

**UNIVERSIDADE FEDERAL DE OURO PRETO
INSTITUTO DE CIÊNCIAS EXATAS E BIOLÓGICAS
DEPARTAMENTO DE COMPUTAÇÃO**

**Uma abordagem multiobjetivo para o problema de
planejamento operacional de lavra: Parte II**

Relatório Final, referente ao período março de 2012 a fevereiro de 2013, apresentado à Universidade Federal de Ouro Preto, como parte das exigências do programa de iniciação científica - PROBIC/FAPEMIG.

Equipe:

Marcone Jamilson Freitas Souza (DECOM/UFOP) - Orientador

Alexandre Costa Barbosa (Bolsista de Iniciação Científica / FAPEMIG)

Vitor Nazário Coelho (Bolsista do Programa Ciência sem Fronteiras)

Igor Machado Coelho (Mestre em Ciência da Computação / IC-UFF) - Co-orientador

Data de início do projeto: 01/03/2012

Data de fim do projeto: 28/02/2013

Ouro Preto - Minas Gerais - Brasil

01 de Abril de 2013

Resumo

Este trabalho tem seu foco no planejamento operacional de lavra em minas a céu aberto. O problema tratado consiste na mistura de minérios provenientes de várias frentes de lavra, para formar um produto, levando-se em consideração vários objetivos conflitantes, como: obtenção das metas de produção e qualidade para o produto formado, e minimização do número de veículos necessários ao processo produtivo. Considera-se o sistema de alocação dinâmica de caminhões, o que significa que, após as descargas nos pontos de basculamento, cada caminhão pode se dirigir a uma frente diferente para novo carregamento, aumentando a produtividade da frota. Na abordagem multiobjetivo não há uma única solução que satisfaça a todos os objetivos. O que se procura é um conjunto de soluções não-dominadas, também chamadas de soluções eficientes, ou Fronteira de Pareto, cabendo ao tomador de decisões a escolha da solução mais adequada. Neste trabalho foi desenvolvido um novo algoritmo heurístico multiobjetivo baseado em busca local, denominado GRASP-2PPLS. Ele combina os procedimentos *Greedy Randomized Adaptive Search Procedures* (GRASP) e *Two-phase Pareto local search* (2PPLs). Assim como nos trabalhos anteriores, o algoritmo foi aplicado de forma a determinar um conjunto de soluções não-dominadas de boa qualidade e em um tempo computacional aceitável para a tomada de decisão em situações práticas. A fase de construção do procedimento GRASP é usada para gerar a população inicial dos algoritmos. As aproximações para a Fronteira de Pareto geradas pelos algoritmos desenvolvidos foram comparadas entre si tendo em vista as métricas de hipervolume, cobertura e espaçamento. Os resultados computacionais encontrados mostraram a superioridade do algoritmo GRASP-2PPLS quando comparado aos dois algoritmos desenvolvidos na pesquisa anterior, GRASP-MOVNS e GRASP-NSGAI-PR. O algoritmo proposto neste trabalho foi capaz de encontrar Fronteiras de Pareto mais diversificadas e com uma melhor qualidade nas soluções. Por fim, ressalta-se a combinação dos procedimentos 2PPLS e o MOVNS, que obteve bons conjuntos referências.

Palavras-chave: Planejamento operacional de lavra, Metaheurísticas, GRASP, NSGA-II, Otimização Multiobjetivo

Alexandre Costa Barbosa

Bolsista

Marcone Jamilson Freitas Souza

Orientador

Sumário

Lista de Figuras

Lista de Algoritmos p. 7

Lista de Tabelas p. 8

1 Introdução p. 9

1.1 O Problema de Planejamento Operacional de Lavra p. 9

1.2 Objetivos do trabalho p. 11

1.3 Estrutura do trabalho p. 12

2 Revisão Bibliográfica p. 13

3 Heurísticas p. 16

3.1 Metaheurísticas p. 16

3.2 Otimização Mono-objetivo p. 17

3.2.1 Introdução p. 17

3.2.2 Heurísticas Mono-objetivo p. 18

3.2.2.1 *Greedy Randomized Adaptive Search Procedure* (GRASP) p. 19

3.3 Otimização Multiobjetivo p. 20

3.3.1 Introdução p. 20

3.3.2 Metas da Otimização Multiobjetivo p. 22

3.3.3 Diferenças com a otimização Mono-objetivo p. 22

3.3.4 Heurísticas de Otimização Multiobjetivo p. 23

3.3.4.1	Computação Evolucionária	p. 23
3.3.4.2	Two-phase Pareto local search (2PPLs)	p. 24
3.3.5	Métricas de Avaliação de Desempenho	p. 26
3.3.5.1	Hipervolume	p. 27
3.3.5.2	Espaçamento	p. 29
3.3.5.3	Cobertura	p. 30
3.3.5.4	Cardinalidade	p. 30
4	O Planejamento Operacional de Lavra Abordado	p. 31
4.1	Introdução	p. 31
4.2	Características do Problema de Alocação Abordado	p. 34
5	Metodologia Heurística	p. 36
5.1	Representação de uma solução	p. 36
5.2	Geração de uma solução inicial	p. 37
5.3	Estruturas de vizinhança	p. 40
5.4	Função de avaliação	p. 43
5.5	Algoritmo proposto	p. 44
5.6	Algoritmos e dados para geração de novos problemas-teste	p. 45
6	Resultados Parciais	p. 51
6.1	Descrição dos problemas-teste	p. 51
6.2	Pesos e parâmetros utilizados	p. 52
6.3	Ambiente de desenvolvimento	p. 52
6.4	Resultados e análise	p. 53
6.5	Geração de novos problemas-teste	p. 56
6.5.1	Resultados computacionais aplicados ao novo problema-teste	p. 58

7 Conclusões e Trabalhos Futuros	p. 61
Referências	p. 63
Anexos	p. 67

Lista de Figuras

1	Soluções Pareto ótimo locais e globais (ALEXANDRE, 2010)	p. 22
2	Metas da Otimização Multiobjetivo (DEB, 2001)	p. 27
3	Distribuição $\times$ Convergência - 1 (DEB, 2001)	p. 28
4	Distribuição $\times$ Convergência - 2 (DEB, 2001)	p. 28
5	Hipervolume gerado pelas soluções não-dominadas de um Fronteira de Pareto hipotética	p. 29
6	Equipamentos de carga e transporte	p. 31
7	Britador	p. 32
8	Modelo de Caminhão (ARAÚJO, 2008)	p. 33
9	Modelo de Carregadeira - L1850 (ARAÚJO, 2008)	p. 33
10	Exemplo de operação em uma mina a céu aberto	p. 35
11	Exemplo de Solução para o POLAD	p. 42
12	Exemplo de aplicação dos movimentos	p. 42
13	Conjunto com 47 soluções não-dominadas - opm90	p. 60

Lista de Algoritmos

1	Greedy Randomized Adaptive Search Procedure	p. 19
2	Construção GRASP	p. 20
3	2PPLs com VNS	p. 25
4	addSolution	p. 26
5	ConstróiSoluçãoEstéril	p. 38
6	ConstróiSoluçãoMinério	p. 40
7	2PPLs com VNS	p. 45
8	ConstroiConjuntoInicial	p. 45
9	GeraProblemaTeste	p. 48
10	geraDadosBasicos	p. 49

Lista de Tabelas

1	Exemplo de características de uma solução para o POLAD	p. 37
2	Parâmetros	p. 46
3	Limites dos Parâmetros	p. 46
4	Frota de Caminhões	p. 47
5	Frota de Carregadeiras	p. 47
6	Compatibilidade Caminhões x Carregadeiras x Descargas	p. 47
7	Melhores valores	p. 51
8	Pesos adotados	p. 52
9	GRASP-2PPLS $\times$ GRASP-MOVNS $\times$ GRASP-NSGAI I-PR: Hipervolume	p. 54
10	GRASP-2PPLS $\times$ GRASP-MOVNS $\times$ GRASP-NSGAI I-PR: Espaçamento	p. 54
11	GRASP-2PPLS $\times$ GRASP-MOVNS: Cobertura	p. 55
12	GRASP-2PPLS $\times$ GRASP-MOVNS: Cardinalidade	p. 56
13	Comparação de resultados: GRASP-2PPLS $\times$ GGVNS	p. 56
14	Pesos geração problemas-teste	p. 57
15	Metas de produção minério e estéril - opm90	p. 57
16	Frota de Caminhões - opm90	p. 58
17	Frota de Carregadeiras - opm90	p. 58
18	Frentes Minério e Estéril - opm90	p. 59
19	Soluções CPLEX	p. 60
20	Comparação de resultados: CPLEX $\times$ GGVNS $\times$ GRASP-MOVNS	p. 60

1 Introdução

1.1 O Problema de Planejamento Operacional de Lavra

As mineradoras realizam suas atividades em minas subterrâneas ou a céu aberto. Em minas a céu aberto as atividades de carregamento e transporte ocorrem da seguinte maneira: os caminhões se deslocam até a frente de lavra, que são os pontos da mina onde o minério e o estéril são retirados, são carregados pelos equipamentos de carga e em seguida se dirigem aos pontos de descarga, onde descarregam o minério e o estéril. Os pontos de descarga podem ser pilhas de estéril, material que não é aproveitado pelo processo; pilhas de homogeneização, quando é transportada uma quantidade de minério maior do que a usina pode beneficiar ou quando é necessário “misturar” os minérios antes de iniciar o beneficiamento, e usina de tratamento, onde se inicia o beneficiamento de minério.

Para fornecer minério de qualidade uniforme para o processo é necessário misturar minério de diferentes qualidades proveniente de várias partes da mina ou de diferentes minas com o objetivo de assegurar a uniformidade da alimentação, já que mudanças são usualmente acompanhadas de aumento do custo total da operação (CHANDA; DAGDELEN, 1995).

A atividade de transporte de material é um dos mais importantes aspectos na operação de minas a céu aberto (ALARIE; GAMACHE, 2002). Segundo Maran e Topuz (1988), sistemas de transporte nessas minas envolvem grande volume de capital e recursos. O objetivo do problema de transporte é mover o material retirado da mina para a usina de modo que o custo seja minimizado, uma vez que o custo associado influencia a escolha de onde retirar minério (GERSHON, 1982).

Minas a céu aberto utilizam dois critérios para o transporte de material por caminhões: alocação estática e alocação dinâmica. Na alocação estática, os caminhões seguem uma trajetória fixa entre um ponto de carga e outro de descarga, ou seja, os caminhões ficam fixos a esses dois pontos durante um determinado período de tempo. Já na alocação dinâmica, os caminhões não ficam vinculados a uma mesma rota; assim, a cada descarga, o caminhão pode ser direcionado

a um ponto de carga não necessariamente o mesmo da viagem anterior.

A alocação estática é o método mais utilizado nas minerações de pequeno e médio porte por não apresentar a obrigatoriedade de utilização de um sistema automático de alocação, conhecido como sistema de despacho. Esse método, entretanto, proporciona menor produtividade em função da possibilidade de formação de filas de caminhões e ociosidade dos equipamentos de carga (RODRIGUES, 2006).

A vantagem da alocação dinâmica de caminhões é que com essa estratégia há uma maior produtividade da frota. Esse aumento de produtividade pode refletir um aumento na produção da mina ou a redução do número de equipamentos necessários para manter o mesmo nível de produção. Um algoritmo eficiente para a alocação dinâmica de caminhões é importante porque ele integra um sistema de despacho computadorizado. Um sistema de despacho reúne, ainda, um algoritmo de sequenciamento de viagens, um sistema de comunicação entre os equipamentos de carga e caminhões e uma central de comandos. Segundo White e Olson (1986), para que o sistema de despacho de caminhões seja completo é importante que o sistema de monitoramento dos equipamentos seja preciso e confiável, de modo que as operações da mina possam ser otimizadas em tempo real.

O custo de instalação de sistemas de despacho depende do tamanho da mina e do tipo de operação. Esse custo inibia a sua utilização por mineradoras de pequeno e médio porte. A partir da década de 90, em consequência da evolução da informática, o custo desses sistemas foi consideravelmente reduzido. Essa redução no custo levou ao aumento no número de mineradoras e empreiteiras que utilizam esse tipo de sistema. Segundo Rodrigues (2006), atualmente cerca de 35 minas fazem uso desses sistemas no Brasil, com diferentes níveis de automação.

Ao contrário das abordagens anteriores, em que o POLAD era tratado como um problema de otimização mono-objetivo com soma ponderada de três objetivos, pretende-se no presente projeto fazer uma abordagem multiobjetivo ao mesmo. O projeto visa, assim, a estudar, desenvolver e implementar algoritmos de otimização multiobjetivo para o POLAD. Espera-se conceber um algoritmo que seja capaz de produzir soluções aproximadas para o conjunto Pareto-ótimo, deixando à escolha do tomador de decisão a solução mais atrativa para os interesses da empresa

No presente trabalho, tem-se como foco o problema de planejamento operacional de lavra, considerando alocação dinâmica de caminhões, doravante referenciado por POLAD. Tradicionalmente, o POLAD tem sido tratado como um problema de otimização mono-objetivo com soma ponderada de três objetivos: a minimização dos desvios de qualidade, a minimização dos desvios de produção e a minimização do número de caminhões necessários ao processo. Neste

trabalho propõe-se tratá-lo por uma abordagem multiobjetivo. Desta forma, o que se procura é um conjunto de soluções não-dominadas, também chamadas de soluções eficientes, ou Fronteira de Pareto, cabendo ao tomador de decisões a escolha da solução mais adequada às suas necessidades.

1.2 Objetivos do trabalho

Este trabalho teve como objetivo geral desenvolver um algoritmo multiobjetivo eficiente para resolver o problema de planejamento operacional de lavra com alocação dinâmica de caminhões (POLAD).

Os objetivos específicos estabelecidos foram os seguintes:

- (a) Fazer uma revisão de literatura sobre os métodos utilizados para resolver o problema de planejamento de lavra em minas a céu aberto;
- (b) Fazer uma revisão de literatura sobre técnicas de otimização discreta multiobjetivo;
- (c) Desenvolver um algoritmo de otimização multiobjetivo utilizando o procedimento *Two-phase Pareto local search* (2PPLs);
- (d) Testar os métodos desenvolvidos, sempre que possível, em casos reais da indústria extrativa brasileira;
- (e) Criação de um novo conjunto de problemas-teste;
- (f) Produzir um artigo que possa ser apresentado e publicado nos anais de um evento científico nacional;
- (g) Contribuir com a divulgação de técnicas de otimização aplicadas à resolução do problema, possibilitando à indústria extrativa nacional melhorar sua produtividade e tornar-se mais competitiva;
- (h) Contribuir com a formação de recursos humanos especializados nessa área do conhecimento;
- (i) Contribuir para a consolidação das linhas de pesquisa “Otimização e simulação de operações de lavra em minas a céu aberto e subterrâneas” e “Otimização Combinatória” do grupo de Logística e Pesquisa Operacional da UFOP;

1.3 Estrutura do trabalho

Este trabalho está dividido em sete capítulos, incluindo esta introdução, onde o problema de planejamento operacional de lavra é contextualizado.

No Capítulo 2 é apresentada uma revisão bibliográfica sobre os diversos métodos utilizados na resolução do POLAD, bem como a forma com que diversos autores tratam esse problema.

No Capítulo 3 são descritos os principais conceitos e características dos Algoritmos Mono-objetivo e, também, são definidos alguns conceitos de algoritmos multiobjetivo, em especial, o método *Two-phase Pareto local search* (2PPLs), adaptado neste trabalho para a resolução do POLAD. Por fim, é apresentada uma breve revisão sobre os métodos de comparação e validação, de forma a testar a eficiência dos algoritmos desenvolvidos.

No Capítulo 4 é apresentado o problema abordado em detalhes.

No Capítulo 5 são descritos os algoritmos desenvolvidos para resolver o POLAD.

No Capítulo 6 são apresentados os resultados parciais da pesquisa, utilizando o algoritmo multiobjetivo desenvolvido neste trabalho.

Por fim, no Capítulo 7 são apresentadas as conclusões e apontados os trabalhos futuros.

2 *Revisão Bibliográfica*

White e Olson (1986) propuseram um algoritmo que é a base para o sistema *DISPATCH*, que vem operando em muitas minas em todo o mundo. Uma solução é obtida em duas etapas. Na primeira, baseada em programação linear, realiza-se uma otimização do problema da mistura de minérios tendo como objetivo a minimização de uma função de custo que considera o ritmo de lavra, a qualidade da mistura, o atendimento às taxas de alimentação da usina de beneficiamento e o remanuseio de material. As restrições do modelo estão relacionadas às capacidades de produção dos equipamentos de carga, à qualidade da mistura e às taxas de alimentação mínima requerida da usina de beneficiamento. A segunda etapa do algoritmo, a qual é resolvida por programação dinâmica, usa um modelo semelhante ao de White, Arnold e Clevenger (1982), diferenciando-se deste por utilizar como variável de decisão o volume de material transportado por hora em uma determinada rota, ao invés da taxa de caminhões por hora. É considerada, ainda, a presença de pilhas de estocagem. Nesta segunda etapa do algoritmo, o objetivo é minimizar a necessidade de transporte de material na mina.

Chanda e Dagdelen (1995) desenvolveram um modelo de programação linear por metas para resolver um problema de mistura de minérios no planejamento de curto prazo em uma mina de carvão. A função objetivo do modelo consistia na soma ponderada de três objetivos distintos: maximizar um critério econômico, minimizar os desvios de produção requeridos e minimizar os desvios de qualidade relativos aos valores desejados para os parâmetros de controle. Nenhuma alocação de equipamento de carga e transporte foi considerada nesse modelo.

Alvarenga (1997) desenvolveu um programa para o despacho ótimo de caminhões em uma mineração de ferro, a céu aberto, com o objetivo de minimizar o tempo de fila da frota de caminhões, aumentar a produtividade desta e melhorar a qualidade do minério lavrado. No trabalho desenvolvido, que é base do sistema SMART MINE, atualmente muito utilizado em várias minas brasileiras, foi aplicada uma técnica estocástica de otimização, o algoritmo genético com processamento paralelo.

Merschmann (2002) desenvolveu um sistema de otimização e simulação para análise de

cenário de produção em minas a céu aberto. O sistema, denominado OTISIMIN (Otimizador e Simulador para Mineração), foi desenvolvido em dois módulos. O primeiro corresponde ao módulo de otimização onde um modelo de programação linear foi construído e resolvido e o segundo a um módulo de simulação que permite ao usuário utilizar os resultados obtidos na resolução do modelo de programação linear como dados de entrada para a simulação. O módulo de otimização foi elaborado com o objetivo de otimizar o processo de mistura de minérios oriundos das várias frentes de lavra de forma a atender as especificações de qualidade impostas pela usina de tratamento e realizar a alocação de equipamentos (caminhões, carregadeiras e/ou escavadeiras) às frentes de lavra, considerando tanto alocação dinâmica quanto estática dos caminhões. O modelo de otimização desenvolvido não considera metas de produção e qualidade, nem a redução do número de caminhões necessários ao sistema de produção.

Em Costa, Souza e Pinto (2004) e Costa, Souza e Pinto (2005) foram apresentados e modelados problemas relativos à mistura de minérios provenientes de várias frentes de lavra, levando-se em consideração metas de produção e qualidade, alocação dinâmica e estática de caminhões, restrições operacionais e alocação dos equipamentos de carga e transporte necessários ao processo. Os modelos considerados foram baseados em programação linear por metas e representaram um avanço em relação àqueles de Merschmann (2002), isto porque, além de contemplarem mais situações reais, reduziam significativamente o número de restrições do problema.

Como relatado nesses trabalhos, o POLAD é um problema da classe NP-difícil e, como tal, métodos exatos de solução têm aplicabilidade restrita. Desta forma, a abordagem mais comum a ser utilizada passou a ser feita por meio de procedimentos heurísticos, como relatado em Costa (2005), que desenvolveu um algoritmo heurístico baseado em *Greedy Randomized Adaptive Search Procedures* - GRASP (FEO; RESENDE, 1995; RESENDE; RIBEIRO, 2003, 2010) e VNS (MLADENOVIC; HANSEN, 1997; HANSEN; MLADENOVIC, 2001) para o POLAD usando seis tipos diferentes de movimentos para explorar o espaço de soluções. Foi feita uma comparação entre os resultados obtidos por esse algoritmo heurístico e os encontrados pelo otimizador LINGO, versão 7, aplicado a um modelo de programação matemática desenvolvida pelos autores, publicado em (COSTA; SOUZA; PINTO, 2004). Mostrou-se que o algoritmo heurístico desenvolvido foi capaz de encontrar soluções de melhor qualidade mais rapidamente.

Guimarães, Pantuza e Souza (2007) apresentaram um modelo de simulação computacional para validar resultados obtidos pela aplicação de um modelo de programação matemática na determinação do ritmo de lavra em minas a céu aberto. Dessa maneira, foi possível validar os resultados da otimização, já que na modelagem de otimização não é possível tratar a variabilidade nos tempos de ciclo e a ocorrência de fila.

Em Coelho, Ribas e Souza (2008), o POLAD é resolvido por um algoritmo heurístico, denominado GVILS, que combina os procedimentos heurísticos GRASP, VND e ILS (LOURENÇO; MARTIN; STÜTZLE, 2003). O algoritmo GVILS faz uso de oito movimentos para explorar o espaço de soluções. Além dos desvios de produção e qualidade, procurou-se minimizar, também, o número de veículos. Usando quatro problemas-teste da literatura, o GVILS foi comparado com o otimizador CPLEX 9.1 aplicado a um modelo de programação matemática. Foram realizados testes envolvendo 15 minutos de processamento. Em dois dos problemas, o algoritmo proposto mostrou-se bastante superior; enquanto nos dois outros ele foi competitivo com o CPLEX, produzindo soluções médias com valores até 0,08% piores, na média.

Souza et al. (2010) propuseram um algoritmo, denominado GGVNS, que combina as metaheurísticas *General Variable Neighborhood Search* - GVNS (HANSEN; MLADENOVIC; PÉREZ, 2008) e o procedimento GRASP. Do procedimento GRASP utilizou-se a fase de construção para produzir soluções viáveis e de boa qualidade rapidamente. O GVNS foi escolhido devido a sua simplicidade, eficiência e capacidade natural de sua busca local para lidar com diferentes vizinhanças. Os autores compararam os resultados gerados pelo GGVNS com aqueles alcançados pelo otimizador CPLEX 11.01, utilizando oito problemas-teste. Os experimentos computacionais mostraram que o algoritmo GVNS era competitivo e capaz de encontrar soluções próximas do ótimo (com um *gap* < 1%) na maioria das instâncias, demandando um pequeno tempo computacional. Coelho et al. (2011b) apresentaram uma paralelização do algoritmo sequencial de Souza et al. (2010). Comparando a versão paralela e a versão sequencial, observou-se a supremacia da versão paralela, tanto em termos de qualidade da solução final quanto na variabilidade.

Coelho et al. (2011a) propõem um algoritmo evolutivo inspirado em Estratégias Evolutivas para resolver o POLAD mono-objetivo. O algoritmo desenvolvido utilizou o procedimento GRASP para gerar a população inicial entregue à Estratégia Evolutiva (ES) (BEYER; SCHWEFEL, 2002). Essa nova abordagem mostrou ser a mais eficiente até o momento, visto que os experimentos computacionais realizados mostraram a efetividade desse algoritmo quando comparado a outros da literatura.

Em termos de abordagem multiobjetivo para POLAD, o único trabalho encontrado na literatura foi o de Pantuza (2011). Este autor propôs um algoritmo genético multiobjetivo híbrido baseado no procedimento *Nondominated Sorting Genetic Algorithm II* (NSGA-II) (DEB et al., 2002). Na abordagem utilizada, foram considerados três objetivos conflitantes: minimizar o número de caminhões necessários para o processo de produção, minimizar os desvios em relação às metas dos teores dos parâmetros de qualidade e minimizar os desvios de produção de minério. Os resultados do modelo de otimização foram validados pela simulação.

3 *Heurísticas*

As heurísticas são técnicas que visam a obtenção de soluções de boa qualidade em um tempo computacional aceitável. Essas técnicas, no entanto, não garantem a obtenção da solução ótima para o problema, nem são capazes de garantir o quão próximo a solução obtida está da ótima.

As heurísticas podem ser construtivas ou de refinamento. As construtivas têm por objetivo construir uma solução, usualmente, elemento a elemento. A escolha de cada elemento está, geralmente, relacionada a uma determinada função que o avalia de acordo com sua contribuição para a solução. Tal função é bastante relativa, pois varia conforme o tipo de problema abordado.

As heurísticas de refinamento, também chamadas de mecanismos de busca local, são técnicas baseadas na noção de vizinhança. Para definirmos o que é uma vizinhança, seja S o espaço de busca de um problema de otimização e f a função objetivo a minimizar. O conjunto $N(s) \subseteq S$, o qual depende da estrutura do problema tratado, reúne um número determinado de soluções s' , denominado vizinhança de s . Cada solução $s' \in N(s)$ é chamada de vizinho de s e é obtida a partir de uma operação chamada de movimento.

Em linhas gerais, esses métodos partem de uma solução inicial s_0 , percorrem o espaço de busca por meio de movimentos, passando de uma solução para outra que seja sua vizinha.

3.1 **Metaheurísticas**

Metaheurísticas são procedimentos destinados a resolver aproximadamente um problema de otimização, tendo a capacidade de escapar das armadilhas dos ótimos locais, ainda distantes de um ótimo global. Elas podem ser de busca local ou populacional. Na primeira, a exploração do espaço de soluções é feita por meio de movimentos, os quais são aplicados a cada passo sobre a solução corrente, gerando outra solução promissora em sua vizinhança. Já na segunda, trabalha-se com um conjunto de soluções, re combinando-as com o intuito de aprimorá-las.

3.2 Otimização Mono-objetivo

3.2.1 Introdução

Os algoritmos de otimização são estratégias inteligentes para solucionar problemas de minimização (ou de maximização) de funções¹, em um determinado domínio, definido pelo conjunto de restrições nas variáveis de decisão. Segundo Goldberg (1989), o problema de otimização mono-objetivo pode ser matematicamente formulado como:

minimize:

$$f(x) \tag{3.1}$$

s.a:

$$g(x) \leq 0 \tag{3.2}$$

$$h(x) = 0 \tag{3.3}$$

$$x \in X \subset \mathbb{R}^N$$

sendo x o vetor de variáveis de decisão ou de otimização com N elementos, X um sub-espço de $\mathbb{R}^N$, $f(x) : \mathbb{R}^N \rightarrow \mathbb{R}$, $g(x) : \mathbb{R}^N \rightarrow \mathbb{R}^j$ e $h(x) : \mathbb{R}^N \rightarrow \mathbb{R}^k$ são, respectivamente, a função objetivo, o vetor de restrições de desigualdades e o vetor restrições de igualdade.

Para a solução dos problemas de otimização, dois grupos de métodos de otimização se destacam: *i*) métodos determinísticos e *ii*) métodos probabilísticos. No primeiro, os métodos são caracterizados por necessitarem de cálculos de derivadas das funções e fazem a busca da solução ótima gerando uma sequência de pontos segundo a expressão $x^{t+1} = x^t + \alpha d^t$, onde d^t é o vetor de busca, cuja expressão matemática contém informações de derivada das funções²

Os algoritmos determinísticos, por necessitarem de informações de derivadas da função objetivo (caso irrestrito), não garantem a convergência para a solução ótima global quando esta função é multimodal.

No segundo grupo, os métodos não necessitam do cálculo da derivada das funções; portanto a função não necessita ser contínua. Além disto, com estes métodos é possível encontrar a solução ótima global de funções multimodais, contudo, não é garantido que esta solução seja encontrada. A desvantagem dos métodos probabilísticos em relação aos determinísticos é que eles necessitam de maior número de cálculo de funções e são, portanto, mais custosos do ponto

¹As funções podem ser mono ou multivariáveis, lineares ou não-lineares, contínuas ou descontínuas

²No caso do método do gradiente, a direção de busca para o problema de minimização irrestrita: minimize $f(x)$ é $d^t = -\nabla f(x)|_{x=x^t}$

de vista computacional.

Os métodos da computação evolucionária são probabilísticos. Segundo Back, Hammel e Schwefel (1997), a computação evolucionária teve origem na década de 50, contudo, sem grande desenvolvimento nas três primeiras décadas, principalmente devido à falta de computadores eficientes na época. Na década de 70, trabalhos de Rechenberg, Schwefel e Fogel foram de grande importância para a mudança da imagem da computação evolucionária. Podem ser citados, em especial, a publicação do livro de John Holland - “Adaptation in Natural and Artificial Systems” em 1975. Nesse trabalho, Holland propõe um método de otimização baseado na seleção e genética natural, que, posteriormente, ficou conhecido como Algoritmos Genéticos (AGs). Os AGs têm sido utilizados na otimização de diversos problemas em várias áreas da ciência. Trabalhos importantes como otimização de dispositivos eletromagnéticos (VASCONCELOS, 1994), Otimização de Funções Matemáticas, Otimização Combinatória, Otimização de Planejamento, Problema do Caixeiro Viajante, Problema de Roteamento de Veículos, Otimização de *Layout* de Circuitos, Síntese de Circuitos Eletrônicos, citados em Michalewicz (1984), são exemplos de aplicações do uso de AGs.

A ideia dos algoritmos presentes na computação evolucionária é evoluir populações de indivíduos (soluções candidatas) na direção do ótimo. Segundo Back, Hammel e Schwefel (1997), dentre os algoritmos existentes na computação evolucionária, a principal diferença entre eles está na representação de seus indivíduos e nas operações realizadas para a geração de novos pontos no espaço de otimização. Esse tópico será melhor discutido na Seção 3.3.4.1.

Nos casos em que se utilizam métodos de otimização determinísticos, o resultado final do processo de otimização é uma única solução, a qual é aceita pelo tomador de decisões ou não. No caso de se utilizar algoritmos evolucionários, o resultado final é mais flexível, pois o tomador de decisões tem à sua disposição não só a melhor opção, mas também outras que podem ser tão interessantes quanto à melhor solução encontrada.

No contexto da otimização mono-objetivo, são apresentados a seguir os algoritmos referenciados neste trabalho.

3.2.2 Heurísticas Mono-objetivo

Dentre as várias heurísticas conhecidas na literatura, neste trabalho foram utilizadas três heurísticas mono-objetivo descritas nas seções a seguir.

3.2.2.1 Greedy Randomized Adaptive Search Procedure (GRASP)

GRASP (FEO; RESENDE, 1995; RESENDE; RIBEIRO, 2003, 2010) é um método iterativo constituído basicamente de duas fases: uma fase de construção e uma fase de busca local, cujo objetivo é convergir à solução encontrada na fase de construção para um ótimo local. A Algoritmo 1 apresenta o pseudocódigo básico do método GRASP para um problema de minimização.

Algoritmo 1: Greedy Randomized Adaptive Search Procedure

Entrada: Inteiro $GRASP_{max}$, Função $f(\cdot)$

Saída: Solução s^* melhor quanto à função f em $GRASP_{max}$ iterações

```

1  $f^* \leftarrow \infty$ 
2 para cada uma das  $GRASP_{max}$  iterações faça
3   Construa uma solução  $s_0$  por uma heurística parcialmente gulosa
4   Submeta  $s_0$  a um procedimento de busca local, retornando  $s$ 
5   se  $f(s) < f^*$  então
6      $s^* \leftarrow s$ 
7      $f^* \leftarrow f(s)$ 
8   fim
9 fim
10 retorna  $s^*$ 

```

A primeira fase do GRASP é a de construção, na qual uma solução viável é construída elemento a elemento. Cada elemento ainda não usado na solução é avaliado por uma função gulosa g e compõe uma lista, denominada de Lista de Candidatos (LC). Por meio de um fator $\alpha \in [0, 1]$ é criada uma Lista Restrita de Candidatos (LRC), cujos elementos i são os melhores da LC segundo a função g e satisfazem a condição $g_i \leq g_{\min} + \alpha \times (g_{\max} - g_{\min})$, sendo $g_{\min}$ o valor do elemento com a melhor avaliação segundo g e $g_{\max}$, o de pior avaliação. Definida a LRC, seleciona-se, aleatoriamente, um candidato da LRC e, em seguida, atualizam-se as listas LC e LRC. O método é interrompido quando $LC = \emptyset$.

De acordo com Feo e Resende (1995), o parâmetro α , que determina o tamanho da LRC, influencia significativamente a qualidade e diversidade das soluções geradas durante a fase de construção. Valores de α muito baixos (próximos de zero), ou seja, que determinam um tamanho muito limitado para a LRC, geram soluções próximas à solução puramente gulosa e implicam em uma baixa diversidade das soluções finais. Já uma escolha de α próxima da seleção puramente aleatória (valores de α próximos a 1) leva a uma grande diversidade de soluções construídas mas, por outro lado, muitas das soluções construídas são de qualidade inferior, tor-

nando mais lento o processo de busca local.

O pseudocódigo da fase de construção é apresentado na Figura 2:

Algoritmo 2: Construção GRASP

Entrada: Lista de elementos candidatos LC, Função $g(\cdot)$

Saída: Solução s construída de forma parcialmente gulosa quanto à função g

```

1  $s \leftarrow \emptyset$ 
2 enquanto a solução  $s$  não estiver totalmente construída faça
3   | Classifique os elementos de LC de acordo com a função  $g$ 
4   | Crie uma LRC, composta pelos melhores elementos da LC
5   | Selecione aleatoriamente um elemento de LRC e inclua-o na solução  $s$ 
6   | Atualize as listas LC e LRC, eliminando o elemento candidato inserido em  $s$ 
7 fim
```

A segunda fase do GRASP consiste em refinar a solução gerada pela fase de construção, aplicando um método de busca local. A velocidade de convergência para um ótimo local irá depender da qualidade da solução construída. Quanto melhor for a qualidade da solução gerada pela heurística de construção, maior será a velocidade de convergência desta solução para um ótimo local.

3.3 Otimização Multiobjetivo

3.3.1 Introdução

Um Problema de Otimização Multiobjetivo (MOOP, do inglês *Multi-Objective Optimization Problem*) possui um conjunto de funções objetivo a serem otimizadas (maximizadas ou minimizadas). Matematicamente, de acordo com Dias e Vasconcelos (2002), o MOOP pode ser formulado como:

minimize:

$$f(x) = \{f_1(x), f_2(x), \dots, f_M(x)\} \quad (3.4)$$

s.a:

$$g(x) = \{g_1(x), g_2(x), \dots, g_J(x)\} \leq 0 \quad (3.5)$$

$$h(x) = \{h_1(x), h_2(x), \dots, h_K(x)\} = 0 \quad (3.6)$$

$$x = \{x_1, x_2, \dots, x_N\} \in X \subset \mathbb{R}^N$$

$$y = \{y_1, y_2, \dots, y_M\} \in Y$$

em que X é o espaço de decisão e Y o espaço dos objetivos.

Na solução de problemas multiobjetivo é gerado ao final um conjunto de soluções, conhecidas como soluções não-dominadas ou de soluções eficientes. Para compreender o que são estas soluções não-dominadas, é necessário apresentar algumas definições.

Definição 1 : Um vetor x^1 domina um vetor x^2 (matematicamente se escreve $x^1 \prec x^2$), quando a avaliação do primeiro não é pior do que a avaliação do segundo em nenhum dos objetivos e é melhor em pelo menos um. Matematicamente, pode-se dizer que $x^1 \prec x^2$ quando se verifica a seguinte relação matemática:

$$\text{Se } \forall i \in \{1, \dots, M\}, y(x^1) \leq y(x^2) \wedge \exists i \in \{1, \dots, M\} \mid y(x^1) < y(x^2)$$

Definição 2 :

Se um vetor x^1 não é dominado por nenhum vetor x^2 qualquer, em todo o espaço viável, diz-se que x^1 é uma solução eficiente, não-dominada, ou solução Pareto-ótima.

Assim, utilizando estas definições, quando um conjunto de soluções finito P é encontrado, se torna possível realizar comparações das soluções duas a duas, dividindo estas soluções em um grupo chamado de soluções dominadas e de soluções não-dominadas P' . As soluções de P' são não-dominadas por qualquer outra solução presente em P . Se o conjunto não-dominado P' abrange a totalidade do espaço de busca factível, ele é chamado de conjunto Pareto-ótimo global.

A Figura 1 ilustra os espaços das variáveis de decisão e dos objetivos. É também mostrado nesta figura a fronteira Pareto-ótima global. Nesta figura, há dois conjuntos Pareto-ótimos que são não-dominados localmente, mostrando a sua vizinhança no seu espaço de dois objetivos e no espaço de variáveis (à direita).

A fronteira Pareto ótimo, ilustrada na Figura 1, é formada por valores das funções objetivo $f(x) = (f_1(x), \dots, f_m(x))$ correspondentes a cada solução no espaço de busca. Logo, para cada uma das soluções encontradas no espaço de variáveis, estas soluções são representadas no espaço dos objetivos, avaliando cada uma delas em cada um dos objetivos existentes.

Um dos objetivos principais de algoritmos que solucionam problemas com múltiplos objetivos é encontrar soluções o mais próximo possível da fronteira de Pareto, e, ainda no universo de soluções encontradas, buscarem uma maior diversidade possível.

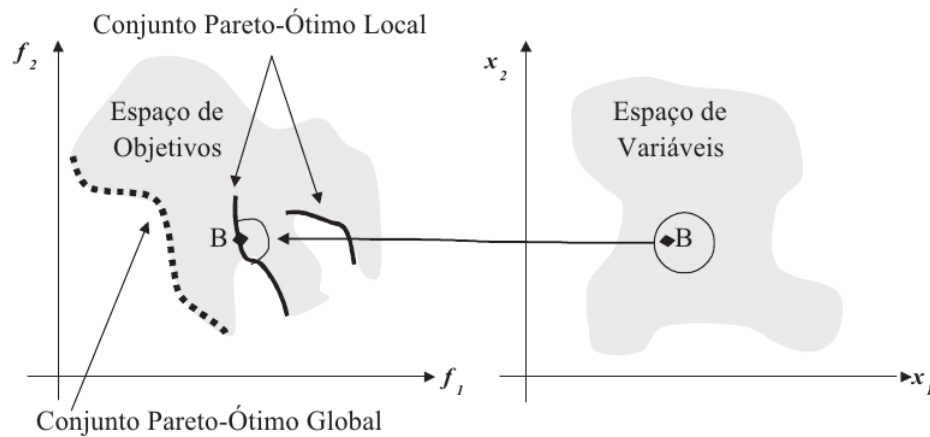


Figura 1: Soluções Pareto ótimo locais e globais (ALEXANDRE, 2010)

3.3.2 Metas da Otimização Multiobjetivo

Se não existe nenhuma informação adicional sobre a importância de cada um dos objetivos, todas as soluções Pareto-ótimas são igualmente importantes. Na prática (em empresas, indústrias e em vários outros setores), o tomador de decisão (*decision maker*) define qual a melhor solução a ser utilizada no momento. Deb (2001) assinala três importantes metas em otimização multiobjetivo:

1. Encontrar um conjunto de soluções que esteja o mais próximo possível da Fronteira de Pareto;
2. Encontrar um conjunto de soluções com a maior diversidade possível;
3. Realizar as duas metas anteriores com a maior eficiência computacional possível.

3.3.3 Diferenças com a otimização Mono-objetivo

Deb (2001) identifica três importantes aspectos que diferenciam a otimização multiobjetivo da otimização mono-objetivo:

1. Em problemas de otimização com um único objetivo, a meta é encontrar uma solução ótima global. Se a função objetivo desses problemas for multimodal, poderia existir mais ótimo global. Neste caso, todos os ótimos são equivalentes. Por outro lado, em MOOP, determinar o conjunto de soluções da fronteira de Pareto é tão importante quanto preservar a diversidade neste conjunto. Um algoritmo eficiente para otimização multiobjetivo deve considerar ambos os aspectos;

2. Um MOOP trabalha com dois espaços (das variáveis e dos objetivos) ao invés de um. Problemas de objetivo simples trabalham unicamente no espaço de variáveis pois procuram apenas uma solução no espaço de objetivos. Manter a diversidade em ambos espaços complica mais o problema, dado que a proximidade de duas soluções no espaço de variáveis *não implica* proximidade no espaço de objetivos.
3. Os métodos tradicionais de otimização multiobjetivo reduzem o conjunto de funções objetivo a uma função simples, a qual pondera cada objetivo. Estes métodos podem também tratar cada objetivo separadamente, utilizando os demais objetivos como restrições. Portanto, um MOOP pode ser convertido, por meio de algumas técnicas, em um problema de otimização simples.

3.3.4 Heurísticas de Otimização Multiobjetivo

3.3.4.1 Computação Evolucionária

Como relatado em (ALEXANDRE, 2010), a computação evolucionária é uma área de pesquisa que busca encontrar soluções eficientes para problemas de grande complexidade. As características principais dos algoritmos evolucionários são:

- São baseados na teoria da evolução de Darwin;
- Trabalham com populações de possíveis soluções;
- São aplicados em diversas áreas tais como otimização mono e multiobjetivo, classificação de padrões, diagnóstico de falhas incipientes, entre outras;
- São fáceis de serem adaptados a diferentes problemas da engenharia e não dependem de características específicas das funções envolvidas no modelo matemático dos problemas;
- São capazes de encontrar boas soluções para problemas com elevado grau de complexidade;
- São simples e fáceis de serem implementados.
- Não garante que seja encontrada a solução ótima para os problemas.
- Utiliza o tempo computacional para avaliação de cada solução gerada.

Os Algoritmos Genéticos (AGs), introduzidos por John Holland, na década de 70, fazem parte da área de Computação Evolutiva, que constitui uma família de métodos computacionais inspirados na evolução natural das espécies. Goldberg (1989) afirma em seu trabalho que o uso de AGs para a solução de problemas multiobjetivo teve início quando Schaffer (SCHAFER, 1985) implementou a primeira versão de um AG multiobjetivo denominado VEGA (*Vector Evaluated Genetic Algorithm*). Esse algoritmo considera uma população de N indivíduos e M objetivos, dividida em m subpopulações com N/m indivíduos em cada uma delas. O operador de seleção dos AGs é aplicado separadamente para cada uma das subpopulações, isto é, para a subpopulação m considera-se apenas o m -ésimo objetivo para fins da seleção, e, posteriormente, une-se estas subpopulações e aplicam-se os outros operadores genéticos de cruzamento e mutação. Além disso, Goldberg (1989) propôs várias abordagens para estender as aplicações de AGs para problemas multiobjetivos. Uma delas propõe um procedimento para ordenação de soluções baseado no conceito de dominância de Pareto. Neste caso, o valor da aptidão de uma solução é proporcional ao número de soluções que ela domina.

O trabalho de Coello (2006) apresenta uma visão geral da história da otimização multiobjetivo. Ele divide os algoritmos até então existentes em duas gerações. A primeira delas envolve algoritmos que possuem como característica a ênfase maior na simplicidade. Entre esses algoritmos destacam-se o VEGA, já discutido anteriormente, o *Nondominated Sorting Genetic Algorithm* (NSGA) (SRINIVAS; DEB, 1994), o *Niched-Pareto Genetic Algorithm* (NPGA) (HORN; NAFLIOTIS; GOLDBERG, 1994) e o *Multi-Objective Genetic Algorithm* (MOGA) (FONSECA; FLEMING, 1993). A segunda geração dos algoritmos dá maior ênfase à eficiência. Entre os algoritmos classificados nessa segunda geração estão: *Strength Pareto Evolutionary Algorithm* (SPEA) e *Strength Pareto Evolutionary Algorithm II* (SPEA2) (ZITZLER; LAUMANN; THIELE, 2001), *Pareto Archived Evolution Strategy* (PAES) (KNOWLES; CORNE, 1999) e o *Nondominated Sorting Genetic Algorithm II* (NSGA-II) (DEB et al., 2002).

Dentre os diversos métodos utilizados na literatura para se encontrar soluções não-dominadas, neste trabalho o foco será dado ao algoritmo *Two-phase Pareto local search* (2PPLs).

3.3.4.2 Two-phase Pareto local search (2PPLs)

O algoritmo 2PPLs é composto de duas fases:

1. Na primeira fase, uma população inicial diversificada e com boa aproximação dos extremos de um conjunto eficiente é gerada;
2. Na segunda fase, aplica-se o método Pareto Local Search (PLS) a cada um dos indivíduos

da população

ou seja, **Pareto Local Search é visto como uma generalização multiobjetivo do método hill-climbing!!**

A combinação do PLS com uma boa população inicial fornece melhores resultados do que o método PLS básicos

Algoritmo 3: 2PPLs com VNS

Entrada: Aproximação inicial de um conjunto eficiente P_0 ; Vizinhanças $\mathcal{N}_k(x)$;

Saída: Conjunto Eficiente Xe

```

1  $Xe \leftarrow P_0; P \leftarrow P_0; P_a \leftarrow \emptyset$ 
2  $k \leftarrow 1$  {Tipo de estrutura de vizinhança corrente}
3 enquanto  $k \leq r$  faça
4   para todo  $p \in P$  faça
5     para todo  $p' \in N_k(p)$  faça
6       se  $f(p) \not\preceq f(p')$  então
7          $\text{addSolution}(Xe, p', f(p'), Added)$ 
8         se  $Added = \text{verdadeiro}$  então
9            $\text{addSolution}(P_a, p', f(p'))$ 
10        fim
11      fim
12    fim
13  fim
14  se  $P_a \neq \emptyset$  então
15     $P \leftarrow P_a; P_a \leftarrow \emptyset; k \leftarrow 1$ 
16  senão
17     $k \leftarrow k + 1$ 
18     $P \leftarrow Xe \setminus \{x \in Xe \mid \text{Pareto Local ótimo em relação } \mathcal{N}_k(x)\}$ 
19  fim
20 fim
21 retorna  $Xe$ 

```

O procedimento `addSolution` (linha 5 do Algoritmo 8), encontra-se detalhado no Algoritmo 4, que adiciona as soluções não-dominadas ao conjunto Xe de soluções potencialmente eficientes.

Na linha 4 aplica-se o método Pareto Local Search (PLS) a cada um dos indivíduos da população. O método PLS é visto como uma generalização multiobjetivo do método da descida (LUST; TEGHEM, 2010). Quando pelo menos uma solução não-dominada é adicionada ao conjunto Xe , linha 14 do Algoritmo 3, a vizinhança corrente retorna à primeira vizinhança do conjunto de vizinhanças $\mathcal{N}_k(x)$, passado como parâmetro do algoritmo. Caso não haja nenhum vizinho a ser adicionado ao conjunto de soluções não-dominadas Xe , a próxima estrutura de vizinhança é acionada (linha 17). O algoritmo termina sua execução quando todas as vizinhanças forem percorridas. A linha 18 assegura que algoritmo não percorra soluções já visitadas.

Algoritmo 4: addSolution

Entrada: População Xe potencialmente eficiente; Solução s , e sua avaliação $z(s)$

Saída: Xe ; Added (opcional)

```

1 Added ← verdadeiro
2 para todo  $x \in Xe$  faça
3   se  $z(x) \preceq z(s)$  então
4     Added ← falso
5     Break
6   fim
7   se  $z(s) \prec z(x)$  então
8      $Xe \leftarrow Xe \setminus x$ 
9   fim
10 fim
11 se Added = verdadeiro então
12    $Xe \leftarrow Xe \cup s$ 
13 fim
14 retorna  $Xe$ 

```

3.3.5 Métricas de Avaliação de Desempenho

Métricas de avaliação de desempenho são bastante utilizadas a fim de mensurar características de algoritmos, ajudando a entender seu comportamento no domínio do problema e permitindo uma avaliação mais concreta do desempenho do algoritmo. Porém, comparar experimentalmente o desempenho de um ou vários algoritmo multiobjetivos não é uma tarefa trivial (DEB, 2001; ZITZLER; DEB; THIELE, 2000). As métricas também são um importante parâmetro de comparação entre algoritmos, uma vez que muitas vezes é difícil perceber qual algoritmo

apresenta um melhor conjunto de soluções para o problema. As duas principais metas da otimização multiobjetivo são a convergência e a diversidade das soluções encontradas. A Figura 2 ilustra ambas as metas. É importante observar que a diversidade e a convergência são conflitantes entre si; logo, utilizar apenas uma métrica não avaliará por completo a Frente de Pareto analisada.

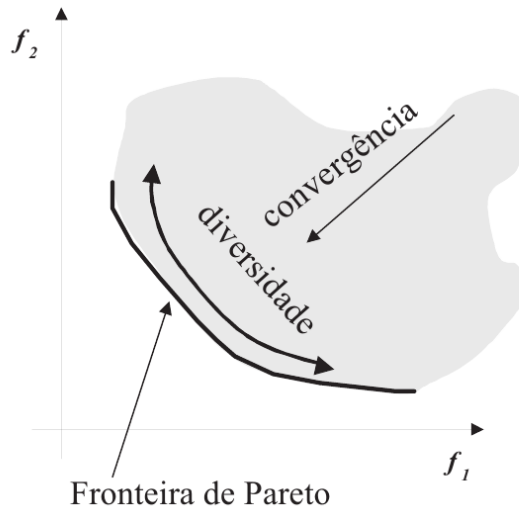


Figura 2: Metas da Otimização Multiobjetivo (DEB, 2001)

A Figura 3 mostra as soluções não-dominadas obtidas por dois algoritmos hipotéticos A e B, neste caso, o algoritmo A possui uma convergência, enquanto que o algoritmo B obteve uma Fronteira de Pareto bem diversificada. Já na Figura 4, é notório que a tarefa de comparar as Fronteiras de Pareto não é trivial, sendo difícil determinar qual algoritmo obteve o melhor desempenho.

Neste trabalho foram utilizadas três métricas de comparação: Hipervolume, Espaçamento e Cobertura. Uma breve revisão sobre essas três métricas é apresentada a seguir.

3.3.5.1 Hipervolume

O indicador de hipervolume (ZITZLER; THIELE, 1998) mensura o volume da região coberta entre os pontos das soluções do *front* encontrado e um ponto de referência. Matematicamente, para cada solução i pertencente à Fronteira de Pareto Q , é construído um hipercubo v_i de acordo com uma ponto de referência W_0 . Uma maneira fácil de determinar este ponto é contruir um vetor com os piores valores de função objetivo. O resultado da métrica é a união de todos os hipercubos encontrados. Nessa métrica, um alto valor de hipervolume indica que houve um elevado espalhamento entre as soluções extremas do Pareto encontrado e indica, também, que houve uma maior convergência, pois a convergência aumenta o volume em relação ao ponto de

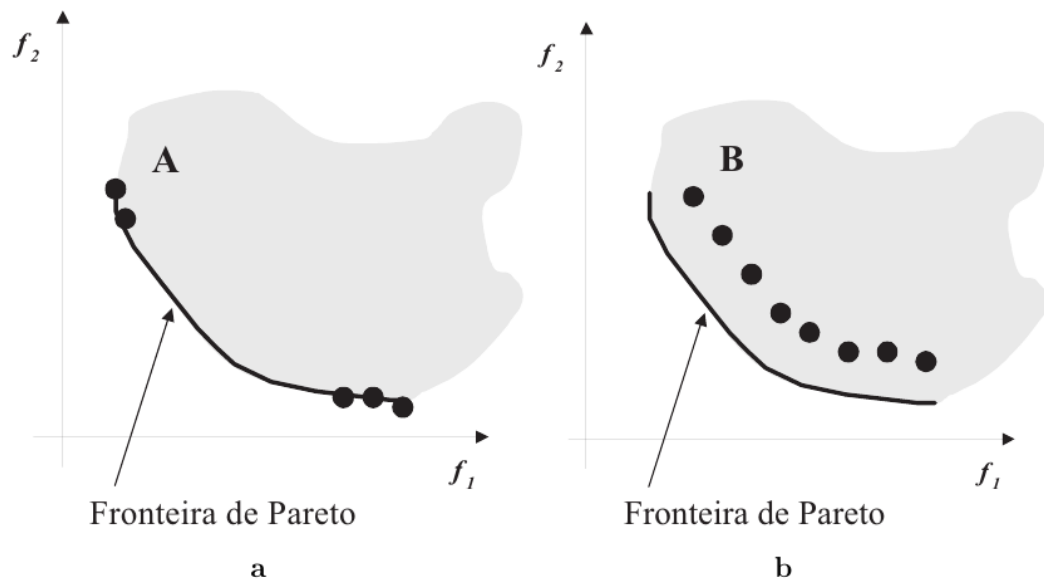


Figura 3: Distribuição $\times$ Convergência - 1 (DEB, 2001)

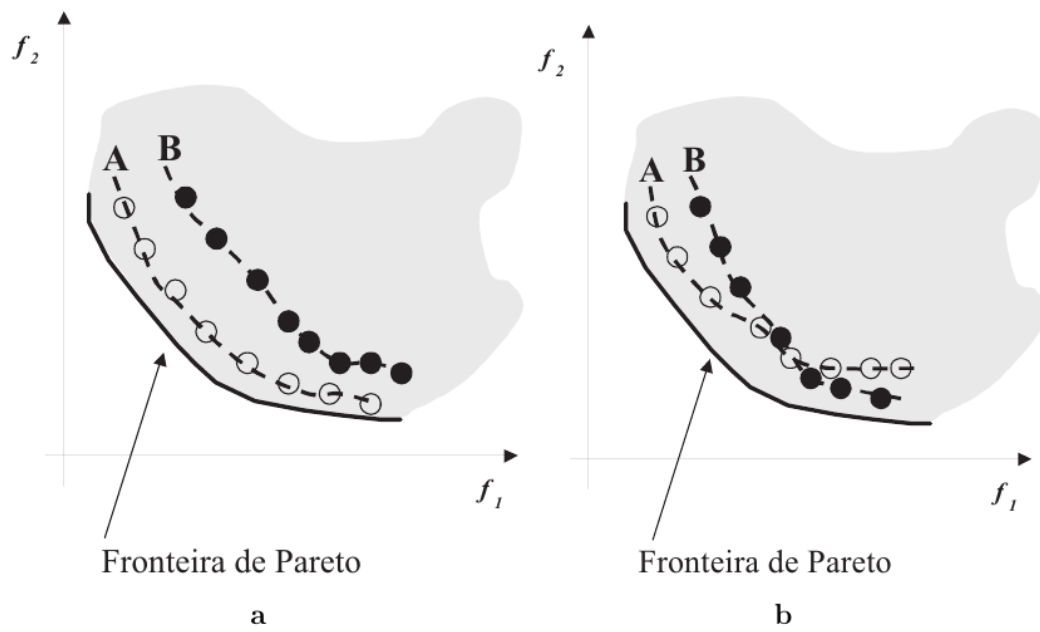


Figura 4: Distribuição $\times$ Convergência - 2 (DEB, 2001)

referência. A figura Essa métrica apresenta um elevado custo computacional e quando se tem um número de objetivos maior que dois, o seu cálculo passa a não ser trivial. Fonseca, Paquete e Lopez-Ibanez (2006), Beume et al. (2009) propuseram uma ferramenta computacional eficiente para o cálculo do hipervolume, a qual foi utilizada neste trabalho.

A Figura 5 ilustra o hipervolume para um Fronteira de Pareto com dois objetivos.

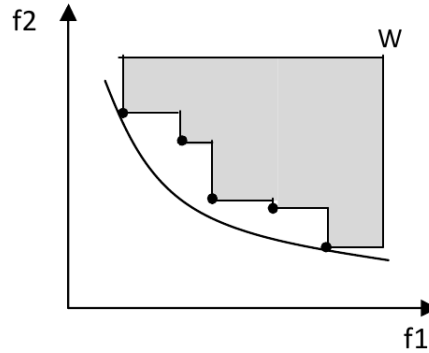


Figura 5: Hipervolume gerado pelas soluções não-dominadas de um Fronteira de Pareto hipotética

3.3.5.2 Espaçamento

Schott (1995) propôs uma métrica de espaçamento que mensura as distribuições das soluções na Fronteira de Pareto. Essa métrica calcula a distância relativa entre soluções consecutivas na Fronteira. A Eq. (3.7) descreve o cálculo dessa métrica.

$$SS = \sqrt{\frac{1}{|Q|} \sum_{i=1}^{|Q|} (d_i - \bar{d})^2} \quad (3.7)$$

Na Eq. (3.7), $|Q|$ é número de soluções da Frente de Pareto, $d_i = \min_{j \in Q | i \neq j} \sum_{m=1}^M |f_m^i - f_m^j|$ e $\bar{d}$ é a média de todos os d_i , ou seja, $\bar{d} = \sum_{i=1}^{|Q|} \frac{d_i}{|Q|}$. O parâmetro M indica o número de objetivos do problema e, por fim, f_m^i e f_m^j indicam os valores do objetivo m para as soluções i e j , respectivamente. Como toda medida de variância, quanto menor for o valor do espaçamento, melhor será a distribuição das distâncias d_i . Consequentemente, as soluções da Frente de Pareto estarão separadas mais uniformemente. Um valor da métrica igual a zero significa que todas as soluções estão equidistantes na Frente de Pareto analisada.

3.3.5.3 Cobertura

A métrica de cobertura (ZITZLER; THIELE, 1998) é capaz de mensurar quanto um determinado conjunto de soluções domina outro conjunto. Para duas Frentes de Pareto A e B, a cobertura $C(A, B)$ é calculada de acordo com a equação 3.8 que indica a porcentagem de indivíduos em B que são fracamente dominados por indivíduos de A.

$$C(A, B) = \frac{|b \in B; \exists a \in A : a \succeq b|}{|B|} \quad (3.8)$$

Como se pode observar, o valor de $C(A, B)$ está dentro do intervalo $[0, 1]$. O valor $C(A, B) = 1$ indica que todas as soluções em B são fracamente dominadas por A. Por outro lado, $C(A, B) = 0$ indica que nenhuma das soluções em B são fracamente dominadas por A. É fácil observar que se $C(A, B) = X$ e $C(B, A) = Y \nRightarrow X + Y = 1$, ou seja, $C(A, B)$ não é, necessariamente, igual a $1 - C(B, A)$.

3.3.5.4 Cardinalidade

Uma medida de cardinalidade é a porcentagem de pontos do Conjunto de Referência *Ref* encontradas no Conjunto de Soluções Aproximadas A. A dificuldade é a obtenção do Conjunto de Referência *Ref*, pois na maioria dos problemas reais, pode ser impraticável o mapeamento de todas as soluções deste conjunto. Outra medida de cardinalidade melhorada é a porcentagem de pontos heurísticos não dominados por pontos de referência. Entretanto, esta medida não considera a distribuição dos pontos heurísticos e a distância em relação aos pontos de referência.

4 *O Planejamento Operacional de Lavra Abordado*

4.1 Introdução

O Problema de Planejamento de Lavra a Céu Aberto em Mineração envolve a alocação de máquinas e caminhões às frentes de lavra. As Figuras 6 e 7 e ilustram, respectivamente, um equipamento de carga abastecendo um caminhão em uma frente e um caminhão depositando o minério (ou estéril) no britador.



Figura 6: Equipamentos de carga e transporte

Cada frente de lavra contém uma determinada quantidade de material (minério ou estéril), com características físicas, químicas e econômicas diferenciadas, denominadas parâmetros de controle. Como exemplo típico de parâmetros de controle, tem-se: Fe, SiO₂, H₂O, Mn, P, granulometria. Para satisfazer as especificações exigidas pelos clientes, é necessário selecionar as frentes a serem lavradas e seu ritmo de lavra, os quais devem ser determinados proporcional-



Figura 7: Britador

mente. Para a operação de minério e estéril, a mina conta com uma frota limitada de equipamentos de carga, os quais devem ser alocados às frentes de lavra e operarem em uma faixa de produtividade que torne viável sua utilização (COSTA, 2005).

Considera-se que o transporte do material retirado da frente de lavra é realizado por uma frota de caminhões com capacidades de carga diferentes, a Figura 8 ilustra um modelo de caminhão para transporte de estéril e minério. Esses caminhões são alocados às frentes de lavra dinamicamente, tentando-se evitar a formação de filas, ou seja, o caminhão é alocado a um ponto de carga ou basculamento que proporcione o menor tempo de fila possível.

O ritmo de lavra é determinado pelas capacidades de operação dos equipamentos de carga e transporte alocados às diversas frentes. A Figura 9 mostra um tipo de equipamento de carga utilizado para o carregamento de minério e estéril na frente de lavra.

Em minas a céu aberto, são utilizados dois critérios para a alocação de caminhões: alocação estática e alocação dinâmica. No sistema de alocação dinâmica, os caminhões não ficam fixos a uma determinada rota, como no sistema de alocação estática. Eles podem ser direcionados a diferentes frentes de lavra, onde esteja um equipamento de carga compatível. Esta estratégia faz aumentar a produtividade da frota e proporciona, segundo Costa (2005), um aumento na capacidade de produção da mina ou mesmo a redução do número de equipamentos necessários para manter o mesmo nível de produção.



Figura 8: Modelo de Caminhão (ARAÚJO, 2008)



Figura 9: Modelo de Carregadeira - L1850 (ARAÚJO, 2008)

4.2 Características do Problema de Alocação Abordado

O problema abordado neste trabalho é o de Planejamento Operacional de Lavra com alocação dinâmica de caminhões (POLAD), sendo estes de capacidades diferentes.

Sendo a alocação dinâmica, ao descarregar o material, seja no britador (ou pilhas de estoque próximas ao britador) ou na pilha de estéril, o caminhão é direcionado a uma frente, não necessariamente a mesma da viagem anterior.

Admite-se que há um conjunto de carregadeiras de diferentes produtividades, sendo este conjunto menor que o de frentes às quais elas serão alocadas.

Considera-se o planejamento para uma hora de produção, sendo este aplicado até uma frente exaurir ou ocorrer uma quebra de equipamento, situação na qual deve ser feito outro planejamento.

Dado o elevado custo de uma carregadeira, é imposto um limite mínimo de produção para cada carregadeira para justificar economicamente sua utilização.

Finalmente, considera-se uma taxa de utilização máxima para os caminhões. Por exemplo, supondo uma taxa de utilização máxima de 85%, um caminhão l de 80 t de capacidade, deveria trabalhar 51 ($= 0,85 \times 60$) minutos, no máximo, em uma hora. Isso é adotado para retratar uma situação mais real, uma vez que um caminhão não fica todo o tempo em atividade. Além disso, essa taxa de utilização máxima tem por objetivo, também, modelar a variabilidade nos tempos de ciclo dos caminhões.

Por fim, a Figura 10 ilustra, de uma forma geral, o problema de planejamento operacional de lavra com alocação dinâmica de caminhões. Nota-se que não temos nenhuma carregadeira alocada na frente com teor de 50% Fe, situação que ocorre na prática, pois nem sempre o número de carregadeiras é suficiente para alocar um equipamento em cada frente de lavra. A Figura 10 mostra, também, que a frota de caminhões e carregadeiras é heterogênea, ou seja, é comum encontrar equipamentos de transporte e carga com diferentes capacidade e compatibilidade, ou seja, nem todo caminhão é compatível com o equipamento de carga alocado na frente de lavra.

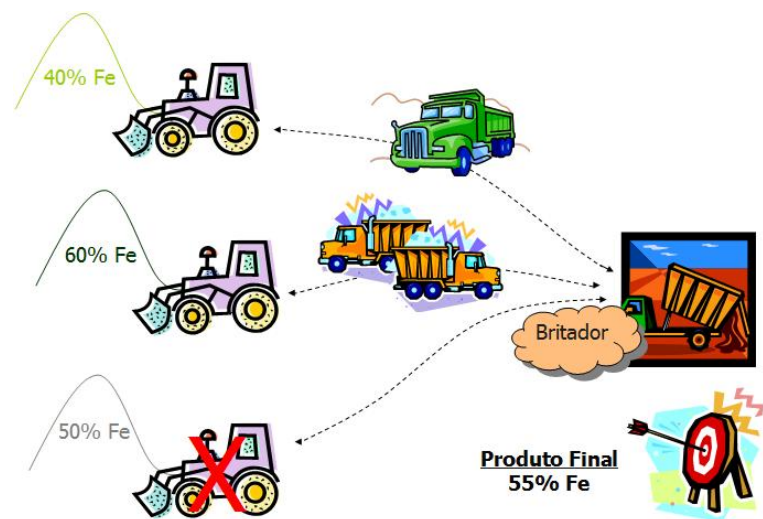


Figura 10: Exemplo de operação em uma mina a céu aberto

5 Metodologia Heurística

Neste capítulo são apresentados os métodos heurísticos propostos para resolver o POLAD multiobjetivo. A Seção 5.1 descreve como uma solução do POLAD é representada nos algoritmos desenvolvidos neste trabalho. A heurística utilizada para geração da solução inicial é apresentada na Seção 5.2. Na Seção 5.3 são apresentados os movimentos que constituem as estruturas de vizinhança utilizadas para resolução do problema. A Seção 5.4 mostra como uma solução é avaliada. A Seção 5.5 detalha o algoritmo desenvolvido para resolver o MOPOLAD, denominado GRASP-2PPLS, combina os procedimentos *Greedy Randomized Adaptive Search Procedures* (GRASP) e *Two-phase Pareto local search* (MOVNS).

5.1 Representação de uma solução

Uma solução do POLAD é representada por uma matriz $R_{|F| \times (1+|V|)}$ de valores inteiros, sendo F o conjunto de frentes e V o conjunto de caminhões.

Para clareza de apresentação, a matriz $R_{|F| \times (1+|V|)}$ é decomposta em duas submatrizes Y e N , com $R = [Y|N]$, sendo $Y = (y_i)_{|F| \times 1}$ e $N = (n_{il})_{|F| \times |V|}$. A submatriz $Y_{|F| \times 1}$ representa a alocação dos equipamentos de carga ao conjunto F de frentes e o respectivo *status* de cada um desses equipamentos com relação ao fato de estarem ativos ou não. Em cada célula y_i da matriz $Y_{|F| \times 1}$ representa-se a carregadeira k alocada à frente i . Um valor D significa que não existe carregadeira alocada. Se não houver viagens feitas a uma frente i , a carregadeira k associada a tal frente é considerada *inativa* e não é penalizada por produção abaixo da mínima para este equipamento de carga, restrições da formulação de programação modelo matemática de Souza et al. (2010). A submatriz $N = (n_{il})_{|F| \times |V|}$ representa o número de viagens realizadas pelos caminhões l às frentes i . Um valor 0 (zero) significa que não há viagem para aquele caminhão, enquanto um valor X informa que há incompatibilidade entre o caminhão e a carregadeira alocada àquela frente.

A Tabela 1 exemplifica uma solução para uma instância do problema. Nesta tabela, as

linhas representam as frentes de lavra disponíveis no conjunto F , a coluna $CARGA$ representa a alocação dos equipamentos de carga às frentes de lavra e as demais colunas indicam o número de viagens que serão realizadas pelo conjunto V de caminhões disponíveis.

Tabela 1: Exemplo de características de uma solução para o POLAD

	<i>Carga</i>	<i>Cam</i> ₁	<i>Cam</i> ₂	...	<i>Cam</i> _v
F_1	$\langle Car_1, 1 \rangle$	8	X	...	X
F_2	$\langle D, 0 \rangle$	0	0	...	0
F_3	$\langle Car_8, 0 \rangle$	0	0	...	0
...	...	...	...	...	...
F_F	$\langle Car_5, 1 \rangle$	0	9	...	3

Neste exemplo observa-se, na coluna $CARGA$, linha F_1 , a dupla $\langle Car_1, 1 \rangle$, indicando que o equipamento de carga Car_1 está alocado à frente F_1 e em operação. Na coluna $CARGA$, linha F_3 , a dupla $\langle Car_8, 0 \rangle$ indica que o equipamento de carga Car_8 está alocado à frente F_3 , mas não está em operação. Observa-se, ainda, na coluna $CARGA$, linha F_2 , o valor $\langle D, 0 \rangle$ informando que não existe equipamento de carga alocado à frente F_2 e que, portanto, esta frente está disponível. As demais colunas representam o número de viagens a serem realizadas por um caminhão a uma frente, considerando a compatibilidade entre o caminhão e o equipamento de carga alocado à frente. As células com os valores X indicam incompatibilidade entre um caminhão e o respectivo equipamento de carga.

A partir de Y , N e dos tempos de ciclo dados na matriz $TC = (tc_{il})_{|F| \times |V|}$ são determinados o ritmo de lavra em cada frente e o somatório dos tempos de ciclo de cada caminhão.

5.2 Geração de uma solução inicial

Uma solução inicial para o problema é feita em duas etapas. Na primeira, realizamos a alocação das carregadeiras e a distribuição das viagens às frentes estéril, e na segunda, às frentes de minério. Esta estratégia é adotada tendo em vista que nas frentes de estéril o importante é atender à produção e não é necessário observar a qualidade.

Na primeira etapa utilizamos uma heurística gulosa, cujo pseudocódigo está descrito no Algoritmo 5.

Algoritmo 5: Constrói Solução Estéril

Entrada: T, S, W

Saída: Solução de estéril S_w

$T \leftarrow$ Conjunto de caminhões ordenados por suas capacidades (o primeiro é o de maior capacidade).

$S \leftarrow$ Conjunto de carregadeiras ordenadas pelas produtividades (a primeira é a de maior produtividade).

$W \leftarrow$ Conjunto de frentes de estéril ordenadas pelas massas (a primeira é a de maior massa).

enquanto a produção de estéril for menor que a produção recomendada **e** existirem frentes de estéril não utilizadas **faça**

 Selecione a primeira frente de estéril i do vetor W ;

se não há carregadeira alocada à frente i **então**

se Todas as carregadeiras estão alocadas **então** Remova a frente i de W

senão

 Atualize S_w alocando a maior carregadeira disponível à frente i ;

fim

fim

se A frente i não foi removida de W **então**

 Encontre um caminhão $l \in T$ tal que: a) Seja compatível com a carregadeira alocada à frente i ; b) Seja possível realizar mais uma viagem; c) Sua capacidade não viole a produção máxima da carregadeira;

se O caminhão l existe **então** Atualize S_w , alocando a maior quantidade possível de viagens do caminhão l à frente de estéril i ;

senão

 Remova a frente i de W ;

fim

fim

fim

Retorne S_w ;

Na segunda etapa, utilizamos uma heurística que aplica *GRASP*_{max} vezes a fase de construção do procedimento *GRASP* e retorna a melhor das soluções construídas, desta feita incluindo-se as frentes de minério. A justificativa para esse procedimento é que a busca local de nosso algoritmo é muito custosa computacionalmente. Assim, a mesma requer uma boa solução inicial, o que, de acordo com (LOURENÇO; MARTIN; STÜTZLE, 2003), aceleraria a convergência para um ótimo local.

Para cada construção, utilizamos uma função guia g que relaciona os valores de desvio de qualidade em relação à meta. De acordo com esta função, é mais indicado selecionar uma frente

que minimize os desvio de qualidade dos parâmetros de controle.

Inicialmente, todas as frentes i candidatas são ordenadas de acordo com os valores de g_i e inseridas em uma lista de candidatos LC . De LC é extraída uma lista restrita de candidatos LRC contendo as frentes de minério mais bem qualificadas de acordo com a função guia. A cardinalidade desta lista, isto é, $\lceil \gamma \times |LC| \rceil$ é definida pelo parâmetro $\gamma \in [0, 1]$. A estratégia utilizada para escolher uma frente i consiste em atribuir, primeiramente, uma classificação probabilística para cada frente candidata da LRC . A função $bias(r) = 1/(r)$ é associada à frente que está na r -ésima posição na classificação. Cada frente candidata é, então, escolhida com probabilidade $p(r) = bias(r) / \sum_{i=1, \dots, |LRC|} bias(i)$. Em seguida, o algoritmo escolhe aleatoriamente uma frente de minério i de LRC , adicionando-a à solução parcial. O Algoritmo 6 descreve este procedimento de construção.

Algoritmo 6: Constrói Solução Minério

Entrada: S_w, γ, g, T, S

Saída: Solução S_0

$S_0 \leftarrow S_w$

$T \leftarrow$ Conjunto de caminhões ordenados pelas suas capacidades (o primeiro é o de menor capacidade).

$S \leftarrow$ Conjunto de carregadeiras ordenadas por suas produtividades (a primeira é a que de maior produtividade).

enquanto a produção de minério for menor que a produção recomendada **e** existirem frentes de minério não utilizadas **faça**

$LC \leftarrow$ Conjunto de frentes de minério ordenadas de acordo com a função g ;

$|LRC| = \lceil \gamma \times |LC| \rceil$;

Selecione uma frente $i \in LRC$ de acordo com a função *bias*;

se não há carregadeira alocada à frente i **então**

se Todas as carregadeiras estão alocadas **então** Remova a frente i de LC

senão

| Atualize S_0 alocando a carregadeira de maior capacidade à frente i ;

fim

fim

se A frente i não foi removida de LC **então**

Encontre um caminhão $l \in T$ tal que: a) Seja compatível com a carregadeira alocada à frente i ; b) Seja possível realizar mais uma viagem; c) Sua capacidade não viole a produção máxima da carregadeira;

se O caminhão l existe **então** Atualize S_0 , alocando a maior quantidade possível de viagens do caminhão l à frente de estéril i ;

senão

| Remova a frente i de W ;

fim

fim

fim

Retorne S_0 ;

5.3 Estruturas de vizinhança

Para explorar o espaço de soluções do POLAD foram desenvolvidos 8 tipos diferentes de movimentos, apresentados a seguir, para definir oito estruturas de vizinhança $N^k(s)$. Os seis primeiros movimentos, e suas devidas estruturas de vizinhança, foram propostos em Costa (2005). Souza et al. (2010) propôs as demais vizinhanças e relatou a boa capacidade exploratória desses

movimentos.

Uma breve descrição dos movimentos segue abaixo:

Movimento Número de Viagens - $N^{NV}(s)$: Este movimento consiste em aumentar ou diminuir o número de viagens de um caminhão l em uma frente i onde esteja operando um equipamento de carga compatível. Desta maneira, neste movimento uma célula n_{il} da matriz N tem seu valor acrescido ou decrescido de uma unidade.

Movimento Carga - $N^{CG}(s)$: Consiste em trocar duas células distintas y_i e y_k da matriz Y , ou seja, trocar os equipamentos de carga que operam nas frentes i e k , caso as duas frentes possuam equipamentos de carga alocados. Havendo apenas uma frente com equipamento de carga, esse movimento consistirá em realocar o equipamento de carga à frente disponível. Para manter a compatibilidade entre carregadeiras e caminhões, as viagens feitas às frentes são realocadas junto com as frentes escolhidas.

Movimento Realocar Viagem de um Caminhão - $N^{VC}(s)$: Consiste em selecionar duas células n_{il} e n_{kl} da matriz N e repassar uma unidade de n_{il} para n_{kl} . Assim, um caminhão l deixa de realizar uma viagem em uma frente i para realizá-la em outra frente k . Restrições de compatibilidade entre equipamentos são respeitadas, havendo realocação de viagens apenas quando houver compatibilidade entre eles.

Movimento Realocar Viagem de uma Frente - $N^{VF}(s)$: Duas células n_{il} e n_{ik} da matriz N são selecionadas e uma unidade de n_{il} é realocada para n_{ik} . Isto é, esse movimento consiste em realocar uma viagem de um caminhão l para um caminhão k que esteja operando na frente i . Restrições de compatibilidade entre equipamentos são respeitadas, havendo realocação de viagens apenas quando houver compatibilidade entre eles.

Movimento Operação Frente - $N^{OF}(s)$: Este movimento consiste em remover o equipamento de carga que está operando na frente i . O movimento remove todas as viagens feitas nessa frente, deixando o equipamento de carga inativo. O equipamento retorna à operação assim que uma nova viagem de um caminhão é associada com ela.

Movimento Operação Caminhão - $N^{OC}(s)$: Consiste em selecionar uma célula n_{il} da matriz N e zerar seu conteúdo, isto é, retirar de atividade um caminhão l que esteja operando em uma frente i .

Movimento Troca de Viagens - $N^{VT}(s)$: Duas células da matriz N são selecionadas e uma unidade de uma célula passa para a outra, isto é, uma viagem de um caminhão a uma frente passa para outro caminhão a outra frente.

Movimento Troca de Carregadeiras - $N^{CT}(s)$: Duas células distintas y_i e y_k da matriz Y tem seus valores permutados, ou seja, os equipamentos de carga que operam nas frentes i e k são trocados. Analogamente ao movimento CG , os equipamentos de carga são trocados, mas as viagens feitas às frentes não são alteradas. Para manter a compatibilidade entre carregadeiras e caminhões, as viagens feitas a frentes com equipamentos de carga incompatíveis são removidas.

A Figura 11 mostra uma possível solução para o problema.

$$s = \left[\begin{array}{c|cccc} 1 & 2 & 4 & 3 & 0 \\ -1 & 0 & 0 & 0 & 0 \\ 3 & 1 & 0 & 3 & 2 \\ 2 & -1 & 4 & 2 & 1 \end{array} \right]$$

Figura 11: Exemplo de Solução para o POLAD

Desta forma, a Figura 12 mostra como ficaria a solução da Figura 11 após uma aplicação aleatória de cada um dos movimentos descritos.

$s \oplus m^{NV} = \left[\begin{array}{c cccc} 1 & 2 & 4 & 3 & 0 \\ -1 & 0 & 0 & 0 & 0 \\ 3 & 1 & 0 & *4 & 2 \\ 2 & -1 & 4 & 2 & 1 \end{array} \right]$	$s \oplus m^{CG} = \left[\begin{array}{c cccc} *3 & *1 & *0 & *3 & *2 \\ -1 & 0 & 0 & 0 & 0 \\ *1 & *2 & *4 & *3 & *0 \\ 2 & -1 & 4 & 2 & 1 \end{array} \right]$
$s \oplus m^{VC} = \left[\begin{array}{c cccc} 1 & 2 & *3 & 3 & 0 \\ -1 & 0 & 0 & 0 & 0 \\ 3 & 1 & *1 & 3 & 2 \\ 2 & -1 & 4 & 2 & 1 \end{array} \right]$	$s \oplus m^{VF} = \left[\begin{array}{c cccc} 1 & *1 & *5 & 3 & 0 \\ -1 & 0 & 0 & 0 & 0 \\ 3 & 1 & 0 & 3 & 2 \\ 2 & -1 & 4 & 2 & 1 \end{array} \right]$
$s \oplus m^{OF} = \left[\begin{array}{c cccc} 1 & 2 & 4 & 3 & 0 \\ -1 & 0 & 0 & 0 & 0 \\ 3 & *0 & *0 & *0 & *0 \\ 2 & -1 & 4 & 2 & 1 \end{array} \right]$	$s \oplus m^{OC} = \left[\begin{array}{c cccc} 1 & 2 & 4 & *0 & 0 \\ -1 & 0 & 0 & 0 & 0 \\ 3 & 1 & 0 & 3 & 2 \\ 2 & -1 & 4 & 2 & 1 \end{array} \right]$
$s \oplus m^{VT} = \left[\begin{array}{c cccc} 1 & *1 & 4 & 3 & 0 \\ -1 & 0 & 0 & 0 & 0 \\ 3 & 1 & 0 & *4 & 2 \\ 2 & -1 & 4 & 2 & 1 \end{array} \right]$	$s \oplus m^{CT} = \left[\begin{array}{c cccc} *3 & 2 & 4 & 3 & 0 \\ -1 & 0 & 0 & 0 & 0 \\ *1 & 1 & 0 & 3 & 2 \\ 2 & -1 & 4 & 2 & 1 \end{array} \right]$

Figura 12: Exemplo de aplicação dos movimentos

5.4 Função de avaliação

Na abordagem multiobjetivo temos os seguintes objetivos conflitantes:

- Minimizar o número de caminhões necessários para o processo de produção
- Minimizar os desvios de produção
- Minimizar os desvios dos parâmetros de qualidade

Na abordagem mono-objetivo do POLAD, a função objetivo é dada pela equação (5.1), que representa a soma ponderada desses três objetivos:

$$f^{MP}(s) = \sum_{j \in T} \lambda_j^- d_j^- + \sum_{j \in T} \lambda_j^+ d_j^+ + \alpha^- P_m^- + \alpha^+ P_m^+ + \beta^- P_e^- + \beta^+ P_e^+ + \sum_{l \in V} \omega_l U_l \quad (5.1)$$

Como uma solução s pode não respeitar todas as restrições, penalizamos uma solução de acordo com a equação (5.2):

$$I(s) = f^P(s) + \sum_{j \in P} f_j^q(s) + \sum_{l \in T} f_l^u(s) + \sum_{k \in S} f_k^c(s) \quad (5.2)$$

em que:

$f^P(s)$ avalia s quanto ao desrespeito aos limites de produção estabelecidos para a quantidade de minério e estéril;

$f_j^q(s)$ avalia s quanto à inviabilidade em relação ao j -ésimo parâmetro de controle;

$f_l^u(s)$ avalia s quanto ao desrespeito do atendimento da taxa de utilização máxima do l -ésimo caminhão;

$f_k^c(s)$, que avalia s quanto ao desrespeito aos limites de produtividade da carregadeira k .

Logo, obtêm-se a função de avaliação dada pela Eq. (5.3):

$$f(s) = f^{MP}(s) + I(s) \quad (5.3)$$

Na equação (5.3), a primeira parcela é a função objetivo propriamente dita, $f^{MP}(s)$ (equação do modelo de programação matemática de Souza et al. (2010)), e a segunda é composta pelas funções que penalizam a ocorrência de inviabilidade na solução corrente. Assim, a função f mensura o desvio dos objetivos considerados e penaliza o não atendimento às restrições do problema.

Decompondo a função (5.1) em três objetivos conflitantes, temos:

$$f^{MP}(s) = \underbrace{\sum_{j \in T} \lambda_j^- d_j^- + \sum_{j \in T} \lambda_j^+ d_j^+}_{z_1(s)} + \underbrace{\alpha^- P_m^- + \alpha^+ P_m^+ + \beta^- P_e^- + \beta^+ P_e^+}_{z_2(s)} + \underbrace{\sum_{l \in V} \omega_l U_l}_{z_3(s)} \quad (5.4)$$

sendo

- Qualidade dos parâmetros de qualidade na mistura

$$z_1(s) = \sum_{j \in T} \lambda_j^- d_j^- + \sum_{j \in T} \lambda_j^+ d_j^+ \quad (5.5)$$

- Desvios de Produção

$$z_2(s) = \alpha^- P_m^- + \alpha^+ P_m^+ + \beta^- P_e^- + \beta^+ P_e^+ \quad (5.6)$$

- Utilização dos Caminhões

$$z_3(s) = \sum_{l \in V} \omega_l U_l \quad (5.7)$$

Por fim, segue a função de avaliação z , descrita pela equação (5.8), com os três objetivos a serem minimizados:

$$z(s) = (z_1(s) + I_1(s), z_2(s) + I_2(s), z_3(s) + I_3(s)) \quad (5.8)$$

5.5 Algoritmo proposto

O algoritmo proposto neste trabalho, denominado GRASP-2PPLS, combina os procedimentos GRASP e Two-phase Pareto Local Search com VNS - 2PPLs (LUST; TEGHEM; TUYTTENS, 2011).

Algoritmo 7: 2PPLs com VNS

Entrada: $graspMax$; Vizinhanças $\mathcal{N}_k(x)$

Saída: Conjunto Eficiente X_e

```

1  $P_0 \leftarrow \text{ConstroiConjuntoInicial}(graspMax)$ 
2  $X_e \leftarrow \text{2PPLs-VNS}(P_0, \mathcal{N}_k(x))$ 
3 retorna  $X_e$ 

```

Um conjunto de soluções não-dominadas inicial (linha 1 do Algoritmo 7) é gerada pelo procedimento parcialmente guloso GRASP, conforme detalhado no Algoritmo 8, as etapas deste algoritmo foram descritas na Seção 5.2. O procedimento `addSolution` (linha 5 do Algoritmo 8), já detalhado na Seção 3.3.4.2, adiciona as soluções criadas pelo procedimento GRASP na população X_e .

Algoritmo 8: ConstroiConjuntoInicial

Entrada: $graspMax$;

Saída: Aproximação de um conjunto eficiente X_e

```

1 para  $i \leftarrow 1$  até  $graspMax$  faça
2    $s_w \leftarrow \text{ConstróiSoluçãoEstéril}()$ 
3   Gere um número aleatório  $\gamma \in [0, 1]$ 
4    $s_i \leftarrow \text{ConstróiSoluçãoMinério}(s_w, \gamma)$ 
5    $\text{addSolution}(X_e, s_i, f(s_i))$ 
6 fim
7 retorna  $X_e$ 

```

O algoritmo GRASP-2PPLS (Algoritmo 7) segue a mesma estrutura proposta por Lust, Teghem e Tuytens (2011), na qual o procedimento 2PPLS, já detalhado na seção 3.3.4.2, é utilizado em conjunto com um mecanismo de troca de vizinhanças, espelhado no método VNS. O método 2PPLS é composto de duas fases. Na primeira, um conjunto inicial diversificado e com uma boa aproximação dos extremos de um conjunto eficiente é gerado (linha 1 do Algoritmo 7). Já na segunda fase, aplica-se o método 2PPLS (linha 2 do Algoritmo 7) no conjunto eficiente gerado na linha 1.

5.6 Algoritmos e dados para geração de novos problemas-teste

Os dados a seguir foram fornecidos por uma empresa de extração de minério brasileira.

A Tabela 2 apresenta os valores dos parâmetros encontrados nas frentes de extração desta empresa. A variável $MinI_j^{min}$ representa o valor mínimo do parâmetro j . Já as variáveis $MinI_j^{max}$, $MinI_j^{Desv.Pad.}$ e $MinI_j^{media}$ representam, respectivamente, o valor máximo, a média e o desvio padrão do parâmetro j em todo conjunto de frentes do Minério I nesta empresa. A mesma nomenclatura é utilizada para o minérios do tipo II e III.

Tabela 2: Parâmetros

		Par1	Par2	Par3	Par4	Par5	Par6	Par7	Par8
Minério I	$MinI_j^{min}$	45,32	12,68	0,402	0,010	17,44	0,38	7,12	42,01
	$MinI_j^{max}$	59,55	31,92	1,693	0,107	52,46	2,17	45,90	87,64
	$MinI_j^{Desv.Pad.}$	3,62	5,05	0,248	0,015	7,69	0,36	7,28	7,53
	$MinI_j^{media}$	52,47	22,29	0,893	0,028	31,78	0,94	28,06	57,31
Minério II	$MinII_j^{min}$	56,39	1,98	1,696	0,081	4,66	1,05	10,64	24,30
	$MinII_j^{max}$	60,40	6,75	5,297	0,249	21,50	4,12	21,07	29,16
	$MinII_j^{Desv.Pad.}$	1,02	1,39	1,160	0,058	4,97	0,95	2,73	1,79
	$MinII_j^{media}$	58,03	3,84	3,773	0,144	10,06	2,44	14,04	26,62
Minério III	$MinIII_j^{min}$	58,32	2,11	0,634	0,018	3,60	0,53	13,08	25,59
	$MinIII_j^{max}$	67,44	9,43	3,728	0,181	29,19	4,21	42,31	73,18
	$MinIII_j^{Desv.Pad.}$	2,29	1,79	0,795	0,046	4,73	0,79	7,98	12,78
	$MinIII_j^{media}$	62,78	4,59	1,573	0,070	10,35	1,43	24,63	44,22

Neste conjunto de dados, foram determinadas duas frentes de descarga, cada um com os seus respectivos limites. A Tabela 3 apresenta a meta, o limite inferior e superior dos parâmetros considerados. Será denominada por LI_j^d , M_j^d , LS_j^d cada elemento dessa tabela, sendo d a descarga e j o parâmetro de controle. Deste modo, o Limite Superior do parâmetro 4 na descarga 2 é representado pela variável LS_4^2 .

Tabela 3: Limites dos Parâmetros

Limite Inferior - LI								
Descarga	Par1	Par2	Par3	Par4	Par5	Par6	Par7	Par 8
1	52	0	0	0,000	0	0	19,62	47,00
2	62	4	0	0,040	0	0	0	0
Meta - M								
Descarga	Par1	Par2	Par3	Par4	Par5	Par6	Par7	Par 8
1	53,1	20,08	1,120	0,0469	17,58	1,05	25	55
2	63,3	4,49	1,773	0,0468	3	1	26	43
Limite Superior - LS								
Descarga	Par1	Par2	Par3	Par4	Par5	Par6	Par7	Par 8
1	55	21,00	1,387	0,10	31,99	1,40	27,86	60,00
2	64	100	100	100	100	100	100	100

As Tabelas 4 e 5 apresentam, respectivamente, as frotas de caminhões e carregadeiras utilizadas na extração de desses três tipos de minérios. Denomina-se por nT_f e cT_f o número de caminhões disponíveis da frota f e suas respectivas capacidades. Da mesma forma, as variáveis nS_f e pS_f representam a quantidade e a produção das carregadeiras da frota f .

Tabela 4: Frota de Caminhões

Caminhões		
Frota	Quantidade - nT_f	Capacidade - cT_f
1	15	230
2	8	45

Tabela 5: Frota de Carregadeiras

Carregadeiras		
Frota	Quantidade - nS_f	Prod./h - pS_f
1	5	1500
2	1	2000
3	3	350

Finalmente, a Tabela 6 apresenta a compatibilidade entre caminhões, carregadeiras e descargas. Cada célula dessa tabela será representada pela variável $comp_{(fT, fS)}^d$, sendo d a descarga, fT a frota do caminhão e fS a frota da carregadeira. Assim sendo, verifica-se que a variável $comp_{(2,3)}^2$ possui o valor 1, ou seja, é possível que os caminhões da frota 2 sejam carregados pelas carregadeiras da frota 3 e descarreguem na descarga 2.

Tabela 6: Compatibilidade Caminhões x Carregadeiras x Descargas

		Carregadeiras		
	Caminhões	1	2	3
Descarga1	1	1	1	0
	2	0	0	0
Descarga2	1	1	1	1
	2	0	0	1

O algoritmo 9 descreve o procedimento utilizado para geração de um problema-teste proposto neste trabalho.

Algoritmo 9: GeraProblemaTeste

Entrada: Número frentes $|F|$ e matriz parâmetros de controle de qualidade *frentesTeores*

Entrada: $txFrentesMinerioEsteril$, $nTiposCam$, $nTiposCarregadeiras$ e $prodMax$

Entrada: p_{QU}^{LI} , p_{QU}^{LS} , p_{Est}^{LI} , p_{Est}^M , p_{Est}^{LS} , p_{Min}^{LI} , p_{Min}^M e p_{Min}^{LS}

Entrada: $txCamFre$ taxa de caminhões por frente, $pCam$ e $pCarreg$ número percentual de caminhões e carregadeiras

Saída: Problema-teste POLAD

1 **Imprime** $opmTesteInstance(|F|, frentesTeores, geraDadosBasicos(|F|, p_{QU}^{LI}, p_{QU}^{LS}, p_{Est}^{LI}, p_{Est}^M, p_{Est}^{LS}, p_{Min}^{LI}, p_{Min}^M, p_{Min}^{LS}, txCamFre, pCam, pCarreg))$

O algoritmo 9 possui como dados primordiais de entrada o número de frentes e a matriz de teores associada a essas frentes. A matriz *frentesTeores* é gerada utilizando os dados $MinI_j^{max}$, $MinI_j^{min}$, $MinI_j^{Desv.Pad.}$ e $MinI_j^{media}$ apresentados na Tabela 2, analogamente, utilizando, também, os limites estabelecidos para o minérios II e III.

Algoritmo 10: geraDadosBasicos

Entrada: $txFrentesMinerioEsteril$, $nTiposCam$, $nTiposCarregadeiras$ e $prodMax$

Entrada: p_{QU}^{LI} , p_{QU}^{LS} , p_{Est}^{LI} , p_{Est}^M , p_{Est}^{LS} , p_{Min}^{LI} , p_{Min}^M e p_{Min}^{LS}

Entrada: $txCamFre$, $pCam$, $pCarreg$ e $carregProducaoMax$

Saída: $tCarregadeiras$, $tCam$ e $(p_{Est}^{LI}, p_{Est}^M, p_{Est}^{LS}, p_{Min}^{LI}, p_{Min}^M, p_{Min}^{LS}) \times prodMax$

Saída: $estMin$, $frente_{QU}$, $total_{QU}^{Min}$, $total_{QU}^{Est}$ e $tCiclo$

```

1  para  $c \leftarrow 1$  até  $nTiposCam$  faça
2       $nCam = pCam[c] \times |F| \times txCamFre$ 
3      para  $i \leftarrow 1$  até  $nCam$  faça
4           $tCam \leftarrow tCam \cup \{c\}$ 
5      fim
6  fim
7   $prodDemanda \leftarrow prodMax$ 
8  enquanto  $prodDemanda > 0$  faça
9      Gere um número aleatório  $x \in [0, 1]$ 
10     para  $s \leftarrow 1$  até  $nTiposCarregadeiras$  faça
11          $x- = pCarreg[s]$ 
12         se  $x < 0$  então
13              $prodDemanda \leftarrow prodDemanda - carregProducaoMax[s]$ 
14              $tCarregadeiras \leftarrow tCarregadeiras \cup \{s\}$ 
15         fim
16     fim
17 fim
18 Atualize  $prodMax$ 
19  $li_{QU} \leftarrow p_{QU}^{LI} \times prodMax$ ;  $ls_{QU} \leftarrow p_{QU}^{LS} \times prodMax$ 
20 para  $i \leftarrow 1$  até  $|F|$  faça
21     se  $i < |F| \times txFrentesMinerio$  então  $estMin[i] \leftarrow 1$ 
22     senão
23          $estMin[i] \leftarrow 0$ 
24     fim
25      $qu \leftarrow rand() \bmod (ls_{QU} - li_{QU} + 1) + li_{QU}$ 
26      $frente_{QU}[i] \leftarrow qu$ 
27     se  $estMin[i] == 1$  então  $total_{QU}^{Min} \leftarrow total_{QU}^{Min} + qu$ 
28     senão
29          $total_{QU}^{Est} \leftarrow total_{QU}^{Est} + qu$ 
30     fim
31      $tCiclo[i] \leftarrow geraTempoCiclo()$ 
32 fim
33 Retorne  $(tCarregadeiras, tCam, (p_{Est}^{LI}, p_{Est}^M, p_{Est}^{LS}, p_{Min}^{LI}, p_{Min}^M, p_{Min}^{LS}) \times prodMax, estMin, frente_{QU}, total_{QU}^{Min}, total_{QU}^{Est}, tCiclo)$ 

```

O Algoritmo 10 apresenta como são definidos o número de caminhões e carregadeiras para cada tipo apresentado nas Tabelas 4 e 5, bem como, a massa de minério ou estéril das frentes e o tempo de ciclo dos caminhões.

As variáveis p_{Est}^{LI} , p_{Est}^M , p_{Est}^{LS} , p_{Min}^{LI} , p_{Min}^M e p_{Min}^{LS} mensuram os percentuais, com base na produção máxima $prodMax$, para as metas e limites das frentes de minério/estéril. A linha 33 retorna esses valores multiplicados pela $prodMax$ atualizada. As variáveis p_{QU}^{LI} e p_{QU}^{LS} são utilizadas nas linhas 19 e 25 de forma a determinar os limites de massa dessas frentes. A variável $carregProducaoMax$ indica a capacidade máxima de cada tipo de carregadeira.

A linha 2 do Algoritmo 10 determina o número de caminhões do tipo i que serão disponibilizados no problema-teste que está sendo gerado. Esse valor é determinado pela multiplicação entre a porcentagem de caminhões do tipo c ($pCam[c]$), o número de $|F|$ e a taxa de caminhões por frente $txCamFre$. A linha 4 empilha os caminhões no vetor $tCam$.

Faz-se necessário determinar o número de carregadeiras necessárias para o processo de produção. Essa etapa ocorre entre as linhas 8 e 16 do Algoritmo 10, são geradas carregadeiras até que a produção desejada ($prodDemanda$) seja atendida. A linha 9 gera um número aleatório entre o intervalo $[0, 1]$, esse número determina os tipos de carregadeiras que serão geradas em cada iteração. A linha 11 subtrai o valor da variável x pela porcentagem de carregadeiras do tipo s , um valor de x igual a 1 resulta no empilhamento de uma carregadeira de cada tipo. A linha 13 atualiza a produção em demanda ($prodDemanda$) e a linha 14 empilha as carregadeiras no vetor $tCarregadeiras$.

A linha 18 atualiza a produção máxima após o empilhamento das carregadeiras. Esse valor é atualizado com a soma da produção de cada carregadeira inserida no problema-teste.

A linha 31 do Algoritmo 9 chama o gerador de número reais $geraTempoCiclo()$ que determina o tempo ciclo da frente i .

Por fim, a linha 1 do Algoritmo 9 utiliza todos os dados de entrada juntamente com aqueles gerados pelo procedimento $geraDadosBasicos$ (Algoritmo 10), realizando a impressão desses dados no formato adequado para execução dos algoritmos e do modelo matemático, apresentados nos Capítulo 5.

6 *Resultados Parciais*

Descrevem-se, neste capítulo, alguns resultados parciais do algoritmo heurístico proposto neste trabalho, nomeado GRASP-2PPLS. Na Seção 6.1 são descritos os problemas-teste usados para comparar o desempenho dos algoritmos. Na Seção 6.2 são apresentados os valores dos parâmetros e pesos utilizados. Na Seção 6.3 é apresentado o ambiente de desenvolvimento dos algoritmos, bem como o de teste dos mesmos. Na Seção 6.4 são mostrados os resultados computacionais dos algoritmos testados.

6.1 Descrição dos problemas-teste

Para testar o algoritmo desenvolvido neste trabalho, foi usado um conjunto de 8 problemas-teste da literatura, disponíveis em <http://www.decom.ufop.br/prof/marcone/projects/mining.html>. Estes problemas-teste foram os mesmos utilizados em Souza et al. (2010) para validar o algoritmo *GGVNS*.

Os melhores resultados da literatura para os problemas-teste analisados são apresentados na Tabela 7. Na coluna “Opt.” indicamos por “✓” as instâncias nas quais o otimizador matemático CPLEX 11.02 obteve o valor ótimo da função.

Tabela 7: Melhores valores

Problema-Teste	Melhor da Literatura	Opt. [†]
opm1	227.12	
opm3	256.37	
opm2	164,027.15	✓
opm4	164,056.68	✓
opm5	227.04	
opm6	236.58	
opm7	164,017.46	✓
opm8	164,018.65	✓

6.2 Pesos e parâmetros utilizados

Os pesos adotados na função de avaliação são apresentados na Tabela 8 e são os mesmos de Costa (2005).

Tabela 8: Pesos adotados

Pesos	Descrição	Valor
γ	Penalidade por tonelada abaixo dos limites inferiores ou acima dos limites superiores de produção (estéril/minério)	1000
α	Penalidade por tonelada abaixo ou acima da meta de produção (estéril/minério)	100
Φ_j	Penalidade por tonelada abaixo do limite mínimo de especificação, ou acima do limite de especificação para os parâmetros de controle j	100
λ	Penalidade por tonelada abaixo ou acima da meta de qualidade	1
ω	Penalidade pelo uso de um caminhão	1
T_{x_l}	Taxa máxima de utilização de um caminhão	85%
Ω^+	Penalidade por utilização acima da taxa máxima de utilização de um caminhão	1000
Ψ_k^-	Penalidade por tonelada abaixo do limite mínimo de produção, ou acima do limite de produção para a carregadeira k	1000

6.3 Ambiente de desenvolvimento

Os experimentos foram testados em um microcomputador DELL XPS 8300 Intel Core i7-2600, 8MB Cache, 3.4GHz, 16GB RAM, sob sistema operacional Ubuntu 10.10.

Os algoritmos foram implementados em C++ com auxílio do *framework* OptFrame (COELHO et al., 2011, 2010)¹ (COELHO et al., 2011, 2010). Essa escolha foi devido a seu arcabouço de fácil utilização, que inclui:

- Representações de soluções e populações;
- Arcabouços de métodos heurísticos e metaheurísticos;
- Interface para análise dos resultados;
- Fórum de discussão com os colaboradores do projeto.

O *framework* OptFrame é, basicamente, uma estrutura computacional que agiliza o desenvolvimento de algoritmos heurísticos. O objetivo do *framework* é fornecer uma simples interface em C++ para componentes comuns de metaheurísticas baseadas em trajetória e população,

¹ disponível em <http://sourceforge.net/projects/optframe/>

aplicadas a problemas de otimização combinatória. Uma vez que muitos métodos são comuns na literatura, o OptFrame fornece uma implementação eficiente para versões simples de algumas heurísticas e metaheurísticas, todavia, o usuário pode desenvolver versões “mais inteligentes” e mais robustas, aplicadas ao seu problema específico. Além disso, o OptFrame possui suporte ao desenvolvimento de sistemas em paralelo, tanto para memória compartilhada quanto para memória distribuída. OptFrame tem sido aplicado com sucesso para modelar e resolver vários problemas combinatórios, mostrando um bom equilíbrio entre a flexibilidade e eficiência.

6.4 Resultados e análise

De modo a comparar o algoritmo desenvolvido neste trabalho, GRASP-2PPLS, foi realizada uma comparação deste algoritmo com dois algoritmos da literatura propostos por Coelho et al. (2012). O primeiro deles, denominado GRASP-MOVNS, combina os procedimentos GRASP e *Multiobjective Variable Neighborhood Search* (MOVNS). O segundo algoritmo, denominado GRASP-NSGAII-PR, combina os procedimentos GRASP, *Nondominated Sorting Genetic Algorithm II* (NSGA-II) e o procedimento de Reconexão por Caminhões (PR, do inglês *Path Relinking*), como operador de cruzamento.

Para verificação do desempenho do algoritmo desenvolvido, foram usadas as métricas de Hipervolume (ZITZLER; THIELE, 1998), Espaçamento (SCHOTT, 1995), Cobertura (ZITZLER; THIELE, 1998) e Cardinalidade (ZITZLER; THIELE, 1998). As Tabelas 9, 10, 11 e 12 mostram os resultados obtidos. A coluna “Instância” indica a instância utilizada; a coluna “Melhor” indica o melhor valor da métrica analisada obtido nas 30 execuções e as colunas “Média” e “Desv. Pad.” indicam, respectivamente, a média e desvio padrão da amostra.

A Tabela 9 mostra os valores da métrica hipervolume para o algoritmo desenvolvido neste trabalho e os dois algoritmos multiobjetivos da literatura. O cálculo dessa métrica foi realizado com o auxílio da ferramenta computacional de Beume et al. (2009). Analisando-se os resultados, verifica-se que o algoritmo GRASP-2PPLS foi capaz de obter os conjuntos de soluções não-dominadas com as melhores convergências em seis instâncias, além de apresentar as melhores médias em cinco delas. Ressalta-se, ainda, que o algoritmo GRASP-MOVNS obteve boas frentes de Pareto nas instâncias opm1, opm2, opm5 e opm6.

A Tabela 10 apresenta a comparação dos resultados utilizando a métrica de Espaçamento. Percebe-se que a variante GRASP-2PPLS obteve os melhores resultados, pois foi capaz de gerar conjuntos mais uniformemente espaçados.

Os algoritmos baseados em procedimentos de busca local, GRASP-2PPLS e GRASP-MOVNS,

Tabela 9: GRASP-2PPLS $\times$ GRASP-MOVNS $\times$ GRASP-NSGAI-PR: Hipervolume

Instância	Hipervolume (10^9)								
	GRASP-2PPLS			GRASP-MOVNS			GRASP-NSGAI-PR		
	Melhor	Média	Desv. Pad.	Melhor	Média	Desv. Pad.	Melhor	Média	Desv. Pad.
opm1	81,71	78,13	1,54	80,37	78,23	1,12	78,43	75,68	0,89
opm2	76,47	75,15	0,56	79,49	76,12	1,49	75,26	73,15	1,05
opm3	77,94	71,17	4,18	76,97	70,42	2,18	69,95	64,98	2,38
opm4	76,80	67,45	4,67	75,61	66,94	3,00	65,09	59,75	2,89
opm5	81,88	78,04	1,41	80,91	78,67	1,37	77,36	75,68	0,72
opm6	80,00	77,97	0,91	80,43	77,23	1,20	77,19	74,32	0,83
opm7	78,78	75,28	1,88	78,48	73,39	3,54	76,15	73,69	2,28
opm8	78,88	73,77	2,26	78,57	72,96	3,90	76,05	73,62	1,57

Tabela 10: GRASP-2PPLS $\times$ GRASP-MOVNS $\times$ GRASP-NSGAI-PR: Espaçamento

Instância	Espaçamento								
	GRASP-2PPLS			GRASP-MOVNS			GRASP-NSGAI-PR		
	Melhor	Média	Desv. Pad.	Melhor	Média	Desv. Pad.	Melhor	Média	Desv. Pad.
opm1	1177,93	4566,51	5058,15	1662,54	2868,67	1099,72	2589,54	6924,42	5007,23
opm2	520,91	1166,88	389,56	1066,13	2041,59	970,75	1812,14	6557,20	5030,65
opm3	4,00	3824,96	3443,70	2250,96	6188,99	3331,59	4850,50	14368,62	5173,60
opm4	3,77	2243,74	1235,66	2411,89	8232,88	3957,29	4865,89	18295,27	7046,64
opm5	1251,93	4795,68	3882,54	1400,84	2563,71	1057,14	2578,39	7544,48	5529,35
opm6	697,34	1212,87	344,61	1087,95	2025,84	695,00	2218,05	5963,77	4650,80
opm7	0,00	6676,54	8042,17	2917,66	8311,12	3437,74	4924,61	14673,57	4646,02
opm8	0,00	6381,38	7678,35	2258,14	8045,67	4684,92	4974,85	14874,73	5897,70

apresentaram os melhores resultados em relação às métricas de hipervolume e espaçamento. Deste modo, as métricas de Cobertura e Cardinalidade foram aplicadas, apenas, entre esses dois algoritmos. A Tabela 11 mostra o resultado da comparação entre esses dois algoritmos com relação à Cobertura e a Tabela 12, em relação à Cardinalidade.

Para utilizar a métrica de cardinalidade, foi criado um conjunto Pareto Referência ($|Ref|$) obtido em duas etapas. Primeiramente, foram realizadas 30 execuções de duas horas com cada uma das instâncias. Essa bateria de teste consistiu na execução dos algoritmos GRASP-2PPLS e GRASP-MOVNS, alternadamente. Dado um conjunto inicial de soluções não-dominadas, o algoritmo GRASP-2PPLS teve como critério de parada o ótimo local em relação às estruturas de vizinhanças utilizadas; em seguida, o conjunto de soluções não-dominadas obtido era entregue ao algoritmo GRASP-MOVNS que refinava essa fronteira de Pareto com um tempo máximo de processamento de 5 minutos e, novamente, o algoritmo GRASP-2PPLS era acionado. Por fim, o conjunto Pareto Referência foi obtido com a união de todas as fronteiras de Pareto encontradas nas 30 execuções, mais as fronteiras obtidas em todos os experimentos executados neste trabalho. Os conjuntos $|D_1|$ e $|D_2|$, representam as cardinalidades dos conjuntos de Pareto obtidos com a união das soluções não-dominadas encontradas pelos algoritmos GRASP-2PPLS e GRASP-MOVNS, respectivamente. Tais conjuntos referências encontram-se disponíveis em: <http://www.decom.ufop.br/prof/marcone/projects/mining.html>.

Tabela 11: GRASP-2PPLS $\times$ GRASP-MOVNS: Cobertura							
Instância	Tempo (min)	Cobertura					
		$C(\text{GRASP-2PPLS}, \text{GRASP-MOVNS})$			$C(\text{GRASP-MOVNS}, \text{GRASP-2PPLS})$		
		Melhor	Média	Desv. Padrão	Melhor	Média	Desv. Padrão
opm1	2	1,00	0,79	0,14	0,13	0,02	0,03
opm2	2	1,00	0,80	0,03	0,10	0,06	0,11
opm3	2	1,00	0,65	0,21	0,25	0,02	0,05
opm4	2	0,95	0,57	0,23	0,35	0,04	0,09
opm5	2	0,73	0,38	0,14	0,57	0,22	0,14
opm6	2	0,96	0,79	0,11	0,18	0,07	0,06
opm7	2	0,90	0,34	0,25	0,50	0,11	0,17
opm8	2	0,90	0,41	0,29	1,00	0,11	0,25

Os resultados apresentados na Tabela 11 mostram que o algoritmo GRASP-2PPLS foi capaz de gerar melhores conjuntos que o algoritmo GRASP-MOVNS em 7 instâncias.

A Tabela 12 mostra o número de soluções não-dominadas obtidas pelos algoritmos GRASP-2PPLS e GRASP-MOVNS, bem como o número de soluções pertencentes aos conjuntos de referência.

Analisando a Tabela 12 nota-se que o algoritmo GRASP-2PPLS foi capaz de encontrar um total de 289 soluções pertencentes aos conjuntos de referência. Já o algoritmo GRASP-MOVNS foi capaz de encontrar apenas 57 soluções. Este resultado mostra a robustez do algoritmo

Tabela 12: GRASP-2PPLS $\times$ GRASP-MOVNS: Cardinalidade

Instância	$ Ref $	GRASP-2PPLS		GRASP-MOVNS	
		$ D_1 $	$ Ref \cap D_1 $	$ D_2 $	$ Ref \cap D_2 $
opm1	234	117	18	141	4
opm2	271	175	35	144	0
opm3	151	125	33	74	1
opm4	148	152	9	87	1
opm5	172	99	41	128	38
opm6	209	200	97	121	3
opm7	104	62	17	78	2
opm8	139	104	39	69	8
Total de Soluções	1428	1034	289	842	57

GRASP-2PPLS de encontrar, em tempo reduzido, um considerável número de soluções não-dominadas pertencentes às aproximações para os conjuntos Pareto de Referência.

Por fim, buscou-se verificar se esta nova proposta multiobjetivo conseguiria gerar uma boa solução mono-objetivo. A Tabela 13 compara o desempenho do algoritmo GRASP-2PPLS frente ao do algoritmo GGVNS de Souza et al. (2010), com relação às melhores soluções mono-objetivo obtidas. Analisando-a, percebe-se que o algoritmo GRASP-2PPLS mostrou-se competitivo com o algoritmo mono-objetivo da literatura GGVNS, uma vez que obteve soluções melhores ou iguais em sete instâncias.

Tabela 13: Comparação de resultados: GRASP-2PPLS $\times$ GGVNS

Instância	Tempo	GRASP-2PPLS	GGVNS
		Melhor	Melhor
opm1	2	228,12	230,12
opm2	2	256,37	256,37
opm3	2	164046,32	164039,12
opm4	2	164074,32	164099,66
opm5	2	227,04	228,09
opm6	2	236,35	236,58
opm7	2	164018,81	164021,28
opm8	2	164022,63	164023,73

6.5 Geração de novos problemas-teste

Os pesos adotados na geração dos problemas-teste foram determinados de forma empírica. A Tabela 14 apresenta esses valores.

O procedimento *geraTempoCiclo()* foi calibrado de forma empírica, sendo o valor retornado por esse procedimento igual a $500 + rand() \bmod 1000) / 100$. A função *rand()* gera um

Tabela 14: Pesos geração problemas-teste

Pesos	Descrição	Valor
$prodMax$	Produção máxima esperada (t.)	15000
$pLIQU$	Percentual da produção que determina o limite inferior da massa de uma frente	0.05
$pLSQU$	Percentual da produção que determina o limite superior da massa de uma frente	0.3
$pPLEst$	Percentual da produção máxima para limite inferior do estéril	0.22
$pPREst$	Percentual da produção máxima para a meta do estéril	0.23
$pPUEst$	Percentual da produção máxima para limite superior do estéril	0.29
$pPLMin$	Percentual da produção máxima para limite inferior do minério	0.51
$pPRMin$	Percentual da produção máxima para limite meta do minério	0.74
$pPUMin$	Percentual da produção máxima para limite superior do minério	0.89
$txCamFre$	Taxa do número de caminhões por frente	0.7
$pCam$	Porcentagem de cada frota de caminhão [45,110,230]	[0.55,0.30,0.15]
$pCarreg$	Porcentagem de cada frota de carregadeira [350,1500,200]	[0.55,0.30,0.15]

número aleatório no intervalo $[0, 1]$.

Foi gerado um novo problema-teste, denominado opm90, com os parâmetros definidos a seguir. A Tabela 15 apresenta os limites de produção minério e estéril. As Tabelas 16 e 17 apresentam as frotas de caminhões e carregadeiras desse novo problema-teste.

Por fim, a Tabela 18 apresenta os teores das frentes e os tempos de ciclo. O número de frentes $nFrentes$ foi fixado em 90, sendo 30 frentes de cada tipo de minério apresentado na Tabela 2. Foi considerada a frente de Descarga 2, onde todos os caminhões e carregadeiras são utilizados. A Tabela 6 apresenta as respectivas compatibilidades. Além disso, foi criada uma nova frota de caminhões, frota 3, com capacidade $cT_f = 110$. Essa nova frota, assim como a frota 2, possui compatibilidade com todas as carregadeiras na Descarga 2. Desta forma, têm-se $nTiposCam = 3$ e $nTiposCarregadeiras = 3$.

Tabela 15: Metas de produção minério e estéril - opm90

	Parâmetros		
	LI	M	LS
estéril	3322	3473	4379
minério	7701	11174	13439

Tabela 16: Frota de Caminhões - opm90

Caminhões		
Frota	Quantidade - nT_f	Capacidade - cT_f
1	23	230
2	86	45
3	47	110

Tabela 17: Frota de Carregadeiras - opm90

Carregadeiras		
Frota	Quantidade - nS_f	Prod./h - pS_f
1	2	2000
2	6	1500
3	6	350

6.5.1 Resultados computacionais aplicados ao novo problema-teste

Com intuito de analisar e avaliar o novo problema-teste, opm90, gerado neste trabalho, foram aplicados os algoritmos multiobjetivos GRASP-2PPLS e GRASP-MOVNS e o algoritmo mono-objetivo GGVNS de Souza et al. (2010).

As soluções obtidas pelo CPLEX 12 em duas baterias de execuções, um de duas horas e a outra de dois minutos, são apresentadas na Tabela 19.

Os resultados obtidos pelo *solver* matemático CPLEX, Tabela 19, fornecem uma dimensão do espaço de soluções, que, em 118 minutos conseguiu obter apenas mais 14 soluções inteiras. O elevado GAP, 99.63%, indica a distância entre o valor da solução inteira e o melhor nó real.

Para os algoritmos heurísticos foram executadas baterias de 30 execuções de dois minutos.

Dada a elevada dimensão do problema-teste gerado e o custo computacional da busca-local utilizando as estruturas de vizinhanças apresentadas na seção 5.3, o algoritmo GRASP-2PPLS foi visto como inaplicável.

O conjunto de soluções não-dominadas obtidos com a união das execuções do algoritmo GRASP-MOVNS é apresentado na Figura 13. Verifica-se que o número de caminhões utilizados varia entre 47 até 66, de um total de caminhões disponíveis, ou seja, pode-se diminuir o número de caminhões disponíveis neste problema-teste.

A Tabela 20 mostra uma comparação entre os resultados obtidos pelo CPLEX e os algoritmos heurísticos. Nesta Tabela é apresentada a melhor solução do CPLEX, a melhor solução mono-objetivo obtida nas 30 execuções do algoritmo GRASP-MOVNS e, para o algoritmo GGVNS,

Tabela 19: Soluções CPLEX

tempo (s)	Melhor solução inteira	Melhor nó	GAP (%)	Número de soluções inteiras
7200	11934,56	44,19	99,63	34
120	13604,02	44,19	99,68	20

é apontado também a média e o desvio padrão.

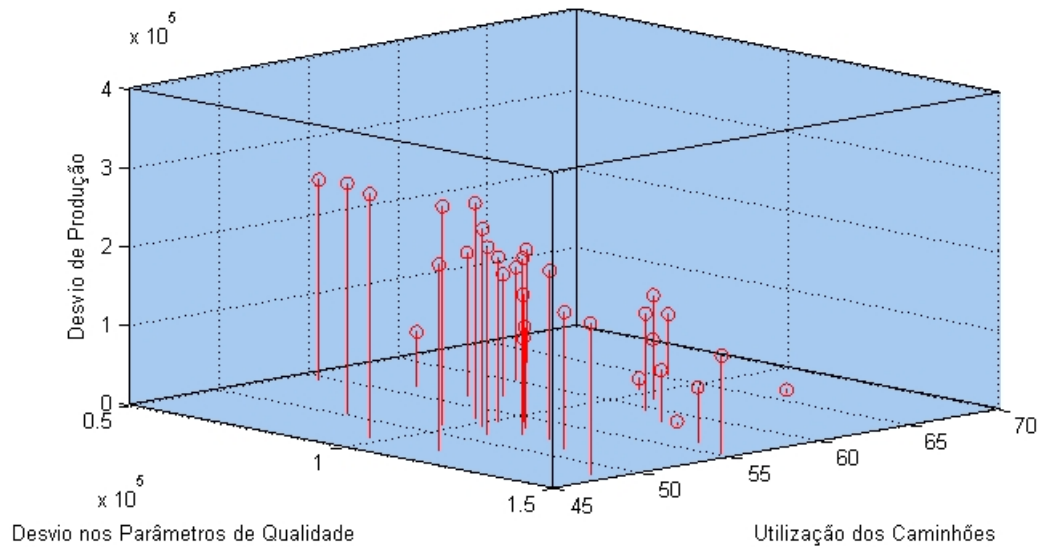


Figura 13: Conjunto com 47 soluções não-dominadas - opm90

Tabela 20: Comparação de resultados: CPLEX $\times$ GGVNS $\times$ GRASP-MOVNS

Método	Melhor	Média	Desv. Pad. (%)
CPLEX - 7200 segundos	11934,56	*	*
CPLEX - 120 segundos	13604,02	*	*
GRASP-MOVNS	58884,93	*	*
GGVNS	581397,18	641393,54	7,58%

A Tabela 20 mostra que o algoritmo multiobjetivo GRASP-MOVNS foi capaz de encontrar uma melhor solução mono-objetivo que o algoritmo da literatura GGVNS. Todavia, essa solução ainda mostra-se distante da solução encontrada pelo CPLEX, tanto em duas horas quanto em dois minutos de execução. Ressalta-se, ainda que o algoritmo GGVNS não foi capaz de encontrar nenhuma solução viável para o problema em sete das 30 execuções.

7 *Conclusões e Trabalhos Futuros*

Este trabalho teve seu foco no problema de planejamento operacional de lavra considerando alocação dinâmica de caminhões, POLAD.

O POLAD foi tratado como um problema de otimização multiobjetivo, uma vez que ele considera vários objetivos conflitantes, como: obtenção das metas de produção e qualidade para o produto formado, e minimização do número de veículos necessários ao processo produtivo.

Na abordagem multiobjetivo não há uma única solução que satisfaça a todos os objetivos. O que se procura é um conjunto de soluções não-dominadas, também chamadas de soluções eficientes, ou Fronteira de Pareto, cabendo ao tomador de decisões a escolha da solução mais adequada.

Em virtude da complexidade combinatória do problema, foi proposto um algoritmo heurístico multiobjetivo, denominado GRASP-2PPLS, que combina os procedimentos *Greedy Randomized Adaptive Search Procedures* (GRASP) e *Two-phase Pareto local search* (2PPLS).

Usando oito problemas-teste da literatura e quatro métricas de comparação, os algoritmos foram comparados com dois algoritmos da literatura. O primeiro deles, denominado GRASP-NSGAII-PR, combina os procedimentos *Nondominated Sorting Genetic Algorithm II* (NSGA-II) e o procedimento de Reconexão por Caminhões (PR, do inglês *Path Relinking*), como operador de cruzamento. O segundo algoritmo, denominado GRASP-MOVNS, combina os procedimentos GRASP e *Multiobjective Variable Neighborhood Search* (MOVNS).

O algoritmo GRASP-2PPLS foi comparado aos algoritmos GRASP-MOVNS e GRASP-NSGAII-PR, obtendo, em todos os problemas-testes utilizados, frentes de pareto mais diversificadas e com uma melhor convergência. Por fim, utilizando uma função mono-objetivo, as soluções obtidas pelo algoritmo GRASP-2PPLS foram comparadas a um algoritmo da literatura mono-objetivo, denominado GGVNS, de Souza et al. (2010). O algoritmo GRASP-MOVNS foi capaz de obter melhores soluções em quatro problemas-testes, mostrando assim o poderio deste algoritmo tanto para aplicações de otimização multiobjetivo quanto para aplicações de otimização mono-objetivo.

Dado que a tomada de decisão no problema em pauta tem que ser rápida, os resultados encontrados validam a utilização dos algoritmos propostos enquanto ferramenta de apoio à decisão.

Adicionalmente, foi desenvolvido um algoritmo gerador de novos problemas-teste. Um novo problema teste, denominado opm90, de grande porte e apenas com fins acadêmicos, foi gerado a partir de dados fornecidos por uma empresa de mineração brasileira. Resultados computacionais mostraram a ineficiência dos métodos atualmente desenvolvidos na solução de problemas de planejamento operacional de lavra de grande escala. Torna-se, assim, necessário o aperfeiçoamento das heurísticas até então desenvolvidas, implementando mecanismos de reavaliação rápida que otimizem a avaliação dos movimentos gerados dentro das busca-locais, de forma análoga às Estruturas Auxiliares de Dados propostas por Penna, Subramanian e Ochi (2011).

Para trabalhos futuros, propõe-se a criação de mais um novo conjunto de problemas-teste utilizando o embasamento teórico e prático desenvolvido neste trabalho. Propõe-se também a execução de testes exaustivos, de forma a obter conjuntos de referência com maior convergência e diversidade. Para validação desses resultados, outras métricas de validação de desempenho de algoritmos de otimização multiobjetivo devem ser implementadas. É também proposto como trabalho futuro a implementação de novos métodos e estratégias de resolução para o MOPO-LAD.

Referências

- ALARIE, S.; GAMACHE, M. Overview of solution strategies used in truck dispatching systems for open pit mines. *International Journal of Surface Mining, Reclamation and Environment*, v. 16, p. 59–76, 2002.
- ALEXANDRE, R. F. *Modelagem, Simulação da Operação e Otimização Multiobjetivo Aplicada ao Problema de Despacho de Veículos em Minas a Céu Aberto*. Dissertação (Dissertação de Mestrado) — Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Minas Gerais, Belo Horizonte, 2010.
- ALVARENGA, G. B. *Despacho ótimo de caminhões numa mineração de ferro utilizando algoritmo genético com processamento paralelo*. Dissertação (Dissertação de Mestrado) — Programa de Pós-Graduação em Engenharia Elétrica/UFGM, Belo Horizonte, 1997.
- ARAÚJO, F. C. R. *Planejamento Operacional de Lavra com Alocação Dinâmica de Caminhões: Abordagens Exata e Heurística*. Dissertação (Dissertação de Mestrado) — Programa de Pós-Graduação em Engenharia Mineral do Departamento de Engenharia de Minas da Escola de Minas da Universidade Federal de Ouro Preto, Ouro Preto, 2008.
- BACK, T.; HAMMEL, U.; SCHWEFEL, H.-P. Evolutionary computation: Comments on the history and current state. *IEEE Transactions on Evolutionary Computation*, v. 1, p. 1097–1100, 1997.
- BEUME, N. et al. On the complexity of computing the hypervolume indicator. *Evolutionary Computation, IEEE Transactions on*, v. 13, n. 5, p. 1075 –1082, 2009.
- BEYER, H. G.; SCHWEFEL, H. P. Evolution strategies - a comprehensive introduction. *Natural Computing*, v. 1, p. 3–52, 2002.
- CHANDA, E. K. C.; DAGDELEN, K. Optimal blending of mine production using goal programming and interactive graphics systems. *International Journal of Surface Mining, Reclamation and Environment*, v. 9, p. 203–208, 1995.
- COELHO, I. M. et al. A computational framework for combinatorial optimization problems. In: *VII ALIO/EURO Workshop on Applied Combinatorial Optimization*. Porto: [s.n.], 2011. p. 51–54.
- COELHO, I. M. et al. Optframe: a computational framework for combinatorial optimization problems. In: *XLII Simpósio Brasileiro de Pesquisa Operacional*. Bento Gonçalves, RS: [s.n.], 2010. p. 1–12.
- COELHO, I. M.; RIBAS, S.; SOUZA, M. J. F. Um algoritmo baseado em grasp, vnd e iterated local search para a resolução do planejamento operacional de lavra. In: *XV Simpósio de Engenharia de Produção*. Bauru/SP: [s.n.], 2008.

- COELHO, V. N. et al. Estratégias evolutivas aplicadas a um problema de programação inteira mista. In: A (Ed.). *Anais X Congresso Brasileiro de Inteligência Computacional (CBIC)*. Fortaleza/CE: [s.n.], 2011. v. 1, p. 1–8.
- COELHO, V. N. et al. Pggvns: Um algoritmo paralelo para o problema de planejamento operacional de lavra. In: *Anais XVIII Simpósio de Engenharia de Produção (SIMPEP)*. Bauru/SP: [s.n.], 2011. v. 1, p. 1–14.
- COELHO, V. N. et al. Uma abordagem multiobjetivo para o problema de planejamento operacional de lavra. In: *XVI Simpósio Brasileiro de Pesquisa Operacional - XVI CLAIO/XLIV SBPO*. Rio de Janeiro, RJ: [s.n.], 2012. p. 1–12.
- COELLO, C. C. Evolutionary multi-objective optimization: a historical view of the field. *Computational Intelligence Magazine, IEEE*, v. 1, n. 1, p. 28–36, 2006. ISSN 1556-603X.
- COSTA, F. P. *Aplicações de Técnicas de Otimização a Problemas de Planejamento Operacional de Lavra em Minas a Céu Aberto*. Dissertação (Dissertação de Mestrado) — Departamento de Engenharia de Minas/EM/UFOP, Ouro Preto, 2005.
- COSTA, F. P.; SOUZA, M. J. F.; PINTO, L. R. Um modelo de alocação dinâmica de caminhões. *Revista Brasil Mineral*, v. 231, p. 26–31, 2004.
- COSTA, F. P.; SOUZA, M. J. F.; PINTO, L. R. Um modelo de programação matemática para alocação estática de caminhões visando ao atendimento de metas de produção e qualidade. *Revista da Escola de Minas*, v. 58, p. 77–81, 2005.
- DEB, K. *Multiobjective Optimization Using Evolutionary Algorithms*. Chichester, U.K.: John Wiley & Sons, 2001.
- DEB, K. et al. A fast and elitist multiobjective genetic algorithm: NSGA-II. *Evolutionary Computation, IEEE Transactions on*, v. 6, n. 2, p. 182–197, 2002. ISSN 1089-778X.
- DIAS, A.; VASCONCELOS, J. A. Multiobjective genetic algorithms applied to solvoptimization problems. *IEEE Transactions on Magnetcs*, v. 38, p. 1133–1136, 2002.
- FEO, T. A.; RESENDE, M. G. C. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, v. 6, p. 109–133, 1995.
- FONSECA, C.; PAQUETE, L.; LOPEZ-IBANEZ, M. An improved dimension-sweep algorithm for the hypervolume indicator. In: *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*. [S.l.: s.n.], 2006. p. 1157 – 1163.
- FONSECA, C. M.; FLEMING, P. J. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In: *Proceedings of the Fifth International Conference*. San Mateo, California: Morgan Kauffman Publishers, 1993.
- GERSHON, M. A linear programming approach to mine scheduling optimization. In: *Proceedings of the 17th Application of computers and operations research in the mineral industry*. New York: [s.n.], 1982. p. 483–493.
- GOLDBERG, D. E. *Genetic Algorithms in Search, Optimization and Machine Learning*. 1. ed. Berkeley: Addison-Wesley Professional, 1989. Hardcover.

- GUIMARÃES, I. F.; PANTUZA, G.; SOUZA, M. J. F. Modelo de simulação computacional para validação dos resultados de alocação dinâmica de caminhões com atendimento de metas de qualidade e de produção em minas a céu aberto. In: *Anais do XIV Simpósio de Engenharia de Produção (SIMPEP)*. Bauru, CD-ROM: [s.n.], 2007. p. 11.
- HANSEN, P.; MLADENOVIC, N. Variable neighborhood search: Principles and applications. *European Journal of Operational Research*, v. 130, p. 449–467, 2001.
- HANSEN, P.; MLADENOVIC, N.; PÉREZ, J. A. M. Variable neighborhood search: methods and applications. *4OR: Quarterly journal of the Belgian, French and Italian operations research societies*, v. 6, p. 319–360, 2008.
- HORN, J.; NAFPLIOTIS, N.; GOLDBERG, D. A niched pareto genetic algorithm for multiobjective optimization. In: *Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on*. [S.l.: s.n.], 1994. p. 82–87.
- KNOWLES, J.; CORNE, D. The pareto archived evolution strategy: a new baseline algorithm for pareto multiobjective optimisation. In: *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*. [S.l.: s.n.], 1999. v. 1, p. 98–105.
- LOURENÇO, H. R.; MARTIN, O. C.; STÜTZLE, T. Iterated local search. In: GLOVER, F.; KOCHENBERGER, G. (Ed.). *Handbook of Metaheuristics*. Boston: Kluwer Academic Publishers, 2003.
- LUST, T.; TEGHEM, J. Two-phase pareto local search for the biobjective traveling salesman problem. *Journal of Heuristics*, Springer Netherlands, v. 16, p. 475–510, 2010. ISSN 1381-1231.
- LUST, T.; TEGHEM, J.; TUYTTENS, D. Very large-scale neighborhood search for solving multiobjective combinatorial optimization problems. In: TAKAHASHI, R. et al. (Ed.). *Evolutionary Multi-Criterion Optimization*. [S.l.]: Springer Berlin / Heidelberg, 2011, (Lecture Notes in Computer Science, v. 6576). p. 254–268.
- MARAN, J.; TOPUZ, E. Simulation of truck haulage systems in surface mines. *International Journal of Surface Mining*, v. 2, p. 43–49, 1988.
- MERSCHMANN, L. H. C. *Desenvolvimento de um sistema de otimização e simulação para análise de cenários de produção em minas a céu aberto*. Dissertação (Dissertação de Mestrado) — Programa de Engenharia de Produção/COPPE/UFRJ, Rio de Janeiro, 2002.
- MICHALEWICZ, Z. *Genetic Algorithms+Data Structures=Evolution Programs*. [S.l.]: Springer- Verlag, 1984.
- MLADENOVIC, N.; HANSEN, P. A variable neighborhood search. *Computers and Operations Research*, v. 24, p. 1097–1100, 1997.
- PANTUZA, G. *Métodos de Otimização Multiobjetivo e de Simulação aplicados ao Problema de Planejamento Operacional de Lavra em Minas a Céu Aberto*. Dissertação (Dissertação de Mestrado) — Programa de Pós-Graduação em Engenharia Mineral - PPGEM da Universidade Federal de Ouro Preto, Ouro Preto, 2011.

- PENNA, P.; SUBRAMANIAN, A.; OCHI, L. An Iterated Local Search heuristic for the Heterogeneous Fleet Vehicle Routing Problem. *Journal of Heuristics*, Springer Netherlands, p. 1–32, 2011. ISSN 1381-1231. Disponível em: <<http://dx.doi.org/10.1007/s10732-011-9186-y>>.
- RESENDE, M. G. C.; RIBEIRO, C. C. Greedy randomized adaptive search procedures. In: GLOVER, F.; KOCHENBERGER, G. (Ed.). *Handbook of Metaheuristics*. Boston: Kluwer Academic Publishers, 2003. p. 219–242.
- RESENDE, M. G. C.; RIBEIRO, C. C. Greedy randomized adaptive search procedures: Advances, hybridizations, and applications. In: GENDREAU, M.; POTVIN, J. (Ed.). *Handbook of Metaheuristics*. 2. ed. New York: Springer, 2010. p. 283–319.
- RODRIGUES, L. F. *Análise comparativa de metodologias utilizadas no despacho de caminhões em minas a céu aberto*. Dissertação (Mestrado) — Programa de Pós-Graduação em Engenharia de Produção, Escola de Engenharia, UFMG, Belo Horizonte, 2006.
- SCHAFFER, J. D. Multiple objective optimization with vector evaluated genetic algorithms. In: *Proceedings of the 1st International Conference on Genetic Algorithms*. Hillsdale, NJ, USA: L. Erlbaum Associates Inc., 1985. p. 93–100. Disponível em: <<http://dl.acm.org/citation.cfm?id=645511.657079>>.
- SCHOTT, J. R. *Fault tolerant design using single and multicriteria genetic algorithm optimization*. Dissertação (Dissertação de Mestrado) — Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1995.
- SOUZA, M. J. F. et al. A hybrid heuristic algorithm for the open-pit-mining operational planning problem. *European Journal of Operational Research, EJOR*, v. 207, p. 1041–1051, 2010.
- SRINIVAS, N.; DEB, K. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, v. 2, p. 221–248, 1994.
- VASCONCELOS, J. A. *Optimization de Forme des Structures Électromagnétiques*. Dissertação (Tese de Doutorado) — Ecole Centrale de Lyon, França, 1994.
- WHITE, J. W.; ARNOLD, M. J.; CLEVINGER, J. G. Automated open-pit truck dispatching at Tyrone. *Engineering and Mining Journal*, v. 183, n. 6, p. 76–84, 1982.
- WHITE, J. W.; OLSON, J. P. Computer-based dispatching in mines with concurrent operating objectives. *Mining Engineering*, v. 38, n. 11, p. 1045–1054, 1986.
- ZITZLER, E.; DEB, K.; THIELE, L. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evol. Comput.*, MIT Press, Cambridge, MA, USA, v. 8, p. 173–195, June 2000. ISSN 1063-6560.
- ZITZLER, E.; LAUMANN, M.; THIELE, L. *SPEA2: Improving the Strength Pareto Evolutionary Algorithm*. Zurich, Switzerland, 2001.
- ZITZLER, E.; THIELE, L. Multiobjective optimization using evolutionary algorithms - a comparative case study. In: EIBEN, A. et al. (Ed.). *Parallel Problem Solving from Nature - PPSN V*. Springer Berlin / Heidelberg, 1998, (Lecture Notes in Computer Science, v. 1498). p. 292–301. ISBN 978-3-540-65078-2. Disponível em: <<http://dx.doi.org/10.1007/BFb0056872>>.

Anexos

Como produtos do desenvolvimento deste trabalho foram geradas as seguintes produções no ano de 2012:

COELHO, V. N. ; SOUZA, M.J.F. ; COELHO, I. M. ; GUIMARAES, F. G. ; LUST, T; CRUZ, R. C. Multi-objective approaches for the open-pit mining operational planning problem. *Electronic Notes in Discrete Mathematics*, v. 39, p.233-240, 2012b.

COELHO, V. N. ; SOUZA, M.J.F. ; COELHO, I. M. ; GUIMARAES, F. G. ; LUST, T. Algoritmos Multiobjetivos para o Problema de Planejamento Operacional de Lavra. In: XV SPOLM, 2012, Rio de Janeiro/RJ. XV Simpósio de Pesquisa Operacional e Logística da Marinha, 2012. v. 1. 12p.

Já como produtos do desenvolvimento do trabalho anterior a este, relativos ao período 2011-2012, foram geradas as seguintes produções no ano de 2012:

COELHO, V. N.; SOUZA, M. J. F.; COELHO, I. M.; CRUZ, R. C.; GUIMARAES, F. G. MULTI-OBJECTIVE APPROACHES FOR THE OPEN-PIT MINING OPERATIONAL PLANNING PROBLEM. In: META'12, 2012, Port El-Kantaoui, Tunísia. International Conference on Metaheuristics and Natural Inspired Computing, 2012. v. 1. 2p.

COELHO, V. N.; SOUZA, M. J. F.; COELHO, I. M.; GUIMARAES, F. G.; CRUZ, R. C. UMA ABORDAGEM MULTIOBJETIVO PARA O PROBLEMA DE PLANEJAMENTO OPERACIONAL DE LAVRA. In: XVI CLAIO / XLIV SBPO, 2012, Rio de Janeiro/RJ. XVI Simpósio Brasileiro de Pesquisa Operacional, 2012. v. 1. 12p.