

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA E
INFORMÁTICA INDUSTRIAL

JONAS KRAUSE

**PROGRAMAÇÃO MATEMÁTICA E EVOLUÇÃO DIFERENCIAL
PARA A OTIMIZAÇÃO DE REDES DE DUTOS**

DISSERTAÇÃO

CURITIBA

2014

JONAS KRAUSE

**PROGRAMAÇÃO MATEMÁTICA E EVOLUÇÃO DIFERENCIAL
PARA A OTIMIZAÇÃO DE REDES DE DUTOS**

Dissertação apresentada ao Programa de Pós-graduação em Engenharia Elétrica e Informática Industrial da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do grau de “Mestre em Ciências” – Área de Concentração: Engenharia de Computação.

Orientador: Heitor Silvério Lopes, Dr

Co-orientador: Lauro César Galvão, Dr

CURITIBA

2014

Dados Internacionais de Catalogação na Publicação

- K 91 Krause, Jonas
Programação matemática e evolução diferencial para otimização de redes de dutos / Jonas Krause. – 2014.
77 f. : il. ; 30 cm
- Orientador: Heitor Silvério Lopes.
Coorientador: Lauro César Galvão.
Dissertação (Mestrado) – Universidade Tecnológica Federal do Paraná. Programa de Pós-graduação em Engenharia Elétrica e Informática Industrial. Curitiba, 2014.
Bibliografia: f. 53-55.
1. Otimização combinatória. 2. Oleodutos. 3. Programação linear. 4. Programação inteira. 5. Programação (Matemática.). 6. Computação evolutiva. 7. Engenharia elétrica – Dissertações. I. Lopes, Heitor Silvério, orient. II. Galvão, Lauro César, coorient. III. Universidade Tecnológica Federal do Paraná. Programa de Pós-graduação em Engenharia Elétrica e Informática Industrial. IV. Título.

CDD (22. ed.) 621.3

Biblioteca Central da UTFPR, Campus Curitiba

Título da Dissertação Nº. 648

“Programação Matemática e Evolução Diferencial para a Otimização de Redes de Dutos.”

por

Jonas Krause

Orientador: Prof. Dr. Heitor Silvério Lopes

Esta dissertação foi apresentada como requisito parcial à obtenção do grau de MESTRE EM CIÊNCIAS – Área de Concentração: Engenharia de Computação do Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial – CPGEI – da Universidade Tecnológica Federal do Paraná – UTFPR, às 14h do dia 16 de dezembro de 2013. O trabalho foi aprovado pela Banca Examinadora, composta pelos professores doutores:

Prof. Dr. Leandro Magatão
(Presidente – UTFPR)

Prof. Dr. Eduardo Gontijo Carrano
(UFMG)

Prof. Dr. Lauro César Galvão
(UTFPR)

Visto da coordenação:

Prof. Ricardo Lüders, Dr.
(Coordenador do CPGEI)

AGRADECIMENTOS

Gostaria de agradecer a todos aqueles que de alguma forma contribuíram para a conclusão deste trabalho.

Aos professores Dr. Heitor Silvério Lopes e Dr. Lauro Cesar Galvão pelas oportunidades e orientações.

Aos colegas do laboratório BIOINFO, em especial aos colegas Rafael Stubs Parpinelli e Hugo Alberto Perlin.

Agradeço a minha esposa, Eliane Teliz Carnero, que me apoiou e me incentivou em todos os momentos.

RESUMO

KRAUSE, Jonas. PROGRAMAÇÃO MATEMÁTICA E EVOLUÇÃO DIFERENCIAL PARA A OTIMIZAÇÃO DE REDES DE DUTOS. 78 f. Dissertação – Programa de Pós-graduação em Engenharia Elétrica e Informática Industrial, Universidade Tecnológica Federal do Paraná. Curitiba, 2014.

A otimização de uma rede de transporte de derivados de petróleo é um problema complexo e abordado na literatura atual. A modelagem matemática deste problema proposta neste trabalho cria um problema de otimização combinatorial. Métodos de resolução deste problema através da programação linear inteira mista e de algoritmos heurísticos de evolução diferencial (Evolução Diferencial Binária e Evolução Diferencial Discretizada) são propostos utilizando variáveis binárias. Os resultados encontrados com a programação linear apresentam valores ótimos para os *benchmarks* com pequenos espaços de busca e valores sub-ótimos para grandes. Resultados utilizando a evolução diferencial também são apresentados como uma alternativa de baixo esforço computacional. A aplicação destes métodos proporciona alternativas para o transporte de diferentes produtos em um horizonte de tempo definido e compara os métodos heurísticos com codificações binárias e contínuas. Tais resultados incentivam a utilização de algoritmos heurísticos com codificação contínua e apontam os métodos de discretização como alternativas eficazes para a resolução de problemas discretos.

Palavras-chave: Otimização Combinatorial, Oleodutos, Programação Linear Inteira Mista, Evolução Diferencial

ABSTRACT

KRAUSE, Jonas. MATHEMATICAL PROGRAMMING AND DIFFERENTIAL EVOLUTION FOR THE OPTIMIZATION OF PIPELINE NETWORKS. 78 f. Dissertação – Programa de Pós-graduação em Engenharia Elétrica e Informática Industrial, Universidade Tecnológica Federal do Paraná. Curitiba, 2014.

The optimization of an pipeline network is a complex problem and addressed in the current literature. The mathematical modeling of this problem proposed in this paper creates a problem of combinatorial optimization. Methods for solving this problem using linear mixed integer programming and heuristic algorithms of differential evolution (Binary Differential Evolution and Discretized Differential Evolution) are proposed using binary variables. The results obtained with the linear programming have optimal values for the benchmarks with small search spaces and sub-optimal for large values. Results using the differential evolution are also presented as an alternative low computational effort. The application of these methods provides alternatives for transporting different products in a defined time horizon and compare heuristic methods with continuous and binary encodings. Such results encourage the use of heuristic algorithms with continuous coding and the point discretization methods as effective for solving problems discrete alternatives.

Keywords: Combinatorial Optimization, Oil Pipeline, Mixed Integer Linear Programming, Differential Evolution

LISTA DE FIGURAS

FIGURA 1	– Etapas do algoritmo de ED	21
FIGURA 2	– Ilustração de uma mutação simples da ED em um espaço 2-D	24
FIGURA 3	– Exemplo do processo de <i>crossover</i> binomial	25
FIGURA 4	– Pseudocódigo da EDB	26
FIGURA 5	– Pseudocódigo da EDD	27
FIGURA 6	– Rede de Polidutos no Brasil	31
FIGURA 7	– Rede de Polidutos na região sudeste	32
FIGURA 8	– Exemplo da disposição dos dutos discretizados	34
FIGURA 9	– Rede simplificada	36
FIGURA 10	– Rede simplificada estruturada pelos pesos P_d	38
FIGURA 11	– Melhor solução para instância 03	48
FIGURA 12	– Melhor solução para instância 14	49
FIGURA 13	– Boxplot - resultados BDE e DDE	50
FIGURA 14	– Boxplot - resultados BDE e DDE	50
FIGURA 15	– Melhor solução para instância 01	58
FIGURA 16	– Melhor solução para instância 02	59
FIGURA 17	– Melhor solução para instância 04	60
FIGURA 18	– Melhor solução para instância 05	61
FIGURA 19	– Melhor solução para instância 06	62
FIGURA 20	– Melhor solução para instância 07	63
FIGURA 21	– Melhor solução para instância 08	64
FIGURA 22	– Melhor solução para instância 09	65
FIGURA 23	– Melhor solução para instância 10	66
FIGURA 24	– Melhor solução para instância 11	67
FIGURA 25	– Melhor solução para instância 12	68
FIGURA 26	– Melhor solução para instância 13	69
FIGURA 27	– Melhor solução para instância 15	70
FIGURA 28	– Boxplot - resultados BDE e DDE	72
FIGURA 29	– Boxplot - resultados BDE e DDE	72
FIGURA 30	– Boxplot - resultados BDE e DDE	73
FIGURA 31	– Boxplot - resultados BDE e DDE	73
FIGURA 32	– Boxplot - resultados BDE e DDE	74
FIGURA 33	– Boxplot - resultados BDE e DDE	74
FIGURA 34	– Boxplot - resultados BDE e DDE	75
FIGURA 35	– Boxplot - resultados BDE e DDE	75
FIGURA 36	– Boxplot - resultados BDE e DDE	77
FIGURA 37	– Boxplot - resultados BDE e DDE	77
FIGURA 38	– Boxplot - resultados BDE e DDE	78
FIGURA 39	– Boxplot - resultados BDE e DDE	78

LISTA DE TABELAS

TABELA 1	– Lista de <i>solvers</i>	18
TABELA 2	– Lista de trabalhos correlatos	29
TABELA 3	– Pesos de cada duto (WESTPHAL et al., 2011)	37
TABELA 4	– <i>Benchmarks</i>	43
TABELA 5	– <i>Parâmetros</i>	45
TABELA 6	– Resultados da EDB e EDD	46

LISTA DE SIGLAS

CE	Computação Evolutiva
PLIM	Programação Linear Inteira Mista
EDB	Evolução Diferencial Binária
EDD	Evolução Diferencial Discretizada
PM	Programação Matemática
ED	Evolução Diferencial
PM	Taxa de mutação
CR	Taxa de <i>crossover</i>
PR	Taxa de perturbação
AEM	Algoritmo Evolutivo Multiobjetivo
NV	Número de Variáveis
EB	Espaço de Busca

LISTA DE SÍMBOLOS

r	Quantidade de refinarias
a	Quantidade de tanques de armazenamento
c	Quantidade de clientes finais
D	Dimensão do problema
G	Número de gerações
H	Horizonte de Tempo
D	Número de dutos
W	Peso do duto
P	Número de produtos
k	Período de tempo
Z	Função objetivo
Q_c	Quantidade de entrega

SUMÁRIO

1 INTRODUÇÃO	13
1.1 MOTIVAÇÃO	14
1.2 OBJETIVOS	15
1.2.1 Objetivo Geral	15
1.2.2 Objetivos Específicos	15
1.3 ORGANIZAÇÃO DA DISSERTAÇÃO	15
2 FUNDAMENTAÇÃO TEÓRICA	17
2.1 PROGRAMAÇÃO MATEMÁTICA	17
2.1.1 <i>Solvers</i>	18
2.1.1.1 <i>SIMPLEX</i>	18
2.1.1.2 <i>Branch-and-Bound</i>	19
2.2 META-HEURÍSTICAS	19
2.2.1 Algoritmos Populacionais	20
2.2.1.1 Algoritmos de Inteligência de Enxames	20
2.2.1.2 Algoritmos Evolutivos	21
2.3 EVOLUÇÃO DIFERENCIAL	21
2.3.1 Seleção	23
2.3.2 Mutação	23
2.3.3 <i>Crossover</i>	24
2.4 EVOLUÇÃO DIFERENCIAL BINÁRIA	25
2.5 EVOLUÇÃO DIFERENCIAL DISCRETIZADA	26
2.6 TRABALHOS CORRELATOS	28
3 METODOLOGIA	30
3.1 DEFINIÇÃO DO PROBLEMA	30
3.2 MODELAGEM MATEMÁTICA DO PROBLEMA	31
3.3 DISCRETIZAÇÃO	33
3.4 ANÁLISE DE COMPLEXIDADE	35
3.5 MODELO SIMPLIFICADO	36
3.5.1 Função Objetivo	38
3.5.2 Restrições	39
4 EXPERIMENTOS E RESULTADOS	42
4.1 PROGRAMAÇÃO LINEAR INTEIRA MISTA	42
4.2 EVOLUÇÃO DIFERENCIAL	44
5 CONCLUSÕES FINAIS	51
5.0.1 Trabalhos futuros	52
REFERÊNCIAS	54
Apêndice A – SOLUÇÕES ÓTIMAS (01, 02, 04, 05 E 06) E SUB-ÓTIMAS (07, 08, 09, 10, 11, 12, 13 E 15) DA PLIM	57
Apêndice B – BOXPLOTS (01, 02, 03, 08, 09, 10, 11 E 13) - DIFERENÇA SIGNIFICATIVA	71
Apêndice C – BOXPLOTS (04, 05, 06 E 07) - DIFERENÇA PODE NÃO SER SIGNIFICATIVA	71

FICATIVA	76
-----------------------	-----------

1 INTRODUÇÃO

O problema de otimização de uma rede de distribuição de derivados de petróleo é um problema de escalonamento (*scheduling*) de alta complexidade. Dependendo de sua estrutura, a rede pode conter dezenas de dutos interligando cada uma das refinarias e tanques de armazenamento. Encontrar a melhor combinação de bombeamento de cada produto no menor tempo possível otimiza a utilização da rede. Diversos métodos podem ser empregados para as soluções deste problema. Um dos métodos mais difundidos na literatura é a programação linear inteira mista. Este método utiliza equações lineares para representar os objetivos e restrições reais, buscando a melhor combinação para a solução do problema.

Métodos heurísticos bioinspirados também são opções viáveis, principalmente em problemas com um grande número de variáveis e que exigem um enorme esforço computacional. A inspiração na natureza para a resolução de problemas complexos tem sido utilizada desde os anos 50 (FRASER, 1957) e seus benefícios são inquestionáveis. Um dos processos mais utilizados para encontrar possíveis soluções de um problema é a simulação da evolução dos seres vivos através de gerações. A ideia inicial baseia-se na teoria de Darwin e utiliza a evolução de uma população de soluções a partir de sua implementação em computador (HOLLAND, 1975). Outros comportamentos observados na natureza também inspiraram a criação de algoritmos que simulam insetos sociais, táticas de caça e técnicas de localização. Estes conceitos são atualmente empregados em diversas disciplinas de ciências naturais, engenharias e ciência da computação.

As metodologias bioinspiradas buscam soluções ótimas ou sub-ótimas no espaço de soluções candidatas, enquanto gerenciam operadores computacionais de busca local e global. Tais algoritmos são conhecidos como meta-heurísticas e podem ser definidos como um procedimento iterativo de busca inspirado em mecanismos biológicos. A designação meta-heurística é dada ao método quando este algoritmo heurístico é aplicado de forma genérica em problemas de otimização. As meta-heurísticas baseadas na teoria de Darwin geralmente são associadas à evolução de uma população através de gerações, buscando um indivíduo mais adaptável ao problema proposto. Estes algoritmos são exemplos de algoritmos populacionais e utili-

zam indivíduos iniciais aleatórios a serem evoluídos pelos processos de mutação e reprodução com herança genética. A linha de pesquisa que trata destes algoritmos é conhecida como Computação Evolutiva (CE) e abrange os estudos de estratégias evolutivas, sistemas classificadores, programação genética, programação evolutiva, algoritmos genéticos e diversos outros algoritmos populacionais como a evolução diferencial.

A principal vantagem da CE está na possibilidade de encontrar soluções boas para problemas complexos (WESTPHAL et al., 2011). Com os objetivos e restrições matematicamente definidos, a CE utiliza procedimentos genéricos e flexíveis para buscar uma solução de boa qualidade mesmo sem garantir a obtenção da solução ótima. Tais procedimentos podem ser adaptados para uma ampla gama de problemas, fornecendo robustez e flexibilidade ao método. Sendo assim, a CE pode ser entendida como um conjunto de técnicas a serem aplicadas na solução de problemas complexos para os quais outras técnicas conhecidas são inviáveis ou dificilmente aplicáveis.

Os problemas de otimização são os que mais se beneficiaram com o uso de técnicas da CE. Muitos problemas de engenharia, por exemplo, podem ser adequadamente apresentados como problemas de otimização (MICHALEWICZ; FOGEL, 2004). Tais problemas geralmente são de alta complexidade, não podendo ser resolvidos em tempo polinomial. A aplicação de meta-heurísticas para a resolução destes problemas de otimização apresenta um novo paradigma, pois se abre mão da garantia de obtenção da solução ótima para se conquistar boas soluções com um esforço computacional menor. Um problema de otimização bastante difundido na literatura é o problema de *scheduling* (agendamento de recursos) e este pode ser modelado como um problema combinatorial. Sendo assim, este trabalho foca na otimização do problema de *scheduling* de redes de distribuição de derivados de petróleo. Este problema é um exemplo clássico de alta complexidade e serve de ponto de partida para a descrição e formalização dos conceitos e técnicas de CE aqui apresentadas.

1.1 MOTIVAÇÃO

A otimização de uma rede de dutos de derivados de petróleo está diretamente ligada a uma grande quantidade de tomadas de decisões. Uma rede de distribuição é composta por refinarias, terminais e parques de armazenagem interligados por dutos. As atividades e operações associadas a uma rede de distribuição normalmente combinam operações contínuas com a determinação de alocação de recursos para as várias tarefas envolvidas na produção (KONDILI et al., 1993). A melhor alocação destes multi-recursos resulta em uma otimização da lucratividade e na minimização dos custos. Vários são os fatores a serem considerados nas diretrizes de

produção: alterações nas demandas, especificações dos produtos, datas de entrega, qualidade e quantidade dos produtos, disponibilidade e desempenho das unidades de processamento. Tais variáveis fazem parte de um sistema de controle da refinaria que deve ser o mais rápido e eficiente possível para a otimização de toda a estrutura de polidutos. Na literatura, trabalhos de Souza et al. (2010) e de Westphal et al. (2011) abordam este mesmo problema. Tais trabalhos apresentam estudos de casos específicos e soluções sub-ótimas, porém não apresentam suas respectivas soluções ótimas. Uma nova modelagem matemática binária para este problema, a criação de (*benchmarks*) utilizando programação matemática e o uso de meta-heurísticas baseadas na evolução diferencial são as principais motivações deste trabalho.

1.2 OBJETIVOS

1.2.1 OBJETIVO GERAL

Criação de um conjunto de práticas, funções e restrições lineares com diferentes produtos em uma rede simplificada de distribuição de derivados de petróleo e encontrar soluções adequadas os *benchmarks* propostos.

1.2.2 OBJETIVOS ESPECÍFICOS

- Modelar matematicamente o problema para r refinarias, a tanques de armazenamento e c clientes finais.
- Utilizar a Programação Linear Inteira Mista (PLIM) como método exato e encontrar as soluções ótimas de cada *benchmark* proposto.
- Utilizar as meta-heurísticas de Evolução Diferencial Binária (EDB) e a Evolução Diferencial Discretizada (EDD) como método alternativo de baixo esforço computacional.
- Comparar resultados encontrados pela PLIM e os métodos heurísticos de EDB e EDD.

1.3 ORGANIZAÇÃO DA DISSERTAÇÃO

Este trabalho aborda a criação de técnicas de otimização para problemas de *scheduling* de derivados do petróleo. A modelagem matemática e o modelo simplificado do problema possui todas as características de um modelo real e pode ser aplicado à infra-estruturas do país inteiro. O algoritmo de evolução diferencial é o método heurístico populacional utilizado para buscar soluções de boa qualidade e rápidas.

O Capítulo 2 apresenta o método de busca da solução ótima utilizando PLIM, as diversas meta-heurísticas utilizadas neste tipo de problema e uma revisão da literatura sobre problemas de otimização de *scheduling* de polidutos na indústria do petróleo.

O Capítulo 3 descreve o desenvolvimento da solução proposta, discorrendo o problema e analisando sua complexidade. Esta metodologia utiliza uma modelagem matemática discreta, um método de busca exaustiva (PLIM) e algoritmos inspirados na evolução diferencial (EDB e EDD).

No Capítulo 4 são apresentados os resultados alcançados pelos algoritmos de evolução diferencial e comparados com soluções ótimas e sub-ótimas encontradas pela programação matemática.

No Capítulo 5 é apresentada a discussão dos resultados, as conclusões finais do trabalho e propostas para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

A modelagem matemática discreta de problemas reais utilizando a PLIM é um dos métodos mais utilizados na literatura atualmente. Porém, o tempo de resolução de modelos discretos pode ser muito elevado. De acordo com Schrage (2000), o tempo médio de processamento da programação linear está diretamente relacionado ao número de variáveis e ao quadrado do número de restrições do modelo. Métodos bioinspirados são uma alternativa com um baixo esforço computacional para para modelos complexos e dinâmicos com muitas variáveis.

A inspiração na natureza é uma tendência em pesquisas com modelos computacionais inteligentes (ZONTA et al., 2005). Para cientistas da computação, matemáticos e engenheiros, os modelos heurísticos buscam soluções de boa qualidade para problemas complexos reais. Mesmo sem a garantia de encontrar a solução ótima, estes métodos geralmente encontram soluções boas em um tempo computacional razoável. Este Capítulo apresenta um breve resumo sobre programação matemática, as meta-heurísticas baseadas em evolução diferencial e os diversos trabalhos correlatos ao problema de *scheduling* da rede de polidutos.

2.1 PROGRAMAÇÃO MATEMÁTICA

A Programação Matemática (PM) consiste na formulação de um modelo similar aos procedimentos e variáveis reais, empregando símbolos matemáticos para representar suas relações. O planejamento da PM engloba técnicas e algoritmos para elaborar e solucionar modelos expressos matematicamente (PUCCINI, 1989). O modelos de PM também podem ser entendidos como um conjunto de equações, inequações e dependências lógicas que representam relacionamentos em estruturas reais. Estes relacionamentos são representados por variáveis de decisão, parâmetros, restrições e a função objetivo. Esta função corresponde a uma função matemática que avalia a qualidade da solução obtida através das variáveis de decisão empregadas, geralmente está associada a uma maximização ou minimização.

As variáveis utilizadas na PM podem representar situações reais de tomada de decisão (geralmente associada a variáveis binárias), problemas combinatórios (variáveis inteiras)

e situações reais com valores contínuos (variáveis reais contínuas). A vertente mais utilizada na PM é a Programação Linear, pois utiliza variáveis contínuas e apresentam comportamento linear, tanto nas restrições quanto na função objetivo (GOLDBARG; LUNA, 2000). Análogamente, a Programação Binária e Programação Inteira utilizam somente variáveis binárias e inteiras.

Outra vertente muito utilizada na literatura é a Programação Linear Inteira Mista. Esta utiliza variáveis contínuas, inteiras e binárias. Sendo assim, as variáveis inteiras são empregadas para representar condições de indivisibilidade e as binárias para tomadas de decisões. Todas estas variantes da PM requerem um grande esforço computacional quando o problema proposto possui um grande espaço de busca, pois seu tempo de processamento está diretamente ligado à quantidade de variáveis do problema. A PLIM foi utilizada por este trabalho para encontrar a solução ótima dos *benchmarks* propostos.

2.1.1 SOLVERS

São *softwares* que repetem seguidamente um procedimento sistemático até a resolução do problema. Atualmente várias empresas disponibilizam *solvers* no mercado, a Tabela 1 apresenta uma pequena lista de desenvolvedores, seus respectivos *solvers* e classifica os *softwares* em comerciais ou livres.

Tabela 1: Lista de *solvers*

Desenvolvedor	<i>Solver</i>	Licença
IBM	IBM-CPLEX	Comercial
Fico Corporation	FICO-XPress	Comercial
Lindo Systems	LINDO/LINGO	Comercial
Gurobi Optimization	Gurobi	Comercial
Microsoft	Solver - Excel	Comercial
OpenOffice.org	Solver - Calc	Livre
LGPL (Lesser General Pulic License)	LP_Solve	Livre
GLPK (GNU Linear Programming Kit)	GLPK	Livre

Este trabalho optou por utilizar o LP_Solve como *software* de otimização por ser um *solver* livre e utilizar a PLIM. Este *solver* utiliza não somente o método SIMPLEX mas também é baseado no algoritmo de *branch-and-bound*.

2.1.1.1 SIMPLEX

Os modelos matemáticos que representam sistemas reais podem englobar milhares de variáveis. Sendo assim, criou-se um método (algoritmo) baseado em repetições e ciclos para

soluções destes modelos. Este algoritmo iterativo foi formalizado por Dantzig (1963) como o método SIMPLEX e tem sido utilizado desde então para encontrar a solução ótima de modelos de programação linear. O SIMPLEX é considerado como um dos mais significativos avanços na matemática do século XX.

Este método é formado por um grupo de critérios para escolha de soluções básicas que melhorem o desempenho do modelo. Para isso, o problema deve apresentar uma solução básica inicial. As soluções básicas subsequentes são calculadas com a troca de variáveis básicas por não básicas, gerando novas soluções. Os critérios para escolha de vetores e conseqüentemente das variáveis que entram e saem para a formação da nova base constituem o centro do método SIMPLEX.

Entretanto, o método esbarrava numa grande necessidade computacional que, até então, era muito escassa. Nas últimas décadas houve uma grande evolução no *hardware* e *software* dos computadores e a aplicação do SIMPLEX vem se tornando mais acessível para usuários de computadores domésticos.

2.1.1.2 *BRANCH-AND-BOUND*

Este algoritmo (BB ou B&B) foi inicialmente proposto por Land e Doig (1960) e consiste em dividir para conquistar. Tal método enumera sistematicamente as soluções candidatas e descarta grandes subconjuntos de soluções ineficazes. Para isso, são utilizados limites superiores e inferiores estimados da quantidade de soluções sendo otimizadas. O método de *branch-and-bound* é um algoritmo utilizado para encontrar as melhores soluções de vários problemas de otimização, especialmente otimização discreta e combinatorial.

2.2 META-HEURÍSTICAS

Os procedimentos heurísticos e meta-heurísticos propõem soluções alternativas às modelagens puramente matemáticas (LOPES et al., 2013). Tais procedimentos são geralmente aplicados a problemas de pesquisa operacional, uma área interdisciplinar do conhecimento voltada ao desenvolvimento de modelos matemáticos e algoritmos para a resolução de problemas complexos reais. A busca pela solução exata em problemas reais com um grande número de variáveis e restrições requer um enorme esforço computacional. Portanto, um equilíbrio entre precisão e rapidez pode ser necessário para a grande maioria destes problemas. As meta-heurísticas são uma alternativa viável para problemas que necessitam de um enorme esforço computacional. Esta sessão apresenta algoritmos populacionais que podem ser aplicados ao

problema proposto e subseções para os algoritmos de inteligência de enxames e algoritmos evolutivos. Como foco no algoritmo populacional da evolução diferencial, esta sessão também apresenta o método utilizado por este trabalho para a resolução do problema proposto.

2.2.1 ALGORITMOS POPULACIONAIS

Os algoritmos populacionais são caracterizados por terem seus indivíduos representando as soluções do problema. Cada indivíduo da população é codificado com a dimensão necessária para representar uma possível solução e passam por processos computacionais que imitam os processos naturais.

2.2.1.1 ALGORITMOS DE INTELIGÊNCIA DE ENXAMES

Os sistemas de inteligência artificial que simulam o comportamento coletivo dos indivíduos de uma população são chamados de inteligência de enxames. Estes algoritmos populacionais bioinspirados simulam o comportamento de bandos, cardumes e manadas. O PSO (*Particle Swarm Optimization*) é um dos mais difundidos na literatura e consiste na otimização por enxame de partículas. Este método foi introduzido por (KENNEDY; EBERHART, 1995) e foi inspirado no comportamento coletivo de bandos e pássaros na sua interação durante o voo. Tais comportamentos foram modelados como um mecanismo de otimização sendo a área sobrevoada o espaço de busca do problema.

A inspiração em insetos sociais deu origem a um modelo que simula uma colônia de insetos. Outro exemplo de comportamento coletivo complexo é a colônia de formigas que examinam eficientemente diversas regiões em busca de alimentos. O algoritmo de otimização bioinspirado na colônia de formigas (*Ant Colony Optimization - ACO*) é um dos mais encontrados na literatura. Dorigo (1999) mostra que o comportamento das formigas pode ser explicado utilizando um modelo probabilístico em que cada formiga decide aleatoriamente o caminho a seguir com base na quantidade de feromônio existente no solo. Quanto maior a concentração desta substância química deixada pelas formigas nos caminhos já percorridos, maior a chance de outras formigas seguirem esse caminho.

Outro algoritmo populacional que simula uma colônia é a colônia artificial de abelhas (*Artificial Bee Colony - ABC*), proposto por Karaboga (2005). Este algoritmo se caracteriza por possuir uma população diferenciada, onde algumas abelhas atuam como exploradoras e buscam boas fontes de alimento, algumas como trabalhadoras e outras como escoteiras. Neste modelo, cada local com alimento representa uma possível solução do problema e o comportamento das

abelhas visa buscar a melhor solução possível como sendo a melhor fonte de alimento para a colônia.

2.2.1.2 ALGORITMOS EVOLUTIVOS

Dentre os algoritmos populacionais existe a classe dos algoritmos evolutivos. Estes algoritmos também utilizam uma população inicial aleatória, porém usam as teorias de Darwin para evoluir os indivíduos por um número predeterminado de gerações. O algoritmo populacional evolutivo mais difundido na literatura é o Algoritmo Genético (AG). Apresentado por Holland (1975), este algoritmo simula a evolução de uma ou mais populações através de gerações. Os processos de seleção, mutação e cruzamento (*crossover*) simulam a reprodução da população e garantem a herança genética. Seu objetivo é encontrar o indivíduo mais adaptável ao problema e, conseqüentemente, uma boa solução.

2.3 EVOLUÇÃO DIFERENCIAL

O algoritmo de Evolução Diferencial (ED) apresentado por Storn e Price (1995) também utiliza uma população inicial aleatória como o AG e busca a evolução de seus indivíduos através das gerações. Porém, seus indivíduos são codificados como vetores reais e os processos de seleção, mutação e *crossover* são adaptados para lidar com este tipo de indivíduo. Sua principal característica é a mutação, pois utiliza uma diferença vetorial (aleatória ou elitizada) para gerar novos indivíduos.

A ED é, sem dúvida, um dos algoritmos mais poderosos de otimização estocástica de parâmetros contínuos (DAS; SUGANTHAN, 2011). A ED trabalha com etapas computacionais semelhantes a outros algoritmos evolutivos. No entanto, ao contrário dos outros algoritmos evolutivos apresentados, as variantes da ED buscam mutar os membros da população com a diferença escalar de vetores selecionados. A ED é um algoritmo que funciona através de um ciclo simples de fases. A Figura 1 apresenta estas fases e o fluxo de atividades do algoritmo.

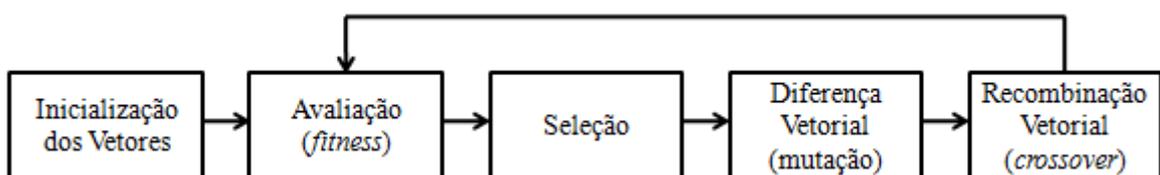


Figura 1: Etapas do algoritmo de ED

Para iniciar este processo, as possíveis soluções do problema são representadas por

um vetor $\vec{X} = [x_1, x_2, x_3, \dots, x_D]$ e cada elemento x_i , com $i = 1, 2, 3, \dots, D$ onde D é a dimensão do problema, pode assumir um valor real. Cada uma das possíveis soluções é avaliada através da função objetivo do problema. Assim cada indivíduo é classificado pelo resultado desta função e os melhores indivíduos são selecionados para gerarem a próxima população. Conseqüentemente, a tarefa de otimização é basicamente uma busca pelo vetor \vec{X}^* que minimiza ou maximiza a função objetivo do problema. A função objetivo para os problemas de minimização se resume à Equação 1.

$$f(\vec{X}^*) \leq f(\vec{X}_i) \quad \forall i \quad (1)$$

Analogamente, para problemas de maximização a função objetivo é representada pela Equação 2.

$$f(\vec{X}^*) \geq f(\vec{X}_i) \quad \forall i \quad (2)$$

A busca pelo melhor indivíduo \vec{X}^* visa encontrar o ótimo global num espaço \mathfrak{R}^D . Tal processo consiste na seleção dos indivíduos para a mutação e o *crossover*, resultando em indivíduos que integram a nova geração da população. Cada geração será denotada na ED pela letra G , sendo $G = 0, 1, \dots, G_{max}$.

Como cada elemento dos vetores está suscetível a mudanças através das gerações, a notificação representada na Equação 3 é adotada para representar o i -ésimo vetor da população na geração G .

$$\vec{X}_i^G = [x_1^G, x_2^G, x_3^G, \dots, x_D^G] \quad (3)$$

Cada um destes indivíduos passará pelos processos de mutação e *crossover* através das gerações preestabelecidas. O algoritmo de ED apresenta algumas variantes de acordo com estes dois processos. As variantes seguem a estrutura de *DE/x/y/z*, onde x é processo de mutação, y a quantidade de diferenças vetoriais utilizadas e z o tipo de *crossover*. A variante mais utilizada na literatura é também a estratégia de maior sucesso da ED e denomina-se *DE/rand/1/bin* (BABU; MUBEEN, 2005). Nesta estratégia o processo de mutação utiliza vetores aleatórios, uma única diferença vetorial para perturbar o indivíduo e o *crossover* acontece de forma binomial. Outras estratégias visam o elitismo no processo de mutação, na variante *DE/best/1/bin* o melhor indivíduo é selecionado como base para o processo. A variante *DE/rand-to-best/1/bin* também utiliza o melhor indivíduo durante a mutação, porém um indivíduo aleatório é utilizado

como base e o melhor indivíduo na diferença vetorial. Estas variantes também podem utilizar duas diferenças vetoriais, criando as variantes *DE/rand/2/bin*, *DE/best/2/bin* e *DE/rand-to-best/2/bin*. Outras variantes utilizam um processo diferente de *crossover*, onde seus vetores são cruzados por um processo exponencial ao invés do binomial. As Subseções 2.3.1 a 2.3.3 abordam como funcionam os processos de seleção, mutação e *crossover* para a estratégia *DE/rand/1/bin*.

2.3.1 SELEÇÃO

A seleção dos vetores que serão preservados para a próxima geração é feita utilizando um critério guloso. Esta seleção depende diretamente do valor da função objetivo de cada indivíduo, para os problemas de minimização os novos indivíduos com o valor menor que o atual passam a ser incorporados na nova geração de indivíduos. Analogamente para problemas de maximização os novos indivíduos são incorporados a população quando o valor da função objetivo for maior que o atual. Para manter o tamanho da população constante ao longo das gerações subsequentes, o processo de seleção determina quais dos novos indivíduos substituirão os indivíduos atuais.

A Equação 4 abaixo apresenta o esquema de seleção para uma função objetivo de minimização.

$$x_i^{G+1} = \begin{cases} u_i^{G+1}, & \text{se } f(u_i^{G+1}) < f(x_i^G) \\ x_i^G, & \text{caso contrário} \end{cases} \quad (4)$$

Neste processo o número de indivíduos da população nunca se altera e a qualidade das soluções encontradas (indivíduos evoluídos) melhora ou se mantém estável, mas nunca se deteriora.

2.3.2 MUTAÇÃO

Biologicamente o termo mutação significa uma mudança repentina nas características genéticas de um cromossomo. No contexto da CE a mutação também pode ser vista como uma mudança ou perturbação no indivíduo utilizando um elemento aleatório. Na ED o vetor selecionado pela taxa de mutação (PM) para ser a base do novo indivíduo é chamado de vetor alvo, o vetor obtido após o processo de mutação é conhecido como vetor doador e finalmente após o processo de recombinação do doador (*crossover*) é chamado de vetor teste (*trial vector*). Na estratégia mais simples de mutação da ED, o vetor doador é criado para cada i -ésimo vetor

alvo \vec{X}_i^G da população atual e dois outros vetores aleatórios e distintos $\vec{X}_{r_1}^G$ e $\vec{X}_{r_2}^G$ são utilizados para mutação. Esta operação de mutação é descrita na Equação 5, que apresenta o vetor doador \vec{V}_i^G e um valor escalar F (normalmente pertencente ao intervalo de $[0,4$ e $1]$) que dimensiona a diferença vetorial dos outros dois vetores selecionados.

$$\vec{V}_i^G = \vec{X}_{r_1}^G + F \times (\vec{X}_{r_2}^G - \vec{X}_{r_3}^G) \quad (5)$$

Este processo é ilustrado no espaço de parâmetros 2-D na Figura 2.

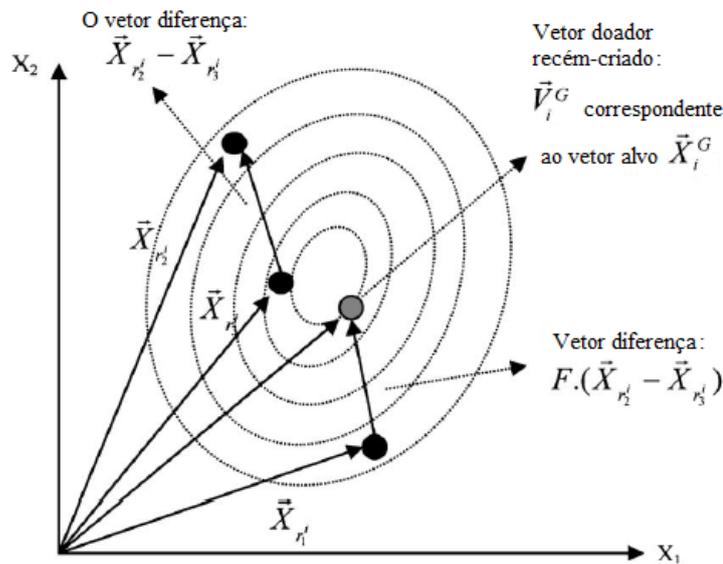


Figura 2: Ilustração de uma mutação simples da ED em um espaço 2-D (DAS; SUGANTHAN, 2011)

2.3.3 CROSSOVER

Um operador de cruzamento é utilizado após a mutação dos indivíduos para manter a diversidade da população e garantir a herança genética das gerações anteriores. Neste processo o vetor doador \vec{V}_i^G troca componentes com o vetor alvo \vec{X}_i^G para formar o vetor teste $\vec{U}_i^G = [u_1^G, u_2^G, u_3^G, \dots, u_i^G]$.

O *crossover* binomial é realizado em cada dimensão do indivíduo sempre que um valor gerado aleatoriamente é menor ou igual ao parâmetro CR . Neste caso, o número de dimensões herdadas do vetor doador tem uma distribuição binomial. Este esquema está representado na Equação 6 abaixo, onde $i = \{1, \dots, D\}$, um valor aleatório $rand_i \in [0, 1]$ e $CR \in [0, 1]$.

$$u_i^G = \begin{cases} v_i^G, & \text{se } rand_i \leq CR \quad \text{ou} \quad i = i_{rand} \\ x_i^G, & \text{caso contrário} \end{cases} \quad (6)$$

Este esquema também garante que u_i^G recebe pelo menos um componente de v_i^G . A Figura 3 representa graficamente o processo de *crossover* binomial para valores de $i = \{1, \dots, 11\}$.

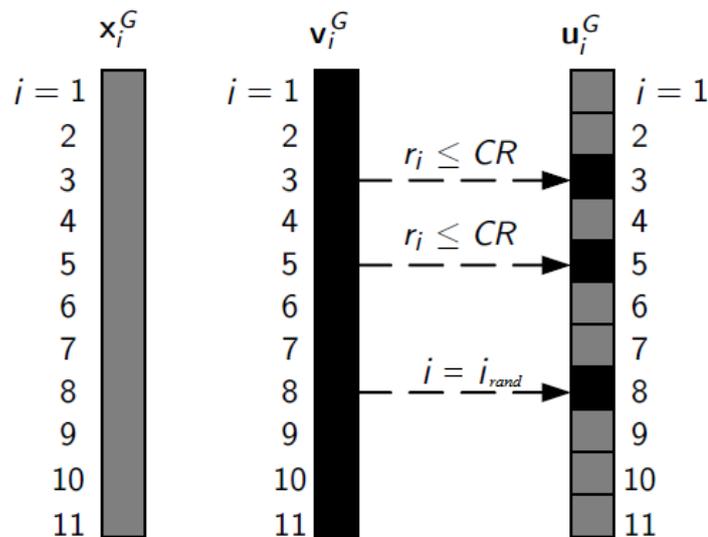


Figura 3: Exemplo do processo de *crossover* binomial
(DAS; SUGANTHAN, 2011)

2.4 EVOLUÇÃO DIFERENCIAL BINÁRIA

O algoritmo de Evolução Diferencial Binária (EDB) é uma meta-heurística populacional introduzido por Krause e Lopes (2012). A EDB é um algoritmo inspirado na ED padrão (STORN; PRICE, 1995) e no processo de mutação do AG (HOLLAND, 1975). Adaptado para problemas binários, esta meta-heurística é um conjunto de mecanismos de gerenciamento atuando sobre métodos heurísticos de ED. Uma estrutura algorítmica geral que pode ser aplicada a diferentes problemas de otimização. A ED é concebida para espaços contínuos, enquanto esta variante do método teve sua codificação adaptada para o espaço binário. Portanto, a codificação do indivíduo também pode ser interpretada como um vetor binário.

Como o processo de mutação da ED utiliza a diferença de vetores, a mutação da EDB foi adaptada para a troca de um ou mais *bit(s)* do vetor. Esta modificação cria soluções não testadas anteriormente e possibilita uma boa diversidade. Este processo de mutação foi inspirado no AG e adaptado para a sua utilização dentro da ED padrão. O EDB também foi adaptado

com um parâmetro de perturbação, esta taxa de perturbação (PR) controla quais indivíduos participarão dos processo de mutação e *crossover*. A estrutura de perturbação busca balancear a mutação inspirada no AG com o *crossover* binomial da ED. Esta modificação garante também que pelo menos um indivíduo da população passará por estes processos e terá pelo menos uma dimensão alterada. Todo o processo de *crossover* foi mantido como na ED padrão.

A Figura 4 apresenta o pseudocódigo da EDB.

```

1: Parâmetros:  $D, N, PM, PR$ 
2: Gerar população inicial  $\vec{x}_i \in \{0,1\}$ 
3: Calcular função fitness  $f(\vec{x}_i)$  de cada indivíduo
4: ENQUANTO condição de parada não atingida FAÇA
5:   PARA  $i = 1$  ATÉ  $N$  FAÇA
6:     Escolha aleatoriamente o índice de um indivíduo:  $s = \text{rndint}(1, N)$  com  $s \neq i$ 
7:     Escolha aleatoriamente o índice de uma variável:  $j_{\text{rand}} = \text{rndint}(1, D)$ 
8:     Solução teste  $\vec{y}$  recebe  $\vec{x}_i$ 
9:     PARA  $j = 1$  ATÉ  $D$  FAÇA
10:      SE  $\text{rndreal}(0,1) < PR$  ou  $j = j_{\text{rand}}$  ENTÃO {Perturbação de  $\vec{y}$ }
11:        SE  $\text{rndreal}(0,1) < PM$  ENTÃO {Mutação}
12:          InverterBit( $\vec{y}_j$ )
13:        SENÃO {Crossover}
14:           $\vec{y}_j = \vec{x}_{sj}$ 
15:      FIM SE
16:    FIM SE
17:  FIM PARA
18:  Calcular  $f(\vec{y})$ 
19:  SE  $f(\vec{y}) > f(\vec{x}_i)$  ENTÃO {Atualiza solução corrente}
20:     $\vec{x}_i$  recebe  $\vec{y}$ 
21:  FIM SE
22: FIM PARA
23: Encontrar a melhor solução corrente  $\vec{x}^*$ 
24: FIM ENQUANTO
25: Apresentar resultados

```

Figura 4: Pseudocódigo da EDB
(KRAUSE; LOPES, 2012)

2.5 EVOLUÇÃO DIFERENCIAL DISCRETIZADA

A Evolução Diferencial Discretizada (DDE) introduzida por Krause et al. (2013) é uma meta-heurística populacional baseada na ED e adaptada para problemas combinatoriais. Sua principal característica é o processo de discretização que todas as dimensões (D) do indivíduo

são submetidas para ter o seu valor de *fitness* avaliado. Este processo consiste no uso de uma função sigmóide que transforma valores contínuos em discretos. Tal função foi escolhida por ser um dos métodos de discretização mais eficientes e utilizados na literatura (KRAUSE et al., 2013). Para o problema de *scheduling* proposto, a Equação 7 foi utilizada para transformar as dimensões contínuas x_i entre [-1 e 1] em binárias [0 ou 1].

$$\vec{X} = \begin{cases} 1, & \text{se } \frac{2}{1+e^{(-2 \cdot x_i)}} - 1 > 0, \\ 0, & \text{caso contrário} \end{cases} \quad (7)$$

Esta característica permite que a EDD continue buscando soluções mais adaptáveis em espaços de busca contínuos para problemas discretos. Usando esta estratégia, cada dimensão do indivíduo evolui continuamente entre -1 e 1 ao invés de saltar entre 0 ou 1. Cada dimensão evolui gradativamente ao redor do zero usando valores contínuos para encontrar a melhor combinação binária para o problema proposto.

A Figura 5 apresenta o pseudocódigo da EDD.

```

1: Função  $f(x)$ : DE( $NP$ ,  $CR$ ,  $F$ ,  $range$ )
2: Gerar população inicial aleatória  $x$ 
3: Calcular função fitness  $f(x)$  de cada indivíduo
4: ENQUANTO condição de parada não atingida FAÇA
5:   PARA  $i = 1$  ATÉ  $NP$  FAÇA
6:      $v_{i,G+1} \leftarrow \text{mutação}(x_{i,G}, F)$ 
7:      $u_{i,G+1} \leftarrow \text{crossover}(x_{i,G}, v_{i,G+1}, CR)$ 
8:   FIM PARA
9:   SE sigmóide ( $u_{i,G+1}$ ) > 0 ENTÃO
10:     $u_{i,G+1} \leftarrow 1$ 
11:   SE NÃO
12:     $u_{i,G+1} \leftarrow 0$ 
13:   FIM SE
14:    $fit_u \leftarrow f(u)$ 
15:   PARA  $i = 1$  ATÉ  $NP$  FAÇA
16:     SE  $fit_u(i) > fit_x(i)$  ENTÃO
17:        $x_{i,G+1} \leftarrow u_{i,G+1}$ 
18:     SE NÃO
19:        $x_{i,G+1} \leftarrow x_{i,G}$ 
20:   FIM SE
21: FIM PARA
22: FIM ENQUANTO
23: Apresentar resultados

```

Figura 5: Pseudocódigo da EDD
(KRAUSE et al., 2013)

2.6 TRABALHOS CORRELATOS

Este trabalho utiliza como ponto de partida os estudos de casos realizados por Westphal et al. (2011). Neste artigo, Westphal apresenta um modelo simplificado contendo algumas das estruturas presentes em polidutos reais. O modelo simula uma rede interligando refinarias, depósitos e clientes finais; entre os depósitos, um poliduto bidirecional. São apresentadas soluções sub-ótimas para cada estudo de caso utilizando uma modelagem temporal discreta. Tais soluções são encontradas a partir da meta-heurística de algoritmos micro-genéticos.

Este mesmo modelo do problema de *scheduling* foi o foco de estudo de Souza et al. (2010), porém a meta-heurística utilizada foi o algoritmo transgenético (uma variação do AG). Por também ser um método estocástico, o algoritmo transgenético também apresenta soluções sub-ótimas para as instâncias propostas. A modelagem temporal discreta também é utilizada em estruturas com um único poliduto. Este é o estudo desenvolvido por Magatão et al. (2004) e Rejowski e Pinto (2004). Estes artigos consideram um único duto interligando um ou diversos destinos e bombeando diversos produtos. A resolução para ambos os artigos foi a utilização da PLIM encontrando assim a solução ótima para o esta modelagem do problema.

Outros trabalhos buscam uma modelagem específica para uma rede de polidutos real (ou uma parte dela) e utilizam a modelagem temporal como contínua. A estrutura mais complexa representada atualmente no Brasil é a rede de polidutos na área de São Paulo e é o foco de estudos de Stebel et al. (2012) e Magatão et al. (2012). Em ambos os estudos a PLIM foi utilizada para encontrar a solução ótima. Nesta modelagem o problema é descrito com um grande número de dutos bidirecionais e uma grande quantidade de produtos e depósitos. Um trabalho similar é apresentado por Neves et al. (2007), porém com um número reduzido de produtos e dutos.

Uma rede de polidutos simplificada com modelagem temporal contínua também é foco de outros dois trabalhos. Garcia et al. (2004) utilizam um único duto bidirecional ligando os depósitos e Cafaro e Cerdá (2012) utilizam depósitos com dupla finalidade de receber e injetar produtos. No primeiro estudo utiliza-se uma abordagem mista de algoritmos evolutivos multiobjetivos com PLIM. O segundo utiliza somente a PLIM para solucionar o problema onde cada duto desta rede leva os diversos produtos a destinos diferentes.

Estudos similares abordam a estrutura com um único poliduto, buscando a melhor combinação para o bombeamento de diversos produtos. Utilizando um poliduto unidirecional e depósitos com uma dupla finalidade, Yongtu et al. (2012) propuseram uma solução sub-ótima utilizando o Algoritmo Evolutivo Multiobjetivo (AEM). Outra estrutura com um único poli-

duto e depósitos que recebem e injetam produtos foi o foco de MirHassani et al. (2013), a resolução em tempo contínuo foi encontrada através da PLIM. Nos estudos de Cafaro e Cerdá (2004, 2008), Relvas et al. (2006), Cafaro et al. (2011), MirHassani et al. (2011), Cafaro e Cerdá (2012) existe um único poliduto em uma única direção. Os problemas de *scheduling* modelados nestes trabalhos representam, em sua maioria, um pedaço de uma rede de polidutos real. Estes estudos buscaram a solução ótima utilizando PLIM. A Tabela 2 apresenta um resumo dos trabalhos correlatos encontrados na literatura e classifica de acordo com a modelagem temporal e a quantidade de dutos.

Tabela 2: Lista de trabalhos correlatos

Autor	Dutos	Produtos	Refinarias	Depósitos	Destinos	Método	Modelagem Temporal
(WESTPHAL et al., 2011)	10	4	2	2	3	Alg. Micro-Genéticos	Discreto
(SOUZA et al., 2010)	10	4	2	2	3	Alg. Transgenético	Discreto
(MAGATÃO et al., 2004)	1	8	1	-	4	PLIM	Discreto
(REJOWSKI; PINTO, 2004)	1	4	1	-	4	PLIM	Discreto
(STEBEL et al., 2012)	30	14	4	6	4	PLIM	Contínuo
(MAGATÃO et al., 2012)	30	14	4	6	4	PLIM	Contínuo
(NEVES et al., 2007)	15	10	3	1	5	PLIM	Contínuo
(GARCIA et al., 2004)	7	4	2	2	3	AEM+PLIM	Contínuo
(CAFARO; CERDÁ, 2012)	7	4	2	2	7	PLIM	Contínuo
(YONGTU et al., 2012)	1	3	1	12	3	AEM	Contínuo
(MIRHASSANI et al., 2013)	1	3	2	-	3	AEM	Contínuo
(CAFARO; CERDÁ, 2004)	1	4	1	-	5	PLIM	Contínuo
(CAFARO; CERDÁ, 2008)	1	4	1	-	5	PLIM	Contínuo
(RELVAS et al., 2006)	1	6	1	-	6	PLIM	Contínuo
(CAFARO et al., 2011)	1	4	1	-	3	PLIM	Contínuo
(MIRHASSANI et al., 2011)	1	4	1	-	4	PLIM	Contínuo
(CAFARO; CERDÁ, 2012)	1	4	1	-	3	PLIM	Contínuo

3 METODOLOGIA

3.1 DEFINIÇÃO DO PROBLEMA

A distribuição de derivados de petróleo é um problema que depende diretamente da rede estruturada para o seu transporte. A alta taxa de ocupação dos dutos cria a necessidade de uma otimização, organizando o envio dos produtos em cada duto de acordo com a demanda e capacidade de armazenamento de cada tanque. A otimização da rede visa uma solução rápida e conseqüentemente mais barata para o transporte dos produtos. Tais produtos são transportados em sucessivos envios de bateladas, onde uma batelada é a quantidade de um determinado produto transportado por um duto.

Geralmente um produto é transportado de refinarias para centros de armazenagem, em seguida, para um destino final (cliente final, portos ou tanques de armazenagem de outras empresas). Estes dutos são, na sua grande maioria, bidirecionais, geralmente interligando os tanques de armazenagem. Nestes dutos os diversos produtos podem fluir de um tanque de armazenagem para outro em qualquer direção, possibilitando a troca de produtos entre os tanques. Algumas redes também utilizam dutos unidirecionais, com um único sentido da refinaria para os tanques de armazenagem e dos tanques para o destino final.

Um determinado duto pode vir a apresentar restrições de uso devido a uma necessidade específica de bombeamento ou uma eventual manutenção. Neste caso a rede inteira é afetada podendo não satisfazer a demanda requerida dos produtos. Nestas situações um remanejo nas bateladas dos produtos para outros dutos e tanques passa a ser uma necessidade imediata. Para isto, o sistema da rede deve recalcular o mais rápido possível o novo caminho de cada produto.

O problema real pode ser extremamente complexo, pois, à medida que se aumenta a quantidade de dutos, aumenta exponencialmente a complexidade da rede. A Figura 6 representa uma estrutura real de polidutos da empresa Petrobras.

Esta estrutura é composta por vários polidutos uni e bidirecionais por todo o país, concentrando suas principais atividades na região sudeste. A modelagem desta estrutura na

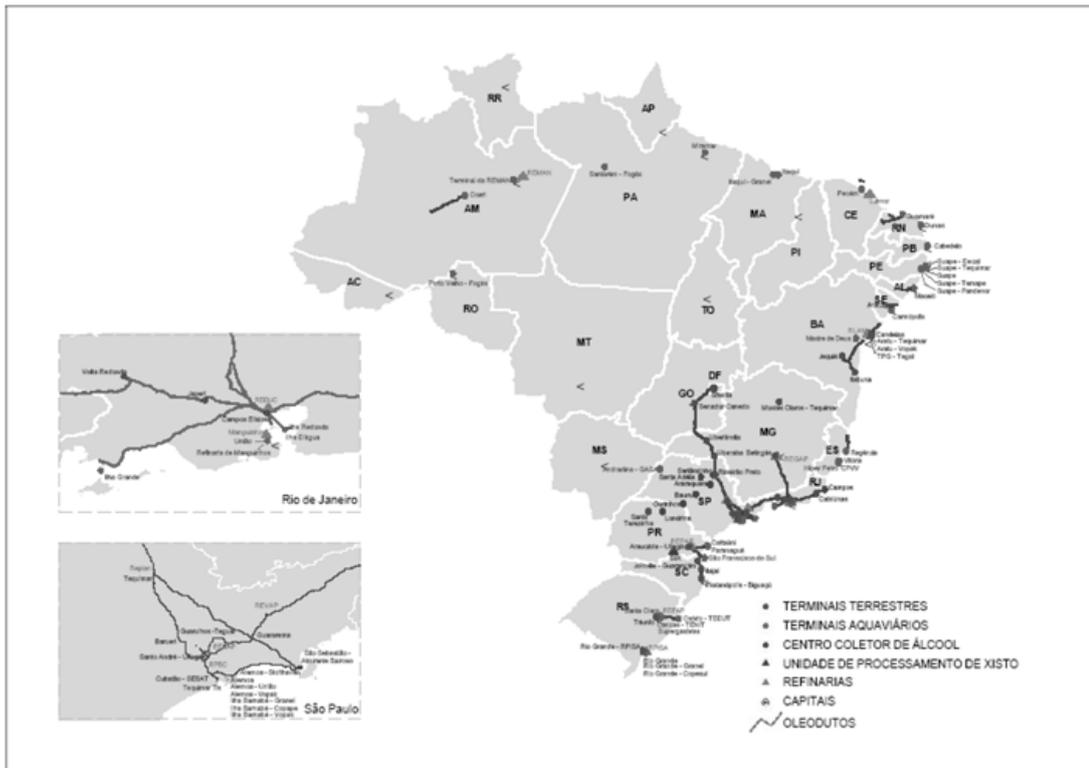


Figura 6: Rede de Polidutos no Brasil
(MAGATÃO et al., 2012)

região sudeste pode ser apresentada de acordo com a Figura 7.

Neste esquema as refinarias são representadas por N3, N4, N5 e N6. Os terminais intermediários por N1, N8, N9, N11, N12 e N13. Os clientes finais como N2 e N14 e os portos como N7 e N10. Neste modelo todos os dutos são bidirecionais, possibilitando assim a troca de qualquer produto entre refinarias, tanques de armazenamento e destinos finais.

3.2 MODELAGEM MATEMÁTICA DO PROBLEMA

As redes de polidutos atualmente utilizadas pela Petrobrás e outras petrolíferas ao redor do mundo consistem em uma estrutura de refinarias (R), tanques de armazenamento (A) e clientes finais (C). Cada duto desta rede pode interligar uma refinaria a tanques de armazenamento, tanques entre si com dutos bidirecionais, tanques de armazenamento a clientes finais ou ainda diretamente refinarias a clientes finais. Estes dutos serão representados aqui pela origem (R ou A) e destino (A ou C), considerando uma quantidade r de refinarias, a de tanques de armazenamento, c de clientes finais e um horizonte de tempo H . Como cada duto conduzirá diferentes produtos por diferentes distâncias, geralmente é associado um tempo máximo para o sistema encontrar a melhor combinação de bateladas dos produtos.

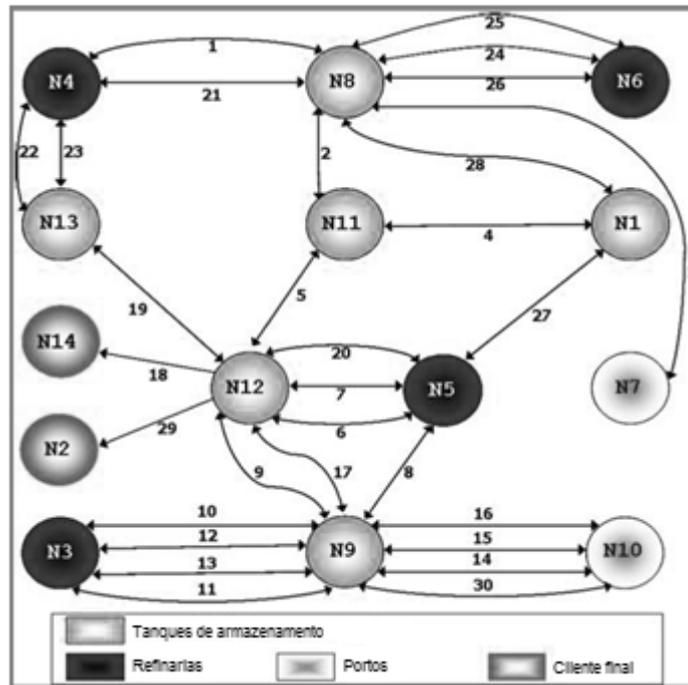


Figura 7: Rede de Polidutos na região sudeste
(MAGATÃO et al., 2012)

As Equações 8, 9 e 10 representam as possíveis refinarias, tanques e clientes do sistema.

$$\vec{R} = \{R_1, R_2, \dots, R_r\} \quad (8)$$

$$\vec{A} = \{A_1, A_2, \dots, A_a\} \quad (9)$$

$$\vec{C} = \{C_1, C_2, \dots, C_c\} \quad (10)$$

Cada possível duto entre as refinarias e os tanques de armazenamento da rede é representado pelo seu nó de origem e seu nó de destino, identificando a refinaria que fornece o produto e o tanque de destino. A matriz que define os dutos de \vec{R} até \vec{A} , interligando as refinarias aos tanques de armazenamento é apresentada na Equação 11.

$$\vec{R} \times \vec{A} = \begin{bmatrix} R_1A_1 & R_1A_2 & \dots & R_1A_a \\ R_2A_1 & R_2A_2 & \dots & R_2A_a \\ \vdots & \vdots & \ddots & \vdots \\ R_rA_1 & R_rA_2 & \dots & R_rA_a \end{bmatrix} \quad (11)$$

Analogamente, a modelagem matemática para dutos bidirecionais entre os tanques de armazenamento é uma matriz quadrada $\vec{A} \times \vec{A}$. Nesta matriz a diagonal principal é nula (valores de $A_1A_1, A_2A_2, \dots, A_aA_a$ iguais a zero), pois representaria dutos que interligam um tanque de

armazenamento a ele mesmo. Sendo assim, o duto representado por A_1A_2 considera a direção de A_1 para A_2 e o duto A_2A_1 com direção de A_2 para A_1 . Sua única restrição é que os dois dutos não podem ser utilizados ao mesmo tempo, pois caracteriza uma colisão entre os produtos. A matriz que define os dutos interligando os tanques de armazenamento é apresentada na Equação 12.

$$\vec{A} \times \vec{A} = \begin{bmatrix} 0 & A_1A_2 & \dots & A_1A_a \\ A_2A_1 & 0 & \dots & A_2A_a \\ \vdots & \vdots & \ddots & \vdots \\ A_aA_1 & A_aA_2 & \dots & 0 \end{bmatrix} \quad (12)$$

As matrizes $\vec{A} \times \vec{C}$ (representando os dutos dos tanques de armazenamento até clientes finais) e $\vec{R} \times \vec{C}$ (representando os dutos das refinarias até os clientes finais) são definidas analogamente a Equação 11. Caso exista mais de um duto interligando cada refinaria a tanques ou clientes, estes dutos também podem ser representados por um vetor de N dimensões. Por exemplo, caso o modelo tenha 4 dutos interligando a refinaria R_1 ao tanque de armazenamento A_2 , estes dutos podem ser representados pelo vetor \vec{RA}_N , com $N = \{1, \dots, 4\}$.

3.3 DISCRETIZAÇÃO

O modelo proposto utiliza dutos discretizados transformando o problema contínuo em um problema combinatorial. Assim, cada elemento das matrizes $\vec{R} \times \vec{A}$, $\vec{A} \times \vec{C}$ e $\vec{R} \times \vec{C}$ pode ser enumerado. Assumindo o valor de D como o número máximo de dutos, os dutos existentes na estrutura real podem ser representados por X_d , com $d = 1, \dots, D$.

As hipóteses para a discretização deste modelo assume que cada unidade discretizada transporta o mesmo volume (m^3) e na mesma quantidade de tempo (hora). Tais hipóteses também consideram que todos os dutos possuem a mesma vazão e transportam todos os produtos na mesma velocidade. Para este modelo não é considerado qualquer restrição na ordem dos produtos (qualquer produto pode preceder ou suceder qualquer produto) e nem o tempo necessário de *setup* (tempo necessário para preparar a batelada do produto). Sendo assim, cada duto possui um valor inteiro W associado representando a quantidade do produto e o tempo necessário para o transporte. Como exemplo, o duto X_d com peso $W_d = 3$ representará neste modelo um duto que transporta $3000 m^3$ de qualquer produto no período de 3 horas.

Para evitar a fragmentação da batelada, este modelo assume que, após o início do bombeamento do produto para o duto, este deve ser preenchido por completo, evitando assim o

transporte de pequenas quantidades dos produtos pela rede. Portanto, cada duto é discretizado de acordo com o quantidade e tempo para o transporte da batelada de um ponto a outro.

Neste trabalho utiliza-se variáveis binárias para representar a existência (1) ou não (0) de cada produto em um determinado duto e período de tempo. Sendo P os diversos tipos de produtos transportados pela rede, cada variável binária também depende do produto transportado. Portanto, as variáveis binárias definidas dependem de $d = \{1, \dots, D\}$, $p = \{1, \dots, P\}$ e $t = \{0, \dots, H-1\}$. Nesta modelagem é utilizado o valor de k para representar o período de tempo que o produto p está presente no duto d . Este período de tempo refere-se ao processo de encher o tanque por completo, determinado pelo peso W_d de cada duto. A Equação 13 mostra como cada valor de k é encontrado para um horizonte H de tempo.

$$k = W_d \times t \quad k < H \quad (13)$$

Portanto, as variáveis binárias do modelo são definidas pela Equação 14.

$$X_{p,d,k} = \begin{cases} 1 \\ 0 \end{cases} \quad \forall p, d, k \quad (14)$$

A Figura 8 exemplifica o uso das Equações 13 e 14 apresentando a estrutura da rede em um horizonte $H = 12$, onde $W_1=2$, $W_2=3$ e $W_d=4$.

$d \backslash t$	0	1	2	3	4	5	6	7	8	9	10	11
1 ($W_1=2$)	$X_{p,1,0}$		$X_{p,1,2}$		$X_{p,1,4}$		$X_{p,1,6}$		$X_{p,1,8}$		$X_{p,1,10}$	
2 ($W_2=3$)	$X_{p,2,0}$			$X_{p,2,3}$			$X_{p,2,6}$			$X_{p,2,9}$		
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
d ($W_d=4$)	$X_{p,d,0}$				$X_{p,d,4}$				$X_{p,d,8}$			

Figura 8: Exemplo da disposição dos dutos discretizados

Esta codificação permite a modelagem do problema e a criação da função objetivo como a minimização do tempo de entrega.

3.4 ANÁLISE DE COMPLEXIDADE

A complexidade do problema proposto está ligada ao Número de Variáveis (NV) do modelo. O Espaço de Busca (EB) criado pelo modelo é a soma das possíveis combinações de cada variável. Como cada variável binária cria duas únicas opções, a Equação 15 representa este EB.

$$EB = 2^{NV} \quad (15)$$

Consequentemente, o tamanho do EB aumenta exponencialmente dependendo do NV. Neste modelo, o NV depende diretamente do número de dutos (D), o peso de cada duto (W_d), o número de produtos (P) e o horizonte de tempo (H). A Equação 16 apresenta um cálculo do total de variáveis que o modelo necessita.

$$NV = P \times \sum_{d=1}^D \left(\frac{H}{W_d} \right) \quad (16)$$

O número de dutos de uma rede polidutos varia de acordo com sua estrutura real. Portanto, o valor de D está associado ao número de refinarias (r), tanques de armazenamento (a) e clientes finais (c). Considerando as matrizes $\vec{R} \times \vec{A}$, $\vec{A} \times \vec{A}$ e $\vec{A} \times \vec{C}$ apresentadas na Seção 3.2, o número de dutos é a soma das variáveis destas matrizes. Com isso, o número total de dutos e, conseqüentemente, de variáveis aumenta ou diminui com qualquer alteração na estrutura. Porém, o impacto maior na complexidade do modelo está ligado ao número de tanques de armazenamento. Por possuir dutos bidirecionais e funcionar como intermédio entre as refinarias e os clientes finais, o número de dutos aumenta mais quando adicionado um tanque de armazenamento do que quando adicionado uma refinaria ou cliente final.

O número de produtos (P) é outro fator que aumenta a complexidade do modelo. Analogamente ao número de dutos, o aumento no número de produtos aumenta consideravelmente o número de variáveis. O horizonte de tempo (H) também é um fator importante, caso o modelo busque horizontes muito grandes (meses e anos) o NV aumenta proporcionalmente. Consequentemente, a complexidade do modelo apresentado aumenta para cada novo produto e horizontes de tempo maiores.

3.5 MODELO SIMPLIFICADO

As matrizes $\vec{R} \times \vec{A}$, $\vec{A} \times \vec{A}$, e $\vec{A} \times \vec{C}$ compõem o conjunto necessário de dutos para representar grande parte das estruturas de polidutos presente no país. A modelagem matemática aqui descrita possibilita a aplicação do método em outras topologias de redes, deixando em aberto o número de refinarias, tanques de armazenamento e destinos. Para exemplificar a sua aplicação, um modelo simplificado hipotético é utilizado.

O modelo tratado neste trabalho é uma simplificação de uma rede real com um número reduzido de refinarias, tanques de armazenamento e clientes finais. Este modelo busca representar as características de uma estrutura real utilizando dutos uni e bidirecionais. A rede em discussão é representada na Figura 9 e proposta inicialmente por Souza et al. (2010) e Westphal et al. (2011).

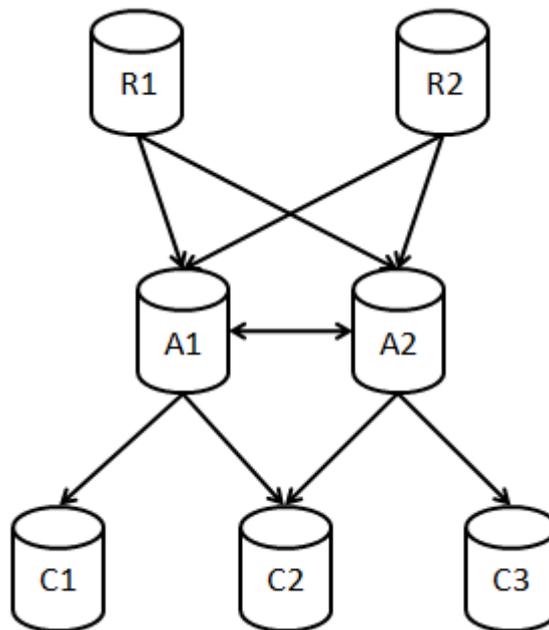


Figura 9: Rede simplificada

Esta rede representa duas refinarias (R_1 e R_2), dois tanques de armazenamento (A_1 e A_2) e três clientes finais (C_1 , C_2 e C_3). As setas representam a ligação entre as refinarias e os tanques de armazenamento unidirecionalmente (R_1A_1 , R_1A_2 , R_2A_1 e R_2A_2), bidirecionalmente entre os tanques de armazenamento (A_1A_2 e A_2A_1) e unidirecionalmente entre os tanques de armazenamento e os clientes finais (A_1C_1 , A_1C_2 , A_2C_2 e A_2C_3). Neste modelo o número de refinarias, tanques de armazenamento e clientes finais definem os valores de $r=2$, $a=2$ e $c=3$. As matrizes que representam este modelo simplificado são apresentadas nas Equações 17, 18 e 19.

$$\vec{R} \times \vec{A} = \begin{bmatrix} R_1A_1 & R_1A_2 \\ R_2A_1 & R_2A_2 \end{bmatrix} \quad (17)$$

$$\vec{A} \times \vec{A} = \begin{bmatrix} 0 & A_1A_2 \\ A_2A_1 & 0 \end{bmatrix} \quad (18)$$

$$\vec{A} \times \vec{C} = \begin{bmatrix} A_1C_1 & A_1C_2 & 0 \\ 0 & A_2C_2 & A_2C_3 \end{bmatrix} \quad (19)$$

Este modelo considera que os produtos são entregues na forma de bateladas discretas. Uma batelada representa o volume de um produto a ser transportado em cada duto. Com o objetivo de evitar as fragmentações dos produtos enviados, este modelo assume cada batelada como o preenchimento total do duto. Para simplificar o problema, cada duto é considerado com o mesmo diâmetro e mesmas características. Todos os produtos fluem com a mesma velocidade e ocupam o mesmo volume no duto. Sendo assim, cada duto é definido com uma distância normalizada em termos de unidades de tempo necessárias para que o volume da batelada seja transportada. A Tabela 3 enumera os dutos e apresenta os pesos associados (W_d), valores sugeridos por Westphal et al. (2011) em seus estudos de casos e adaptáveis a valores contínuos para modelagens reais.

Tabela 3: Pesos de cada duto (WESTPHAL et al., 2011)

Duto	Numeração do duto	Peso do duto
R_1A_1	D_1	$W_1 = 1$
R_1A_2	D_2	$W_2 = 3$
R_2A_1	D_3	$W_3 = 3$
R_2A_2	D_4	$W_4 = 2$
A_1A_2	D_5	$W_5 = 3$
A_2A_1	D_6	$W_6 = 3$
A_1C_1	D_7	$W_7 = 4$
A_1C_2	D_8	$W_8 = 2$
A_2C_2	D_9	$W_9 = 3$
A_2C_3	D_{10}	$W_{10} = 2$

A Figura 10 modela a estrutura da rede utilizando os pesos atribuídos a cada duto representando cada duto.

A rede de polidutos real possui ainda outros fatores a serem considerados, como a quantidade de produtos e quais refinarias produzem quais produtos. Este modelo simula a distribuição de até seis produtos ($P=6$). Neste modelo simplificado, a refinaria R_1 produz somente os produtos 1, 2 e 3 e a refinaria R_2 somente os produtos 4, 5 e 6. No caso do duto

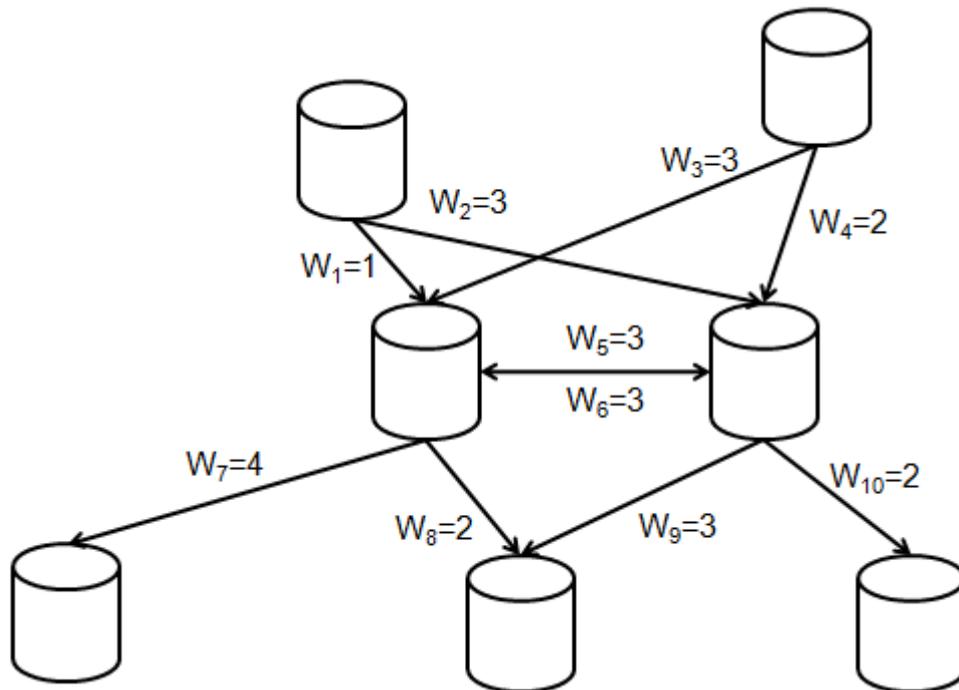


Figura 10: Rede simplificada estruturada pelos pesos P_d

bidirecional definido pela matriz $\vec{A} \times \vec{A}$, esta conexão é desdobrada em duas possíveis conexões, o duto A_1A_2 e o duto A_2A_1 (representando o caminho de ida e de volta). Consequentemente, os dutos A_1A_2 e A_2A_1 representam um único duto bidirecional e não podem ser utilizados ao mesmo tempo, evitando assim possíveis colisões.

Considera-se ainda a tancagem agregada do produto, isto é, cada tanque de armazenamento e clientes finais possui a quantidade de tanques necessária para cada produto que ele possa receber. Este modelo simplificado utiliza um horizonte de tempo máximo de 48 horas ($H=48$), este valor é comumente usado para representar um horizonte de dois dias. Com o modelo definido, busca-se a solução do problema como a melhor combinação de bateladas enviadas para satisfazer a demanda de cada cliente.

3.5.1 FUNÇÃO OBJETIVO

O objetivo principal do modelo é entregar a quantidade exata de cada produto exigido por cada cliente no menor tempo possível. Matematicamente este objetivo pode ser representado como a minimização do tempo de entrega e a demanda exigida por cada cliente como restrições de fornecimento e entrega. A Equação 20 representa a função objetivo Z do modelo proposto.

$$Z = \min \left(\sum_{p=1}^P \sum_{d=1}^D \sum_{k=0}^{H-1} (W_d \times (k+1)) \times X_{p,d,k} \right) \quad (20)$$

Esta função pode ser interpretada como a minimização do tempo, onde o período de tempo W_d é o peso de cada variável binária. Portanto, para as variáveis iniciais ($k=0$) também fazem parte da função objetivo, o peso adotado para a função Z é acrescido de uma unidade de tempo ($k+1$).

3.5.2 RESTRIÇÕES

As restrições do modelo buscam representar linearmente as restrições de uma rede real, garantido assim a unicidade dos produtos em cada duto, a unicidade do duto bidirecional, a ordem das bateladas e as quantidades de fornecimento e entrega.

Para representar restrições de produção, este modelo assume que a refinaria R_1 produz somente os produtos 1, 2 e 3 e a refinaria R_2 produz somente os produtos 4, 5 e 6. Consequentemente, as variáveis $X_{1,3,k}$, $X_{1,4,k}$, $X_{2,3,k}$, $X_{2,4,k}$, $X_{3,3,k}$, $X_{3,4,k}$, $X_{4,1,k}$, $X_{4,2,k}$, $X_{5,1,k}$, $X_{5,2,k}$, $X_{6,1,k}$, $X_{6,2,k}$ podem ser excluídas do modelo.

- **Unicidade dos dutos:** Esta restrição garante que somente um único produto p seja bombeado em cada duto a cada instante de tempo. A Equação 21 apresenta restrições de unicidade modeladas matematicamente.

$$\sum_{p=1}^P X_{p,d,k} \leq 1 \quad (21)$$

- **Unicidade do duto bidirecional:** O duto bidirecional definido pela matriz $\vec{A} \times \vec{A}$ é desdobrado em duas possíveis conexões, o duto A_1A_2 e o duto A_2A_1 . Esta restrição garante que quando o duto A_1A_2 está sendo utilizado, o duto A_2A_1 não pode ser utilizado e vice-versa. A Equação 22 apresenta as restrições do duto bidirecional modeladas matematicamente.

$$\sum_{p=1}^P (X_{p,5,k} + X_{p,6,k}) \leq 1 \quad (22)$$

- **Ordem dos dutos:** Esta restrição garante que a matriz $\vec{A} \times \vec{C}$ que transporta os produtos para os clientes finais só os façam para produtos provenientes dos dutos anteriores. Por exemplo, o duto D_7 que abastece o cliente 1 depende dos dutos D_1 e D_5 para receber

os produtos 1, 2 e 3 e dos dutos D_3 e D_5 para os produtos 4, 5 e 6. Analogamente os dutos D_8 , D_9 e D_{10} também dependem de dutos anteriores. Os dutos D_5 e D_6 também precisam respeitar a ordem das bateladas, pois dependem do bombeamento dos produtos pelos dutos D_1 , D_2 , D_3 e D_4 . A Equação 23 apresenta as restrições de ordem modeladas para os dutos D_7 e D_8 , que recebem os produtos 1, 2 e 3 ($p=\{1, 2, 3\}$) dos dutos D_1 e D_6 . Analogamente, a Equação 24 apresenta as restrições de ordem modeladas para os produtos 4, 5 e 6 ($p=\{4, 5, 6\}$) provenientes dos dutos D_3 e D_6 .

$$\sum_{k=0}^{H-1} (W_7 \times X_{p,7,k+W_7} + W_8 \times X_{p,8,k+W_8}) \leq \sum_{k=0}^{H-1} (W_1 \times X_{p,1,k} + W_6 \times X_{p,6,k}) \quad (23)$$

$$\sum_{k=0}^{H-1} (W_7 \times X_{p,7,k+W_7} + W_8 \times X_{p,8,k+W_8}) \leq \sum_{k=0}^{H-1} (W_3 \times X_{p,3,k} + W_6 \times X_{p,6,k}) \quad (24)$$

A Equação 25 apresenta as mesmas restrições de ordem para os dutos D_9 e D_{10} , que recebem os produtos 1, 2, e 3 ($p=\{1, 2, 3\}$) dos dutos D_2 e D_5 . Analogamente, a Equação 26 para os produtos 4, 5 e 6 ($p=\{4, 5, 6\}$) provenientes dos dutos D_4 e D_5 .

$$\sum_{k=0}^{H-1} (W_9 \times X_{p,9,k+W_9} + W_{10} \times X_{p,10,k+W_{10}}) \leq \sum_{k=0}^{H-1} (W_2 \times X_{p,2,k} + W_5 \times X_{p,5,k}) \quad (25)$$

$$\sum_{k=0}^{H-1} (W_9 \times X_{p,9,k+W_9} + W_{10} \times X_{p,10,k+W_{10}}) \leq \sum_{k=0}^{H-1} (W_4 \times X_{p,4,k} + W_5 \times X_{p,5,k}) \quad (26)$$

A ordem dos dutos D_5 e D_6 são apresentadas nas Equações 27 e 28 respectivamente.

$$\sum_{k=0}^{H-1} (W_5 \times X_{p,5,k+W_5}) \leq \sum_{k=0}^{H-1} (W_1 \times (X_{1,1,k} + X_{2,1,k} + X_{3,1,k}) + W_3 \times (X_{4,3,k} + X_{5,3,k} + X_{6,3,k})) \quad (27)$$

$$\sum_{k=0}^{H-1} (W_6 \times X_{p,6,k+W_6}) \leq \sum_{k=0}^{H-1} (W_2 \times (X_{1,2,k} + X_{2,2,k} + X_{3,2,k}) + W_4 \times (X_{4,4,k} + X_{5,4,k} + X_{6,4,k})) \quad (28)$$

- **Quantidades de fornecimento e entrega:** Esta restrição garante que todo o produto entregue para um cliente tenha sido fornecido pelas refinarias. Ela também garante a entrega da quantidade de cada produto p exigida por cada cliente. O modelo ainda permite que o

bombeamento total das refinarias seja maior que a quantidade total entregue. Tais produtos extras ficam armazenados nos tanques intermediários e possibilitam uma entrega mais rápida. A Equação 29 define que a quantidade total de produtos fornecidos pela refinaria seja maior ou igual ao a quantidade exigida (Q). A Equação 30 define que a quantidade total de produtos entregues aos clientes finais seja igual ao a quantidade exigida.

$$\sum_{p=1}^P \sum_{k=0}^{H-1} (W_1 \times X_{p,1,k} + W_2 \times X_{p,2,k} + W_3 \times X_{p,3,k} + W_4 \times X_{p,4,k}) \geq Q \quad (29)$$

$$\sum_{p=1}^P \sum_{k=0}^{H-1} (W_7 \times X_{p,7,k} + W_8 \times X_{p,8,k} + W_9 \times X_{p,9,k} + W_{10} \times X_{p,10,k}) = Q \quad (30)$$

Para os clientes 1 e 3, que são abastecidos através de um único duto, as Equações 31 e 32 representa as restrições de entrega sendo Q_c a quantidade de cada um dos produtos a ser entregue para cada cliente.

$$\sum_{k=0}^{H-1} (W_7 \times X_{p,7,k}) = Q_1 \quad (31)$$

$$\sum_{k=0}^{H-1} (W_{10} \times X_{p,10,k}) = Q_3 \quad (32)$$

Os dutos D_8 e D_9 abastecem o cliente 2 simultaneamente. Portanto, a quantidade Q_2 exigida por este cliente é a soma dos produtos transportados por estes dois dutos. A Equação 33 representa esta restrição.

$$\sum_{k=0}^{H-1} (W_8 \times X_{p,8,k} + W_9 \times X_{p,9,k}) = Q_2 \quad (33)$$

A implementação do modelo proposto na PLIM e nos algoritmos heurísticos da ED foi feita utilizando a função objetivo e restrições aqui descritas. Para a PLIM foi utilizado o LP_Solve, *software* livre descrito na Seção 2.1. Para o modelo heurístico, cada uma das variáveis do problema foi associada a uma dimensão do cromossomo. Portanto, o algoritmo populacional da ED foi implementado com indivíduos codificados para receber variáveis binárias em cada uma de suas dimensões.

4 EXPERIMENTOS E RESULTADOS

Neste Capítulo são apresentados os experimentos e resultados obtidos a partir da modelagem matemática descrita no Capítulo 3. A implementação da PLIM utiliza o LP_solve 5.5.2.0, um *software* livre para programação linear baseado nos métodos SIMPLEX e *Branch-and-bound*. Para os métodos heurísticos baseados na ED 3.6¹, as implementações foram feitas em ANSI C. O *hardware* utilizado para todos os experimentos consiste em cinco (5) máquinas com oito (8) processadores Intel(R) Core(TM) i7 (3,5 GHz), memória RAM de 8GB e sistema operacional Ubuntu Server 12.04.

4.1 PROGRAMAÇÃO LINEAR INTEIRA MISTA

Os experimentos utilizando a PLIM buscaram através dos métodos SIMPLEX e B&B a melhor combinação de variáveis binárias para esta modelagem. Esta combinação é a solução ideal (valor ótimo) do problema e representa a melhor sequência de bateladas enviadas no menor tempo possível. O ótimo é alcançado codificando a função objetivo e restrições do modelo simplificado para a linguagem do LP_solve. Assim, cada um dos *benchmarks* foi submetido ao processamento por um período máximo de 7 dias. Em poucas horas de processamento (5 horas) os *benchmarks* 01, 02 e 03 com espaços de busca menores ($P=2$) alcançaram valores ótimos. Com o aumento da quantidade de produtos (P) transportados o número de variáveis do modelo aumenta proporcionalmente. Consequentemente, o espaço de busca (EB) aumenta exponencialmente (Equação 15), aumentando também o tempo de processamento. Para os *benchmarks* 04, 05 e 06 ($P=3$) o tempo de processamento necessário chegou a seis (6) dias e encontrou as soluções ótimas destas instâncias. Para os outros *benchmarks* ($P=4,5,6$) o tempo de processamento não foi o suficiente, encontrando assim somente valores sub-ótimos. Tais sub-ótimos são soluções válidas com uma excelente qualidade, porém não são a melhor solução para o modelo proposto. Para estes *benchmarks* seria necessário um esforço computacional maior e consequentemente mais tempo de processamento.

¹<http://http.icsi.berkeley.edu/~storn/>

A Tabela 4 apresenta os *benchmarks* propostos, o número de produtos (p) transportados, as refinarias (R_1 e R_2) que produzem cada produto, o número de variáveis binárias (NV), a quantidade exigida por cada um dos três clientes (Q_1 , Q_2 e Q_3) e o valor da função objetivo encontrado.

Tabela 4: Benchmarks

<i>Benchmark</i>	P	R_1	R_2	NV	Q_1	Q_2	Q_3	Ótimo* / Sub-ótimo
01	2	1	4	320	12	12	12	658*
02	2	1	4	320	12	24	12	1100*
03	2	1	4	320	12	24	18	1450*
04	3	1,2	4	478	8	8	8	803*
05	3	1,2	4	478	12	12	12	1654*
06	3	1,2	4	478	12	18	12	2288*
07	4	1,2	4,5	640	8	8	8	1245
08	4	1,2	4,5	640	8	16	8	2024
09	4	1,2	4,5	640	8	18	10	3206
10	5	1,2,3	4,5	812	8	8	6	1626
11	5	1,2,3	4,5	812	8	12	6	2184
12	5	1,2,3	4,5	812	8	12	8	2966
13	6	1,2,3	4,5,6	960	4	6	4	1090
14	6	1,2,3	4,5,6	960	4	8	4	1285
15	6	1,2,3	4,5,6	960	4	12	6	2496

As melhores soluções foram projetadas em gráficos, representando cada produto (P) em cada instante de tempo (t). Como exemplo de solução ótima, a solução encontrada para o *benchmark 03* é apresentada na Figura 11. Esta figura apresenta cada duto D_d em cada instante de tempo, demonstrando o uso de cada um dos dutos apenas dois produtos.

A Figura 12 também representa uma solução viável encontrada, porém é uma solução sub-ótima do *benchmark 14*. Nesta solução é possível perceber os diversos "buracos" no gráfico, representando que o duto não está sendo usado ou esperando para escoar o produto. Estas lacunas são consequências do truncamento da rede e a espera pelo bombeamento do produto pelas refinarias.

As Figuras 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26 e 27 apresentadas no apêndice deste trabalho representam as demais soluções.

Em todas as soluções encontradas, os dutos D_1 , D_2 , D_3 e D_4 foram utilizados continuamente para fornecer a quantidade de produtos necessária o mais rápido possível. Os dutos D_5 e D_6 , que representam o duto bidirecional, foram mais utilizados pelos *benchmarks* com muitos produtos ($p=5,6$). Tais dutos ajudam no remanejamento dos produtos entre os tanques de armazenamento e se faz necessário para transportar maiores quantidades no mesmo horizonte de tempo.

Os dutos D_7 , D_8 , D_9 e D_{10} entregam os produtos aos clientes finais e dependem da disponibilidade de cada produto nos tanques de armazenamento. Os dutos D_7 , D_8 , D_9 e D_{10} entregam os produtos aos clientes finais e dependem da disponibilidade de cada produto nos tanques de armazenamento. Portanto, estes dutos devem esperar o bombeamento dos produtos através dos dutos anteriores até os tanques de armazenamento. Conseqüentemente, estes dutos possuem intervalos em que não são utilizados pois devem esperar os dutos subseqüentes terminarem as suas bateladas.

4.2 EVOLUÇÃO DIFERENCIAL

A discretização da ED pelos métodos apresentados nas Seções 2.4 e 2.5 resultou em duas meta-heurísticas, a EDB e a EDD. Cada método possui uma adaptação diferente para trabalhar em espaços binários. A EDB possui uma codificação que permite somente valores binários. A EDD busca por soluções em um espaço contínuo, similar à ED original. A discretização da EDD pela função sigmóide discretiza os valores contínuos em valores binários para a avaliação do *fitness*.

Por serem meta-heurísticas, a EDB e a EDD iniciam seus indivíduos com valores aleatórios e buscam a evolução desta população. Conseqüentemente, seus resultados são influenciados pelo número de indivíduos da população, o número de gerações e os seus parâmetros de controle. O número de indivíduos e gerações foram selecionados seguindo a literatura (PRICE et al., 2005). A população de 300 indivíduos e o valor de 20000 gerações foram selecionados para os experimentos. Tais valores são utilizados por Price et al. (2005) em problemas com grandes espaços de busca.

A configuração dos parâmetros de cada algoritmo também é baseada nos parâmetros da literatura. Porém, os *benchmarks* (02, 05, 08, 11 e 14) foram testados utilizando uma margem de $\pm 5\%$ para cada parâmetro. Os experimentos com a EDB usaram os parâmetros de 10% para a taxa de mutação e 50% para a taxa de perturbação (KRAUSE; LOPES, 2012). Portanto, foram realizados experimentos fatoriais com 5%, 10% e 15% para a mutação e 45%, 50% e 55% para a perturbação da EDB. Os melhores resultados alcançados nestes testes utilizaram os parâmetros de 15% e 50% respectivamente. Com uma taxa de mutação mais elevada, o EDB busca uma diversidade maior. Esta característica mostra a necessidade de uma busca global maior.

Com o mesmo procedimento, foram configurados os parâmetros da EDD. A mutação de 10% e *crossover* de 80% são sugeridos na literatura (KRAUSE et al., 2013) e também foram

testados. Portanto, os valores de 5%, 10% e 15% foram testados para a mutação e os valores de 75%, 80% e 85% para o *crossover*. Para estes experimentos os valores sugeridos na literatura proporcionaram os melhores resultados. No total, foram realizados 9000 experimentos para o ajuste dos parâmetros de ambos os métodos.

A Tabela 5 resume os parâmetros utilizados nos experimentos da EDB e da EDD.

Tabela 5: Parâmetros

Parâmetro	EDB	EDD
População	300	300
Gerações	20000	20000
PM	15%	10 %
PR	50%	-
CR	-	80%

Após o ajuste dos parâmetros e a configuração do número de indivíduos e gerações, cada *benchmark* foi processado 100 vezes. O tempo de processamento de cada algoritmo para as 100 rodadas variou de acordo com o espaço de busca. Conseqüentemente, quanto maior o número de variáveis, maior a dimensão (D) de cada indivíduo e maior o tempo de processamento necessário. Os *benchmarks* 01, 02 e 03 necessitaram, em média, 2 horas para a EDB e 4 horas para a EDD. O tempo de processamento cresce para os demais *benchmarks* de acordo com o número de produtos (P) transportados. A EDB necessitou em média de 3 horas para os *benchmarks* 04, 05 e 06 (P=3), 4 horas para os *benchmarks* 07, 08 e 09 (P=4), 5 horas para os *benchmarks* 10, 11 e 12 (P=5) e 7 horas para os *benchmarks* 13, 14 e 15 (P=6). A EDD demandou mais tempo de processamento e completou todas as iterações após 4 horas (média) para os *benchmarks* 01, 02 e 03, 5 horas para os *benchmarks* 04, 05 e 06, 6 horas para os *benchmarks* 07, 08 e 09, 8 horas para os *benchmarks* 10, 11 e 12 e 11 horas para os *benchmarks* 13, 14 e 15.

Após as 100 rodadas, cada uma das soluções encontradas apresenta uma alternativa para o transporte dos produtos. Estas alternativas não otimizam ao máximo o uso dos dutos, porém são soluções viáveis que podem ser utilizadas pelas refinarias quando não se pode esperar o processamento completo da PLIM.

A Tabela 6 apresenta os resultados encontrados por cada método. Para cada coluna dos métodos (EDB e EDD), o melhor indivíduo/solução é apresentado (Melhor) junto com a média das 100 soluções encontradas e o desvio padrão (Média \pm DP). A coluna porcentagem (%) calcula o quão longe do melhor valor conhecido (Ótimo ou Sub-Ótimo) está a solução encontrada por ambos os métodos.

Estes resultados também apresentam o comportamento de cada método. Em todos os

Tabela 6: Resultados da EDB e EDD

<i>Benchmarks</i>	<i>p</i>	Ótimo* / Sub-Ótimo	EDB			EDD		
			Melhor	Média ± DP	%	Melhor	Média ± DP	%
01	2	658*	658	664,40 ± 11,04	0,00%	658	720,20 ± 36,59	0,00%
02	2	1100*	1100	1128,33 ± 14,98	0,00%	1100	1198,55 ± 46,74	0,00%
03	2	1450*	1450	1458,55 ± 24,58	0,00%	1450	1541,90 ± 57,43	0,00%
04	3	803*	857	866,30 ± 15,03	6,72%	803	867,62 ± 46,32	0,00%
05	3	1654*	1747	1752,68 ± 17,89	5,62%	1654	1739,40 ± 51,56	0,00%
06	3	2288*	2446	2460,84 ± 24,59	6,91%	2358	2479,30 ± 76,73	3,06%
07	4	1245	1345	1370,03 ± 33,49	8,03%	1303	1396,79 ± 59,88	4,66%
08	4	2024	2177	2213,92 ± 37,98	7,56%	2136	2331,53 ± 74,49	5,53%
09	4	3206	3517	3564,02 ± 73,66	9,70%	3438	3625,86 ± 98,89	7,24%
10	5	1626	1770	1819,23 ± 30,77	8,86%	1715	1928,06 ± 82,30	5,47%
11	5	2184	2401	2506,54 ± 59,06	9,94%	2392	2673,06 ± 98,31	9,52%
12	5	2966	3344	3481,07 ± 80,27	12,75%	3307	3485,51 ± 107,32	11,50%
13	6	1090	1210	1244,26 ± 48,68	11,01%	1201	1292,96 ± 94,40	10,18%
14	6	1285	1493	1688,22 ± 61,30	16,19%	1479	1734,32 ± 101,79	15,10%
15	6	2496	3153	3240,42 ± 94,29	26,32%	3000	3372,58 ± 138,35	20,19%

benchmarks propostos, a EDD encontrou soluções mais próximas do ótimo/sub-ótimo conhecido. Apesar disso, a média e o desvio padrão das 100 rodadas da EDD são maiores do que a média e o desvio padrão da EDB. Porém, estes resultados sugerem a EDD como um algoritmo de maior busca global. Este fato está provavelmente associado ao processo original de mutação da ED que foi mantido durante as iterações da EDD. Contudo, ambos os métodos apresentam soluções viáveis para o problema combinatorial proposto.

Alguns testes de hipóteses foram realizados para comparar os resultados encontrados por cada um dos métodos. A análise estatística das médias através do *boxplot* aponta diferenças significativas na maioria dos *benchmarks* testados. Os *benchmarks 01, 02, 03, 08, 09, 10, 11, 13 e 15* apresentaram tais diferenças pois os intervalos entre o primeiro e terceiro quartis não se sobrepõem nos *boxplots*. A Figura 13 representa esta característica.

As Figuras 28, 29, 30, 31, 32, 33, 34 e 35 são apresentadas no apêndice do trabalho.

Os *benchmarks 04, 05, 06, 07 e 12* apresentaram uma sobreposição dos intervalos em seus *boxplots*. A Figura 14 exemplifica esta sobreposição dos 100 resultados encontrados por cada método no *benchmark 12*.. Esta característica aponta que a diferença entre os resultados encontrados pode não ser significativa.

As Figuras 36, 37, 38 e 39 são apresentadas no apêndice do trabalho.

Nestes casos, os testes estatísticos de normalidade e não-paramétrico foram utilizados para determinar a significância entre os resultados encontrados. O teste Shapiro-Wilk foi utilizado com confiança de 95% para determinar se os dados seguem uma distribuição normal. Todos os cinco *benchmarks* testados negam a hipótese do teste e desviam de uma distribuição

normal. Consequentemente, um teste não-paramétrico deve ser utilizado. Na comparação de dois métodos, o teste de Postos com Sinais de Wilcoxon aponta os *benchmarks 04, 05 e 12* sem diferença significativa, ou seja, com resultados similares. Para os *benchmarks 06 e 07*, este teste aponta uma diferença significativa (98% e 99% respectivamente) entre os resultados encontrados.

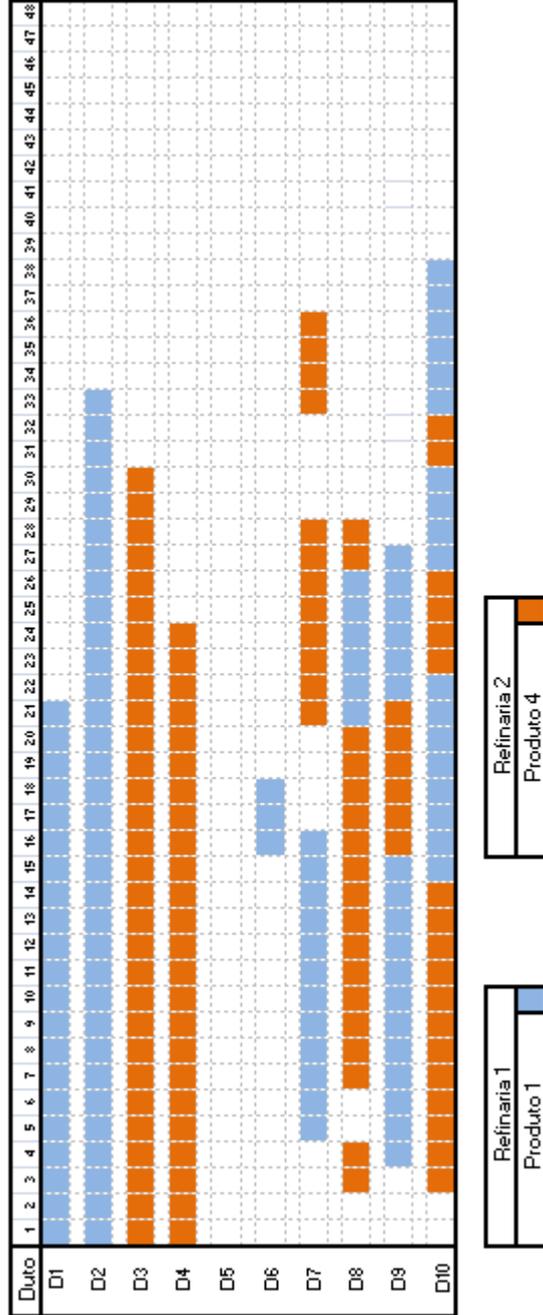


Figura 11: Melhor solução para instância 03

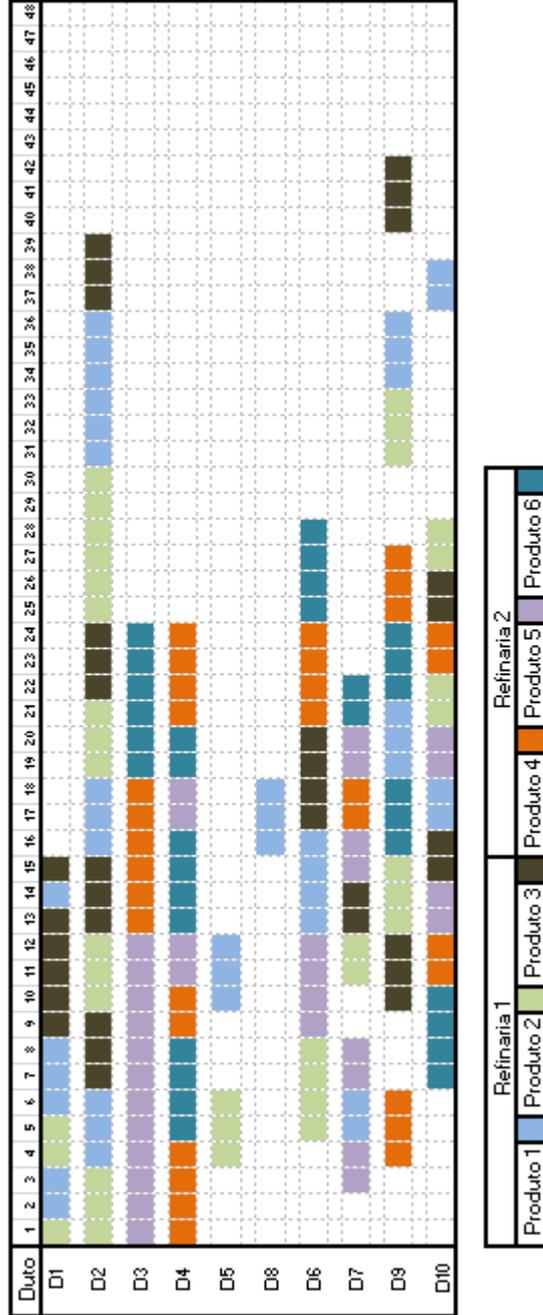


Figura 12: Melhor solução para instância 14

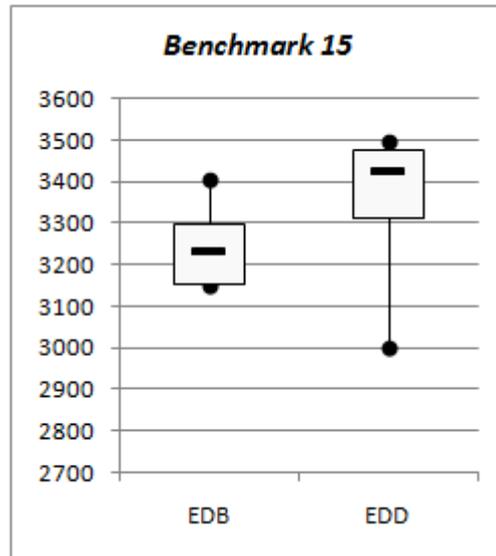


Figura 13: Boxplot - resultados BDE e DDE

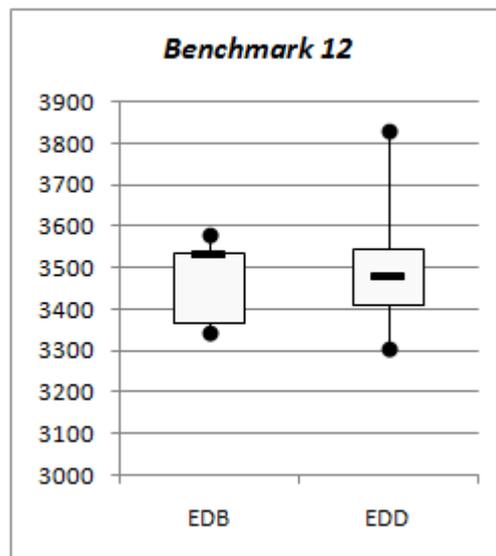


Figura 14: Boxplot - resultados BDE e DDE

5 CONCLUSÕES FINAIS

Neste trabalho foi proposto uma modelagem matemática discreta para um problema real. A aplicabilidade deste modelo em uma rede existente proporciona alternativas para a sequência de bateladas. Este modelo também permite o estudo de diferentes topologias de redes, com diferentes tamanhos de dutos e um número variado de produtos em diferentes horizontes de tempo. Porém, esta modelagem leva a um problema combinatorial complexo e a necessidade de um grande esforço computacional para resolvê-lo.

A otimização da rede de distribuição de derivados de petróleo é essencial para a redução do custo vinculado ao transporte e para o escoamento correto dos produtos aos clientes finais. Para exemplificar a aplicabilidade da modelagem proposta, a rede simplificada descrita na Seção 3.5 busca utilizar esta modelagem para a PLIM e para duas meta-heurísticas, a EDB e a EDD. Estas duas versões foram recentemente propostas na literatura e buscam comparar possíveis codificações da ED, ambas aplicadas para problemas binários.

Os *benchmarks* propostos apresentam um conjunto de regras e restrições que simulam uma situação real. Uma representação mais detalhada destas situações traz a possibilidade de criar novos estudos de casos. A dificuldade de representar todas as restrições do problema faz com que os *benchmarks* propostos sejam simplificados, pois diversas situações reais podem exigir o acréscimo de novas restrições e modificações na modelagem atual. A criação de uma modelagem matemática é imprescindível para a implementação de qualquer algoritmo na resolução do problema, seja através de uma programação matemática ou de qualquer meta-heurística.

Os experimentos realizados buscaram testar a modelagem proposta e atingiram o objetivo geral proposto na Subseção 1.2.1. Porém, somente os *benchmarks* 01, 02, 03, 04, 05 e 06 encontraram o valor ótimo para esta modelagem. Os outros *benchmarks* (07, 08, 09, 10, 11, 12, 13, 14 e 15) alcançaram valores sub-ótimos. Os objetivos específicos propostos na Subseção 1.2.2 também foram atingidos. A modelagem matemática apresentada permite a implementação de diferentes métodos, variando o número de refinarias, tanques de armazenamento, clientes fi-

nais, produtos e horizonte de tempo. A PLIM foi utilizada com sucesso na busca de soluções para a modelagem proposta e os resultados encontrados pelas meta-heurísticas apontam a EDB e a EDD como métodos viáveis que necessitam de um esforço computacional aceitável.

Os resultados da EDB e EDD também revelam o funcionamento de cada algoritmo. Os resultados de média e de desvio padrão da EDD apontam este algoritmo como um método eficaz de busca global. Esta característica é conhecida da ED devido ao seu processo de mutação, que utiliza a diferença vetorial para a perturbação dos indivíduos da população. Este processo de mutação e a codificação contínua foram mantidos no algoritmo de EDD e são os responsáveis pela diversidade dos resultados apresentados. Portanto, a EDD, quando comparada à EDB, apresenta diferentes (melhores e piores) soluções para o problema proposto. Tais resultados também sugerem que algoritmos concebidos para espaços contínuos podem ser eficientemente aplicados em problemas discretos. A utilização da função sigmóide no processo de discretização da EDD também sugere a aplicação desta meta-heurística para outros problemas combinatoriais binários e inteiros.

A alta complexidade do problema proposto é uma das principais motivações deste trabalho. A modelagem apresentada é uma inovadora e tem como objetivo adaptar o problema para espaços binários. Portanto, este problema discretizado é ideal para testar algoritmos heurísticos binários. Com um grande número de variáveis e diversas restrições, os algoritmos heurísticos necessitam de um bom equilíbrio entre a sua capacidade de busca global e busca local para encontrar soluções boas com um esforço computacional menor.

Este trabalho também contribui propondo *benchmarks* para o modelo simplificado. Estes *benchmarks* podem servir de ponto de partida para outros pesquisadores testarem novos métodos heurísticos. A possibilidade de criação de *benchmarks* novos e mais complexos também proporciona uma grande diversidade de estudos de casos. A busca por outras meta-heurísticas binárias para solucionar este problema proporciona novas linhas de pesquisa. A utilização de métodos de discretização de algoritmos contínuos para problemas discretos aumenta ainda mais o leque de possíveis meta-heurísticas aplicáveis a este problema.

5.0.1 TRABALHOS FUTUROS

As soluções ótimas dos *benchmarks 07, 08, 09, 10, 11, 12, 13, 14 e 15* propostos podem ser encontradas com mais tempo de processamento ou utilizando recursos computacionais mais avançados como supercomputadores com altíssima velocidade de processamento. A modelagem matemática apresentada também permite a criação de novos *benchmarks* com diferentes horizontes de tempo, produtos transportados e diversos dutos de diferentes tama-

nhos. A variação do número de refinarias, tanques de armazenamento e clientes finais sugere novos estudos de casos com diferentes topologias de redes. Outras meta-heurísticas como o AG, PSO, ABC e métodos híbridos também podem ser implementadas utilizando esta mesma modelagem. A utilização da EDD para outros problemas discretos também é um objetivo futuro. Outras versões do método com diferentes processos de discretização podem ser aplicados às mais diversas modelagens.

REFERÊNCIAS

- BABU, P. C. B.; MUBEEN, J. Multiobjective differential evolution (MODE) for optimization of adiabatic styrene reactor. **Chemical Engineering Science**, v. 60, n. 17, p. 4822–4837, 2005.
- CAFARO, D. C.; CERDÁ, J. Optimal scheduling of multiproduct pipeline systems using a non-discrete milp formulation. **Computers & Chemical Engineering**, v. 28, n. 10, p. 2053–2068, 2004.
- CAFARO, D. C.; CERDÁ, J. Dynamic scheduling of multiproduct pipelines with multiple delivery due dates. **Computers & Chemical Engineering**, v. 32, n. 4, p. 728–753, 2008.
- CAFARO, D. C.; CERDÁ, J. Rigorous scheduling of mesh-structure refined petroleum pipeline networks. **Computers & Chemical Engineering**, v. 38, n. 10, p. 185–203, 2012.
- CAFARO, V. G. et al. Detailed scheduling of operations in single-source refined products pipelines. **Industrial & Engineering Chemistry Research**, v. 50, n. 10, p. 6240–6259, 2011.
- DANTZIG, G. **Linear Programming and Extensions**. Princeton, USA: Princeton University Press, 1963.
- DAS, S.; SUGANTHAN, P. N. Differential evolution: A survey of the state-of-the-art. **IEEE Transactions on Evolutionary Computation**, v. 15, n. 1, p. 4–31, 2011.
- DORIGO, M. Ant colony optimization: A new meta-heuristic. In: **Proceedings of the Congress on Evolutionary Computation**. Piscataway, USA: IEEE Press, 1999. p. 1470–1477.
- FRASER, A. S. Simulation of genetic systems by automatic digital computers. II. effects of linkage on rates of advance under selection. **Australian Journal of Biological Science**, v. 10, p. 492–499, 1957.
- GARCIA, J. et al. Hybrid heuristic and mathematical programming in oil pipelines networks. In: **Evolutionary Computation**. Piscataway, USA: IEEE Press, 2004. v. 2, p. 1479–1486.
- GOLDBARG, M.; LUNA, H. **Otimização combinatória e programação linear: modelos e algoritmos**. Rio de Janeiro, RJ: Campus, 2000.
- HOLLAND, J. H. **Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence**. Ann Arbor, USA: University of Michigan Press, 1975.
- KARABOGA, D. **An idea based on honey bee swarm for numerical optimization**. Kayseri, Turquia: Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.
- KENNEDY, J.; EBERHART, R. C. Particle swarm optimization. In: **IEEE Congress on Evolutionary Computation**. Piscataway, USA: IEEE Press, 1995. v. 4, p. 1942–1948.

KONDILI, E.; PANTELIDES, C.; SARGENT, R. A general algorithm for short-term scheduling of batch operations. **Computers & Chemical Engineering**, v. 17, n. 2, p. 211 – 227, 1993.

KRAUSE, J.; CORDEIRO, J. A.; LOPES, R. S. P. H. S. A survey of swarm algorithms applied to discrete optimization problems. In: YANG, X. et al. (Ed.). **Swarm Intelligence and Bio-Inspired Computation: Theory and Applications**. New York, USA: Elsevier Science & Technology Books, 2013. v. 1, p. 169–192.

KRAUSE, J.; LOPES, H. Proposta de um algoritmo inspirado em evolução diferencial aplicado ao problema multidimensional da mochila. In: **Anais do Encontro Nacional de Inteligência Artificial**. Curitiba, PR: SBC, 2012.

KRAUSE, J.; PARPINELLI, R.; LOPES, H. A comparison of differential evolution algorithm with binary and continuous encoding for the MKP. In: **Proceedings of BRICS-CCI**. Recife, PE: IEEE Press, 2013.

LAND, A. H.; DOIG, A. G. An automatic method of solving discrete programming problems. **Econometrica**, v. 28, n. 3, p. 497–520, 1960.

LOPES, H. S.; RODRIGUES, L. C. de A.; STEINER, M. T. A. (Ed.). **Meta-Heurísticas em Pesquisa Operacional**. 1. ed. Curitiba, PR: Omnipax, 2013.

MAGATÃO, L.; ARRUDA, L. V. R.; JR., F. N. A mixed integer programming approach for scheduling commodities in a pipeline. **Computers & Chemical Engineering**, v. 28, n. 1-2, p. 171–185, 2004.

MAGATÃO, S. N. B. et al. Planning and sequencing product distribution in a real-world pipeline network: An MILP decomposition approach. **Industrial & Engineering Chemistry Research**, v. 51, n. 12, p. 4591–4609, 2012.

MICHALEWICZ, Z.; FOGEL, D. B. **How to Solve It: Modern Heuristics**. 2. ed. Berlin, Germany: Springer, 2004.

MIRHASSANI, S.; ABBASI, M.; MORADI, S. Operational scheduling of refined product pipeline with dual purpose depots. **Applied Mathematical Modelling**, v. 37, n. 8, p. 5723 – 5742, 2013.

MIRHASSANI, S. A.; MORADI, S.; TAGHINEZHAD, N. Algorithm for long-term scheduling of multiproduct pipelines. **Industrial & Engineering Chemistry Research**, v. 50, n. 24, p. 13899–13910, 2011.

NEVES, F. et al. An efficient approach to the operational scheduling of a real-world pipeline network. **Computer Aided Chemical Engineering**, v. 24, p. 697–702, 2007.

PRICE, K.; STORN, R.; LAMPINEN, J. **Differential Evolution: A Practical Approach to Global Optimization**. Heidelberg, Germany: Springer, 2005. (Natural Computing).

PUCCINI, A. L. **Programação linear**. Rio de Janeiro, RJ: Livros Técnicos e Científicos, 1989. (Aplicações de computadores).

REJOWSKI, R.; PINTO, J. M. Efficient MILP formulations and valid cuts for multiproduct pipeline scheduling. **Computers & Chemical Engineering**, v. 28, n. 8, p. 1511–1528, 2004.

RELVAS, S. et al. Pipeline scheduling and inventory management of a multiproduct distribution oil system. **Industrial & Engineering Chemistry Research**, v. 45, n. 23, p. 7841–7855, 2006.

SCHRAGE, L. **Optimization Modeling with LINGO**. Chicago, USA: Lindo Publishing, 2000.

SOUZA, T. C. N. de; GOLDBARG, E. F. G.; GOLDBARG, M. C. Transgenetic algorithm for the biobjective oil derivatives distribution problem. In: **IEEE Congress on Evolutionary Computation**. Piscataway, USA: IEEE Press, 2010. p. 1–8.

STEBEL, S. L. et al. Mixed integer linear programming formulation for aiding planning activities in a complex pipeline network. **Industrial & Engineering Chemistry Research**, v. 51, n. 35, p. 11417–11433, 2012.

STORN, R.; PRICE, K. **Differential Evolution: A simple and efficient adaptive scheme for global optimization over continuous spaces**. Berkeley, USA: International Computer Science Institute, Berkeley University, 1995.

WESTPHAL, H.; NEVES-JR., F.; ARRUDA, L. V. R. de. Computação evolucionária em problemas de engenharia. In: LOPES, H. S.; TAKAHASHI, R. H. C. (Ed.). 1. ed. Curitiba (PR): Omnipax, 2011. cap. Algoritmo micro-genético aplicado ao scheduling de uma rede de distribuição de derivados de petróleo, p. 331–354.

YONGTU, L.; MING, L.; NI, Z. A study on optimizing delivering scheduling for a multiproduct pipeline. **Computers & Chemical Engineering**, v. 44, n. 9, p. 127–140, 2012.

ZONTA, T. et al. Inteligência computacional aplicada a resolução do problema do corte unidimensional. **Revista Eletrônica de Sistemas de Informação**, v. 4, n. 2, p. 1677–3071, 2005.

APÊNDICE A – SOLUÇÕES ÓTIMAS (01, 02, 04, 05 E 06) E SUB-ÓTIMAS (07, 08, 09, 10, 11, 12, 13 E 15) DA PLIM

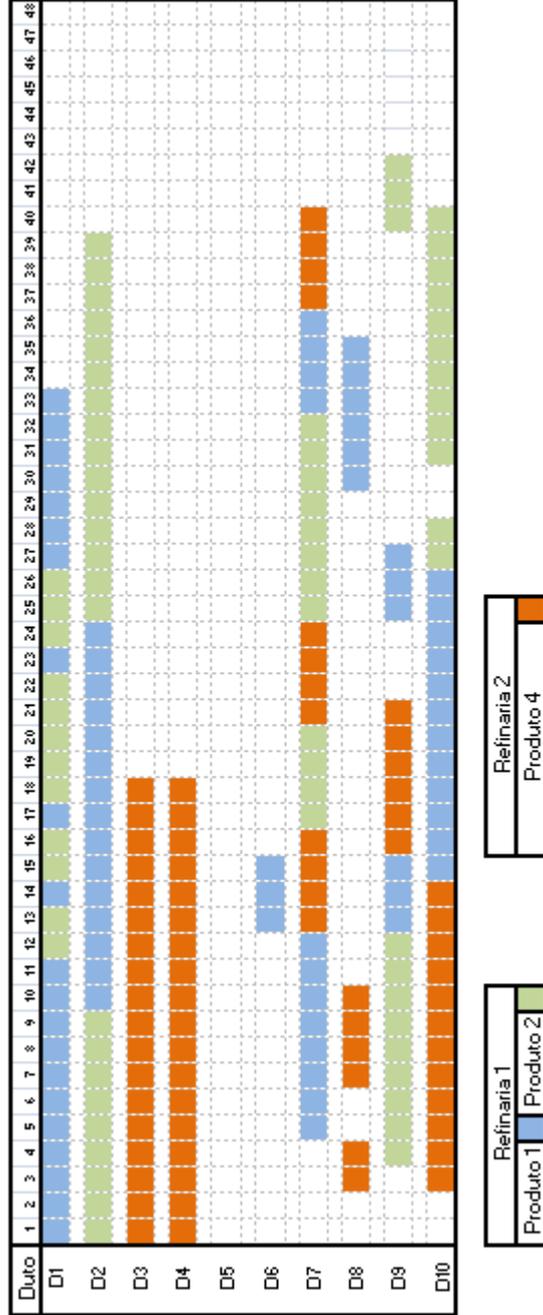


Figura 18: Melhor solução para instância 05

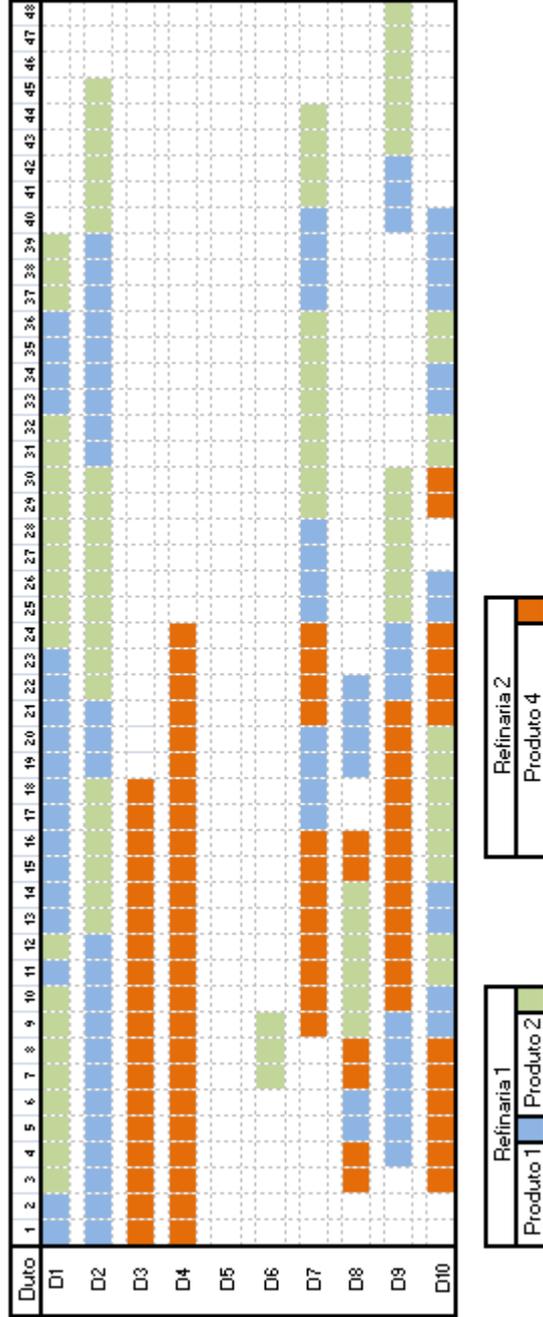


Figura 19: Melhor solução para instância 06

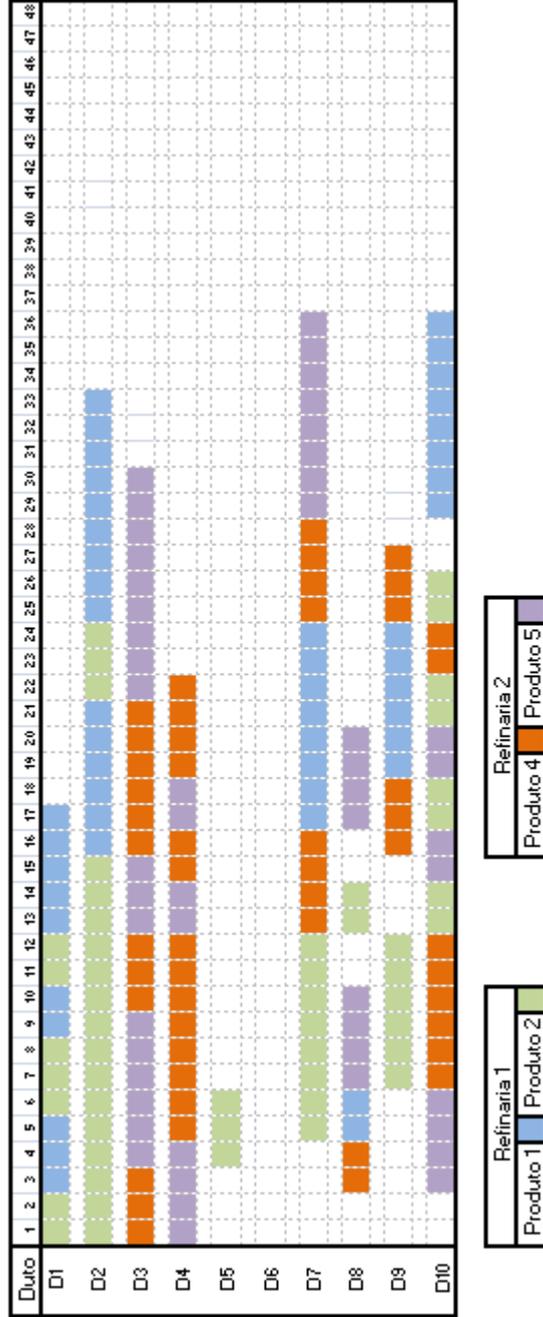


Figura 20: Melhor solução para instância 07

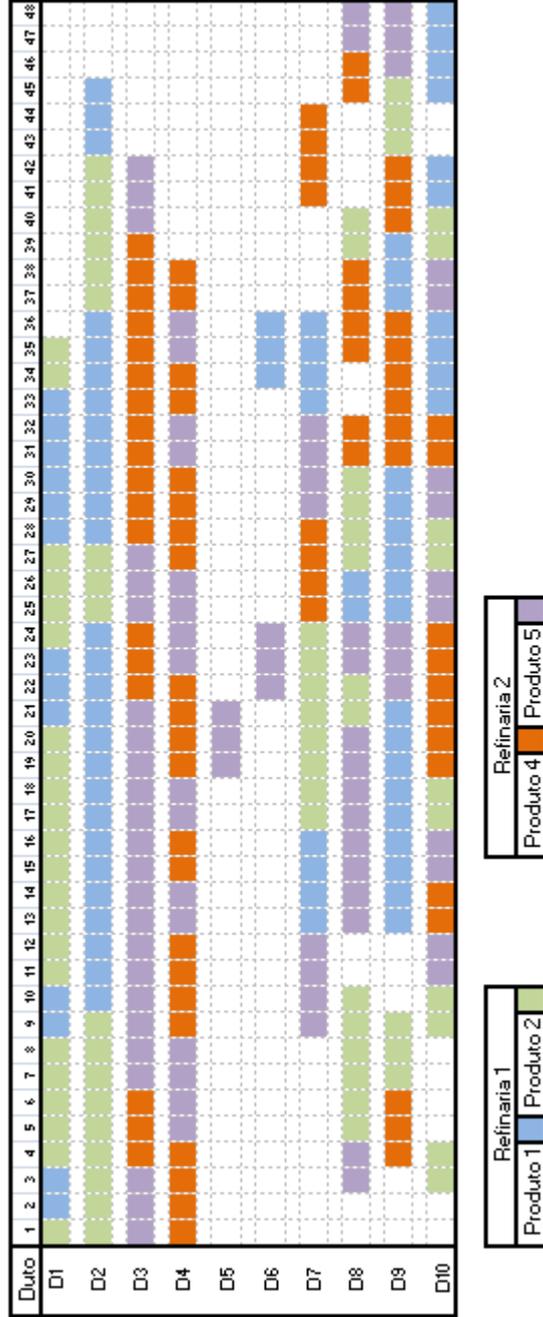


Figura 22: Melhor solução para instância 09

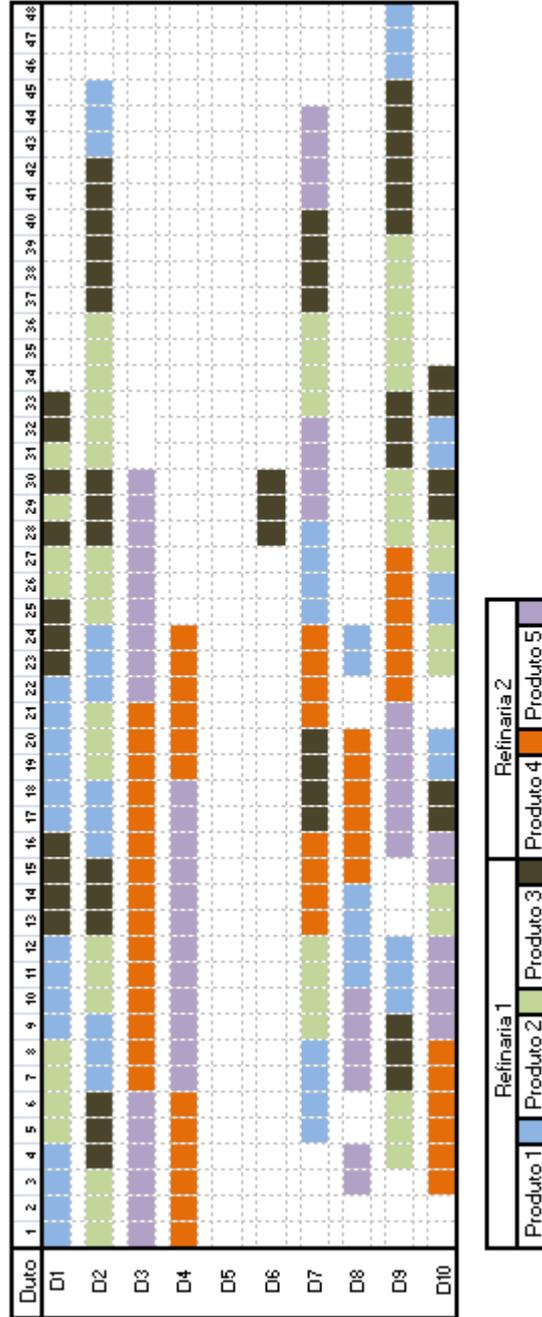


Figura 24: Melhor solução para instância 11

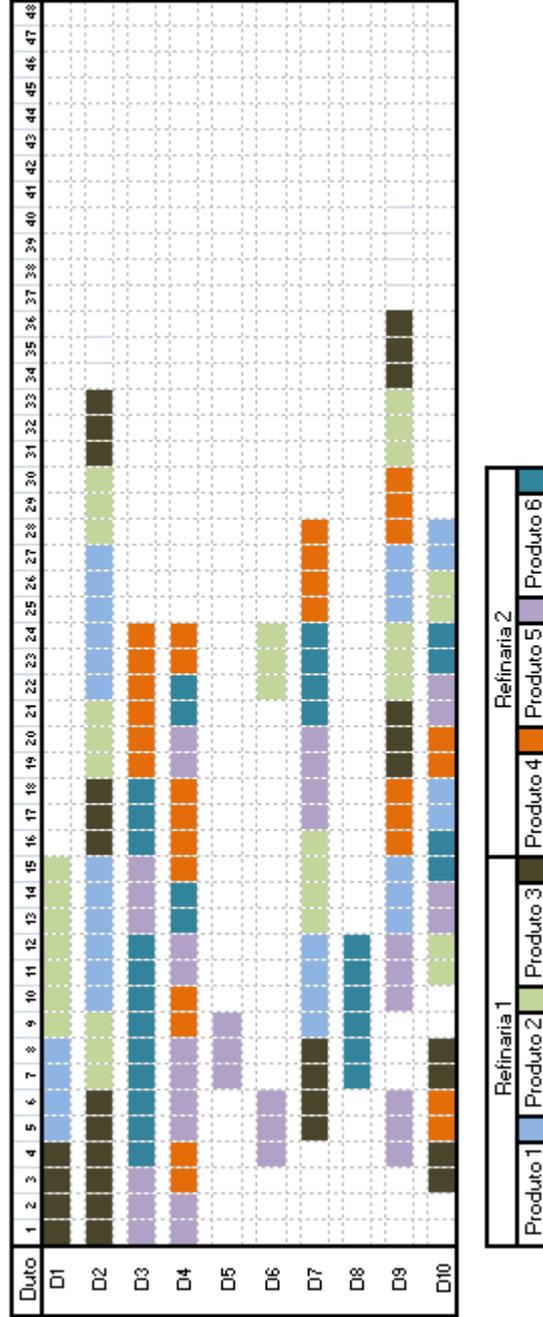


Figura 26: Melhor solução para instância 13

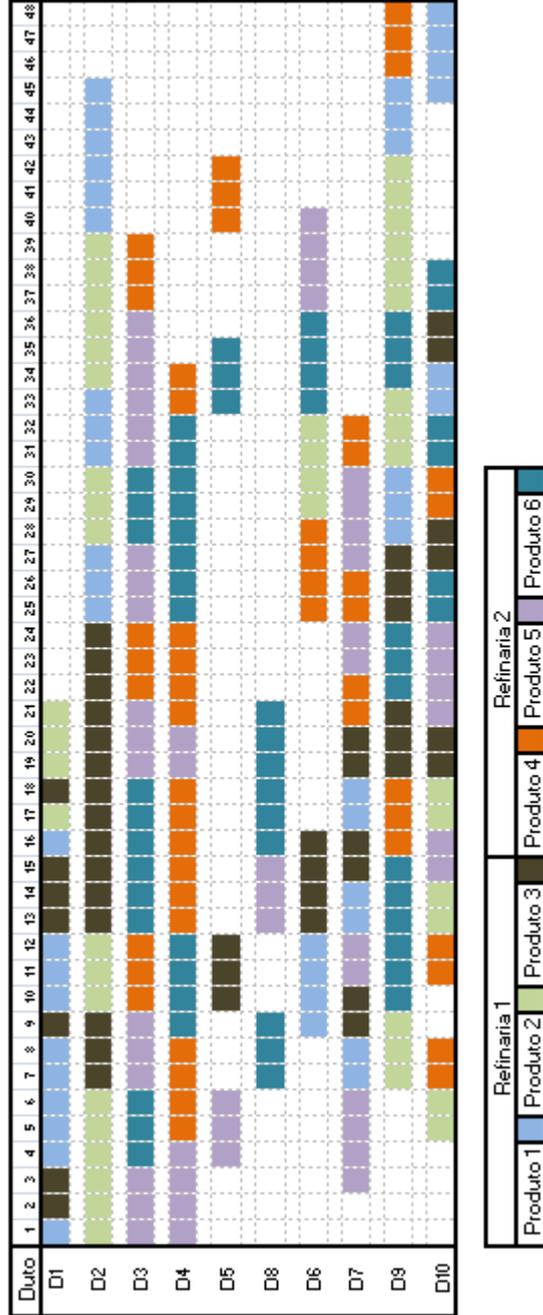


Figura 27: Melhor solução para instância 15

**APÊNDICE B – BOXPLOTS (01, 02, 03, 08, 09, 10, 11 E 13) - DIFERENÇA
SIGNIFICATIVA**

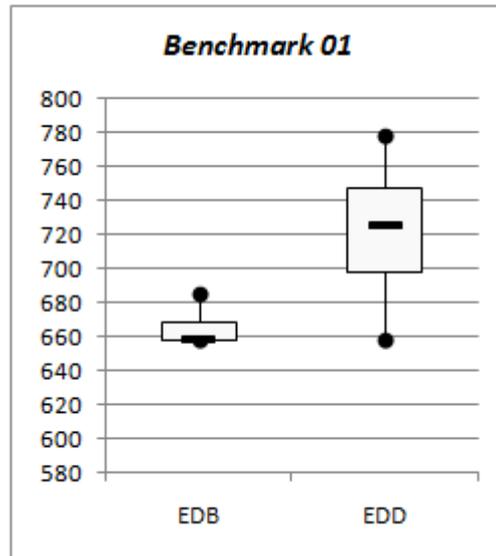


Figura 28: Boxplot - resultados BDE e DDE

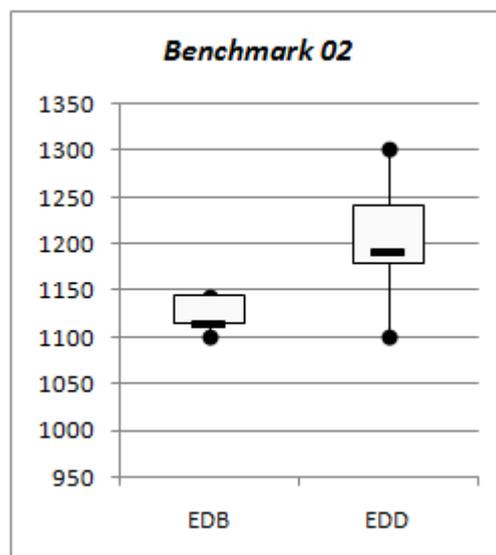


Figura 29: Boxplot - resultados BDE e DDE

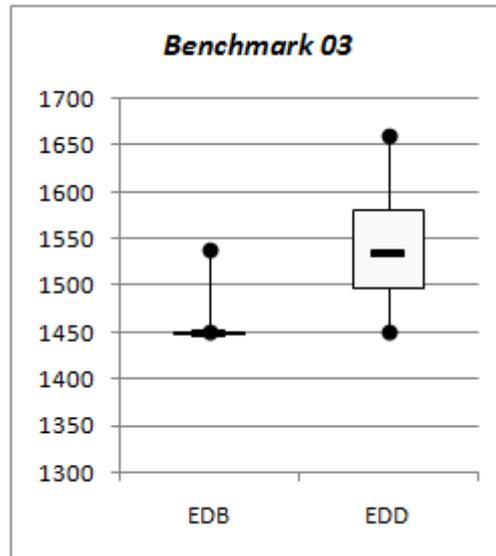


Figura 30: Boxplot - resultados BDE e DDE

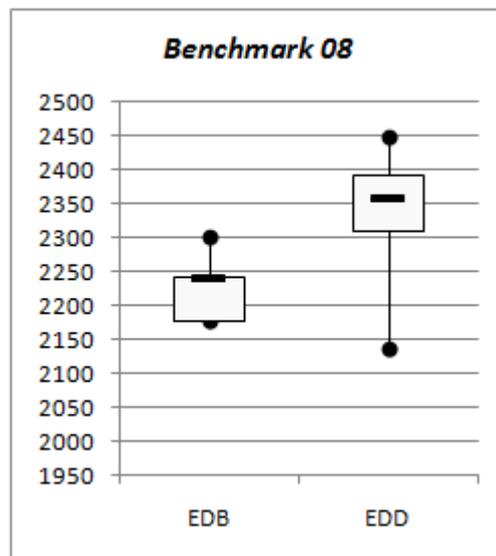


Figura 31: Boxplot - resultados BDE e DDE

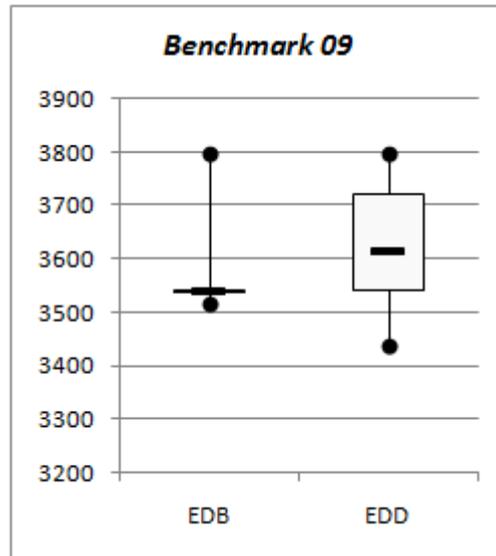


Figura 32: Boxplot - resultados BDE e DDE

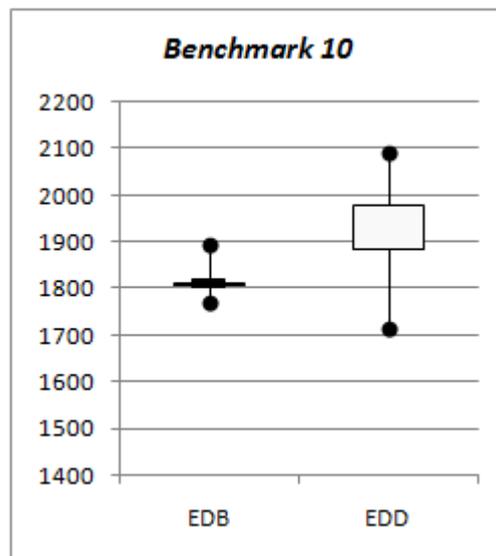


Figura 33: Boxplot - resultados BDE e DDE

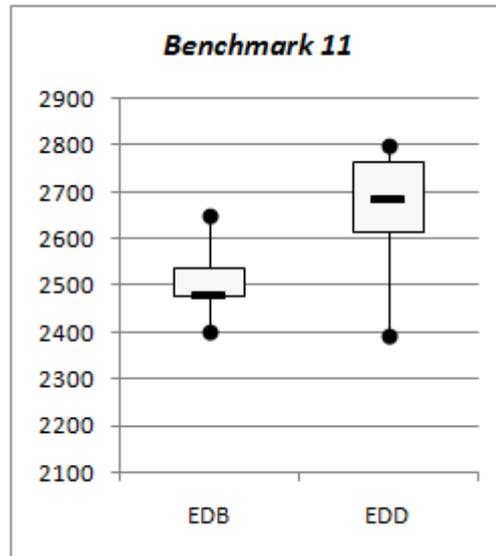


Figura 34: Boxplot - resultados BDE e DDE

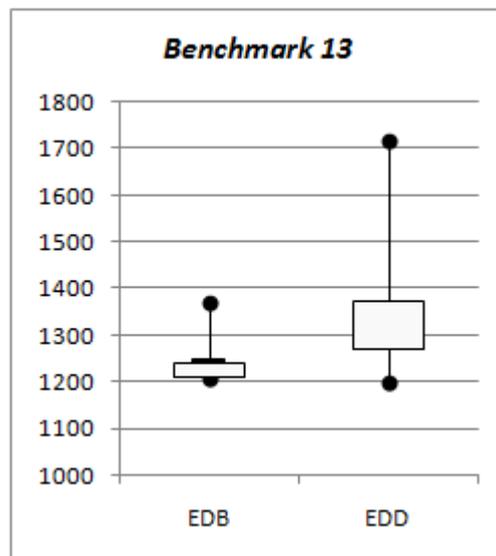


Figura 35: Boxplot - resultados BDE e DDE

APÊNDICE C – BOXPLOTS (04, 05, 06 E 07) - DIFERENÇA PODE NÃO SER SIGNIFICATIVA

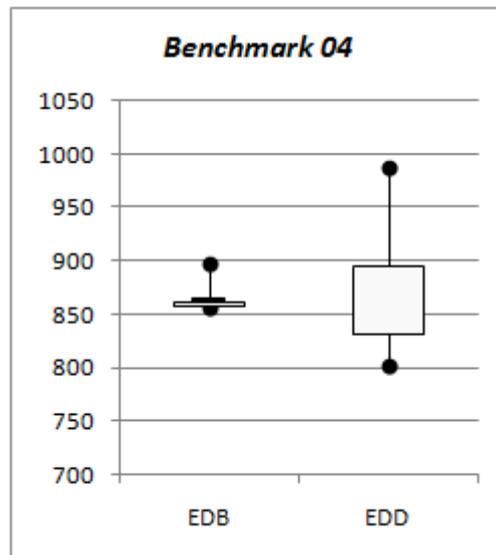


Figura 36: Boxplot - resultados BDE e DDE

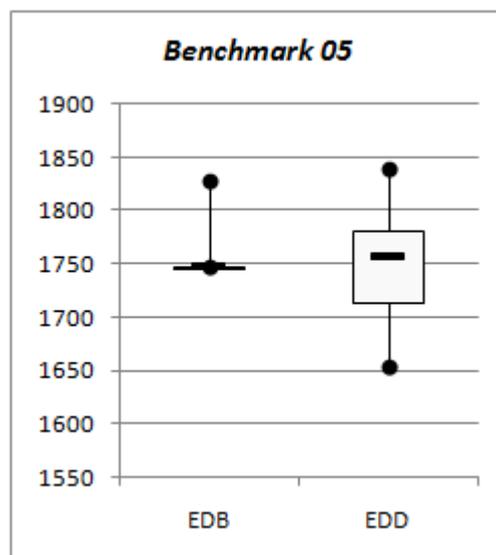


Figura 37: Boxplot - resultados BDE e DDE

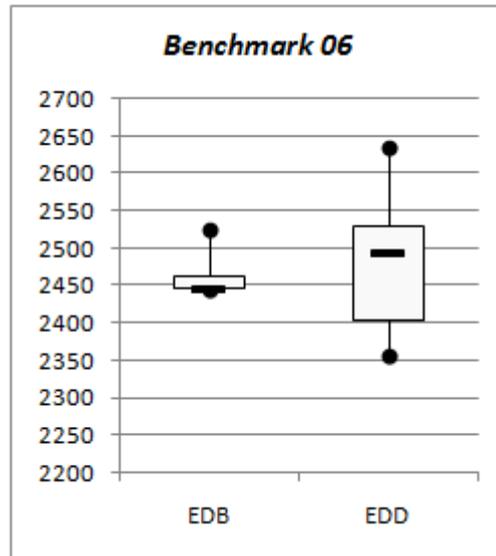


Figura 38: Boxplot - resultados BDE e DDE

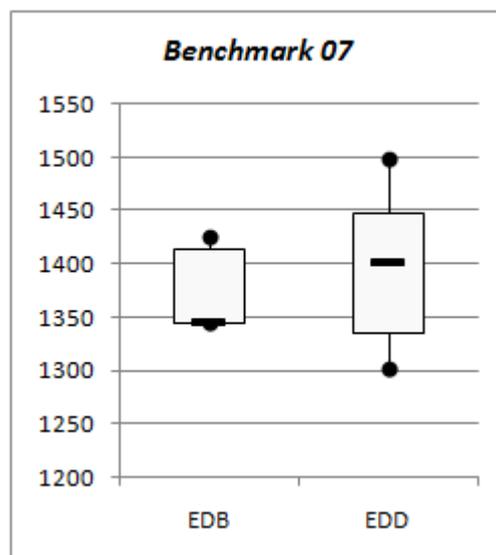


Figura 39: Boxplot - resultados BDE e DDE