), $rand(\)$ é um valor aleatório uniforme no intervalo $[0,1]$ e n é o número de partículas na população. A forma como os parâmetros w , c_1 e c_2 são definidos durante a execução do algoritmo depende da versão do algoritmo MOPSO. No caso da versão OMOPSO (Sierra & Coello, 2005), estes parâmetros são escolhidos aleatoriamente em cada iteração dentro de um intervalo pré-definido: $w \in [0.1,0.5]$ e $c_1, c_2 \in [1.5,2]$. Outras características do algoritmo OMOPSO são as seguintes:

- Uma nova posição x_i de uma partícula substitui o seu $pbest_i$ atual se $pbest_i$ é dominado por x_i ou se são não dominados entre si.
- Partículas não dominadas são guardadas num arquivo externo. Este arquivo tem uma dimensão máxima pré-definida. Quando a capacidade do arquivo é atingida, uma partícula apenas entra no arquivo após ser eliminada outra do arquivo.
- A cada partícula do arquivo externo é atribuída uma medida de distância de *crowding* (Deb et al., 2002). Esta medida é utilizada para selecionar o líder ($gbest_i$) de cada partícula da população e para selecionar uma partícula do arquivo externo para ser substituída quando o arquivo está cheio. Na seleção do líder, é efetuado um torneio binário, que consiste na seleção aleatória de duas partículas do arquivo para competirem entre si, sendo escolhida a partícula que tiver maior valor de distância de *crowding*. Para retirar uma partícula do arquivo selecionamos a partícula que tiver o menor valor de distância de *crowding*, uma vez que significa que está mais próxima dos seus vizinhos.
- É utilizado um operador de mutação designado turbulência, com uma certa probabilidade. Esta operação é efetuada após a execução das operações (2.1) e (2.2). A população é dividida em três partes. A um terço é aplicada a mutação uniforme, a outro terço é aplicada uma mutação não-uniforme e o restante terço da população não é alterado.

O algoritmo OMOPSO foi comparado com outros algoritmos MOPSO e MOEA em Durillo et al. (2009) e devido ao seu bom desempenho o algoritmo OMOPSO tem sido considerado como representativo do estado da arte das abordagens MOPSO. O pseudo-código do algoritmo OMOPSO é apresentado no Algoritmo 2.2.

Algoritmo 2.2.

Algoritmo OMOPSO – pseudo-código

Inicializa o contador de iterações, $k = 1$

Inicializa aleatoriamente cada partícula x_i^k da população e a sua velocidade

$$v_i^k, i = 1, \dots, n$$

Avalia cada partícula x_i^k , calculando $f(x_i^k) = (f_1(x_i^k), f_2(x_i^k), \dots, f_m(x_i^k))$, $i = 1, \dots, n$

Inicializa $pbest_i$ com a posição atual da partícula x_i^k , $i = 1, \dots, n$

Insera as partículas não dominadas no arquivo externo

Enquanto o critério de paragem não é atingido

Para todas as partículas $i, i = 1, \dots, n$

 Seleciona o líder, $gbest_i$ do arquivo externo

 Atualiza a velocidade da partícula v_i^{k+1} , usando (2.1)

 Atualiza a posição da partícula x_i^{k+1} , usando (2.2)

 Efetua a mutação da partícula x_i^{k+1} , aplicando o operador turbulência

 Avalia a partícula x_i^{k+1} , calculando $f(x_i^{k+1})$

 Atualiza $pbest_i$ da partícula i

Termina Para

 Atualiza o arquivo externo de soluções não dominadas

Termina Enquanto

Devolve o arquivo de soluções não dominadas

2.3.3 Algoritmo MOSADE

Na Evolução Diferencial (DE) a população evolui com base em três mecanismos: mutação, recombinação e seleção. Existem várias formas de definir estas operações. Vamos considerar a parametrização DE/rand/1/bin. Esta notação significa que, no operador de mutação, cada solução base ($x_{r_1}^k$ em 2.3) é escolhida aleatoriamente, é utilizado apenas um vetor diferença ($(x_{r_2}^k - x_{r_3}^k)$ em 2.3) e o operador de recombinação utiliza a distribuição binomial. A recombinação discreta com probabilidade CR utilizada em (2.4) faz com que o número de valores herdados de v_i^{k+1} siga uma distribuição binomial.

Em cada iteração k e para cada solução i , o algoritmo começa por aplicar um operador mutação. Para efetuar esta operação, são escolhidas aleatoriamente três

partículas da população, $x_{r_1}^k, x_{r_2}^k$ e $x_{r_3}^k$. Com as partículas escolhidas é construída uma nova solução, que se designa por vetor dador:

$$v_i^{k+1} = x_{r_1}^k + F \times (x_{r_2}^k - x_{r_3}^k) \quad (2.3)$$

onde $F > 0$ é o parâmetro de mutação. Este parâmetro controla a extensão do movimento efetuado. Baixos valores de F favorecem a exploração local e valores elevados favorecem a exploração de novas regiões. De seguida é aplicado o operador recombinação. O vetor teste u_i^{k+1} é obtido a partir dos elementos do vetor x_i^k e do vetor dador v_i^{k+1} :

$$u_{ij}^{k+1} = \begin{cases} v_{ij}^{k+1}, & \text{se } rand(j) \leq CR \text{ ou } j = randint(i) \\ x_{ij}^k, & \text{caso contrário} \end{cases}, j = 1, \dots, d \quad (2.4)$$

onde $rand(j) \sim U[0,1]$, CR é a probabilidade de recombinação e $randint(i)$ é um valor aleatório inteiro entre 1 e d . A consideração de $j = randint(i)$ em (2.4) tem como intenção obrigar a que o vetor u_i^{k+1} tenha pelo menos um elemento diferente do vetor original x_i^k .

Para decidir qual a solução que irá sobreviver para a próxima geração é efetuada uma operação de seleção. No algoritmo MOSADE (Wang et al., 2010), a operação de seleção baseia-se na relação de dominância entre as duas soluções e, no caso de serem mutuamente não dominadas, é selecionada a solução com maior valor de distância de *crowding*. Esta operação é dada pela expressão

$$x_i^{k+1} = \begin{cases} x_i^k, & \text{se } x_i^k \prec u_i^{k+1} \\ u_i^{k+1}, & \text{se } u_i^{k+1} \prec x_i^k, i = 1, \dots, n \\ LC(u_i^{k+1}, x_i^k), & \text{se } u_i^{k+1} \nprec x_i^k \end{cases} \quad (2.5)$$

onde $LC(u_i^{k+1}, x_i^k)$ significa a solução x_i^k ou u_i^{k+1} com maior valor de distância de *crowding*, se nenhuma delas dominar a outra.

No algoritmo MOSADE, as partículas não dominadas encontradas são armazenadas num arquivo externo. As soluções neste arquivo são distinguidas através de um mecanismo de preservação de diversidade. Os parâmetros F e CR são definidos de forma independente para cada partícula e podem ser modificados de forma adaptativa ao longo da execução do algoritmo. Isto significa que, se uma solução ainda não foi melhorada durante um certo número de iterações, os parâmetros F e CR atribuídos a essa

solução são substituídos por valores gerados num intervalo pré-definido. Os autores argumentam que o algoritmo DE é muito sensível aos valores dos parâmetros e com esta estratégia os parâmetros não necessitam de ser afinados. O algoritmo MOSADE foi também desenvolvido para resolver problemas com restrições, em que é adotado o mecanismo para tratamento de restrições apresentado em Deb et al. (2002). Este mecanismo é baseado no conceito de não dominância e no valor total de violação das restrições, para cada solução. Esta abordagem foi testada em problemas *benchmark* (Zitzler et al., 2000) (Deb et al., 2005) e obteve resultados competitivos. O pseudo-código do algoritmo MOSADE é apresentado no Algoritmo 2.3.

Algoritmo 2.3

Algoritmo MOSADE – pseudo-código

Inicializa o contador de iterações, $k = 1$

Inicializa aleatoriamente cada solução da população, $x_i^k, i = 1, \dots, n$

Avalia cada solução x_i^k , calculando $f(x_i^k) = (f_1(x_i^k), f_2(x_i^k), \dots, f_m(x_i^k)), i = 1, \dots, n$

Inicializa o arquivo externo de soluções não dominadas

Enquanto o critério de paragem não é atingido

Para todas as soluções $i, i = 1, \dots, n$

 Seleciona aleatoriamente $r_1 \neq r_2 \neq r_3 \in \{1, 2, \dots, n\}$

 Cria o vetor dador v_i^{k+1} usando (2.3)

 Cria o vetor u_i^{k+1} por recombinação de x_i^k e v_i^{k+1} , usando (2.4)

 Avalia u_i^{k+1} , calculando $f(u_i^{k+1})$

 Seleciona entre x_i^k e u_i^{k+1} para definir x_i^{k+1} , usando (2.5)

Termina Para

 Atualiza o arquivo de soluções não dominadas

Termina Enquanto

 Devolve o arquivo de soluções não dominadas

2.3.4 Algoritmo MOEM

Na abordagem EM cada solução é representada por uma partícula com uma determinada carga. A carga de uma partícula depende dos respetivos valores das funções objetivo. Esta abordagem é inspirada na lei de Coulomb, onde a qualidade das partículas é avaliada pelo valor das suas cargas e em que maior valor de carga representa mais qualidade. As partículas movem-se por um mecanismo de atração-repulsão. As partículas

com melhor avaliação atraem as outras e as partículas com pior avaliação repelem as restantes.

O algoritmo EM foi adaptado à resolução de problemas multiobjetivo (MOEM) por Tsou e Kao (2006). O algoritmo MOEM é baseado em três componentes: carga individual, força total e procedimento de pesquisa local. O algoritmo começa com uma população gerada aleatoriamente. Em seguida a população evolui por pesquisa local e por movimentação baseada no cálculo das cargas individuais e das forças de atração / repulsão, até que um critério de paragem pré-definido seja satisfeito.

Para efetuar a pesquisa local, construímos um arquivo local S , constituído por um número s pré-definido de soluções selecionadas aleatoriamente do arquivo de soluções não dominadas da população. Caso este arquivo tenha uma dimensão inferior a s , os dois arquivos coincidem. O processo de pesquisa local é aplicado a cada partícula do arquivo local S , aplicando uma perturbação a cada variável da partícula. Para determinar o valor desta perturbação, começamos por considerar um parâmetro δ pré-definido. Obtemos o valor máximo possível para a perturbação de uma variável multiplicando o parâmetro δ pela maior amplitude entre todas as variáveis do problema, utilizando a expressão (2.6):

$$leng = \delta \times \max_{j=1, \dots, d} \{u_j - l_j\} \quad (2.6)$$

Cada variável da solução será então modificada por um valor aleatório no intervalo $[0, leng]$. Após todas as variáveis terem sido modificadas a partícula é avaliada. Se a nova partícula domina a anterior, esta é substituída. O processo é repetido um determinado número ($lsit$) de vezes para cada partícula em cada iteração.

A carga de uma partícula depende dos seus valores das funções objetivo e dos valores de todos os outros membros da população. No algoritmo MOEM, a carga de uma partícula i é dada por

$$q_i = \exp \left(-d \frac{\min_{x_p \in S} \|f(x_i) - f(x_p)\|}{\sum_{j=1}^n \min_{x_p \in S} \|f(x_j) - f(x_p)\|} \right), i = 1, \dots, n \quad (2.7)$$

onde S é o arquivo local.

A força exercida entre duas partículas é inversamente proporcional ao quadrado da sua distância e diretamente proporcional ao produto das suas cargas. A força individual que cada partícula x_r exerce noutra partícula x_i relembra este princípio sendo dada por (2.8):

$$F^{ir} = \begin{cases} (x_i - x_r) \frac{q_i q_r}{\|x_i - x_r\|^2}, & \text{se } x_i \prec x_r \\ (x_r - x_i) \frac{q_i q_r}{\|x_i - x_r\|^2}, & \text{caso contrário} \end{cases}, r = 1, \dots, n (r \neq i) \quad (2.8)$$

A força total exercida na partícula i é a soma das forças individuais:

$$F^i = \sum_{\substack{r=1 \\ r \neq i}}^n F^{ir} \quad (2.9)$$

Após obter o vetor força total, é efetuado o movimento de cada partícula i de acordo com a expressão (2.10)

$$x_{ij} = \begin{cases} x_{ij} + \lambda \frac{F_j^i}{\|F^i\|} (u_j - x_{ij}), & \text{se } F_j^i > 0 \\ x_{ij} + \lambda \frac{F_j^i}{\|F^i\|} (x_{ij} - l_j), & \text{se } F_j^i \leq 0 \end{cases}, j = 1, \dots, d \quad (2.10)$$

onde λ é um valor aleatório tal que $\lambda \sim U[0,1]$ e l_j, u_j são os limites superior e inferior de cada componente j da partícula x_i , respetivamente, impostos pelas restrições do problema. F_j^i é a componente j da força total exercida na partícula i .

As partículas não dominadas obtidas durante a execução do algoritmo são armazenadas no arquivo externo. Em Tsou e Kao (2006) é utilizada a técnica de *clustering* proposta em Zitzler et al. (1999) para manter a diversidade do arquivo não dominado.

Os principais passos do algoritmo MOEM são descritos no Algoritmo 2.4 e no Algoritmo 2.5 são apresentados os detalhes do procedimento de pesquisa local.

Algoritmo 2.4

Algoritmo MOEM – pseudo-código

Inicializa o contador de iterações, $k = 1$

Inicializa aleatoriamente cada partícula da população, $x_i^k, i = 1, \dots, n$

Avalia cada partícula x_i^k , calculando $f(x_i^k) = (f_1(x_i^k), f_2(x_i^k), \dots, f_m(x_i^k)), i = 1, \dots, n$

Insera as partículas não dominadas no arquivo A

Enquanto o critério de paragem não é atingido

 Seleciona aleatoriamente as partículas do arquivo A a inserir no arquivo local S

 Efetua a pesquisa local em S

Para $i = 1, \dots, n$

 Calcula a carga (q_i) da partícula i , usando (2.7)

 Calcula a força total (F^i) exercida na partícula x_i , usando (2.8) e (2.9)

 Movimenta a partícula i , usando (2.10)

 Atualiza o arquivo não dominado A

Termina Para

$k \leftarrow k + 1$

Termina Enquanto

Devolve o arquivo não dominado A

Algoritmo 2.5.

Pesquisa Local (algoritmo MOEM) – pseudo-código

$$leng = \delta \times \max_{j=1, \dots, d} \{u_j - l_j\}$$

Para $i = 1, \dots, |S|$

Inicializa o contador de pesquisa local, $count = 0$

Enquanto $count < lsit$

$$z \leftarrow x_i$$

Para $j = 1, \dots, d$

Seleciona dois valores aleatórios $\lambda_1, \lambda_2 \in [0, 1]$

Se $\lambda_1 > 0.5$ **Então**

$$z_j \leftarrow z_j + \lambda_2 \times leng$$

Caso contrário

$$z_j \leftarrow z_j - \lambda_2 \times leng$$

Termina Se

Termina Para

Se $z \prec x_i$ **Então**

Atualiza A com z

$$x_i \leftarrow z$$

Termina Se

$$count \leftarrow count + 1$$

Termina Enquanto

Termina Para

2.4 Algoritmo EMOEM

2.4.1 Descrição

Com o intuito de melhorar o algoritmo MOEM desenvolvemos o algoritmo *Enhanced* MOEM (EMOEM) (Carrasqueira et al., 2014, 2015a) que modifica algumas das principais operações do algoritmo MOEM. No algoritmo MOEM o movimento de cada partícula (2.10) é influenciado pelo seu vetor força (2.8) – (2.9), que depende das cargas de todas as outras partículas da população (2.7). Tal como pode ser observado em (2.7), o cálculo da carga é efetuado utilizando um fator (d). A utilização deste fator permite acentuar as diferenças de desempenho entre as partículas. Em (2.7) o fator de ponderação utilizado é o número de variáveis (d). No entanto, as diferenças entre os

valores da carga de partículas muito afastadas entre si no espaço dos objetivos podem ser anuladas quando o rácio entre o número de variáveis e o tamanho da população é pequeno. No algoritmo EMOEM o número de variáveis é substituído pelo tamanho da população no fator utilizado para calcular a carga. Assim, no algoritmo EMOEM a carga é obtida utilizando a expressão (2.11).

$$q_i = \exp \left(-n \frac{\min_{x_p \in S} \|f(x_i) - f(x_p)\|}{\sum_{j=1}^n \min_{x_p \in S} \|f(x_j) - f(x_p)\|} \right), i = 1, \dots, n \quad (2.11)$$

Utilizando (2.11) acentuam-se as diferenças entre partículas com bom e mau desempenho, dado que as partículas fracas diminuem as suas cargas e as mais fortes aumentam-nas.

Adicionalmente, o movimento efetuado pela expressão (2.10) foi também modificado. No algoritmo MOEM, o movimento individual pode ser efetuado numa direção diferente da indicada pelo vetor força por duas razões. A primeira razão é que cada coordenada do vetor força é multiplicada pela correspondente coordenada de um vetor amplitude que determina a direção da pesquisa. Cada componente j deste vetor amplitude resulta da diferença entre x_{ij} e o limite inferior da variável j , ou da diferença entre o limite superior da variável j e x_{ij} , conforme a coordenada j do vetor força é negativa ou positiva, respetivamente. Dado que as coordenadas do vetor amplitude não são todas iguais, a direção do movimento é alterada. Na Fig. 2.2 está representado um exemplo desta situação, onde os vetores a tracejado representam a direção do vetor força. A mudança de direção apenas não se verifica quando a distância de x_i ao respetivo limite é a mesma em todas as dimensões. A segunda razão é que o limite escolhido (limite superior ou inferior) depende de a força ser positiva ou negativa. Há muitas situações em que as componentes individuais da força têm sinais diferentes. Nestes casos são utilizados diferentes vetores limite, o que desvia a partícula da direção indicada pelo seu vetor força. Resumindo, concluímos que em muitos casos a direção do movimento é diferente da direção definida pela força. Isto pode enviesar o movimento da partícula, não a conduzindo em direção às melhores soluções. Para ultrapassar este problema, foi incorporada uma modificação no mecanismo de atualização da posição do algoritmo EMOEM. Na nova expressão de atualização de posição, a componente relativa às coordenadas do vetor da amplitude é eliminada para garantir que o movimento efetuado

pela partícula segue a direção do seu vetor força. A Fig. 2.3 representa um exemplo do mecanismo de atualização de posição utilizado pelo algoritmo EMOEM, no qual cada partícula se move de acordo com a expressão (2.12)

$$x_{ij} = x_{ij} + \lambda \frac{F_j^i}{\|F^i\|}, j = 1, \dots, d \quad (2.12)$$

onde λ é um valor aleatório uniforme no intervalo $[0,1]$. Dado que o vetor força é normalizado, as variáveis devem ser consideradas no intervalo $[0,1]$. Para satisfazer este requisito é efetuada uma mudança de variável. Assim, antes de atualizar a posição da partícula, as suas variáveis são transformadas para valores do intervalo $[0,1]$ utilizando a expressão

$$x_{ij} \leftarrow \frac{x_{ij} - l_j}{u_j - l_j}, j = 1, \dots, d \quad (2.13)$$

onde l_j e u_j são os limites inferior e superior da variável j , respetivamente. Isto assegura que cada variável da partícula se mantém no intervalo $[0,1]$ e a partícula está pronta para ser atualizada. A direção do movimento não se altera pelo facto de a partícula ocupar uma determinada posição no espaço de pesquisa. Assim, o movimento efetuado pelo algoritmo EMOEM ultrapassa as situações de enviesamento identificadas no algoritmo MOEM e, em simultâneo, procura manter a admissibilidade das soluções. Em alguns casos, uma solução pode-se tornar não admissível mas nessa situação os valores das variáveis são movidos para o correspondente limite mais próximo. A implementação e teste do algoritmo EMOEM mostrou que este se comporta globalmente melhor do que o algoritmo MOEM, tendo aumentado a convergência e a diversidade na maioria dos problemas, como se poderá analisar nas experiências computacionais apresentadas na seção 2.5.4.

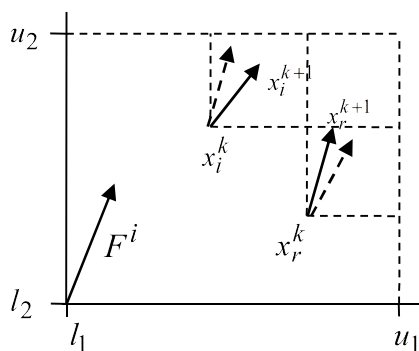


Fig. 2.2 atualização de posição – algoritmo MOEM

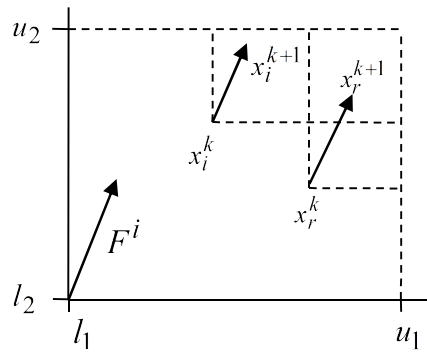


Fig. 2.3 atualização de posição – algoritmo EMOEM

Para melhorar o desempenho do algoritmo EMOEM propomos ainda uma outra modificação, que melhora o procedimento de pesquisa local. Os testes realizados mostraram que a nova versão supera a anterior. Descrevemos de seguida o novo procedimento de pesquisa local.

No mecanismo eletromagnético as cargas individuais e o vetor força aplicado a uma partícula são afetados por todas as outras partículas. Isto pode impedir que as partículas com melhor desempenho alcancem ainda melhores posições no espaço de pesquisa. Em contraste, a pesquisa local pode fornecer um processo de exploração no qual uma partícula selecionada não seja condicionada pelas outras partículas. Assim, o procedimento de pesquisa local é vital para o sucesso do algoritmo. A estratégia local do algoritmo MOEM apresentada no Algoritmo 2.5 perturba aleatoriamente cada variável de uma partícula num passo de comprimento pré-definido. Se a nova solução domina a atual, esta é substituída e a pesquisa termina; caso contrário, é efetuada uma nova pesquisa aleatória sobre a mesma partícula, que se repete até perfazer o número de iterações pré-definido para este procedimento. Esta situação é ilustrada na Fig. 2.4, onde o extremo de cada vetor indica uma solução alternativa para ser analisada em termos de dominância relativamente à solução x_i .

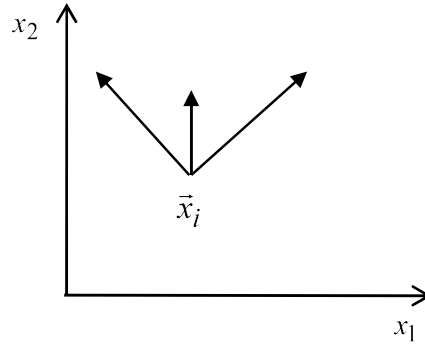


Fig. 2.4 pesquisa local – algoritmo MOEM

Analisando o comportamento do algoritmo MOEM, verificamos que é frequente o procedimento de pesquisa local produzir poucas novas soluções não dominadas. Pretendendo melhorar a capacidade do algoritmo MOEM para convergir para a frente ótima de Pareto, introduzimos um novo procedimento de pesquisa local inspirado no algoritmo de Hooke e Jeeves (1961) e na sua aplicação bem sucedida em outros algoritmos de otimização multiobjetivo (Mousa et al., 2012). A pesquisa local proposta pretende explorar direções de pesquisa promissoras. Considerando uma solução x_i selecionada do arquivo local, perturba-se uma variável de cada vez com uma certa probabilidade conduzindo a uma nova solução. Se esta solução temporária (seja z_i) for dominada por x_i , ou se forem não dominadas entre si, realiza-se uma nova perturbação. Caso contrário, x_i é substituída por z_i e efetua-se uma pesquisa a partir de z_i , seguindo a direção definida por x_i e z_i . Quando esta pesquisa produz uma solução y_i que domina z_i , y_i substitui z_i e a pesquisa prossegue.

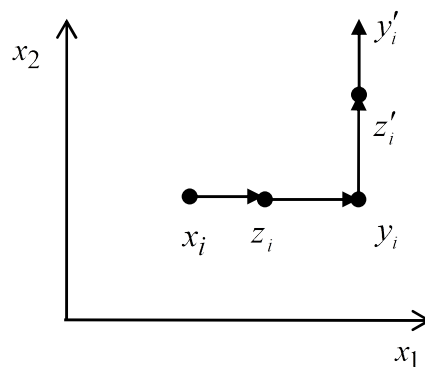


Fig. 2.5 pesquisa local – algoritmo EMOEM

Na Fig. 2.5 é ilustrado este processo. Considerando a pesquisa local a iniciar na solução x_i , estão indicadas as direções de pesquisa bem sucedidas. Modificando a variável x_{i1} em x_i , obtém-se z_i . Considerando que z_i domina x_i , a pesquisa continua nesta direção e y_i é retida considerando que domina z_i . De seguida, começando a partir de y_i e modificando a variável y_{i2} em y_i , obtém-se a solução z'_i . A pesquisa poderia ainda prosseguir para y'_i se z'_i dominasse y_i . Considerando estas duas variáveis da solução x_i , foram encontradas quatro soluções sucessivas não dominadas. Este exemplo ilustra o propósito desta estratégia, i.e. tirar partido de uma direção bem sucedida e prosseguir a pesquisa nessa direção. Após realizarmos algumas experiências computacionais, estabelecemos a probabilidade de perturbação de uma variável como $p_m = 1/3$, dado que foi este o valor que produziu melhores resultados. O pseudo-código desta pesquisa local é apresentado no Algoritmo 2.6.

Algoritmo 2.6.

Pesquisa Local (algoritmo EMOEM) – pseudo-código

Para $i = 1, \dots, |S|$

Para $j = 1, \dots, d$

$leng = \delta \times (u_j - l_j)$

 Seleciona um valor aleatório $\lambda_1 \in [0,1]$

Se $\lambda_1 < p_m$

 Inicializa o contador de pesquisa local, $count = 0$

Enquanto $count < lsit$

$trial = false$

 Seleciona um valor aleatório $\lambda_2 \in [0,1]$

$z_j \leftarrow x_{ij} + \lambda_2 \times leng$

Se $z \prec x_i$

$y_j \leftarrow z_j + \delta \times (z_j - x_{ij})$

$trial = true$ //pesquisa bem sucedida

Caso contrário

$z_j \leftarrow x_{ij} - \lambda_2 \times leng$ // nova perturbação a partir de x

Se $z \prec x_i$

$y_j \leftarrow z_j + \delta \times (z_j - x_{ij})$

$trial = true$ //pesquisa bem sucedida

Termina Se

Termina Se

Se $trial$ **Então** // se pesquisa bem sucedida, atualiza x_i

Se $y \prec z$

$x_i \leftarrow y$

Caso contrário

$x_i \leftarrow z$

Termina Se

$count \leftarrow lsit$ // pára a pesquisa

 Atualiza A com x_i

Caso contrário // pesquisa mal sucedida

$count \leftarrow count + 1$ // faz nova iteração

Termina Se

Termina Enquanto

Termina Se

Termina Para

Termina Para

2.5 Resultados experimentais

Vamos começar por definir as medidas de desempenho utilizadas para avaliar os algoritmos. Posteriormente serão apresentados os problemas teste e as parametrizações dos algoritmos. Em seguida são mostrados e analisados os resultados computacionais. Apresentamos ainda as considerações finais relativas ao trabalho que desenvolvemos neste domínio.

2.5.1 Medidas de desempenho

Os algoritmos multiobjetivo são avaliados em termos de convergência para a frente ótima de Pareto e relativamente à diversidade das soluções obtidas. Existem várias medidas para avaliar o desempenho de um algoritmo multiobjetivo, relativamente a um ou a ambos os eixos de avaliação. As medidas podem ainda ser unárias, se o seu cálculo requerer apenas um conjunto de soluções a avaliar, ou binárias, se forem necessários dois conjuntos de soluções aproximadas para efetuar o seu cálculo. Nas medidas de desempenho unárias, a cada conjunto de soluções a avaliar é atribuído um valor que representa um certo aspeto de qualidade. Nas medidas binárias, a cada par de conjuntos de dados a avaliar é atribuído um valor que indica a qualidade de um conjunto em relação ao outro.

Consideremos as medidas de desempenho hipervolume (HV) (Zitzler & Thiele, 1999) e Distância Geracional Invertida (IGD), ambas unárias. O hipervolume é calculado medindo o volume do espaço entre a frente não dominada e um ponto de referência, sendo geralmente considerado o ponto nadir, podendo também ser outro ponto que seja dominado por este. O ponto nadir (P) é obtido considerando em cada função objetivo o pior valor obtido em todas as soluções ótimas de Pareto. Maiores valores de HV indicam maior qualidade do conjunto solução obtido pelo algoritmo. A Fig. 2.6 ilustra o cálculo do hipervolume num problema de minimização com duas funções objetivo.

A Distância Geracional Invertida é a soma das distâncias de cada ponto da verdadeira frente ótima de Pareto ao ponto mais próximo do conjunto de soluções não dominadas encontrado pelo algoritmo. Ambos os indicadores medem a convergência e a diversidade do conjunto de soluções obtido. O cálculo do indicador IGD requer o conhecimento da verdadeira frente de Pareto do problema. Quanto menor o valor de IGD, melhor é a aproximação. Para avaliar o algoritmo EMOEM e compará-lo com os outros algoritmos, são utilizadas estas duas medidas de desempenho: HV e IGD.

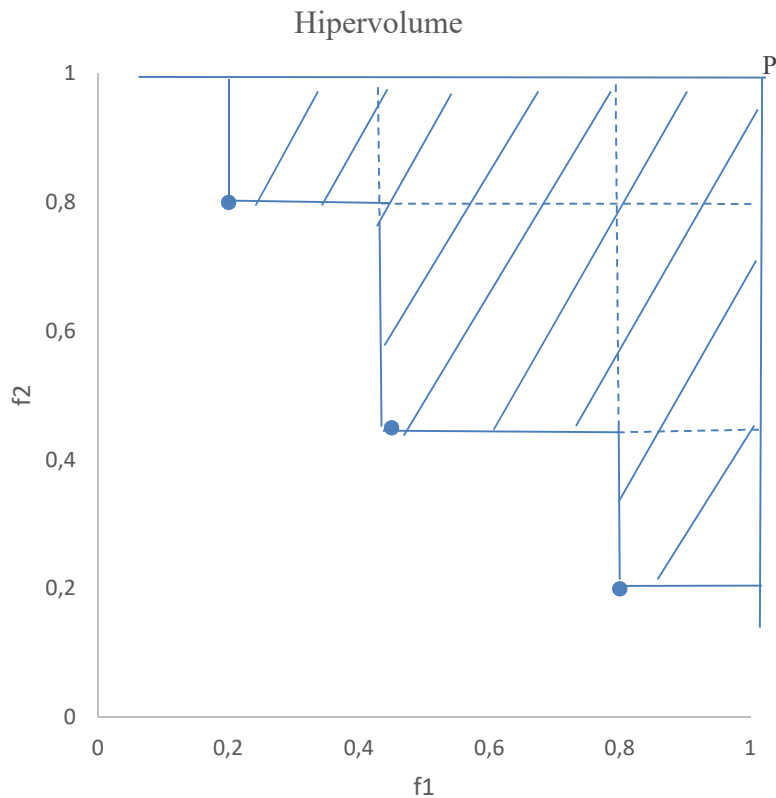


Fig. 2.6 Cálculo do hipervolume num problema de minimização

2.5.2 Problemas teste

Consideremos onze problemas teste, distribuídos da forma seguinte: 5 problemas bi-objetivo do conjunto de problemas ZDT (Zitzler et al., 2000); 3 problemas teste do conjunto DTLZ (Deb et al., 2005) (problemas dtlz1, dtlz2 e dtlz3), considerando formulações tri-objetivo; no domínio da otimização com restrições consideramos os problemas CONSTR, SRN e TNK (Deb et al., 2002).

2.5.3 Parametrização dos algoritmos

Os algoritmos foram implementados em Matlab e foram efetuadas 30 corridas de cada algoritmo para cada problema. O tamanho da população e do arquivo não dominado é de 100 indivíduos para todos os algoritmos. No algoritmo EMOEM, o arquivo local está limitado a 5 elementos e o número de iterações da pesquisa local é $lsit = 5$. Para equilibrar o esforço computacional de todos os algoritmos, o critério de paragem do algoritmo EMOEM é a realização de 50000 e 100000 avaliações das funções objetivo,

para problemas com duas e três funções objetivo, respectivamente; para os algoritmos MOEM, NSGA-II, OMOPSO e MOSADE, o critério de paragem é de 25000 e 50000 avaliações das funções objetivo, para problemas com duas e três funções objetivo, respectivamente. Os valores diferentes do critério de paragem adotados para o algoritmo EMOEM são justificados pela necessidade de nivelar o custo da execução computacional. É importante mencionar que aproximadamente metade das avaliações das funções objetivo efetuadas pelo algoritmo EMOEM ocorrem no procedimento de pesquisa local e o custo computacional por avaliação de função neste procedimento é muito menor do que no ciclo principal do algoritmo.

2.5.4 Comparação dos algoritmos EMOEM, MOEM, OMOPSO, MOSADE e NSGA-II

Os algoritmos utilizados nesta comparação são os que têm o melhor desempenho das suas classes. Nas Fig. 2.7-2.9 estão representadas as melhores soluções obtidas pelos algoritmos MOEM e EMOEM nos problemas ZDT1, SRN e TNK. Analisando os gráficos verificamos o melhor desempenho do algoritmo EMOEM, conseguindo uma melhor caracterização da frente ótima de Pareto do problema. O algoritmo MOEM apenas consegue maior número de soluções no problema TNK, o que se traduz numa melhor representação da frente ótima de Pareto deste problema. A Tabela 2.1 contém os valores da mediana e do intervalo inter-quartis do hipervolume obtidos pelos algoritmos em todos os problemas de teste. Para cada problema, o melhor valor é assinalado a negrito e o segundo melhor valor está sublinhado. No problema DTLZ1 apenas os algoritmos EMOEM e NSGA-II obtiveram soluções dentro do hipercubo definido pelo conjunto não dominado e pelo ponto de referência. No problema DTLZ3 ocorre a mesma situação, não obstante a mediana ser zero para o algoritmo EMOEM. Tal como podemos observar na Tabela 2.1, o algoritmo EMOEM ocupa o segundo lugar no que se refere ao desempenho de acordo com o hipervolume para a maior parte dos problemas. As diferenças são estatisticamente significativas ao nível de significância $\alpha = 0.05$ para todos os problemas resolvidos. Para esta análise foi utilizado o teste não paramétrico Kruskal-Wallis, por estarmos a comparar mais do que dois algoritmos e por os resultados de cada um dos algoritmos não seguirem a distribuição normal.

No caso da medida de desempenho IGD, os resultados obtidos para os cinco algoritmos são apresentados na Tabela 2.2. Estes resultados confirmam os resultados

observados pelo indicador hipervolume. O algoritmo EMOEM obtém melhores resultados do que o algoritmo MOSADE em todos os problemas com exceção dos problemas ZDT1, ZDT3 e DTLZ2, e é melhor do que o algoritmo MOEM em todos os problemas. Podemos observar que o algoritmo EMOEM mantém a sua posição relativa nos problemas com restrições. O algoritmo MOEM apresenta o pior desempenho entre todos os algoritmos utilizados nas experiências. Os resultados obtidos são estatisticamente significativos ao nível $\alpha = 0.05$ para todas as instâncias.

Tabela 2.1. Mediana e Intervalo Inter-Quartis (IQR) dos valores do hipervolume obtidos pelos algoritmos EMOEM, MOEM, OMOPSO, MOSADE e NSGA-II.

Problema	EMOEM		MOEM		OMOPSO		MOSADE		NSGA-II	
	Mediana	IQR	Mediana	IQR	Mediana	IQR	Mediana	IQR	Mediana	IQR
ZDT1	<u>0.6371</u>	0.0149	0.5544	0.1718	0.6312	0.0035	0.6261	0.0019	0.6571	0.0007
ZDT2	<u>0.3152</u>	0.0129	0.0000	0.0000	0.3010	0.2272	0.2749	0.0050	0.3248	0.0005
ZDT3	0.7246	0.0618	0.0001	0.1024	0.7020	0.0127	<u>0.7417</u>	0.0031	0.7736	0.0006
ZDT4	<u>0.5489</u>	0.0242	0.1491	0.0256	0.1010	0.0058	0.1333	0.0163	0.6569	0.0013
ZDT6	0.0398	0.0185	0.0000	0.0000	<u>0.2681</u>	0.0285	0.0000	0.0000	0.3162	0.0007
DTLZ1	<u>0.5705</u>	0.5237	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.9683	0.0018
DTLZ2	0.1410	0.0268	0.2176	0.0267	0.3203	0.0106	<u>0.4061</u>	0.0033	0.3633	0.0090
DTLZ3	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.3338	0.0352
CONSTR	<u>3.5797</u>	0.0550	3.1829	0.0847	2.6038	0.0015	3.5653	0.0500	5.7584	0.0032
SRN	<u>24331</u>	157.07	23871	297.03	24910	14.799	22096	1561.7	23782	231.02
TNK	0.3187	0.0015	0.3200	0.0029	0.3222	0.0010	<u>0.3239</u>	0.0006	0.3240	0.0007

Tabela 2.2. Mediana e Intervalo Inter-Quartis (IQR) dos valores de IGD obtidos pelos algoritmos EMOEM, MOEM, OMOPSO, MOSADE e NSGA-II.

Problema	EMOEM		MOEM		OMOPSO		MOSADE		NSGA-II	
	Mediana	IQR	Mediana	IQR	Mediana	IQR	Mediana	IQR	Mediana	IQR
ZDT1	0.0308	0.0256	0.1041	0.1231	<u>0.0215</u>	0.0023	0.0255	0.0014	0.0064	0.0003
ZDT2	<u>0.0125</u>	0.0086	0.7722	0.1257	0.0206	0.4431	0.0411	0.0051	0.0063	0.0004
ZDT3	0.0496	0.0394	0.8544	0.6901	0.0377	0.0070	<u>0.0165</u>	0.0013	0.0066	0.0005
ZDT4	<u>0.0863</u>	0.0288	0.5473	0.0513	0.5564	0.0075	0.5202	0.0203	0.0065	0.0004
ZDT6	0.5605	0.0686	2.9764	0.6501	<u>0.0236</u>	0.0186	1.2474	0.0751	0.0065	0.0008
DTLZ1	<u>0.2516</u>	0.5278	1.3785	1.0316	14.884	3.473	10.474	5.1913	0.0314	0.0033
DTLZ2	0.4645	0.1473	0.1763	0.0205	0.0933	0.0059	0.0714	0.0054	<u>0.0779</u>	0.0055
DTLZ3	<u>4.5693</u>	12.650	17.482	7.7891	174.98	16.579	180.36	19.424	0.0886	0.0253
CONSTR	<u>0.1747</u>	0.0309	1.1218	0.1394	0.8356	0.0004	0.1813	0.0921	0.1732	0.0009
SRN	<u>2.2151</u>	0.4660	2.7933	0.7269	1.0298	0.0613	8.0903	2.6726	4.4872	0.8205

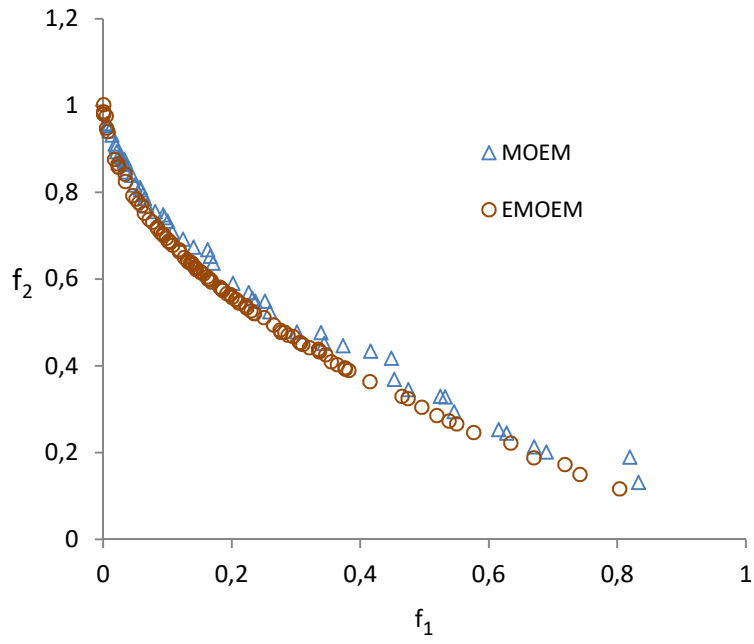


Fig. 2.7 Problema ZDT1

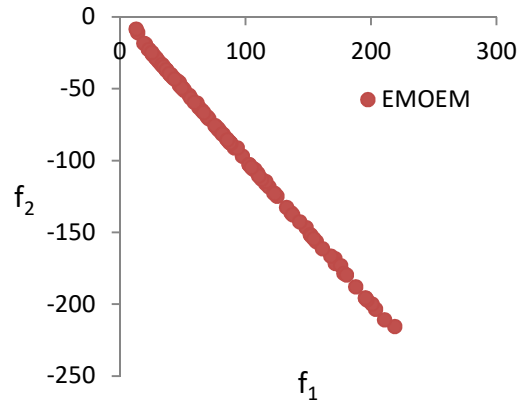
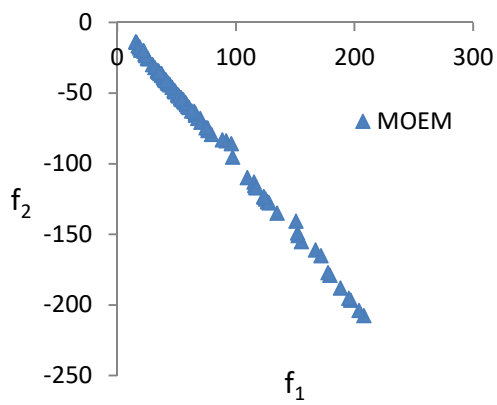


Fig. 2.8 Problema SRN

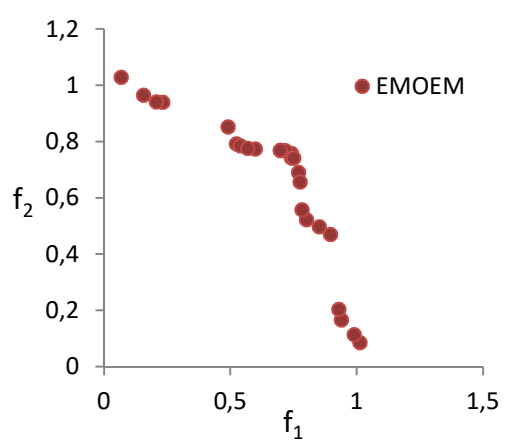
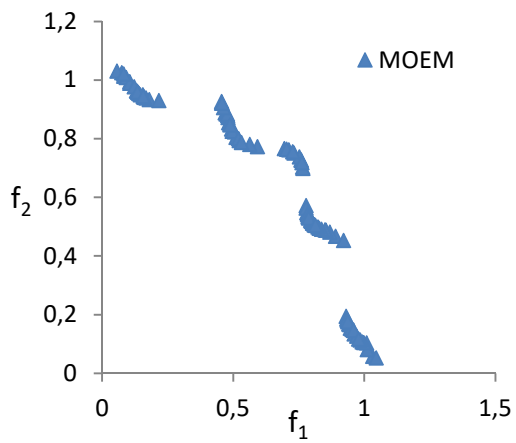


Fig. 2.9 Problema TNK

Na Tabela 2.3 são apresentados a mediana e o intervalo inter-quartil dos tempos de execução de cada um dos algoritmos nos problemas *benchmark* ZDT e DTLZ num computador Intel core 2 Duo com processador 2.4 GHz e 4 GB memória RAM. Apesar de o algoritmo EMOEM efetuar mais avaliações de funções, o seu custo computacional ainda é inferior ao do algoritmo MOEM. Genericamente, o tempo despendido por cada algoritmo na resolução de um problema depende fortemente do desempenho do algoritmo nesse problema, designadamente do número de soluções não dominadas obtidas, porque um melhor desempenho do algoritmo geralmente corresponde a um aumento do tamanho do arquivo não dominado. Em situações de pior desempenho dos algoritmos foram obtidas poucas soluções não dominadas e o esforço computacional decresceu substancialmente. Esta situação ocorreu, por exemplo, nos problemas DTLZ1 e DTLZ3, nos algoritmos MOEM, MOSADE e OMOPSO.

Em geral, o algoritmo NSGA-II apresenta melhores valores para as medidas de desempenho hipervolume e IGD do que os restantes algoritmos. No entanto, também exige um maior esforço computacional para alcançar um número de soluções não dominadas semelhante ao obtido pelos algoritmos OMOPSO e EMOEM.

Tabela 2.3. Mediana e Intervalo Inter-Quartis (IQR) dos tempos de execução obtidos pelos algoritmos EMOEM, MOEM, OMOPSO, MOSADE e NSGA-II.

Problema	EMOEM		MOEM		OMOPSO		MOSADE		NSGA-II	
	Mediana	IQR	Mediana	IQR	Mediana	IQR	Mediana	IQR	Mediana	IQR
ZDT1	160.01	23.954	275.86	4.5162	106.31	3.9780	56.862	2.4414	353.23	2.3127
ZDT2	147.66	60.275	297.31	7.5426	111.64	58.048	48.867	4.1223	424.01	1.5561
ZDT3	149.84	18.588	271.52	3.4008	59.936	1.7901	45.123	1.5873	440.55	33.080
ZDT4	170.66	12.800	203.10	2.0826	31.949	2.5662	17.730	0.0203	420.68	15.296
ZDT6	146.03	4.6917	212.47	9.5746	21.965	7.5582	12.347	0.3549	411.30	1.4898
DTLZ1	445.30	40.084	204.10	4.0560	14.851	1.0608	42.346	9.6760	816.46	2.5740
DTLZ2	699.95	12.862	247.70	1.7667	121.91	2.7378	447.82	183.57	842.17	2.1216
DTLZ3	331.34	28.396	214.67	3.7128	20.389	0.7488	42.175	3.3930	813.12	2.3868

Notamos que o algoritmo EMOEM conseguiu melhorar substancialmente o desempenho relativamente ao algoritmo original MOEM. No entanto este algoritmo foi superado pelo algoritmo de referência NSGA-II na maioria dos problemas *benchmark* utilizados, embora tenha ficado na segunda posição em muitos problemas, com melhor desempenho do que os restantes algoritmos analisados.

2.6 Considerações finais

Têm sido propostas diversas meta-heurísticas para resolver problemas de otimização mono e multiobjetivo. A abordagem EM é uma meta-heurística relativamente recente que tem demonstrado bom desempenho em problemas mono-objetivo (Alikani et al., 2009) (Tavakkoli-Moghaddam et al., 2009). Além disso, têm sido propostas na literatura diversas modificações do algoritmo inicial, que melhoraram o seu desempenho em diversos problemas teste. O algoritmo MOEM (Tsou et al., 2006) representa uma primeira tentativa de aplicação da abordagem EM em problemas de otimização multiobjetivo. No entanto, este algoritmo mostrou um fraco desempenho em comparação com outras meta-heurísticas representativas do estado da arte.

A nossa motivação para o estudo nesta área e para o desenvolvimento do algoritmo EMOEM deriva de experiências preliminares que efetuámos, nas quais a abordagem EM mostrou um elevado desempenho em problemas mono-objetivo por comparação com a abordagem PSO. No entanto, o algoritmo MOEM original (Tsou et al., 2006) demonstrou fortes dificuldades em diversos problemas multiobjetivo. O algoritmo EMOEM pretende ultrapassar o enviesamento dos mecanismos da carga individual e da atualização de posição, além de modificar também o procedimento de pesquisa local. O comportamento do algoritmo EMOEM representa uma melhoria significativa relativamente ao algoritmo MOEM, aumentando a qualidade das soluções e o número de soluções não dominadas encontradas. O algoritmo EMOEM mostrou-se ainda competitivo quando comparado com os resultados dos outros algoritmos representativos do estado da arte (OMOPSO, MOSADE, NSGA-II), mas não os conseguiu superar.

Tendo em conta os resultados globais obtidos, iremos concentrar a investigação nos algoritmos evolucionários multiobjetivo (em particular o algoritmo NSGA-II) e na otimização por enxame de partículas (tendo por base o algoritmo OMOPSO) para os desenvolvimentos seguintes no domínio dos problemas de otimização em dois níveis multiobjetivo.

3 Abordagens PSO e evolucionária para problemas em dois níveis multiobjetivo

3.1 Motivação

Os problemas de otimização com decisões hierárquicas surgem em diversos contextos de aplicação. Estes problemas caracterizam-se pelo facto de as funções objetivo e as restrições dependerem da concretização de variáveis de outros problemas de otimização que não podem ser considerados ao mesmo nível, tipicamente representando interesses antagónicos num contexto de um processo de decisão sequencial. De uma forma simples, temos um problema de otimização incluído nas restrições de outro problema de otimização. Temos, assim, um primeiro nível de otimização e um segundo nível, cujas soluções vão influenciar os resultados do primeiro. Neste caso temos um problema de otimização em dois níveis (*bi-level* - BL). Em determinadas situações podemos inclusivamente ter mais do que dois níveis de decisão. Neste caso temos um problema multi-nível.

Os problemas BL são constituídos por um problema de nível superior e por um problema de nível inferior, que está incluído nas restrições do primeiro. O decisor do nível superior é o líder e o decisor do nível inferior é o seguidor. A resolução do problema BL exige, para cada conjunto de valores das variáveis do nível superior, a resolução de um problema de otimização no nível inferior. O objetivo do líder é otimizar a sua função objetivo, mas deve ter em conta no seu processo de decisão a reação por parte do seguidor porque esta afeta o valor da função objetivo do líder e eventualmente a admissibilidade da solução. Este facto eleva a dificuldade do problema, tornando muito difícil a sua resolução, mesmo no caso linear onde o problema é *NP-hard* (Dempe, 2002).

Encontramos problemas de otimização em dois níveis em setores tão diversos como a produção industrial, o desenho de redes de transportes e gestão de tráfego, as redes informáticas ou os mercados de energia.

Têm sido desenvolvidos esforços de investigação significativos no sentido de reformular o problema BL, permitindo a sua resolução através de métodos de programação matemática (Ruuska et al., 2011). Contudo, apenas foram estabelecidas condições para transformação do problema em dois níveis num problema de otimização com um único nível quando o problema verifica exigentes propriedades de linearidade ou convexidade (Eichfelder, 2010). Nestes casos, as condições de Karush-Kuhn-Tucker (KKT) permitem transformar o problema BL num problema equivalente com um único nível que pode ser tratado utilizando técnicas de programação matemática adequadas.

Um problema BL pode ter múltiplas funções objetivo no nível superior, no nível inferior ou em ambos os níveis. Se o problema BL contiver um problema multiobjetivo apenas no nível inferior, é comum designar-se por problema BL semi-vetorial (BLSV). A existência de múltiplas soluções ótimas de Pareto no problema do nível inferior para uma dada solução do nível superior dificulta o processo de decisão e a abordagem do problema de otimização multiobjetivo BL (BLMOP) ou BLSV. O seguidor poderá escolher qualquer uma das soluções ótimas de Pareto do seu problema. A decisão do seguidor tem impacto nos valores das funções objetivo do problema do nível superior, sem que o líder possa influenciar o resultado a partir da altura em que definiu as variáveis do nível superior. Assim, quer o problema do nível superior seja mono-objetivo ou multiobjetivo, o líder terá sempre de escolher uma solução antecipando a decisão do seguidor relativa à seleção de uma solução de entre o conjunto de soluções ótimas de Pareto. Caso o líder não possua informação que lhe permita antecipar a reação do seguidor, poderá adotar uma de duas perspetivas extremas, otimista ou pessimista, relativamente à decisão do seguidor. Na perspetiva otimista o líder considera que o seguidor escolhe, de entre o conjunto de soluções ótimas de Pareto para o seu problema, a solução mais favorável para o líder. Na perspetiva pessimista o líder assume que o seguidor escolhe a pior solução para o líder. Recentemente o problema BLMOP tem sido estudado por diversos investigadores, tendo sido propostos alguns algoritmos para o resolver. Todas estas abordagens adotam uma perspetiva otimista do problema.

Eichfelder (2007, 2010) apresentou novos resultados teóricos e propôs um algoritmo para resolver o problema BLMOP com problemas bi-objetivo em ambos os níveis e uma única variável no nível superior. Nishizaki e Sakawa (1999), Shi e Xia (2001) e Abo-Sinna e Baky (2007) propuseram algoritmos iterativos para resolver o problema BLMOP. Os algoritmos evolucionários têm também sido utilizados para tratar o problema BLMOP. Deb e Sinha (2009) propuseram um algoritmo evolucionário

(BLEMO) baseado no algoritmo multiobjetivo NSGA-II. O algoritmo BLEMO tem sido utilizado para resolver problemas BLMOP, incluindo problemas com restrições no nível superior dependentes de variáveis de nível superior. Em Sinha e Deb (2009) foram consideradas regras adicionais para melhorar a capacidade de o algoritmo encontrar a frente ótima de Pareto, nomeadamente a exigência de permanência de uma solução não dominada na população do problema do nível inferior durante um determinado número de gerações para que esta solução possa ser aceita como solução do problema BL. Sinha et al. (2015) consideram o problema BLMOP assumindo que o líder conhece o processo de tomada de decisão do seguidor, o qual é representado através da maximização de uma função valor que agrega os múltiplos objetivos do seguidor. Assim, o problema do nível inferior é transformado num problema mono-objetivo, sendo apresentada uma única solução do problema do seguidor por cada solução do líder. Para reduzir o esforço computacional, algumas soluções do problema do nível inferior são obtidas a partir de aproximações para as variáveis do nível inferior usando funções quadráticas das variáveis do nível superior, após serem conhecidas outras soluções do problema do nível inferior determinadas anteriormente pelo algoritmo. Sinha et al. (2017) propõem um algoritmo evolucionário para resolver o problema BLMOP, usando aproximações quadráticas para obter um conjunto de soluções não dominadas do problema multiobjetivo do nível inferior parametrizadas pelo vetor de variáveis do problema do nível superior. A ideia essencial é que, conhecendo um conjunto aproximado de soluções para um certo número de problemas do nível inferior (dependentes de diferentes concretizações das variáveis do nível superior), não é necessário resolver um problema de otimização para obter um conjunto aproximado de soluções para outras instanciações do problema multiobjetivo do nível inferior. A abordagem proposta determina uma aproximação do conjunto de soluções não dominadas de um problema do nível inferior a partir de conjuntos de soluções não dominadas correspondentes a soluções vizinhas no problema do nível superior. A aplicação desta técnica exige a verificação de condições matemáticas restritas. Em muitos problemas onde estas condições não se verificam, os autores propõem a aplicação desta técnica na determinação de uma aproximação inicial do conjunto de soluções do problema do nível inferior, de forma a acelerar a convergência do algoritmo evolucionário. A otimização com enxame de partículas (PSO) tem também sido utilizada para resolver problemas BL com múltiplos objetivos em um ou em ambos os níveis. Halter e Mostaghim (2006) propuseram um algoritmo baseado em enxame de partículas para resolver um problema real BL com restrições lineares. Alves (2012) propôs um

algoritmo MOPSO BL para resolver problemas BL lineares com múltiplos objetivos no nível superior e um único objetivo no nível inferior. Este algoritmo foi posteriormente melhorado em Alves e Costa (2014). Os algoritmos em Alves (2012), Alves e Costa (2014) e Halter e Mostaghim (2006) exploram a linearidade das funções objetivo e/ou das restrições para resolver o problema do nível inferior. Zhang et al. (2012) apresentaram um algoritmo baseado em otimização multiobjetivo com enxame de partículas para resolver o problema BLMOP. Para evitar convergência prematura na otimização com enxame de partículas, foi mais tarde introduzido um operador cruzamento (Zhang et al., 2013).

Neste capítulo apresentamos um novo algoritmo baseado em PSO (MO-BLPSO) para o problema BLMOP comparando-o com o algoritmo BLEMO.

3.2 Formulação

Começamos por apresentar a definição de um problema de programação BL com uma única função objetivo em cada nível:

$$\begin{aligned} & \min_{x_u} F(x_u, x_l) \\ & \text{s.a. } G(x_u, x_l) \leq 0 \\ & \quad x_l \in \arg \min_{x_l} \{f(x_u, x_l) : g(x_u, x_l) \leq 0\} \end{aligned} \quad (3.1)$$

onde $x_u \in \mathfrak{R}^{n_1}$ é um vetor de dimensão n_1 das variáveis do nível superior e $x_l \in \mathfrak{R}^{n_2}$ é um vetor de dimensão n_2 das variáveis do nível inferior; $F(x_u, x_l)$ e $f(x_u, x_l)$ são as funções objetivo do líder e do seguidor, respetivamente. Dado que o seguidor otimiza $f(x_u, x_l)$ após x'_u ter sido selecionado, x_u assume um vetor constante sempre que $f(x_u, x_l)$ é otimizado. A reação racional do seguidor dado um x'_u é:

$$\Psi(x'_u) = \left\{ x_l \in \mathfrak{R}^{n_2} : x_l \in \arg \min_{x_l} \{f(x'_u, x_l) : g(x'_u, x_l) \leq 0\} \right\}$$

A região admissível para o problema BL (3.1), também designada por região induzida, é $IR = \{(x_u, x_l) \in \mathfrak{R}^{n_1+n_2} : G(x_u, x_l) \leq 0, x_l \in \Psi(x_u)\}$. O problema BL é o problema do líder. É de notar que não é clara a definição do valor da função objetivo $F(x_u, x_l)$ do ponto de vista do líder (que apenas tem o controlo de x_u) se o conjunto das soluções ótimas do problema do nível inferior não for singular. A existência de soluções ótimas alternativas

no nível inferior coloca dificuldades ao líder, na medida em que ele desconhece qual delas o seguidor vai escolher. Esta escolha pode afetar significativamente o valor da função objetivo do líder e, conseqüentemente, a sua decisão. Em muitos estudos esta dificuldade não é abordada, sendo assumida a existência de uma única solução no nível inferior. Outros estudos consideram uma abordagem otimista ou pessimista. A abordagem otimista (a mais comum) assume que o líder é livre para selecionar a solução que melhor serve os seus interesses entre as soluções ótimas para o seguidor. Neste caso, a otimização do nível superior é tomada relativamente a x_u e x_l , significando que o líder tem a capacidade de influenciar as decisões do seguidor (Tsoukalas et al., 2009). Assim, o problema pode ser escrito como:

$$\min_{x_u, x_l} \{F(x_u, x_l) : x_l \in \Psi(x_u), G(x_u, x_l) \leq 0\}.$$

Quando não é permitida a cooperação entre o líder e o seguidor, ou se o líder for avesso ao risco e pretender limitar os danos resultantes da escolha de uma solução indesejável por parte do seguidor, pode ser admitida uma abordagem pessimista. Neste caso o líder prepara-se para o pior cenário, ou seja, escolhe a decisão x_u que tem melhor desempenho na perspetiva da “pior” resposta do seguidor entre as soluções que para ele são ótimas (alternativas) (Tsoukalas et al., 2009). Isto conduz ao seguinte problema:

$$\min_{x_u} \left\{ \max_{x_l} \{F(x_u, x_l) : x_l \in \Psi(x_u)\} : G(x_u, x_l) \leq 0 \right\}$$

A discussão entre as abordagens otimista e pessimista em problemas BL tem sido apresentada em vários estudos, tais como Dempe (2002) e Tsoukalas et al. (2009), entre outros.

Neste capítulo vamos considerar ambos os problemas dos níveis superior e inferior como problemas de otimização multiobjetivo, considerando uma abordagem otimista. Um problema BLMOP pode ser formulado genericamente por

$$\begin{aligned} \min_{x_u \in X} F &= (F_1(x_u, x_l), \dots, F_m(x_u, x_l)) \\ \text{s.a. } G(x_u, x_l) &\leq 0 \\ x_l &\in \arg \min_{x_l \in Y} \{f = (f_1(x_u, x_l), \dots, f_n(x_u, x_l)) : g(x_u, x_l) \leq 0\} \end{aligned} \quad (3.2)$$

onde $F(x)$ e $f(x)$ são os vetores das funções objetivo do nível superior e inferior a otimizar, $x = (x_u, x_l)$ com $x_u \in \mathfrak{R}^{n_1}$ e $x_l \in \mathfrak{R}^{n_2}$ os vetores das variáveis de decisão do

nível superior e inferior, e $G(x)$, $g(x)$ as funções das restrições do nível superior e inferior, respectivamente. X e Y definem limites inferiores/superiores nas variáveis de decisão x_u e x_l .

3.3 Algoritmo BLEMO

O algoritmo BLEMO (Deb e Sinha, 2009) utiliza os operadores do algoritmo evolucionário NSGA-II para resolver os problemas multiobjetivo dos níveis superior e inferior que compõem o problema BLMOP (3.2). O algoritmo utiliza uma estrutura adequada para representar a população, que é composta por N indivíduos divididos em n_s sub-populações, representada na Fig. 3.1. Todos os membros de uma sub-população têm o mesmo vetor x_u . Assim, na população existem apenas n_s vetores das variáveis de decisão do nível superior diferentes e em cada sub-população existem $N_l = N/n_s$ indivíduos.

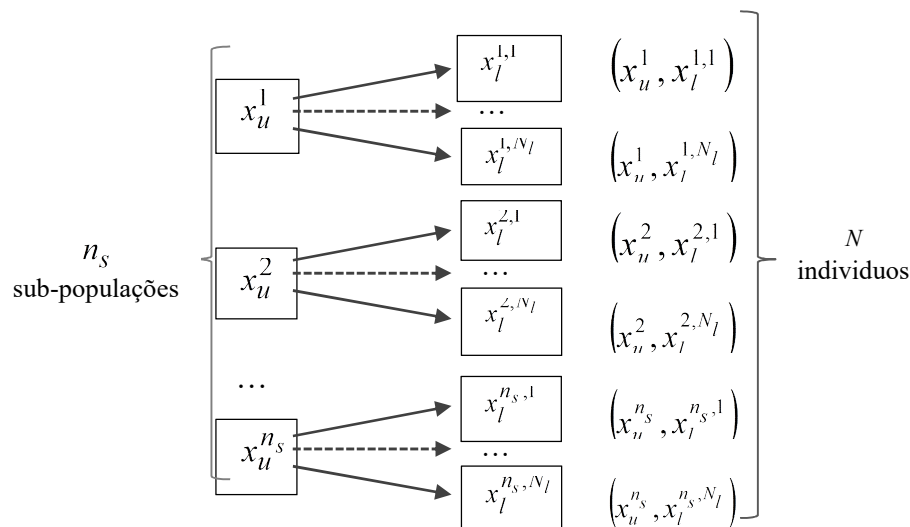


Fig. 3.1 Estrutura da população

O algoritmo BLEMO utiliza o mecanismo de ordenação por frentes de não dominância e o operador de distância de *crowding* introduzidos em Deb et al. (2002) para classificar cada membro da população. Assim, a cada indivíduo é atribuída uma ordem de não dominância (ND_u) e uma medida de distância de *crowding* (CD_u) relativas aos objetivos e restrições do nível superior. São também calculadas a ordem de não

dominância (ND_i) e a medida de distância de *crowding* (CD_i) relativas aos objetivos e restrições do nível inferior dentro de cada sub-população. Cada iteração do nível superior do algoritmo BLEMO começa por criar uma nova população, utilizando os operadores genéticos seleção, cruzamento e mutação. Em seguida são calculadas as soluções não dominadas do problema do nível inferior para cada sub-população. O algoritmo BLEMO efetua esta operação utilizando o algoritmo NSGA-II. No final de cada iteração, o algoritmo BLEMO guarda as soluções não dominadas do problema (3.2) num arquivo externo. Pretendendo impedir que soluções dominadas do nível inferior (i.e. soluções não admissíveis para o problema (3.2)) entrem no arquivo, é exigido que este apenas possa receber os indivíduos que tenham permanecido na população pelo menos durante um dado número de gerações r . Este arquivo é ordenado utilizando a não dominância e a distância de *crowding*.

3.4 Algoritmo MO-BLPSO

Nesta secção propomos um algoritmo baseado na abordagem PSO para resolver um problema de otimização BL multiobjetivo em ambos os níveis (MO-BLPSO) (Carrasqueira et al., 2015b). O algoritmo utiliza um esquema semelhante ao utilizado no algoritmo BLEMO para a interação entre as fases de otimização dos níveis superior e inferior e na representação da população, dividindo-a em sub-populações, em que todos os membros da mesma sub-população têm o mesmo vetor x_u . O algoritmo utiliza um arquivo externo para guardar as soluções não dominadas do problema BL, atualizado após cada iteração do nível superior e considera também um arquivo do nível inferior para guardar as soluções não dominadas obtidas durante as iterações do nível inferior. Este arquivo guia a pesquisa no nível inferior. O algoritmo MO-BLPSO utiliza as principais opções e parâmetros do algoritmo OMOPSO (Reyes-Sierra & Coello, 2005) para resolver ambos os problemas de otimização, dos níveis superior e inferior.

O algoritmo MO-BLPSO começa por inicializar a população P_t (iteração inicial $t = 1$) aleatoriamente. De seguida são efetuadas T iterações no problema do nível superior. Em cada iteração o algoritmo MO-BLPSO procura soluções não dominadas do nível superior. Em cada iteração principal (nível superior), o algoritmo MO-BLPSO calcula soluções não dominadas do problema do nível inferior. As soluções admissíveis do problema do nível superior estão entre estas soluções. Na rotina do nível inferior são

efetuadas T_l iterações do algoritmo OMOPSO para cada solução de nível superior. No final de cada rotina do nível inferior, o arquivo A que contém as soluções não dominadas do nível superior é atualizado.

Descrevemos de seguida cada um dos passos executados pelo algoritmo MO-BLPSO em cada iteração do nível superior. O algoritmo começa por criar as n_s sub-populações a partir da população da iteração anterior. Para criar cada sub-população, é selecionado um vetor do nível superior x_u de uma partícula da população por torneio binário usando (ND_u) e (CD_u) lexicograficamente, o qual é de seguida atualizado usando os operadores do algoritmo OMOPSO. Selecionam-se então aleatoriamente N_l partículas da população atual P_t e atualizam-se os vetores do nível inferior x_l usando os operadores do algoritmo OMOPSO. Para o nível superior, a partícula $gbest$ é selecionada do arquivo não dominado A por torneio binário, selecionando as partículas com maiores valores de CD_u . Para o nível inferior, em cada sub-população n_s a partícula $gbest$ é selecionada do arquivo não dominado dessa sub-população B_t^s , utilizando o processo de torneio binário relativamente aos valores CD_l dando preferência às partículas com maior valor de distância de *crowding*. As soluções x_l são então concatenadas com a solução x_u para formar uma sub-população. Notemos que todas as N_l soluções de uma mesma sub-população têm o mesmo vetor x_u . De seguida, as partículas da sub-população são avaliadas e são atribuídos os valores de ordem de não dominância (ND_l) e de distância de *crowding* (CD_l) a cada indivíduo. Estas operações são repetidas até n_s sub-populações terem sido obtidas. Finalmente reúnem-se todas as sub-populações numa população (Q_t) e são atribuídos os valores de ordem de não dominância de nível superior (ND_u) e de distância de *crowding* (CD_u) a cada indivíduo em Q_t .

O ciclo do nível inferior do algoritmo é executado usando o algoritmo OMOPSO para resolver o problema do nível inferior para cada solução de nível superior x_u . Assim, para cada sub-população s , o arquivo de soluções não dominadas B_t^s é inicializado identificando o vetor x'_u dos elementos do arquivo A mais próximo do vetor x_u da sub-população s , em termos de distância euclideana, e selecionando todos os elementos do arquivo A cujo vetor do nível superior é x'_u para serem inseridos no arquivo B_t^s . A velocidade, posição e melhor posição obtida para cada partícula, são também inicializadas

da forma usual utilizada na abordagem PSO. O ciclo do nível inferior é então executado durante T_l iterações, sendo em cada iteração executada a rotina *OMOPSOlowerlevel*. O vetor *gbest* é escolhido por torneio binário entre os elementos do arquivo B_t^s . Após cada iteração da rotina *OMOPSOlowerlevel* a sub-população é avaliada e são atualizadas a posição *pbest*, a ordem de não dominância do nível inferior (ND_l) e a distância de *crowding* (CD_l) de cada partícula. O arquivo não dominado B_t^s é então atualizado, utilizando as partículas não dominadas ($ND_l = 1$) da sub-população. Este arquivo é ainda ordenado por ordem decrescente da distância de *crowding* dos seus membros, por forma a privilegiar a seleção de partículas líder localizadas em regiões de menor concentração. Após serem realizadas estas operações é executada uma nova iteração do ciclo do nível inferior, até serem concluídas T_l iterações.

Após terminar este processo, os elementos do arquivo B_t^s são testados para entrar no arquivo não dominado A . Este procedimento é repetido para todas as sub-populações. Pode acontecer que algumas partículas do arquivo do nível inferior não sejam verdadeiras soluções ótimas de Pareto para o problema. Estas partículas são não admissíveis para o problema do nível superior, apesar de poderem dominar algumas soluções ótimas de Pareto verdadeiras do problema do nível superior. Assim, deve ser garantido que essas soluções não entram no arquivo A . Para ultrapassar esta questão é utilizado um procedimento para filtrar as partículas que entram neste arquivo. Deste modo, apenas são testadas para entrar no arquivo A as partículas que permaneceram no arquivo do nível inferior B_t^s pelo menos durante r iterações. Além disso, apenas entram no arquivo A as soluções que são não dominadas no nível superior relativamente a qualquer membro deste arquivo. Os elementos do arquivo A que são dominados pelo novo membro são então removidos de A . Após a movimentação de toda a população juntam-se as n_s sub-populações, sendo obtida a população atualizada Q_t .

A nova população a considerar na iteração seguinte do problema (3.1) será obtida a partir das populações P_t e Q_t . Para efetuar a seleção juntam-se as duas populações P_t e Q_t numa única população R_t composta por $2N$ partículas, mantendo o agrupamento das partículas em sub-populações de acordo com a estrutura ilustrada na Fig. 3.1. De forma a selecionar as melhores partículas relativamente ao problema do nível superior, os elementos da população R_t são classificados em termos de não dominância (ND_u) e

distância de *crowding* (CD_u) no nível superior. Note-se que, em cada sub-população, as partículas estão classificadas relativamente à não dominância e distância de *crowding* no nível inferior através dos valores ND_l e CD_l , respetivamente. Após serem classificadas as partículas, a população R_t é ordenada por ordem crescente dos valores ND_u e ND_l e por ordem decrescente dos valores CD_u , lexicograficamente. Ou seja, começam por ser selecionadas as partículas não dominadas relativamente a toda a população no nível superior e relativamente à respetiva sub-população no nível inferior pertencentes a regiões de menor concentração de partículas. Seguem-se as partículas não dominadas relativamente à restante população, ou seja, os que têm valores $ND_u = 2$ observando igualmente a ordem dos restantes critérios (CD_u e ND_l), e assim sucessivamente para a restante população. A nova população S_t é obtida percorrendo os elementos de R_t e inserindo a sub-população a que a partícula selecionada pertence, até serem inseridas n_s sub-populações, ou seja, até a população S_t conter N membros.

As sub-populações de S_t oriundas da população P_t são então atualizadas relativamente ao problema do nível inferior, dado que não foram movimentadas nesta iteração. Assim, é executado o ciclo do nível inferior para estas sub-populações. A nova população P_{t+1} é criada juntando as n_s sub-populações de S_t . Após serem efetuadas T iterações, o algoritmo devolve o arquivo A contendo as soluções não dominadas no nível superior como as soluções do problema BLMOP (3.2).

Deve ser referido que, na primeira iteração do algoritmo, o arquivo A é um conjunto vazio. Assim, é criado um pré-arquivo (*Pre A*) contendo as soluções não dominadas existentes na população inicial, sendo usado em substituição de A em todos os passos da iteração $t = 1$.

Os passos do algoritmo MO-BLPSO são detalhados no Algoritmo 3.1, que chama a rotina *OMOPSOlowerlevel* descrita no Algoritmo 3.2.

Algoritmo 3.1.

Algoritmo MO – BLPSO – pseudo-código

Inicializa o contador de iterações do nível superior, $t = 1$

Inicializa aleatoriamente cada partícula da população P_t :

$$x_{s_j}^t = \left(x_{u_s}^t, x_{l_j}^t \right), j = 1, \dots, N_l, s = 1, \dots, n_s$$

Avalia cada partícula da população

Classifica todos os membros de cada sub-população, calculando os valores ND_l e CD_l

Classifica as partículas da população, calculando os valores ND_u e CD_u

Pre A = ϕ , $A = \phi$

Insera as partículas de P_t com $ND_u = 1$ e $ND_l = 1$ no arquivo *Pre A*

Enquanto $t < T$

 // cria uma nova população Q_t como a união de n_s sub-populações Q_t^s

$$Q_t = \phi$$

Para $s = 1, \dots, n_s = N / N_l$

 Seleciona uma partícula de P_t por torneio binário, usando ND_u e CD_u

 Atualiza a componente $x_{u_s}^t$ dessa partícula, usando (2.1) e (2.2) e aplica o

operador turbulência

 Seleciona aleatoriamente N_l partículas da população P_t

Para $j = 1, \dots, N_l$

 Cria $x_{l_j}^t$, usando (2.1) e (2.2) e aplica turbulência

$$\text{Concatena } x_{s_j}^t \leftarrow \left(x_{u_s}^t, x_{l_j}^t \right)$$

$$Q_t^s \leftarrow Q_t^s \cup \left\{ x_{s_j}^t \right\}$$

Termina Para

 Avalia a sub-população Q_t^s , avaliando F, G, f, g

 Atribui os valores ND_l e CD_l a cada $x_{s_j}^t, j = 1, \dots, N_l$

Termina Para

$$Q_t \leftarrow \bigcup_{s=1}^{n_s} Q_t^s$$

Para $s = 1, \dots, n_s$

$$t_l = 1$$

Se $t > 1$

 Seleciona $B_t^s = A^s$, o sub-arquivo de A^s mais próximo de Q_t^s

Caso contrário

 Seleciona $B_t^s = \text{Pre } A^s$, o sub-arquivo de *Pre A* mais próximo de Q_t^s

Termina Se

$$x_{s_j}^{t_l} \leftarrow x_{s_j}^t, j = 1, \dots, N_l$$

Inicializa $pbest_j$ com o indivíduo $x_{l_j}^{t_l}, j = 1, \dots, N_l$

Enquanto $t_l \leq T_l$

$$OMOPSO_{lowerlevel}(Q_t^s, B_t^s, t_l)$$

Classifica os indivíduos de Q_t^s calculando os valores ND_l e CD_l

Atualiza o arquivo de soluções não dominadas do nível inferior B_t^s

$$t_l \leftarrow t_l + 1$$

Termina Enquanto

Atualiza o arquivo externo A das soluções não dominadas

Termina Para

$$Q_t \leftarrow \bigcup_{s=1}^{n_s} Q_t^s$$

$$R_t \leftarrow P_t \cup Q_t$$

Atribui os valores ND_u e CD_u a cada indivíduo $x_{s_j}^t, j = 1, \dots, 2N$ de R_t

Cria a população S_t com metade dos indivíduos de R_t , usando ND_u, CD_u e ND_l

Seja $I = \{1, \dots, I\}$ o conjunto dos índices das sub-populações de S_t oriundas de P_t

Para cada sub-população $S_t^i, i \in I$

$$t_l = 1$$

Se $t > 1$

Seleciona $B_t^i = A^i$, o sub-arquivo de A mais próximo de S_t^i

Caso contrário

Seleciona $B_t^i = Pre A^i$, o sub-arquivo de $Pre A$ mais próximo de S_t^i

Termina Se

$$x_{s_j}^{t_l} \leftarrow x_{s_j}^t, j = 1, \dots, N_l$$

Inicializa $pbest_j$ com o indivíduo $x_{l_j}^{t_l}, j = 1, \dots, N_l$

Enquanto $t_l \leq T_l$

$$Omopsolowerlevel(S_t^i, B_t^i, t_l)$$

Classifica os indivíduos de S_t^i calculando os valores ND_l e CD_l

Atualiza o arquivo de soluções não dominadas do nível inferior B_t^i

$$t_l \leftarrow t_l + 1$$

Termina Enquanto

Atualiza o arquivo externo A das soluções não dominadas

Termina Para

Atribui valores ND_u e CD_u a cada indivíduo $x_{s_j}^t$, $s = 1, \dots, n_s$, $j = 1, \dots, N_l$ de S_t

$P_{t+1} \leftarrow S_t$

$t \leftarrow t + 1$

Termina Enquanto

Devolve o arquivo de soluções não dominadas A

Algoritmo 3.2.

Rotina $OMOPSO_{lowerlevel}(P, B, t_l)$ – pseudo-código

Para cada partícula $x_i^{t_l} \in P$, $i = 1, \dots, N_l$

Seleciona o líder $gbest_i$, do arquivo B , por torneio binário

Atualiza a velocidade da partícula $v_i^{t_l+1}$, usando (2.1)

Atualiza a posição da partícula $x_i^{t_l+1}$ usando (2.2)

Aplica turbulência à partícula $x_i^{t_l+1}$ e avalia $f(x_i^{t_l+1})$

Atualiza $pbest_i$ da partícula i , substituindo-a por $x_i^{t_l+1}$ no caso de $pbest_i$ ser dominada por esta ou se forem mutuamente não dominadas entre si

Termina Para

A estrutura do problema BLMOP exige a execução de uma certa sequência de operações dos níveis superior e inferior. Assim, efetuar uma sequência de várias iterações da pesquisa do nível superior sem executar a fase de otimização do nível inferior pode tornar-se um desperdício de tempo porque as soluções obtidas são não admissíveis para o problema BLMOP se forem dominadas no problema do nível inferior. No nosso algoritmo apenas as soluções obtidas após uma execução do nível inferior são aceites para entrar no arquivo A . Este processo difere do algoritmo apresentado em Zhang et al. (2012, 2013), no qual uma solução entra no arquivo diretamente após uma execução do nível superior, sem efetuar a otimização do nível inferior. Estes autores não revelam como resolvem os inconvenientes resultantes desta operação. Testámos este processo e obtivemos soluções a entrarem no arquivo A sendo dominadas para o problema do nível inferior. Assim que estas soluções entram no arquivo pode tornar-se impossível retirá-las

e inserir outras soluções não dominadas (de nível superior) que contenham verdadeiras soluções ótimas de Pareto do nível inferior porque as novas soluções, admissíveis, podem ser dominadas no nível superior pelas soluções não admissíveis que já estão no arquivo A . Efetuámos alguns testes com o algoritmo proposto por Zhang et al. e confirmámos esta dificuldade, a qual é ultrapassada pelo algoritmo MO-BLPSO aqui proposto.

3.5 Análise comparativa entre os algoritmos MO-BLPSO e BLEMO

Para analisar o desempenho do algoritmo MO-BLPSO, serão utilizados problemas teste existentes na literatura. Um dos problemas considerados pode ser parametrizado relativamente ao número de variáveis do nível inferior. Assim, serão consideradas três variantes deste problema. Para efetuar uma análise comparativa será considerado o algoritmo BLEMO. Apresentamos inicialmente os problemas teste utilizados e de seguida analisaremos os resultados obtidos por ambos os algoritmos MO-BLPSO e BLEMO.

3.5.1 Problemas teste

São resolvidas quatro instâncias de dois problemas BLMOP de Deb e Sinha (2009), considerando três dimensões diferentes para o segundo problema.

Problema 3.1

$$\begin{aligned}
 \min F_1(x_u, x_l) &= x_{l_1} - x_u \\
 \min F_2(x_u, x_l) &= x_{l_2} \\
 \text{s.a } G(x_u, x_l) &= 1 + x_{l_1} + x_{l_2} \geq 0 \\
 \min f_1(x_l) &= x_{l_1} \\
 \min f_2(x_l) &= x_{l_2} \\
 \text{s.a } g(x_u, x_l) &= x_u^2 - x_{l_1}^2 - x_{l_2}^2 \geq 0 \\
 -1 \leq x_{l_1}, x_{l_2} &\leq 1, 0 \leq x_u \leq 1
 \end{aligned}$$

Este problema tem uma variável do nível superior e duas variáveis do nível inferior. Verifica-se também a dependência da restrição do problema do nível superior das variáveis do nível inferior, o que dificulta a resolução do problema.

Problema 3.2

$$\begin{aligned}
 \min F_1(x_u, x_l) &= (x_{l_1} - 1)^2 + \sum_{i=1}^K x_{l_{i+1}}^2 + x_u^2 \\
 \min F_2(x_u, x_l) &= (x_{l_1} - 1)^2 + \sum_{i=1}^K x_{l_{i+1}}^2 + (x_u - 1)^2 \\
 \text{s.a } \min f_1(x_u, x_l) &= x_{l_1}^2 + \sum_{i=1}^K x_{l_{i+1}}^2 \\
 \min f_2(x_u, x_l) &= (x_{l_1} - x_u)^2 + \sum_{i=1}^K x_{l_{i+1}}^2 \\
 &-1 \leq x_{l_1}, x_{l_2}, \dots, x_{l_{K+1}}, x_u \leq 2
 \end{aligned}$$

Este problema tem uma variável do nível superior e $K+1$ variáveis do nível inferior. Para cada valor de x_u apenas uma solução de toda a frente ótima de Pareto do problema do nível inferior é ótima de Pareto para o problema do nível superior. Isto torna o problema difícil de resolver. O conjunto das soluções ótimas de Pareto deste problema é dado por

$$P^* = \{x_{l_i} = 0, i = 2, \dots, K + 1, x_{l_1} = x_u, x_u \in [0.5, 1]\}$$

Para este problema, consideramos três instâncias: $K = 1$, $K = 5$ e $K = 13$.

3.5.2 Medidas de desempenho

Para avaliar o algoritmo proposto MO-BLPSO e compará-lo com o algoritmo BLEMO, serão utilizadas duas medidas de desempenho unárias: o indicador de hipervolume (HV) (Fonseca et al., 2006) (Zitzler & Thiele, 1999), e a Distância Geracional Invertida (IGD, cf. seção 2.5.1); e uma medida de desempenho binária: a Cobertura de dois conjuntos (C), que definimos de seguida.

Cobertura de dois Conjuntos (C): Sejam A e B dois conjuntos de soluções a avaliar. A medida de desempenho $C(A, B)$ é definida no intervalo $[0, 1]$ e indica a proporção de pontos em B que são dominados por ou iguais a pontos em A :

$$C(A, B) = \frac{|b \in B : \exists a \in A : a \prec b|}{|B|}, \text{ onde } a \prec b \text{ significa que } a \text{ domina ou é igual a } b.$$

$C(A, B) = 1$ significa que todos os pontos em B são dominados por ou iguais a pontos em A , enquanto $C(A, B) = 0$ significa que nenhum ponto em B é dominado por ou igual a pontos em A .

Considerando as medidas de desempenho unárias, a aproximação da frente ótima de Pareto é tanto melhor quanto menor for o valor IGD e quanto maior for o valor do hipervolume. Para analisar o desempenho de dois algoritmos A e B relativamente à medida de cobertura de dois conjuntos, são calculados os valores $C(A,B)$ para avaliar a qualidade relativa das soluções do algoritmo A e $C(B,A)$ para avaliar a qualidade relativa das soluções do algoritmo B . O algoritmo A obtém melhor desempenho se $C(B,A) < C(A,B)$, caso contrário, se $C(A,B) < C(B,A)$, o melhor desempenho é obtido pelo algoritmo B .

3.5.3 Parametrização dos algoritmos

Os algoritmos foram implementados em Matlab e os testes executados num computador Intel core 2 Duo com processador 2.4 GHz. Foram definidos os seguintes parâmetros para os dois algoritmos MO-BLPSO e BLEMO: $N = 240$, $N_l = 60$, $T_l = 80$ e $T = 100$ para o problema 3.1 e $N = 240$, $N_l = 60$, $T_l = 80$ e $T = 100$ para o problema 3.2. Estes parâmetros foram selecionados por experimentação. Foram efetuadas 10 corridas independentes de cada algoritmo para cada problema.

3.5.4 Resultados

As soluções obtidas (na melhor corrida) pelos algoritmos MO-BLPSO e BLEMO para o problema 3.1 são apresentadas nas Fig. 3.1 e 3.2, respetivamente. Os dois algoritmos fazem uma boa aproximação da verdadeira frente ótima de Pareto do problema estando as soluções bem dispersas ao longo de toda a frente.

Nas Fig. 3.3-3.4 estão representadas as soluções obtidas na melhor corrida para o problema 3.2 (para $K = 1$) pelos algoritmos BLEMO e MO-BLPSO, respetivamente. Podemos observar na Fig. 3.3 que as soluções obtidas pelo algoritmo BLEMO atingem a frente ótima de Pareto do problema. O algoritmo MO-BLPSO consegue igualmente obter uma boa caracterização desta frente (Fig. 3.4), sendo os resultados muito competitivos quando comparados aos obtidos pelo algoritmo BLEMO.

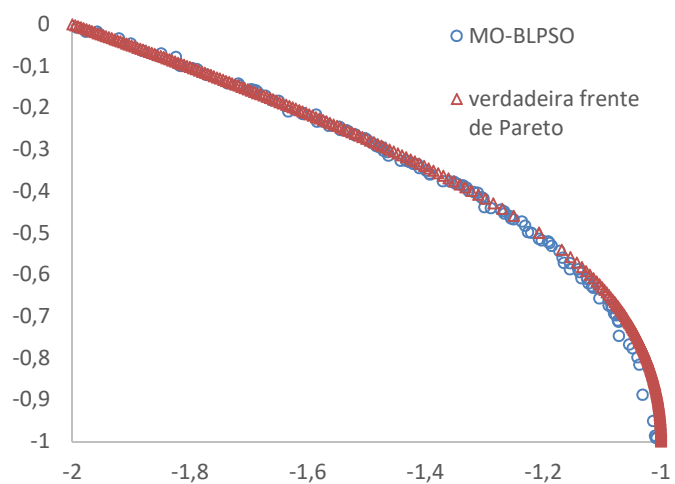


Fig. 3.1 Problema 3.1. Algoritmo MO-BLPSO

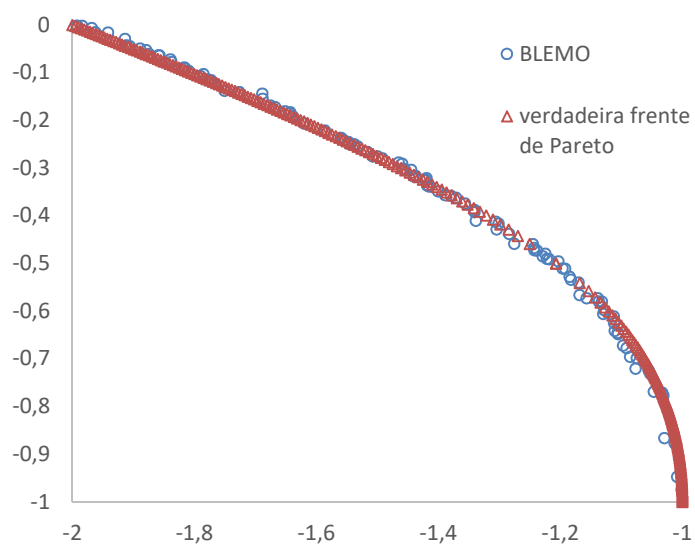


Fig. 3.2 Problema 3.1. Algoritmo BLEMO

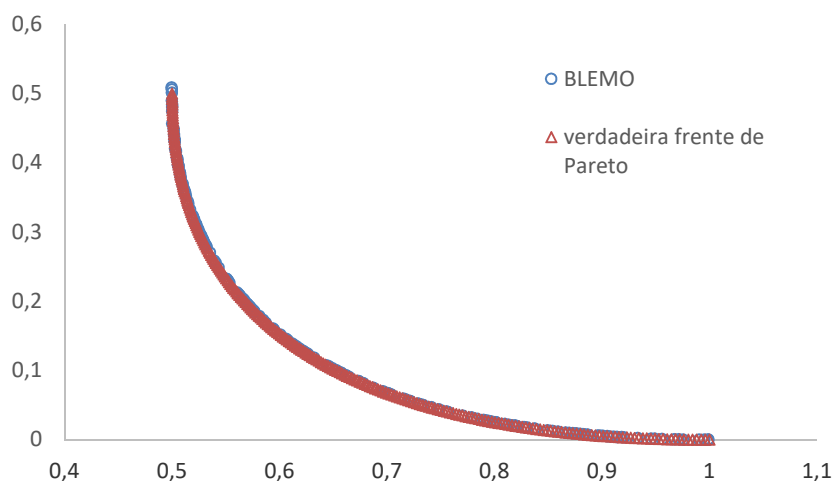


Fig. 3.3 Problema 3.2. Algoritmo BLEMO

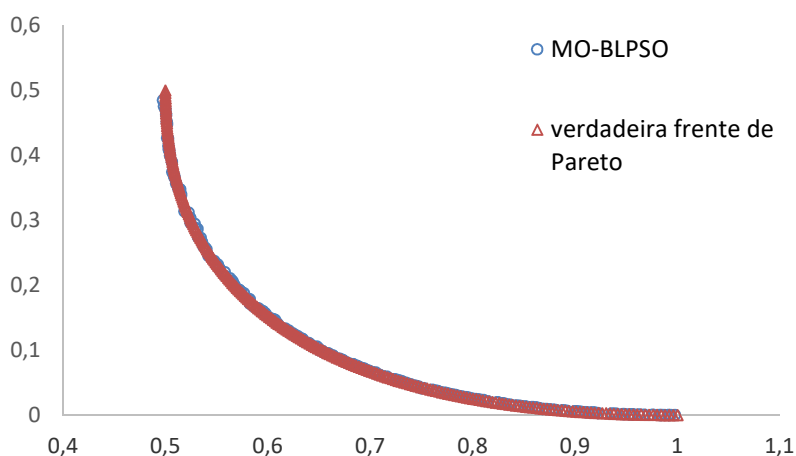


Fig. 3.4 Problema 3.2. Algoritmo MO-BLPSO

Na tabela 3.1 são apresentados os valores das medidas hipervolume e IGD obtidos pelos algoritmos MO-BLPSO e BLEMO para todas as instâncias consideradas dos problemas 3.1 e 3.2. Os algoritmos têm um desempenho semelhante, segundo estas medidas. Observando os valores IGD podemos concluir que os dois algoritmos obtêm soluções bem distribuídas ao longo de toda a frente ótima de Pareto em todas as instâncias testadas. O algoritmo MO-BLPSO obteve valores de hipervolume ligeiramente melhores em três das quatro instâncias testadas e o algoritmo BLEMO obteve melhores

valores de IGD em todas as instâncias. Em ambas as medidas de desempenho os valores dos algoritmos MO-BLPSO e BLEMO estão bastante próximos.

Tabela 3.1 Valor médio do hipervolume e IGD obtido pelos algoritmos MO-BLPSO e BLEMO.

Problema	MO-BLPSO		BLEMO	
	Hipervolume	IGD	Hipervolume	IGD
Problema 1	0.3166	0.01403	0.3064	0.01024
Problema 2 (K=1)	0.2073	0.00442	0.2071	0.00421
Problema 2 (K=5)	0.2062	0.01425	0.2049	0.00985
Problema 2 (K=13)	0.2030	0.02634	0.2054	0.00847

Adicionalmente comparamos o desempenho dos dois algoritmos utilizando a Cobertura de dois conjuntos. Os valores obtidos por esta medida de desempenho são apresentados na Tabela 3.2. Verificamos que apenas um pequeno número de soluções do algoritmo MO-BLPSO são dominadas por ou iguais a soluções obtidas pelo algoritmo BLEMO. O algoritmo BLEMO revela um desempenho muito inferior segundo esta medida em qualquer das instâncias consideradas. No problema 3.1, verificamos que, em média, 65% das soluções obtidas pelo algoritmo BLEMO são dominadas por ou iguais a soluções obtidas pelo algoritmo MO-BLPSO, enquanto que para o mesmo problema a percentagem média de soluções obtidas pelo algoritmo MO-BLPSO que são dominadas por ou iguais a soluções obtidas pelo algoritmo BLEMO é de apenas 18%.

Tabela 3.2 Valor médio da Cobertura de dois conjuntos aplicada aos algoritmos BLEMO e MO-BLPSO.

Problema	C(BLEMO, MO-BLPSO)	C(MO-BLPSO, BLEMO)
Problema 3.1	0.18	0.65
Problema 3.2 (K=1)	0.08	0.46
Problema 3.2 (K=5)	0.06	0.82
Problema 3.2 (K=13)	0.10	0.82

Esta medida permite concluir que, embora os conjuntos de soluções obtidas pelos dois algoritmos forneçam boas aproximações da frente ótima de Pareto dos problemas, cuja diferença de qualidade não é perceptível ou é muito pequena nos gráficos ou nas medidas de HV e IGD, as soluções obtidas pelo MO-BLPSO aproximam-se mais das

soluções ótimas de Pareto, dominando uma grande percentagem de soluções obtidas pelo BLEMO, enquanto que a relação inversa se verifica numa percentagem muito inferior.

3.6 Considerações finais

Neste capítulo desenvolvemos um algoritmo de otimização multiobjetivo baseado em enxame de partículas para resolver o problema de otimização em dois níveis com múltiplas funções objetivo em ambos os níveis. A principal dificuldade na resolução destes problemas resulta de existirem soluções para o problema BLMOP que contenham soluções dominadas para o nível inferior, as quais podem dominar soluções que contenham verdadeiras soluções ótimas de Pareto nos dois níveis. As primeiras, ao serem aceites para entrar no arquivo não dominado do nível superior, apesar de não serem admissíveis para o problema BLMOP, impedem a entrada no arquivo das verdadeiras soluções ótimas de Pareto. Para evitar esta situação, as soluções do nível inferior devem ser calculadas com muita precisão, tentando garantir que são de facto soluções ótimas de Pareto para este problema. O algoritmo MO-BLPSO demonstrou capacidade para ultrapassar esta dificuldade na resolução dos problemas teste. Os resultados obtidos por este algoritmo são semelhantes aos do algoritmo BLEMO, representativo do estado da arte. Até ao momento, existia apenas um algoritmo PSO para resolver problemas BLMOP, mas que revelou um mau desempenho em algumas experiências efetuadas, não sendo capaz de ultrapassar as dificuldades mencionadas acima. Este facto motivou-nos para desenhar um algoritmo baseado em PSO para resolver problemas BLMOP tendo em conta essas dificuldades. Verificamos que o algoritmo MO-BLPSO tem um desempenho similar ao algoritmo BLEMO (baseado no algoritmo NSGA-II), se considerarmos as medidas de desempenho hipervolume e IGD. Tendo em conta este resultado, utilizámos também a medida de desempenho Cobertura de dois conjuntos e verificámos que, segundo esta medida, o algoritmo MO-BLPSO tem um desempenho muito superior ao algoritmo BLEMO.

Nos próximos capítulos vamos considerar um problema relativo à interação entre retalhista e consumidor de energia elétrica formulado como um problema de otimização em dois níveis. Os algoritmos MO-BLPSO e BLEMO serão utilizados para resolver o problema apresentado, adaptando o seu funcionamento e mecanismos à estrutura do problema.

4 Um modelo BL para o estudo das interações retalhista – consumidor de energia elétrica

4.1 Introdução

No capítulo anterior desenvolvemos um algoritmo para resolução de problemas de otimização em dois níveis. Estes problemas surgem em diferentes domínios, sempre que exista uma relação hierárquica envolvendo dois decisores. No mercado de energia elétrica podemos encontrar esta relação entre o retalhista e o consumidor. Considerando que estes intervenientes pretendem otimizar os respetivos proveitos e custos, podemos modelar o seu relacionamento através de um problema de otimização em dois níveis. Desenvolvemos neste capítulo um modelo para este problema, o qual será tratado utilizando dois algoritmos, um evolucionário e outro PSO, baseados nos algoritmos apresentados no capítulo 3 (Carrasqueira et al., 2017). Este modelo é ainda reescrito de forma equivalente num modelo BL em que o problema do nível inferior é transformado num problema de otimização linear inteira mista (MILP – *mixed integer linear problem*), podendo ser resolvido por um método de programação matemática que determina a solução exata do problema para cada instanciação das variáveis do nível superior. No entanto, a dimensão do modelo aumenta muito o número de variáveis, pelo que a aplicação deste método está limitada a dimensões pequenas do problema, acima das quais não será possível aplicá-lo. Os resultados obtidos permitirão avaliar melhor o desempenho dos algoritmos que usam apenas meta-heurísticas.

Neste capítulo considera-se um modelo BL mono-objetivo, em que o objetivo do retalhista é a maximização do lucro e o objetivo do consumidor é a minimização do custo com a eletricidade. No capítulo seguinte será feita uma extensão deste modelo para um modelo semi-vetorial em que o consumidor pretende não só minimizar o custo como também minimizar o possível desconforto associado à deslocação no tempo do período de funcionamento dos seus equipamentos elétricos.

4.2 Enquadramento do problema e abordagens

A cadeia de abastecimento de eletricidade inclui unidades de geração, operadores da rede de transporte e distribuição, e comercializadores (retalhistas) que fornecem os consumidores. Tipicamente os retalhistas compram eletricidade no mercado grossista e vendem depois a eletricidade a consumidores finais para operar uma diversidade de equipamentos preenchendo as necessidades de serviços de energia. Em geral, os preços no mercado grossista são variáveis e definidos por leilões ou por contratos bilaterais. Os mercados grossistas de eletricidade podem incluir mercados de futuros para aquisição e geração futura, mercados com base na procura (*spot*, dia seguinte) com uma componente de contratação diária e uma componente intra-diária de ajustamento, e mercado de serviços auxiliares (tempo real) lidando com o equilíbrio da geração e do consumo.

Os preços de eletricidade analisados do ponto de vista dos consumidores podem ser estáticos ou dinâmicos. Preços estáticos incluem tarifas planas (preço €/kWh uniforme, por vezes com uma componente de potência contratada €/kW) nas quais o retalhista assume o risco da variação dos preços do mercado, e tarifas com diferenciação horária (ToU – *time-of-use*) nas quais os preços são estabelecidos antecipadamente para períodos de tempo específicos, incluindo *critical peak pricing* (CPP) no qual o retalhista pode esporadicamente estabelecer um preço muito elevado para um período limitado de tempo. Assim, é possível enviar sinais económicos em períodos de quebra de geração ou quando a segurança do sistema está comprometida e mesmo evoluir para preços em tempo real refletindo o custo atual no consumo real nesse período. Esquemas de preços ToU, CPP e preço tempo real (RTP – *real time pricing*) requerem contadores de eletricidade tecnologicamente mais avançados, cuja instalação em larga escala é esperada no âmbito da implementação das redes inteligentes (*smart grids*).

O mercado retalhista de eletricidade desenvolveu-se de forma a satisfazer as necessidades dos consumidores, funcionando num único sentido: retalhista → consumidor. Neste modelo o retalhista fornece a energia elétrica ao consumidor a um preço fixo e não tem conhecimento antecipado de quando e como o cliente utiliza a energia. Este sistema torna-se desadequado para representar a evolução esperada do mercado de eletricidade tendo ainda em conta a penetração crescente de fontes renováveis. No mercado liberalizado o retalhista adquire a energia a preços variáveis dependentes do período em que a energia é fornecida. Por sua vez, a atual estrutura tarifária não permite que o retalhista reflita completamente para o consumidor o

preço real da energia de uma forma eficiente, ou seja, cobrar ao consumidor o custo real da energia. Outro inconveniente do modelo atualmente em prática é a existência de períodos de picos de consumo, que podem obrigar à utilização de centrais de geração mais dispendiosas e provocar congestão das infraestruturas de transporte e distribuição. Considerando a tendência de aumento de consumo de energia elétrica decorrente de vários fatores derivados do desenvolvimento económico e tecnológico, incluindo a previsível disseminação de veículos elétricos, estas ineficiências do sistema tornam-se particularmente relevantes. O modelo atual não fornece ao consumidor o estímulo tarifário necessário para que este altere os seus padrões de consumo de acordo com a flexibilidade de uso dos seus equipamentos. Se os consumidores fossem sujeitos a preços de energia variáveis ao longo do dia de alguma forma refletindo os custos de geração, isto é, se fossem oferecidas tarifas dinâmicas, os consumidores possivelmente adotariam ações para calendarizar o seu consumo, designadamente através de sistemas automáticos de gestão de energia, de forma a minimizar a sua conta de eletricidade tendo em conta a qualidade de serviço associada à utilização dos equipamentos, nomeadamente requisitos de conforto. O aproveitamento da flexibilidade que os consumidores geralmente têm na calendarização das suas cargas seria do maior interesse dos diversos intervenientes na cadeia de abastecimento da eletricidade. Os consumidores poderiam ver a sua conta de eletricidade diminuir e os retalhistas poderiam tomar decisões mais acertadas relativamente à compra e venda de eletricidade. Outras empresas intervenientes, nos níveis da distribuição e do transporte, seriam também potenciais beneficiários porque poderiam usar os recursos libertados para uma melhor gestão das redes e da disponibilidade da energia distribuída baseada nas fontes renováveis.

Retalhistas e consumidores têm interesses conflitantes. Os retalhistas pretendem maximizar os seus lucros pela venda de eletricidade e os consumidores pretendem minimizar os seus custos garantindo os requisitos de qualidade do serviço de energia associado com o funcionamento dos equipamentos nos períodos de tempo apropriados. Além do conflito entre os objetivos de cada interveniente, existe uma relação hierárquica entre retalhistas e consumidores dado que os primeiros determinam os preços e os últimos reagem calendarizando os seus consumos tendo em conta os preços definidos. Esta interação pode ser modelada através de um problema de otimização em dois níveis (BL).

As capacidades das tecnologias de informação e comunicação associadas à evolução do sistema elétrico para redes inteligentes permitem a interação bi-direcional entre retalhistas e consumidores. Deste modo, o retalhista pode enviar a informação sobre

preços de eletricidade variáveis, recebendo por seu turno a resposta do consumidor através do ajustamento da calendarização da utilização dos seus equipamentos com o intuito de minimizar a sua conta de eletricidade.

Têm sido propostos diversos modelos na literatura considerando a gestão do lado da procura (*Demand-Side Management* - DSM) no mercado residencial (Soares et al., 2014). A DSM tem muitos efeitos benéficos, possibilitando uma melhor utilização da capacidade de geração de energia, incluindo facilitar a integração de fontes renováveis, e das infraestruturas de rede, contribuindo para evitar ou adiar novos investimentos, diminuindo os picos de consumo, reduzindo os níveis de emissões de carbono e melhorando a sustentabilidade geral do sistema. Os programas DSM têm reemergido no contexto das redes inteligentes, permitindo aos utilizadores finais redefinir os seus padrões de consumo de energia e obter vantagem das tarifas dinâmicas. Tradicionalmente estes programas incluem o controlo direto das cargas e a deslocação temporal de cargas, e devem ser concebidos para induzir a aceitação por parte do consumidor.

Têm sido propostas várias abordagens utilizando estratégias de preços variáveis para diminuir os picos de consumo. Na estratégia de preços horários do dia seguinte, os consumidores recebem no dia anterior, ou algumas horas antes, os preços para as próximas 24 horas. Os consumidores devem então reagir calendarizando a utilização dos seus equipamentos, alterando o seu perfil de consumo diferindo ou antecipando o momento de uso de alguns equipamentos para obter um equilíbrio satisfatório entre a minimização da conta de eletricidade e a maximização do bem estar em termos de conforto associado com os serviços de energia fornecidos (aquecimento de água, lavagem e secagem de roupa, carga do veículo elétrico, etc.). Alguns autores modelam a interação entre retalhistas e consumidores utilizando modelos de otimização BL procurando encontrar uma solução de equilíbrio entre esses aspetos de avaliação. Em Meng & Zeng (2013) o problema BL é convertido num modelo num só nível com restrições de complementaridade resultantes das condições de Karush-Kuhn-Tucker e é utilizado um algoritmo de planos de corte para resolver o problema. Zugno et al. (2013) propõem um modelo BL no qual o objetivo do retalhista é a maximização do lucro e o objetivo do consumidor é a minimização do custo e do desconforto, sendo as funções objetivo e as restrições do retalhista e do consumidor lineares. No nível superior (retalhista) são utilizadas variáveis estocásticas para determinar os preços no mercado grossista enquanto no nível inferior (consumidor) são considerados equipamentos flexíveis e equipamentos inflexíveis; o desconforto é medido face a uma amplitude da temperatura interior

considerada confortável. Os autores transformam o modelo BL num modelo MILP, que é resolvido na plataforma GAMS pelo *solver* CPLEX. Zhang et al. (2011) apresentam um modelo BL com múltiplas funções objetivo no nível superior para maximizar o lucro das empresas fornecedoras de energia. O consumidor pode escolher a empresa fornecedora procurando minimizar a sua conta de eletricidade. O problema BL é resolvido utilizando a abordagem PSO em ambos os níveis. No nível inferior o consumidor determina a quantidade de energia a adquirir a cada retalhista, tendo por base os preços oferecidos por estes. No nível superior cada um dos retalhistas otimiza a sua função objetivo, tendo conhecimento das decisões do consumidor e mantendo as variáveis relativas aos restantes retalhistas inalteradas. Cada retalhista determina o desvio entre o lucro obtido com a sua estratégia ótima e a estratégia atual. O algoritmo termina quando a soma dos desvios de todos os retalhistas for inferior a uma determinada tolerância pré-definida. Sekizaki et al. (2016) propõem um modelo BL incluindo a tarifa de rede na avaliação do lucro esperado pelo retalhista e o custo para os consumidores resultante da não utilização dos seus equipamentos, que é proporcional à quantidade de carga reduzida face ao nível ideal (i.e. aquele que maximiza a sua utilidade). Para cada tipo de consumidor (residencial, comercial, industrial e um tipo misto resultante da combinação dos anteriores) é estabelecida uma função objetivo que resulta da soma do custo de aquisição de energia ao retalhista com a “desutilidade” associada à redução do consumo, que é definida individualmente para cada tipo de carga. Estes autores consideram também diferentes cenários para tratar a incerteza dos preços no mercado grossista. O modelo BL é resolvido através de um algoritmo genético no nível superior e de um *solver* de programação linear no nível inferior. Meng e Zeng (2014) consideram cargas não deslocáveis, interruptíveis, não interruptíveis e cargas cujo consumo pode ser reduzido. Estes autores consideram a relação entre um retalhista e N consumidores, modelada por um problema BL multi-seguidor. O problema do retalhista é resolvido por um algoritmo genético. O problema do consumidor é linear e é resolvido de forma exata por um *solver* de PL. Meng e Zeng (2015) consideram as cargas do consumidor relativas a equipamentos interruptíveis e não interruptíveis. É desenvolvido um modelo de aprendizagem para identificar o padrão de utilização de cada equipamento. A resposta dos consumidores pode ser obtida através de um sistema programado automaticamente ou pode ser inserida manualmente pelos próprios consumidores. O problema é formulado através de um modelo BL, sendo o problema do nível superior resolvido através de algoritmos genéticos e o problema do nível inferior por um método de programação matemática. Em Bu et al. (2011) o

retalhista adquire energia de duas fontes distintas, decidindo a quantidade de energia a adquirir de cada opção considerando diferentes cenários. O objetivo do consumidor é maximizar a sua função utilidade baseada na quantidade de energia adquirida e no preço pago. Em Yang et al. (2012) o objetivo do consumidor é uma função de satisfação relativa à diferença entre carga efetiva e nominal definindo uma estratégia de preços variáveis e comparando com a estratégia de preços ToU e com uma tarifa fixa. O problema é resolvido por uma técnica de programação matemática, otimizando a função objetivo do retalhista a partir do conhecimento da função utilidade do consumidor. Lu et al. (2016) apresentam um resumo dos problemas de otimização multi-nível relacionados com tomada de decisão envolvendo hierarquias, em diversos domínios. O resumo inclui a descrição de diferentes tipos de problemas BL e aplicações práticas incluindo estratégias de oferta em mercados de eletricidade.

Para obter uma caracterização mais realista do consumo de eletricidade devem ser definidos os ciclos de funcionamento típicos dos equipamentos controláveis. Neste sentido, Soares et al. (2014) consideram os principais equipamentos geralmente existentes numa habitação e apresentam um ciclo de operação específico para cada equipamento, estabelecendo o seu potencial para ser controlado.

No nosso trabalho vamos formular a interação entre retalhistas (que estabelecem tarifas dinâmicas) e consumidores (que ajustam o consumo modificando a calendarização habitual do funcionamento dos equipamentos) através de um problema de otimização BL. No nível superior, a função objetivo é a maximização do lucro do retalhista e no nível inferior a função objetivo é a minimização da conta de eletricidade do consumidor. Começamos por aplicar um algoritmo híbrido (H-BLEA), transformando o problema do nível inferior num problema MILP e resolvendo-o de forma exata. O problema do nível superior é resolvido utilizando um algoritmo evolucionário e é utilizado o *solver* MILP CPLEX para obter a solução ótima (exata) para o problema do nível inferior para cada conjunto de valores das variáveis do nível superior (preços da eletricidade durante o período de planeamento). Seguidamente apresentamos dois algoritmos BL baseados em populações, um baseado num algoritmo evolucionário (BLEA) e o outro na otimização com enxame de partículas (BLPSO) para resolver o problema BL através de meta-heurísticas. Dado que os consumidores têm diferentes preferências relativamente à aceitação de um possível desconforto associado à alteração do período de funcionamento dos equipamentos face aos padrões habituais, são definidas janelas temporais admissíveis para cada equipamento. São considerados três perfis para modelar diferentes tipos de

consumidores, o que é refletido em restrições mais ou menos estritas do problema do nível inferior relativamente aos períodos de tempo aceitáveis para funcionamento dos equipamentos e, assim, na dificuldade em determinar soluções ótimas para este problema (i.e., soluções admissíveis para o problema BL). É considerado um perfil base representando os períodos habituais de funcionamento de cada equipamento. Os outros dois perfis consideram períodos de funcionamento mais reduzidos e períodos mais alargados. Os resultados dos três algoritmos (H-BLEA, BLPSO e BLEA) são comparados para os três perfis de consumidores.

4.3 Modelo BL para o problema do mercado retalhista de eletricidade

O retalhista e os consumidores têm interesses opostos: o retalhista pretende maximizar o lucro e os consumidores pretendem minimizar a conta de eletricidade tendo em consideração as restrições de conforto associadas aos períodos de tempo para operação dos equipamentos (cargas). Modelámos esta relação como um problema BL onde o retalhista é o líder e o consumidor o seguidor. O período de planeamento é dividido em sub-períodos $P_i (i = 1, \dots, I)$ e o retalhista determina o preço da eletricidade x_i para cada sub-período i . Estes preços definidos pelo retalhista são limitados a valores mínimos (\underline{x}_i) e máximos (\bar{x}_i) e é imposto um preço médio (x^{AVG}) relativo a todo o período de planeamento. O consumidor tem uma carga base relativa a equipamentos não controláveis e um conjunto de equipamentos cujo ciclo de operação pode ser estabelecido em intervalos de tempo aceitáveis especificados de acordo com as suas preferências, hábitos e requisitos de conforto. Em reação aos preços de energia, o consumidor pretende determinar o tempo (z_j) em que cada equipamento j deve iniciar o seu ciclo de funcionamento para minimizar a conta de eletricidade assegurando que todo o ciclo está dentro do correspondente intervalo de tempo de conforto T_j .

Notação

T = número de intervalos (minutos, quartos de hora, meias-horas ou outros períodos de tempo) em que o período de planeamento é dividido ($t = 1, \dots, T$). Seja $T = \{1, \dots, T\}$.

J = número de equipamentos cuja utilização pode ser calendarizada pelo sistema de gestão de energia do consumidor ($j = 1, \dots, J$).

C_t = potência contratada no tempo t do período de planeamento (kW).

π_t = preço da energia no tempo t do período de planeamento (€/kWh), onde h é a duração de um intervalo como uma fração de uma hora (para permitir a discretização mais conveniente do período de planeamento tendo em conta o problema concreto).

b_t = carga base não controlável no tempo t do período de planeamento (kW).

d_j = duração (número de intervalos de tempo) do ciclo de operação da carga j .

$g_j(r)$ = potência requerida pela carga j no tempo r do seu ciclo de operação ($r = 1, \dots, d_j$) (kW).

$T_j = [T1_j, T2_j] \subseteq T$: intervalo de tempo no qual a carga j pode operar.

I = número de sub-períodos de tempo $P_i \subset T$ nos quais o retalhista pode definir diferentes preços de eletricidade (ToU) ($i=1, \dots, I$).

$P1_i, P2_i$: intervalos de tempo que delimitam cada sub-período $P_i, i = 1, \dots, I$, tais que

$P_i = [P1_i, P2_i]$ e $\bigcup_{i=1}^I P_i = T$. \bar{P}_i denota a amplitude de P_i , i.e. $\bar{P}_i = P2_i - P1_i + 1$.

\bar{x}_i = preço máximo no sub-período P_i (€/kWh).

\underline{x}_i = preço mínimo no sub-período P_i (€/kWh).

x^{AVG} = preço médio no período de planeamento (€/kWh).

Variáveis de decisão do nível superior:

x_i = preço durante o sub-período P_i (€/kWh); $i=1, \dots, I$.

Variáveis de decisão do nível inferior:

z_j = tempo de início do ciclo de funcionamento da carga j ; $j = 1, \dots, J$.

Variáveis auxiliares do nível inferior:

u_{jt} = variável binária que indica se o ciclo de operação da carga j está “on” ou “off” no tempo t do período de planeamento; $j = 1, \dots, J$; $t = 1, \dots, T$.

p_{jt} = potência pedida à rede pela carga j no tempo t do período de planeamento (kW);
 $j = 1, \dots, J$; $t = 1, \dots, T$.

No Modelo BL1, a função objetivo do decisor do nível superior (4.1) é o lucro do retalhista (receitas provenientes da venda de eletricidade aos consumidores menos o custo de aquisição da energia). As variáveis x e z , variáveis de decisão do nível superior e inferior neste modelo, correspondem às variáveis x_u e x_l na formulação BL (secção 3.2), respetivamente. As restrições (4.2 – 4.4) definem os limites superior e inferior para os preços de eletricidade em cada sub-período P_i e definem um preço médio no período de planeamento T . A expressão (4.5) é a função objetivo do consumidor, que pretende minimizar a conta de eletricidade (para abastecer as cargas não controláveis e as deslocáveis). A expressão (4.6) define o valor das variáveis binárias auxiliares u_{jt} em função das variáveis de decisão z_j , no período de operação da carga j . As variáveis u_{jt} são usadas em (4.7) para obter o valor da potência requerida à rede pela carga j em cada tempo t do período de planeamento, i.e. o tempo $r = t - z_j + 1$ do ciclo de operação da carga iniciado em z_j . As restrições (4.8) impõem que a potência contratada nunca seja excedida. As restrições (4.9) definem os intervalos de tempo onde é permitida a operação de cada carga j de acordo com as preferências do consumidor.

Nota: se existirem diferentes intervalos disjuntos possíveis para a operação da carga j , as restrições (4.9) devem ser substituídas por restrições disjuntas tais como $(T1_j^1 \leq z_j \leq T2_j^1 - d_j + 1) \vee (T1_j^2 \leq z_j \leq T2_j^2 - d_j + 1) \vee \dots$, onde $[T1_j^k, T2_j^k]$, $k = 1, 2, \dots$ são os diferentes intervalos de tempo nos quais a carga j pode funcionar.

Modelo BL 1

$$\max F = \sum_{i=1}^I x_i \sum_{t \in P_i} \left(b_t + \sum_{j=1}^J p_{jt} \right) - \sum_{t=1}^T \pi_t \left(b_t + \sum_{j=1}^J p_{jt} \right) \quad (4.1)$$

s. a

$$x_i \leq \bar{x}_i \quad i=1, \dots, I \quad (4.2)$$

$$x_i \geq \underline{x}_i \quad i=1, \dots, I \quad (4.3)$$

$$\frac{1}{T} \sum_{i=1}^I \bar{P}_i x_i = x^{AVG} \quad (4.4)$$

$$\min f = \sum_{i=1}^I x_i \sum_{t \in P_i} \left(b_t + \sum_{j=1}^J p_{jt} \right) \quad (4.5)$$

s.a.

$$u_{jt} = \begin{cases} 1 & \text{se } z_j \leq t \leq z_j + d_j - 1 \\ 0 & \text{caso contrário} \end{cases} \quad j=1, \dots, J; t=1, \dots, T \quad (4.6)$$

$$p_{jt} = g_j (t - z_j + 1) u_{jt} \quad j=1, \dots, J; t=1, \dots, T \quad (4.7)$$

$$\sum_{j=1}^J p_{jt} + b_t \leq C_t \quad t=1, \dots, T \quad (4.8)$$

$$T1_j \leq z_j \leq T2_j - d_j + 1 \quad j=1, \dots, J \quad (4.9)$$

O Modelo BL 1 pode ser escrito de uma forma equivalente, reformulando o problema do nível inferior como um problema MILP. Assim, apresentamos o Modelo BL 2, que é um modelo de programação matemática equivalente ao Modelo BL 1. Neste novo modelo não são consideradas as variáveis $z_j, j=1, \dots, J$ (que especificam o tempo t de início do ciclo de funcionamento do equipamento j), sendo utilizadas variáveis binárias w_{jrt} que indicam se o equipamento j está “on” ou “off” no tempo t do período de planeamento e está no tempo r do seu ciclo de operação. As variáveis u_{jt} já não são

necessárias pois podem ser obtidas a partir de $w_{jrt} : u_{jt} = \sum_{r=1}^{d_j} w_{jrt}$. As variáveis do nível superior (x_i) do Modelo BL 1 mantêm-se inalteradas no Modelo BL 2.

Notação

Os dados são os mesmos do Modelo BL 1, com uma diferença: cada função $g_j(r)$ é substituída pela correspondente série de valores g_{jr} para cada tempo r do ciclo de operação da carga j :

g_{jr} = potência requerida pelo equipamento j no tempo r do seu ciclo de operação ($r = 1, \dots, d_j$) (kW).

Para evitar ambiguidade entre pontos no tempo do ciclo de operação de cada equipamento e pontos no tempo do período de planeamento, referimo-nos ao tempo r do ciclo de operação como estado r .

Variáveis de decisão do nível inferior:

w_{jrt} = variável binária indicando se o equipamento j está “on” (1) ou “off” (0) no tempo t do período de planeamento e no estado r do seu ciclo de operação.

Para não aumentar desnecessariamente o número de variáveis w_{jrt} , estas variáveis são definidas apenas para t no período de tempo permitido para a operação de cada equipamento. Assim, as variáveis w_{jrt} são definidas para

$$j = 1, \dots, J, r = 1, \dots, d_j, t = T1_j, \dots, T2_j.$$

Variáveis auxiliares do nível inferior:

p_{jt} = potência requerida à rede pela carga j no tempo t do período de planeamento (kW).

Modelo BL 2

$$\max F = \sum_{i=1}^I x_i \sum_{t \in P_i} \left(b_t + \sum_{j=1}^J p_{jt} \right) - \sum_{t=1}^T \pi_t \left(b_t + \sum_{j=1}^J p_{jt} \right) \quad (4.10)$$

s. a

$$x_i \leq \bar{x}_i, i = 1, \dots, I \quad (4.11)$$

$$x_i \geq \underline{x}_i, i = 1, \dots, I \quad (4.12)$$

$$\frac{1}{T} \sum_{i=1}^I \bar{P}_i x_i = x^{AVG} \quad (4.13)$$

$$\min f = \sum_{i=1}^I x_i \sum_{t \in P_i} \left(b_t + \sum_{j=1}^J p_{jt} \right) \quad (4.14)$$

s.a

$$p_{jt} = \sum_{r=1}^{d_j} g_{jr} w_{jrt}, j = 1, \dots, J; t = T1_j, \dots, T2_j \quad (4.15)$$

$$p_{jt} = 0, \quad j = 1, \dots, J; t < T1_j \vee t > T2_j \quad (4.16)$$

$$\sum_{j=1}^J p_{jt} + b_t \leq C_t \quad t = 1, \dots, T \quad (4.17)$$

$$\sum_{r=1}^{d_j} w_{jrt} \leq 1, \quad j = 1, \dots, J; t = T1_j, \dots, T2_j \quad (4.18)$$

$$w_{j(r+1)(t+1)} \geq w_{jrt}, j = 1, \dots, J; r = 1, \dots, d_j - 1; t = T1_j, \dots, T2_j - 1 \quad (4.19)$$

$$\sum_{t=T1_j}^{T2_j} w_{jrt} = 1, \quad j = 1, \dots, J; r = 1, \dots, d_j \quad (4.20)$$

$$\sum_{t=T1_j}^{T2_j - d_j + 1} w_{j1t} \geq 1, \quad j = 1, \dots, J \quad (4.21)$$

$$w_{jrt} \in \{0,1\}, \quad j = 1, \dots, J; r = 1, \dots, d_j; t = T1_j, \dots, T2_j \quad (4.22)$$

$$p_{jt} \geq 0, \quad j = 1, \dots, J; t = 1, \dots, T \quad (4.23)$$

O problema do nível superior (4.10) – (4.13) e a função objetivo do nível inferior (4.14) mantêm-se os mesmos do Modelo BL 1.

As restrições (4.15) – (4.16) correspondem à restrição (4.7) no Modelo BL 1 para definir as variáveis auxiliares p_{jt} . Dado que estas variáveis são definidas para $t = 1, \dots, T$, estas restrições compreendem dois grupos: (4.15) que define p_{jt} para t no período de tempo permitido para funcionamento do equipamento j (para o qual as variáveis w_{jrt} estão definidas) e (4.16) para t fora deste período de tempo onde p_{jt} é sempre zero.

As restrições (4.17) são iguais às restrições (4.8) no Modelo BL 1.

As restrições (4.18) asseguram que, no tempo t do período de planeamento, cada equipamento j ou está desligado ou está ligado num único estado r do seu ciclo de funcionamento.

As restrições (4.19) asseguram, para cada equipamento j , que se este estiver ligado no tempo t e no estado $r \leq d_j - 1$ do seu ciclo de funcionamento, então deve estar também ligado no tempo $t + 1$ e no estado $r + 1$.

As restrições (4.20) asseguram que cada equipamento j está em funcionamento no estado r exatamente uma única vez.

Note-se que as restrições (4.19) não evitam que um equipamento j comece a funcionar num de tempo posterior a $T2_j - d_j + 1$ e, dado que não consegue terminar até $T2_j$, continuaria a partir de $T1_j$. Por exemplo, consideremos que um equipamento j está no estado $r = 1$ em $t = T2_j - 1$, $r = 2$ em $t = T2_j$ e então passaria para $r = 3$ em $t = 1$, $r = 4$ em $t = 2$, etc.; este esquema de funcionamento não é admissível na prática mas não viola as restrições (4.19). Assim, são impostas as restrições (4.21) que asseguram que cada equipamento j começa o seu funcionamento (estado $r = 1$) no máximo no tempo $T2_j - d_j + 1$ de forma a que possa terminar até $T2_j$. As restrições (4.19) juntamente com as restrições (4.20) e (4.21) asseguram que o equipamento j está em funcionamento exatamente d_j intervalos de tempo consecutivos, forçando w_{jrt} a ser zero quando o equipamento j está desligado.

4.4 Algoritmo híbrido evolucionário com *solver* MILP: H-BLEA

O algoritmo híbrido proposto, H-BLEA, utiliza um algoritmo evolucionário e um *solver* MILP para resolver o problema BL formulado no Modelo BL 2.

O algoritmo evolucionário é aplicado ao problema do nível superior (4.10) – (4.13). Cada indivíduo da população representa um conjunto de preços de eletricidade $x' = (x'_1, x'_2, \dots, x'_I)$. Para cada x' é determinada a solução ótima (exata) do problema do nível inferior (4.14) – (4.23) com $x = x'$. Seja y' a solução ótima obtida para esta instância do nível inferior, com y' dado pelos valores das variáveis de decisão do nível inferior, w_{jrt} e p_{jt} . Cada solução (x', y') para o problema BL é então avaliada pela função objetivo do nível superior F em (4.10).

O problema do nível inferior foi modelado usando a linguagem AMPL e o algoritmo evolucionário foi implementado na linguagem Matlab. Para cada indivíduo x' , o problema MILP do nível inferior é resolvido de forma exata utilizando o *solver* CPLEX incorporado no sistema AMPL, que por sua vez é chamado a partir do algoritmo evolucionário. Os preços de eletricidade (x_i) são os únicos parâmetros que mudam de uma chamada para outra. A descrição do algoritmo evolucionário é apresentada no Algoritmo 4.1.

Algoritmo 4.1

Algoritmo evolucionário – pseudo-código

Inicializa uma população Pop de N indivíduos $x' = (x'_1, x'_2, \dots, x'_l)$ satisfazendo as restrições (4.11) – (4.13)

Para cada indivíduo $x' \in Pop$, resolve o problema do nível inferior (4.14) – (4.23) com $x = x'$ usando o *solver* MILP. Seja y' a solução obtida.

Avalia cada solução (x', y') para o problema BL calculando $F(x', y')$

Enquanto o critério de paragem não é atingido

Repete

Seleciona dois progenitores x' e x'' de Pop , um aleatoriamente e outro por torneio binário decidido por F

$x^c \leftarrow$ cruzamento de x' e x''

$x^m \leftarrow$ mutação de x^c

Procedimento de reparação (x^m)

Até terem sido gerados N descendentes que formam o conjunto $Desc$

Para cada x^m em $Desc$, resolve o problema do nível inferior (4.14) – (4.23) com

$x = x^m$ usando o *solver* MILP. Seja y^m a solução ótima obtida

Avalia cada solução (x^m, y^m) calculando $F(x^m, y^m)$

Cria $NextPop$ copiando a melhor solução obtida até ao momento (pode estar em Pop ou em $Desc$) e efetua $N - 1$ torneios binários sem reposição entre indivíduos de $Desc$ e Pop . Atualiza a população Pop para a próxima geração com $NextPop$

Termina Enquanto

Devolve a solução $(x', y') \in Pop$ com melhor F

Algoritmo 4.2

Procedimento de reparação (x) – pseudo-código

$$s = \sum_{i=1}^I \bar{P}_i x_i$$

Seja $A = \{1, 2, \dots, I\}$, o conjunto de índices i de x que podem ser modificados

Enquanto ($s \neq Tx^{AVG}$) e ($A \neq \phi$) **Faz**

$$\Delta = Tx^{AVG} - s$$

$$P = \sum_{i \in A} \bar{P}_i$$

Para cada $x_i, i \in A$

$$x_i \leftarrow x_i + \Delta / P$$

Termina Para

Para $i = 1$ até I

Se $x_i < \underline{x}_i$ **então**

$$x_i \leftarrow \underline{x}_i$$

$$A \leftarrow A \setminus \{i\}$$

Se não Se $x_i > \bar{x}_i$ **então**

$$x_i \leftarrow \bar{x}_i$$

$$A \leftarrow A \setminus \{i\}$$

Termina Se

Termina Para

$$s = \sum_{i=1}^I \bar{P}_i x_i$$

Termina Enquanto

Se $s = Tx^{AVG}$ **então** Devolve x

Se não Elimina x /* $A = \phi$ */

Termina Se

O processo de reprodução utilizado no problema do nível superior cria N soluções *Desc*, cada uma gerada a partir da seleção de dois progenitores da população atual *Pop*. É aplicado um torneio binário no qual dois indivíduos de *Pop* são escolhidos aleatoriamente sendo o melhor selecionado para progenitor. No processo de cruzamento há 50% de possibilidade deste indivíduo ser o primeiro ou o segundo progenitor. O outro progenitor é selecionado aleatoriamente entre os elementos de *Pop*. É então aplicado um operador de cruzamento de um ponto de corte para obter um descendente a partir dos dois progenitores selecionados. Assim, se o primeiro progenitor for $x' = (x'_1, x'_2, \dots, x'_I)$ e o

segundo for $x'' = (x''_1, x''_2, \dots, x''_I)$, o descendente é $x^c = (x'_1, \dots, x'_{i_1}, x''_{i_1+1}, \dots, x''_I)$ onde i_1 é o ponto de corte obtido aleatoriamente no intervalo $[2, I - 1]$.

A mutação é aplicada ao vetor x^c com uma probabilidade P_m de modificar cada gene de x^c obtendo-se x^m . Para uma dada variável x_i^c , a mutação consiste em adicionar ou subtrair uma perturbação positiva gerada aleatoriamente no intervalo entre 0 e $0.2(\bar{x}_i - \underline{x}_i)$. Se x_i^m estiver fora dos limites impostos pelas restrições (4.11) e (4.12), então é colocado no limite mais próximo e o seu índice é excluído de um conjunto A das variáveis que podem ser modificadas pelo procedimento de reparação. Se x^m não for admissível porque não cumpre a restrição (4.13), então é reparado usando o procedimento descrito no Algoritmo 4.2, podendo ser eliminado se não for reparável por este procedimento. O processo de seleção, cruzamento, mutação e reparação é repetido até serem gerados N descendentes. Os descendentes formam a população $Desc$, que vai competir com Pop para determinar a população da próxima geração.

O indivíduo com o melhor valor de F obtido até ao momento sobrevive de uma geração para a seguinte. Este indivíduo é o primeiro a ser inserido na população da próxima geração. Os outros $N - 1$ indivíduos são obtidos através da seleção por torneio binário entre um indivíduo da população $Desc$ e outro de Pop , ambos escolhidos aleatoriamente, sendo o indivíduo com o melhor valor de F escolhido para integrar a população da geração seguinte. Todos os indivíduos incluídos na nova população são excluídos da população original (Pop ou $Desc$), de forma a que um indivíduo não possa ser selecionado mais do que uma vez.

4.5 Algoritmos baseados em EA e PSO

Nesta secção são propostos dois algoritmos para resolver o modelo BL 1 apresentado na seção 4.3 usando meta-heurísticas baseadas em populações para resolver os problemas de ambos os níveis. Os algoritmos baseados em populações permitem explorar em simultâneo diferentes conjuntos de preços de eletricidade (soluções do nível superior) e, para cada um deles, diferentes planos de calendarização dos equipamentos deslocáveis (soluções do nível inferior). As abordagens propostas são um algoritmo BL evolucionário (BLEA) e um algoritmo BL de otimização por enxames de partículas

(BLPSO). Os algoritmos usam a mesma estrutura para representar as populações. Em ambos os casos a população é composta por N indivíduos divididos em n_s sub-populações, cada uma contendo N_l indivíduos (ver Fig. 4.1). Todos os indivíduos (x, z) de uma sub-população têm os mesmos valores para o conjunto de variáveis do nível superior (i.e., o mesmo x). O vetor de valores reais do nível superior $x = (x_1, x_2, \dots, x_I)$ representa uma definição de preços de eletricidade e o vetor de valores inteiros do nível inferior $z = (z_1, z_2, \dots, z_J)$ representa os pontos no tempo em que cada carga inicia o seu ciclo de funcionamento.

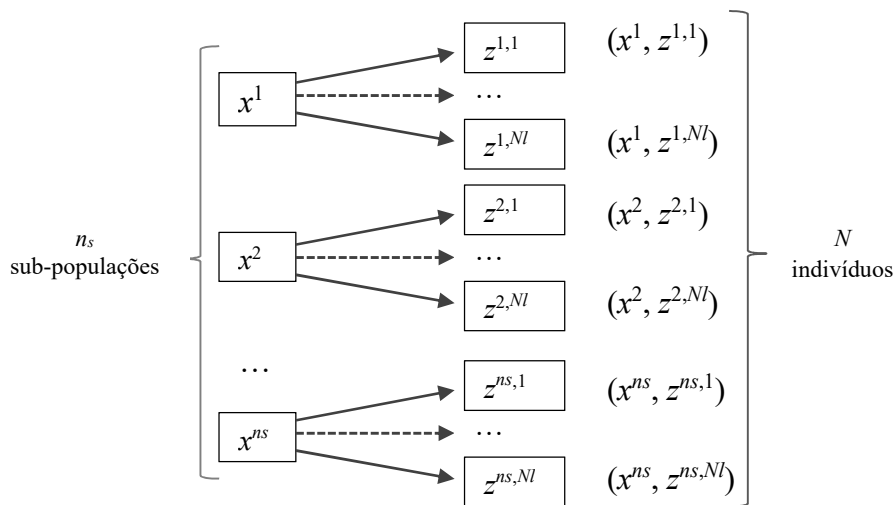


Fig. 4.1 Estrutura da população

4.5.1 Algoritmo BLEA

O algoritmo BLEA resolve o problema aplicando operadores genéticos aos indivíduos da população em ambos os sub-problemas dos níveis superior e inferior. O ciclo principal do algoritmo consiste em três passos. No primeiro passo é criada uma nova população $Pop1$ (com a estrutura representada na Fig. 4.1) através dos operadores de seleção cruzamento e mutação. Os indivíduos da população são inicialmente gerados aleatoriamente nos intervalos pré-definidos; primeiro são geradas as variáveis do nível superior e em seguida são obtidas as variáveis do nível inferior. Para obter um x admissível este vetor é sujeito a um processo de reparação descrito no Algoritmo 4.2 para satisfazer a restrição (4.4) que define um preço médio no período de planejamento.

Nos casos em que não é possível obter um vetor x admissível, é gerado um novo vetor x e o processo de reparação é reinicializado. Após ser obtido o vetor x , é gerado o

vetor z do nível inferior satisfazendo as restrições (4.9). Se o vetor não satisfizer as restrições (4.8), o que acontece quando a potência contratada é excedida num certo intervalo de tempo, é gerado um novo vetor z .

O pseudo-código do algoritmo BLEA é apresentado no Algoritmo 4.3. Os passos 2 e 3 constituem o ciclo principal do algoritmo que é efetuado durante K iterações. No passo 2 é criada uma nova população $Pop2$ usando os operadores de seleção, cruzamento e mutação para obter as componentes dos níveis superior e inferior de cada solução. Para cada sub-população de $Pop2$, primeiro é criado o vetor x do nível superior, que é o mesmo para todos os indivíduos dessa sub-população (Fig. 4.1), e em seguida é obtido o vetor z do nível inferior para cada indivíduo.

Para obter o vetor x do nível superior, são selecionados dois progenitores de diferentes sub-populações de $Pop1$: um é obtido por torneio binário selecionando a solução com melhor valor da função objetivo do nível superior e o outro é escolhido aleatoriamente. É aplicado um operador cruzamento (com um ponto de corte) para gerar um vetor x^c , com igual possibilidade de cada vetor progenitor dar origem à primeira ou à segunda parte de x^c . Em seguida é aplicado o operador mutação e o vetor resultante, x^m , é sujeito ao procedimento de reparação descrito no Algoritmo 4.2 caso não seja admissível (as restrições (4.2) – (4.4) são as mesmas restrições (4.11) – (4.13) do modelo BL 2). As operações algorítmicas no nível superior são as mesmas do algoritmo H-BLEA.

Para procurar obter o vetor z ótimo do nível inferior para o vetor x de cada sub-população de $Pop2$, são efetuadas as operações algorítmicas do nível inferior durante k_l iterações, evoluindo os indivíduos da sub-população através de seleção, cruzamento e mutação. Estes operadores são aplicados de uma forma similar à utilizada no nível superior. Após obter z^c por cruzamento de um ponto, é aplicado o operador de mutação a cada variável z_j^c com probabilidade P_m . A mutação consiste na adição ou subtração a z_j^c de uma perturbação positiva gerada aleatoriamente no intervalo entre 0 e $0.2(T2_j - d_j + 1 - T1_j)$ para gerar z_j^m . Se alguma variável do vetor z^m ficar fora dos limites do período de funcionamento admissível para a operação da carga definido pelas restrições (4.9), essa variável é deslocada para o limite mais próximo. Se z^m conduzir à

violação das restrições (4.8), devido a (4.6) e (4.7), o processo é repetido gerando um novo vetor z até ser obtido um indivíduo admissível.

Em cada iteração do nível inferior, este processo de seleção, cruzamento e mutação é repetido para os N_l indivíduos de cada sub-população, gerando a nova população *Desc*. A população *Desc* vai competir com a população atual *Pop2* para gerar a nova população (*Pop2*) para a próxima iteração do nível inferior.

No passo 3 as populações *Pop1* e *Pop2* são juntas e a melhor solução do nível inferior de cada sub-população é identificada. Em seguida, é selecionada a melhor solução de acordo com a função objetivo do nível superior entre as soluções identificadas, e a sub-população que contenha esta solução é incluída na nova *Pop1* para a próxima iteração. As restantes $n_s - 1$ sub-populações são escolhidas por torneio binário sem reposição entre *Pop1* e *Pop2*. Os torneios binários são efetuados entre as melhores soluções identificadas em cada sub-população de acordo com a função objetivo do nível superior.

O pseudo-código do algoritmo BLEA é apresentado no Algoritmo 4.3.

Algoritmo 4.3

Algoritmo BLEA - pseudo-código

(Passo 1) Inicializa $Pop1$ contendo n_s sub-populações $Pop1_s$, $s = 1, \dots, n_s$, com N_l indivíduos cada: gera aleatoriamente n_s vetores x admissíveis e, para cada um, gera aleatoriamente N_l vetores z admissíveis.

Avalia cada indivíduo da população (x, z) calculando $F(x, z)$ e $f(x, z)$.

Para $k = 1, \dots, K$

(Passo 2)

Para $s = 1, \dots, n_s$

Repete

Seleciona x' e x'' de $Pop1$, um aleatoriamente e o outro por torneio binário

$x^c \leftarrow$ Cruzamento de x' e x''

$x^m \leftarrow$ Mutação de x^c

Procedimento de reparação (x^m)

Até x^m ser admissível para o nível superior

Cria $Pop2_s$ com x^m e todos os z em $Pop1_s$:

$$Pop2_s = \left\{ (x^m, z^{s,1}), (x^m, z^{s,2}), \dots, (x^m, z^{s,N_l}) \right\}$$

Para $k_l = 1, \dots, K_l$

$Desc_s = \phi$

Para $n_l = 1, \dots, N_l$

Repete

Seleciona $z^{s,a}$ e $z^{s,b}$ de $Pop2_s$, um aleatoriamente e o outro por torneio binário

$z^{s,c} \leftarrow$ Cruzamento de $z^{s,a}$ e $z^{s,b}$

$z^{s,m} \leftarrow$ Mutação de $z^{s,c}$

Até $z^{s,m}$ ser admissível para o nível inferior

Insera $(x^m, z^{s,m})$ como (x^s, z^{s,n_l}) na sub-população $Desc_s$

Termina Para n_l

Seleciona os melhores N_l indivíduos de $Pop2_s$ e de $Desc_s$ de acordo com f para criar a nova $Pop2_s$ para a próxima iteração ($k_l + 1$).

Termina Para k_l

Termina Para s

/* seleciona a população para a próxima geração */

(Passo 3) Atualiza $Pop1$ para a geração $k+1$ copiando a sub-população de $Pop1 \cup Pop2$ contendo a melhor solução de acordo com F encontrada até ao momento, e efetuando $n_s - 1$ torneios binários sem reposição entre as sub-populações de $Pop1$ e $Pop2$.

Termina Para k

4.5.2 Algoritmo BL de otimização por enxame de partículas (BLPSO)

Enquanto o algoritmo evolucionário cria uma nova população em cada geração, na abordagem PSO os indivíduos (partículas) da população são movimentados de uma iteração para a seguinte mudando as suas posições. Em cada iteração do algoritmo PSO, as partículas são movimentadas de acordo com as equações (4.24) e (4.25).

$$v^q \leftarrow wv^q + c_1rand() (pbest^q - y^q) + c_2rand() (gbest - y^q) \quad (4.24)$$

$$y^q \leftarrow y^q + v^q, q = 1, \dots, n \quad (4.25)$$

Nestas equações y denota a posição da partícula, v é o vetor velocidade, w é o peso de inércia, v_1 e v_2 são os parâmetros cognitivo e social, $rand()$ é um valor uniforme aleatório no intervalo $[0,1]$, q é o índice da partícula e n é o número de partículas da população. $pbest^q$ representa a melhor posição (solução) obtida pela partícula q e $gbest$ representa a melhor posição alcançada por todas as partículas da população durante as iterações já efetuadas. O parâmetro w é linearmente decrescente, conforme apresentado em (Shi & Erberhart, 1999), para aumentar a capacidade de o algoritmo explorar novas regiões nas primeiras iterações e de explorar localmente a região alcançada nas últimas iterações.

O algoritmo começa por inicializar a população de N partículas divididas em n_s sub-populações, cada uma com $N_l = \frac{N}{n_s}$ partículas (ver Fig.4.1). Cada partícula tem duas componentes, a de nível superior x e a de nível inferior z . Para cada sub-população o vetor do nível superior é o mesmo e é inicialmente gerado aleatoriamente; é aplicado o processo de reparação descrito para os algoritmos H-BLEA e BLEA para garantir que x satisfaz a restrição (4.4). Se o procedimento de reparação não for bem sucedido em obter um x admissível, o processo é repetido até ser obtido um vetor do nível superior admissível. Em seguida são gerados aleatoriamente os vetores z do nível inferior para as partículas da sub-população respeitando (4.9). Se a solução correspondente ao vetor z de uma partícula não satisfizer as restrições (4.8), é efetuada uma nova tentativa e o processo é repetido até ser obtido um vetor z admissível.

A melhor posição alcançada por uma partícula no ciclo principal, $pbest1$, é inicializada com a posição atual e a melhor solução encontrada na população é guardada no vetor temporário pre_gbest para ser usado na primeira iteração do algoritmo. Esta solução é provisória porque a fase de otimização do nível inferior ainda não foi

executada. Na iteração seguinte é utilizado $gbest$ que é a melhor partícula de toda a população de acordo com F .

O ciclo principal do algoritmo é executado durante K iterações. Primeiro é atualizada a componente do nível superior (x) de cada partícula usando os operadores PSO (4.24) e (4.25) com $y = x$ e são verificadas as restrições (4.2), (4.3) e (4.4) para assegurar a admissibilidade da solução resultante. No caso de violação das primeiras duas restrições, a variável correspondente é alterada para o limite mais próximo; para a restrição (4.4) é efetuado o procedimento de reparação apresentado no Algoritmo 4.2.

Após obter o vetor do nível superior de cada sub-população, são executadas as operações relativas ao nível inferior durante k_l iterações procurando obter a solução ótima do nível inferior para cada sub-população. A pesquisa do nível inferior usa $gbest$ (a melhor solução de cada sub-população s de acordo com f) e é definido um $pbest2$ (local) para cada partícula. Em cada iteração do nível inferior são atualizados os vetores do nível inferior das N_l partículas usando (4.24) e (4.25) com $y = z$ e verificada a admissibilidade de cada partícula. Se a partícula atualizada não satisfizer as restrições (4.9) e/ou violar as restrições (4.8), é efetuado um procedimento de reparação durante um máximo de L tentativas. Este procedimento de reparação do nível inferior consiste em colocar a partícula no limite mais próximo definido pelas restrições (4.9). A partícula é então movimentada pelas equações (4.24) e (4.25), testando novamente as restrições (4.8). Se a partícula permanecer não admissível por mais do que L iterações, é reinicializada aleatoriamente e o processo de atualização é repetido até ser alcançada uma posição admissível. Após concluir o processo de atualização das partículas do nível inferior, são atualizadas a melhor posição ($pbest2$) de cada partícula e a melhor partícula ($gbest^s$) da sub-população de acordo com f .

No final do processo de otimização do nível inferior, é atualizada a melhor posição ($pbest1$) de cada partícula de acordo com F no ciclo principal. Este passo também atualiza a melhor posição encontrada até ao momento ($gbest$) em toda a população. Note-se que, após a primeira iteração, a melhor posição de toda a população substitui a solução temporária pre_gbest . No nível inferior, $pbest2$ e $gbest^s$ são apenas atualizados se a nova posição da partícula for melhor de acordo com a função objetivo do nível inferior. No nível superior, o processo de atualização de $pbest1$ e $gbest$ começa por analisar o valor associado à posição de cada partícula relativamente a F e, se a nova posição for melhor,

substitui a anterior. Caso contrário, se a nova posição e $pbest1$ da partícula tiverem o mesmo valor F então a comparação é efetuada relativamente à função objetivo do nível inferior; se a nova posição tiver um valor melhor relativamente a f , substitui a anterior. É importante referir que apenas são testadas para substituir a solução $gbest$ de toda a população as partículas que se mantiverem como $gbest^s$ de uma sub-população s durante pelo menos r_0 iterações do nível inferior. Nas experiências realizadas este parâmetro assume uma importância significativa para melhorar a capacidade do algoritmo em obter uma boa solução para o problema. Este processo impõe algum elitismo dado que uma solução $gbest^s$ de uma sub-população s deve provar ser a melhor durante um certo número iterações até ser eleita como a solução $gbest$ global de toda a população.

O pseudo-código do algoritmo BLPSO assim como o procedimento de atualização de $gbest$ e $pbest1$ são apresentados nos algoritmos 4.4 e 4.5, respetivamente.

Algoritmo 4.4

Algoritmo BLPSO - pseudo-código

Inicializa Pop com n_s sub-populações Pop_s , $s = 1, \dots, n_s$, com N_l partículas cada: gera aleatoriamente n_s vetores admissíveis x e, para cada um, gera aleatoriamente N_l vetores admissíveis z .

Avalia cada partícula da população (x, z) calculando $F(x, z)$ e $f(x, z)$.

Inicializa a melhor posição de cada partícula:

$$pbest1^{s,n_l} = (pbest1_x^s, pbest1_z^{s,n_l}) = (x^s, z^{s,n_l}), s = 1, \dots, n_s, n_l = 1, \dots, N_l.$$

Atribui a pre_gbest a melhor partícula de Pop de acordo com F , para ser usada como $gbest$ na primeira iteração.

Para $k = 1, \dots, K$

Para $s = 1, \dots, n_s$

Repete

Atualiza a componente x^s da partícula usando (4.24) e (4.25) com

$$y = x^s, pbest1_x^s \text{ e } gbest_x$$

Procedimento de reparação (x^s)

Até x^s ser admissível para o nível superior

Inicializa a melhor solução $gbest^s$ de Pop_s de acordo com f

Inicializa $pbest2^{s,n_l} = (pbest2_x^s, pbest2_z^{s,n_l}) = (x^s, z^{s,n_l}), n_l = 1, \dots, N_l$

Para $k_l = 1, \dots, K_l$

Para $n_l = 1, \dots, N_l$

$l = 0$

Repete

$l \leftarrow l + 1$

Atualiza o componente z^{s,n_l} de cada partícula usando (4.24) e

(4.25) com $y = z^{s,n_l}$, $pbest2_z^{s,n_l}$ e $gbest_z^s$

Se z^{s,n_l} não satisfaz as restrições (4.9) desloca-o para os limites mais próximos

Se z^{s,n_l} não satisfaz a restrição (4.8) e $l > L$ **Então** z^{s,n_l} é reinicializado e $l = 0$

Até z^{s,n_l} ser admissível para o nível inferior

Atualiza a melhor posição $pbest2^{s,n_l}$ alcançada pela partícula n_l de Pop_s de acordo com f : **Se**

$$f(x^s, z^{s,n_l}) < f(pbest2^{s,n_l}) \text{ Então } pbest2_z^{s,n_l} \leftarrow z^{s,n_l}$$

Termina Para n_l

Atualiza a melhor partícula $gbest^s$ de Pop_s de acordo com f

Termina Para k_l

Termina Para s

Atualiza *gbest* e *pbest1* (Algoritmo 4.5)
Termina Para *k*
 Devolve *gbest*

Algoritmo 4.5

Atualiza *gbest* e *pbest1* - pseudo-código

Para $s = 1, \dots, n_s$

/* atualiza *pbest1* de cada partícula */

Para $n_l = 1, \dots, N_l$

Se $F(x^s, z^{s,n_l}) > F(pbest1^{s,n_l})$ ou $[F(x^s, z^{s,n_l}) = F(pbest1^{s,n_l})$ e

$f(x^s, z^{s,n_l}) < f(pbest1^{s,n_l})]$ **Então**

$(pbest1^{s,n_l}) \leftarrow (x^s, z^{s,n_l})$

Termina Para n_l

/* atualiza *gbest* */

Seja r o número de vezes que $gbest^s$ se manteve em Pop_s

Se $r > r_0$

Atualiza o *gbest* da população *Pop* usando $gbest^s$ de acordo com F

Termina Para s

4.6 Comparação dos algoritmos – Resultados experimentais e discussão

4.6.1 Dados do problema e definição de parâmetros

Para efetuar uma análise comparativa entre os algoritmos BLPSO, BLEA e H-BLEA, considerámos um problema com três cenários para os períodos de funcionamento dos equipamentos. A maior parte dos dados foi obtida através de auditorias energéticas, tendo alguns valores sido estimados.

É considerado um período de planeamento de 24 horas dividido em intervalos de 15 minutos, conduzindo a $T = \{1, \dots, 96\}$. A carga é composta por uma carga base (apresentada na Fig. 4.2), associada ao consumo de equipamentos cuja utilização não se considera controlável, e cinco equipamentos cujo período de utilização é deslocável no tempo ($J = 5$): máquina de lavar roupa, máquina de lavar loiça, aquecedor elétrico de água – EWH, secador de roupa e veículo elétrico. Os ciclos de funcionamento destes equipamentos estão representados na Fig. 4.3, onde é apresentada a potência requerida pelo equipamento j no estado r do seu ciclo de funcionamento ($r = 1, \dots, d_j$).

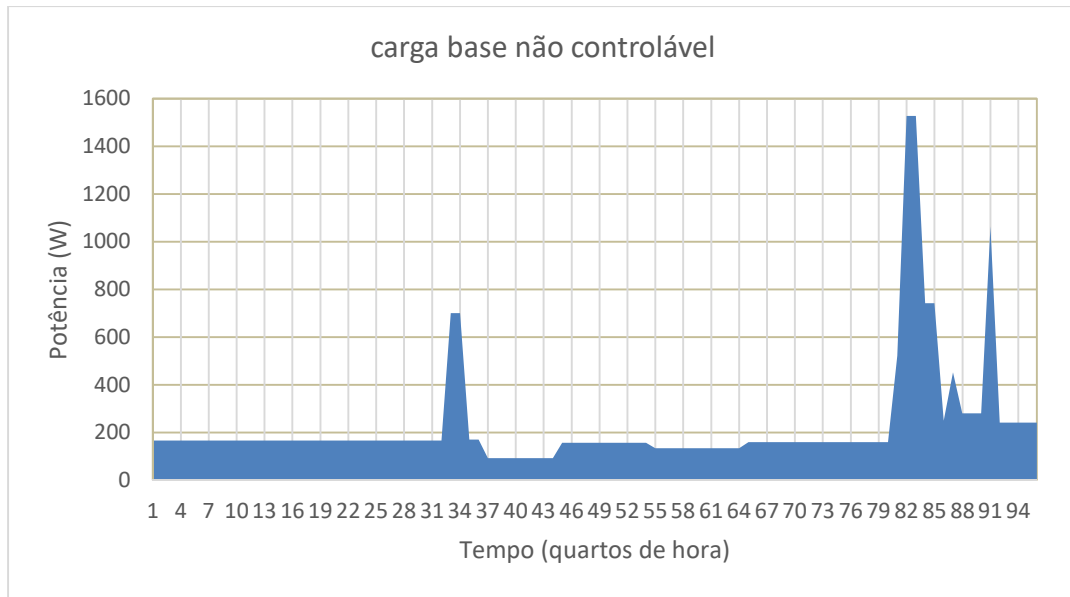


Fig. 4.2 Potência da carga base (não controlável)

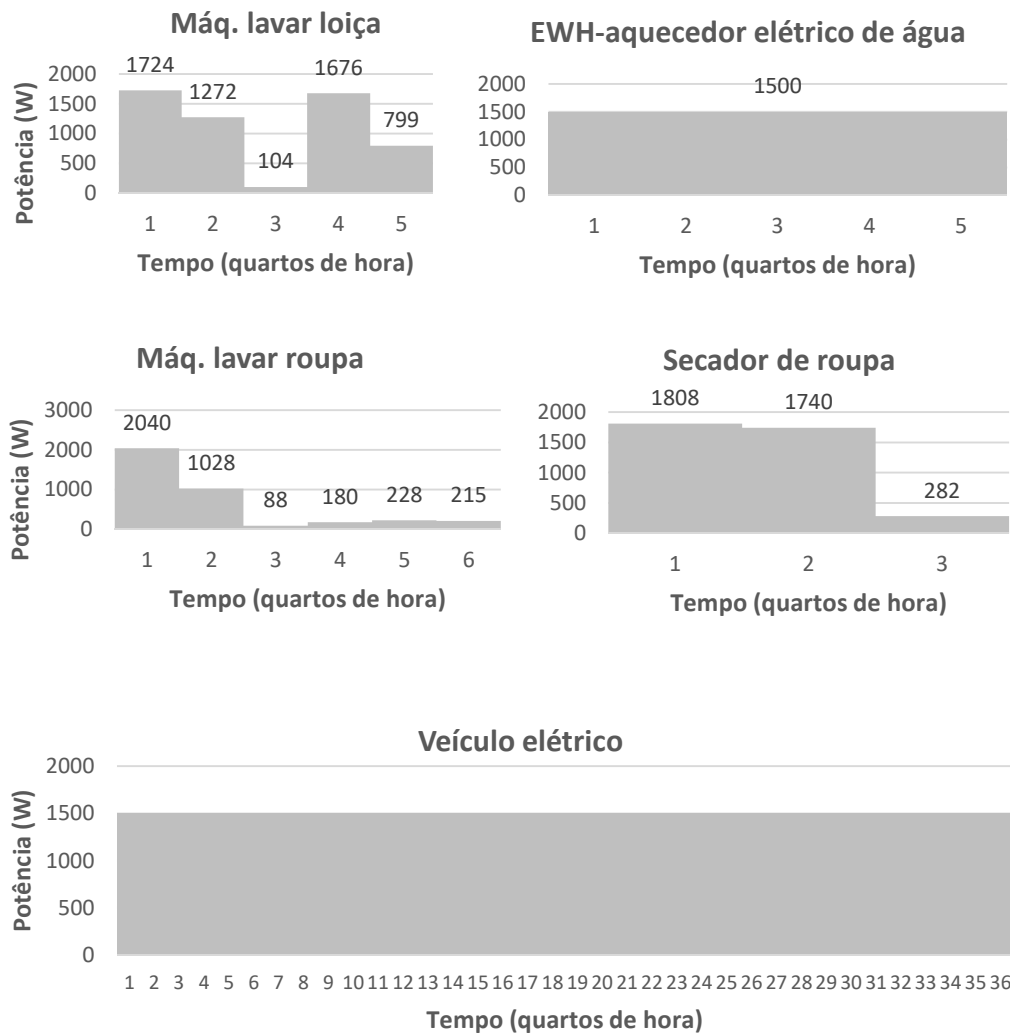


Fig. 4.3 Ciclos de funcionamento dos equipamentos controláveis

Os preços de eletricidade mínimos e máximos que podem ser aplicados ao consumidor em cada um de sete sub-períodos de tempo ($I = 7$) são apresentados na Tabela 4.1. O preço médio de energia ao longo de todo o período de planeamento é definido como $x^{AVG} = 0.116 \text{ €/kWh}$. Os preços de eletricidade suportados pelo retalhista estão apresentados na Fig. 4.4. Todos os preços apresentados estão em €/kWh, sendo depois convertidos para períodos de quarto de hora para serem utilizados no modelo.

Tabela 4.1 Limites máximos e mínimos dos preços de eletricidade

Sub-período	Intervalo inicial (t)	Intervalo final (t)	Preço mínimo (€/kWh)	Preço máximo (€/kWh)
1	1	28	0.04	0.10
2	29	38	0.08	0.24
3	39	44	0.03	0.12
4	45	60	0.10	0.28
5	61	76	0.03	0.12
6	77	84	0.08	0.24
7	85	96	0.04	0.10

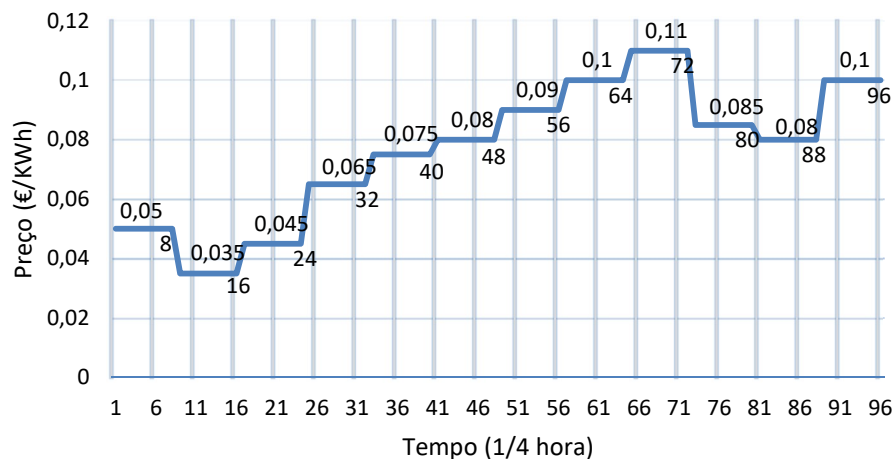


Fig. 4.4 Preços da energia suportados pelo retalhista

O ciclo de funcionamento dos cinco equipamentos cujo período de utilização é deslocável no tempo pode ser calendarizado dentro do período de planeamento, i.e. o seu período de funcionamento habitual pode ser deslocado de acordo com os preços de eletricidade e as preferências do consumidor. Consideramos três perfis de comportamento do consumidor relativamente à extensão e localização temporal do período durante o qual é permitido o funcionamento do equipamento. Além de um perfil base refletindo um padrão de consumo mais habitual, são também considerados um perfil mais restrito, refletindo uma menor disponibilidade para aceitar a deslocação de cargas, e um perfil estendido,

correspondendo a uma maior disponibilidade para aceitar a calendarização das cargas em períodos de tempo mais alargados. Estes perfis estão associados a espaços de pesquisa mais ou menos restritos tendo em conta as restrições (4.9), as quais influenciam a quantidade de potenciais soluções do nível inferior. Os períodos de funcionamento permitidos para utilização dos equipamentos relativos a cada perfil são mostrados nas Tabelas 4.2-4.4.

Além do cenário inicial, consideramos dois cenários (variantes) de cada um dos perfis, correspondendo a deslocações nos períodos de funcionamento permitidos para o funcionamento das cargas. Pretende-se deste modo confirmar a estabilidade dos algoritmos em cada um dos cenários apresentados. Assim, obtemos as variantes $v1$ e $v2$ de cada um dos três perfis (base, restrito e estendido), cada uma obtida deslocando o período de funcionamento permitido de todas as cargas para uma zona diferente do período de planeamento.

Tabela 4.2. Períodos permitidos para a operação de cada equipamento no perfil base.

Equipamento	Intervalo inicial	Intervalo final
Máq. Lavar loiça	1	36
Máquina lavar roupa	32	60
Aquecedor elétrico de água (EWH)	24	40
Veículo elétrico	1	48
Secador de roupa	76	96

Tabela 4.3. Períodos permitidos para a operação de cada equipamento no perfil restrito.

Equipamento	Intervalo inicial	Intervalo final
Máq. Lavar loiça	1	34
Máquina lavar roupa	32	50
Aquecedor elétrico de água (EWH)	24	36
Veículo elétrico	1	45
Secador de roupa	70	82

Tabela 4.4. Períodos permitidos para a operação de cada equipamento no perfil estendido.

Equipamento	Intervalo inicial	Intervalo final
Máq. Lavar loiça	1	44
Máquina lavar roupa	28	65
Aquecedor elétrico de água (EWH)	24	45
Veículo elétrico	1	48
Secador de roupa	70	96

Tabela 4.5. Períodos permitidos para a operação de cada equipamento no perfil base e nos cenários $v1$ e $v2$.

Equipamento	Cenário $v1$		Cenário $v2$	
	Int. inicial	Int. final	Int. inicial	Int. Final
Máq. Lavar loiça	28	63	60	95
Máquina lavar roupa	4	32	36	64
Aquecedor elétrico de água (EWH)	40	56	60	76
Veículo elétrico	30	77	49	96
Secador de roupa	56	76	20	40

Tabela 4.6. Períodos permitidos para a operação de cada equipamento no perfil restrito e cenários $v1$ e $v2$.

Equipamento	Cenário $v1$		Cenário $v2$	
	Int. inicial	Int. final	Int. inicial	Int. Final
Máq. Lavar loiça	28	61	60	93
Máquina lavar roupa	14	32	36	54
Aquecedor elétrico de água (EWH)	40	52	60	72
Veículo elétrico	33	77	52	96
Secador de roupa	56	68	26	38

Tabela 4.7. Períodos permitidos para a operação de cada equipamento no perfil estendido e cenários $v1$ e $v2$.

Equipamento	Cenário $v1$		Cenário $v2$	
	Int. inicial	Int. final	Int. inicial	Int. Final
Máq. Lavar loiça	28	71	53	96
Máquina lavar roupa	3	40	35	72
Aquecedor elétrico de água (EWH)	40	61	60	81
Veículo elétrico	30	77	49	96
Secador de roupa	50	76	14	40

Adicionalmente realizamos uma experiência considerando uma maior discretização do período de planeamento, para analisar a escalabilidade de cada um dos algoritmos. É importante referir que uma maior discretização do período de planeamento aumenta significativamente o número de variáveis do Modelo BL2 usado pelo algoritmo híbrido H-BLEA. Nesta experiência, discretizamos o período de planeamento de 24 horas em intervalos de 5 minutos.

No algoritmo BLEA, consideramos uma população de $N = 240$ indivíduos divididos em $n_s = 12$ sub-populações, cada uma contendo $N_l = 20$ indivíduos. O algoritmo BLEA é executado durante $K = 100$ gerações, sendo efetuada, em cada geração, a otimização do nível inferior durante $K_l = 40$ gerações. No algoritmo BLPSO consideramos os parâmetros $N = 240$, $N_l = 30$, $K = 100$ e $K_l = 60$ e o parâmetro $r_0 = 4$. Os valores selecionados para a dimensão da população e número de iterações do nível

inferior foram obtidos após realização de algumas experiências, analisando o comportamento dos algoritmos e tendo em conta que, para os mesmos parâmetros, o algoritmo BLEA tem um custo computacional superior ao algoritmo BLPSO. Definimos o parâmetro w linearmente decrescente de 0.9 (no início do algoritmo) até 0.4 (no final do algoritmo). No algoritmo híbrido H-BLEA, consideramos uma população de $N = 30$ indivíduos (neste caso $n_s = N$ porque a cada solução do nível superior está associada uma única solução do nível inferior). O algoritmo H-BLEA é executado durante $K = 100$ gerações.

Todos os algoritmos foram implementados em Matlab e testados num computador Intel Core I-7 3.2 GHz 32 GB RAM. A pesquisa do nível superior do algoritmo H-BLEA é similar à realizada pelo algoritmo BLEA e o problema do nível inferior é resolvido de forma exata usando o *solver* CPLEX para cada instanciação de x . Assim, as soluções do nível inferior obtidas pelo algoritmo H-BLEA permitem avaliar a capacidade dos algoritmos BLEA e BLPSO para determinar soluções de boa qualidade para o problema do nível inferior. É importante referir que uma solução admissível para o problema BL deve ser uma solução ótima para o problema do nível inferior.

4.6.2 Resultados

O problema foi resolvido para os três perfis mostrados nas Tabelas 4.2 – 4.4, usando os algoritmos BLPSO, BLEA e H-BLEA. A experiência consistiu em executar os algoritmos o número necessário de vezes até obter 30 soluções válidas para cada instância. Após cada execução dos algoritmos BLPSO e BLEA é verificada a validade da solução. Uma solução é válida se for ótima para o problema do nível inferior para os preços estabelecidos no nível superior. Assim, no algoritmo H-BLEA foram necessárias 30 execuções, dado que o problema do nível inferior é resolvido de forma exata. No algoritmo BLPSO, foram necessárias 30 execuções para o perfil estendido; nos outros perfis foram necessárias mais algumas execuções (até mais 30%) para obter 30 soluções válidas. O algoritmo BLEA apresentou dificuldades em obter soluções ótimas do nível inferior para o perfil estendido, onde em média apenas 1 solução é válida em cada 6 execuções. No perfil restrito o algoritmo BLEA apenas necessitou de 30 execuções e no perfil base foram necessárias 33. No perfil estendido o espaço de pesquisa é maior e o algoritmo BLEA tem dificuldade em escapar a soluções sub-ótimas para o problema do nível inferior.

A informação acerca das 30 soluções válidas obtidas para os perfis base, restrito e estendido é apresentada nas Tabelas 4.8 – 4.10. A Tabela 4.8 apresenta o máximo, média, mediana e intervalo interquartil para o lucro do retalhista (F) de entre as 30 corridas e ainda o custo do consumidor (f) na melhor solução para o problema BL, i.e. com o máximo de F . O custo do consumidor e o lucro do retalhista são expressos em €. Os resultados referem-se a um período de 24 horas e correspondem a um grupo de 1000 consumidores com consumos de energia e perfis de resposta similares. A Tabela 4.9 apresenta os preços de energia (valores das variáveis do nível superior) e a Tabela 4.10 o intervalo de início de funcionamento (variáveis do nível inferior) na melhor solução para cada perfil.

Em comparação com o perfil base, a principal dificuldade no perfil restrito consiste em obter soluções admissíveis para o nível inferior, enquanto no perfil estendido a principal dificuldade reside na existência de soluções ótimas locais para o nível inferior. Note-se ainda que a restrição da potência contratada (4.8) e as características dos ciclos de operação das cargas (Fig. 4.3) introduzem dificuldades adicionais na determinação das soluções ótimas.

O algoritmo BLPSO é o que apresenta as melhores soluções em todos os perfis relativamente aos valores máximo e mediana de F (cf. Tabela 4.8 e Fig. 4.5). A Fig. 5 apresenta os diagramas de extremos (*boxplots*) para o valor da função objetivo do retalhista (F), permitindo uma comparação da mediana e dos intervalos interquartis de F nas soluções válidas obtidas pelo três algoritmos nos três perfis. O algoritmo BLPSO é o que apresenta maior variabilidade ao longo das várias execuções. Esta variabilidade deve-se principalmente à presença de *outliers* (cf. Fig. 4.5), o que justifica que a média dos valores de F obtidos pelo algoritmo BLPSO não seja sempre a melhor em todos os perfis.

A avaliação da significância das diferenças do lucro do retalhista ao longo dos três algoritmos é efetuada através do teste estatístico Kruskal-Wallis. Ao nível de significância $\alpha = 0.05$, estas diferenças são estatisticamente significativas para todos os perfis.

Tabela 4.8. Lucro do Retalhista e custo do Consumidor (para a melhor solução de F) para os perfis base, restrito e estendido.

Perfil	Algoritmo	Máximo F	Média F	Mediana F	IQR F	f
Base	BLPSO	1827.639	1800.205	1826.715	10.15	3357.58
	BLEA	1826.810	1799.389	1800.929	23.41	3356.75
	H-BLEA	1818.490	1804.578	1808.060	9.93	3351.68
Restrito	BLPSO	1825.732	1800.213	1814.370	1.16	3378.78
	BLEA	1818.160	1802.236	1802.290	20.69	3329.20
	H-BLEA	1818.500	1813.115	1814.130	5.83	3329.50
Estendido	BLPSO	1493.350	1468.644	1482.100	24.96	3047.81
	BLEA	1471.750	1458.445	1461.450	21.37	3036.34
	H-BLEA	1481.960	1465.075	1466.230	19.30	3040.17

Tabela 4.9. Preços da energia (€/kWh) para cada sub-período na melhor solução obtida para os perfis base, restrito e estendido.

Perfil	Algoritmo	Sub-períodos						
		P_1	P_2	P_3	P_4	P_5	P_6	P_7
		[1, 28]	[29, 38]	[39, 44]	[45, 60]	[61, 76]	[77, 84]	[85, 96]
Base	BLPSO	0.1	0.24	0.12	0.101	0.03	0.24	0.1
	BLEA	0.1	0.24	0.12	0.100128	0.030897	0.24	0.1
	H-BLEA	0.099843	0.239843	0.119835	0.101761	0.031761	0.235828	0.1
Restrito	BLPSO	0.10	0.24	0.12	0.10	0.066648	0.24	0.052470
	BLEA	0.10	0.24	0.12	0.120143	0.048983	0.24	0.049166
	H-BLEA	0.09999	0.23999	0.11999	0.12002	0.051141	0.23999	0.046493
Estendido	BLPSO	0.1	0.24	0.12	0.1	0.060572	0.24	0.060571
	BLEA	0.1	0.24	0.12	0.100492	0.055462	0.236928	0.068776
	H-BLEA	0.099989	0.239989	0.119950	0.100067	0.060634	0.239926	0.060507

Tabela 4.10. Intervalos iniciais e finais de funcionamento dos equipamentos na melhor solução obtida para os perfis base, restrito e estendido.

Perfil	Algoritmo	Máq. lav. Loixa	Máq. lavar roupa	EWH	Veiculo elétrico	Secador de roupa
Base	BLPSO	1-5	45-50	36-40	5-40	85-87
	BLEA	1-5	45-50	36-40	5-40	85-87
	H-BLEA	1-5	48-53	36-40	5-40	85-87
Restrito	BLPSO	5-9	39-44	28-32	9-44	74-76
	BLEA	1-5	39-44	28-32	5-40	73-75
	H-BLEA	1-5	39-44	28-32	5-40	74-76
Estendido	BLPSO	39-43	60-65	39-43	1-36	86-88
	BLEA	40-44	60-65	41-45	1-36	73-75
	H-BLEA	39-43	60-65	41-45	1-36	85-87

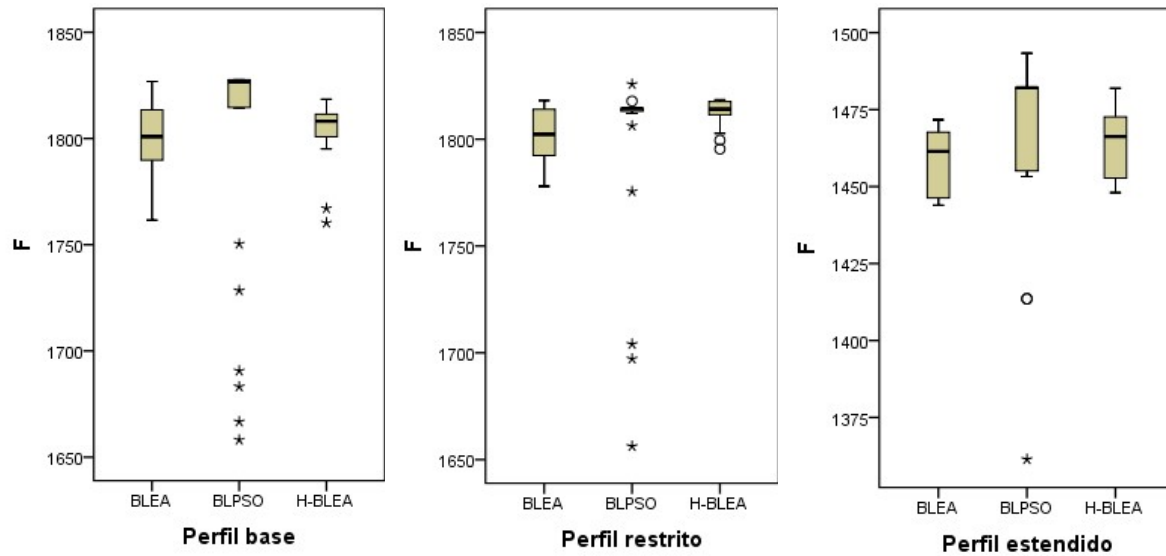


Fig. 4.5. Diagramas de extremos e quartis do lucro do retalhista (F) para os perfis base, restrito e estendido (os símbolos “o” e “*” representam *outliers*, i.e. valores além de 1.5 e 3 vezes o intervalo interquartil, respetivamente).

No perfil base, as soluções ótimas obtidas pelos três algoritmos apenas diferem no período de funcionamento da máquina de lavar roupa, apesar de este ocorrer sempre no mesmo sub-período de preços P_4 . No perfil restrito, as soluções do nível inferior diferem nos períodos de operação de três equipamentos; os períodos de funcionamento da máquina de lavar loiça e do secador de roupa estão dentro dos mesmos sub-períodos de preços (P_1 e P_7 , respetivamente) e o período de operação do veículo elétrico distribui-se ao longo de três sub-períodos de preços (P_1 a P_3) com graus de interseção distintos. Como se poderia esperar, no perfil estendido os algoritmos obtêm soluções do nível inferior diferentes porque o espaço de pesquisa é mais amplo devido aos períodos de tempo disponíveis para funcionamento dos equipamentos serem mais alargados. Em particular, o secador de roupa opera no sub-período P_5 na solução obtida pelo algoritmo BLEA e no sub-período P_7 nas soluções obtidas pelos algoritmos BLPSO e H-BLEA. Isto deve-se ao facto de o preço definido pelo algoritmo BLEA no sub-período P_7 ser superior ao preço no sub-período P_5 e se verificar a relação inversa nos algoritmos H-BLEA e BLPSO.

Na tabela 4.11 são apresentados os tempos de execução (em segundos) de cada algoritmo no perfil base, relativamente às 30 corridas efetuadas. Observamos que os tempos de execução dos algoritmos BLPSO e BLEA são similares, apesar de o algoritmo

BLEA despende um pouco mais de tempo para resolver o problema. O algoritmo H-BLEA requer mais tempo para resolver o problema porque chama um *solver* externo para determinar a solução ótima do problema do nível inferior. O esforço computacional de cada algoritmo é aferido em termos de tempo de execução em vez de número de avaliações da função objetivo porque esta última medida é difícil de obter para o algoritmo H-BLEA. As relações entre os tempos computacionais dos três algoritmos são semelhantes para os restantes perfis.

Tabela 4.11. Tempo de execução (em segundos) dos algoritmos para o perfil base.

Algoritmo	Máximo	Mediana	Média	Desvio Padrão
BLPSO	552.9	534.17	532.96	8.45
BLEA	639.85	629.16	629.22	4.69
H-BLEA	1127.57	1083.48	1087.44	9.52

Apresentamos nas Fig. 4.6-4.8 os diagramas de carga associados à melhor solução para cada perfil, para o cenário original. Todas as soluções foram obtidas pelo algoritmo BLPSO. Na melhor solução para o perfil restrito o pico de consumo é maior, o que decorre da redução do intervalo de tempo disponível para funcionamento da máquina de lavar roupa, que é o equipamento com o maior pico de consumo, apesar de tal ser parcialmente compensado pela deslocação do funcionamento do aquecedor elétrico de água para um período de menor consumo. No perfil estendido, a maior flexibilidade na operação das cargas conduz a uma maior diversidade de deslocações possíveis para obter as soluções ótimas do nível inferior em face dos preços fixados pelo retalhista.

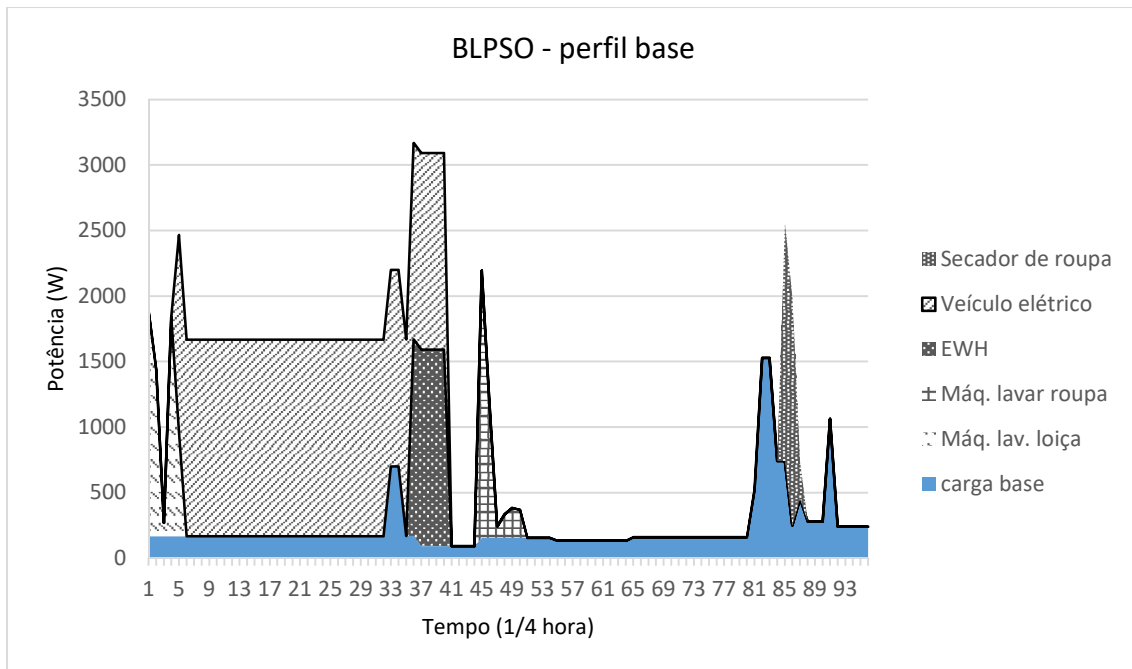


Fig. 4.6 Diagrama de carga para a melhor solução para o perfil base, obtida pelo algoritmo BLPSO

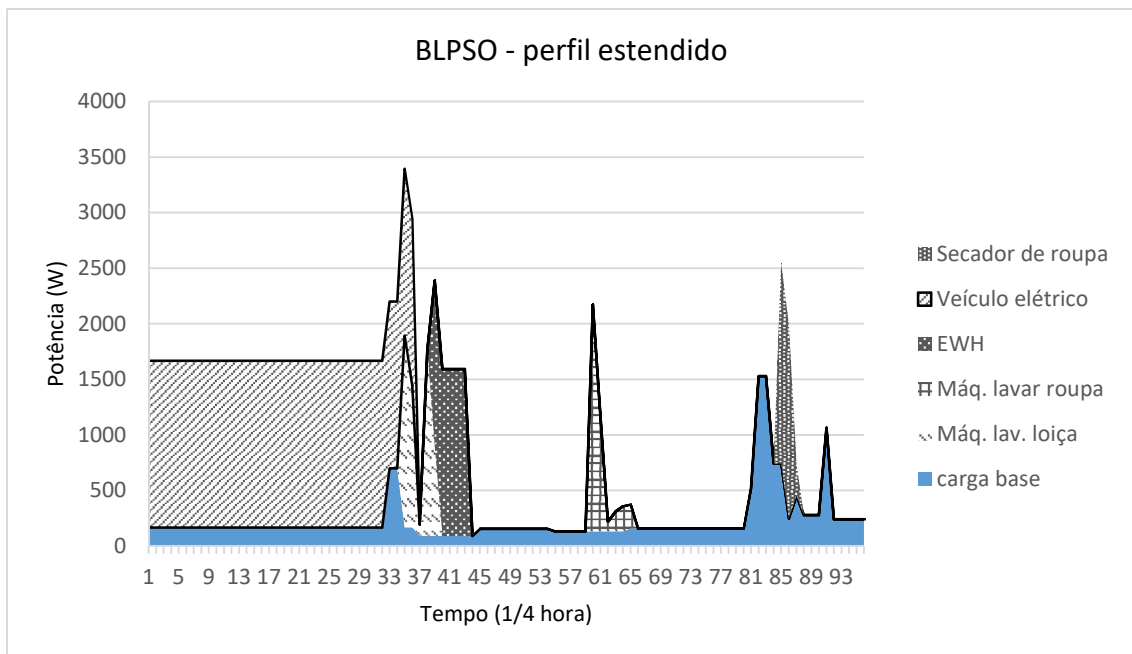


Fig. 4.7 Diagrama de carga para a melhor solução para o perfil estendido, obtida pelo algoritmo BLPSO

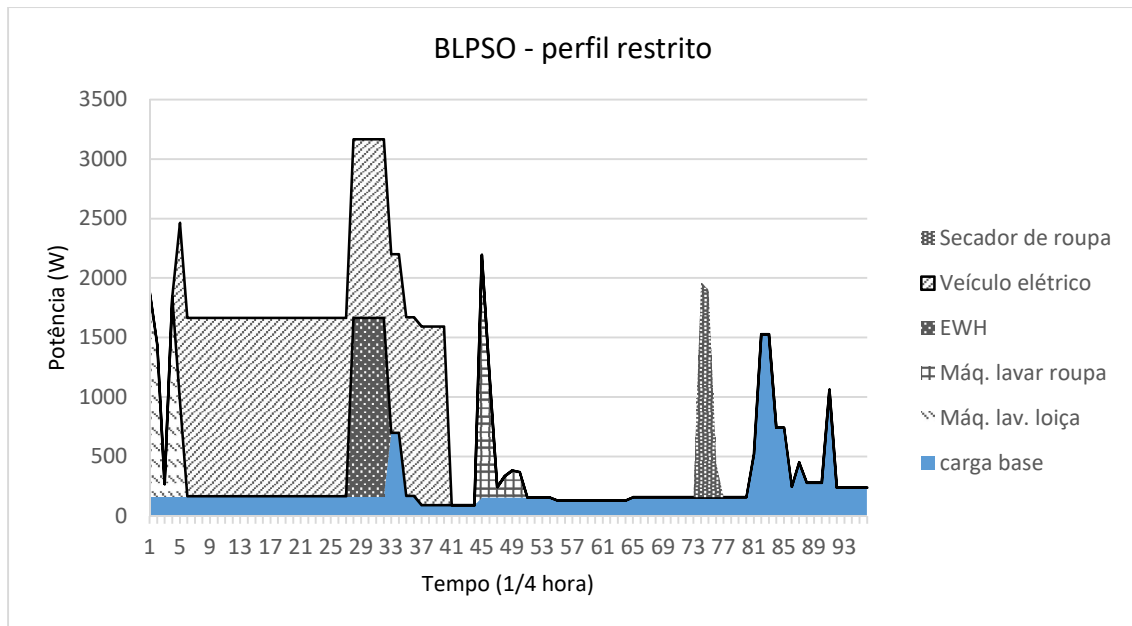


Fig. 4.8 Diagrama de carga da melhor solução obtida pelo algoritmo BLPSO para o perfil restrito

Foram também testadas dois outros cenários (variantes), utilizando os dados das Tabelas 4.5 – 4.7, para avaliar a robustez dos algoritmos. Nestes cenários foram deslocados os intervalos de tempo permitidos para a operação das cargas em cada perfil (base, restrito e estendido) mantendo a amplitude dos intervalos, i.e. foram testados seis problemas adicionais. Os dois cenários têm uma maior simultaneidade de intervalos de funcionamento, o que constitui um maior desafio para os algoritmos. No cenário *v1* os intervalos de funcionamento estão mais concentrados no meio do período de planeamento, onde os limites inferior/superior dos preços são mais elevados. No cenário *v2* os intervalos de funcionamento permitidos para as cargas estão mais concentrados no final do período de planeamento, onde a carga não controlável é superior e os preços de energia suportados pelo retalhista são mais elevados. O cenário *v2* é menos favorável para o lucro do retalhista.

Tabela 4.12. Lucro do Retalhista e custo do Consumidor (para a melhor solução de F) para o cenário vI .

Perfil	Algoritmo	Máximo F	Média F	Mediana F	IQR F	f
Base	BLPSO	1772.873	1684.281	1715.620	13.744	3706.766
	BLEA	1761.623	1720.390	1736.314	13.453	3695.515
	H-BLEA	1735.510	1607.230	1714.415	24.038	3702.190
Restrito	BLPSO	1702.810	1647.313	1650.542	89.539	3703.248
	BLEA	1696.240	1667.751	1678.841	25.988	3696.677
	H-BLEA	1687.660	1615.550	1677.865	15.625	3697.670
Estendido	BLPSO	1769.364	1718.629	1754.967	63.355	3703.257
	BLEA	1760.885	1729.772	1741.903	21.908	3694.778
	H-BLEA	1756.430	1633.370	1723.710	27.415	3701.620

Tabela 4.13. Preços da energia (€/kWh) para cada sub-período na melhor solução obtida para o cenário vI .

Perfil	Algoritmo	Sub-períodos						
		P_1	P_2	P_3	P_4	P_5	P_6	P_7
		[1, 28]	[29, 38]	[39, 44]	[45, 60]	[61, 76]	[77, 84]	[85, 96]
Base	BLPSO	0.04	0.129440	0.12	0.28	0.12	0.08	0.080134
	BLEA	0.040860	0.119607	0.12	0.28	0.12	0.139416	0.046711
	H-BLEA	0.040222	0.119900	0.12	0.28	0.12	0.150840	0.040340
Restrito	BLPSO	0.04	0.12	0.12	0.28	0.12	0.152	0.04
	BLEA	0.041392	0.119907	0.12	0.28	0.12	0.139225	0.045346
	H-BLEA	0.040926	0.119792	0.119864	0.279864	0.119864	0.144679	0.043326
Estendido	BLPSO	0.04	0.120003	0.12	0.28	0.12	0.151991	0.040004
	BLEA	0.041326	0.119601	0.12	0.28	0.12	0.138045	0.046543
	H-BLEA	0.040484	0.119804	0.12	0.28	0.12	0.150382	0.040113

Tabela 4.14. Intervalo inicial de funcionamento dos equipamentos na melhor solução obtida para o cenário vI .

Perfil	Algoritmo	Máq. lav. Loiça	Máq. lavar roupa	EWH	Veículo elétrico	Secador de roupa
Base	BLPSO	28-32	11-16	40-44	30-65	74-76
	BLEA	28-32	10-15	40-44	30-65	73-75
	H-BLEA	28-32	4-9	40-44	30-65	64-66
Restrito	BLPSO	28-32	14-19	40-44	33-68	61-63
	BLEA	28-32	14-19	40-44	33-68	62-64
	H-BLEA	28-32	14-19	40-44	33-68	66-68
Estendido	BLPSO	28-32	10-15	40-44	30-65	74-76
	BLEA	28-32	11-16	40-44	30-65	74-76
	H-BLEA	28-32	9-14	40-44	30-65	72-74

Tabela 4.15. Lucro do Retalhista e custo do Consumidor (para a melhor solução de F) para o cenário $v2$.

Perfil	Algoritmo	Máximo F	Média F	Mediana F	IQR F	F
Base	BLPSO	1086.672	1011.424	1071.978	78.123	3215.326
	BLEA	1085.664	1075.354	1079.630	13.503	3214.347
	H-BLEA	1086.852	1050.229	1070.628	22.661	3215.534
Restrito	BLPSO	1053.966	947.370	1032.238	41.309	3201.299
	BLEA	1058.913	1051.997	1054.952	3.810	3215.364
	H-BLEA	1054.073	1023.373	1015.594	35.460	3206.275
Estendido	BLPSO	1094.108	1008.666	1066.124	75.575	3214.944
	BLEA	1088.045	1081.187	1082.052	5.291	3214.220
	H-BLEA	1089.170	1070.519	1072.001	26.041	3214.819

Tabela 4.16. Preços da energia (€/kWh) para cada sub-período na melhor solução obtida para o cenário $v2$.

Perfil	Algoritmo	Sub-períodos						
		P_1 [1, 28]	P_2 [29, 38]	P_3 [39, 44]	P_4 [45, 60]	P_5 [61, 76]	P_6 [77, 84]	P_7 [85, 96]
Base	BLPSO	0.080880	0.119414	0.12	0.119826	0.12	0.24	0.1
	BLEA	0.083955	0.116542	0.116772	0.117451	0.12	0.24	0.1
	H-BLEA	0.080792	0.119712	0.119996	0.119800	0.12	0.24	0.1
Restrito	BLPSO	0.1	0.102312	0.090954	0.107947	0.12	0.24	0.1
	BLEA	0.081223	0.119270	0.12	0.119316	0.12	0.24	0.1
	H-BLEA	0.075016	0.159792	0.108732	0.109076	0.12	0.24	0.1
Estendido	BLPSO	0.080664	0.119238	0.12	0.120314	0.12	0.24	0.1
	BLEA	0.078783	0.121704	0.12	0.122065	0.12	0.24	0.1
	H-BLEA	0.081	0.118816	0.12	0.119988	0.12	0.24	0.1

Tabela 4.17. Intervalo inicial de funcionamento dos equipamentos na melhor solução obtida para o cenário $v2$.

Perfil	Equipamentos	Máq. lav. Louça	Máq. lavar roupa	EWH	Veículo elétrico	Secador de roupa
Base	BLPSO	60-64	36-41	72-76	61-96	20-22
	BLEA	60-64	36-41	72-76	61-96	20-22
	H-BLEA	60-64	36-41	72-76	61-96	20-22
Restrito	BLPSO	88-92	39-44	60-64	52-87	26-28
	BLEA	72-76	36-41	60-64	61-96	26-28
	H-BLEA	89-93	39-44	60-64	53-88	26-28
Estendido	BLPSO	61-65	35-40	72-76	61-96	14-16
	BLEA	63-67	39-44	72-76	61-96	14-16
	H-BLEA	53-57	35-40	60-64	61-96	14-16

As Tabelas 4.12 - 4.17 apresentam os resultados obtidos para os cenários $v1$ e $v2$ de cada um dos modelos base, restrito e estendido. No cenário $v1$, o algoritmo BLPSO apresenta os melhores resultados para o valor máximo de F em todos os perfis e a melhor mediana no perfil estendido, enquanto o algoritmo BLEA apresenta a melhor mediana nos perfis base e restrito e a melhor média em todos os perfis. No cenário $v2$ o algoritmo BLEA obtém os melhores valores da média e mediana em todos os perfis. Nos dois cenários o algoritmo BLPSO mostra, em geral, uma maior variabilidade, de acordo com a medida IQR de F , i.e. obtém soluções muito boas em algumas corridas mas também soluções piores do que as obtidas pelos outros algoritmos noutras corridas. No cenário $v1$, esta variabilidade justifica que este algoritmo apresente o melhor desempenho no máximo de F mas não na mediana de F . Esta situação parece decorrer de uma maior taxa de reparação requerida pelas soluções não admissíveis do nível inferior devido ao movimento de partículas na otimização por enxame de partículas ser contínuo e, por isso exigir mais arredondamentos das variáveis, em comparação com o algoritmo BLEA. O algoritmo H-BLEA apresenta os piores valores do máximo de F em todos os perfis do cenário $v1$.

A análise das nove instâncias do problema (3 perfis x 3 cenários) conduz às seguintes conclusões principais. O algoritmo BLPSO encontrou a melhor solução em 7 das 9 instâncias, enquanto os algoritmos BLEA e H-BLEA obtiveram, cada um, apenas uma melhor solução, nas restantes 2 instâncias. O algoritmo BLEA teve algumas dificuldades em obter soluções ótimas do problema do nível inferior no perfil estendido, em qualquer dos cenários. Apesar de o algoritmo H-BLEA apresentar o pior valor máximo do lucro do retalhista em 4 das 9 instâncias, os seus valores nunca são mais do que 2% piores relativamente à melhor solução obtida, em todas as instâncias.

Considerámos ainda a discretização do período de planeamento em intervalos de 5 minutos. Começámos por resolver a instância do problema relativa ao perfil base, através do algoritmo H-BLEA. Verificámos que o tempo de execução do algoritmo aumentou cerca de 28 vezes relativamente ao tempo requerido para a resolução do problema com intervalos de 15 minutos. Este aumento do esforço computacional torna o algoritmo H-BLEA impraticável quando se adota uma discretização muito detalhada do período de planeamento. Resolvemos a mesma instância usando os algoritmos BLPSO e BLEA. No algoritmo BLPSO verificou-se um aumento do tempo de execução de 25% e o algoritmo BLEA manteve aproximadamente o mesmo tempo de execução. Estes valores evidenciam

a escalabilidade dos algoritmos BLPSO e BLEA. Pretendemos também analisar a capacidade dos algoritmos BLPSO e BLEA na resolução do problema para os diferentes perfis e cenários e para esta discretização do período de planeamento, avaliando as soluções obtidas. Assim, resolvemos todas as nove instâncias do problema considerando a discretização do período de planeamento em intervalos de 5 minutos, através dos algoritmos BLPSO e BLEA. Os resultados obtidos foram semelhantes aos alcançados para a discretização temporal em intervalos de 15 minutos, o que permite concluir que ambos os algoritmos são escaláveis para problemas de maior dimensão, mantendo os mesmos níveis de desempenho.

4.7 Considerações finais

Neste trabalho considerámos uma representação do problema da interação retalhista-consumidor de energia elétrica através de um modelo BL. Desenvolvemos um algoritmo híbrido H-BLEA e dois algoritmos baseados em populações BLPSO e BLEA para determinar os preços ótimos de energia elétrica a serem definidos pelo retalhista de forma a maximizar o seu lucro sujeito à estratégia de resposta do consumidor procurando minimizar a sua conta de eletricidade. A calendarização do período de funcionamento dos equipamentos depende dos preços definidos pelo retalhista e dos requisitos de conforto do consumidor em termos dos períodos de operação dos equipamentos. O algoritmo H-BLEA resolve o problema do nível inferior de forma exata, pelo que se torna impraticável para problemas de grande dimensão. Os algoritmos BLPSO e BLEA foram desenvolvidos para ultrapassar esta limitação. Para avaliar o desempenho dos algoritmos em diferentes circunstâncias, foram considerados três perfis que refletem restrições mais ou menos exigentes do problema do nível inferior. Foram ainda considerados dois cenários adicionais de cada um dos perfis, correspondentes à deslocação dos períodos admissíveis de funcionamento dos equipamentos. Nestes cenários estes períodos foram deslocados para outras regiões do período de planeamento, traduzindo-se em condições mais exigentes devido à maior simultaneidade dos períodos permitidos para o funcionamento dos diferentes equipamentos.

O algoritmo BLPSO teve melhor desempenho do que os algoritmos BLEA e H-BLEA na maior parte dos casos relativamente à melhor solução entre as 30 soluções válidas obtidas a partir de corridas independentes para as 9 instâncias, mas também

apresentou uma maior variabilidade. A capacidade dos algoritmos BLPSO e BLEA para explorar múltiplas soluções ótimas alternativas para o problema do nível inferior, o que pode representar diferentes valores de lucro para o retalhista para o mesmo custo do consumidor, justifica o facto de estes algoritmos geralmente obterem melhores valores para o lucro do retalhista do que o algoritmo H-BLEA para o valor máximo alcançado em todas as corridas.

Os resultados obtidos com os outros dois cenários reproduziram um comportamento dos algoritmos semelhante ao que tinha sido verificado na versão original do respetivo perfil. Estes resultados reforçam a estabilidade dos algoritmos relativamente ao problema considerado.

No problema apresentado o objetivo do retalhista é maximizar o lucro e o objetivo do consumidor é minimizar o custo. No entanto, o consumidor pode também querer considerar explicitamente o objetivo de minimizar o desconforto sentido pela deslocação temporal do período de funcionamento dos equipamentos em relação ao período preferido. Neste caso obtemos um problema multiobjetivo para o consumidor (nível inferior), para o qual apresentaremos um novo modelo no próximo capítulo.

5 Um modelo BL semi-vetorial para o estudo da interação retalhista - consumidor de energia elétrica

5.1. Motivação

Os problemas BL com uma única função objetivo no nível superior e múltiplas funções objetivo no nível inferior são designados problemas BL semi-vetoriais. Neste caso, dado termos no nível inferior um problema multiobjetivo, não existe uma solução ótima do problema do nível inferior para cada decisão do líder mas um conjunto de soluções ótimas de Pareto. Surge, assim, a dificuldade de o líder antecipar qual a solução que o seguidor irá escolher de entre um conjunto de soluções ótimas de Pareto do problema do nível inferior que conduzem a valores distintos para a função objetivo do líder. A abordagem otimista assume que o seguidor seleciona a solução que é melhor para o líder. No caso oposto, a abordagem pessimista assume que o seguidor escolhe a pior solução na perspectiva do líder.

A designação de problema BL semi-vetorial para o problema BL com múltiplas funções objetivo no nível inferior foi introduzida por Bonnel (2006), que estabeleceu as condições de otimalidade necessárias para uma abordagem otimista. Bonnel e Morgan (2006) propuseram uma abordagem baseada numa função penalidade para a resolução do problema, mas não apresentaram resultados numéricos. Ankhili e Mansouri (2009) desenvolveram um método de função penalidade exato para o problema BL semi-vetorial em que o problema do nível inferior é linear. Na sequência do trabalho de Ankhili e Mansouri, Zheng e Wan (2011) propuseram um novo método de função penalidade (com dois parâmetros de penalidade) para tratar o mesmo problema. Calvete e Galé (2011) abordaram igualmente o caso linear para o problema do nível inferior, considerando ainda que todas as restrições do problema BL são lineares e que a função objetivo do nível superior é quase-côncava. Estes autores propuseram um algoritmo enumerativo exato e uma abordagem heurística baseada num algoritmo genético. Todos os trabalhos mencionados consideram a formulação otimista do problema BL semi-vetorial. Mais recentemente, Liu et al. (2014) desenvolveram as condições de otimalidade necessárias

para a formulação pessimista usando uma técnica de escalarização (que converte o problema do nível inferior num programa escalar paramétrico), seguindo uma abordagem próxima da apresentada em Dempe et al. (2013) para a formulação otimista.

Vamos apresentar de seguida as características das abordagens pessimista e otimista no problema BL semi-vetorial, analisando o risco de se assumir uma abordagem otimista numa situação de decisão real.

5.2 Abordagens pessimista vs. otimista no problema BL semi-vetorial

Como já referimos nos capítulos 3 e 4, um problema BL pode ter várias funções objetivo no nível superior, no nível inferior, ou em ambos os níveis. A existência de múltiplos objetivos aumenta significativamente a dificuldade do problema, em particular a existência de múltiplos objetivos no nível inferior com ou sem múltiplos objetivos no nível superior.

Num problema BL semi-vetorial (BLSV) o problema do nível inferior é um problema de otimização multiobjetivo. As dificuldades inerentes à antecipação da reação do seguidor são abordadas em Nishizaki e Sakawa (1999). Estes autores propuseram um algoritmo iterativo para o problema BL com múltiplas funções objetivo nos dois níveis que assume que o líder tem alguma expectativa sobre a reação do seguidor, podendo ser otimista, pessimista ou o líder tem conhecimento sobre as preferências do seguidor (uma antecipação resultante do comportamento passado do seguidor).

Note-se que, também no caso de problemas BL mono-objetivo, a solução para o seguidor pode não ser única se existirem soluções ótimas alternativas para o problema do nível inferior. Neste caso, a abordagem mais comum é a otimista, assumindo-se que existe algum tipo de cooperação entre o líder e o seguidor ou uma compensação efetuada pelo líder com o intuito de influenciar a decisão do seguidor. A maioria da investigação teórica e algorítmica sobre problemas de otimização BL assume que a solução ótima do problema do nível inferior é única ou adota a abordagem otimista. A abordagem pessimista é considerada mais difícil de tratar.

Também no caso de problemas BL multiobjetivo ou semi-vetoriais a maior parte da literatura adota a abordagem otimista, assumindo que o seguidor seleciona sempre a solução ótima de Pareto do problema do nível inferior que é melhor para o líder. No entanto, num problema BLSV cada solução do problema do nível inferior é uma solução de compromisso entre as múltiplas funções objetivo, não se tratando de soluções

alternativas com o mesmo valor para uma única função objetivo (do nível inferior). No caso de ter múltiplas funções objetivo, o seguidor não é, em geral, indiferente em relação à escolha de uma determinada solução. Esta escolha depende das preferências do seguidor face às suas múltiplas funções objetivo. Poderá, pois, ser irrealista assumir uma perspectiva otimista, considerando que o seguidor irá escolher a melhor solução para o líder.

5.3 Problema BL bi-objetivo no nível inferior para a interação retalhista-consumidor no mercado de eletricidade

Consideremos de novo o problema da interação retalhista – consumidor no mercado de eletricidade apresentado no capítulo 4. Na formulação apresentada estabelecemos que a única função objetivo do consumidor (decisor do nível inferior) correspondia à minimização da sua conta de eletricidade. Vamos agora admitir que, adicionalmente, o consumidor pretende considerar uma outra função objetivo relativa ao desconforto resultante da deslocação temporal do funcionamento dos equipamentos que adota, face aos padrões habituais, para tentar diminuir a conta de eletricidade. Assim, nos períodos mais preferidos para a utilização dos equipamentos o desconforto será mínimo. Ao invés, nos períodos menos desejados a função desconforto terá um valor máximo. O objetivo do retalhista continua a ser a maximização do seu lucro. Este problema corresponde a um modelo BL semi-vetorial, bi-objetivo no nível inferior, que pode ser representado pela formulação seguinte:

$$\max F = \sum_{i=1}^I x_i \sum_{t \in P_i} \left(b_t + \sum_{j=1}^J p_{jt} \right) - \sum_{t=1}^T \pi_t \left(b_t + \sum_{j=1}^J p_{jt} \right) \quad (5.1)$$

s.a

$$x_i \leq \bar{x}_i \quad i = 1, \dots, I \quad (5.2)$$

$$x_i \geq \underline{x}_i \quad i = 1, \dots, I \quad (5.3)$$

$$\frac{1}{T} \sum_{i=1}^I \bar{P}_i x_i = x^{AVG} \quad (5.4)$$

$$\min f_1 = \sum_{i=1}^I \left(x_i \sum_{t=T_{i-1}+1}^{T_i} \left(b_t + \sum_{j=1}^J p_{jt} \right) \right) \quad (5.5)$$

$$\min f_2 = \sum_{j=1}^J \sum_{t=1}^T e_{jt} u_{jt}$$

s. a

$$u_{jt} = \begin{cases} 1 & \text{se } z_j \leq t \leq z_j + d_j - 1 \\ 0 & \text{caso contrário} \end{cases} \quad j = 1, \dots, J; t = 1, \dots, T \quad (5.6)$$

$$p_{jt} = g_j (t - z_j + 1) u_{jt} \quad j = 1, \dots, J; t = 1, \dots, T \quad (5.7)$$

$$\sum_{j=1}^J p_{jt} + b_t \leq C_t \quad t = 1, \dots, T \quad (5.8)$$

$$T1_j \leq z_j \leq T2_j - d_j + 1 \quad j = 1, \dots, J \quad (5.9)$$

Este modelo é obtido a partir do Modelo BL1 apresentado no capítulo 4, adicionando uma nova função objetivo no nível inferior (f_2) para incorporar no processo de otimização o possível desconforto sentido pelo consumidor pela deslocação temporal da operação dos seus equipamentos em relação aos períodos especificados como desejáveis. Assim, o consumidor pretende minimizar a sua conta de eletricidade (f_1) e o desconforto (f_2). Na expressão da função objetivo f_2 (5.5), e_{jt} representa um coeficiente de desconforto associado ao funcionamento do equipamento j no intervalo de tempo t . Este coeficiente varia entre 0 e 1, em que 0 corresponde aos intervalos de tempo de maior preferência para o consumidor e 1 representa o máximo desconforto, embora o consumidor ainda admita o funcionamento do equipamento nestes intervalos de tempo.

A minimização da conta de eletricidade é alcançada calendarizando o funcionamento dos equipamentos para períodos onde o custo da eletricidade seja mais

reduzido face aos preços definidos pelo retalhista para cada período. A minimização do desconforto é obtida quando o consumidor calendariza o funcionamento dos seus equipamentos para períodos onde o funcionamento dos equipamentos não é penalizado. A função objetivo relativa ao desconforto é definida pelo consumidor tendo em conta os seus padrões habituais de utilização dos serviços de energia bem como a sua disponibilidade para aceitar eventuais ajustamentos nas suas rotinas.

Os objetivos de minimização do custo da energia e do desconforto são, em geral, conflitantes. Notemos que os preços da energia tendem a ser superiores em períodos de maior consumo refletindo os maiores custos incorridos pelo retalhista nestes períodos. Estes períodos de maior consumo correspondem a padrões de utilização dos equipamentos preferidos pela maioria dos consumidores, ou seja, tendencialmente o desconforto será menor nestes períodos (tendo em conta as rotinas típicas de utilização dos serviços de energia). Esta relação entre os objetivos do consumidor conduz à necessidade de procurar soluções de compromisso entre a minimização do custo da energia e a minimização do desconforto. Por outro lado, verifica-se também neste problema uma relação direta entre o custo da energia para o consumidor e o lucro do retalhista. Assim, quando a conta de energia do consumidor aumenta, o lucro do retalhista também aumenta e, inversamente, a redução da conta de energia do consumidor corresponde à redução do lucro do retalhista.

Tratando-se de um modelo BLSV, a resolução do problema do consumidor (nível inferior) corresponde a obter um conjunto de soluções ótimas de Pareto. Considerando que o retalhista adota uma abordagem otimista, e tendo em conta a relação entre os objetivos acima descritos, a solução correspondente conduz a um maior custo da energia para o consumidor. Neste caso pressupõe-se que o consumidor privilegia o conforto (i.e., favorece a minimização de f_2), sujeitando-se assim a um maior custo (i.e., admitindo um maior valor de f_1). Ou seja, perante um conjunto de soluções ótimas de Pareto, o consumidor seleciona a que minimiza o seu desconforto.

Note-se, no entanto, que o pressuposto da abordagem otimista pode não ser realista e outras opções devem ser exploradas. Num comportamento oposto ao anterior, o retalhista pode assumir que o consumidor escolhe a solução ótima de Pareto com um menor custo de energia, admitindo que o consumidor privilegia a minimização da sua conta de eletricidade aceitando assim incorrer em algum desconforto. Do ponto de vista do retalhista, esta opção corresponde a uma abordagem pessimista do problema BL.

Para uma análise mais completa do problema, tanto a solução otimista como a pessimista deverão ser obtidas com o intuito de analisar as consequências dos diferentes comportamentos do consumidor. Estas soluções balizam um conjunto de soluções ótimas de Pareto no nível inferior a que correspondem valores da função objetivo do retalhista (lucro) que se situam no intervalo definido pelos correspondentes valores da função objetivo associados às soluções otimista e pessimista. O conhecimento das soluções extremas (otimista e pessimista) permite ao retalhista identificar antecipadamente o intervalo de valores onde estará contido o lucro correspondente à solução que será escolhida pelo consumidor, para um dado vetor de preços de energia estabelecido pelo retalhista. Apresentamos de seguida as formulações matemáticas que definem a solução otimista e a solução pessimista no problema BLSV (5.1) – (5.9).

Considerando uma abordagem otimista por parte do líder, o problema BLSV pode ser formulado como:

$$\max_{x,z} \{F(x,z) : z \in \Psi(x), G(x) \leq 0\}$$

em que $\Psi(x)$ representa o conjunto das soluções ótimas de Pareto do problema do nível inferior (5.5) – (5.9) para um dado x e $G(x) \leq 0$ as restrições do nível superior (5.2) – (5.4).

Admitindo uma abordagem pessimista, o líder escolhe a solução x que tem melhor desempenho na perspectiva da “pior” resposta do seguidor entre as soluções ótimas de Pareto do seu problema. Neste caso o problema BLSV pode ser formulado como:

$$\max_x \left\{ \min_z \{F(x,z) : z \in \Psi(x)\} : G(x) \leq 0 \right\}$$

5.4 Algoritmos populacionais para o problema BL semi-vetorial para a interação retalhista-consumidor no mercado de eletricidade

Para resolver o problema (5.1) – (5.9) vamos utilizar as abordagens algorítmicas baseadas em PSO e EA desenvolvidas no capítulo anterior, adaptando-as às características do problema BL semi-vetorial. A existência de duas funções objetivo no nível inferior influencia as operações dos algoritmos no problema do nível superior. Apresentamos de seguida a descrição de cada um dos algoritmos BL-MOPSO e BL-MOEA, baseados em otimização por enxame de partículas e em algoritmos evolucionários, respetivamente.

5.4.1 Algoritmo BL-MOPSO

O problema BL semi-vetorial (5.1) – (5.9) contém um problema com duas funções objetivo no nível inferior e a sua resolução requer, em geral, a obtenção de um conjunto de soluções não dominadas que caracterizem a frente ótima de Pareto do problema. Assim, o algoritmo BLPSO apresentado no capítulo 4 deverá ser modificado, de forma a utilizar um algoritmo multiobjetivo MOPSO para resolver o problema do nível inferior, tal como foi considerado no capítulo 3. Apesar de focarmos este estudo no cálculo das soluções extremas, otimista e pessimista, o desenvolvimento desta metodologia permite obter um algoritmo genérico que poderá ser usado para outros problemas BLSV e/ou para o cálculo de outras soluções que resultem da incorporação de preferências do consumidor. Notemos, no entanto, que para este problema concreto não era necessário caracterizar toda a frente ótima de Pareto do nível inferior para cada instanciação das variáveis do nível superior porque sabemos que as soluções otimista e pessimista estão nos extremos da frente ótima de Pareto do nível inferior. As alterações no algoritmo BLPSO ocorrem também na seleção dos melhores indivíduos do problema do nível superior. Para lidar com a característica multiobjetivo deste problema são utilizadas as medidas de não dominância ND_l e de distância de *crowding* CD_l . Para guiar a pesquisa no nível inferior em cada sub-população s , é utilizado um arquivo A_s de soluções não dominadas. No início do procedimento de otimização do nível inferior para cada sub-população, este arquivo começa por conter as soluções não dominadas existentes nessa sub-população. No final de cada iteração do processo de resolução do problema do nível inferior, este arquivo é atualizado com as soluções não dominadas encontradas. Notemos que, relativamente ao problema BLSV, tanto a escolha da melhor posição de cada partícula, como da melhor solução da população, estão dependentes da abordagem seguida. O algoritmo BL-MOPSO incorpora estas modificações, sendo definidas duas versões do algoritmo, uma para a abordagem pessimista e outra para a abordagem otimista. As duas versões distinguem-se na seleção da melhor partícula para o nível superior, efetuada entre os elementos do arquivo A_s de cada sub-população s , após ser realizado o processo de resolução do problema do nível inferior sobre essa sub-população. Assim, na versão otimista é selecionada a partícula com maior valor de F e na versão pessimista é selecionada a partícula com menor valor de F . Em qualquer uma das versões, a melhor solução de todas as sub-populações (candidata a solução final) é a que apresenta o maior

valor de F de entre as partículas selecionadas de cada sub-população s . Apresentamos de seguida o pseudo-código deste algoritmo.

Algoritmo 5.1

Algoritmo BL-MOPSO - pseudo-código

Inicializa Pop com n_s sub-populações Pop_s com N_l partículas cada: gerar aleatoriamente n_s vetores admissíveis x e, para cada uma, gera aleatoriamente N_l vetores admissíveis z .

Avalia cada indivíduo da população (x, z) calculando $F(x, z)$ e $f(x, z)$.

Inicializa a melhor posição de cada partícula:

$$pbest1^{s,n_l} = \left(pbest1_x^s, pbest1_z^{s,n_l} \right) = \left(x^s, z^{s,n_l} \right), s = 1, \dots, n_s, n_l = 1, \dots, N_l.$$

Atribui a pre_gbest a melhor partícula de Pop de acordo com F , para ser usada como $gbest$ na primeira iteração.

Para $k = 1, \dots, K$

Para $s = 1, \dots, n_s$

Repete

Atualiza a componente x^s da partícula usando (4.24) e (4.25) com $y = x^s$, $pbest1_x^s$ e $gbest_x$

Procedimento de reparação (x^s)

Até x^s ser admissível para o nível superior

$$A_s = \emptyset$$

Insera em A_s todas as partículas não dominadas da sub-população s e ordena A_s por ordem decrescente de F

Inicializa $gbest_z^s$ com o primeiro elemento de A_s (versão otimista) ou com o último elemento de A_s (versão pessimista)

Inicializa $pbest2^{s,n_l} = \left(pbest2_x^s, pbest2_z^{s,n_l} \right) = \left(x^s, z^{s,n_l} \right), n_l = 1, \dots, N_l$

Para $k_l = 1, \dots, K_l$

Para $n_l = 1, \dots, N_l$

$$l = 0$$

Repete

$$l \leftarrow l + 1$$

Atualiza a componente z^{s,n_l} de cada partícula usando (4.24) e (4.25) com $y = z^{s,n_l}$, $pbest2_z^{s,n_l}$ e $gbest_z^s$

Se z^{s,n_l} não satisfaz as restrições (5.10) **então** empurra-a para os limites mais próximos

Se z^{s,n_l} não satisfaz a restrição (5.9) e $l > L$ **então** z^{s,n_l} é reinicializada e $l = 0$

Até z^{s,n_l} ser admissível para o nível inferior

Atualiza a melhor posição $pbest2^{s,n_l}$ alcançada pela partícula n_l de Pop_s de acordo com f : **Se** $f(x^s, z^{s,n_l}) \prec f(pbest2^{s,n_l})$ **então**
 $pbest2_z^{s,n_l} \leftarrow z^{s,n_l}$

Termina Para n_l

Atualiza o arquivo A_s das soluções não dominadas da sub-população s e ordena A_s por ordem decrescente de F .

Atualiza $gbest_z^s$ com o primeiro elemento de A_s (versão otimista) ou com o último elemento de A_s (versão pessimista)

Termina Para k_l

Termina Para s

Para $s = 1, \dots, n_s$

Para $n_l = 1, \dots, N_l$

Se $F(x^s, z^{s,n_l}) > F(pbest1^{s,n_l})$ **então**

$pbest1^{s,n_l} \leftarrow (x^s, z^{s,n_l})$ // Atualiza $pbest1^{s,n_l}$

Termina Para n_l

Se $(k = 1, s = 1)$ **ou** $F(x^s, gbest_z^s) > F(gbest)$ **então**

$gbest \leftarrow (x^s, gbest_z^s)$ // Inicializa ou atualiza $gbest$

Termina Para s

Termina Para k

Devolve $gbest$

5.4.2 Algoritmo BL-MOEA

O algoritmo BL evolucionário (BL-MOEA) para resolução do problema BL semi-vetorial é desenvolvido com base no algoritmo BLEA, desenhado para problemas BL mono-objetivo em ambos os níveis. Para resolver o problema multiobjetivo será utilizado o algoritmo NSGA-II, cujo bom desempenho foi já por nós verificado anteriormente em diversos estudos. Além da utilização do algoritmo NSGA-II para resolver o problema do nível inferior, será também modificado o processo de seleção de cada nova sub-população a incluir na população da nova geração. No algoritmo BLEA a escolha de cada sub-população é realizada por torneio binário, vencendo a que tiver o indivíduo com maior valor da função objetivo do nível superior. No caso do problema BL semi-vetorial, a escolha depende da abordagem seguida pelo líder. Na abordagem otimista, o líder escolhe a sub-população que tiver o indivíduo não dominado no nível inferior com maior valor da função objetivo do nível superior. Na abordagem pessimista é primeiro selecionado o

conjunto das soluções pessimistas, constituído pela solução não dominada de cada sub-população pior para o líder, i.e., com menor valor de F . Depois é escolhida a sub-população que contém a solução do conjunto das soluções pessimistas que seja melhor para o líder, i.e., com maior valor de F .

Apresentamos de seguida o pseudo código do algoritmo BL-MOEA.

Algoritmo 5.2

Algoritmo BL-MOEA – pseudo-código

(Passo 1) Inicializa $Pop1$ contendo n_s sub-populações $Pop1_s$ com N_l indivíduos cada: gera aleatoriamente n_s vetores admissíveis x e, para cada um, gera aleatoriamente N_l vetores admissíveis z .

Avalia cada indivíduo da população (x,z) calculando $F(x,z)$ e $f(x,z)$.

Para $k = 1, \dots, K$

(Passo 2)

Para $s = 1, \dots, n_s$

Repete

Seleciona x' e x'' de $Pop1_s$ um aleatoriamente e o outro por torneio binário

$x^{cross} \leftarrow$ Cruzamento de x' e x''

$x^{mut} \leftarrow$ Mutação de x^{cross}

Procedimento de reparação (x^{mut})

Até x^{mut} ser admissível para o nível superior

Cria $Pop2_s$ com x^{mut} e todos os z em $Pop1_s$:

$$Pop2_s = \left\{ (x^{mut}, z^{s,1}), (x^{mut}, z^{s,2}), \dots, (x^{mut}, z^{s,N_l}) \right\}$$

Para $k_l = 1, \dots, K_l$

$Desc_s = \phi$

Para $n_l = 1, \dots, N_l$

Repete

Seleciona $z^{s,'}$ e $z^{s,''}$ de $Pop2_s$, um aleatoriamente e o outro por torneio binário

$z^{s,cross} \leftarrow$ Cruzamento de $z^{s,'}$ e $z^{s,''}$

$z^{s,mult} \leftarrow$ Mutação de $z^{s,cross}$

Até $z^{s,mult}$ ser admissível para o nível inferior

Insera $(x^{mut}, z^{s,mult})$ como (x^s, z^{s,n_l}) na sub-população $Desc_s$

Termina Para n_l

Classifica cada membro da subpopulação, calculando ND_l e CD_l

Seleciona os melhores N_l indivíduos de $Pop2_s$ e de $Desc_s$ de acordo com os valores ND_l e CD_l , lexicograficamente, para criar a

nova $Pop2_s$ para a próxima iteração ($k_l + 1$).

Termina Para k_l

Termina Para s

/* seleciona a população para a próxima geração */

(Passo 3)

Para $s = 1, \dots, n_s$

$sol1(s) \leftarrow$ solução com maior F (versão otimista) ou com menor F (versão pessimista), entre as soluções não dominadas para o nível inferior de $Pop1_s$

$sol2(s) \leftarrow$ solução com maior F (versão otimista) ou com menor F (versão pessimista), entre as soluções não dominadas para o nível inferior de $Pop2_s$

Termina Para

Atualiza $Pop1$ para a geração $k+1$ copiando a sub-população de $Pop1 \cup Pop2$ correspondente à solução com maior F entre as soluções de $sol1 \cup sol2$ e efetuando $n_s - 1$ torneios binários sem reposição determinados pelo maior valor de F entre os elementos de $sol1$ e os elementos de $sol2$ para decidir as outras sub-populações de $Pop1 \cup Pop2$ a copiar

Termina Para k

$bestsol \leftarrow$ solução com maior F entre os elementos da sub-população 1 de $Pop1$ não dominados para o problema do nível inferior

Devolve $bestsol$

5.5 Exemplo de aplicação dos algoritmos BL-MOPSO e BL-MOEA no problema BL semi-vetorial para a interação retalhista-consumidor de energia elétrica

Consideremos o problema do estabelecimento de preços por um retalhista no mercado de energia elétrica. O retalhista pretende maximizar o seu lucro e o consumidor pretende minimizar a sua conta de eletricidade, tirando partido dos preços diferenciados em diferentes períodos horários, bem como o possível desconforto sentido pela deslocação temporal da utilização dos seus equipamentos para períodos menos preferidos face aos seus hábitos. Apresentamos de seguida os dados do problema (5.1) – (5.9) para o caso em estudo, cuja formulação foi apresentada na secção 5.3.

5.5.1 Dados do problema

Consideremos o problema de calendarização do consumo de energia elétrica de um grupo de consumidores domésticos, clientes do mesmo retalhista, que apresentam o mesmo padrão de preferências de utilização dos serviços de energia. O retalhista define um preço para cada um dos 7 sub-períodos de preços, verificando a restrição do preço médio pré-definido. Os equipamentos controláveis são os apresentados no capítulo 4. Assim, além de uma carga base não controlável, os consumidores utilizam 5 equipamentos distintos, cada um com um ciclo de funcionamento e preferências de utilização específicos. Para este efeito, será considerado um perfil de utilização dos equipamentos onde são definidos intervalos de tempo específicos onde cada equipamento pode funcionar (Tabela 5.1).

Tabela 5.1 Períodos de tempo permitidos para a operação de cada equipamento

Equipamento	Intervalo inicial	Intervalo final
Máquina de lavar loiça	1	44
Máquina de lavar roupa	28	65
Aquecedor de água (EWH)	24	45
Veículo elétrico	1	48
Máquina secar roupa	70	96

A função relativa ao desconforto associado a cada equipamento é definida no intervalo $[0,1]$ ao longo de todo o período de planeamento. Os intervalos de utilização dos equipamentos incluem períodos onde o desconforto é mínimo (nulo), máximo (um) e gradualmente crescente ou decrescente. Definimos uma função desconforto para cada um dos equipamentos, ao longo de todo o período de planeamento. Assim, apresentamos nas Fig. 5.1-5.5 a representação gráfica dos coeficientes de desconforto relativos a cada equipamento (coeficientes e_{jt} na função objetivo f_2 do nível inferior – (5.5)).

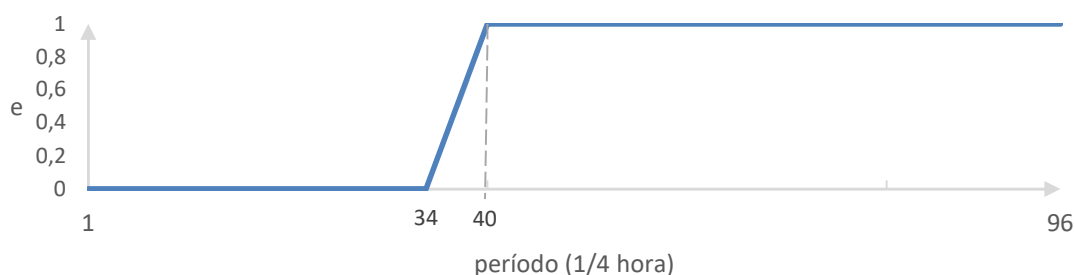


Fig 5.1 Coeficientes de desconforto relativos à utilização da máquina de lavar loiça

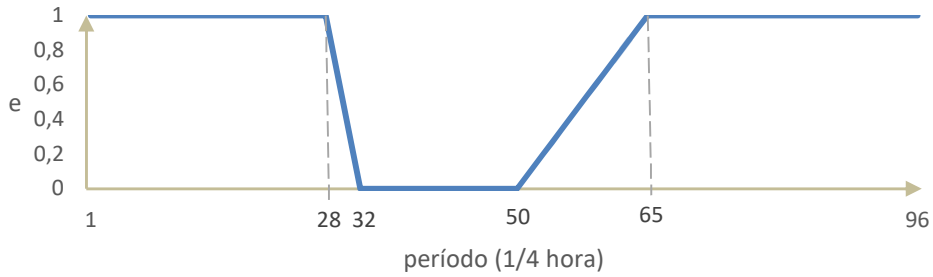


Fig 5.2 Coeficientes de desconforto relativos à utilização da máquina de lavar roupa

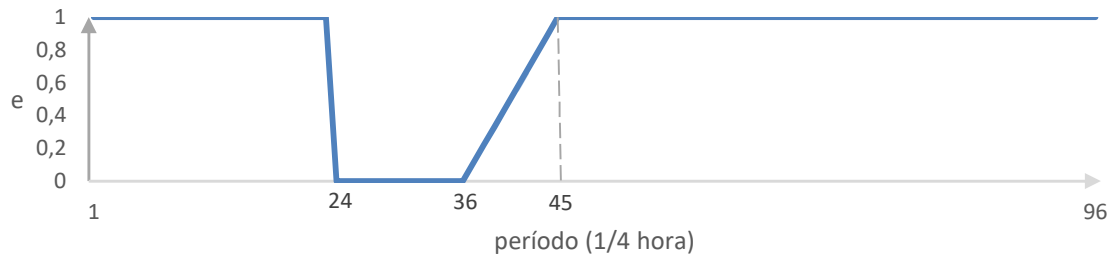


Fig 5.3 Coeficientes de desconforto relativos à utilização do aquecedor elétrico de água

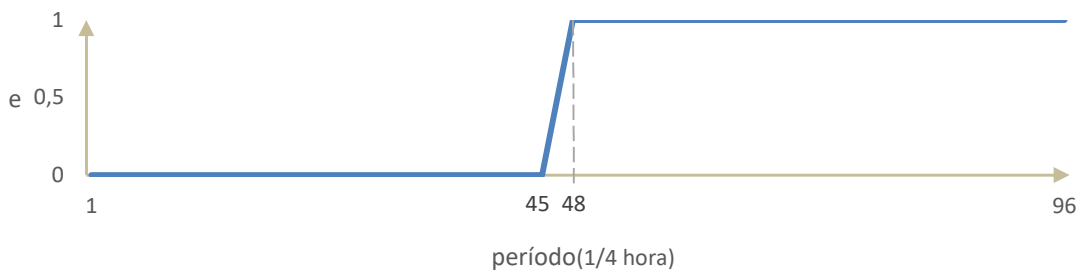


Fig 5.4 Coeficientes de desconforto relativos ao carregamento do veículo elétrico

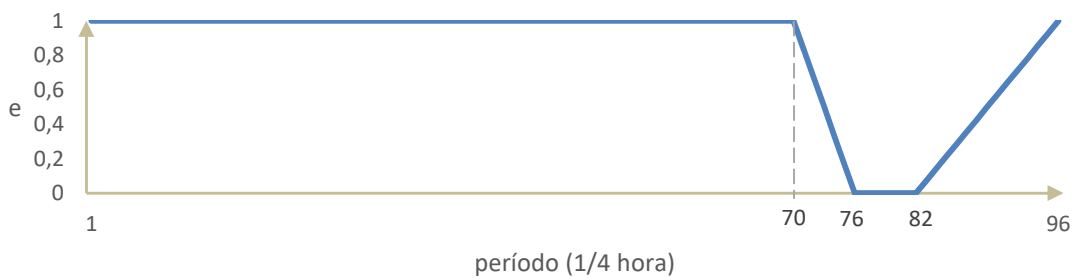


Fig 5.5 Coeficientes de desconforto relativos à utilização do secador de roupa

No algoritmo BL-MOPSO definimos os parâmetros $N = 400$, $N_l = 50$, $K = 100$ e $K_l = 80$. Para as operações do algoritmo MOPSO, utilizado para resolver o problema do nível inferior, consideramos a parametrização utilizada no algoritmo OMOPSO apresentado no capítulo 2. No algoritmo BL-MOEA consideramos os parâmetros

$N = 320, N_I = 40, K = 100$ e $K_I = 60$. Cada um dos algoritmos é executado 10 vezes para cada instância considerada.

5.5.2 Resultados

Em cada instância do problema as soluções otimista e pessimista são obtidas efetuando execuções independentes do respectivo algoritmo.

As soluções obtidas por cada um dos algoritmos BL-MOPSO e BL-MOEA considerando ambas as perspectivas otimista e pessimista estão apresentadas na Tabela 5.2. Os preços da energia e os períodos de funcionamento dos equipamentos relativos a estas soluções são apresentados nas Tabelas 5.3 e 5.4, respetivamente.

Analisámos as soluções apresentadas na Tabela 5.2, tendo verificado que são efetivamente soluções ótimas de Pareto do problema do nível inferior.

Na perspectiva otimista, o desconforto é nulo nas soluções obtidas por ambos os algoritmos, o que decorre de nesta perspectiva se privilegiar o conforto aceitando o aumento do custo. O algoritmo BL-MOPSO consegue obter a melhor solução, garantindo um lucro para o retalhista ligeiramente superior ao obtido pelo algoritmo BL-MOEA. Contudo, no conjunto das 10 corridas efetuadas, o algoritmo BL-MOEA gerou soluções de boa qualidade e com menor variabilidade, como pode ser avaliado pela mediana e pelo *IQR* de F .

Na perspectiva pessimista, o lucro do retalhista diminuiu significativamente nas soluções obtidas por ambos os algoritmos. Note-se que, neste problema, adotar a perspectiva pessimista corresponde a assumir que a prioridade do consumidor é reduzir a conta de eletricidade em detrimento do conforto. O algoritmo BL-MOEA obteve melhores soluções e com menor variabilidade no conjunto das 10 corridas em relação ao algoritmo BL-MOPSO. No nível inferior, o algoritmo BL-MOPSO obteve soluções com menor custo embora com um maior desconforto para o consumidor. Na melhor solução apresentada por este algoritmo, o funcionamento da máquina de lavar roupa foi deslocado para o intervalo 60-65, que corresponde aos menores preços de energia dentro do intervalo onde este equipamento pode ser utilizado (*cf.* Tabela 5.4), conseguindo assim um menor custo.

Note-se, no entanto, que a interpretação de que o algoritmo BL-MOEA obteve melhores soluções do que o algoritmo BL-MOPSO na perspectiva pessimista pode não ser correta por não haver a garantia de que as “piores” soluções para o líder calculadas pelos

algoritmos para cada vetor de preços (a partir das quais é selecionada a solução pessimista como a “melhor das piores”) sejam efetivamente as soluções de mais baixo lucro para o retalhista. Assim, as soluções pessimistas do algoritmo BL-MOEA, aparentemente melhores, podem eventualmente corresponder a uma escolha do consumidor que não seja a pior para o retalhista. Naturalmente, o mesmo pode acontecer com o algoritmo BL-MOPSO. No caso da interpretação de resultados da perspectiva otimista esta questão não se coloca, porque sabemos à partida que as “melhores” soluções para o líder para cada vetor de preços (a partir das quais é selecionada a “melhor das melhores”) apresentam sempre $f_2 = 0$.

Relativamente à definição dos preços da energia por parte do retalhista, verifica-se que os dois algoritmos efetuam um ajustamento dos preços de energia à respetiva perspectiva, diminuindo o preço da energia no período P_6 e aumentando no período P_5 na passagem da perspectiva otimista para a pessimista (*cf.* Tabela 5.3). Note-se que na perspectiva pessimista o consumidor privilegia a diminuição do custo, disponibilizando-se para calendarizar consumos em períodos de preços mais baixos, ainda que com maior desconforto. Assim, analisando a melhor solução encontrada pelo algoritmo BL-MOPSO identificamos a calendarização do funcionamento da máquina de lavar roupa e do secador de roupa para o período P_5 onde se encontram os preços mais baixos, para reduzir o custo de energia. Esta redução é um pouco mitigada pela gestão de preços por parte do retalhista, que subiu os preços do período P_5 relativamente à perspectiva otimista.

Tabela 5.2. Lucro do Retalhista e custo do Consumidor (para a melhor solução de F).

Perspetiva	Algoritmo	Máximo F	Mediana F	IQR F	f_1	f_2
Otimista	BLMOPSO	1922.112	1902.635	28.329	3433.15	0.00
	BLMOEA	1920.290	1918.635	0.28	3413.30	0.00
Pessimista	BLMOPSO	1560.844	1463.831	68.916	3119.53	8.277778
	BLMOEA	1638.177	1528.651	8.41	3175.56	2.722222

Tabela 5.3. Preços da energia para cada sub-período na melhor solução obtida.

Perspetiva	Algoritmo	P_1	P_2	P_3	P_4	P_5	P_6	P_7
		[1, 28]	[29, 38]	[39, 44]	[45, 60]	[61, 76]	[77, 84]	[85, 96]
Otimista	BL-MOPSO	0.10	0.24	0.12	0.120001	0.033036	0.24	0.070618
	BL-MOEA	0.10	0.24	0.12	0.120372	0.030692	0.24	0.073248
Pessimista	BL-MOPSO	0.10	0.24	0.12	0.120245	0.041483	0.234395	0.062768
	BL-MOEA	0.10	0.24	0.12	0.120744	0.065949	0.184779	0.062558

Tabela 5.4. Intervalos iniciais e finais de funcionamento dos equipamentos na melhor solução obtida.

Perspetiva	Algoritmo	Máq. lav. loiça	Máq.lavar roupa	EWH	Veiculo elétrico	Secador de roupa
Otimista	BL-MOPSO	1-5	39-44	28-32	5-40	76-78
	BL-MOEA	1-5	39-44	28-32	5-40	76-78
Pessimista	BL-MOPSO	1-5	60-65	39-43	5-40	74-76
	BL-MOEA	1-5	41-46	38-42	5-40	74-76

5.6 Considerações finais

A consideração de uma nova função objetivo relativa ao desconforto sentido pelo consumidor associado ao deslocamento temporal da utilização dos equipamentos transforma o problema da interação retalhista-consumidor de energia elétrica num problema em dois níveis semi-vetorial. Foi desenvolvido um modelo BL semi-vetorial considerando que o retalhista pretende maximizar o lucro e o consumidor pretende minimizar a conta de eletricidade e o desconforto. Sendo o problema do nível inferior multiobjetivo, para cada definição de preços de eletricidade (valores das variáveis do nível superior), o problema do nível inferior apresenta um conjunto de soluções ótimas de Pareto entre as quais o consumidor deverá selecionar uma. A dificuldade por parte do líder (retalhista) em antecipar a reação do consumidor torna este problema particularmente difícil. Assim, considerámos a posição do líder (retalhista) segundo duas perspetivas relativamente à reação do consumidor: otimista, em que se assume que o consumidor escolhe a solução ótima de Pareto do problema do nível inferior com maior lucro para o retalhista (i.e., o consumidor privilegia o conforto em detrimento do custo), e pessimista, em que se assume que o consumidor escolhe a solução ótima de Pareto com menor lucro para o retalhista (i.e., o consumidor privilegia o custo em detrimento do conforto). Desenvolvemos os algoritmos BL-MOPSO e BL-MOEA para resolução do problema através de meta-heurísticas populacionais baseadas em otimização por enxame de partículas e em algoritmos evolucionários. Verificámos que as soluções não dominadas do problema do nível inferior obtidas por cada um dos algoritmos são efetivamente soluções ótimas de Pareto deste problema. Assim, os algoritmos alcançaram soluções admissíveis do problema BLSV em qualquer das instâncias consideradas. Verificamos que na perspetiva pessimista os lucros do retalhista diminuem significativamente.

Na perspetiva otimista o algoritmo BL-MOPSO obteve a melhor solução, mas o algoritmo BL-MOEA gerou soluções consistentemente de elevada qualidade. Na

perspetiva pessimista o algoritmo BL-MOEA obteve melhores soluções e com menor variabilidade em relação ao algoritmo BL-MOPSO, apesar de essas soluções não serem as verdadeiras pessimistas, ou seja, existem outras soluções ótimas de Pareto com menor lucro para o retalhista para o mesmo conjunto de preços.

6 Conclusões e desenvolvimentos futuros

A otimização multiobjetivo permite tratar de forma mais realista muitos problemas ao incluir explicitamente como funções objetivo os diferentes eixos de avaliação do mérito das soluções potenciais. No caso de termos um problema de otimização com diferentes decisores relacionados através de uma hierarquia, o modelo em dois níveis BL é uma representação adequada. Os modelos multiobjetivo e/ou BL são, em geral, difíceis de resolver usando métodos de programação matemática, em particular quando envolvem funções não lineares e/ou são de natureza combinatória. Neste contexto, as meta-heurísticas têm sido cada vez mais utilizadas para resolver este tipo de problemas pelo facto de, em geral, não requererem a satisfação de condições exigentes de convexidade/diferenciabilidade, serem fáceis de implementar e produzirem bons resultados com um esforço computacional aceitável.

Neste trabalho abordámos problemas de otimização multiobjetivo e BL. Começámos a nossa investigação pela análise de meta-heurísticas populacionais representativas do estado da arte e desenvolvemos um novo algoritmo, *Enhanced Multiobjective Electromagnetism-like Mechanism* (EMOEM), para a resolução de problemas multiobjetivo baseado na abordagem *Electromagnetism-like mechanism* (EM). Este algoritmo melhorou significativamente o desempenho do algoritmo MOEM anteriormente proposto por outros autores. Contudo, existe uma ampla investigação desenvolvida sobre as meta-heurísticas PSO e EA, o que permitiu obter algoritmos com elevado desempenho, como confirmámos na análise comparativa que efetuámos. Assim, prosseguimos o nosso trabalho utilizando as meta-heurísticas PSO e EA para o desenvolvimento de algoritmos dedicados a problemas de otimização em dois níveis.

Os problemas em dois níveis são mais complexos pelo facto de existir uma interação entre dois níveis de otimização. Estes problemas são ainda mais difíceis de resolver no caso do problema BL semi-vetorial ou multiobjetivo em ambos os níveis. Começámos por analisar os problemas BL do ponto de vista algorítmico. Desenvolvemos o algoritmo MO-BLPSO baseado na abordagem PSO para resolução do problema BL com múltiplas funções objetivo em ambos os níveis. Testámos o algoritmo MO-BLPSO em problemas onde é conhecida a verdadeira frente ótima de Pareto e comparámos os resultados com os do algoritmo BLEMO, anteriormente proposto por outros autores. O algoritmo MO-BLPSO mostrou-se muito competitivo nesta análise.

Utilizámos um modelo de otimização BL para representar o problema da interação entre um retalhista e consumidores no fornecimento de energia elétrica ao segmento residencial, no contexto de tarifas variáveis. Começámos por construir um modelo mono-objetivo em ambos os níveis, considerando o lucro do retalhista como a função objetivo do nível superior e o custo da energia elétrica como a função objetivo do consumidor no nível inferior. Este modelo considera três perfis de consumidores, relativos à localização no período de planeamento e à amplitude dos períodos em que é permitido o funcionamento dos equipamentos. Desenvolvemos dois algoritmos populacionais, um baseado na meta-heurística PSO (BLPSO) e outro nos algoritmos evolucionários (BLEA), adaptados às características específicas deste problema. Construimos ainda um modelo equivalente onde o problema do nível inferior é transformado num problema de otimização linear inteira mista (MILP) que pode ser resolvido de forma exata. Desenvolvemos uma abordagem híbrida H-BLEA para resolver este problema. Esta abordagem utiliza um algoritmo evolucionário para resolver o problema do nível superior e resolve o problema do nível inferior de forma exata através de um *solver*. A resolução do problema BL pela abordagem híbrida permitiu avaliar o desempenho dos algoritmos populacionais. Os resultados obtidos confirmaram o bom desempenho dos algoritmos BLPSO e BLEA, os quais alcançaram ainda soluções ótimas alternativas para o problema do nível inferior que não tinham sido identificadas pelo algoritmo H-BLEA. Os algoritmos populacionais têm ainda a vantagem de poderem ser utilizados em problemas de maior dimensão. Assim, aumentámos a discretização temporal do período de planeamento e os algoritmos BLPSO e BLEA continuaram a conseguir determinar a solução do problema, evidenciando a sua robustez. Verificámos que com maior discretização temporal o algoritmo H-BLEA se torna impraticável por não ser possível obter a solução do problema num tempo de execução aceitável.

Tendo em vista uma melhor caracterização do problema, considerámos uma função objetivo adicional no nível inferior associada ao possível desconforto sentido pelo consumidor pela deslocação temporal da utilização dos seus equipamentos em relação ao período mais preferido. O decisor líder do problema BL, i.e. o retalhista, tem neste caso uma dificuldade acrescida, na medida em que o problema do nível inferior tem um conjunto de soluções não dominadas e não é possível antecipar qual será a decisão do seguidor. Considerámos os casos extremos em que o líder antecipa que o seguidor escolherá a decisão mais favorável para o líder (perspetiva otimista) e em que o líder acredita que o seguidor escolherá a solução mais desfavorável para o líder (perspetiva

pessimista). Desenvolvemos os algoritmos BL-MOPSO e BL-MOEA, adaptados dos algoritmos BLPSO e BLEA, para resolver este problema segundo cada uma das perspectivas do líder. Os algoritmos BL-MOPSO e BL-MOEA conseguiram resolver o problema em qualquer das instâncias consideradas.

Verificámos que as meta-heurísticas populacionais utilizadas são adequadas e produziram bons resultados para os problemas considerados. Os algoritmos desenvolvidos poderão ser facilmente adaptados para a resolução de outros problemas deste tipo.

Pretendemos futuramente aprofundar o nosso trabalho desenvolvendo o modelo do problema da interação retalhista-consumidor de energia elétrica de forma a integrar diferentes tipos de consumidores, além dos consumidores residenciais.

É também nosso objetivo introduzir mecanismos de incorporação de preferências para orientação da pesquisa na resolução do problema do nível inferior no modelo BLSV apresentado. Este processo permitirá obter outras soluções deste problema, para além da solução otimista e da solução pessimista.

Referências

- Abo-Sinna, M.A., Baky, A.I. (2007) Interactive Balance Space Approach for Solving Multilevel Multi-objective Programming Problems, *Information Sciences* 177, pp. 3397-3410
- Alikani, M. G., Javadian, N., Tavakkoli-Moghaddan, R. (2009) A novel hybrid approach combining electromagnetism-like method with Solis and Wets local search for continuous optimization problems. *Journal of Global Optimization* 44, pp. 227–234
- Alves, M.J. (2012) Using MOPSO to solve multiobjective bilevel linear problems. In: Marco Dorigo et al. (Eds), *Swarm Intelligence, 8th International Conference, LNCS*, vol.7461, pp. 332–339, Springer-Verlag
- Alves, M.J., Costa, J. P.(2014) An algorithm based on particle swarm optimization for multiobjective bilevel linear problems, *Applied Mathematics and Computation* 247,pp. 547–561
- Alves, M.J., Antunes, C.H., Carrasqueira, P. (2015) A PSO approach to semivectorial bilevel programming: pessimistic, optimistic and deceiving solutions. In: *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation (GECCO'15)*. Madrid, Spain, pp. 599–606
- Alves, M.J., Antunes, C. H., Carrasqueira, P. (2016) A hybrid genetic algorithm for the interaction of electricity retailers with demand response. In: G. Squillero and P. Burelli (Eds.), *EvoApplications 2016, Part I, LNCS 9597*, pp. 459–474, Springer
- Ankhili, Z., Mansouri, A. (2009) An exact penalty on bilevel programs with linear vector optimization lower level. *Eur. J. Oper. Res.* 197, 1, pp. 36-41
- Antunes, C.H., Alves, M. J., Clímaco, J. C. (2016) *Multiobjective Linear and Integer Programming*. Springer International Publishing Switzerland
- Bard, J.F. (1998) *Practical Bilevel Optimization: Algorithms and Applications*. Dordrech: Kluwer Academic Publisher.
- Birbil, S.I., Fang, S. (2003) An electromagnetism-like mechanism for global optimization. *Journal of Global Optimization* 25, pp. 263–282
- Bonnell, H. (2006) Optimality conditions for the semivectorial bilevel optimization problem. *Pac. J. Optim.* 2, 3, pp. 447-468

- Bonnell, H., Morgan, J. (2006) Semivectorial bilevel optimization problem: penalty approach. *J. Optimz. Theory App.*, vol. 131, 3, pp. 365-382
- Bu, S., Yu, F. R., Liu, P. X. (2011) A game-theoretical decision-making scheme for electricity retailers in the smart grid with demand-side management. 2011 IEEE International Conference on Smart Grid Communications - SmartGridComm, pp. 387-391
- Calvete, H. I., Galé, C. (2011) On linear bilevel problems with multiple objectives at the lower level. *Omega* 39, 1, pp. 33-40
- Carrasqueira, P., Alves, M.J., Antunes, C.H. (2014) An improved multiobjective electromagnetism like mechanism algorithm. In: A. I. Esparcia-Alczar and A. M. Mora (eds.). *EvoApplications 2014*, LNCS 8602, pp. 627-638
- Carrasqueira, P., Alves, M.J., Antunes, C.H. (2015a) A Multiobjective Electromagnetism-Like Algorithm with Improved Local Search. In: Almeida, J. P. et al. (eds.) *Operational Research. CIM Series in Mathematical Sciences*, vol 4, pp. 123-144. Springer
- Carrasqueira, P., Alves, M.J., Antunes, C.H. (2015b) A bi-level multiobjective PSO algorithm. In *Proceedings of EMO 2015*, LNCS 9018, Springer, pp. 263-276
- Carrasqueira, P., Alves, M.J., Antunes, C.H. (2017) Bi-level Particle Swarm Optimization and Evolutionary Algorithm Approaches for Residential Demand Response with Different User Profiles. Submitted.
- Deb, K., Pratap, A., Agarwal, S., Meyarivan, T. (2002) A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6 (2), pp. 182- 197
- Deb, K., Sinha, A. (2009) Solving bilevel multi-objective optimization problems using evolutionary algorithms. In: Ehrgott, M. et al. (eds.), *Proceedings of 5th International Conference on Evolutionary Multi-Criterion Optimization (EMO 2009)*, LNCS 5467, pp. 110-124. Springer-Verlag
- Deb, K., Sinha, A. (2010) An efficient and accurate solution methodology for bilevel multi-objective programming problems using a hybrid evolutionary-local-search algorithm. *Evol. Comput.* 18, 3, pp. 403-449

- Deb, K., Thiele, L., Laumanns, M., Zitzler, E. (2005) Scalable Test Problems for Evolutionary Multiobjective Optimization. In: L. J. A. Abraham (ed.), *Evolutionary Multiobjective Optimization. Theoretical Advances and Applications*, pp. 105–145
- Dempe, S. (2002) *Foundations of Bilevel Programming*, Kluwer Academic Publishers, Dordrecht
- Dempe, S. (2009) Bilevel programming implicit function approach. In Floudas and Pardalos (Eds), *Encyclopedia of Optimization*, vol. 2, pp. 260-266
- Dempe, S., Gadhi, N., Zemkoho, A.B. (2013) New optimality conditions for the semivectorial bilevel optimization problem. *J. Optimiz. Theory App.*, vol. 157, 1, pp. 54-74
- Durillo, J.J., Nieto, J.G., Coello, C.A., Luna, F., Alba, E. (2009) Multi-objective particle swarm optimizers: An experimental comparison. 5th International Conference on Evolutionary Multicriterion Optimization. EMO2009, pp. 495–509. Springer
- Eichfelder, C. (2007) Solving nonlinear multiobjective bilevel optimization problems with coupled upper level constraints. Technical Report Preprint No. 320, Preprint Series of the Institute of Applied Mathematics, Univ. Erlangen-Nrnberg, Germany.
- Eichfelder, G. (2010) Multiobjective Bilevel Optimization, *Mathematical Programming*, 123 (2), pp. 419-449
- Fonseca, C.M., Paquete, L., López-Ibáñez, M. (2006) An Improved Dimension-Sweep Algorithm for the Hypervolume. *Proceedings of 2006 IEEE Congress on Evolutionary Computation*, pp.1157-1163
- Glover, F. (1989) Tabu Search - Part I. *ORSA Journal on Computing* , 1, pp. 190-206
- Halter, W., Mostaghim, S. (2006) Bilevel optimization of multi-component chemical systems using particle swarm optimization. In *Proc. of the 2006 Congress on Evolutionary Computation (CEC'06)*, pp. 1240–1247. IEEE Press
- Hooke, R., Jeeves, T.A. (1961) Direct search solution of numerical and statistical problems. *Journal of the ACM*, 8, pp. 212–229
- Kennedy, J, Eberhart, R.C. (1995) Particle swarm optimization. *IEEE International Conference on Neural Network*, pp. 1942–1948
- Kirkpatrick, S., Gelatt, C.D., Vecchi, M. P. (1983) Optimization By Simulated Annealing. *Science*, 220, pp. 671-680

- Liu, B., Wan, Z., Chen, J., Wang, G. (2014) Optimality conditions for pessimistic semivectorial bilevel programming problems. *J. Inequal. Appl.*, 2014:41
- Lu, J., Han, J., Hu, Y., Zhang, G. (2016) Multilevel decision-making: A survey. *Information Sciences* 346–347, pp. 463–487
- Lu, J., Shi, C., Zhang, G. (2006) On bilevel multi-follower decision making: General framework and solutions. *Information Sciences* 176 (11), pp. 1607-1627
- Meng, F.-L., Zeng, X.-J. (2013) An optimal real-time pricing algorithm for the smart grid: a bilevel programming approach. *Imperial College Computing Student Workshop - ICCSW'13*, pp. 81-88
- Meng, F.L., Zeng, X.J. (2014) An optimal real-time pricing for demand-side management: A Stackelberg game and genetic algorithm approach. In: 2014 International Joint Conference on Neural Networks (IJCNN), IEEE, pp. 1703–1710
- Meng, F.L., Zeng, X.J. (2015) Appliance level demand modeling and pricing optimization for demand response management in smart grid. In: 2015 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. IEEE
- Mezura-Montes, E., Reyes-Sierra, M., Coello Coello, C.A. (2008) Multi-Objective Optimization using Differential Evolution: A Survey of the State-of-the-Art. In Chakraborty, U.K. (ed). *Advances in Differential Evolution*, pp. 173-196. Springer, Berlin.
- Mladenović, N., Hansen, P. (1997) Variable neighborhood search, *Comput. Oper. Res.*, 24, pp.1097-1100
- Mohsenian-Rad., A.-H., Leon-Garcia, A. (2010) Optimal residential load control with price prediction in real-time electricity pricing environments. In *IEEE Transactions on Smart Grid*, Vol 1, No.2, pp. 120-133
- Mousa, A.A., El-Shorbagy, M.A., Abd-El-Wahed, W.F. (2012) Local search based hybrid particle swarm optimization algorithm for multiobjective optimization. *Swarm and Evolutionary Computation*, 3, pp. 1–14
- Naji-Azimi, Z., Toth, P., Galli, L. (2010) An electromagnetism metaheuristic for the unicost set covering problem. *European Journal of Operational Research* 205, pp. 290–300

- Nishizaki, I., Sakawa, M. (1999) Stakelberg Solutions to Multiobjective Two-Level Linear Programming Problems, *Journal of Optimization Theory and Applications*, 103, pp. 161-182
- Price, K. (1997) Differential evolution vs. the functions of 2nd ICEO. *IEEE International Conference on Evolutionary Computation (ICEC1997)*. Indianapolis,USA, pp. 153–157
- Reyes-Sierra, M., Coello Coello, C.A. (2005) Improving PSO-based multi-objective optimization using crowding, mutation and e-dominance. In: Coello Coello, C.A., Aguirre, A.H., Zitzler, E.(eds.). *EMO2005. LNCS*, vol. 3410, pp.505-519. Springer.
- Rocha, A.M.A.C., Fernandes, E.M.G.P. (2009) A modified electromagnetism-like algorithm based on a pattern search method. In: Mastorakis, N., Mladenov, V., Kontargyri, V.T. (eds). *Proceedings of the European Computing Conference. Lecture Notes in Electrical Engineering*, vol 2, part 9, chapter 12, pp.1035–1042. Springer, Berlin, Heidelberg.
- Ruuska, S., Miettinen, K., Wiecek, M.M. (2012) Connections between single-level and bilevel multiobjective optimization. In: *Journal of Optimization Theory and Applications*, vol153, issue 1, pp. 60-74
- Ruuska, S., Miettinen, K. (2012) Constructing Evolutionary Algorithms for Bilevel Multiobjective Optimization. In: *IEEE Congress on Evolutionary Computation (CEC 2012)*
- Samadi, P., Mohsenian-Rad, H., Schober, R., Wong, V.W.S. (2012) Advanced demand side management for the future smart grid using mechanism design. *IEEE Transactions on Smart Grid*, Vol. 3, No. 3, pp. 1170-1180
- Sekizaki, S., Nishizaki, I., Hayashida, T. (2016) Electricity retail market model with flexible price settings and elastic price-based demand responses by consumers in distribution network. *Electric Power and Energy Systems*, 81, pp. 371-386
- Shi, Y., Eberhart, R. (1999) Empirical study of particle swarm optimization. *Proceedings of the 1999 Congress on Evolutionary Computation*, pp. 1945-1950
- Shi, X., Xia, H. (2001) Model and interactive algorithm of bi-level multi-objective decision-making with multiple interconnecting decision makers. *J. Multi-Criteria Decision Analysis* 10, pp.27–34

- Sinha, A., Deb, K. (2009) Towards understanding evolutionary bilevel multiobjective optimization algorithm. Technical Report Proceedings of the IFAC Workshop on Control Applications of Optimization (6-8 May, 2009, Jyväskylä, Finland), Kanpur, Indian Institute of Technology, India. (Also KanGAL Report No. 2008006)
- Sinha A., Malo P., Deb K. (2015) Towards Understanding Bilevel Multi-objective Optimization with Deterministic Lower Level Decisions. In: Gaspar-Cunha A., Henggeler Antunes C., Coello C. (eds) Evolutionary Multi-Criterion Optimization. EMO 2015. Lecture Notes in Computer Science, vol. 9018. Springer, pp. 426-443.
- Sinha, A., Malo, P., Deb, K. (2016) Evolutionary algorithm for bilevel optimization using approximations of the lower level optimal solution mapping. *European Journal of Operational Research*, vol. 257, Issue 2, pp. 395-411
- Sinha, A., Malo, P., Deb, K. (2017). Approximated set-valued mapping approach for handling multiobjective bilevel problems. *Computers & Operations Research*, 77, pp. 194-209.
- Soares, A., Gomes, A., Antunes, C.H. (2014a) Categorization of residential electricity consumption as a basis for the assessment of the impacts of demand response actions, *Renewable and Sustainable Energy Reviews* 30, pp. 490–503
- Soares, A., Antunes, C.H., Oliveira, C., Gomes, A. (2014b) A multi-objective genetic approach to domestic load scheduling in an energy management system. *Energy* 77, pp. 144–152
- Storn, R., Price, K. (1995) Differential evolution: a simple and efficient adaptive scheme for global optimization over continuous spaces. International Computer Science Institute, Berkeley. Technical Report TR-95-012.
- Tavakkoli-Moghaddam, R., Khalili, M., Naderi, B. (2009) A hybridization of simulated annealing and electromagnetism-like mechanism for job shop problems with machine availability and sequence-dependent setup times to minimize total weighted tardiness. *Soft Computing* 13(10), pp.995-1006
- Tsou, C.-S., Kao, C.-H. (2008) Multi-objective inventory control using electromagnetism-like meta-heuristic. *International Journal of Production Research* 46:14, pp. 3859-3874

- Tsou, C.-S., Kao, C.-H. (2006) An Electromagnetism-Like Meta-Heuristic for Multi-Objective Optimization. Proceedings of 2006 IEEE Congress on Evolutionary Computation, pp. 1172-1178
- Tsou, C.-S., Hsu, C.-H., Yu, F.-J. (2008) Using multi-objective electromagnetism-like optimization to analyze inventory tradeoffs under probabilistic demand. Journal of Scientific & Industrial Research, vol. 67, pp. 569-573
- Tsoukalas, A., Wiesemann, W., Rustem, B. (2009) Global optimisation of pessimistic bi-level problems. Lectures on Global Optimization 55, 215-243
- Wang, Y.-N., Wu, L.-H., Yuan, X.-F. (2010) Multi-objective self-adaptive differential evolution with elitist archive and crowding entropy-based diversity measure. Soft Computing, vol. 14, pp. 193–209. Springer
- Yang, P., Tang, G., Nehorai, A. (2012) A game-theoretic approach for optimal time-of-use electricity pricing. IEEE Transactions on Power Systems 8(2), pp. 884-892
- Zhang, T., Hu, T., Zheng, Y., Guo, X. (2012) An Improved Particle Swarm Optimization for Solving Bilevel Multiobjective Programming Problem. Journal of Applied Mathematics, vol.2012, Article ID 626717, 13 pages
- Zhang, T., Hu, T., Guo, X., Chen, Z., Zheng, Y. (2013) Solving high dimensional bilevel multiobjective programming problem using a hybrid particle swarm optimization algorithm with crossover operator. Knowledge-Based Systems 53, pp. 13-19
- Zhang, G., Zhang, G., Gao, Y., Lu, J. (2011) Competitive Strategic Bidding Optimization in Electricity Markets Using Bilevel Programming and Swarm Technique. IEEE Transactions on Industrial Electronics 58, pp. 2138-2146
- Zheng, Y., Wan, Z. (2011) A solution method for semivectorial bilevel programming problem via penalty method. Journal of Applied Mathematics and Computing 37, 1-2, pp. 207-219
- Zitzler, E., Deb, K., Thiele, L. (2000) Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. Evolutionary Computation 8, pp. 173-195
- Zitzler, E., Thiele, L. (1999) Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. IEEE Transactions on Evolutionary Computation, 3(4), pp.257–271

Zugno, M., Morales, J.M., Pinson, P., Madsen, H. (2013) A bilevel model for electricity retailers' participation in a demand response market environment. *Energy Economics* 36, pp. 182-197

Apêndice

Apresentamos neste apêndice a descrição dos problemas multiobjetivo *benchmark* utilizados nas experiências computacionais relativas aos algoritmos de otimização multiobjetivo implementados no capítulo 2.

Começamos por descrever na Tabela A1 os problemas com duas funções objetivo, pertencentes ao conjunto ZDT, propostos por Zitzler et al. (2000), explicitando para cada problema as funções objetivo, a dimensão (número de variáveis) e a caracterização da frente ótima de Pareto quanto à respectiva convexidade. Estes problemas são obtidos considerando a seguinte estrutura:

$$\begin{aligned} \min F(x) &= (f_1(x), f_2(x)) \\ \text{s. a } f_2(x) &= g(x_2, \dots, x_m)h(f_1(x_1), g(x_2, \dots, x_m)) \\ \text{onde } x &= (x_1, \dots, x_m) \end{aligned}$$

A Tabela A2 contém os problemas com três funções objetivo pertencentes ao conjunto DTLZ (Deb et al., 2005) utilizados no capítulo 2. Estes problemas têm dimensão ajustável dentro de um determinado intervalo. Descrevemos estes problemas identificando as funções objetivo, a dimensão considerada, o domínio das variáveis e a classificação da frente ótima de Pareto.

Na Tabela A3 são descritos os problemas multiobjetivo com restrições CONSTR, SRN e TNK (Deb et al., 2002).

Tabela A1.

Problema	Funções objetivo	Dim(n)	Domínio	caracterização
ZDT1	$f_1(x) = x_1$ $f_2(x) = g(x) \left[1 - \sqrt{\frac{x_1}{g(x)}} \right]$ $g(x) = 1 + 9 \left(\sum_{i=2}^n x_i \right) / (n-1)$	30	[0,1]	Convexa
ZDT2	$f_1(x) = x_1$ $f_2(x) = g(x) \left[1 - (x_1 / g(x))^2 \right]$ $g(x) = 1 + 9 \left(\sum_{i=2}^n x_i \right) / (n-1)$	30	[0,1]	Não convexa
ZDT3	$f_1(x) = x_1$ $f_2 = g(x) \left[1 - \left(\sqrt{x_1 / g(x)} - \frac{x_1}{g(x)} \sin(10\pi x_1) \right) \right]$ $g(x) = 1 + 9 \left(\sum_{i=2}^n x_i \right) / (n-1)$	30	[0,1]	Convexa, desconexa
ZDT4	$f_1(x) = x_1$ $f_2(x) = g(x) \left[1 - \sqrt{\frac{x_1}{g(x)}} \right]$ $g(x) = 1 + 10(n-1) + \sum_{i=2}^n x_i^2 - 10 \cos(4\pi x_1)$	10	$x_1 \in [0,1]$ $x_i \in [-5,5]$ $i = 2, \dots, n$	Não convexa, multimodal (várias frentes)
ZDT6	$f_1(x) = 1 - \exp(-4\pi x_1) \sin^6(6\pi x_1)$ $f_2(x) = g(x) \left[1 - (f_1(x) / g(x))^2 \right]$ $g(x) = 1 + \left[9 \left(\sum_{i=2}^n x_i \right) / (n-1) \right]^{0.25}$	10	[0,1]	Não convexa, não uniformemente espaçada

Tabela A2.

Problema	Funções objectivo	Dim(n)	Domínio	Frente não dominada
DTLZ1	$f_1(x) = \frac{1}{2}x_1x_2(1 + g(x_3))$ $f_2(x) = \frac{1}{2}x_1(1 - x_2)(1 + g(x_3))$ $f_3(x) = \frac{1}{2}(1 - x_1)(1 + g(x_3))$ $g(x) = 100 \left[5 + \sum_{i=1}^5 (x_{3i} - 0.5)^2 - \cos(20\pi(x_{3i} - 0.5)) \right]$	7	[0,1]	Múltiplas frentes locais; frente global sobre um hiper-plano
DTLZ2	$f_1(x) = (1 + g(x_3)) \cos\left(x_1 \frac{\pi}{2}\right) \cos\left(x_2 \frac{\pi}{2}\right)$ $f_2(x) = (1 + g(x_3)) \cos\left(x_1 \frac{\pi}{2}\right) \sin\left(x_2 \frac{\pi}{2}\right)$ $f_3(x) = (1 + g(x_3)) \sin\left(x_1 \frac{\pi}{2}\right)$ $g(x) = \sum_{i=1}^{10} (x_{3i} - 0.5)^2$	12	[0,1]	Frente global no 1º octante, sobre a superfície da esfera unitária
DTLZ3	$f_1(x) = (1 + g(x_3)) \cos\left(x_1 \frac{\pi}{2}\right) \cos\left(x_2 \frac{\pi}{2}\right)$ $f_2(x) = (1 + g(x_3)) \cos\left(x_1 \frac{\pi}{2}\right) \sin\left(x_2 \frac{\pi}{2}\right)$ $f_3(x) = (1 + g(x_3)) \sin\left(x_1 \frac{\pi}{2}\right)$ $g(x) = 100 \left[10 + \sum_{i=1}^{10} (x_{3i} - 0.5)^2 - \cos(20\pi(x_{3i} - 0.5)) \right]$	12	[0,1]	Múltiplas frentes locais, frente global no 1º octante, sobre a superfície da esfera unitária

Tabela A3.

Problema	Funções objectivo	Restrições	Dim(n)	Domínio
CONSTR	$f_1(x) = x_1$ $f_2(x) = (1 + x_2) / x_1$	$g_1(x) = x_2 + 9x_1 \geq 6$ $g_2(x) = -x_2 + 9x_1 \geq 1$	2	[0,1]
SRN	$f_1(x) = (x_1 - 2)^2 + (x_2 - 1)^2 + 2$ $f_2(x) = 9x_1 - (x_2 - 1)^2$	$g_1(x) = x_1^2 + x_2^2 \leq 225$ $g_2(x) = x_1 - 3x_2 \leq -10$	2	[0,1]
TNK	$f_1(x) = x_1$ $f_2(x) = x_2$	$g_1(x) = -x_1^2 - x_2^2 + 1 +$ $0.1 \cos(16 \arctan(x_1 / x_2)) \leq 0$ $g_2(x) = (x_1 - 0.5)^2 + (x_2 - 0.5)^2$	2	[0,1]