

João Carlos Heringer Moreira

**Uma Abordagem com Multi-Mochilas
Multidimensionais para o Problema de Alocação de
Ações de Redução de Perdas na Distribuição de
Energia**

Vitória - ES

Março de 2015

João Carlos Heringer Moreira

**Uma Abordagem com Multi-Mochilas
Multidimensionais para o Problema de Alocação de
Ações de Redução de Perdas na Distribuição de
Energia**

Dissertação apresentada ao Programa de Pós-Graduação em Informática da Universidade Federal do Espírito Santo como requisito parcial para a obtenção do título de Mestre em Informática.

Orientador:
Prof. Dr. Flávio Miguel Varejão

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO

Vitória - ES
Março de 2015

Dissertação de Mestrado sob o título de “*Uma Abordagem com Multi-Mochilas Multidimensionais para o Problema de Alocação de Ações de Redução de Perdas na Distribuição de Energia*”, defendida por João Carlos Heringer Moreira em 16 de Março de 2015, aprovada pela banca examinadora constituída pelos doutores:

Prof. Dr. Flávio Miguel Varejão
Universidade Federal do Espírito Santo
Orientador

Prof. Dr. Maria Claudia Silva Boeres
Universidade Federal do Espírito Santo

Prof. Dr. André Renato Sales Amaral
Universidade Federal do Espírito Santo

Prof. Dr. Alexandre Cláudio B. Delbem
Universidade de São Paulo

Resumo

Em países em desenvolvimento, perdas não-técnicas de energia causadas por fatores externos ao processo de transporte, transformação e distribuição são consideradas pelas companhias de distribuição como algumas das maiores causas de prejuízos. No Brasil, parte dessas perdas pode ser repassada ao consumidor nas tarifas, entretanto o valor máximo deste repasse é limitado pela agência reguladora, como forma de incentivar melhorias por parte das distribuidoras. Este limite é definido na forma de metas de redução de perdas não-técnicas. O problema de otimização abordado neste trabalho trata da redução de perdas não-técnicas do ponto de vista da distribuidora. Para atingir as metas estabelecidas pela agência reguladora, as distribuidoras possuem várias ações de redução de perdas, que devem ser alocadas em planos multianuais. Estes planos tentam atingir a meta estabelecida, respeitando alguns orçamentos disponíveis, e objetivando sempre obter o maior lucro possível com a alocação das ações. Este trabalho aborda o problema da definição destes planos como uma generalização do Problema da Mochila. Uma modelagem formal do problema é definida na forma de um problema de programação inteira e a dificuldade da mesma é analisada através de testes computacionais, utilizando um resolvidor genérico aplicado a uma variedade de instâncias para obter a solução exata. Duas heurísticas são então propostas, a primeira baseada em uma abordagem gulosa e a segunda na metaheurística Busca Tabu, e aplicadas ao problema. Finalmente, as técnicas são comparadas considerando a qualidade das soluções encontradas.

PALAVRAS CHAVES: Problema da Mochila, Metaheurísticas, Otimização Combinatória, Perdas não-técnicas de energia.

Abstract

In developing countries, non-technical energy losses caused by factors unrelated to the transportation, transformation and distribution processes are considered by distribution companies as some of the greatest causes of loss. In Brazil, part of those losses can be passed on to consumers as an increase in the energy bill. However, the maximum value of this increase is limited by the regulatory agency, as a way to encourage distribution companies to make improvements on their activities. This limit is defined in the form of non-technical energy loss reduction goals. The optimization problem addressed in this work treats the loss reduction from the distribution companies point of view. In order to achieve the goals established by the regulatory agency, the companies have several loss reduction actions, which must be allocated in multiyear plans. These plans try to achieve the reduction goals without exceeding predefined budgets, always aiming to obtain the highest possible profit with the actions allocation. This work approaches the problem of those plans definition as a generalization of the Knapsack Problem. A formal model is defined as an integer programming problem and the model's hardness is analysed through computational experiments, using a generic solver applied to a variety of instances to obtain the exact solution. Two heuristics are then proposed, the first one based in a greedy approach and the second on the Tabu Search metaheuristic, and applied to the problem. Finally, the techniques are compared considering the quality of the solutions.

KEYWORDS: Knapsack Problem, Metaheuristics, Combinatorial Optimization, Non-technical energy losses.

Agradecimentos

À minha família, pelo incentivo para seguir sempre em frente, pelo apoio que me foi dado e pela compreensão nos momentos de ausência.

Ao Prof. Flávio, por me permitir desenvolver este trabalho e me guiar durante o desenvolvimento, e pela atenção e incentivo nas horas ruins.

Ao Marcos Daniel pela inestimável ajuda, pelos conselhos e companheirismo.

Sumário

Lista de Tabelas

Lista de Figuras

Lista de Algoritmos

Lista de Siglas

Lista de Símbolos

1	Introdução	14
2	Caracterização do Problema	17
2.1	Definição do Problema	17
2.2	Modelagem Formal	20
3	A Abordagem como um Problema da Mochila	23
3.1	O Problema da Mochila clássico e suas variações	23
3.2	O Problema das Multi-Mochilas Multidimensionais Limitadas Parcialmente Ordenadas	27
4	Solução do Problema	33
4.1	O Algoritmo Guloso	35
4.1.1	Solução Inicial	37
4.1.2	Medida de Eficiência	37
4.2	A Busca Tabu	39

4.2.1	A Lista Tabu e seu gerenciamento	41
4.3	O Gerador de Instâncias	45
4.3.1	O Fator de Correlação entre os Itens de uma Instância	47
5	Resultados Experimentais	49
5.1	Dificuldade do Problema	50
5.2	Qualidade das Heurísticas	52
6	Conclusão e Trabalhos Futuros	56
	Referências Bibliográficas	59
	Apêndice A – Tabelas dos Resultados Experimentais	61

Lista de Tabelas

A.1	Resultados Experimentais para instâncias com forte correlação ($\alpha = 0$)	61
A.1	Resultados Experimentais para instâncias com forte correlação ($\alpha = 0$) (Continuação)	62
A.2	Resultados Experimentais para instâncias com fraca correlação ($\alpha = 0.1$)	62
A.2	Resultados Experimentais para instâncias com fraca correlação ($\alpha = 0.1$) (Continuação)	63
A.2	Resultados Experimentais para instâncias com fraca correlação ($\alpha = 0.1$) (Continuação)	64
A.3	Resultados Experimentais para instâncias sem correlação ($\alpha = 1$)	64
A.3	Resultados Experimentais para instâncias sem correlação ($\alpha = 1$) (Continuação)	65

Lista de Figuras

4.1	Transformação de uma instância do Problema da Mochila Limitada (BKP) com 3 itens em uma instância do Problema da Mochila tradicional (KP) com 13 itens.	34
4.2	Exemplo de codificação da solução de uma instância com 3 itens e M mochilas.	35
4.3	Exemplo do Método da Eliminação Reversa especializado para variáveis limitadas.	44
4.4	Exemplos de correlação entre lucro e custo dos itens de uma instância.	48
5.1	Proporção de instâncias de itens com forte correlação interrompidas ($\alpha = 0.0$).	50
5.2	Proporção de instâncias de itens com fraca correlação interrompidas ($\alpha = 0.1$).	51
5.3	Proporção de instâncias de itens sem correlação interrompidas ($\alpha = 1.0$).	51
5.4	Qualidade das soluções do BTRLP em instâncias com itens com forte correlação ($\alpha = 0.0$).	52
5.5	Qualidade das soluções do BTRLP em instâncias com itens com fraca correlação ($\alpha = 0.1$).	52
5.6	Qualidade das soluções do BTRLP em instâncias com itens sem correlação ($\alpha = 1.0$).	53
5.7	Qualidade das soluções do AGRLP em instâncias com itens com forte correlação ($\alpha = 0.0$).	54
5.8	Qualidade das soluções do AGRLP em instâncias com itens com fraca correlação ($\alpha = 0.1$).	54
5.9	Qualidade das soluções do AGRLP em instâncias com itens sem correlação ($\alpha = 1.0$).	54

Lista de Algoritmos

4.1	Algoritmo Guloso	36
4.2	Busca Tabu	40
4.3	Gerador de Instâncias	46

Lista de Siglas

ANEEL	Agência Nacional de Energia Elétrica
PMMMLPOARE	Problema de Mochilas Múltiplas Multidimensionais Limitadas e Parcialmente Ordenadas Ampliado com Restrições Extras
PMMMLPO	Problema das Multi-Mochilas Multidimensionais Limitadas Parcialmente Ordenadas
AGRLP	Algoritmo Guloso partindo da Relaxação Linear do Problema
BTRLP	Busca Tabu partindo da Relaxação Linear do Problema
RT	Reposicionamento Tarifário
RR	Receita Requerida
RV	Receita Verificada
VPL	Valor Presente Líquido
PTAS	Polynomial Time Approximation Schemes
FPTAS	Fully Polynomial Time Approximation Schemes
CAPEX	Capital Expenditure
OPEX	Operational Expenditure
MER	Método da Eliminação Reversa
LT	Lista Tabu
LTA	Lista Tabu Ativa

Lista de Símbolos

M	Número de anos ou mochilas
L	Número de recursos ou dimensões
$o_{i,l}$	Orçamento para o ano i do recurso l
g_i	Meta de redução de perdas para o ano i
r	Taxa de atualização do capital
N	Número de ações ou itens
v_j	Valor da unidade de energia recuperada pela ação j
m_j	Mercado da ação j
$u_{i,j}$	Mercado anual da ação j para o ano i
$c_{j,l}$	Custo da ação j no recurso l
$e_{j,k}$	Energia recuperada pela ação j no k -ésimo ano após ser executada
D_j	Conjunto de duplas $(d, Q_{j,d})$ que representa as dependências da ação j
$Q_{j,d}$	Quantidade de vezes que a ação d deve ser executada antes de cada execução da ação j
\bar{x}	Uma possível solução do problema de otimização
$x_{j,i}$	A quantidade de vezes que a ação (item) j está sendo executada no ano (mochila) i em uma solução do problema de otimização
$R_{i,j,k}$	Energia recuperada no ano i do planejamento pelas execuções da ação j no ano k do planejamento
$V_i(\bar{x})$	Lucro total no ano i do planejamento
$C_i(\bar{x})$	Custo total no ano i do planejamento
p_j	Lucro do item j
w_j	Peso do item j
x_j	Quantidade de itens j presentes na mochila

b_j	Quantidade máxima de itens do tipo j que podem estar na mochila
c	Capacidade da mochila
$w_{j,r}$	Peso do item j na dimensão (recurso) r da mochila
c_r	Capacidade da dimensão (recurso) r da mochila
R	Número de dimensões ou recursos de uma mochila
c_i	Capacidade da mochila i em problemas da mochila múltipla
A	Conjunto de duplas (i, j) que representa dependências entre ações i e j em problemas da mochila parcialmente ordenada
$c_{i,r}$	Capacidade da dimensão (recurso) r na mochila i
$b_{j,i}$	Quantidade máxima de itens do tipo j na mochila i
mov	Movimento feito em uma solução nas heurísticas
$m\bar{ov}$	Movimento inverso ao movimento mov feito em uma solução nas heurísticas
$x(i)$	Solução da iteração i da heurística
d_r	Valor da variável dual associada a uma restrição

1 Introdução

Os principais objetivos de uma empresa distribuidora de energia são entregar energia aos consumidores com qualidade e com baixos custos e remunerar adequadamente o capital dos investidores. Com o objetivo de garantir condições favoráveis para que o mercado de energia elétrica se desenvolva com equilíbrio entre os agentes e beneficiando a sociedade, foi criada em 1996 a agência reguladora chamada de Agência Nacional de Energia Elétrica (ANEEL).

Entre suas atribuições, a ANEEL deve estimular a redução da energia perdida durante a distribuição. Esta perda de energia pode ser classificada em duas categorias: as perdas técnicas, causadas por fenômenos físicos inerentes ao processo de distribuição de energia, e as perdas não-técnicas, causadas por outros fatores externos como furtos ou falhas na medição do consumo de energia. Um dos mecanismos da ANEEL para estimular a redução de perdas é a definição de metas de perda de energia, que indicam a quantidade máxima de energia perdida pela distribuidora por causas não-técnicas que pode ser repassada aos consumidores na forma de aumento nas tarifas. As distribuidoras, por sua vez, fazem seus planejamentos de forma a tentar atingir esta meta, mas sempre tentando obter o máximo retorno financeiro com as ações tomadas para reduzir a perda de energia.

Segundo o balanço de 2011 da companhia distribuidora local [ABREU ET AL. 2011], aproximadamente 40 milhões de reais foram utilizados em ações de redução de perdas não-técnicas naquele ano. Hoje estas ações são escolhidas a mão pelos especialistas, visando, por tentativa e erro, escolher a combinação de ações que obtenha o maior lucro para a distribuidora. Dado o alto valor deste orçamento e a quantidade de ações disponível para o especialista, é interessante uma abordagem mais sistemática na escolha destas ações, de forma a facilitar a tomada de decisão do especialista e, conseqüentemente, aumentar o lucro da empresa distribuidora.

Neste contexto, esse trabalho descreve a aplicação de alguns métodos de otimização a um problema real: o da escolha de um conjunto de ações de redução de perdas não-técnicas de energia elétrica em um portfólio pré-definido, visando maximizar o retorno do capital empregado na execução das ações e se sujeitando a algumas restrições. Posto isto, pode-se listar os principais

objetivos deste trabalho como:

1. Entender melhor o problema da escolha de ações de redução de perdas não-técnicas, suas características e restrições, para poder modelá-lo formalmente visando a aplicação de algoritmos de otimização;
2. Testar a dificuldade das instâncias do modelo desenvolvido;
3. Resolver o modelo definido, escolhendo uma combinação de ações que maximize o lucro obtido pela empresa distribuidora com a execução das mesmas.

Para atingir estes objetivos, uma modelagem matemática é definida a partir da apresentação do domínio do problema da distribuidora. Esta modelagem é abordada neste trabalho como uma generalização do Problema da Mochila [PISINGER 1995], que passa a ser chamada de PMMMLPOARE (Problema de Mochilas Múltiplas Multidimensionais Limitadas e Parcialmente Ordenadas Ampliado com Restrições Extras). Uma outra generalização do Problema da Mochila, nomeada de PMMMLPO (Problema das Multi-Mochilas Multidimensionais Limitadas Parcialmente Ordenadas), é então construída a partir de algumas outras generalizações do problema tradicional conhecidas na literatura: Mochila Limitada, Mochila Multidimensional, Mochilas Múltiplas e Mochila Parcialmente Ordenada [KELLERER ET AL. 2004]. É mostrado que o PMMMLPOARE é uma generalização do PMMMLPO, e as dificuldades dos dois são relacionadas.

Além disso, com o objetivo de resolver o PMMMLPOARE, uma vez que encontrar a solução exata do problema pode ser inviável computacionalmente, duas heurísticas são desenvolvidas: o AGRLP (Algoritmo Guloso partindo da Relaxação Linear do Problema), uma heurística baseada em uma abordagem gulosa e o BTRLP (Busca Tabu partindo da Relaxação Linear do Problema), uma heurística baseada na metaheurística Busca Tabu. Uma vez que a distribuidora local não foi capaz de fornecer um número suficiente de instâncias para executar os testes necessários, um gerador de instâncias artificiais do problema teve de ser desenvolvido.

Finalmente, um conjunto de testes computacionais é feito. Primeiramente, um conjunto de instâncias do PMMMLPOARE com as dimensões que se espera encontrar na prática é criado utilizando o gerador de instâncias. Em seguida, essas instâncias são resolvidas utilizando o resolvidor genérico CPLEX, para testar a dificuldade das mesmas e obter algumas soluções para comparação. Por fim, as mesmas instâncias são resolvidas utilizando as duas heurísticas, e os resultados obtidos são comparados com o resolvidor genérico e entre elas.

As principais contribuições deste trabalho podem ser então resumidas em:

- Definição e modelagem formal do problema de alocação de ações de redução de perdas não-técnicas de energia, com a criação de duas novas variações do problema da mochila tradicional (PMMMLPO e PMMMLPOARE);
- Criação de um gerador de instâncias aleatórias do problema, que pode ser utilizado para criar instâncias artificiais com dimensões parecidas com as dimensões das instâncias que se espera encontrar na prática;
- Criação de duas novas heurísticas para o problema, a primeira baseada em uma abordagem Gulosa e a segunda em uma Busca Tabu;
- Um estudo empírico da dificuldade das instâncias do problema criadas com o gerador de instâncias aleatório, utilizando o resolvidor genérico CPLEX;
- Uma comparação entre as soluções encontradas pelo AGRLP, BTRLP e CPLEX, em relação à qualidade das soluções.

O restante desta dissertação está dividido em 5 capítulos:

- O Capítulo 2 apresenta melhor o domínio do problema e a descrição do mesmo a partir do ponto de vista da empresa distribuidora de energia. Partindo desta definição, é mostrada a modelagem matemática proposta.
- No Capítulo 3 várias generalizações conhecidas do Problema da Mochila são descritas, sendo posteriormente utilizadas para criar uma única generalização. Esta generalização é então comparada à modelagem do problema, mostrando suas similaridades e diferenças e a relação entre a dificuldade das duas.
- O Capítulo 4 apresenta as duas heurísticas desenvolvidas para resolver o problema: o AGRLP e o BTRLP. São apresentados pseudocódigos das duas e discutidos os detalhes específicos da implementação de cada uma delas. Além disso, é apresentado o gerador de instâncias aleatórias criado para gerar instâncias do problema e suas motivações.
- O Capítulo 5 dedica-se a apresentar o resultados computacionais obtidos com os experimentos. São mostrados neste capítulo primeiramente os testes feitos com o resolvidor genérico, seguidos dos resultados obtidos com as duas heurísticas e acompanhados das análises realizadas com estes experimentos computacionais.
- O Capítulo 6 fecha o trabalho apresentando as conclusões obtidas e alguns possíveis trabalhos futuros.

2 Caracterização do Problema

Este capítulo apresenta a definição do problema a ser resolvido a partir do ponto de vista da empresa distribuidora. Primeiramente, o domínio do problema é apresentado na seção 2.1, e uma definição informal do problema é feita. A maioria das informações disponíveis nesta seção foram obtidas no site da ANEEL [Anl] e em conversas com os especialistas. Em seguida, na seção 2.2 parte-se dessa definição informal para o desenvolvimento do modelo matemático, que será utilizado no restante deste trabalho.

2.1 Definição do Problema

Um dos objetivos da Agência Nacional de Energia Elétrica (ANEEL) é garantir que os serviços de distribuição de energia sejam fornecidos ao consumidor com continuidade, qualidade e preços justos, garantindo que a receita obtida pela distribuidora seja suficiente para cobrir os custos operacionais e de manutenção além de remunerar adequadamente o capital investido. Para garantir preços justos para o consumidor, bem como o equilíbrio econômico-financeiro da distribuidora, a ANEEL é responsável por estabelecer as tarifas finais ao consumidor e os mecanismos de reajuste dessas tarifas.

Entre os mecanismos de reajuste das tarifas definidos pela ANEEL se encontram as *Revisões Tarifárias Periódicas*. Essas revisões são efetuadas após um período definido no contrato de concessão (3 anos no caso da distribuidora local), com o objetivo de analisar o equilíbrio da concessão. Durante esta revisão, é feito, entre outros procedimentos, o cálculo do *Reposicionamento Tarifário (RT)*, ou seja, o índice de reajuste na tarifa a ser aplicado naquele período (após correções).

O RT tem como objetivo o reequilíbrio econômico-financeiro da concessão, através do reajuste (para mais ou menos) em relação ao último RT. O cálculo desse índice considera a razão entre a *Receita Requerida (RR)*, que corresponde à receita que a concessionária precisaria obter para cobrir os custos operacionais e remunerar os investimentos, e a *Receita Verificada*

(RV), que é a receita obtida pela concessionária com o fornecimento da energia. É no cálculo da Receita Requerida que os custos da aquisição de energia junto às empresas geradoras são repassados ao consumidor.

Vale citar que, pela própria natureza do processo de transmissão e distribuição de energia, nem toda a energia comprada será efetivamente entregue ao consumidor, pois ocorrerão *perdas* no caminho. Essas perdas podem ser divididas em *perdas técnicas*, que estão intimamente ligadas aos processos de transporte, transformação e distribuição de energia, e *perdas não-técnicas*, que representam as demais perdas, como furtos de energia, consumidores sem medidores e falhas na medição ou no faturamento. A existência dessas perdas significa que, para vender uma quantidade x de eletricidade, a empresa distribuidora precisa comprar das geradoras $x + y$, onde y representa a parcela de eletricidade perdida.

A princípio, as perdas de eletricidade por causas não-técnicas poderiam ser repassadas aos consumidores. Entretanto, como forma de estimular a melhoria na qualidade dos serviços, a ANEEL não permite que esse repasse seja feito indiscriminadamente. Baseando-se em estudos de outras concessionárias de porte parecido, ela estabelece um limite considerado aceitável para as perdas-não técnicas, que é conhecido como *Meta de Perdas Não-Técnicas*. Para exemplificar, suponha uma empresa distribuidora com um nível de perdas não-técnicas de 15%. Para conseguir vender 85GW de eletricidade, esta empresa precisa comprar 100 GW. Se a ANEEL estabelece uma meta de perdas hipotética de 10%, a empresa fica autorizada a repassar o valor de 10 dos 15 GW perdidos aos consumidores, e deve mitigar os outros 5 GW perdidos, ou assumir o prejuízo.

Na prática, este cenário incentiva as empresas distribuidoras a combater a perda da energia acima da meta, pois como a perda não pode ser repassada ao consumidor em forma de aumento nas tarifas, é melhor que a eletricidade seja efetivamente vendida, e não desperdiçada. No entanto, não vale a pena para a distribuidora mitigar essas perdas além da meta, já que isso significaria uma redução na Receita Requerida e um aumento na Receita Verificada, e consequentemente uma redução no Reposicionamento Tarifário.

Geralmente, essa meta é definida durante a Revisão Tarifária Anual, e a partir dela é derivada uma *Curva de Redução de Perdas Não-Técnicas*, que estabelece metas anuais de redução de perdas para o período coberto pela revisão, com o objetivo de, ao final do período, atingir a meta estabelecida. Isso possibilita que a distribuidora, baseada na curva definida pela ANEEL, possa definir um *Plano de Redução de Perdas Não-Técnicas*. Este plano consiste na curva anual de redução de perdas, definida anteriormente, um portfólio de *Ações de Redução de Perdas Não-Técnicas*, orçamentos anuais para diferentes recursos financeiros, como por exemplo,

orçamentos para compra de peças e orçamentos para pagamento de pessoal, que são utilizados para a execução das ações e uma *taxa de atualização do capital*.

Com a definição deste plano, o problema a ser resolvido pela distribuidora pode então ser resumido em: *Como distribuir adequadamente as ações de redução de perdas não-técnicas durante o período desejado, respeitando algumas restrições, sem ultrapassar as metas de redução anuais nem os orçamentos definidos e maximizando o retorno financeiro das ações executadas?*

Uma grande variedade de ações compõe o portfólio citado anteriormente, como a troca de medidores defeituosos, instalação de medidores, troca de transformadores e doação de geladeiras mais eficientes, por exemplo. Cada uma dessas ações possui características únicas, que devem ser levadas em conta no momento do planejamento. Uma delas é o *mercado* da ação, que indica a quantidade máxima de vezes que a ação pode ser executada durante todo o período do planejamento. Algumas ações, como a troca de medidores, são geralmente mais fáceis de ser executadas, e podem, portanto, ser executadas mais vezes durante o planejamento. Entretanto, nenhuma ação pode ser executada infinitamente, todas estão limitadas pelos seus mercados. Outra característica das ações é o *mercado anual*. Análoga ao *mercado* das ações, essa característica define quantas vezes determinada ação pode ser tomada em cada ano. Essa restrição impede que o mercado de alguma ação seja executado totalmente em um ano apenas, e pode ser utilizada para modelar, por exemplo, limitações no fornecimento de peças.

Outros aspectos importantes de cada ação são o custo da ação, a quantidade de energia recuperada, o valor da energia recuperada e a dependência entre ações. Primeiramente, cada ação consome porções diferentes de cada orçamento, esse é o *custo* da ação. Além disso, apesar de consumir os orçamentos somente no ano em que foram executadas, a *energia recuperada* pela ação pode estar dividida em vários anos. Por exemplo, uma ação *a* executada no ano *y*, consome uma porção dos orçamentos do ano *y*, mas recupera energia no ano *y* e em todos os anos seguintes, impactando, portanto, na meta de todos os anos seguintes ao que foi tomada. Outro ponto importante é que cada ação é tomada em um tipo de rede específica, como redes de alta ou baixa tensão, que pode ter um valor da unidade de energia recuperada diferente. Por isso, para que seja possível calcular o lucro de cada ação, deve-se ser definido também o *valor da energia* recuperada para cada ação. Finalmente, algumas ações podem requerer que outras ações sejam executadas antes delas. Um possível exemplo seria a necessidade de instalar novos postes ou equipamentos para poder instalar novos medidores. Essa situação requer que para cada ação seja definida uma relação de *dependência*, que indica de quais ações ela depende.

2.2 Modelagem Formal

Para que seja possível aplicar ao problema métodos de otimização, deve-se primeiramente defini-lo formalmente. Nesta modelagem, considera-se que o único objetivo é maximizar o *valor presente líquido (VPL)*, ou seja, maximizar o retorno financeiro dos investimentos feitos, considerando uma taxa de atualização do capital, para um plano de M anos, dados:

- o orçamento anual $o_{i,l}$, para um conjunto de L recursos, $1 \leq i \leq M$, $1 \leq l \leq L$;
- a meta de redução anual g_i , $1 \leq i \leq M$, que representa a perda de energia que deve ser reduzida no ano i ;
- a taxa de atualização do capital r , que representa a desvalorização anual do investimento. Essa taxa é constante para todos os anos e investimentos.

O investimento a ser feito consiste em escolher um subconjunto de ações em um portfólio de tamanho N . Cada uma das j ações deste portfólio, $1 \leq j \leq N$, possui algumas características:

- o *valor da eletricidade* v_j , que representa o valor da unidade de eletricidade para cada ação j do portfólio, ou seja, o valor do kilowatt-hora;
- m_j , o número máximo de vezes que a ação j pode ser executada durante todo o plano, também conhecido como o *mercado* da ação;
- $u_{j,i}$, o número máximo de vezes que a ação j pode ser executada no ano i , ou seja, o *mercado anual* da ação;
- $c_{j,l}$, quanto do recurso l cada execução da ação j consome, o *custo* da ação;
- $e_{j,k}$, a *energia recuperada* pela ação j no k -ésimo ano após ser executada;
- um conjunto D_j de pares $(d, Q_{j,d})$ representando as *dependências* da ação j . Para cada execução da ação j , cada ação $d \in D_j$ deve ser executada previamente um número de vezes definido por $Q_{j,d} \in \mathbb{R}^+$.

O objetivo é encontrar uma solução \bar{x} , ou seja, um conjunto de valores para as variáveis $x_{j,i}$, $\forall i, j$, $x_{j,i} \in \mathbb{N}$, que maximize o VPL. Esta solução representa o número de vezes que a ação j será executada no i -ésimo ano do planejamento. Para introduzir a equação do VPL e a função objetivo do problema, três equações auxiliares devem ser definidas:

- A primeira delas, a equação 2.1, representa a redução de perdas de energia no i -ésimo ano pela ação j tomada no k -ésimo ano do planejamento. Em outras palavras, esta equação representa quanto as execuções da ação j no ano k do planejamento evitarão de perdas de energia em determinado ano i do planejamento.

$$R_{i,j,k}(\bar{x}) = x_{j,k} \cdot e_{j,i-k+1} \quad (2.1)$$

- Já a equação 2.2 representa o lucro total anual V_i , que é a soma de toda energia recuperada no ano i multiplicada pelo valor da energia:

$$V_i(\bar{x}) = \sum_{j=1}^N \sum_{k=1}^i R_{i,j,k}(\bar{x}) \cdot v_j, \quad (2.2)$$

- Finalmente, a equação 2.3 representa o custo total anual C_i , ou seja, qual o custo de todas as ações tomadas no i -ésimo ano:

$$C_i(\bar{x}) = \sum_{j=1}^N \sum_{l=1}^L x_{j,i} \cdot c_{j,l} \quad (2.3)$$

Por definição, $V_i - C_i$ é o fluxo de caixa total no i -ésimo ano e o VPL é a soma de todos os fluxos de caixa anuais, ajustados pela taxa de atualização do capital definida no início desta seção, para todos os anos. Quanto maior o VPL, mais a distribuidora está lucrando com a execução das ações. Assim, o objetivo neste problema é maximizar o VPL, e a função objetivo é representada na equação 2.4:

$$\max_{\bar{x}}(O(\bar{x})) = \max_{\bar{x}} \left(\sum_{i=1}^M \frac{V_i(\bar{x}) - C_i(\bar{x})}{(1+r)^i} \right) \quad (2.4)$$

A função objetivo apresentada na equação 2.4 deve ser maximizada respeitando as restrições do problema. São elas:

- A restrição anual de orçamento (equação 2.5), que evita que o custo da solução ultrapasse os orçamentos anuais para cada recurso do planejamento,

$$\sum_{j=1}^N x_{j,i} \cdot c_{j,l} \leq o_{i,l} \quad \forall i, l, \quad (2.5)$$

- a restrição de mercado (equação 2.6), que evita que a solução encontrada ultrapasse o

limite global do número de alocações de cada ação,

$$\sum_{i=1}^M x_{j,i} \leq m_j \quad \forall j, \quad (2.6)$$

- a restrição de mercado anual (equação 2.7), análoga à restrição de mercado, mas evita ultrapassar o limite anual de alocações de cada ação,

$$x_{j,i} \leq u_{j,i} \quad \forall j, i, \quad (2.7)$$

- a restrição anual de meta de redução (equação 2.8), utilizada para garantir que as metas de redução de perdas não serão ultrapassadas,

$$\sum_{j=1}^N \sum_{k=1}^i R_{i,j,k}(\bar{x}) \leq g_i \quad \forall i, \quad (2.8)$$

- a restrição de dependência entre ações (equação 2.9), que garante que as relações de dependência serão respeitadas para todos os anos e ações do planejamento

$$\forall j, k \quad \sum_{i=1}^k x_{d,i} \geq \sum_{i'=1}^k x_{j,i'} \cdot Q_{j,d} \quad \forall d \in D_j. \quad (2.9)$$

3 A Abordagem como um Problema da Mochila

Neste trabalho, aborda-se o modelo definido no capítulo 2 como uma generalização do *Problema da Mochila*, que é muito conhecido na literatura. A formulação exata do modelo definido, entretanto, se mostrou muito peculiar, e não foi encontrado nenhum trabalho abordando um modelo que englobasse tantas variações do Problema da Mochila quanto este.

As seções seguintes descrevem o Problema da Mochila clássico e algumas de suas generalizações mais conhecidas. Posteriormente, é construída uma nova generalização do problema constituída das características de cada uma dessas generalizações e são mostradas as semelhanças e diferenças entre essa nova generalização e o modelo definido no capítulo anterior. Finalmente, a dificuldade do modelo proposto anteriormente é justificada considerando as dificuldades de todas as generalizações relacionadas no decorrer do capítulo.

3.1 O Problema da Mochila clássico e suas variações

O *Problema da Mochila* tradicional, também conhecido como *Problema da Mochila 0-1*, é um dos mais conhecidos da área de Otimização Combinatória. Ele pode ser formalmente definido como: Dada um mochila de capacidade c e um conjunto de itens N , contendo n itens j , cada um deles com um lucro p_j e um peso w_j associados, qual é a melhor combinação possível desses itens, visando maximizar o lucro total sem exceder a capacidade da mochila?

Esta pergunta pode ser formulada como a solução do seguinte problema de Programação Linear Inteira [PISINGER 1995]:

$$\begin{aligned}
&\text{maximizar} && \sum_{j=1}^N p_j \cdot x_j \\
&\text{sujeito a} && \sum_{j=1}^N w_j \cdot x_j \leq c, \\
&&& x_j \in \{0, 1\}, \quad j = 1, \dots, N.
\end{aligned} \tag{3.1}$$

Uma das possíveis generalizações do problema clássico é chamada de *Problema da Mochila Limitada*. Nesta variação, um item j pode ser colocado na mochila até b_j vezes. Segue abaixo a formulação desta variação:

$$\begin{aligned}
&\text{maximizar} && \sum_{j=1}^N p_j \cdot x_j \\
&\text{sujeito a} && \sum_{j=1}^N w_j \cdot x_j \leq c, \\
&&& x_j \leq b_j \\
&&& 0 \leq x_j, \quad x_j \in \mathbb{Z}, \quad j = 1, \dots, N.
\end{aligned} \tag{3.2}$$

Boa parte dos métodos utilizados para resolver este tipo de problema se baseia na transformação da variável limitada em variáveis binárias, transformando o problema limitado em um problema 0-1 com um número maior de variáveis [KELLERER ET AL. 2004] e resolvendo o problema resultante com técnicas criadas para o Problema da Mochila 0-1 tradicional.

Outra possível generalização do problema clássico envolve mochilas com múltiplas restrições, como peso e volume, por exemplo. Neste problema, a mochila possui R dimensões, cada uma delas com uma capacidade c_r diferente, e conseqüentemente, a alocação de um item na mochila agora consome w_r recursos de cada dimensão. Esta nova generalização, que possui diversas aplicações práticas, é conhecida como *Problema da Mochila Multidimensional*, ou *Problema da Mochila d -Dimensional*, e pode ser formulada como [PISINGER 1995]:

$$\begin{aligned}
&\text{maximizar} && \sum_{j=1}^N p_j \cdot x_j \\
&\text{sujeito a} && \sum_{j=1}^N w_{j,r} \cdot x_j \leq c_r, \quad r = 1, \dots, R, \\
&&& x_j \in \{0, 1\}, \quad j = 1, \dots, N.
\end{aligned} \tag{3.3}$$

Devido a dificuldade computacional observada na prática esta variação é muito utilizada também como forma de comparação entre diversas implementações de metaheurísticas e, por

isso, é a variação do problema tradicional com a maior literatura disponível. Na década de 90, várias metaheurísticas foram aplicadas com sucesso a este problema, como as Buscas Tabu de [DAMMEYER , VOSS 1993, HANAFI , FREVILLE 2001] e o Algoritmo Genético de [CHU , BEASLEY 1998]. Este último, além de ser uma das primeiras aplicações de Algoritmos Genéticos ao problema, contribuiu com o estabelecimento de um conjunto de instâncias difíceis, conhecido como “OR-Library” que até hoje é muito utilizado como padrão em testes envolvendo o problema da Mochila Multidimensional. Posteriormente, [VASQUEZ , VIMONT 2005] apresentam uma Busca Tabu com elementos de Programação Linear que conseguiu os melhores resultados conhecidos para as instâncias de testes comumente usadas. Uma análise dos métodos exatos e heurísticos disponíveis até então é apresentada em [FREVILLE 2004]. Outros dois trabalhos recentes sobre esta variação do problema são apresentados em [BOUSSIER ET AL. 2010] e [DELLA CROCE , GROSSO 2012]. O primeiro apresenta um algoritmo exato combinando uma Busca por Resolução, um Branch & Bound e uma Busca em Profundidade, explorando características especiais do problema. O algoritmo resultante é mais rápido que as abordagens exatas existentes até então e conseguiu encontrar melhores resultados em várias das instâncias da OR-Library. O segundo trabalho apresenta uma heurística baseada em conceitos de Programação Linear, com a vantagem de ser altamente paralelizável e apresenta os melhores resultados conhecidos na literatura para as instâncias da OR-Library.

Vários outros trabalhos recentes testam metaheurísticas com o Problema da Mochila Multidimensional. Em [AL-SHIHABI , ÓLAFSSON 2010], um algoritmo híbrido utilizando Colônia de Formigas, Partições Aninhadas e Programação Linear é apresentado, e produz resultados próximos aos melhores conhecidos. [BEHESHTI ET AL. 2013] e [CHIH ET AL. 2014] apresentam abordagens baseadas em Otimização por Enxame de Partículas Binárias. O primeiro utiliza um Enxame de Partículas Binárias Aceleradas e o segundo um Enxame de Partículas Binárias com aceleração variando com o tempo. Ambos reportam resultados melhores que outras implementações da mesma metaheurística aplicadas ao problema da mochila multidimensional, mas piores que os melhores resultados obtidos até então para as instâncias da OR-Library. Em [WANG ET AL. 2013], o Algoritmo da Mosca da Fruta é adaptado para variáveis binárias, e a implementação resultante é testada utilizando as instâncias da OR-Library. Em [MARTINS ET AL. 2014], a eficiência da aplicação de “linkage-learning” a este problema é investigada, concluindo que a aplicação do mesmo não levou a resultados melhores para o problema.

Considerando agora que ao invés de várias dimensões na mesma mochila, existam M mochilas com capacidades c_i distintas e cada item j pode ser alocado em uma das mochilas. Esta generalização é conhecida como *Problema das Mochilas Múltiplas* e pode ser formulada da seguinte maneira [KELLERER ET AL. 2004]:

$$\begin{aligned}
& \text{maximizar} && \sum_{i=1}^M \sum_{j=1}^N p_j \cdot x_{j,i} \\
& \text{sujeito a} && \sum_{j=1}^N w_j \cdot x_{ij} \leq c_i, && i = 1, \dots, M, \\
& && \sum_{i=1}^M x_{j,i} \leq 1, && j = 1, \dots, N, \\
& && x_{j,i} \in \{0, 1\}, && i = 1, \dots, M, \quad j = 1, \dots, N.
\end{aligned} \tag{3.4}$$

Em [AMARANTE ET AL. 2013], esta generalização do problema da mochila é utilizada para modelar um problema de alocação de máquinas virtuais em computação em nuvem. O modelo é resolvido utilizando a metaheurística Colônia de Formigas e obtém, para certas classes de instâncias, resultados melhores que outras abordagens já aplicadas ao problema. Outro trabalho recente envolvendo essa generalização aparece em [PATVARDHAN ET AL. 2014], no qual o problema das mochilas múltiplas é resolvido utilizando um Algoritmo Evolucionário de Inspiração Quântica.

Outra possível generalização do problema tradicional que envolve um ordenamento parcial entre os itens é chamada de *Problema da Mochila Parcialmente Ordenada*. Este ordenamento entre os itens implica em dizer que, se o item i precede o item j , para que seja possível colocar o item j na mochila, o item i também deve estar na mochila. Esta generalização pode ser formulada da seguinte forma [KELLERER ET AL. 2004]:

$$\begin{aligned}
& \text{maximizar} && \sum_{j=1}^N p_j \cdot x_j \\
& \text{sujeito a} && \sum_{j=1}^N w_j \cdot x_j \leq c \\
& && x_d \geq x_j, && (j, d) \in A \\
& && x_j \in \{0, 1\}, && j = 1, \dots, N
\end{aligned} \tag{3.5}$$

Neste caso, as precedências são modeladas como um conjunto A de tuplas (i, j) , e cada uma dessas tuplas representa uma relação do tipo "o item i precede o item j ".

Uma revisão sobre o *Problema da Mochila com Restrições de Vizinhaça*, que é uma generalização do Problema da Mochila Parcialmente Ordenada é feita em [BORRADAILE ET AL. 2012]. Em [LAMBERT , NEWMAN 2014], esta variação do Problema da Mochila é utilizada para modelar um problema de sequenciamento de blocos em mineração a céu aberto.

Todos os problemas mencionados nas equações 3.1 a 3.5 são difíceis de resolver otimamente. Eles pertencem à classe de problemas \mathcal{NP} -hard [KELLERER ET AL. 2004], e não são conhecidos algoritmos para resolvê-los em tempo polinomial, a não ser que $\mathcal{P} = \mathcal{NP}$ [GAREY, JOHNSON 1978]. Para o problema tradicional 0-1 e sua variação limitada, existem algoritmos pseudopolinomiais, que se comportam como algoritmos polinomiais se os coeficientes do problema forem limitados, baseados em programação dinâmica e FPTAS (*Fully Polynomial Time Approximation Schemes*), que são algoritmos utilizados para obter soluções aproximadas, com o tempo de execução crescendo polinomialmente em relação ao tamanho das instâncias e ao tamanho do erro aceitável [KELLERER ET AL. 2004]. Enquanto isso, as outras variações são difíceis de aproximar e só são conhecidos PTAS (*Polynomial Time Approximation Schemes*) para resolvê-las de forma aproximada [KOLLIPOULOS, STEINER 2007, PUCHINGER ET AL. 2006, DAWANDE ET AL. 2000], com o tempo de execução crescendo polinomialmente em relação ao tamanho da instância, porém exponencialmente em relação ao erro aceitável.

3.2 O Problema das Multi-Mochilas Multidimensionais Limitadas Parcialmente Ordenadas

Ao considerar um problema com as características de todas as variações apresentadas na seção 3.1, chega-se a uma generalização que pode ser chamada de *Problema das Multi-Mochilas Multidimensionais Limitadas Parcialmente Ordenadas* (PMMMLPO), mostrada na equação 3.6.

A modelagem do problema definido no capítulo 2 possui muitas similaridades à formulação do PMMMLPO. As ações, anos e recursos da modelagem podem ser relacionados, respectivamente, aos itens, mochilas e recursos deste problema. Quando se considera que cada ação pode ser executada u vezes, ao invés de $\{0,1\}$ vezes por ano, a modelagem pode ser descrita como um *Problema da Mochila Limitada*. Além disso, o plano de redução de perdas é definido para vários anos, e considerando cada ano como uma mochila diferente, a modelagem também se encaixa como um *Problema das Mochilas Múltiplas*. Atualmente, a distribuidora local de energia considera dois tipos de recursos, conhecidos como orçamentos CAPEX e OPEX. Cada ação tomada pode consumir uma fração de cada um desses recursos, o que significa que o custo das ações é calculado em 2 dimensões, ou seja, o modelo também se encaixa como uma generalização do *Problema da Mochila Multidimensional*. Algumas das ações definidas pela distribuidora só podem ser tomadas depois que outras forem executadas, o que se encaixa na definição do *Problema da Mochila Parcialmente Ordenada*.

$$\begin{aligned}
& \underset{\bar{x}}{\text{maximizar}} && \left(\sum_{i=1}^M \sum_{j=1}^N x_{j,i} \cdot p_j \right) \\
& \text{sujeito a} && \sum_{j=1}^N x_{j,i} \cdot w_{j,r} \leq c_{i,r} \quad \forall i, r \\
& && x_{j,i} \leq b_{j,i} \quad \forall j, i \\
& && \sum_{i=1}^k x_{d,i} \geq \sum_{i'=1}^k x_{j,i'} \quad \forall (j, d) \in A \quad \forall k \leq M \\
& \text{onde} && x_{j,i} \in N, \quad i \leq M, \quad j \leq N, \quad r \leq R
\end{aligned} \tag{3.6}$$

Apesar de todas as semelhanças descritas anteriormente, percebe-se que a modelagem do problema definido no capítulo 2 não se encaixa exatamente no PMMMLPO. Primeiramente, mesmo que uma ação consuma recursos somente do ano em que foi alocada, ela pode impactar a meta de recuperação de energia de vários anos posteriores, como definido na restrição representada pela equação 2.8. Além disso, enquanto o problema da Mochila Parcialmente Ordenada (equação 3.5) considera somente dependências simples (uma ação x deve ser executada uma vez antes da ação y), o modelo admite que uma ação possa precisar que outra seja executada mais de uma vez antes dela. Além disso, ao contrário do problema da distribuidora, o PMMMLPO não possui uma taxa de atualização do capital.

Considerando o problema da distribuidora com dependências simples, ações recuperando energia somente no ano em que foram tomadas e taxa de atualização de capital igual a zero, características que podem ser representadas no modelo definido no capítulo 2, é possível interpretá-lo como uma generalização do PMMMLPO. Assim, a partir deste ponto refere-se como *Problema de Mochilas Múltiplas Multidimensionais Limitadas e Parcialmente Ordenadas Ampliado com Restrições Extras*, ou PMMMLPOARE ao modelo definido no capítulo 2 e apresentado na equação 3.7, que é utilizado neste trabalho para representar o problema da distribuidora de energia.

$$\begin{aligned}
& \underset{\bar{x}}{\text{maximizar}} && \left(\sum_{i=1}^M \frac{V_i(\bar{x}) - C_i(\bar{x})}{(1+r)^i} \right) && \text{(VPL)} \\
& \text{sujeito a} && \sum_{j=1}^N x_{j,i} \cdot c_{j,l} \leq o_{i,l} \quad \forall i, l && \text{(Orçamentos)} \\
& && \sum_{i=1}^M x_{j,i} \leq m_j \quad \forall j && \text{(Mercado)} \\
& && x_{j,i} \leq u_{j,i} \quad \forall j, i && \text{(Mercado Anual)} \\
& && \sum_{j=1}^N \sum_{k=1}^i R_{i,j,k}(\bar{x}) \leq g_i \quad \forall i && \text{(Meta)} \\
& && \sum_{i=1}^k x_{d,i} \geq \sum_{i'=1}^k x_{j,i'} \cdot Q_{j,d} \quad \forall d \in D_j \quad \forall j, k && \text{(Dependências)} \\
& \text{onde} && R_{i,j,k}(\bar{x}) = x_{j,k} \cdot e_{j,i-k+1} && \text{(Recuperação)} \\
& && V_i(\bar{x}) = \sum_{j=1}^N \sum_{k=1}^i R_{i,j,k}(\bar{x}) \cdot v_j && \text{(Lucro Anual)} \\
& && C_i(\bar{x}) = \sum_{j=1}^N \sum_{l=1}^L x_{j,i} \cdot c_{j,l} && \text{(Custo anual)} \\
& && x_{j,i} \in N, \quad i \leq M, \quad j \leq N, \quad l \leq L
\end{aligned} \tag{3.7}$$

Com o objetivo de provar a complexidade do PMMMLPOARE, será mostrada uma redução do PMMMLPOARE para o PMMMLPO, mostrando que o segundo é um caso especial do primeiro. Primeiramente, considere uma instância do PMMMLPOARE onde uma ação só recupera energia no ano em que foi tomada (consideração 1). Fazendo essa consideração tem-se que:

$$R_{i,j,k}(\bar{x}) = \begin{cases} x_{j,k} \cdot e_{j,i-k+1} = x_{j,k} \cdot e_{j,1}, & \forall i = k \\ 0, & \forall i \neq k \end{cases} \tag{3.8}$$

A partir da equação 3.8, o lucro anual pode ser simplificado para:

$$\begin{aligned}
V_i(\bar{x}) &= \sum_{j=1}^N \sum_{k=1}^i R_{i,j,k}(\bar{x}) \cdot v_j \\
&= \sum_{j=1}^N \sum_{k=i}^i R_{i,j,k}(\bar{x}) \cdot v_j \\
&= \sum_{j=1}^N x_{j,i} \cdot e_{j,1} \cdot v_j
\end{aligned} \tag{3.9}$$

Considere agora que nessa instância, o valor da energia recuperada pela ação j no ano em que foi executada multiplicado pelo valor da energia recuperada é duas vezes a soma dos custos da mesma ação em todos os recursos (consideração 2), como na equação equação 3.10:

$$2 \cdot \sum_{l=1}^L x_{j,i} \cdot c_{j,l} = x_{j,i} \cdot e_{j,1} \cdot v_j \quad (3.10)$$

Utilizando as equações 3.9 e 3.10, é possível simplificar a equação que calcula o fluxo de caixa anual, $V_i - C_i$:

$$\begin{aligned} V_i - C_i &= \sum_{j=1}^N x_{j,i} \cdot e_{j,1} \cdot v_j - \sum_{j=1}^N \sum_{l=1}^L x_{j,i} \cdot c_{j,l} \\ &= \sum_{j=1}^N \left(x_{j,i} \cdot e_{j,1} \cdot v_j - \sum_{l=1}^L x_{j,i} \cdot c_{j,l} \right) \\ &= \sum_{j=1}^N x_{j,i} \cdot e_{j,1} \cdot v_j - \frac{x_{j,i} \cdot e_{j,1} \cdot v_j}{2} \\ &= \sum_{j=1}^N \frac{x_{j,i} \cdot e_{j,1} \cdot v_j}{2} \\ &= \sum_{j=1}^N x_{j,i} \cdot \frac{e_{j,1} \cdot v_j}{2} \end{aligned} \quad (3.11)$$

Utilizando a fórmula simplificada para o fluxo de caixa anual (equação 3.11), e considerando que nesta instância a taxa de atualização de capital r é 0 (consideração 3), a nova função objetivo pode ser calculada como:

$$\begin{aligned} \text{maximizar}_{\bar{x}} \quad & \left(\sum_{i=1}^M \frac{V_i(\bar{x}) - C_i(\bar{x})}{(1+r)^i} \right) = \\ \text{maximizar}_{\bar{x}} \quad & \left(\sum_{i=1}^M \frac{\sum_{j=1}^N x_{j,i} \cdot \frac{e_{j,1} \cdot v_j}{2}}{(1+0)^i} \right) = \\ \text{maximizar}_{\bar{x}} \quad & \left(\sum_{i=1}^M \sum_{j=1}^N x_{j,i} \cdot \frac{e_{j,1} \cdot v_j}{2} \right) \end{aligned} \quad (3.12)$$

Se nessa instância o mercado for ignorado, ou seja, não existir um limite global para a execução das ações, a fórmula do mercado pode ser ignorada (consideração 4). Considerando também que nesta instância só existem dependências simples (consideração 5), o conjunto de dependências pode ser modelado simplesmente como um conjunto de tuplas com os índices das ações que possuem as dependências. Ou seja, o conjunto de dependências D pode ser represen-

tado como um conjunto de duplas (j, d) , onde cada uma das duplas do conjunto representa uma dependência entre a ação j e a ação d (para cada execução da ação j , a ação d deve ter sido executada uma vez anteriormente). Assim, a restrição de dependência para esta instância pode ser simplificada (equação 3.13):

$$\sum_{i=1}^k x_{d,i} \geq \sum_{i'=1}^k x_{j,i'} \cdot Q_{j,d} \quad \forall d \in D_j \quad \forall j, k \quad \rightarrow \quad \sum_{i=1}^k x_{d,i} \geq \sum_{i'=1}^k x_{j,i'} \quad \forall (j, d) \in D \quad \forall k \leq M \quad (3.13)$$

Por último, utilizando a equação 3.8, a restrição de meta também pode ser simplificada (equação 3.14):

$$\sum_{j=1}^N \sum_{k=1}^i R_{i,j,k}(\bar{x}) \leq g_i \quad \forall i \quad \rightarrow \quad \sum_{j=1}^N \sum_{k=i}^i x_{j,k} \cdot e_{j,1} \leq g_i \quad \forall i \quad \rightarrow \quad \sum_{j=1}^N x_{j,i} \cdot e_{j,1} \leq g_i \quad \forall i \quad (3.14)$$

É fácil perceber que a equação para a restrição de meta simplificada, apresentada na equação 3.14, tem a mesma forma da equação da restrição de orçamento, logo, a recuperação de energia e metas anuais podem ser visualizadas como apenas mais um recurso e orçamentos respectivamente. Assim, o modelo resultante para esta instância do PMMMLPOARE, feito à partir da equação 3.7, com as simplificações propostas nas equações 3.12, 3.13 e 3.14 é apresentado na equação 3.15

$$\begin{aligned} & \underset{\bar{x}}{\text{maximizar}} && \left(\sum_{i=1}^M \sum_{j=1}^N x_{j,i} \cdot \frac{e_{j,1} \cdot v_j}{2} \right) && \text{(Objetivo)} \\ & \text{sujeito a} && \sum_{j=1}^N x_{j,i} \cdot c_{j,l} \leq o_{i,l} \quad \forall i, l && \text{(Orçamentos)} \\ & && x_{j,i} \leq u_{j,i} \quad \forall j, i && \text{(Mercado Anual)} \\ & && \sum_{i=1}^k x_{d,i} \geq \sum_{i'=1}^k x_{j,i'} \quad \forall (j, d) \in D \quad \forall k \leq M && \text{(Dependências)} \\ & \text{onde} && x_{j,i} \in N, \quad i \leq M, \quad j \leq N, \quad l \leq L && \end{aligned} \quad (3.15)$$

Fazendo a comparação entre as equações 3.6, que representa o PMMMLPO, e 3.15, que mostra uma versão do PMMMLPOARE com as considerações 1 à 5 e as consequentes simplificações (equações 3.12 a 3.14), é possível perceber que as duas são essencialmente iguais, após a substituição das seguintes variáveis:

- as variáveis $w_{j,r}$, $c_{i,r}$ e $b_{j,i}$ do PMMMLPO são, respectivamente, as variáveis $c_{j,l}$, $o_{i,l}$ e $u_{j,i}$ desta versão do PMMMLPOARE;
- a variável R do PMMMLPO é a variável L do PMMMLPOARE;
- o conjunto A do PMMMLPO é o conjunto D do PMMMLPOARE;
- a variável p_j do PMMMLPO pode ser vista como o termo $\frac{e_{j,1} \cdot v_j}{2}$ do PMMMLPOARE.

Assim, fica demonstrado que uma instância qualquer deste caso especial do PMMMLPOARE pode ser resolvida como uma instância do PMMMLPO. Ou seja, o PMMMLPO é um caso especial do PMMMLPOARE e, em outras palavras, o PMMMLPOARE é uma generalização do PMMMLPO. Como o PMMMLPOARE é uma generalização do PMMMLPO, pode-se dizer que o primeiro é pelo menos tão difícil quanto o segundo, que por sua vez é uma generalização do problema clássico e suas variações apresentadas na seção 3.1. Conclui-se então que tanto o PMMMLPOARE quanto o PMMMLPO são tão difíceis quanto o problema clássico e suas variações. Logo, não existem algoritmos capazes de resolvê-los em tempo polinomial, considerando $\mathcal{P} \neq \mathcal{NP}$.

4 Solução do Problema

Nos capítulos 2 e 3, apresentou-se a modelagem do problema real como uma generalização do problema da mochila, chamada de PMMMLPOARE, e foram apresentadas as variações do problema da mochila que o compõem. Neste capítulo, são apresentadas as duas heurísticas desenvolvidas para resolver o PMMMLPOARE: um algoritmo implementado utilizando uma estratégia Gulosa e uma implementação da Busca Tabu.

A primeira abordagem, utilizando a estratégia Gulosa, foi escolhida porque mesmo com sua simplicidade de implementação, ela consegue alcançar bons resultados em respeito à qualidade das soluções encontradas nas instâncias utilizadas nos testes, executando muito rapidamente. Já a segunda, baseada na Metaheurística Busca Tabu, foi escolhida pois ao contrário da busca Gulosa simples, ela é uma busca local que consegue escapar de ótimos locais. Além disso, abordagens baseadas em Buscas Tabu já foram utilizadas com sucesso em outros trabalhos envolvendo variações do Problema da Mochila e outros problemas de Otimização Combinatória.

A seguir é mostrada a estrutura da solução, que é uma característica compartilhada pelas duas heurísticas. Esta introdução é seguida de seções dedicadas a mostrar as duas heurísticas desenvolvidas e suas características. Ao final do capítulo, uma seção motiva e mostra um gerador de instâncias artificiais desenvolvido para gerar as instâncias utilizadas nos testes.

Antes de explicar os detalhes da implementação de cada uma das heurísticas, é útil mostrar a codificação utilizada para representar as soluções, já que ela é a mesma nas duas heurísticas. Como foi mostrado no capítulo 2, no problema abordado, ao contrário do problema da mochila tradicional, mais de uma cópia de um item pode estar presente na mochila. Essa característica leva, então, a duas formas principais para a codificação da solução.

Na primeira opção, a quantidade de cada item alocada por mochila é codificada em um formato binário, ou seja, cada uma das variáveis inteiras do problema é decomposta em $\lfloor \log(b_{ji}) \rfloor + 1$ variáveis binárias artificiais, onde b_{ji} é o limite superior da quantidade de itens j que pode ser alocada na mochila i . Os lucros e pesos de cada uma dessas novas variáveis artificiais é um múltiplo dos valores da variável original por um fator de dois. Esta decomposição é exemplifi-

cada na figura 4.1, com uma transformação de uma instância do Problema da Mochila Limitada em uma do Problema da Mochila 0-1 Tradicional.

(BKP)		(KP)				
item tipo 1: $b_1 = 7$ $\lfloor \log(7) \rfloor = 2$	lucro	p_1	$2p_1$	$4p_1$		
	peso	w_1	$2w_1$	$4w_1$		
item tipo 2: $b_2 = 16$ $\lfloor \log(16) \rfloor = 4$	lucro	p_2	$2p_2$	$4p_2$	$8p_2$	p_2
	peso	w_2	$2w_2$	$4w_2$	$8w_2$	w_2
item tipo 3: $b_3 = 18$ $\lfloor \log(18) \rfloor = 4$	lucro	p_3	$2p_3$	$4p_3$	$8p_3$	$3p_3$
	peso	w_3	$2w_3$	$4w_3$	$8w_3$	$3w_3$

Figura 4.1: Transformação de uma instância do Problema da Mochila Limitada (BKP) com 3 itens em uma instância do Problema da Mochila tradicional (KP) com 13 itens.

Percebe-se que a decomposição apresentada nesta primeira opção introduz uma grande quantidade de variáveis extras no problema, aumentando o consumo de memória dos algoritmos e introduzindo certa ineficiência no processo de resolução [GLOVER 1990]. Além disso, apesar deste processo garantir que o limite de cada tipo de item por mochila (o mercado anual) não será excedido, o PMMMLPOARE ainda possui outras restrições, como o mercado global e as restrições de dependência. Assim, para garantir que essas outras restrições sejam satisfeitas, seria necessário introduzir um processo de codificação e decodificação da solução, o que aumentaria ainda mais a dificuldade de implementação das heurísticas e custo das computacionais das mesmas. Além disso, utilizando este processo, a variação de apenas uma unidade em uma dessas variáveis artificiais pode causar uma grande variação nas variáveis reais do problema, e o mesmo pode acontecer no sentido inverso.

Na segunda opção, mais prática e intuitiva, a solução do problema é representada diretamente como um vetor de inteiros x , de tamanho igual ao número de itens(ações) vezes o número de mochilas(anos). Cada posição, ou *variável* $x_{j,i}$ deste vetor guarda quantos itens do tipo j estão alocados na mochila i . Esta opção é ilustrada na figura 4.2.

<i>Mochilas:</i>	1			2				<i>M</i>		
Solução:	$x_{1,1}$	$x_{2,1}$	$x_{3,1}$	$x_{1,2}$	$x_{2,2}$	$x_{3,2}$...	$x_{1,M}$	$x_{2,M}$	$x_{3,M}$
<i>Itens:</i>	1	2	3	1	2	3		1	2	3

Figura 4.2: Exemplo de codificação da solução de uma instância com 3 itens e M mochilas.

Por ser mais simples, não precisar de nenhum processo complexo de codificação e decodificação, consumir menos memória, e ter um mapeamento “direto” com as variáveis reais do problema, a segunda abordagem foi escolhida para codificar as soluções dos algoritmos neste trabalho. Um ponto importante a se observar é que esta codificação permite a representação de soluções inviáveis do problema, logo, os algoritmos que a utilizarem precisam de mecanismos para garantir a viabilidade das soluções.

4.1 O Algoritmo Guloso

A primeira heurística desenvolvida foi uma busca local utilizando uma estratégia gulosa. Este tipo de estratégia consiste em resolver o problema em uma sequência de passos, onde a cada passo se faz a melhor escolha momentânea, sem se preocupar com o impacto futuro desta escolha e sem desfazer escolhas passadas.

Em geral, esse tipo de estratégia tem a desvantagem de ser incapaz de escapar de ótimos locais, pois não permite desfazer passos da busca. Por outro lado, uma das vantagens é que algoritmos que a implementam geralmente convergem relativamente mais rápido, sendo esta uma abordagem interessante quando se desejam soluções viáveis rápidas. Outra característica desejável é que os algoritmos desenvolvidos com essa abordagem são, em geral, relativamente simples de entender e implementar.

O algoritmo 4.1 descreve a heurística gulosa desenvolvida para o PMMMLPOARE. Esta heurística, classificada como uma heurística construtiva de melhoramento, recebe uma solução inicial incompleta, ou seja, uma solução à qual ainda podem ser adicionados itens e melhora esta solução a cada iteração. Ela recebe também a definição do problema, ou seja, os metadados que carregam as características da instância, como o número de anos (mochilas) e ações (itens) do planejamento e as características das ações. A partir daí, algoritmo pode ser dividido em 3 etapas: Na primeira etapa, o algoritmo gera uma lista com todas as possíveis *alocações* que podem ser executadas em uma solução. Essas alocações são todas as possíveis combinações

de item versus mochila, que representam a adição de uma cópia de algum tipo de item a uma mochila. O segundo passo do algoritmo consiste em ordenar esta lista em ordem decrescente, de acordo com uma medida da *eficiência* (ver seção 4.1.2) de cada uma das alocações. Na terceira parte, esta lista com as alocações é utilizada para completar a solução inicial e retornar a solução completa encontrada.

Algoritmo 4.1: Algoritmo Guloso

Entrada: Definição: def , Solução Inicial: $S_{inicial}$
Saída: Melhor Solução Encontrada: S_{melhor}

```

1 início
2    $S_{melhor} \leftarrow S_{inicial}$ 
3    $ListaAlocacoes \leftarrow GeraAlocacoes(def)$ 
4    $ListaAlocacoes \leftarrow OrdenaDecrescente(ListaAlocacoes, def, eficiencia)$ 
5    $Melhorou \leftarrow True$ 
6   enquanto  $Melhorou$  faça:
7      $Melhorou \leftarrow False$ 
8     para  $Alocacao \in ListaAlocacoes$  && not  $Melhorou$  faça:
9       se  $eViavel(S_{melhor}, Alocacao)$  então
10         $S_{melhor} \leftarrow S_{melhor} + Alocacao$ 
11         $Melhorou \leftarrow True$ 
12      fim
13    fim
14  fim
15  retorna  $S_{melhor}$ 
16 fim

```

Na linha 3, a função $GeraAlocacoes()$ gera a partir da definição do problema, uma lista com todas as possíveis alocações de itens versus mochilas. Para exemplificar, em um problema com duas mochilas e três tipos de itens, a lista de alocações gerada seria $\{(1,1), (1,2), (2,1), (2,2), (3,1), (3,2)\}$, onde cada uma das duplas da lista representa a adição de uma cópia de um tipo de item a uma mochila. Em seguida essa lista de alocações é ordenada decrescentemente na linha 4, em função das características de cada alocação obtidas da definição do problema e de uma medida de eficiência. A partir daí, a solução inicial é melhorada iterativamente no laço entre as linhas 6 e 14. A cada passo do laço, a lista de possíveis alocações, que já está ordenada em ordem decrescente de eficiência, é percorrida procurando a alocação que produza uma solução viável com a maior eficiência possível. Uma vez que uma dessas alocações é encontrada, a solução é atualizada com a alocação (linha 10), e o algoritmo volta para o laço externo. Este laço externo é repetido indefinidamente, até que todas as alocações da lista levem a soluções inviáveis. Neste ponto, o algoritmo para e retorna a solução viável encontrada.

Observando o algoritmo, percebem-se duas decisões importantes a serem tomadas durante a implementação: a solução inicial que o algoritmo recebe e a medida de eficiência a ser utilizada

para ordenar as alocações. As seções 4.1.1 e 4.1.2 se dedicam a explicar cada uma dessas decisões.

4.1.1 Solução Inicial

A primeira decisão importante ao implementar este algoritmo é escolher qual será a solução inicial. A princípio, uma possível solução inicial seria uma sem nenhum item. Outras possíveis soluções iniciais poderiam ser geradas de forma aleatória. Entretanto, durante os testes notou-se que nenhuma das duas opções obtinha resultados melhores do que utilizar a solução do problema relaxado linearmente, truncada para se tornar viável.

A princípio, uma desvantagem dessa abordagem poderia ser o tempo gasto para resolver a relaxação linear do problema. Entretanto, os testes mostraram que os resolvidores modernos, como o *CPLEX*, conseguem resolver esta relaxação facilmente, em tempo desprezível. Não faria sentido, portanto, não utilizar no algoritmo um ponto de partida tão bom, que consome tão poucos recursos computacionais.

A obtenção da solução inicial pode então ser resumida em duas partes: a resolução da relaxação linear do problema, seguida pelo truncamento da solução obtida, observando para que as restrições de dependência permaneçam respeitadas.

4.1.2 Medida de Eficiência

O último aspecto importante a ser definido é a maneira de avaliar a eficiência de cada alocação. Esta medida de eficiência é utilizada nos algoritmos para comparar diferentes alocações e escolher qual seria mais atrativa para o algoritmo, em outras palavras, qual alocação deveria ser feita para encontrar melhores soluções.

A primeira possível maneira de medir a eficiência da alocação é considerar apenas o lucro p_j do item j , obtido ao fazer a alocação deste item em uma mochila (equação 4.1):

$$ef_{lucro} = p_j \quad (4.1)$$

Em geral, heurísticas gulosas utilizadas para resolver o Problema da Mochila costumam utilizar a noção de pseudo-utilidade, ou custo-benefício. Essa medida é usualmente definida para o problema da mochila 0-1 tradicional (equação 3.1) como a razão entre o lucro p_j do item j e o seu custo w_j . Entretanto, essa noção de custo benefício fica um tanto nebulosa para problemas que tratam de mais de um recurso e tem, portanto, mais de um custo. Uma generalização desta

ideia seria utilizar a razão entre o lucro e a somatória dos R recursos consumidos. Essa seria, então, uma segunda possível medida de eficiência:

$$ef_{cxb} = \frac{P_j}{\sum_{r=1}^R w_{j,r}} \quad (4.2)$$

Apesar de ser uma medida de eficiência viável, é fácil perceber que a equação acima não leva em consideração diferentes ordens de magnitude entre os os recursos. Isso pode ser resolvido se fazendo um escalonamento, levando em consideração não só a quantidade de recursos consumida, mas também a quantidade de recursos disponível inicialmente, c_r [PUCHINGER ET AL. 2006]:

$$ef_{escalado} = \frac{P_j}{\sum_{r=1}^R \frac{w_{j,r}}{c_r}} \quad (4.3)$$

Outra medida de eficiência, proposta em [PIRKUL 1987] e aplicada com sucesso por [CHU , BEASLEY 1998], envolve utilizar como multiplicador dos custos da ação o valor das *variáveis duais* obtidas resolvendo a relaxação linear do problema. Em outras palavras, o multiplicador d_r aplicado ao custo do item em relação ao recurso r é o *preço sombra* da restrição correspondente na solução da relaxação linear do problema [CHU , BEASLEY 1998]:

$$ef_{dual} = \frac{P_j}{\sum d_r \cdot c_{j,r}} \quad (4.4)$$

Durante a implementação do algoritmo guloso foram executados vários testes para verificar qual medida de eficiência proporcionaria melhores resultados. Partindo de uma solução vazia, os melhores resultados foram obtidos com $ef_{escalado}$, e os piores com ef_{lucro} . Entretanto, ao executar os testes partindo da solução da relaxação linear do problema truncada, o quadro se inverteu, e a medida ef_{lucro} passou a ser a melhor.

A conclusão obtida foi de que, partindo da solução vazia seria melhor utilizar $ef_{escalado}$, pois vários itens ainda seriam adicionados e, portanto, a vantagem desta medida teria tempo para fazer efeito e produzir soluções melhores. Entretanto, ao partir da solução truncada da relaxação linear do problema as mochilas já estão bem cheias, e como poucos itens serão adicionados, vale a pena ser mais agressivo e colocar logo itens que possuam o maior lucro possível.

Enfim, o algoritmo guloso escolhido implementa o código apresentado no algoritmo 4.1, partindo da solução truncada da relaxação linear do problema original, utilizando como medida de eficiência ef_{lucro} descrita na equação 4.1. A esta implementação se refere a partir deste ponto

como *Algoritmo Guloso partindo da Relaxação Linear do Problema*, ou *AGRLP*.

4.2 A Busca Tabu

A segunda heurística construída é uma implementação da metaheurística Busca Tabu [GLOVER 1989]. Esta metaheurística tem sido uma das mais utilizadas nas últimas décadas para atacar problemas de Otimização Combinatória, e por um bom tempo uma busca tabu apresentada em [VASQUEZ , HAO 2001], e posteriormente melhorada em [VASQUEZ , VIMONT 2005], foi o método que encontrou soluções de melhor qualidade para o Problema da Mochila Multidimensional.

Assim como o algoritmo guloso explicado na seção anterior, a Busca Tabu é essencialmente um método de busca local, logo, como a maioria dos métodos simples de busca local, ela teria uma tendência a ficar presa em ótimos locais. Para contornar este problema, a Busca Tabu permite que, não existindo mais nenhum movimento viável que melhore a solução corrente, seja feito um movimento que leve a uma solução pior, com a expectativa que no longo prazo esta escolha leve a soluções melhores. Além disso, neste método as características de movimentos feitos em pontos anteriores da busca são levados em consideração na hora de escolher o próximo movimento. Por ter estas características, a Busca Tabu consegue escapar melhor de ótimos locais e levar a soluções melhores que buscas locais simples.

Antes de continuar com a explicação da heurística, é importante explicar brevemente alguns termos que serão utilizados mais adiante. O primeiro deles são os *movimentos*. Movimentos são as operações efetuadas em uma solução que a levam a outra. Ao conjunto de soluções que podem ser alcançadas a partir da solução x utilizando os movimentos disponíveis, dá-se o nome de *Vizinhança* de x . Um componente característico da Busca Tabu é a Lista Tabu. Essa lista contém movimentos ou características que levam a soluções indesejadas, e é utilizada durante a busca para proibir certos movimentos, como forma de escapar de ótimos locais e direcionar a busca a soluções mais promissoras.

O pseudocódigo apresentado no algoritmo 4.2 descreve a heurística implementada. Ele recebe como entrada uma solução inicial e tenta, a cada iteração, melhorar a solução atual

fazendo uma busca em sua vizinhança por soluções melhores.

Algoritmo 4.2: Busca Tabu

Entrada: Solução Inicial: $S_{inicial}$
Saída: Melhor Solução Encontrada: S_{melhor}

```

1 início
2    $S_{atual} \leftarrow S_{inicial}$ 
3    $S_{melhor} \leftarrow S_{inicial}$ 
4    $ListaTabu \leftarrow \emptyset$ 
5   enquanto  $\neg condicaoParada$  faça:
6      $movimentos \leftarrow Vizinhanca(S_{atual})$ 
7      $movimento \leftarrow MelhorMovimentoNaoTabu(movimentos)$ 
8      $S_{atual} \leftarrow S_{atual} + movimento$ 
9      $TabuList \leftarrow TabuList + movimento$ 
10    se  $S_{atual} \geq S_{melhor}$  então
11       $S_{melhor} \leftarrow S_{atual}$ 
12       $ResetaListaTabu()$ 
13    fim
14  fim
15  retorna  $S_{melhor}$ 
16 fim

```

A parte mais importante da busca tabu acontece no laço das linhas 5 a 14. Primeiramente, todos os movimentos que levam aos possíveis vizinhos da solução atual são obtidos na linha 6. A partir desta lista de movimentos, na linha 7, a função *MelhorMovimentoNaoTabu()* busca o movimento que proporciona o maior aumento na função objetivo representada na solução atual e que não é proibido. Caso não exista nenhum movimento permitido que melhore a solução atual, é escolhido o movimento que a piore menos, e que também seja permitido. Em seguida, a solução atual é atualizada com o movimento escolhido (linha 8), a lista tabu é atualizada com o movimento escolhido (linha 9) e finalmente, caso a solução atual seja melhor que a melhor solução encontrada até então, a melhor solução é atualizada com a solução atual (linha 11) e a lista tabu é resetada, removendo todos os seus itens.

Assim como o algoritmo guloso, a busca tabu recebe como entrada uma solução inicial que será melhorada iterativamente. Mais uma vez, durante os testes o algoritmo foi testado iniciando de uma solução vazia, de soluções aleatórias e da solução do problema relaxado linearmente truncada, e a última opção obteve os melhores resultados, sendo então escolhida como a solução inicial na versão final do algoritmo implementado.

O espaço de busca que se desejou atingir com esse algoritmo foi o conjunto de todas as soluções viáveis, ou seja, soluções que respeitam as restrições do problema. Por isso, uma estrutura de vizinhança simples pôde ser implementada, que ainda cobre todo o espaço desejado. Para uma solução x , essa estrutura é representada por todas as soluções que podem ser alcan-

çadas incrementando ou decrementando alguma variável x_j de x , desde que todas as restrições permaneçam respeitadas. Logo, um movimento *mov* no algoritmo representa incrementar ou decrementar uma das variáveis da solução, mantendo a viabilidade da mesma.

Vale citar também que durante os testes medidas de eficiência análogas as apresentadas na seção 4.1.2 foram testadas na função *MelhorMovimentoNaoTabu()* para comparar os movimentos e, mais uma vez, a que considera somente o aumento na função objetivo foi a que obteve melhores resultados. Talvez isso seja devido ao fato de que, como comentado em [GLOVER 1989], a chegada a um ótimo local na busca tabu não representa uma barreira ao algoritmo, então, vale a pena ser mais agressivo nas escolhas ao invés de tentar adiar a chegada a este ponto.

Outro aspecto importante a ser considerado quando se implementa uma Busca Tabu é a implementação da Lista Tabu e seu gerenciamento. Este aspecto é tratado na seção seguinte. A Busca Tabu implementada de acordo com o algoritmo 4.2, utilizando a solução inicial e estrutura de vizinhança explicadas anteriormente e a Lista Tabu explicada na seção 4.2.1, será chamada a partir daqui de *Busca Tabu partindo da Relaxação Linear do Problema*, ou *BTRLP*.

4.2.1 A Lista Tabu e seu gerenciamento

Um dos aspectos mais importantes ao se implementar uma Busca Tabu é definir como implementar e gerenciar a Lista Tabu. Esta lista é utilizada no algoritmo para “proibir” certos movimentos da vizinhança de uma solução, a fim de guiar a busca local a regiões mais interessantes do espaço de busca, onde espera-se poder encontrar soluções distintas das já visitadas e potencialmente melhores.

A princípio, poderia-se guardar todas as soluções visitadas em uma lista. Assim, ao fazer um movimento que levasse da solução x à solução x' , a solução x seria guardada nesta lista com todas as suas características, que no caso do problema aqui resolvido, seriam todas as quantidades de cada item em cada mochila. Conseqüentemente, para saber se um movimento é tabu, ou seja, para saber se um movimento é proibido pela Lista Tabu, basta percorrer esta lista verificando se o movimento leva a solução atual a alguma solução presente na lista, que significaria que o movimento leva a uma solução que já foi visitada. Dado um tamanho suficiente a esta lista, é fácil perceber que ela funcionaria no propósito de evitar que o algoritmo andasse em círculos, já que não seria possível visitar uma solução já visitada anteriormente. Entretanto, o tempo de processamento para verificar a lista e fazer a comparação solução a solução, e o consumo de memória para guardar essa informação das soluções completas seriam altíssimos.

Outra possível abordagem para a implementação da Lista Tabu, com o objetivo de combater o alto consumo de memória e processamento seria guardar, ao invés de todas as soluções visitadas, somente os movimentos que foram feitos nas transições entre uma solução e outra. Assim quando o algoritmo fizesse a transição de uma solução x para outra x' , não seria guardada a solução x completa na lista, e sim o movimento mov que quando aplicado a x leva a x' . Ao verificar se um possível movimento é tabu, seria necessário percorrer esta lista verificando apenas se o movimento está presente ou não. Este tipo de implementação, embora diminua o consumo de recursos computacionais quando comparada a abordagem anterior, permite que o algoritmo desfaça um movimento assim que o mesmo for feito, levando a uma solução que já foi visitada.

Segundo [GLOVER 1989], este tipo de gerenciamento da Lista Tabu, onde é prevenida a repetição ao invés da reversão de movimentos não funciona muito bem na prática. Uma opção tão simples quanto esta mas que geralmente apresenta resultados melhores é, ao invés de guardar na lista o movimento mov que leva uma solução x a x' , guardar o inverso dele, ou seja, o movimento \overline{mov} , que levaria de x' a x . Claramente, este método tem as mesmas vantagens do anterior em termos de consumo de recursos, e ainda previne que o algoritmo volte a visitar soluções por onde já passou. Agora, a desvantagem deste método é que este tipo de lista tabu é muito restritivo, pois ao se proibir um movimento em específico, é proibido na verdade todo um conjunto de soluções que só seriam alcançáveis se aplicando este movimento.

Uma maneira de se resolver este problema de uma lista tabu muito restritiva é utilizar um gerenciamento dinâmico da lista tabu. Um desses métodos, proposto em [GLOVER 1990], é conhecido com Método da Eliminação Reversa (MER), e foi aplicado com sucesso em diversas Buscas Tabu desenvolvidas para atacar o Problema da Mochila Multidimensional [DAMMEYER , VOSS 1993, HANAFI , FREVILLE 2001, VASQUEZ , VIMONT 2005]. Embora esses trabalhos apliquem o MER a problemas com variáveis binárias, e no PMMMLPOARE as variáveis tenham limites superiores maiores que um, existe uma especialização do MER que torna esse método aplicável ao PMMMLPOARE.

Neste método, ainda existe uma lista tabu utilizada para guardar todos os movimentos executados durante a busca. Entretanto, a simples presença de um movimento nesta lista não implica que ele representa um movimento proibido, e sim que ele representa um movimento *potencialmente* proibido. Vale lembrar aqui que se um movimento mov representa adicionar uma unidade a uma variável x_j do problema, então seu complemento, \overline{mov} representa retirar uma unidade da mesma variável.

A gerência da Lista Tabu (LT) é feita com o auxílio de outra lista chamada de *Lista Tabu*

Ativa (LTA). Enquanto a LT guarda os movimentos que já foram feitos durante a busca, e potencialmente não podem ser desfeitos, a LTA guarda quais movimentos efetivamente não podem ser desfeitos na iteração, para não levar a soluções já visitadas. Logo, a cada iteração da busca, a LT é atualizada com a adição do movimento feito e a LTA é totalmente recalculada considerando os movimentos presentes na LT.

A atualização da LTA é feita da seguinte forma [GLOVER 1990]: Considerando a LT como uma lista contendo todos os movimentos executados nas iterações anteriores ordenados pela iteração em que foram executados, sempre que um novo movimento mov é feito na iteração q do algoritmo, ele é adicionado ao final desta LT que agora tem a forma $LT = (mov(1), mov(2), \dots, mov(q))$, e um rastreo é feito nesta lista para produzir a LTA. Especificamente, a LT é rastreada do fim até o começo, e a cada $mov(i)$ visitado, um vetor *desvio* guardando as diferenças entre a solução da iteração i , $x(i)$, e a solução corrente $x(q)$ é atualizado, associado a uma variável n que guarda a soma dos valores absolutos presentes neste vetor.

A atualização do vetor *desvio* durante o rastreo de $mov(q)$ a $mov(i)$ ocorre da seguinte maneira: o vetor é inicializado com todos os j elementos contendo o valor 0. Cada movimento $mov(h)$ deste rastreo representa o aumento ou decréscimo de uma variável x_j , então, baseado nisso, o j -ésimo componente do vetor é ajustado na direção contrária, e a variável n é atualizada de acordo.

Código da Lista Tabu

j significa aumentar a variável x_j em 1 unidade

$-j$ significa reduzir a variável x_j em 1 unidade

Lista Tabu (LT)

i	1	2	3	4	5	6	7	8	9	10	11
mov(i)	-3	-3	-1	3	-4	2	-4	3	4	4	-2

Rastreio da LT utilizando o Método da Eliminação Reversa

i	desvio(j)				n	Lista Tabu Ativa (LTA)
	j = 1	2	3	4		
11		1			1	aumentar x_2
10		1		-1	2	
9		1		-2	3	
8		1	-1	-2	4	
7		1	-1	-1	3	
6			-1	-1	2	
5			-1		1	diminuir x_3
4			-2		2	
3	1		-2		3	
2	1		-1		2	
1	1				1	aumentar x_1

Figura 4.3: Exemplo do Método da Eliminação Reversa especializado para variáveis limitadas.

Não é difícil perceber que a solução $x(q+1)$ será a mesma que $x(i)$ somente se $n(i) = 0$. Assim, se durante este rastreio $n(i)$ assume o valor 1, a única forma de um possível movimento $mov(q+1)$ levar a uma solução idêntica a $x(i)$ é se $mov(q+1)$ for igual ao último elemento não zero do vetor *desvio*. Neste caso, o movimento representado neste elemento é considerado tabu e adicionado à LTA. Na iteração seguinte, para verificar se o movimento mov é tabu, basta verificar se ele está presente na LTA. A figura 4.3 mostra a atualização da LTA ao se inserir um movimento no final da LT.

4.3 O Gerador de Instâncias

Para testar a dificuldade do PMMMLPOARE e comparar a qualidade das soluções geradas pelas abordagens propostas nas seções anteriores, é necessária uma quantidade de instâncias do problema suficiente para que seja possível fazer conclusões úteis. Infelizmente, a distribuidora local de energia não foi capaz de fornecer esta quantidade de instâncias, logo, um gerador de instâncias artificiais teve de ser criado para a condução dos testes. Este gerador cria instâncias de acordo com a formulação descrita no capítulo 2, com parâmetros que podem ser ajustados para criar instâncias de diferentes tamanhos e características.

O gerador criado recebe como parâmetros:

- Y , o número de anos;
- A , o número de ações;
- R , o número de recursos ou orçamentos;
- α , o nível de correlação entre o custo e o lucro das ações.

O algoritmo de geração de instâncias é composto de três partes principais: A geração das ações, a geração dos orçamentos e metas anuais e a geração das dependências entre ações. Ao gerar as ações, o primeiro componente a ser criado é o valor da energia recuperada pela ação (v_a), seguido dos mercados anual ($u_{a,y}$) e global (m_a). Nesta parte também é criado o custo da ação em cada orçamento ($c_{a,r}$), e a curva de redução de perda de energia da ação, que é simplesmente um vetor representando quanta perda de energia será evitada em cada ano após a ação ser executada. A segunda parte do algoritmo envolve a criação das metas anuais de redução de perdas (g_y) e o orçamento anual para cada recurso ($o_{y,r}$). A última parte gera algumas dependências aleatórias entre as ações. O pseudo-código para o gerador é apresentado

no algoritmo 4.3.

Algoritmo 4.3: Gerador de Instâncias

Entrada: Y, A, R, α

Saída: Instância gerada

```

1 início
2    $r \leftarrow 0.10$ 
3   para  $a = 1$  a  $A$  faça:
4      $v_a \leftarrow RealAleatorio(1.0, 2.0)$ 
5     para  $y = 1$  a  $Y$  faça:
6        $u_{a,y} \leftarrow InteiroAleatorio(0, 50)$ 
7     fim
8      $m_a \leftarrow 0.9 * Soma(u_a)$ 
9     para  $r = 1$  a  $R$  faça:
10       $c_{a,r} \leftarrow RealAleatorio(1.0, 100.0)$ 
11    fim
12    para  $y = 1$  a  $Y$  faça:
13       $W_y \leftarrow RealAleatorio(0.0, 1.0)$ 
14    fim
15    para  $y = 1$  a  $Y$  faça:
16       $W_y \leftarrow W_y / Soma(W_y)$ 
17    fim
18    para  $y = 1$  a  $Y$  faça:
19       $e_{a,y} \leftarrow ((1 - \alpha) \times ((2 \times Soma(c_a) + 10) / v_a) + (\alpha \times RealAleatorio(1.0, 100.0) \times R)) \times W_y \times (1 + r)^y$ 
20    fim
21  fim
22  para  $y = 1$  a  $Y$  faça:
23     $aux \leftarrow 0.5 * Soma(e_y) / Y$ 
24     $g_y \leftarrow RealAleatorio(0.95 * aux, 1.05 * aux)$ 
25    para  $r = 1$  a  $R$  faça:
26       $aux \leftarrow 0.5 * Soma(c_{y,r}) / Y$ 
27       $o_{y,r} \leftarrow RealAleatorio(0.95 * aux, 1.05 * aux)$ 
28    fim
29  fim
30  para  $i = 1$  a  $A/10$  faça:
31    GeraDependencia()
32  fim
33 fim

```

As linhas 3 a 21 mostram a criação das A ações. O valor da energia gerada é criado na linha 4 como um valor real entre 1 e 2. O mercado anual da ação é gerado na linha 6 como um inteiro aleatório entre 0 e 50 e o mercado global é criado na linha 8, como 90% da soma dos mercados anuais. O mercado global é criado um pouco abaixo da soma dos mercados anuais, pois se fosse igual ou maior a restrição de mercado global seria inútil, já que nunca seria atingida. Na linha 10, são criados os custos da ação em cada recurso. As linhas 12 a 17 constróem um vetor aleatório de tamanho Y onde a soma de todos os componentes é 1. Este

vetor é utilizado para criar a curva de redução de perdas na linha 19.

O total de perdas evitadas pela ação é criado em função de dois componentes: a soma dos custos da ação acrescida de um valor fixo e um número aleatório. O parâmetro de correlação α é utilizado para dosar quando de cada componente será utilizado na criação da curva. Quando for 0, somente o primeiro componente é utilizado, o que significa que a quantidade de energia recuperada, e conseqüentemente o lucro da ação, é fortemente correlacionada ao custo total da ação, ou seja, ações que consomem mais recursos recuperam mais energia. No outro extremo, quando o valor do parâmetro de correlação for 1, somente a segunda componente, aleatória, é utilizada, então o lucro da ação não tem relação com o custo da mesma. A influência e o significado desse fator de correlação são melhores explicados na seção 4.3.1. O total de perdas evitadas é então dividido pelos anos da ação, para gerar a Curva de Redução de Perdas, utilizando o vetor normalizado criado anteriormente.

Nas linhas 22 a 29, a meta de redução e orçamentos para cada recurso são criados, para cada ano. As metas anuais foram criadas como a metade da soma do total de energia recuperada por todas as ações, dividido pelo total de anos ($\pm 5\%$). Essa abordagem foi escolhida para garantir que as soluções não teriam poucas ações, nem seriam compostas de quase todas as possíveis alocações de ações. Os orçamentos foram criados de forma análoga, mas um orçamento por recurso por ano é criado.

A linha 31 cria dependências entre as ações. Para cada execução do loop, a função *Gera-Dependencia()* escolhe duas ações e gera uma dependência entre elas, tomando cuidado para evitar dependências cíclicas, pois nesse caso as ações envolvidas nunca satisfariam a restrição de dependência, e nunca seriam escolhidas.

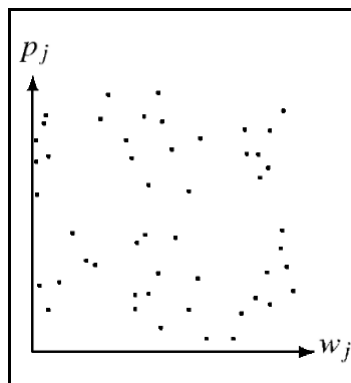
4.3.1 O Fator de Correlação entre os Itens de uma Instância

Segundo [PISINGER 2005], um dos fatores que influencia fortemente a dificuldade do Problema da Mochila tradicional é o nível de correlação existente entre o lucro e o custo dos itens da instância do problema. Aparentemente, quanto maior o nível de correlação, mais difícil o problema é para os algoritmos exatos. Baseado no nível de correlação, [PISINGER 2005] sugere então alguns tipos de instâncias que podem refletir características interessantes, entre elas:

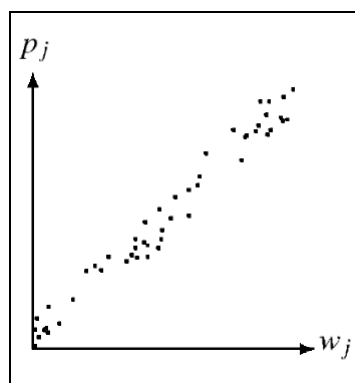
- *Não correlacionadas*: são as instâncias onde o lucro e custo dos itens não possuem nenhuma correlação. Este tipo de instância é geralmente fácil de resolver, e ilustra os casos reais onde é razoável assumir que o lucro de um item não depende de nenhuma maneira de seu custo.

- *Fracamente correlacionadas*: são as instâncias onde o lucro de um item é diretamente proporcional ao custo do mesmo, acrescido de um valor fixo e um pequeno valor aleatório. Este tipo de instância é, talvez, o mais parecido com instâncias encontradas em problemas reais, onde o retorno de um investimento é geralmente proporcional ao seu custo, com pequenas variações.
- *Fortemente correlacionadas*: são as instâncias onde o lucro de um item é diretamente proporcional ao custo do mesmo, acrescido de um valor fixo. Esse tipo de instância é, em geral, um dos mais difíceis de resolver.

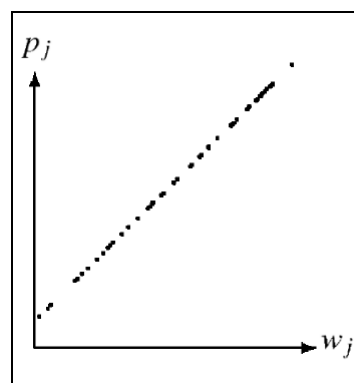
A figura 4.4 ilustra cada um destes tipos. Nela, cada gráfico representa uma instância do problema de um dos tipos citados acima, sendo cada ponto do gráfico um item da instância. O eixo p_j representa o lucro do item e o eixo w_j representa o custo do item.



(a) Sem correlação



(b) Fraca correlação



(c) Forte correlação

Figura 4.4: Exemplos de correlação entre lucro e custo dos itens de uma instância.

5 Resultados Experimentais

Com o objetivo de testar as heurísticas implementadas e a dificuldade das instâncias criadas pelo gerador, foram executados testes computacionais com os dois algoritmos e um resolvidor genérico.

Primeiramente, as instâncias geradas foram resolvidas com o resolvidor genérico, visando verificar a dificuldade das mesmas. O resolvidor utilizado foi o CPLEX 12.5.0 [Cpl 2014] em sua configuração padrão.

Em seguida, os algoritmos AGRLP e BTRLP foram executados nas mesmas instâncias. Enquanto o AGRLP não possui nenhum parâmetro ajustável, o BTRLP possui 2, a condição de parada, que durante os testes foi 10000 iterações e o tamanho máximo da Lista Tabu que foi 10000 movimentos. Estes parâmetros foram escolhidos empiricamente, experimentando diferentes combinações.

Todos estes experimentos foram executados em máquinas com processadores Intel Core i5-3570 @ 3.40 GHz, com 8GB de memória RAM e o sistema operacional Ubuntu 13.04. As duas heurísticas e o gerador foram implementados em Java e testados na JVM 1.7.

Para executar os experimentos várias instâncias artificiais tiveram que ser criadas. Essas instâncias foram geradas utilizando o gerador de instâncias apresentado na seção 4.3, para combinações de 3 a 6 anos, 25, 50 e 100 ações, 1, 2 e 4 recursos, e α como 0.0, 0.1 e 1.0. Estes valores para as quantidades de anos, ações e recursos foram escolhidos para simular as dimensões das instâncias que se espera encontrar na prática. Já os valores de α foram escolhidos para testar se a dificuldade do PMMMLPOARE também é sensível ao fator de correlação como o problema 0-1 tradicional, e representam respectivamente instâncias com forte correlação, fraca correlação e sem correlação entre os itens. Assim, foram geradas 100 instâncias para cada possível combinação destes parâmetros, totalizando 10800 instâncias.

A próxima subseção mostra uma análise da dificuldade utilizando os resultados dos testes com o CPLEX, e em seguida são mostrados os resultados dos testes das duas heurísticas implementadas.

5.1 Dificuldade do Problema

Para analisar a dificuldade das instâncias do problema geradas, todas elas foram testadas no CPLEX para se obter o tempo de resolução e as soluções resultantes. Durante os testes preliminares, enquanto algumas destas instâncias eram solucionadas rapidamente pelo resolvidor, outras demoravam demais para serem resolvidas. Por isto uma estratégia alternativa para a execução dos testes foi adotada: instâncias que demorassem mais de 20 minutos para serem resolvidas foram consideradas proibitivas para a continuidade do experimento. Quando uma execução do CPLEX chegava então a 20 minutos, ela era interrompida, e a solução no momento da interrupção era salva.

As figuras 5.1, 5.2 e 5.3 mostram os resultados destes testes, considerando as taxas de interrupção. Cada uma das três figuras representa uma das configurações de correlação, e cada um de seus heatmaps representa uma quantidade de recursos. As cores nos heatmaps representam a razão de instâncias interrompidas durante a execução do CPLEX, variando das combinações de ações/anos onde nenhuma instância foi interrompida (os quadrados mais claros) às combinações onde todas as instâncias foram interrompidas (quadrados mais escuros). O número exato de instâncias interrompidas para cada configuração está listado nas tabelas disponíveis no Apêndice A.

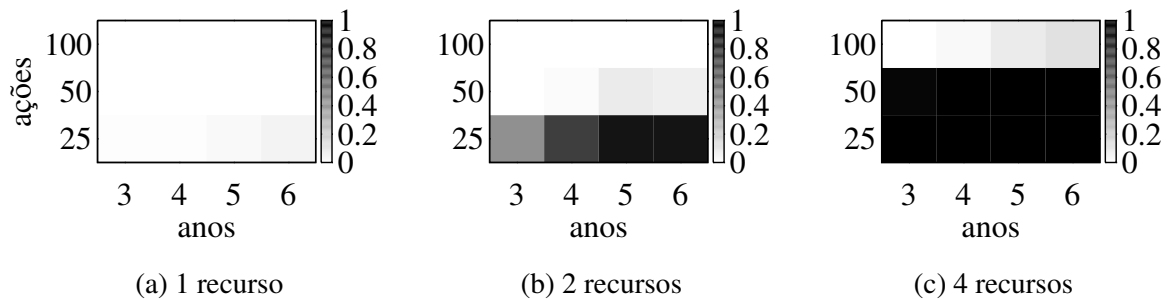


Figura 5.1: Proporção de instâncias de itens com forte correlação interrompidas ($\alpha = 0.0$).

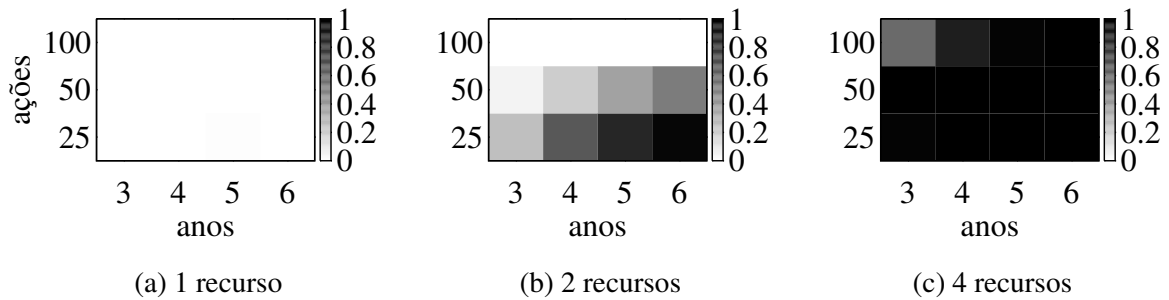


Figura 5.2: Proporção de instâncias de itens com fraca correlação interrompidas ($\alpha = 0.1$).

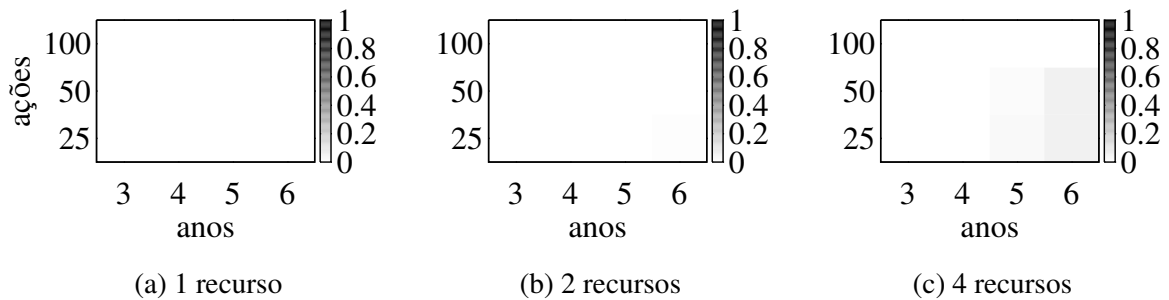


Figura 5.3: Proporção de instâncias de itens sem correlação interrompidas ($\alpha = 1.0$).

Comparando as figuras 5.1 e 5.2 à figura 5.3 pode-se perceber que as primeiras possuem tons muito mais escuros, o que significa que o nível de correlação realmente afeta a dificuldade de resolução das instâncias. Como esperado, instâncias sem correlação entre os itens são claramente mais fáceis para o CPLEX, com poucas instâncias de tamanhos maiores sendo interrompidas. Entretanto, ao contrário do esperado, as figuras 5.1 e 5.2 mostram que as instâncias geradas com fraca correlação parecem mais difíceis que as com forte correlação.

A respeito da quantidade de anos e recursos das instâncias, os resultados obtidos batem com o que era esperado, pois as figuras 5.1 a 5.3 mostram que quanto maiores as instâncias são nestes dois parâmetros, mais difícil de serem resolvidas. Ainda assim, comparando as três figuras e observando a dificuldade quando se varia o número de ações mantendo os outros parâmetros fixos, as instâncias com 25 ações aparentam ser mais difíceis de resolver que as com 50 e 100.

Em relação ao tempo de execução, o CPLEX precisou de 380 segundos, na média de todas as instâncias. Já a média do gap medido para todas as instâncias ficou em 0.01%. Isso significa que, na média, o CPLEX encontrou soluções que estavam a no máximo 0.01% da solução ótima.

5.2 Qualidade das Heurísticas

Para analisar a qualidade das soluções obtidas pelas heurísticas, as mesmas instâncias testadas anteriormente foram resolvidas aplicando o AGRLP e o BTRLP, e as soluções encontradas foram comparadas às soluções obtidas pelo CPLEX antes de atingir o limite de tempo. As figuras 5.4 a 5.6 mostram os resultados desta comparação para o BTRLP. Da mesma forma que na seção anterior, cada uma das figuras mostra uma das configurações de correlação testadas, e cada um dos heatmaps representa a quantidade de recursos daquele conjunto de instâncias. A cor nos heatmaps representa a qualidade média das soluções encontradas em instâncias com determinada combinação de anos x ações. A escala de cor representa a média da razão entre as soluções encontradas para determinadas instâncias e a melhor solução conhecida, ou seja, quanto mais escuro, melhor a qualidade das soluções encontradas. As mesmas representações são utilizadas nas figuras 5.7 a 5.9 para apresentar os resultados obtidos pelo AGRLP. A média da razão para as cem instâncias com cada combinação de parâmetros está listada nas tabelas do Apêndice A.

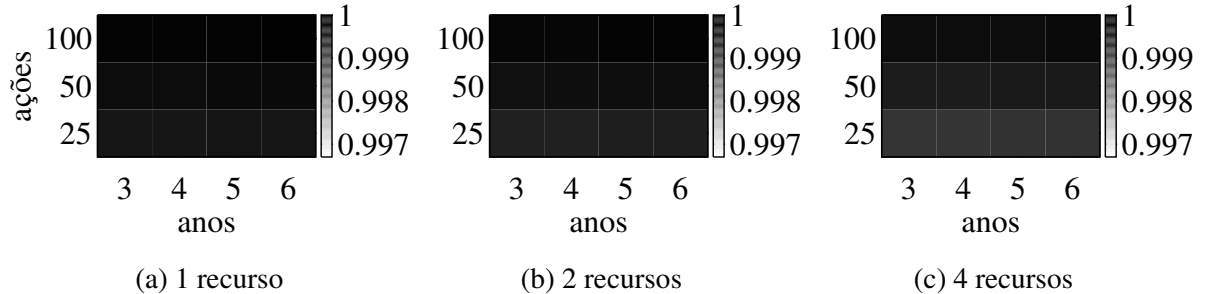


Figura 5.4: Qualidade das soluções do BTRLP em instâncias com itens com forte correlação ($\alpha = 0.0$).

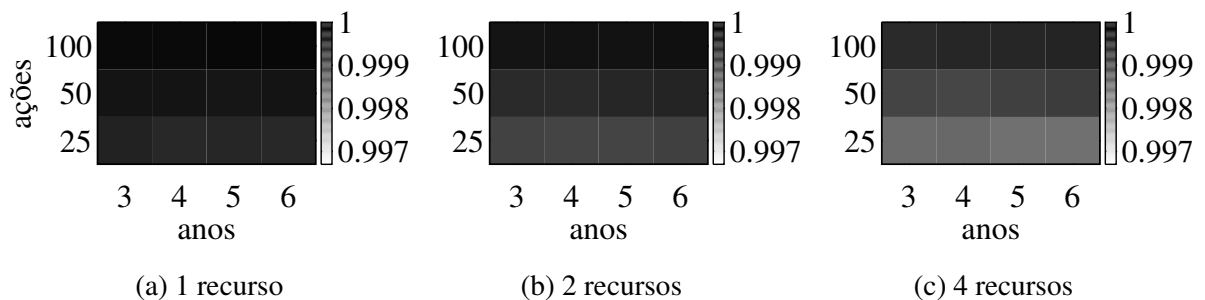


Figura 5.5: Qualidade das soluções do BTRLP em instâncias com itens com fraca correlação ($\alpha = 0.1$).

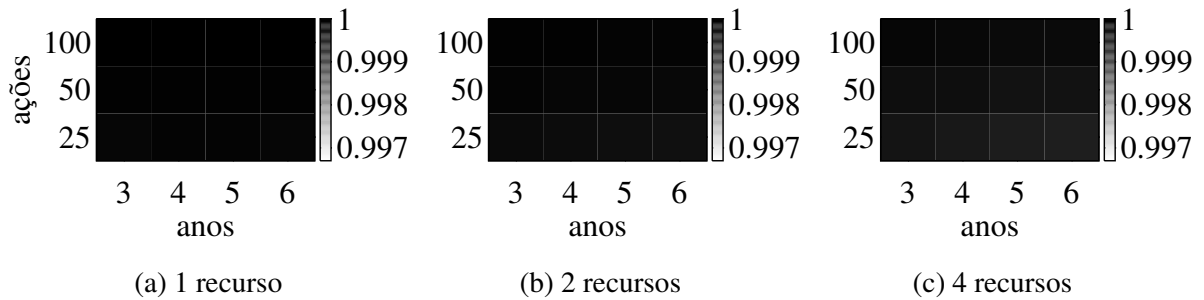


Figura 5.6: Qualidade das soluções do BTRLP em instâncias com itens sem correlação ($\alpha = 1.0$).

Observando as figuras 5.4, 5.5 e 5.6, que mostram os resultados dos testes executados com o BTRLP, pode-se perceber novamente algum impacto do fator de correlação na qualidade das soluções obtidas com o algoritmo, pois os heatmaps das instâncias com algum nível de correlação se apresentam mais claros que os das instâncias sem correlação. Em especial, nas instâncias com fraca correlação (figura 5.5) o algoritmo parece obter as piores soluções.

Comparando os resultados a respeito do número de recursos nas instâncias, também pode se perceber que as instâncias com maior número de recursos são as mais difíceis de resolver. Essa influência pode ser percebida principalmente na figura 5.5. Novamente, ao observar a variação da qualidade das soluções relacionando-a à variação da quantidade de ações das instâncias, nas instâncias com 25 ações o algoritmo obtém soluções piores que nas instâncias com mais itens. Nestes testes, não foi possível perceber a influência da quantidade de anos na qualidade das soluções.

Nas figuras 5.7, 5.8 e 5.9 o resultado da aplicação do AGRLP é apresentado. Mais uma vez, é possível perceber uma diminuição da qualidade das soluções com o aumento da quantidade de recursos e do nível de correlação das instâncias. Além disso, novamente o aumento da quantidade de ações das instâncias parece propiciar que o algoritmo ache soluções melhores, e a influência do aumento do número de anos não foi perceptível também no AGRLP.

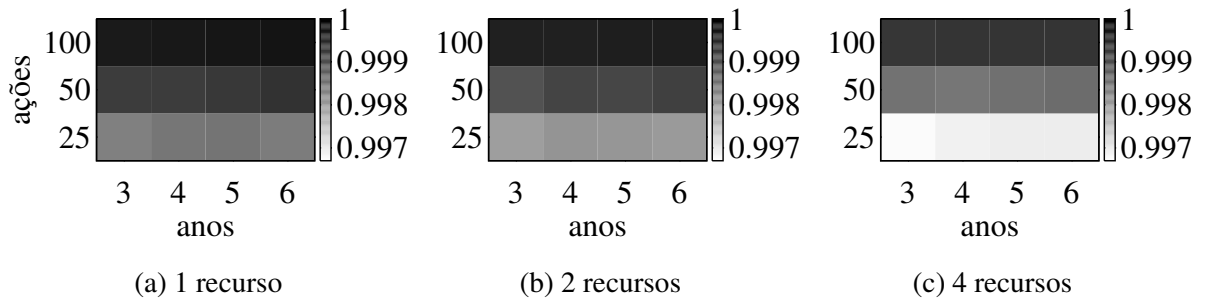


Figura 5.7: Qualidade das soluções do AGRLP em instâncias com itens com forte correlação ($\alpha = 0.0$).

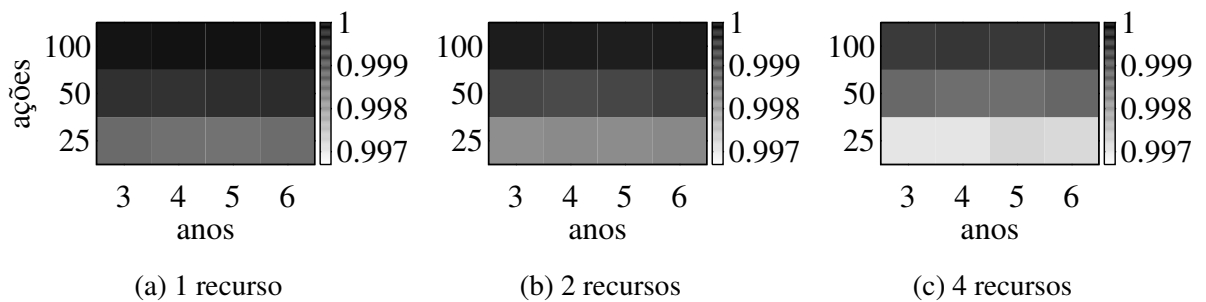


Figura 5.8: Qualidade das soluções do AGRLP em instâncias com itens com fraca correlação ($\alpha = 0.1$).

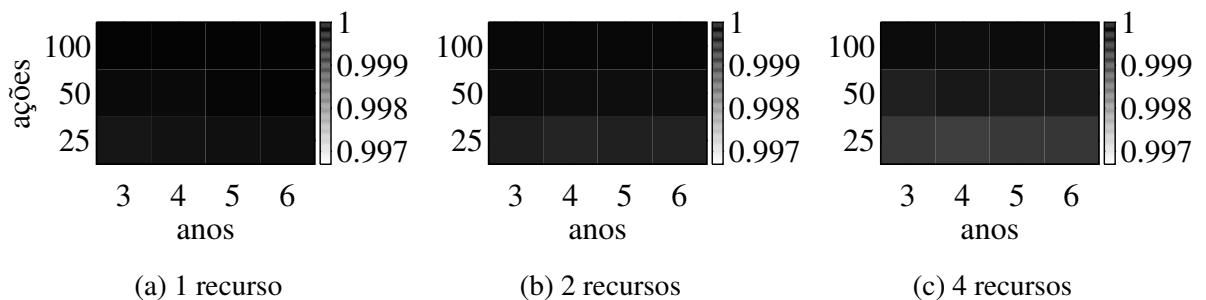


Figura 5.9: Qualidade das soluções do AGRLP em instâncias com itens sem correlação ($\alpha = 1.0$).

Observando as figuras, pode-se fazer também uma comparação entre as duas heurísticas propostas. Comparando o grupo das figuras 5.4, 5.5 e 5.6 com o das 5.7, 5.8 e 5.9, o tom mais escuro do primeiro em relação ao segundo é evidente. Isto indica que, para estas instâncias, o BTRLP se sai melhor que o AGRLP em relação à qualidade das soluções encontradas na média.

Mesmo considerando que o ponto de partida das heurísticas já era uma boa solução, já que estava na média a 99.652% da melhor solução conhecida, as heurísticas foram capazes de

melhorar o resultado. Na média, as soluções encontradas pelo AGRLP ficaram a 99.912% das melhores soluções conhecidas e o BTRLP obteve resultados ainda melhores, com soluções a 99.967% da melhores, na média com todos os tipos de instâncias.

Quanto ao tempo de execução, o BTRLP teve um tempo de execução máximo de 2 minutos, com uma média de 30 segundos. O AGRLP teve um tempo de execução desprezível, menor que 1 segundo, para todas as instâncias.

6 Conclusão e Trabalhos Futuros

Este trabalho teve como principais objetivos entender e (1) modelar o problema de alocação de ações de redução de perda de energia, (2) atestar a dificuldade do modelo gerado e (3) aplicar formas de resolução exatas e heurísticas ao problema.

O problema de alocação de ações de redução de perda de energia, visando aumentar o lucro da distribuidora foi modelado como uma junção de várias generalizações do famoso Problema da Mochila. Assim, a meta (1) foi alcançada ao se juntar o Problema da Mochila Limitada, o Problema da Mochila Multidimensional, o Problema das Mochilas Múltiplas e o Problema da Mochila Parcialmente Ordenada em um só modelo para descrever o problema original.

Ao se buscar atingir a meta (1), outra contribuição interessante foi alcançada, com a definição de duas generalizações para o Problema da Mochila, nomeadas neste trabalho como PMMMLPO e PMMMLPOARE. A primeira é composta de várias generalizações conhecidas do Problema da Mochila 0-1 tradicional e a segunda é uma generalização da primeira com algumas restrições menos rígidas, utilizada para modelar o problema original da distribuidora. Estas duas generalizações se mostraram bastante peculiares, e não foram encontrados trabalhos anteriores tratando das mesmas. Devido à generalidade delas, espera-se que elas possam ser utilizadas para modelar mais problemas reais com maior nível de detalhes.

Como forma de atestar a dificuldade do modelo e atingir o objetivo (2), primeiramente foi feita a comparação do modelo da distribuidora, o PMMMLPOARE, com o PMMMLPO e as variações do problema da mochila tradicional que o compõem. Desta forma, foi possível mostrar que como as variações do problema da mochila que formam o PMMMLPO são \mathcal{NP} -hard, o PMMMLPO também o é, e conseqüentemente, como o PMMMLPOARE é uma generalização do PMMMLPO, ele também é \mathcal{NP} -hard.

Para executar os testes computacionais seria necessária uma quantidade suficiente de instâncias do problema real para que fosse possível obter conclusões úteis. Como não foi possível conseguir esta quantidade de instâncias da distribuidora local, outra contribuição foi feita na forma da criação de um gerador de instâncias aleatórias, detalhado neste trabalho, capaz de

gerar instâncias com diversas configurações.

Ainda como forma de atingir o objetivo (2), instâncias aleatórias do modelo foram geradas e resolvidas com o conhecido resolvidor genérico CPLEX. Estes experimentos mostraram que, como esperado, após um pequeno aumento em algumas das dimensões das instâncias, o resolvidor já não consegue encontrar a solução ótima dentro do tempo estabelecido. Foi atestado também que embora não exatamente como previsto na literatura, a dificuldade do problema da distribuidora também é sensível ao aumento no nível de correlação entre o lucro e o custo das ações da instância.

Para atingir o objetivo (3), além da utilização do CPLEX duas heurísticas foram propostas, uma baseada em uma abordagem gulosa (AGRLP) e outra baseada na metaheurística Busca Tabu (BTRLP). Ambas foram testadas nas mesmas instâncias resolvidas anteriormente com o CPLEX, e as soluções encontradas pelas heurísticas foram comparadas com as melhores soluções conhecidas. Esta comparação mostrou que as duas heurísticas encontraram soluções de boa qualidade, muito próximas das melhores soluções conhecidas. Em particular, o BTRLP se saiu melhor que o AGRLP em relação à qualidade das soluções encontradas, embora precise de um tempo de execução maior.

Apesar disso, vale citar que as melhores soluções encontradas para as instâncias foram as fornecidas pelo CPLEX. Além disso, mesmo nos casos em que foi interrompido antes do final da execução, nos piores casos a solução encontrada estava a no máximo 0,35% da solução ótima do problema. Esta porcentagem representaria, segundo o relatório financeiro da distribuidora local, R\$ 140.000,00 [ABREU ET AL. 2011]. Sendo o método que encontrou as melhores soluções para todas as instâncias e tendo encontrado excelentes soluções mesmo no caso onde foi interrompido, o CPLEX mostrou que, embora talvez não seja o resolvidor mais adequado para instâncias com dimensões gigantescas, para as instâncias do problema com dimensões que se espera encontrar na prática ele pode ser a melhor opção.

Possíveis trabalhos futuros incluem:

- investigar mais profundamente o porquê de a resolução de algumas instâncias demorar muito mais que outras, mesmo tendo a mesma quantidade de anos, ações, recursos e correlação, e serem geradas aleatoriamente da mesma forma;
- investigar o comportamento inesperado observado em relação ao nível de correlação, ou seja, por que as instâncias fracamente correlacionadas parecem mais difíceis que as fortemente correlacionadas;
- implementar melhorias na Busca Tabu, na forma de outros tipos de memória de inter-

mediário e longo prazo, para prover níveis diferentes de intensificação e diversificação durante a busca;

- implementar e aplicar outras heurísticas a este problema, de forma a melhorar os resultados obtidos.

Referências Bibliográficas

- [Anl] Agência Nacional de Energia Elétrica. URL <http://www.aneel.gov.br/>. Acessado em: 19/05/2015.
- [Cpl 2014] IBM ILOG CPLEX Optimization Studio. 2014. URL www.ibm.com/software/commerce/optimization/cplex-optimizer/. Acessado em: 19/05/2015.
- [ABREU ET AL. 2011] Abreu, A. M. B. P. d., Setas, M. N. S. N. F. , Martins, C. S. S. Demonstrações Financeiras 2011. http://www.edp.com.br/conheca-edp/relatorios/Documents/relatorio_2011.pdf, 2011.
- [AL-SHIHABI , ÓLAFSSON 2010] Al-Shihabi, S. , Ólafsson, S. A hybrid of Nested Partition, Binary Ant System, and Linear Programming for the multidimensional knapsack problem. *Computers & Operations Research*, volume 37(2):247 – 255, 2010.
- [AMARANTE ET AL. 2013] Amarante, S. R. M., Roberto, F. M., Cardoso, A. R. , Celestino, J. Using the Multiple Knapsack Problem to Model the Problem of Virtual Machine Allocation in Cloud Computing. *2013 IEEE 16th International Conference on Computational Science and Engineering*, volume 0:476–483, 2013.
- [BEHESHTI ET AL. 2013] Beheshti, Z., Shamsuddin, S. , Yuhaziz, S. Binary Accelerated Particle Swarm Algorithm (BAPSA) for discrete optimization problems. *Journal of Global Optimization*, volume 57(2):549–573, 2013.
- [BORRADAILE ET AL. 2012] Borradaile, G., Heeringa, B. , Wilfong, G. The knapsack problem with neighbour constraints. *Journal of Discrete Algorithms*, volume 16(0):224 – 235, 2012.
- [BOUSSIER ET AL. 2010] Boussier, S., Vasquez, M., Vimont, Y., Hanafi, S. , Michelon, P. A multi-level search strategy for the 0-1 Multidimensional Knapsack Problem. *Discrete Applied Mathematics*, volume 158(2):97–109, 2010.
- [CHIH ET AL. 2014] Chih, M., Lin, C. J., Chern, M. S. , Ou, T. Y. Particle swarm optimization with time-varying acceleration coefficients for the multidimensional knapsack problem. *Applied Mathematical Modelling*, volume 38(4):1338 – 1350, 2014.
- [CHU , BEASLEY 1998] Chu, P. , Beasley, J. A Genetic Algorithm for the Multidimensional Knapsack Problem. *Journal of Heuristics*, volume 4(1):63–86, 1998.
- [DAMMEYER , VOSS 1993] Dammeyer, F. , Voß, S. Dynamic tabu list management using the reverse elimination method. *Annals of Operations Research*, volume 41(2):29–46, 1993.
- [DAWANDE ET AL. 2000] Dawande, M., Kalagnanam, J., Keskinocak, P., Salman, F. S. , Ravi, R. Approximation algorithms for the multiple knapsack problem with assignment restrictions. *Journal of Combinatorial Optimization*, volume 4(2):171–186, 2000.
- [DELLA CROCE , GROSSO 2012] Della Croce, F. , Grosso, A. Improved Core Problem Based Heuristics for the 0/1 Multi-dimensional Knapsack Problem. *Computers and Operations Research*, volume 39(1):27–31, 2012.

- [FREVILLE 2004] Freville, A. The multidimensional 0–1 knapsack problem: An overview. *European Journal of Operational Research*, volume 155(1):1 – 21, 2004.
- [GAREY , JOHNSON 1978] Garey, M. R. , Johnson, D. S. Strong NP-Completeness Results: Motivation, Examples, and Implications. *J ACM*, volume 25(3):499–508, 1978.
- [GLOVER 1989] Glover, F. Tabu Search — Part I. *ORSA Journal on Computing*, volume 1(3):190–206, 1989.
- [GLOVER 1990] Glover, F. Tabu Search — Part II. *ORSA Journal on Computing*, volume 2(1):4–32, 1990.
- [HANAFI , FREVILLE 2001] Hanafi, S. , Freville, A. Extension of Reverse Elimination Method Through a Dynamic Management of the Tabu List. *RAIRO - Operations Research*, volume 35(2):251–267, 2001.
- [KELLERER ET AL. 2004] Kellerer, H., Pferschy, U. , Pisinger, D. Knapsack Problems. Springer, 2004.
- [KOLLIPOULOS , STEINER 2007] Kolliopoulos, S. G. , Steiner, G. Partially ordered knapsack and applications to scheduling. *Discrete Applied Mathematics*, volume 155(8):889 – 897, 2007.
- [LAMBERT , NEWMAN 2014] Lambert, W. , Newman, A. Tailored Lagrangian Relaxation for the open pit block sequencing problem. *Annals of Operations Research*, volume 222(1):419–438, 2014.
- [MARTINS ET AL. 2014] Martins, J. P., Fonseca, C. M. , Delbem, A. C. B. On the performance of linkage-tree genetic algorithms for the multidimensional knapsack problem. *Neurocomputing*, volume 146(0):17 – 29, 2014.
- [PATVARDHAN ET AL. 2014] Patvardhan, C., Bansal, S. , Srivastav, A. Balanced Quantum-Inspired Evolutionary Algorithm for Multiple Knapsack Problem. *International Journal of Intelligent Systems and Applications*, volume 6(11):1–12, 2014.
- [PIRKUL 1987] Pirkul, H. A heuristic solution procedure for the multiconstraint zero-one knapsack problem. *Naval Research Logistics (NRL)*, volume 34(2):161–172, 1987.
- [PISINGER 1995] Pisinger, D. Algorithms for knapsack problems. Tese de Doutorado, University of Copenhagen, 1995.
- [PISINGER 2005] Pisinger, D. Where are the hard knapsack problems. *Computers & Operations Research*, volume 32(9):2271–2284, 2005.
- [PUCHINGER ET AL. 2006] Puchinger, J., Raidl, G. R. , Pferschy, U. The core concept for the multidimensional knapsack problem. Em J. Gottlieb , G. R. Raidl, editores, *Evolutionary Computation in Combinatorial Optimization*, pp. 195–208. Springer Berlin Heidelberg, 2006.
- [VASQUEZ , HAO 2001] Vasquez, M. , Hao, J. A Hybrid Approach for the 01 Multidimensional Knapsack problem. Em *International Joint Conference on Artificial Intelligence*, pp. 328–333. 2001.
- [VASQUEZ , VIMONT 2005] Vasquez, M. , Vimont, Y. Improved results on the 0-1 multidimensional knapsack problem. *European Journal of Operational Research*, volume 165:70–81, 2005.
- [WANG ET AL. 2013] Wang, L., Zheng, X. L. , Wang, S. Y. A novel binary fruit fly optimization algorithm for solving the multidimensional knapsack problem. *Knowledge-Based Systems*, volume 48(0):17 – 23, 2013.

APÊNDICE A – Tabelas dos Resultados Experimentais

As tabelas a seguir mostram, para cada nível de correlação, os resultados detalhados dos experimentos. As colunas Y (anos), A (ações), R (recursos) descrevem a configuração das instâncias, e cada linha representa 100 instâncias de uma dada configuração.

A coluna '# Int.' mostra, entre as 100 instâncias com a configuração, quantas foram interrompidas durante a execução do CPLEX por chegar ao limite de tempo (20 minutos). As seis últimas colunas mostram, para cada heurística, a razão mínima, máxima e média entre a solução da heurística e a melhor solução conhecida.

Tabela A.1: Resultados Experimentais para instâncias com forte correlação ($\alpha = 0$)

Y	A	R	# Int.	AGRLP			BTRLP		
				Min.	Max.	Méd.	Min.	Máx.	Méd.
3	25	1	1	0.99438	0.99973	0.99834	0.99931	0.99998	0.99971
3	25	2	43	0.99426	0.99966	0.99794	0.99904	0.99985	0.99957
3	25	4	100	0.99230	0.99871	0.99676	0.99888	0.99981	0.99932
3	50	1	0	0.99819	0.99992	0.99922	0.99960	0.99998	0.99983
3	50	2	0	0.99705	0.99981	0.99893	0.99951	0.99991	0.99976
3	50	4	97	0.99597	0.99951	0.99851	0.99937	0.99985	0.99963
3	100	1	0	0.99912	1.0	0.99965	0.99972	1.0	0.99993
3	100	2	0	0.99893	0.99992	0.99957	0.99978	1.0	0.99991
3	100	4	0	0.99831	0.99981	0.99929	0.99967	0.99991	0.99982
4	25	1	1	0.99683	0.99970	0.99847	0.99928	0.99996	0.99970
4	25	2	75	0.99637	0.99918	0.99807	0.99912	0.99981	0.99957
4	25	4	100	0.99411	0.99866	0.99688	0.99895	0.99974	0.99931
4	50	1	0	0.99788	0.99979	0.99920	0.99951	0.99999	0.99983

Continua ...

Tabela A.1: Resultados Experimentais para instâncias com forte correlação ($\alpha = 0$) (Continuação)

Y	A	R	# Int.	AGRLP			BTRLP		
				Min.	Max.	Méd.	Min.	Máx.	Méd.
4	50	2	2	0.99819	0.99970	0.99911	0.99960	1.0	0.99979
4	50	4	100	0.99606	0.99934	0.99847	0.99940	0.99981	0.99963
4	100	1	0	0.99915	0.99998	0.99967	0.99982	1.0	0.99993
4	100	2	0	0.99899	0.99995	0.99958	0.99982	0.99998	0.99991
4	100	4	3	0.99856	0.99965	0.99931	0.99973	0.99991	0.99983
5	25	1	3	0.99676	0.99971	0.99848	0.99933	0.99997	0.99973
5	25	2	92	0.99575	0.99932	0.99804	0.99931	0.99993	0.99960
5	25	4	100	0.99444	0.99842	0.99693	0.99883	0.99968	0.99933
5	50	1	0	0.99832	0.99993	0.99927	0.99965	0.99997	0.99985
5	50	2	8	0.99748	0.99978	0.99907	0.99959	0.99994	0.99979
5	50	4	100	0.99658	0.99920	0.99854	0.99952	0.99980	0.99966
5	100	1	0	0.99904	1.0	0.99969	0.99974	1.0	0.99994
5	100	2	0	0.99884	0.99987	0.99959	0.99981	0.99998	0.99992
5	100	4	8	0.99859	0.99977	0.99931	0.99974	0.99992	0.99984
6	25	1	5	0.99633	0.99944	0.99839	0.99921	0.99991	0.99971
6	25	2	92	0.99617	0.99923	0.99800	0.99921	0.99980	0.99959
6	25	4	100	0.99147	0.99873	0.99694	0.99867	0.99970	0.99934
6	50	1	0	0.99838	0.99977	0.99933	0.99962	1.0	0.99986
6	50	2	7	0.99840	0.99969	0.99915	0.99968	0.99993	0.99981
6	50	4	100	0.99692	0.99940	0.99859	0.99949	0.99983	0.99966
6	100	1	0	0.99935	0.99993	0.99973	0.99979	1.0	0.99994
6	100	2	0	0.99909	0.99993	0.99960	0.99984	1.0	0.99992
6	100	4	12	0.99880	0.99960	0.99932	0.99975	0.99991	0.99984

Tabela A.2: Resultados Experimentais para instâncias com fraca correlação ($\alpha = 0.1$)

Y	A	R	# Int.	AGRLP			BTRLP		
				Min.	Max.	Méd.	Min.	Máx.	Méd.
3	25	1	0	0.99704	0.99982	0.99861	0.99895	0.99990	0.99955
3	25	2	25	0.99459	0.99937	0.99815	0.99851	0.99969	0.99913

Continua ...

Tabela A.2: Resultados Experimentais para instâncias com fraca correlação ($\alpha = 0.1$) (Continuação)

<i>Y</i>	<i>A</i>	<i>R</i>	# Int.	AGRLP			BTRLP		
				Min.	Max.	Méd.	Min.	Máx.	Méd.
3	25	4	98	0.99053	0.99919	0.99703	0.99784	0.99939	0.99861
3	50	1	0	0.99864	1.0	0.99937	0.99919	1.0	0.99972
3	50	2	5	0.99782	0.99972	0.99908	0.99884	0.99977	0.99945
3	50	4	100	0.99615	0.99940	0.99863	0.99860	0.99950	0.99910
3	100	1	0	0.99933	0.99997	0.99972	0.99967	1.0	0.99985
3	100	2	0	0.99893	0.99994	0.99961	0.99944	0.99992	0.99974
3	100	4	58	0.99809	0.99967	0.99923	0.99904	0.99976	0.99944
4	25	1	0	0.99689	0.99950	0.99853	0.99899	0.99988	0.99946
4	25	2	65	0.99496	0.99972	0.99821	0.99828	0.99979	0.99909
4	25	4	100	0.99325	0.99847	0.99705	0.99763	0.99944	0.99864
4	50	1	0	0.99859	0.99984	0.99935	0.99940	0.99992	0.99972
4	50	2	20	0.99713	0.99967	0.99903	0.99879	0.99978	0.99944
4	50	4	100	0.99615	0.99941	0.99857	0.99837	0.99953	0.99907
4	100	1	0	0.99932	1.0	0.99974	0.99953	1.0	0.99985
4	100	2	0	0.99914	0.99986	0.99961	0.99953	0.99997	0.99976
4	100	4	88	0.99833	0.99965	0.99927	0.99926	0.99970	0.99948
5	25	1	1	0.99633	0.99969	0.99852	0.99880	0.99984	0.99948
5	25	2	85	0.99638	0.99955	0.99817	0.99845	0.99958	0.99911
5	25	4	100	0.99468	0.99871	0.99724	0.99781	0.99929	0.99855
5	50	1	0	0.99875	0.99982	0.99940	0.99942	0.99990	0.99970
5	50	2	36	0.99800	0.99959	0.99908	0.99910	0.99974	0.99948
5	50	4	100	0.99632	0.99938	0.99856	0.99874	0.99957	0.99915
5	100	1	0	0.99952	0.99997	0.99975	0.99967	0.99997	0.99987
5	100	2	0	0.99904	0.99986	0.99959	0.99951	0.99990	0.99975
5	100	4	98	0.99864	0.99967	0.99927	0.99915	0.99973	0.99950
6	25	1	0	0.99476	0.99949	0.99858	0.99908	0.99986	0.99947
6	25	2	97	0.99596	0.99919	0.99822	0.99845	0.99965	0.99916
6	25	4	100	0.99396	0.99876	0.99720	0.99758	0.99934	0.99853
6	50	1	0	0.99863	0.99985	0.99941	0.99955	0.99994	0.99973
6	50	2	51	0.99840	0.99971	0.99920	0.99924	0.99976	0.99952
6	50	4	100	0.99701	0.99925	0.99866	0.99873	0.99955	0.99921

Continua ...

Tabela A.2: Resultados Experimentais para instâncias com fraca correlação ($\alpha = 0.1$) (Continuação)

<i>Y</i>	<i>A</i>	<i>R</i>	# Int.	AGRLP			BTRLP		
				Min.	Max.	Méd.	Min.	Máx.	Méd.
6	100	1	0	0.99938	0.99998	0.99975	0.99967	1.0	0.99988
6	100	2	0	0.99925	0.99984	0.99961	0.99961	0.99990	0.99976
6	100	4	100	0.99866	0.99966	0.99932	0.99928	0.99971	0.99952

Tabela A.3: Resultados Experimentais para instâncias sem correlação ($\alpha = 1$)

<i>Y</i>	<i>A</i>	<i>R</i>	# Int.	AGRLP			BTRLP		
				Min.	Max.	Méd.	Min.	Máx.	Méd.
3	25	1	0	0.99832	1.0	0.99971	0.99886	1.0	0.99990
3	25	2	0	0.99739	1.0	0.99959	0.99923	1.0	0.99985
3	25	4	0	0.99370	1.0	0.99925	0.99710	1.0	0.99972
3	50	1	0	0.99867	1.0	0.99985	0.99955	1.0	0.99995
3	50	2	0	0.99911	1.0	0.99983	0.99970	1.0	0.99993
3	50	4	0	0.99813	1.0	0.99957	0.99881	1.0	0.99980
3	100	1	0	0.99972	1.0	0.99993	0.99975	1.0	0.99997
3	100	2	0	0.99929	1.0	0.99988	0.99963	1.0	0.99994
3	100	4	0	0.99930	1.0	0.99982	0.99945	1.0	0.99991
4	25	1	0	0.99616	1.0	0.99975	0.99923	1.0	0.99993
4	25	2	0	0.99734	1.0	0.99952	0.99931	1.0	0.99982
4	25	4	0	0.99606	1.0	0.99919	0.99895	1.0	0.99967
4	50	1	0	0.99928	1.0	0.99986	0.99967	1.0	0.99994
4	50	2	0	0.99914	1.0	0.99980	0.99932	1.0	0.99989
4	50	4	0	0.99876	0.99999	0.99968	0.99923	1.0	0.99981
4	100	1	0	0.99951	1.0	0.99994	0.99982	1.0	0.99998
4	100	2	0	0.99951	1.0	0.99989	0.99967	1.0	0.99995
4	100	4	0	0.99927	1.0	0.99979	0.99937	1.0	0.99987
5	25	1	0	0.99816	1.0	0.99978	0.99945	1.0	0.99993
5	25	2	0	0.99810	1.0	0.99958	0.99921	1.0	0.99981
5	25	4	3	0.99512	0.99998	0.99927	0.99822	1.0	0.99963
5	50	1	0	0.99918	1.0	0.99990	0.99974	1.0	0.99996

Continua ...

Tabela A.3: Resultados Experimentais para instâncias sem correlação ($\alpha = 1$) (Continuação)

Y	A	R	# Int.	AGRLP			BTRLP		
				Min.	Max.	Méd.	Min.	Máx.	Méd.
5	50	2	0	0.99932	1.0	0.99980	0.99959	1.0	0.99989
5	50	4	2	0.99861	0.99995	0.99961	0.99913	0.99999	0.99975
5	100	1	0	0.99967	1.0	0.99994	0.99985	1.0	0.99997
5	100	2	0	0.99949	1.0	0.99987	0.99970	1.0	0.99992
5	100	4	0	0.99930	1.0	0.99982	0.99955	1.0	0.99986
6	25	1	0	0.99920	1.0	0.99980	0.99928	1.0	0.99991
6	25	2	1	0.99827	1.0	0.99954	0.99904	1.0	0.99979
6	25	4	6	0.99673	1.0	0.99930	0.99837	1.0	0.99960
6	50	1	0	0.99960	1.0	0.99992	0.99965	1.0	0.99996
6	50	2	0	0.99860	1.0	0.99979	0.99932	1.0	0.99987
6	50	4	6	0.99896	0.99995	0.99962	0.99923	0.99999	0.99975
6	100	1	0	0.99957	1.0	0.99994	0.99987	1.0	0.99997
6	100	2	0	0.99950	1.0	0.99987	0.99966	1.0	0.99993
6	100	4	0	0.99956	1.0	0.99982	0.99960	1.0	0.99986