

UNIVERSIDADE FEDERAL DE UBERLÂNDIA - UFU
FACULDADE DE ENGENHARIA ELÉTRICA - FEELT
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Maicon Fernando Dias Torres

**Rede de Sistemas Embarcados que
utilizam Algoritmos Genéticos aplicado
na otimização de Despacho em
Mineração com Múltiplas Rotas**

Uberlândia
2017

Maicon Fernando Dias Torres

**Rede de Sistemas Embarcados que
utilizam Algoritmos Genéticos aplicado
na otimização de Despacho em
Mineração com Múltiplas Rotas**

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Faculdade de Engenharia Elétrica como parte dos requisitos para a obtenção do título de Mestre em Ciências.

Área de concentração: Sistemas Embarcados

Orientador: Fábio Vincenzi Romualdo da Silva

Uberlândia

2017

Trata-se da versão corrigida da dissertação. A versão original se encontra disponível na EESC/UFU que aloja o Programa de Pós-Graduação de Engenharia Elétrica.

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema de Bibliotecas da UFU, MG, Brasil.

T693r
2017

Torres, Maicon Fernando Dias, 1989-
Rede de sistemas embarcados que utilizam algoritmos genéticos
aplicado na otimização de despacho em mineração com múltiplas rotas /
Maicon Fernando Dias Torres. - 2017.

73 f. : il.

Orientador: Fábio Vincenzi Romualdo da Silva.
Dissertação (mestrado) - Universidade Federal de Uberlândia,
Programa de Pós-Graduação em Engenharia Elétrica.
Disponível em: <http://dx.doi.org/10.14393/ufu.di.2018.1137>
Inclui bibliografia.

1. Engenharia elétrica - Teses. 2. Algoritmos genéticos - Teses. 3.
Mineração a céu aberto - Teses. 4. Otimização estrutural - Teses. I. Silva,
Fábio Vincenzi Romualdo da, 1974- II. Universidade Federal de
Uberlândia. Programa de Pós-Graduação em Engenharia Elétrica. III.
Título.

CDU: 621.3



SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA ELÉTRICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA



Ata da defesa de DISSERTAÇÃO DE MESTRADO junto ao Programa de Pós-graduação em Engenharia Elétrica da Faculdade de Engenharia Elétrica da Universidade Federal de Uberlândia.

Defesa de Dissertação de Mestrado Acadêmico, número ____/2017/PPGEE

Data: 08 de dezembro de 2017

Discente: MAICON FERNANDO DIAS TORRES

Número de matrícula: 11522EEL010

Título do Trabalho: Rede de sistemas embarcados que utilizam algoritmos genéticos aplicado na otimização de despacho em mineração com múltiplas rotas

Área de concentração: Sistemas de Energia Elétrica

Linha de pesquisa: Controle e Automação

As 10:00 horas do dia 08 de dezembro de 2017 na Sala Azul, Campus Santa Mônica da Universidade Federal de Uberlândia, reuniu-se a Banca Examinadora, designada pelo Colegiado do Programa de Pós-graduação em Engenharia Elétrica, assim composta:

Fábio Vincenzi Romualdo da Silva (orientador)

CPF: 191.533.808-51

Kleber Lopes Fontoura

CPF: 542.973.541-87

Márcio José da Cunha

CPF: 298.522.678-33

Iniciando os trabalhos o presidente da mesa Prof. Dr. Fábio Vincenzi Romualdo da Silva apresentou a Comissão Examinadora e o candidato, agradeceu a presença do público, e concedeu ao Discente a palavra para a exposição do seu trabalho. A duração da apresentação do Discente e o tempo de arguição e resposta foram conforme as normas do Programa.

A seguir o senhor presidente concedeu a palavra, pela ordem sucessivamente, aos examinadores, que passaram a argüir o candidato. Ultimada a argüição, que se desenvolveu dentro dos termos regimentais, a Banca, em sessão secreta, atribuiu os conceitos finais.

Em face do resultado obtido, a Banca Examinadora considerou o candidato A provado.

Esta defesa de Dissertação de Mestrado Acadêmico é parte dos requisitos necessários à obtenção do título de Mestre. O competente diploma será expedido após cumprimento dos demais requisitos, conforme as normas do Programa, a legislação pertinente e a regulamentação interna da UFU.

Nada mais havendo a tratar, foram encerrados os trabalhos às 14 horas e 18 minutos. Foi lavrada a presente ata que após lida e achada conforme foi assinada pela Banca Examinadora.

Prof. Dr. Fábio Vincenzi Romualdo da Silva

UFU

Prof. Dr. Márcio José da Cunha

UFU

Prof. Dr. Kleber Lopes Fontoura

CEFET

Este trabalho é dedicado aos meus pais, Sônia e Marcos.

Agradecimentos

O autor agradece à UFU (Universidade Federal de Uberlândia) e a empresa Instale Tecnologia pela disponibilidade de pesquisas, desenvolvimento e testes do trabalho desenvolvido.

*A tomada de uma decisão
anula todas as outras possibilidades.
(o autor)*

Resumo

Torres, M. F. D. **Rede de Sistemas Embarcados que utilizam Algoritmos Genéticos aplicado na otimização de Despacho em Mineração com Múltiplas Rotas.** 75 p. Dissertação de mestrado – Faculdade de Engenharia de Uberlândia, Universidade de Uberlândia, 2017.

A área de mineração é muito rica em processos complexos que necessitam de tecnologias. Os recursos minerais limitados e o difícil acesso elevam o capital investido na extração e no processamento de minérios. Para viabilizar o processo essas empresas buscam sempre otimização com foco na perda zero. Os algoritmos genéticos são muito utilizados em processos de otimização, incluindo o planejamento e a otimização de rotas. Esse trabalho apresenta a aplicação de algoritmo genético no despacho de caminhões em mina a céu aberto, de modo a orientar o operador para que o mesmo siga a melhor rota, visando à redução de perdas e o aumento da produção. Como o operador recebe apenas uma sugestão de rota, uma comparação entre as perdas provocadas pelas rotas sugeridas pelo sistema e as seguidas pelo operador demonstrará a eficiência do sistema, contanto que este fosse seguido, justificando a aplicação do algoritmo genético nesse processo.

Palavras-chave: Algoritmo genético, sistema embarcado despacho em mineração.

Abstract

Torres, M. F. D. **Network of Embedded System using Genetic Algorithm to optimizations in Mineration with multiples routes.** 75 p. Master Thesis – Uberlândia School of Engineering, University of Uberlândia, 2017.

The mining area is very rich in complex processes that require technologies. Limited mineral resources and difficult access increase the capital invested in the extraction and processing of ores. To make the process viable, these companies always seek optimization with a focus on zero loss. Genetic algorithms are widely used in optimization processes, including route planning and optimization. This work presents the application of genetic algorithm in the dispatch of trucks in open pit, in order to guide the operator to follow the best route, aiming at reducing losses and increasing production. As the operator receives only a route suggestion, a comparison between the losses generated by the routes generated by the system and those followed by the operator will demonstrate the efficiency of the system if it were followed, justifying the application of the genetic algorithm in that process.

Keywords: Genetic algorithm, embedded system, mining dispach.

Lista de ilustrações

Figura 1	Exemplo de problema - Lançamento vertical	22
Figura 2	Discretização do espaço de busca	23
Figura 3	a) Cruzamento um ponto; b) Cruzamento dois pontos; c) Cruzamento de pontos aleatórios	26
Figura 4	a) Mutação aleatória e b) Mutação por troca	27
Figura 5	Distância entre dois pontos na superfície da Terra.	33
Figura 6	Cercas eletrônicas em um mapa (verde) com os nomes locais (amarelo).	34
Figura 7	Módulo ESP8266.	36
Figura 8	Interface Homem-Máquina (IHM) utilizando dispositivos <i>mobile</i>	37
Figura 9	Composição do mercado de OS <i>mobile</i> (2016).	38
Figura 10	Grafo de uma mina genérica.	43
Figura 11	Matriz de rotas para o exemplo da Figura 10.	44
Figura 12	Cruzamento de dois indivíduos.	45
Figura 13	Erro de <i>loop</i> ao gerar indivíduo	46
Figura 14	Correção dos laços	46
Figura 15	Erro de caminho sem saída	46
Figura 16	Correção do erro de caminho sem saída.	47
Figura 17	Exemplo de mutação	48
Figura 18	Diagrama do Sistema	49
Figura 19	Diagrama da IHM	50
Figura 20	Beagle Bone Rev.C, computador embarcado.	51
Figura 21	Diagrama da Telemetria	51
Figura 22	Grafo da mina	54
Figura 23	Software de cadastro	55
Figura 24	Software de cadastro - Frentes	55
Figura 25	Software de cadastro - pontos de báscula	56
Figura 26	Software de cadastro	57

Figura 27	Software de cadastro	57
Figura 28	Software de cadastro	58
Figura 29	Tela inicial	59
Figura 30	Senha de proteção	60
Figura 31	Notificação de alteração de rota	61
Figura 32	Tela de notificação em modo noturno	62
Figura 33	Fluxograma da rotina inicial	63
Figura 34	Fluxograma da rotina cerca eletrônica	64
Figura 35	Fluxograma da rotina AG	65
Figura 36	Simulador desenvolvido para a validação	66
Figura 37	Gráfico de Produção Total x N° de caminhões	70

Lista de tabelas

Tabela 1	Exemplo de população inicial hipotética	24
Tabela 2	Exemplo de população final	29
Tabela 3	A atraso em fila com objetivos separados	68
Tabela 4	Produção por caminhão com objetivos separados	68
Tabela 5	Tempo perdido em fila	69
Tabela 6	Produção dos caminhões: Alocação fixa vs dinâmica (multiobjetivo) .	69
Tabela 7	Tempo de execução do AG	71
Tabela 8	Custo dos materiais utilizados	71

Lista de siglas

AG Algoritmo Genético

CAN *Controller Area Network*

DMT Distância Média de Transporte

DMS *Database Management Systems* - Sistema de Gerenciamento de Banco de Dados

GPS Sistema de Posicionamento Global

GUI *Graphical User Interface* - Interface Gráfica do Usuário

HMI *Human-Machine Interface* - Interface Homem-Máquina

IHM Interface Homem-Máquina

IEC 60529 *International Electrotechnical Commission* - Comissão Internacional de Eletrônica e Mecânica

JSON *JavaScript Object Notation*

K Escala de temperatura Kelvin

LCD *Liquid-Crystal Display* - Display de cristal líquido

PO Pesquisa Operacional

TCP *Transmission Control Protocol* - Protocolo de controle de transmissão

TTL *Transistor-Transistor Logic* - Lógica de Transistor-Transistor

UDP *User Datagram Protocol* - Protocolo Dados de Usuário

WAP *Wireless Access Point*

VRP *Vehicle Route Problem* - Problema de Rotas de Veículos

Sumário

1	Introdução	15
2	Revisão Bibliográfica	19
2.1	AG	21
2.1.1	Função objetivo	22
2.1.2	Indivíduos	22
2.1.3	Seleção	24
2.1.4	Cruzamento	25
2.1.5	Mutação	26
2.1.6	Critério de parada	27
2.2	Sistemas Embarcados como plataforma para execução de AG	28
2.2.1	História	29
2.2.2	Gerenciamento de Recursos	31
2.2.3	Periféricos	32
2.2.4	Sistema de Posicionamento Global (GPS)	32
2.3	Comunicação	34
2.3.1	ESP8266	35
2.4	Interface do usuário (Display)	37
2.4.1	Android e Firemonkey	37
3	Problema de Despacho utilizando Algoritmo Genético	40
3.1	Modelo de busca de rotas	40
3.2	Função de aptidão	41
3.3	Criação de indivíduo	43
3.4	Seleção	44
3.5	Cruzamento	45
3.6	Elitismo	47
3.7	Mutação	47

3.8	Critério de parada	48
4	Sistema Desenvolvido	49
4.1	Servidor	49
4.2	IHM	50
4.3	Telemetria	50
4.3.1	Módulo WIFI	52
4.4	Softwares de Cadastro	53
4.4.1	Aplicativo Android	58
4.5	Simulação	59
5	Resultados	67
5.1	Alocação Fixa e Alocação Dinâmica	67
5.1.1	Simulação Específica - Redução de distância, ociosidade e fila . . .	67
5.1.2	Simulação Completa - Alocação Fixa e Dinâmica	69
5.2	Desempenho: Analise do Algoritmo Embarcado	70
6	Discussão e Conclusões	72
	Referências	73

CAPÍTULO 1

Introdução

Ao longo dos últimos anos, empresas de tecnologia passaram a investir em pesquisa e desenvolvimento visando atender seu público exigente. O avanço da tecnologia trouxe recursos que não existiam a alguns anos atrás, como a internet de alta velocidade, tablets, smartphones e computadores embarcados. Dispositivos do tamanho de cartões de crédito possuem poder de processamento superior aos computadores do século XX. O uso desses recursos tem crescido a medida que a tecnologia contida neles avança e seus custos reduzem. A disseminação desses dispositivos trouxe a possibilidade de implementação de soluções alternativas às já existentes no mercado, reduzindo o custo de desenvolvimento, tornando métodos de otimização viáveis.

As tarefas de otimizações são as mais abordadas em trabalhos de diversas áreas atualmente, isso se deve ao fato dos recursos minerais, energéticos e naturais estarem cada vez mais escassos e limitados. Empresas, sob efeito da crise econômica, passam a investir em melhorias para aumentarem sua margem de lucro, por meio da redução de suas perdas, e aumentar sua competitividade no mercado. Nesse sentido, o estudo de otimização de equipamentos está se tornando muito relevante atualmente.

No ramo da mineração, por exemplo, existem poucas empresas no mundo desenvolvendo soluções que gerenciam rotas nas minas, bem como fornecem soluções para telemetria de equipamentos em geral. Considerando que esse ramo de negócio movimenta bilhões de dólares todo ano, investir na otimização de mineração pode resultar na economia de bilhões de dólares todo ano.

Segundo os dados da Secretaria de Geologia, Mineração e Transformação Mineral do ano de 2015, o comércio exterior do setor mineral brasileiro alcançou US\$ 39 bilhões, isso representa 20,4% do total de exportações do Brasil nesse mesmo ano. Em 2016 o total de exportação fechou em US\$ 36,6 bilhões em razão do ciclo da desvalorização dos preços dos minérios mais vendidos, como o minério de ferro. A previsão do Instituto Brasileiro de Mineração (IBRAM) é um aumento de até 5% em 2017. A participação da indústria mineradora no PIB do Brasil é relevante, logo a retração do mercado de minérios força as mineradoras a direcionar seus investimentos para redução dos seus custos e otimização

de seus processos.

Normalmente na natureza, o minério bruto encontra-se abaixo do solo recoberto de outros tipos de minérios denominados estéril. Esse material é movido da cava da mina para depósitos, onde podem ser reaproveitados posteriormente ou simplesmente acumulados, pois grande parte desses materiais não são viáveis de aproveitamento, ou possuem baixo retorno financeiro como a magnetita, gesso, sílica, e a cal. A quantidade de estéril até o minério e sua dificuldade de extração, define o tipo de mina. Quando o minério bruto principal está muito abaixo da superfície, é mais viável cavar galerias de túneis para acessá-lo. Esse tipo de mina denomina-se mina subterrânea e apresenta muitos riscos de acidentes, portanto, devem ser tomados diversos tipos de cuidados especiais, envolvendo a segurança de pessoas e equipamentos. Por outro lado, em minas a céu aberto, o minério está próximo a superfície, tornando viável a remoção da camada de estéreo que o recobre, assim reduz o risco de acidentes, como desmoronamentos, e cuidados especiais exigidos em minas subterrânea, como ventilação forçada por exemplo.

A mineração a céu aberto é o método mais comum no mundo. As minas a céu aberto ocupam 60% de toda a superfície de trabalho (HARTMAN; MUTMANSKY, 2002). A logística da operação neste tipo de mina consiste organizar equipamentos de carga e equipamento de transporte, as frentes de minério (também conhecido como pontos de extração) são locais onde máquinas, como escavadeiras, operam de forma coordenada para extrair o minério e transporta-lo até os equipamentos de transporte (caminhões). Esses veículos transportam o material, minério ou estéreo, da frente de minério até o ponto de basculamento. O ponto de basculamento é onde os caminhões depositam o material transportado para uso posterior como no caso de pilhas de minério ou para processá-lo imediatamente como é o caso dos britadores. Esse equipamento é capaz de triturar o minério para facilitar o beneficiamento pela industria. Cada equipamento tem suas restrições de capacidades de processamento, que deve ser respeitadas para que todas as máquinas trabalhem em sincronismo.

Para manter a produtividade é necessário dimensionar a quantidade de máquinas de carga e caminhões suficiente para atender todas as frentes sem formar filas, considerando o tempo de carregamento, transporte e basculamento dos equipamentos.

O tempo de carregamento é definido como o tempo que uma máquina de carregamento gasta para carregar um caminhão, esse valor depende do tamanho da concha da máquina e capacidade do caminhão, ou seja a quantidade de material que o caminhão é recomendado a transportar por viagem. O tempo de transporte é o tempo que o caminhão gasta para transportar o minério da frente de extração até o ponto de basculamento, esse tempo depende da velocidade do caminhão e da distância do trajeto.

O Tempo de basculamento é o tempo que o caminhão gasta para manobrar e bascular o minério. O tempo de carregamento, transporte e basculamento compõem o ciclo de transporte. Pode acontecer de dois ou mais caminhões se encontrar nas frentes de minério,

nesse caso o primeiro carrega enquanto os demais devem esperar, esse tempo ocioso é denominado tempo de fila, e deve ser minimizado para manter a produção máxima da mina.

Os engenheiros de minas alocam as máquinas segundo dados históricos da mina com foco na redução do tempo de fila, mas como não é possível prever fatores externos como quebra de equipamentos, acidentes, condições climáticas, ao longo do tempo, o tempo de fila aumenta e a alocação inicial aumentando as perdas na produção de minério. Para corrigir esse problema um método denominado “Sistema de Despacho” é utilizado pelas mineradoras.

No “Sistema de Despacho”, um grupo de funcionários, denominados controladores, trabalham durante turnos de 8 ou 6 horas, e são treinados para alterar rotas, direcionando o fluxo de caminhões afim de evitar a formação de filas e perca de produtividades, usando um misto de raciocínio e intuição. Os controladores, muitas vezes, tomam decisões de modo intuitivo, o que nem sempre são ótimas (MARIN, 2009).

Os controladores precisam definir as rotas levando em conta as condições instantâneas dos diversos pontos de carregamento, a situação de cada caminhão e os pontos de descarga. A posição e a velocidade de um determinado caminhão são obtidas por meio de rádio ou, em alguns casos, através de computadores nos caminhões que transferem as informações para um servidor, onde elas podem ser utilizadas pelos controladores(WHITE; OLSON, 1986).

O sistema de despacho é considerado semi-automático quando o computador não comunica diretamente com motorista do caminhão (BONATES; LIZOTTE, 1988), nesse caso o controlador aloca manualmente os equipamentos nas rotas de acordo com a sua visão da operação da mina, geralmente obtida por um acompanhamento em tempo real da posição e do estado da atividade dos equipamentos, como por exemplo: carregando, manobrando, basculando, parado etc, e informa os motoristas por meio de rádio.

A falta de orientação entre os pontos de carga e descarga, e a atuação lenta dos operadores provocam filas de caminhões tanto na origem quanto no destino, esse tempo que os caminhões e máquinas escavadeiras/carregadeiras ficam parados causam prejuízo para a empresa mineradora devido a ociosidade. Essas condições variáveis na mina necessitam de tomada de decisões em tempo real (COX et al., 2017). Uma solução utilizada nesse caso é implantação de um sistema de alocação dinâmica na mineradora.

Na alocação dinâmica o algoritmo parte de uma alocação prévia, dimensionado por um engenheiro, posteriormente durante a operação, sempre que houver alterações nas condições iniciais, o próprio algoritmo modifica as rotas automaticamente, atuando instantaneamente. Assim, um caminhão pode ser direcionado a diferentes frentes, aumentando a produtividade, reduzindo a quantidade de equipamentos necessário para manter um certo nível de produção(SOUZA et al., 2010). Além disso reduz o tempo ocioso das máquinas e dos caminhões, ou em outras palavras, o algoritmo executa as tarefas dos controladores de

modo automático, de modo que os controladores apenas monitorem o processo reduzindo a responsabilidade no processo de despacho.

A alocação dinâmica pode ser encontrada em muitos trabalhos acadêmicos, mas existem muitos métodos utilizados para alcançar o resultado. Grande parte dos trabalhos concentram as informações de todos os caminhões da mina em um computador dedicado, a partir desse, os dados são processados por algorítimos matemáticos com base em programação linear ou por algorítimos estocásticos como Algoritmo Genético (AG). Essa centralização dos dados faz com que o custo de processamento seja muito alto, exigindo um hardware mais caro, a medida que o número de caminhões e da complexidade da mina são aumentados.

A técnica utilizada nesse trabalho visa desenvolver um método alternativo aos existentes, aplicando o método de busca por AG de modo descentralizado, diminuído o custo do elemento mais caro em processos de alocação dinâmica, o servidor. Esse trabalho propõe a implementação de um algoritmo genético de modo que possa ser embarcado e promova o aumento da produção, reduzindo as perdas relacionadas ao tráfego de máquinas no interior de uma mineradora, que resultam em problemas como a formação de filas, gasto desnecessário de combustível e seleção incorreta de prioridades.

Os dados e informações obtidas de uma mina real são utilizadas no software de simulação desenvolvido com o propósito de avaliar a aplicabilidade, eficiência e usabilidade da proposta para a futura implantação na mina. Esse procedimento de pré-teste é necessário porque a operação da mina é ininterrupta e autorização para melhorias somente são obtidas quando avaliadas previamente.

CAPÍTULO 2

Revisão Bibliográfica

Em um contexto geral, o método de otimização para buscar rotas entre origem e destino pode ser encontrado em uma ampla quantidade de trabalhos acadêmicos, assim como sua implantação, principalmente em problemas de logística, que exigem restrições de tempo e custos de transporte.

O Problema de Roteamento de Veículos, conhecido na literatura como *Vehicle Route Problem* - Problema de Rotas de Veículos (VRP) tem sido um alvo de estudos nos últimos anos, principalmente no ramo da mineração, onde temos grandes custos envolvidos no processo. A complexidade do problema somada à necessidade de redução de custos que apresentam essas atividades de carga e transporte de material, tem estimulado diversos profissionais e pesquisadores a realizarem estudos teóricos sobre o tema. O VRP pode ser considerado um problema que exige elevado custo computacional, isto é, muito processamento. Por isso, a busca por algoritmos que exigem pouco processamento para a solução desse tipo de problema, vem sendo intensificada nos dias atuais.(ALVARENGA, 1997).

Hjorring e Holt (1999) afirma que a solução de roteamento de veículos se situa na classe dos problemas difíceis de serem resolvidos, ou seja, é um problema np-complexo. Por isso, exige um custo computacional elevado.

A medida que os pontos de destino, origens e intercessões aumentam, o tempo de execução do Problema de Rotas de Veículos cresce exponencialmente. Por exemplo, se considerarmos n como sendo o número de nós ou entradas do algoritmo a quantidade de passos gastos até encontrar todas as rotas possíveis é fornecida pela fórmula $O(n!)$.

Dois dos algoritmos mais conhecidos para a busca de rotas são o Dijkstra $O(n^2)$ e o de Bellman-Ford $O(n^2)$. O algoritmo de Dijkstra e o algoritmo Bellman-Ford são muito utilizados em problemas de VRP porque fornecem resposta rápida em comparação a algoritmos que utilizam Pesquisa Operacional (PO). No entanto, esses algoritmos são utilizados para resolver VRP que possuem poucas origens e destinos e poucos intercessões, desse modo, apresentando poucas combinações.

Das linhas de pesquisas que estudam soluções ótimas para VRP, existe uma área que

foca no uso de AG para esse fim.

Algoritmos genéticos (GOLDBERG, 1989) são utilizados por pesquisadores para resolver problemas relacionados a busca de otimização de rotas (AHN; RAMAKRISHNA, 2002). Um dos motivos do uso de AG em VRP é que eles são robustos, eficientes e podem trabalhar com uma ampla faixa de funções objetivo(WARDLAW; BHAKTIKUL, 2004). Alguns trabalhos (KANOH; NAKAMURA, 2000) (TANAKA et al., 2004) (INAGAKI; HASEYAMA; KITAJIMA, 1999) utilizam algoritmos genéticos para encontrar múltiplas rotas de modo simultâneo. Outros propõem solução para a sobreposição de rotas (CHAKRABORTY, 2004) (INOUE, 2001), pois muitas das rotas encontradas são semelhantes. Heinen e Osório (2006) propõe a Heurística de Clark e Wright, a Heurística de Mole e Jameson para resolver problemas de VRP. He et al. (2010) AG para despacho de caminhões em minas, obtendo um aumento na eficiência econômica e redução da manutenção.

O trabalho de Cox et al. (2017) mostra que o AG obteve performance quando a mina tem poucos caminhões. Alvarenga (1997) utilizou AG para alocação fixa de caminhões. Para reduzir o tempo de execução o autor utiliza processamento paralelo. No entanto, para minas com muitas origens, destinos e intercessões a utilização de AG pode não ser interessante. Isso ocorre porque os AG's executam muitas operações simples até encontrar uma solução aceitável, sua aproximação depende de probabilidades, isto é, o AG é estocástico. A solução que o AG fornece depende de um número indeterminado de passos, seu tempo de processamento, para um determinado hardware, varia a cada tentativa.

A solução PO baseia-se em modelos matemáticos e executa operações matriciais, ou seja, cálculo do determinante e matriz inversa. Costa, Souza e Pinto (2004) e Melo et al. (2013), por exemplo, executam funções de PO para gerar planilhas com o auxílio das ferramenta Lingo Excel, com foco na qualidade e na produção de minério. A maior vantagem da PO é que esse método sempre encontra uma solução ótima, contanto que essa exista. (SCHULZE, 1998)

Quando o problema está relacionado a um elevado número de origens, destinos e intercessões, tanto a solução que utiliza AG, quanto a que faz uso de PO necessitam de servidores de grande porte, de elevado poder de processamento, pois os cálculos de rotas são executados de centralizado no servidor. Esse tipo de solução necessita de investimento elevado. A medida que se acrescenta mais origens, destinos e intercessões faz-se necessário a substituição ou aquisição de mais servidores de grande porte. Nesse sentido, o investimento em hardware depende do número de rotas que podem ser traçadas entre origem e destino.

Com a intenção de minimizar os investimentos em hardware, este trabalho propõe a descentralização do processamento das rotas. Na estratégia adotada um AG foi implementado e instalado em um sistema embarcado, de modo que a rota que o caminhão deve seguir é calculada pelo sistema embarcado instalado no caminhão. AG's são ade-

quadros para resolver problemas estocas e podem encontrar a solução mais rápido que os métodos matemáticos pois não avalia todas as possíveis soluções aceitáveis para um problema(KURNIAWAN; SULISTIYO; WULANDARI, 2015). O sistema recebe informação da posição de todos os caminhões da mina de modo que o algoritmo implementado leva em consideração as rotas que estão sendo percorridas por todos os veículos e a posição de cada um. Esse procedimento tem como objetivo evitar filas nos pontos de extração de minério, onde os caminhões são carregados e nos pontos de britagem, onde os caminhões entregam as cargas de minério para beneficiamento.

2.1 Algoritmo Genético

No ano de 1958 o botânico Charles Darwin apresentou o conceito da seleção natural, para provar a teoria da evolução das espécies (DARWIN, 1859). Darwin defendia a ideia de que a espécie sobrevivente não é a mais forte, e sim a que sofre o maior número de adaptações ao ambiente. Os algoritmos genéticos foram criados baseando-se no princípio da seleção natural, visando solucionar problemas de otimização por Holland (1992).

No processo de seleção natural, os genes, a cada geração, sofrem alterações que podem ser benéficas ou não. Caso um novo gene proporcione uma nova vantagem evolutiva para um determinado indivíduo, de modo que ele fique melhor adaptado ao ambiente, esse indivíduo terá maior condições de transferir seus genes para as futuras gerações. Desse modo, quando se tenta resolver um problema por meio de algoritmos genéticos, é necessário encontrar uma maneira (denominada em AG de função objetivo¹) de representar as dificuldades impostas pelo ambiente aos indivíduos para que os melhores tenham maior probabilidade de serem selecionados.

Os indivíduos de uma espécie, no contexto de AG é o portador do código genético, que por sua vez, é uma representação do espaço de busca do problema a ser resolvido, em geral na forma de sequências de bits. Cada gene contém informações que compõem a solução do problema.

No processo de evolução natural, os indivíduos mais aptos sobrevivem e cruzam dando origem a novos indivíduos com a combinação das características dos pais, ocasionalmente os filhos sofrem mutações que podem ajudar à adaptação ao ambiente. Em AG os indivíduos, também sofrem mutação e cruzamento até que seja atingida o critério de parada. Assim, após a geração final, a população contém os indivíduos mais evoluídos, desses o que melhor atender a função objetivo é selecionado como a solução do problema.

Atualmente os Algoritmos Genéticos estão entre os algoritmos mais poderosos e bem sucedidos, quando aplicados em buscas estocásticas e técnicas de otimização baseado nos princípios de evolução Li, He e Guo (2006).

¹OBS: Na implementação de um algoritmo AG, a função objetivo é importante porque a partir dela é implementada a função fitness, que determinará o erro entre a resposta calculada e o valor de resposta ideal.

2.1.1 Função objetivo

A função-objetivo é utilizada para verificar se o resultado esperado já foi alcançado com sucesso. Ela pode ser um problema de otimização, um conjunto de teste para identificar os indivíduos mais aptos, ou mesmo uma “caixa preta”, onde sabemos apenas o formato das entradas e nos retorna um valor que queremos otimizar. A grande vantagem dos algoritmos genéticos está no fato de não precisarmos saber como funciona esta função objetivo, apenas tê-la disponível para ser aplicada aos indivíduos e comparar os resultados.

Por exemplo, imagine uma bola de tênis em sua mão e que deseja arremessá-la a uma altura de um metro ($\Delta h = 1$), conforme a Figura 1:

Supondo que a bola de tênis seja lançada com uma velocidade (v_0) e que ela seja superior à ideal, a altura da bola de tênis irá ultrapassar um metro, e se for menor, não atingirá a altura definida.

Nesse caso a função objetivo pode ser representada pela Equação 1, e ela pode ser usada para determinar se a altura exata de um metro foi atingida, em outras palavras, se a velocidade aplicada a bola de tênis foi o valor exato para se alcançar a altura exata de um metro.

No contexto de AG, a v_0 é a variável de controle, cada valor de v_0 será uma solução, mas não necessariamente uma solução boa. O valor ideal de v_0 faz a bola de tênis atingir um metro exato e cair novamente. Em problemas de AG a variável de controle recebe a denominação de indivíduo (ou cromossomo).

$$\Delta h = \frac{v_0^2}{2g} \quad (1)$$

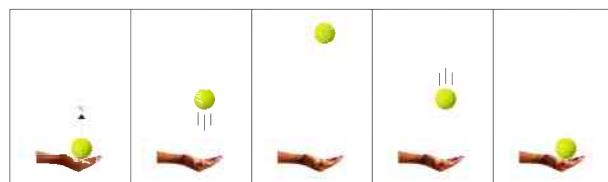


Figura 1 – Exemplo de problema - Lançamento vertical

Fonte: Acervo do Autor.

2.1.2 Indivíduos

O indivíduo pode ser representado (codificado) por vetor, inteiro ou número de ponto flutuante.

Por exemplo: A codificação em números inteiros consiste em reduzir o espaço de busca, convertendo uma faixa de valores contínuos em uma faixa de valores discretos. Em uma faixa de valores contínuos, existe uma infinidade de valores que podem ser considerados soluções para o problema. Para que o AG encontre a solução ideal, é necessário buscar

num conjunto de infinitos valores. Por isso, na prática, delimita-se um conjunto de valores reais ($indivíduo \in \mathbb{R}[min, max]$) na qual deseja-se encontrar a resposta e em seguida, o intervalo de números reais é discretizado, isto é, convertido em um número finito de elementos.

No exemplo da bola de tênis, considerando que o valor de v_0 está contido no conjunto real $\{0,100\}$. Nesse caso, o espaço de busca é infinito dentro do intervalo ($v_0 \in \mathbb{R}[0, 100]$). Mas se discretizarmos o espaço em 1024 partes, por exemplo, reduziria a probabilidade de encontrar a solução de $\frac{1}{\infty}$ para $\frac{1}{1024}$. A Figura 2 mostra o gráfico do deslocamento da bola de tênis $f(v_0)$ (eixo das ordenadas) em função da velocidade inicial v_0 (eixo das abscissas). Esse procedimento, facilita a busca pela solução mais próxima da ideal. Na prática, define-se o intervalo de amostragem de modo que o erro obtido entre o valor encontrado e o ideal seja desprezível, reduzindo, desse modo, o esforço computacional que seria necessário para encontrar o valor ideal.

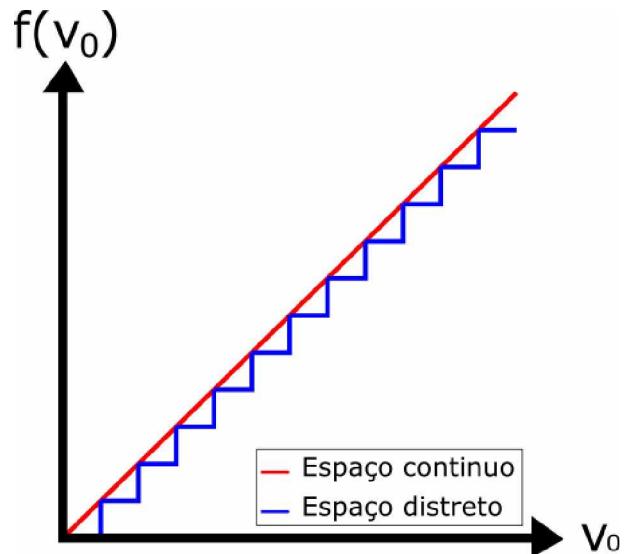


Figura 2 – Discretização do espaço de busca

Fonte: Acervo do Autor.

Nesse exemplo, os indivíduos podem ser criados aleatoriamente por meio de uma função randômica respeitando-se o intervalo entre 0 e $2^{(genes)} - 1$. O Indivíduo é codificado em um número binário que represente seu valor, cujo número de genes é igual a $\log_2(espaço)$. No exemplo da bola de tênis, definimos o espaço de busca entre 0 a 100, dividido em 1024 partes, assim são gerados valores entre 0 e 1023 para representar esses indivíduos, que posteriormente são codificados em números binários, compostos por 10 genes ($\log_2(1024) = 10$) de valores binários. Os indivíduos criados são organizados para constituir a população inicial, na Tabela 1 é mostrado de como ficaria uma população para o exemplo da bola de tênis, para uma população de 13 indivíduos aleatórios.

Tabela 1 – Exemplo de população inicial hipotética

v_0	v_0 discreto	v_0 codificado (indivíduo)
0,0000	0	0000000000
1,5625	16	0001000000
2,6367	27	0000011011
5,4688	56	0000111000
9,6680	99	0001100011
12,2070	125	0111110100
17,1875	176	0101100000
25,0000	256	0100000000
30,3710	311	0100110111
45,8007	469	0111010101
85,1562	872	1101101000
97,5586	999	1111100111
99,5117	1019	1111111011

2.1.3 Seleção

O operador seleção simula a “seleção natural” proposta por Darwn, este escolhe os indivíduos que formarão a próxima população ou que cruzarão para obter novos indivíduos, como base em sorteios com maior probabilidade de selecionar os que tenham maior aptidão, porém aqueles que têm baixa probabilidade também podem ser selecionados.

Os algoritmos genéticos possuem diversos métodos de seleção. A escolha inadequada desse método pode causar atrasos na execução ou não alcançar o ótimo global. A seleção ocorre após o cálculo da aptidão dos elementos gerados. Por meio da aptidão é possível determinar quais os indivíduos possuem maior probabilidade de gerar descendentes que aproximem da solução ótima. A combinação dos indivíduos de baixa aptidão pode gerar um descendente de alta aptidão, devido a características que os demais da elite não possuem.

A técnica da Roleta é um método de seleção muito utilizado devido à simplicidade de execução. Todos os indivíduos recebem uma porcentagem da roleta proporcional à sua aptidão, ou seja, quanto maior for a aptidão de um cromossomo maior será sua probabilidade de ser escolhido. Após definida a roleta, gera-se um número aleatório, dentro dos limites da mesma, deixando a sorte definir qual será o indivíduo selecionado. O processo se repete até que seja obtido um número par necessário para executar o processo de cruzamento.

O torneio é um método parecido com a roleta. No entanto, o sorteio é igualitário, ou seja, todos os indivíduos possuem a mesma probabilidade de serem escolhidos, e só após o sorteio é analisado a aptidão de cada indivíduo. O método consiste em selecionar aleatoriamente “n” indivíduos da população atual e avaliar a aptidão de cada um, aquele que tiver a maior aptidão é selecionado para o cruzamento.

2.1.4 Cruzamento

O cruzamento é a fase do Algoritmo genético que segue após a seleção. Assim como no cruzamento biológico, cada um dos dois progenitores sede parte de suas características para criar um novo indivíduo. Em AG dois indivíduos selecionados pelos métodos apresentados no tópico 2.1.3, trocam parte de seus genes para criar dois indivíduos novos. A população de soluções de alta performance sobrevive e reproduz, aproximando-as cada vez mais das melhores soluções para o problema (COX et al., 2017). Os métodos de cruzamento mais conhecidas são os de ponto único, duplo e cruzamento de pontos aleatórios.

A ideia do ponto único é escolher um aleatoriamente um ponto do código genético para dividi-lo em dois, dessa forma o primeiro filho é formado pela combinação da primeira parte do pai um com a segunda parte do pai dois. O segundo filho é formado pela combinação das duas partes restantes, conforme apresentado na Figura 3a.

No método de cruzamento por dois pontos são selecionados dois pontos aleatoriamente, dividindo o código genético em três partes. Assim o primeiro filho é composto pela combinação da primeira e terceira parte do pai um e a segunda parte do pai dois. O segundo filho é formado pela combinação das três partes restantes, como mostrado na Figura 3b.

O método de cruzamento de pontos aleatórios consiste em utilizar um máscara binária com tamanho igual ao número de genes do indivíduo e é preenchida aleatoriamente de zeros e uns. Em seguida, a rotina compara os genes dos indivíduos com o valor do bit da máscara. Quando a mascara tiver o bit igual a um, o descendente herda o gene do pai um, quando for zero, herda o alelo do pai dois. O segundo descendente é obtido por meio da inversão da mascara e repetindo o processo (Figura 3c).

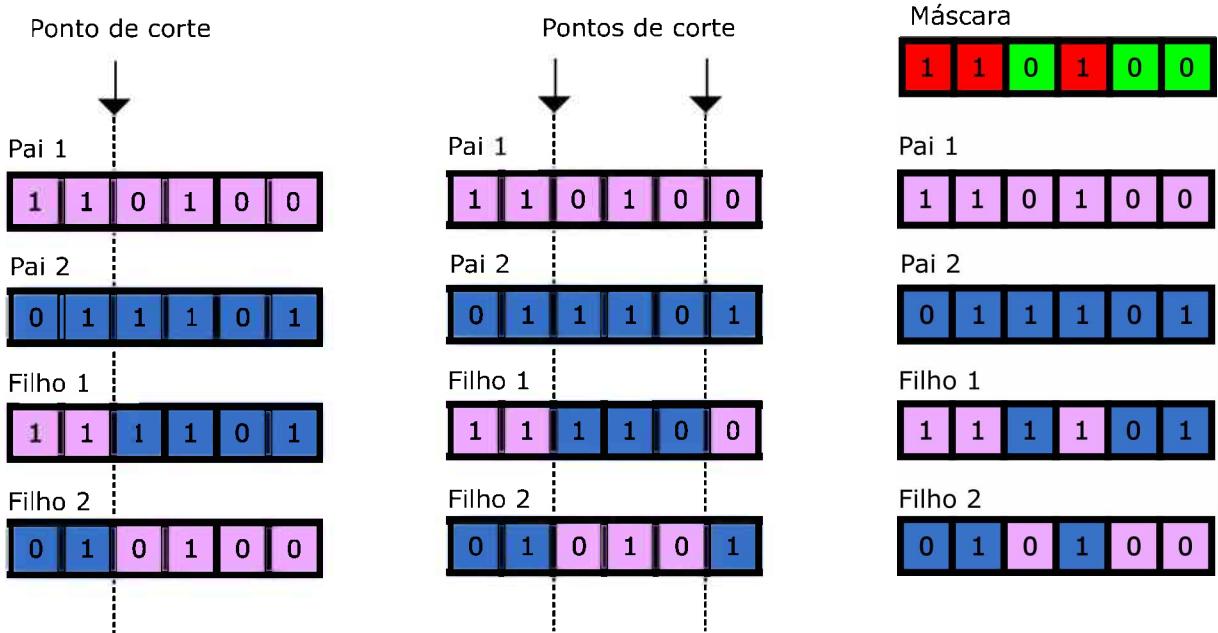


Figura 3 – a) Cruzamento um ponto; b) Cruzamento dois pontos; c) Cruzamento de pontos aleatórios

Fonte: Acervo do Autor, baseado em Mitchell (1998)

2.1.5 Mutação

O processo de mutação consiste em alterar aleatoriamente algumas características dos indivíduos da população. Isso é de vital importância para alcançar a solução ótima para o problema, pois é possível que nenhum dos pais tenha, ou tenha em baixa porcentagem, alguma propriedade especial para acelerar o processo de busca da solução ideal. O principal fator que se deve ter em mente é que a mutação deve atingir apenas a uma pequena parte da população, para evitar a perda de características benéficas ou a intensificação de características negativas.

Dentre as mutações mais utilizadas encontra-se a mutação aleatória e a mutação por trocas. Na mutação aleatória é sorteado um gene, que é alterado de acordo com os valores válidos para o gene, como mostrado na Figura 3a.

Entende-se por valores válidos aqueles que podem ser alterados de acordo com a implementação do gene, isto é, se o gene foi criado com bits, um ou mais bits podem ser substituídos por outros valores. Caso o gene tenha sido implementado com letras do alfabeto, a letra poderá ser substituída por outra.

Na mutação por troca, “n” pares de genes são sorteados aleatoriamente e permutados entre si, como mostrado na Figura 3b.

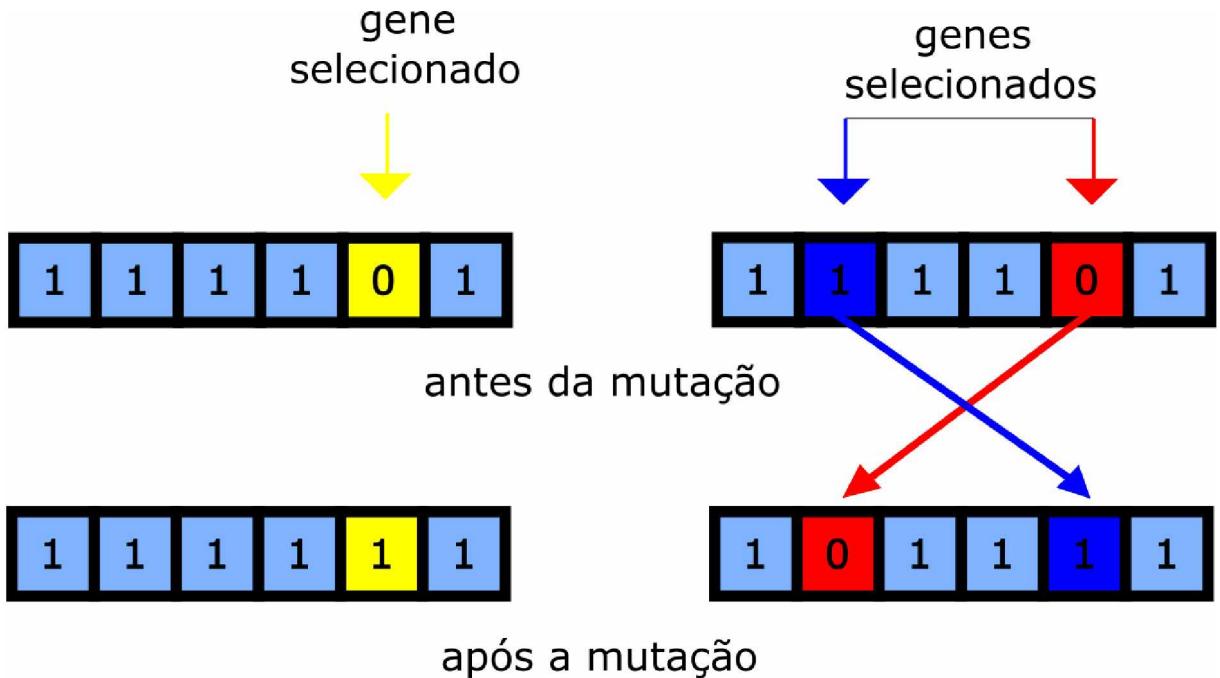


Figura 4 – a) Mutação aleatória e b) Mutação por troca

Fonte: Acervo do Autor, baseado em Mitchell (1998)

2.1.6 Critério de parada

Os AG repetem as etapas de avaliação da população², seleção, cruzamento e mutação, com o objetivo de evoluir os indivíduos e melhorar sua aptidão, aproximando cada vez mais da solução ideal. Para finalizar, o algoritmo deve estabelecer um critério de parada, que pode variar conforme o problema e as condições aceitáveis. A seguir temos alguns critérios de parada:

- **Número de gerações máximo:** O algoritmo para após a executar uma quantidade fixa de gerações;
- **Qualidade da solução:** O algoritmo para quando a aptidão do melhor indivíduo da população é igual ou maior que um erro aceitável;
- **Tempo de execução:** O algoritmo executa até que seja atingido um tempo especificado previamente;
- **Estagnação de aptidão:** O algoritmo para quando a população não consegue evoluir mais, ou seja, não ocorre progresso na aptidão dos indivíduos.

Após atingir o critério de parada, a população é ordenada de acordo com a aptidão dos indivíduos, aquele que tiver a maior aptidão será decodificado para as escala do problema,

²Avaliar a população significa aplicar a função de aptidão (*fitness*) para avaliar a qualidade de cada indivíduo.

considerado essa como a solução do problema, a Equação 2 é a conversão do valor discreto do indivíduo para o valor contínuo, numa faixa delimitada por um valor mínimo (*min*) e um valor máximo (*max*). A Tabela 2 é um exemplo de uma população final para o problema da bola de tênis. O melhor indivíduo encontrada pelo AG é o 45, que ao ser decodificado com a Equação 3 para o espaço continuo, retorna o valor de v_0 igual a 4,3945 m/s. Esse é melhor valor para a velocidade de lançamento da bola de tênis, de modo que a bola de tênis fique mais próxima de um metro, ou mais precisamente, 0,9846 m. O erro de 0,0154 m (1,5%) é o menor erro possível para a codificação em 10 genes (1024 partes), uma quantidade maior de genes poderia ser utilizada para aumentar a precisão.

Após atingir o critério de parada, a população é ordenada de acordo com a aptidão dos indivíduos, aquele que possuir a maior aptidão será a solução e decodificada para fornecer o resultado do problema.

A Equação 2 é a conversão do valor discreto do indivíduo para o valor contínuo, numa faixa delimitada por um valor mínimo (*min*) e um valor máximo (*max*).

Com relação ao exemplo da bola de tênis, a Tabela 2 mostra um exemplo de população final para o problema. O melhor indivíduo encontrado pelo AG é o 45, que ao ser decodificado com a Equação 3 para o espaço continuo, retorna o valor de v_0 igual a 4,3945 m/s. Esse é melhor valor para a velocidade de lançamento da bola de tênis, de modo que a bola de tênis atinja a marca de um metro com menor erro (0,9846 m com erro de 0,0154 m, aproximadamente 1,5%). Para uma codificação em 10 genes (1024 partes), esse é o menor erro possível. Entretanto, uma quantidade maior de genes poderia ser utilizada para aumentar a precisão.

$$Valor = min + \left(\frac{indivíduo \cdot (max - min)}{2^{genes}} \right) \quad (2)$$

$$v_0 = \frac{45 * 100}{1024} = 4,3945 \quad (3)$$

Isso posto, de posse do algoritmo, sua execução pode ser realizada em desktops, servidores, notebooks ou mini computadores. No entanto, nesse trabalho, os algoritmos serão implementado em sistemas embarcados.

2.2 Sistemas Embarcados como plataforma para execução de AG

A evolução da eletrônica tornou possível a redução dos tamanhos dos dispositivos após a substituição de relês e circuitos lógicos integrados por microcontroladores. O resultado desse avanço foi o surgimento de sistemas inteligentes e sistemas dedicados a tarefas específicas. Sistemas embarcados são dispositivos microprocessados focados em executar uma única tarefa, como chips que leem os sensores de temperatura nas geladeiras

Tabela 2 – Exemplo de população final

v_0	v_0 discreto	v_0 codificado (indivíduo)	$f(v_0) = \Delta H(m)$	Erro
4,3945	45	0000101101	0,9846	0,0154
4,3945	45	0000101101	0,9846	0,0154
4,3945	45	0000101101	0,9846	0,0154
4,3945	45	0000101101	0,9846	0,0154
4,4922	46	0000101110	1,0289	0,0289
4,4922	46	0000101110	1,0289	0,0289
4,1992	43	0000101011	0,8990	0,1010
2,7343	28	0000011100	0,3812	0,6188
6,1523	63	0000111111	1,9299	0,9299
0,9766	10	0000001010	0,0486	0,9514
8,7890	90	0001011010	3,9385	2,9385
97,5586	999	1111100111	485,2667	484,2667
99,9023	1023	1111111111	508,8623	507,8623

modernas e controlam o momento exato que o compressor deve ligar. Por outro lado, um computador pessoal, pode ser utilizado para diversas tarefas, dependendo do programa que está sendo executado.

Segue breve histórico do surgimento dos primeiros computadores e sistemas embarcados.

2.2.1 História

Os primeiros computadores, desenvolvidos na década de 50 tinham como componente básico a válvula termiônica ou válvula eletrônica que consistem em um tubo de vidro a vácuo, com filamentos metálicos aquecidos para controlar o fluxo de energia com potencial acima de 80V. Devido às condições exigidas pelas válvulas, o ENIAC foi o primeiro computador inventado, ocupava uma área de 180m², pesando 30Ton, consumindo 200kW e clock de 1kHz, esse foi o marco da primeira geração de computadores.

A segunda geração iniciou no ano de 1953, com o Manchester TC (*Manchester Transistor Computer*), o primeiro computador que substituía as válvulas termoiônicas por transistores. Essa mudança reduziu o consumo para 150W e aumentou o clock para 1MHz. A partir da criação do Manchester TC, os computadores passaram a ser vendidos para empresas.

A terceira geração de computadores é delimitada pelos anos 1965 e 1975. O tamanho físico e o consumo dos computadores foi reduzido após a compactação de arranjos de transistores em circuitos integrados. Além disso a redução do custo financeiro para a fabricação, facilitou a disseminação dos computadores pessoais.

Em paralelo a esses eventos, a empresa japonesa BUSICOM iniciou um projeto para uma calculadora eletrônica no ano 1969, e enviou seus engenheiros para os Estados Unidos, com objetivo de apresentar o projeto ao Marcian Hoff, o especialista em computadores da

Intel Corporation. Hoff percebeu que o projeto tinha um potencial maior que apenas as funções da calculadora, o chip poderia ser modificado para executar as funções segundo sua programação. A Intel comprou a licença da BUSICOM e lançou em 1971 o primeiro processador de 4 bits. Esse processador continha um conjunto de instruções complexas que podia ser combinadas para executar tarefas mais complexas, essa arquitetura ficou conhecida como CISC (*Complex Instruction Set Computer* - Conjunto de instruções complexas).

No final da década de 70, começaram as primeiras observações, buscando tirar o máximo proveito dos processadores. Foi constatado que, para executar uma instrução na arquitetura CISC, são gastos vários acessos a memória, isso tornava o processamento das instruções mais lentas, a medida que os programas ficassem mais complexos. Desse problema seguiram duas vertentes a melhoria da memória ou a troca da arquitetura do processador.

A Intel, protegendo seu produto, passou a investir em estudos de melhoria do hardware das memórias, mesmo sabendo que sempre seria limitada pelo avanço da tecnologia e que o custo desse investimento seria repassado para seus clientes. Outras empresas como a Motorola e MOS Technology também seguiram essa linha de pesquisa.

Em contra partida, o grupo inglês Acorn Computer Group investiu seus estudos no desenvolvimento de uma arquitetura com instruções mais simples e separando a memória de controle e instrução, essa modificação otimizava os recursos do processador. Essa arquitetura ficou conhecida como arquitetura RISC(*Reduced Instruction Set Computer* - Conjunto reduzido de instruções). No ano de 1985, a Arcon desenvolveu o primeiro processador RISC, denominado ARM1 (*Acorn RISC Machine 1*).

Nessa mesma época surgiram os primeiros sistemas operacionais desenvolvidos exclusivamente para processadores CISC, marcando a quarta geração de computadores, e suprimindo a propagação dos processadores ARMs. Em 1990, Apple Computer buscava um processador para construir um computador puramente estático, e se uniu a Acorn e a VLSI Technology formando a ARM Limited (*Advanced RISC Machines* - Máquinas RISC avançadas), e licenciando a produção de processadores ARM por terceiros.

A partir dos anos 90, o ARM dominou o mercado de processadores para sistemas embarcados devido a suas características de simplicidade de fabricação, de baixo consumo e custo.

No ano inicio do século XXI o desenvolvimento e a liberação de projetos de computadores estáticos, deu origem ao termo “*Open Hardware*”. Essa evolução facilitou a expansão de hardwares com processamento suficiente para atender a demanda de tarefas já existente, ganhando força com a tecnologia de IoT (*Internet of Things* - Internet das coisas) e a tecnologia *mobile*.

2.2.2 Gerenciamento de Recursos

Nas ultimas décadas houve um aumento no número de dispositivos *mobile*, como *smartphones*, notebooks, tablets, dentre outros. Ao mesmo tempo intensificaram as críticas sobre a duração da bateria, aquecimentos e sensibilidade a água, poeira e queda. Essas limitações que afetam esses dispositivos também incluem os sistema embarcados. Em alguns casos, sistemas embarcados podem ser aplicados em dispositivos móveis, isso implica na limitação de alguns recursos essenciais. Sistemas embarcados móveis devem possuir sua própria fonte de energia elétrica. Atualmente o mais comum, é o uso de acumuladores de eletricidade, como baterias, de preferência recarregáveis. O consumo deve ser bem gerenciado para manter a carga da bateria por um período maior, ou seja, dispositivos embarcados podem “dormir” (*standby*) por curtos intervalos, quando não estão em operação. Em *standby* os dispositivos desligam os periféricos, executando apenas as funções mais básicas, reduzindo o consumo de energia.

Os dispositivos eletrônicos, em geral, são sensíveis a corpos estranhos, como líquidos, poeira, ou até mesmo a descargas eletrostáticas provocadas pelo contato direto com componentes. Para proteger os circuitos os dispositivos embarcados são montados sobre uma estrutura denominada *case*, que impede parcialmente ou completamente a entrada de poeira, líquido e toque com os componentes internos. O padrão *International Electrotechnical Commission* - Comissão Internacional de Eletromecânica (IEC 60529) classifica grau de proteção dos dispositivos quanto a resistência a entrada de corpo sólidos, líquidos e a impactos mecânicos.

Em sistemas embarcados evita-se o uso de ventoinhas para evitar sobreaquecimento, pois ela é sensível a vibrações mecânicas, geram ruído por efeito magnético e precisam de uma entrada de ar, reduzindo a isoliação contra líquidos e sólidos e consomem muita bateria. Em dispositivos embarcados, para dissipar o calor gerado pelos componentes eletrônicos, costuma projetá-los com microcontroladores e microprocessadores que não necessitam de refrigeração forçada. Caso seja necessário utilizam dissipadores ou o próprio *case* para dissipar o calor.

De qualquer modo, o processamento deve ser otimizado para a execução da tarefa em linguagens com menor nível de abstração e maior portabilidade, facilitando a adaptação do software a outros hardwares embarcados. Isso torna-se necessário devido a constante evolução do hardware pelas fabricantes, devido as diferentes arquiteturas de *hardware* presentes no mercado (8, 16, 32 e 64 bit) e a redução nos custos.

A portabilidade do código pode ser conseguida de diferentes formas, como comentado a seguir:

- ❑ Utilização da biblioteca *C99 Standard Data Types* `<stdint.h>` se a programação for realizada na Linguagem C.
- ❑ Instalação de uma máquina virtual desenvolvida para diversos hardwares, dessa

forma, o executável é compatível com diversos dispositivos sem que seja necessário recompilar o código. Essa estratégia aumenta o consumo de memória e processamento, como ocorre com a linguagem de programação Java.

- ❑ Interpretação de *scripts*, como ocorre na linguagem python, *scripts* Linux, contudo, a desvantagem está na redução da performance.
- ❑ Utilizar ferramentas multiplataforma de desenvolvimento que geram o código uma vez e podem compilar o código para várias plataformas como ocorre o Xamarin da Microsoft, Qt Creator, o Firemonkey da Embarcadero etc.

2.2.3 Periféricos

Microcontrolador ou microprocessador é um único circuito integrado o qual contém um núcleo de processador, memória e periféricos programáveis de entrada e saída. A memória que armazena o programa pode ser RAM, NOR flash ou PROM. Atualmente existe uma vasta variedade de periféricos tanto para a entrada quanto para a saída de dados, dentre eles podemos citar: bluetooth, UART, I2C, 1-Wire, SPI, HDMI, USB, PS2 etc.

É por meio dos periféricos que o microcontrolador ou microprocessador pode se comunicar com dispositivos como mouse, teclado, sensores, atuadores, monitores, sirenes, módulos celulares, GPS, WiFi, dentre outros dispositivos.

O trabalho proposto utiliza GPS para fornecer a localização dos caminhões na mina e WiFi para transmitir e receber os dados do servidor.

2.2.4 GPS

O módulo GPS capta os sinais de micro-ondas de satélites e por meio de triangulação consegue determinar, a posição global do mesmo. O GPS precisa se conectar com no mínimo três satélites para ter uma posição válida, mas a precisão pode melhorar quando o módulo se conecta a mais satélites.

A medição de distância pelas coordenadas geográficas exige o uso de uma equação capaz de converter latitude e longitude em distância, levando em consideração a superfície esférica da Terra, como mostrado na Figura 5. Essa equação foi criada por José de Mendoza y Ríos em 1801, e em 1835 James Inman a denominou equação de Haversine.

A Formula utiliza funções trigonométricas para calcular a distancia em metros em uma esfera como a Terra, por meio das latitudes e longitudes dos locais em questão(ALAM et al., 2016).

$$\text{hav}\left(\frac{d}{r}\right) = \frac{1 - \cos\left(\frac{d}{r}\right)}{2} \quad (4)$$

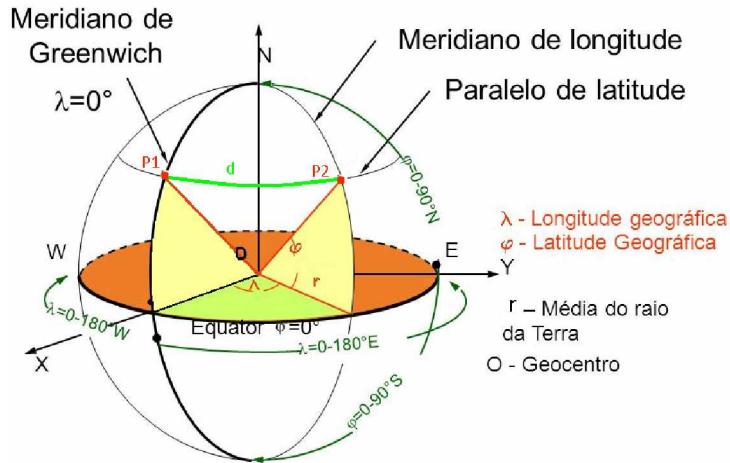


Figura 5 – Distância entre dois pontos na superfície da Terra.

Fonte: <http://slideplayer.com.br/slide/2839790>

$$d = 2r \arcsin \left(\sqrt{\sin^2\left(\frac{\varphi_2 - \varphi_1}{2}\right) + \cos(\varphi_1)\cos(\varphi_2)\sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)} \right) \quad (5)$$

Onde:

- ❑ hav: é a formula de Haversine;
- ❑ d: é a distância entre os dois pontos na Terra;
- ❑ r: é o raio da Terra;
- ❑ φ_1 : é a latitude do ponto1 (P1);
- ❑ φ_2 : é a latitude do ponto2 (P2);
- ❑ λ_1 : é a longitude do ponto 1 (P1);
- ❑ λ_2 : é a longitude do ponto 2 (P2).

A escolha de uma coordenada de referência para o cálculo da distância é a base para um sistema de geolocalização.

No trabalho proposto, o GPS é utilizado para fornecer a latitude e longitude do local em que ele se encontra com o objetivo de se criar uma cerca eletrônica. Ela é um polígono imaginário delimitado em torno de um ponto de referência pré-fixado, como mostrado na Figura 6. Essa técnica permite georreferenciar um objeto munido de GPS e monitorar sua movimentação entre essas áreas.

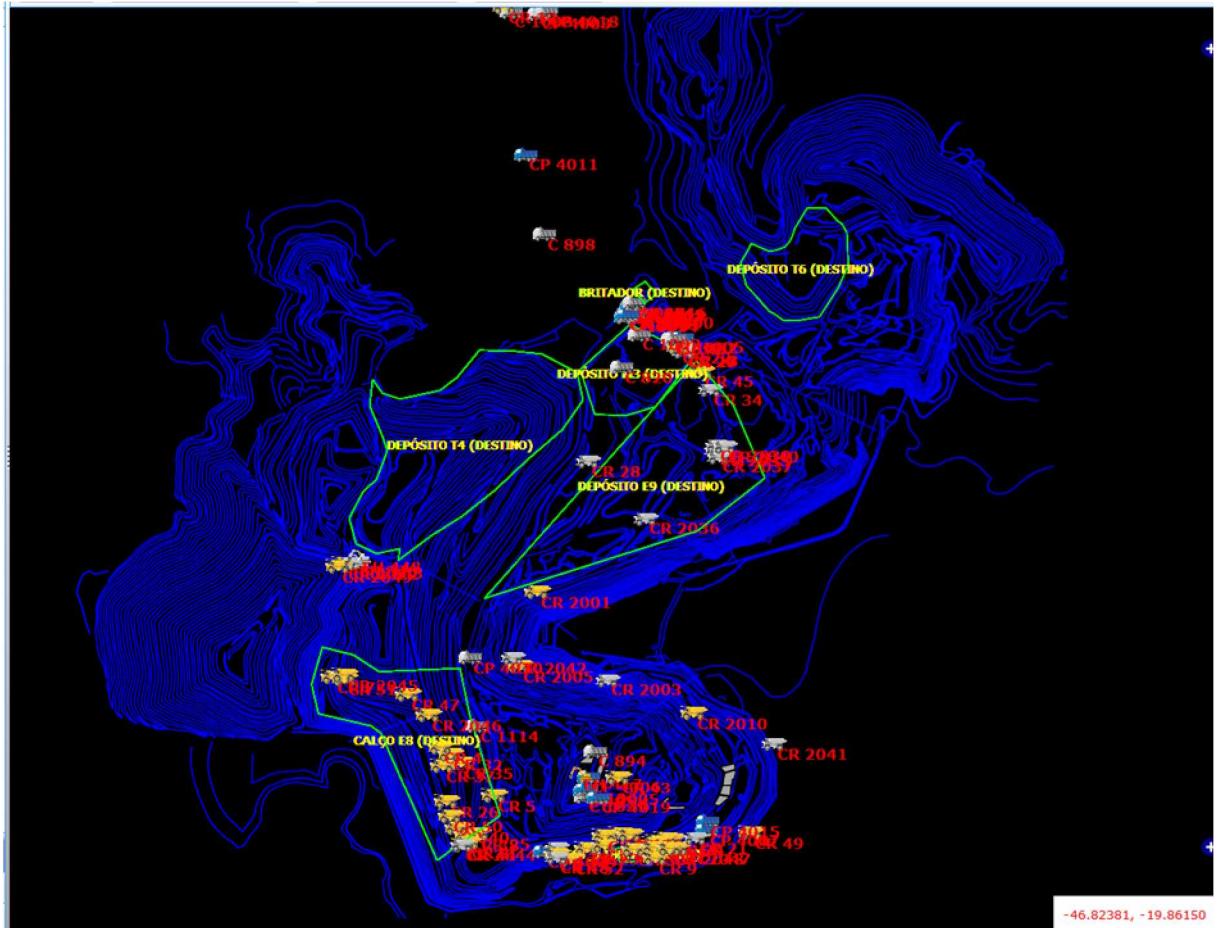


Figura 6 – Cercas eletrônicas em um mapa (verde) com os nomes locais (amarelo).

Fonte: acervo do autor

O polígono pode ser implementado por meio de formas geométricas simples como: quadrado, círculo, polígono, ou formas geométricas aleatórias compostas por diversas coordenadas. No trabalho em questão utiliza cercas eletrônicas circulares por terem o cálculo mais simples e atenderem a proposta. Esse polígono é denominado cerca eletrônica.

A cerca eletrônica circular é levada à prática escolhendo-se um ponto central e um raio. Em seguida utiliza-se a equação 5 para calcular a distância entre o ponto central e a coordenada do GPS. Se a distância calculada for menor que o raio da cerca eletrônica, o objeto está dentro da região.

2.3 Comunicação de dados

O uso da rede sem fio é fundamental para a transmissão de dados entre dispositivos móveis, podendo ser rádio, GPRS ou Wifi. Todavia, no trabalho proposto utilizou-se rede WiFi, porque elas se destacam pela velocidade de transmissão de dados. A comunicação

WiFi realiza a troca de informações entre os caminhões e o servidor. A comunicação consiste na transferência de informações da telemetria e posição dos caminhões da mina.

A rede WiFi suporta o protocolo de comunicação *Transmission Control Protocol* - Protocolo de controle de transmissão (TCP) e *User Datagram Protocol* - Protocolo Dados de Usuário (UDP). O TCP é um dos principais protocolos da Internet, utilizado principalmente em redes locais que precisam de confiabilidade dos dados, pois permite a verificação de erros de transmissão.

Contudo, para proporcionar a confiabilidade o TCP precisa gerar um *overhead* de informações transmitidas, o que cria uma grande latência dos dados na rede.

Embora o TCP seja mais confiável, na maioria das aplicações, não há necessidade de que 100% dos pacotes cheguem ao destinatário. Nesses casos, é mais indicado o uso do protocolo UDP, esse protocolo não possui verificação de erro, no entanto, considerando uma determinada rede, o protocolo UDP pode ser usado para transmitir os dados com uma taxa de transmissão maior, pois não necessita aguardar a confirmação de recebimento do pacote, pelo receptor. O UDP é amplamente utilizado em redes simples de internet local, sendo suportado por diversos módulos WiFi de baixo custo, como é o caso do módulo ESP8266.

2.3.1 ESP8266

O ESP8266, mostrado na Figura 7, é um módulo produzido pela empresa chinesa Espressif. Esse dispositivo une um microcontrolador de 8 bits e 80MHz, a um módulo de comunicação WiFi. Uma das vantagens desse dispositivo é que ele suporta a configuração *Wireless Access Point* (WAP)³, ou, em outras palavras, cria redes WiFi e ao mesmo tempo pode criar um servidor Web, TCP ou UDP.

Uma rede local dentro de um veículo permite que todos os dispositivos, como sensores, display, entre outros periféricos que também possuem WiFi, possam se conectar e comunicar, dispensando o uso de cabeamento. Esse procedimento reduz ruídos e interferências, além de reduzir o tempo gasto durante a instalação e manutenção da solução completa.

Esse módulo pode ser programado de três formas diferentes:

- ❑ A primeira por meio de comandos *Hayes*, mais conhecidos como comandos AT, nesse modo o ESP8266 recebe comandos por meio da porta serial para executar funções pre-configuradas. Esses comandos podem ser enviados por um segundo microcontrolador ou um computador, dispensando a programação do ESP8266;
- ❑ A segunda maneira é a programação na linguagem Lua. A vantagem da linguagem Lua é o fato de ser interpretada por *scripts*, dispensando a compilação do código a cada modificação;

³WAP é um dispositivo de uma rede sem fio que cria a conexão entre todos os dispositivos da rede

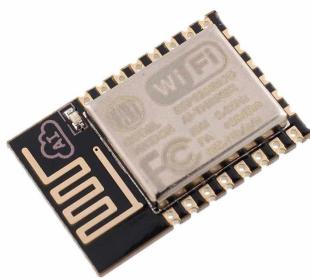


Figura 7 – Módulo ESP8266.

Fonte: <http://www.botnroll.com>

- A terceira é a programação em linguagem C++, com a abstração do Arduíno. Existe uma grande quantidade de bibliotecas que facilitam a implementação dos códigos.

O módulo pode ser configurado para enviar e receber mensagens através do protocolo UDP para um IP fixo. Mas, primeiro é necessário definir o padrão do formato de texto da mensagem que será utilizado.

O *JavaScript Object Notation* (JSON) é um padrão de formato de texto para transmissão de objetos, de forma que seja legível a humanos e é de fácil interpretação pela máquina.

O quadro abaixo mostra um exemplo de um pacote de dados no formato JSON. O agrupamento dos atributos e valores pares contribui para interpretação dos dados. Esse é um dos formatos utilizados em comunicação com servidores e pode ser facilmente decodificado com o propósito de ser utilizado no armazenamento de informações em banco de dados.

```

1 { "MEDICAO": 
2 [
3 { "SENSOR": "1", "TEMPERATURA": "25.5", "HUMIDADE": "65" },
4 { "SENSOR": "2", "TEMPERATURA": "37.0", "HUMIDADE": "67" },
5 { "SENSOR": "3", "TEMPERATURA": "18.0", "HUMIDADE": "60" },
6 { "SENSOR": "4", "TEMPERATURA": "30.5", "HUMIDADE": "78" }
7 ]
8 }
```

Bancos de dados são conjuntos de dados estruturados. Em computadores os dados são organizados por softwares chamados de *Database Management Systems* - Sistema de

Gerenciamento de Banco de Dados (DMS). Estes softwares são responsáveis por controlar a busca e manipulação de dados de modo que apenas usuários autorizados tenham acesso a certos tipos de dados.

No trabalho desenvolvido, as informações de todos os equipamentos da mina, principalmente escavadeiras e caminhões, são armazenadas em um banco de dados.

2.4 Interface do usuário (Display)

Hoje em dia, existem muitos aplicativos para smartphone e tablets que são usados como IHM para o controle remoto de vários sistemas (iluminação, construção inteligente, caixas de mídia, monitoramento de parâmetros em geral), empregados na automação residencial, comercial e industrial, como mostrado a Figura 8.

Nos últimos anos, devido a popularidade, poder de processamento, conectividade, diversidade de aplicativos e disponibilidade de ferramentas de desenvolvimento, os tablets vem sendo cada vez mais utilizados em aplicações onde existe a necessidade de uma IHM. Além disso, atualmente existem muitas ferramentas multiplataformas de desenvolvimento para dispositivos móveis que utilizam sistema operacional Android.



Figura 8 – IHM utilizando dispositivos *mobile*.

Fonte: <https://www.pro-face.com>

2.4.1 Android e Firemonkey

O Android surgiu em 2003. A ideia inicial era trabalhar em sistemas para câmeras digitais, mas como o mercado não era tão amplo a empresa focou no mercado *mobile*. O sistema nasceu *Open Source*, baseado no Kernel Linux, contando com uma interface

simples, funcional e já integrada a diversos instrumentos, a ideia era oferecer um sistema gratuito e também simples aos desenvolvedores.

O android foi comprado pela Google em 2005, nascendo o Google Mobile Division, uma divisão de pesquisa da empresa focada em tecnologia móvel. Em 2007 empresas como a HTC, Dell, Intel, Motorola, Qualcomm, Texas Instruments, Samsung, LG, T-Mobile, Nvidia, o próprio Google em um total de mais de 80 empresas, reuniram-se em forma de um consórcio de tecnologia, com objetivo de criar uma plataforma de código aberto para smartphones. O resultado foi o primeiro Android comercial, lançado oficialmente em 22 de outubro de 2008. Em 2013, o Android estava em 88,7% dos smartphones vendidos no Brasil. No ano seguinte, subiu para 91%. No início de 2016, a fatia do Android já representava 93%. No ano de 2016 foram ativados 296,9 milhões de smartphones, 75 milhões de aparelhos a mais que 2015, isso significa 86,2% do mercado, como mostrado na Figura 9.

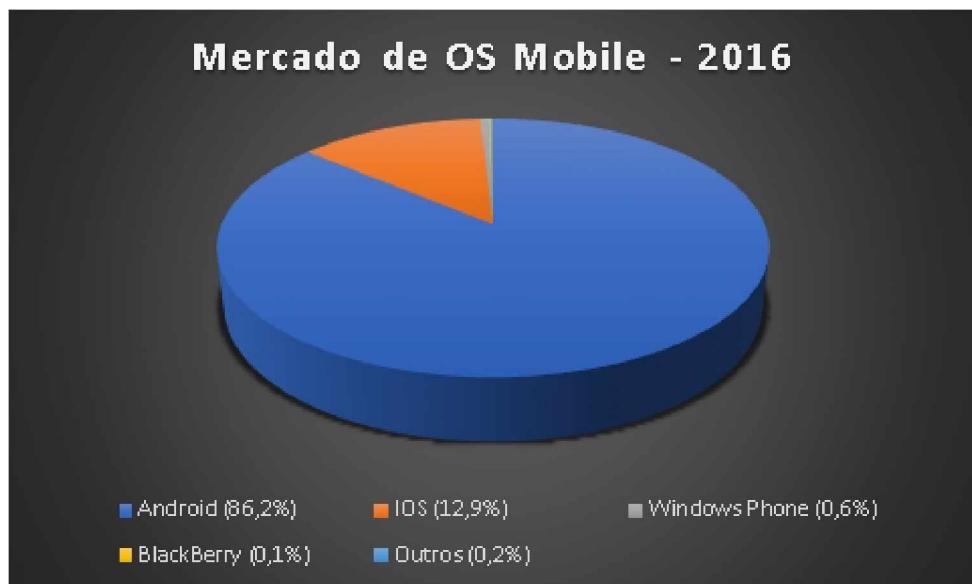


Figura 9 – Composição do mercado de OS *mobile* (2016).

Fonte: <https://www.tecmundo.com.br>.

O sistema operacional Android favorece a aquisição de dispositivos de baixo custo e com a vantagem de ser de Código Aberto (*Open-Source*) (NICOLAE; LUCACI; MOISE, 2013). Ademais, algumas empresas têm investido em compiladores multiplataforma com objetivo de facilitar a implementação de código e garantir a portabilidade para outras plataformas, alguns exemplos conhecidos são o Xamarin e o Firemonkey.

O FireMonkey é baseado numa GUI denominada VGScene, projetada pela empresa russa Ulan-Ude, ou mais precisamente por Eugene Kryukov. A ideia de Eugene era projetar uma GUI baseada em vetores, que futuramente poderia compor a nova geração de GUI de desenvolvimento. Essa empresa foi comprada pela *Embarcadero Technologies* no ano de 2011. Eugene continuou o desenvolvimento do firemonkey vinculando às linguagens

da Embarcadeiro C++ e *object-pascal*. No mesmo ano foi lançado a versão “XE2” que continha o *framework* padrão *Visual Component Library* (VCL) e acionando a primeira versão do firemonkey, com compiladores para Windows e Mac. A partir desse ano a Embarcadeiro lançou novas versões anualmente, adicionando melhorias ao *firemonkey*, como a adição de compiladores para Android em 2013, posteriormente suporte ao window 10, até chegar no ano de 2017 com o lançamento da versão “XE10.2 Tokyo”, com o compilador de codinome “Godzilla” que suporta compilação para a plataforma Linux.

CAPÍTULO 3

Problema de Despacho utilizando Algoritmo Genético

Existem vários modelos de AG que podem ser aplicados para resolver um determinado problema. No entanto, a escolha de um modelo ineficiente, pode exigir mais gerações ou um esforço computacional maior para fornecer a mesma solução. A escolha de um AG depende da aplicação e dos recursos de hardware disponíveis. O modelo mais utilizado em problemas de busca de rotas é o de indivíduos de tamanho variável, que será apresentado a seguir.

3.1 Modelo de busca de rotas

Para definir o modelo utilizado, deve-se partir do mapa da mina e convertê-lo em um grafo. Os pontos estratégicos, como frente de minério, intercessão e pontos de basculamento são denominados nós. Cada nó recebe um número único para representá-lo no algoritmo. As vias que interligam os nós são convertidas em linha reta, distância e será representada pelos ramos do grafo.

As rotas são todos os caminhos que interligam um nó de origem a um de destino. O indivíduo representa uma rota válida partindo de uma frente de extração para um ponto de basculamento ou vice-versa.

Cada indivíduo é representado por um vetor de números inteiros, preenchido pelos nós que formam a rota organizados sequencialmente da origem até o destino. Como exemplo temos o indivíduo [2 3 6 7], onde os números 2,3,6,7 são os pontos estratégicos (nós). Os elementos 2 e 7 são a origem e o destino respectivamente. Os números 3 e 6 são intercessões entre essa rota e outras que levam a outros nós.

Esse tipo de modelamento tem como característica, o fato de todos os indivíduos serem rotas válidas. No entanto a quantidade de nós que compõem uma rota pode variar de acordo com a quantidade de intercessões. Isso resulta em indivíduos de tamanhos diferentes, sendo essa característica um diferencial do AG utilizado nesse trabalho.

3.2 Função de aptidão

Normalmente em problemas de busca de rotas, a aptidão considera a menor distância como o objetivo principal. Contudo, o problema pode levar em conta outros fatores, como trânsito, condições das vias, hábitos comuns dos motoristas, pontos obrigatórios, qualidade do material que está sendo transportado, economia de combustível etc. Em problemas de alocação dinâmica a função de aptidão leva em consideração os seguintes fatores:

- ❑ Minimizar a distância de transporte (DMT);
- ❑ Minimizar a formação de filas;
- ❑ Minimizar a ociosidade das máquinas.

A seguir são apresentados procedimentos para otimizar rotas e filas:

O primeiro procedimento a ser alcançado é encontrar uma rota que represente a menor distância entre o ponto de extração e o de basculamento (Distância Média de Transporte (DMT)), evitando-se desse modo, que o algoritmo crie rotas longas para esperar os outros caminhões terminarem seus processos de carregamento ou descarregamento.

O valor DMT é encontrado pela somatória das distâncias entre os nós que compõem a rota, como mostrado na Equação 6.

$$DMT(x) = \sum_{i=0}^{sizeof(x)-1} d(x_i, x_{i+1}) \quad (6)$$

Onde:

- ❑ x_n : é um ponto de decisão da rota;
- ❑ d : É a distância do percurso entre os dois pontos de decisão da rota.

O segundo procedimento é reduzir a formação de fila. Durante o ciclo de operação de uma mineradora, a frota de caminhões trabalham em dois grupos: o primeiro grupo tem como origem a frente de minério com destino a britagem. O segundo possui a origem na britagem e destino a frente de minério. Desse modo, cada grupo não interfere no outro, todavia, os caminhões podem se encontrar em pontos de carregamento e basculamento, ocasionando filas e perda por tempo ocioso.

O tempo de fila deve ser analisado com cuidado, pois apesar da fila ser uma perda de tempo, a escolha por outra rota mais longa, pode gerar um atraso maior que o tempo esperando em fila. Pensando nisso, é necessário considerar o tempo total até chegar ao destino (incluindo o tempo em fila). E esse tempo pode ser encontrado pelo seguinte algoritmo:

1. Cria uma lista com todos os outros caminhões que sigam para mesma rota deste.

2. Calcula o tempo de transporte ($T_t = \frac{dmt}{vm}$), onde **dmt** é a distância média de transporte e **vm** é a velocidade média do caminhão;
3. Elimina da lista todos caminhões que tenham T_t maior que esse;
4. Cria uma variável para acumular o tempo perdido em fila e atribui o tempo de carregamento da frente T_c (ou basculamento se estiver carregado), a variável é denominada T_f ;
5. Varre a lista procurando por caminhões cuja diferença do tempo de transporte com o em questão, seja menor que T_f . Caso seja, temos uma condição de fila, incrementamos T_f com o tempo gasto para que o caminhão carregue $T_f = T_f + T_c$, e repetimos o passo 5. Quando não for encontrado nenhum caminhão que satisfaça essa condição, o algoritmo terá encontrado o tempo gasto para chegar ao destino (equação 7).

$$T_d = T_t + T_f \quad (7)$$

Onde:

- ❑ T_f : Tempo em parado fila;
- ❑ T_t : Tempo de transporte;
- ❑ T_d : Tempo gasto até o destino (bascular ou carregar).

O terceiro procedimento é a minimização da ociosidade das máquinas de carregamento e do britador. As escavadeiras são alocadas de acordo com sua capacidade de extração, ou seja pela quantidade de toneladas de material que é removida por hora. Se o número de caminhões nessa frente for inferior a sua capacidade, a escavadeira fica ociosa. Por outro lado, se houver mais caminhões do que a escavadeira consegue atender, a formará fila no carregamento. Apesar disso, na prática, as máquinas são alocadas com capacidade superior ao número de caminhões necessários para atender a capacidade da escavadeira.

O número de caminhões é calculado por meio do percentual que se utilizará da escavadeira ($\%P$), levando-se em conta a capacidade máxima dos caminhões (Cp_{max}). Dessa maneira, a alocação dinâmica deve distribuir os caminhões entre as rotas, de modo que todas as frentes mantenham sua produção ideal ($\Delta CP = 0$).

$$\Delta CP = |(\%P \cdot Cp_{max}) - Cp_{atual}| \quad (8)$$

A função de aptidão precisa levar em consideração todos os fatores apresentados. Como as unidades dos termos são: o valor da distância em metros, tempo e erro absoluto, a equação exponencial e^{-x} é usada com objetivo de limitar os valores para a escala unitária

(0..1), pois para valores muito altos, o $\lim_{x \rightarrow \infty} e^{-x} = 0$ e para valores nulos, o $\lim_{x \rightarrow 0} e^{-x} = 1$. Uma constante de peso (K_n) é utilizada para justar a influência de cada fator ao resultado final.

$$f(ind) = K_1 \cdot e^{-DMT(ind)} + K_2 \cdot e^{-T_d} + K_3 \cdot e^{-\Delta CP} \quad (9)$$

3.3 Criação de indivíduo

O indivíduo é implementado por meio de listas de ID positivos e maiores que zero, onde cada ID representa um ponto geográfico de interesse. Cada rota é determinada pela organização sequencial dos ID no vetor implementado em linguagem de programação C++;

O espaço de busca é a área limitada que contém a solução para o problema, dentro dessa área, a solução escolhida pode apresentar um erro aceitável, moderada, ou inaceitável, dependo das restrições estabelecidas. Uma rota inaceitável, por exemplo, as rotas que não chegam aos destinos. No trabalho desenvolvido o espaço de busca é composto apenas por rotas válidas, pois a matriz de rotas é gerada somente com rotas válidas, o que facilita a busca pela melhor rota.

A matriz é preenchida de acordo com o grafo da mina, como o mostrado no exemplo da Figura 10. Em seguida fixa-se um nó de origem e verifica-se quais são os caminhos que saem do nó escolhido e quais são os nós que esses caminhos dão acesso. Por exemplo, se o nó escolhido for o nó 1, de acordo com o grafo, ele dará acesso aos nós 2, 7, 9 e 10. Então, a primeira coluna da matriz conterá 1 nas posições (1,2), (1,7), (1,9) e (1,10) e zero nas outras posições da linha.

A diagonal será sempre nula, pois não existe rota do nó para ele mesmo. A Figura 11 mostra a matriz do grafo apresentado.

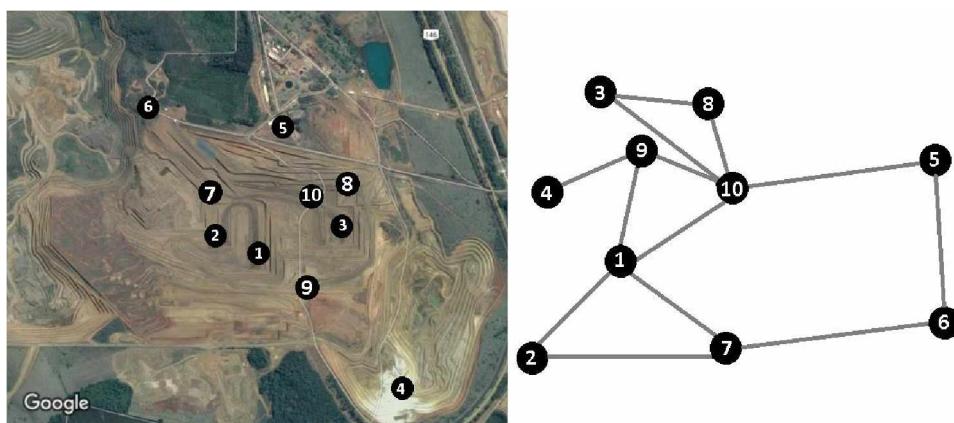


Figura 10 – Grafo de uma mina genérica.

Fonte: Acervo do autor

	Destino									
	1	2	3	4	5	6	7	8	9	10
ORIGEM	1	0	1	0	0	0	1	0	1	1
	2	1	0	0	0	0	1	0	0	0
	3	0	0	0	0	0	0	1	0	1
	4	0	0	0	0	0	0	0	1	0
	5	0	0	0	0	0	1	0	0	1
	6	0	0	0	0	1	0	1	0	0
	7	1	1	0	0	0	1	0	0	0
	8	0	0	1	0	0	0	0	0	1
	9	1	0	0	1	0	0	0	0	1
	10	1	0	1	0	1	0	0	1	0
LEGENDA	0	FALSO								
	1	VERDADEIRO								

Figura 11 – Matriz de rotas para o exemplo da Figura 10.

Fonte: Acervo do autor

A rotina utilizada para construir um indivíduo começa criando uma lista e adiciona o nó origem que passa a ser denominado de pivô. Em seguida, seleciona-se a linha da matriz de rotas referente ao nó acrescentado, e escolhe-se aleatoriamente um nó para ser adicionado a lista. O novo nó adicionado à lista passa a ser o novo pivô e, a partir dele, será selecionada uma nova rota que levará ao próximo nó, que, por sua vez, será o próximo pivô. Então, entende-se por pivô o último nó acrescentado na tabela e na qual será a origem para o próximo nó em direção ao destino. Vale salientar que a escolha é aleatória, porém, somente entre os nós válidos.

O processo se repete até que o destino final seja alcançado (ponto de basculamento ou carregamento). A lista criada será um indivíduo, que irá compor uma população. A lista composta pelos indivíduos forma uma população. A população sofre alterações ao longo do período de evolução, os métodos de seleção, cruzamento e mutação são os agentes que qualificam e modificam a população.

3.4 Seleção

Em problemas de busca de rotas os operadores de seleção são semelhantes aos utilizados em outros algoritmos genéticos. Existem três métodos que são muito utilizados: o método da roleta, do *ranking* e do torneio.

No trabalho em questão os métodos de seleção por torneio, roleta e *ranking* foram experimentados, porém, nos experimentos realizados, somente o método de seleção por torneio forneceu soluções com menor número de gerações, isto é, com menor tempo de processamento.

Por exemplo, o algoritmo do torneio implementado seleciona aleatoriamente um grupo de 5 indivíduos, desse grupo o que possuir maior aptidão é selecionado para a fase de

cruzamento.

3.5 Cruzamento

Em algoritmos de busca de rotas, o cruzamento é diferente do cruzamento utilizado no AG convencional. O método consiste em primeiramente selecionar os dois indivíduos, posteriormente seleciona-se um dos nós comuns, excluindo-se a origem e o destino das duas rotas e em seguida trocam-se as semi-rotas entre os dois indivíduos, conforme o exemplo da Figura 12.

Não é possível cruzar dois indivíduos que não possuam nós em comum. Nesse caso, um novo par é selecionado para o cruzamento. Existe a possibilidade de algum nó, que esteja presente em um dos pais, apareça duplicado no filho, isso cria *loops*, como mostrado na Figura 13, tornando o caminho mais longo do que deve ser.

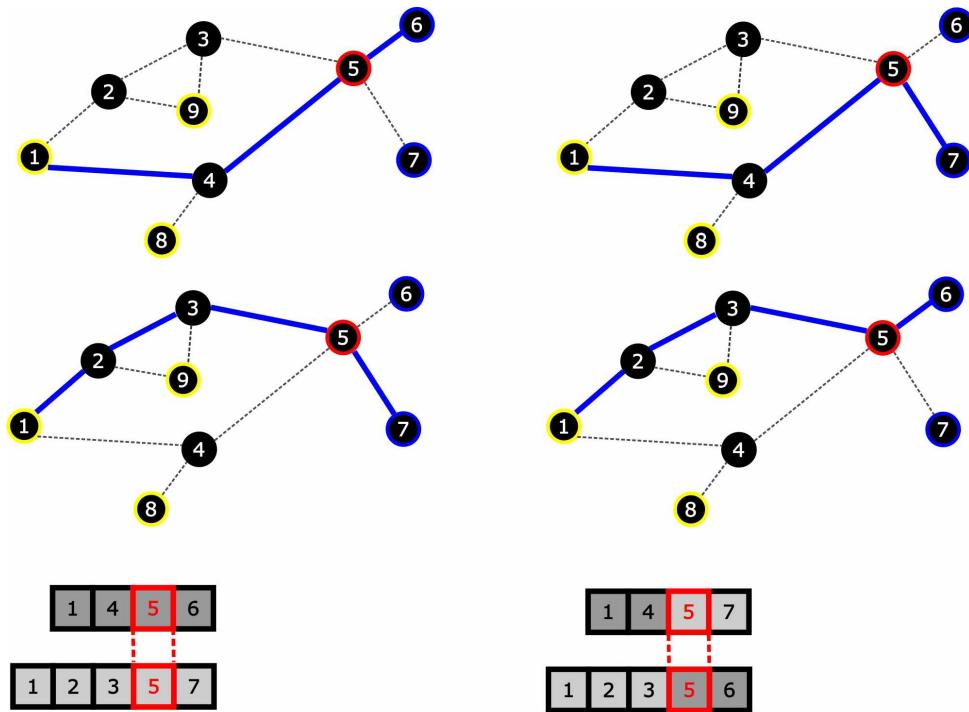


Figura 12 – Cruzamento de dois indivíduos.

Fonte: Acervo do autor

Um algoritmo de correção de laços elimina o segmento do trajeto que leva à interseção duplicada e sorteia um novo caminho alternativo, caso o nó em questão não possua outras opções de rotas, o procedimento é repetido no segmento atual até que haja *loops* na rota.

Assim como o *loop*, o “caminho sem saída” é uma falha do algoritmo de construção de indivíduos. Isso ocorre quando a rota gerada não possui outra opção de formar um trajeto, exceto retroceder, o que torna parte do trajeto inútil e faz com que o caminho fique mais longo que o necessário.

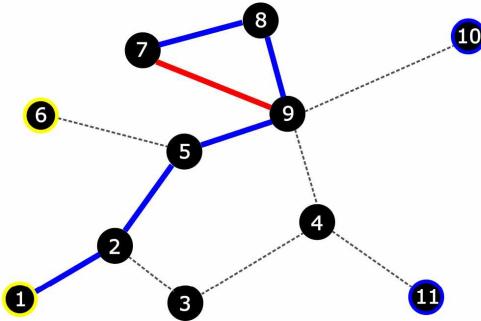


Figura 13 – Erro de loop ao gerar indivíduo

Fonte: Acervo do autor

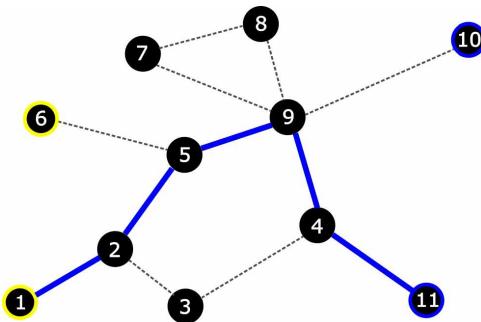


Figura 14 – Correção dos laços

Fonte: Acervo do autor

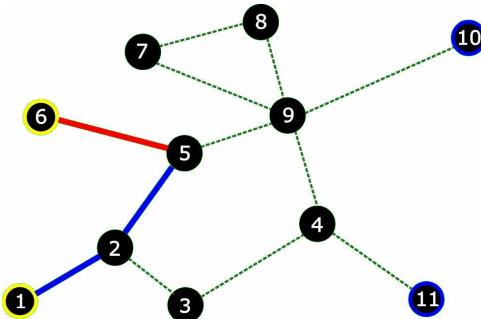


Figura 15 – Erro de caminho sem saída

Fonte: Acervo do autor

Quando o nó sorteado leva a um trajeto inútil, o algoritmo de correção do “caminho sem saída” é acionado. O nó é eliminado e um novo nó é sorteado aleatoriamente, porém o nó excluído não fará mais parte do sorteio. Caso um novo nó não possa ser sorteado, o algoritmo irá retroceder um nó e eliminar aquele que não proporciona uma nova rota, o nó excluído não fará mais parte do sorteio. Esse procedimento se repete até que o destino final seja alcançado.

Como as seleções são aleatórias, existe a possibilidade que os melhores individuo dessa geração não sofram cruzamento, ou, se cruzarem, poderão gerar indivíduos piores. Para

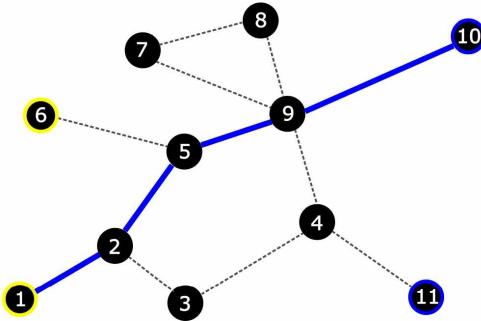


Figura 16 – Correção do erro de caminho sem saída.

Fonte: Acervo do autor

evitar que isso aconteça, existe um algoritmo denominado de elitismo que evita que isso ocorra.

3.6 Elitismo

O elitismo transfere automaticamente os “n” melhores indivíduos para a nova população, antes da aplicação dos operadores de cruzamento e mutação, esse método garante que as melhores soluções da geração atual, persistam na nova geração, acelerando a conversão do algoritmo e evitando a perda de possíveis soluções ótimas, encontradas antes do encerramento do rotina de evolução.

No entanto, o elitismo pode tornar a população homogênea, reduzindo a probabilidade de encontrar uma característica intrínseca de algum indivíduo com baixa aptidão, que, quando combinada com outro indivíduo, produza um indivíduo com aptidão superior a aptidão do melhor indivíduo da população atual. Para resolver esse problema utiliza-se a mutação.

3.7 Mutação

A população deve ser sempre heterogênea para ter características diversificadas que quando combinadas originem soluções diferentes. A mutação deve alterar uma rota de modo que o indivíduo preserve parte das características que possui, impedindo, desse modo, a homogeneização da população.

O método de mutação em busca de rotas consiste na seleção de um nó aleatório, com exceção da origem e do destino. Em seguida, exclui-se toda semi-rota após esse nó. Na sequência, o algoritmo responsável por criar novas rotas é utilizado para completar a rota até o destino.

A Figura 17 mostra a mutação de um indivíduo, onde o gene “4” aleatoriamente é selecionado. Os genes “7” e “8” são eliminados da rota. O indivíduo é preenchido com

genes válidos até encontrar um destino, nesse caso o nó “10”.

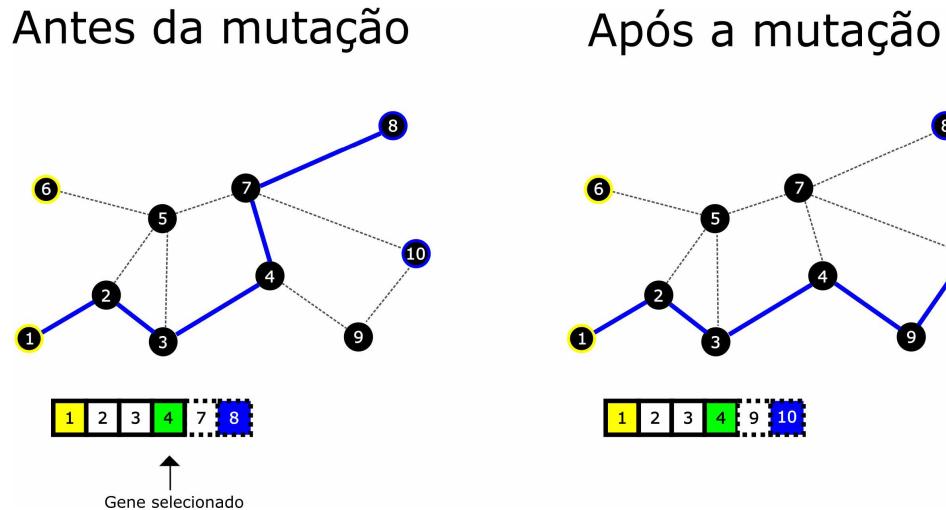


Figura 17 – Exemplo de mutação

Fonte: Acervo do autor

3.8 Critério de parada

O algoritmo genético é executado até que a condição de parada seja atendida. No trabalho proposto, o AG irá evoluir a população até que a condição de estagnação da função de aptidão seja atendida ou o limite de tempo máximo de 5 segundos seja atingido. Esse tempo foi escolhido de modo que o motorista possa ter ciência da nova rota escolhida e possa conduzir o caminhão para uma nova rota em tempo hábil.

Vale salientar que nos testes realizados, o tempo máximo não foi atingido. O AG implementado atingiu a condição de parada por meio da função de aptidão.

A população final é organizada de acordo com a aptidão. O melhor indivíduo da população final é a rota ótima encontrada pelo AG.

CAPÍTULO 4

Sistema Desenvolvido

O AG proposto foi implementado na prática por meio do conjunto de hardware especificado a seguir:

Como mostra a Figura 18, o sistema é composto por três partes: Servidor, IHM e Telemetria.

4.1 Servidor

O Servidor é um computador que sincroniza as informações de todos os embarcados e armazena as informações relevantes. Esse computador não requer um alto poder de processamento como os utilizados em soluções centralizadas de alocação dinâmica de frota. Nesse trabalho um notebook com processador i7 e 8GB de memória foi utilizado como servidor. Esse computador realiza a comunicação entre os embarcados nos caminhões, armazena as informações e sincroniza as informações entre os veículos.

Os caminhões enviam constantemente as suas informação para o servidor, contudo, não recebem informações de outros veículos. Ao entrar em uma cerca eletrônica, o AG deve ser recalculado e para isso o embarcado requer as informações dos demais caminhões. Apenas nesse caso o servidor responde ao embarcado enviando as informações de velocidade, posição, DMT restante, rota dos outros caminhões etc.

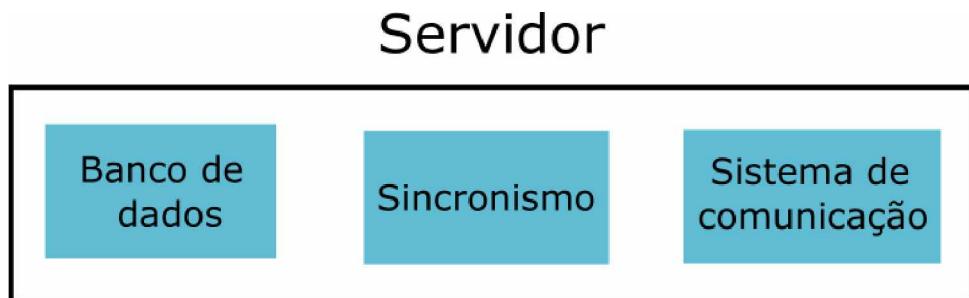


Figura 18 – Diagrama do Sistema

Fonte: Acervo do Autor.

4.2 IHM

A IHM foi implementada por meio de um tablet com o sistema operacional Android 4.4, modelo DELL Venue 7, com 1GB de RAM e processador Intel Atom Z3460 - 1.6 GHz, Dual Core.

Esse tablet executa um aplicativo, cuja função é orientar o usuário sobre qual rota o mesmo deve seguir, além de trocar informações entre usuário, servidor e controladores. A tela *touch screen* facilita a interação do operador com o sistema. Além disso, o tablet já possuem diversos periféricos que podem ser aproveitados para a aplicação, como o GPS por exemplo.

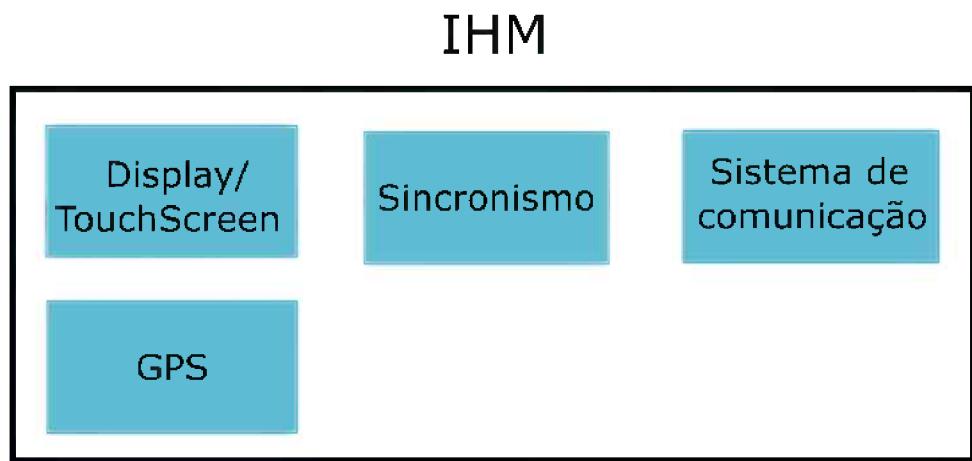


Figura 19 – Diagrama da IHM

Fonte: Acervo do Autor.

O GPS fornece informações de posição do embarcado, e por meio dessa, é calculado a velocidade do veículo. A IHM deve monitorar constantemente a localização obtida pelo GPS e verificar se está contido em uma cerca eletrônica por meio da equação do Haversine. Caso o veículo esteja dentro da cerca, uma requisição de sincronismo é enviada ao servidor e o AG é calculado para selecionar a melhor rota, segundo a condição atual da mina.

O WiFi do tablet permite a conexão com módulos sem fio e estabelece uma comunicação com o servidor por meio da rede.

4.3 Telemetria

A Telemetria utiliza o computador embarcado Beagle Bone Black Rev. C (Figura 20), com o processador AM335x 1GHz ARM® Cortex-A8, memória RAM de 512MB DDR3, executando o Linux (Debian) conectado à IHM por meio de uma rede Wifi interna. A Telemetria têm as seguintes funções:

- Suportar futuramente a adição de sensores;

- ❑ Executar o AG;
- ❑ Reduzir o esforço computacional da IHM;



Figura 20 – Beagle Bone Rev.C, computador embarcado.

Fonte: Acervo do Autor.

A solução desenvolvida deve suportar a adição de sensores afim de aumentar a precisão das informações por meio de sensores externos e sensores originais do caminhão. Um exemplo é a obtenção de informações da rede CAN do caminhão para obter a velocidade do velocímetro do caminhão, que possui um erro muito menor que o GPS.

Telemetria

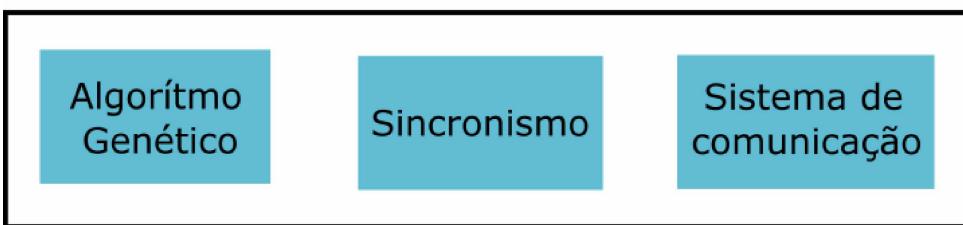


Figura 21 – Diagrama da Telemetria

Fonte: Acervo do Autor.

Outro fator relevante é o tempo de execução do algoritmo. O cálculo do AG tem que ser executado numa janela de tempo estreita. Quanto mais antecipadamente o algoritmo for executado mais tempo o operador terá para tomar as decisões, no entanto, devido a dinâmica da mina, pode ocorrer eventos que alteram a rota preferencial, como a quebra de uma escavadeira, por exemplo. Nesse caso, os pequenos erros podem acumular reduzindo a eficiência do sistema. Portanto o AG deve ser executado o mais rápido possível de modo a sobrar tempo para o motorista tomar ciência da rota e conduzir o caminhão para ela.

4.3.1 Módulo WIFI

O módulo Wifi cria uma rede sem fio local e o conecta à IHM. O computador embarcado da telemetria comunica com o módulo Wifi por meio da porta serial. O protocolo JSON foi utilizado para enviar as mensagens entre a telemetria e a IHM, e entre a IHM e o banco de dados. A padronização das mensagens reduzem os *bugs* na programação e reduz o tempo de implementação dos códigos.

O quadro abaixo mostra um exemplo de mensagem de sincronismo, enviada pelo servidor, com as informações de todos os caminhões e frentes.

Um módulo ESP8266 é utilizado no sistema embarcado que executa o AG para converter um canal de dados serial para UDP e enviá-los diretamente à IHM, sem a necessidade de cabeamento.

```

1
2 { "CM" : [
3     { "TAG": "CM01", "INFO": [ 0,33,35,9,0.472 ] },
4     { "TAG": "CM02", "INFO": [ 1,26,35,2,0.121 ] },
5     { "TAG": "CM03", "INFO": [ 2,37,35,9,0.216 ] }
6 ]
7
8 "FRENTE" : [
9     { "TAG": "1080/M1", "INFO": [ 0, 700, 80,160,35 ] },
10    { "TAG": "1085/M2", "INFO": [ 1,1000,100,115,70 ] },
11    { "TAG": "1140/M1", "INFO": [ 2,1200, 70,130,70 ] }
12 ]
13 }
```

O vetor “INFO” possui as informações relevantes para o cálculo do algoritmo de busca de rota, esse dados respectivamente são:

- Mensagem “CM”:

- ID do veículo:** É o número que identifica o equipamento de transporte;
- Velocidade média:** É a média da velocidade do caminhão ao longo do trajeto;
- Capacidade:** É a quantia de material que é transportado a cada viagem;
- ID do destino:** É o número que identifica o local de carga ou descarga ao fim da rota atual;
- DMT restante:** É a distância em quilômetros até o final da rota.

- Mensagem “FRENTE”:

- ID da frente:** É o número que identifica o local de carregamento;

- ❑ **Taxa de Extração máxima:** É a quantidade máxima de material que pode ser extraído por hora;
- ❑ **Prioridade:** É o percentual da taxa de extração máxima que precisa ser atendida.
- ❑ **Tempo de carregamento:** É o tempo que a escavadeira gasta para carregar um caminhão;
- ❑ **Taxa de extração atual:** É a quantia de material extraído por hora até o momento.

Esse módulo conta com a criptografia WPA2, a mesma utilizada em roteadores. A WAP2 utiliza o protocolo **AES256**(*Advanced Encryption Standard 256 bits*), que dificulta ataques por força bruta, na tentativa de invasão da rede.

4.4 Softwares de Cadastro

O software de cadastro é um formulário onde são registradas as informações sobre os caminhões, as frentes de minério e basculamento. A Figura 22 mostra o grafo da mina utilizada no trabalho. As informações contidas nele foram cadastradas pelo software. O programa de cadastro é organizado em cinco janelas:

- ❑ Adicionar frentes;
- ❑ Adicionar intercessões;
- ❑ Adicionar pontos de basculamento;
- ❑ Adicionar caminhões;
- ❑ Adicionar DMT's;
- ❑ Adicionar caminhos (matriz de rotas).

A tela de cadastro de frente permite editar ou adicionar uma nova frente. Os parâmetros de entrada são o nome ou tag de identificação da frente, a capacidade de extração, a prioridade da frente, o tempo de carregamento e a capacidade extraída.

A tag é um nome de reconhecimento do local, deve ser de conhecimento dos operadores, pois essa será inclusa nas mensagens de orientação do caminho, durante a alocação dinâmica.

A capacidade de extração é um parâmetro definido por engenheiros de minas com base nas informações do fabricante e do material extraído, o valor significa a quantidade máxima de material (em toneladas) que a máquina consegue extrair por hora. Assim como a capacidade de extração a prioridade é definida por especialistas, ela informa a

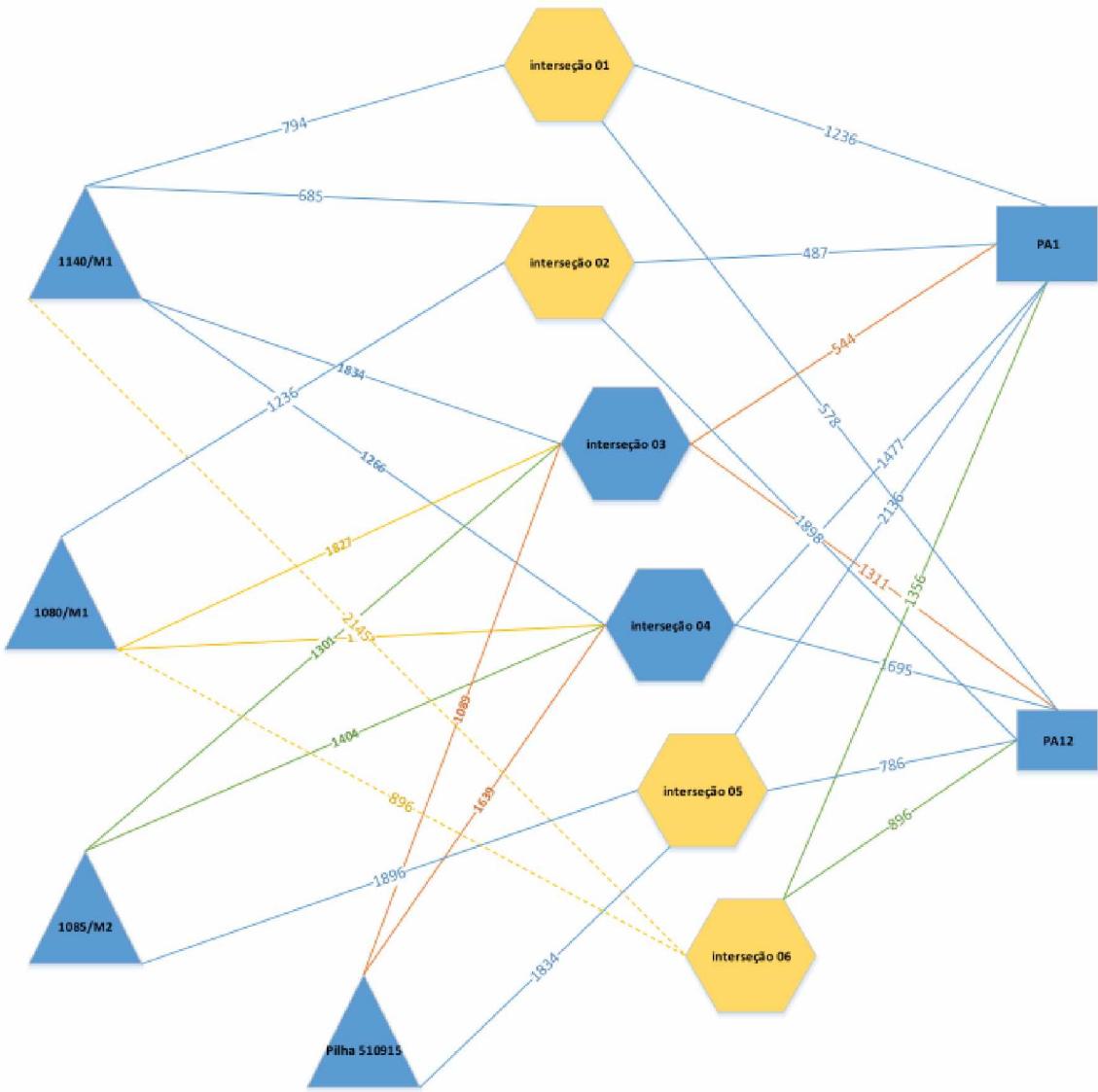


Figura 22 – Grafo da mina

Fonte: Acervo do Autor.

quantidade mínima que deve ser extraída para que a máquina não fique ociosa, esse valor é expresso em porcentagem da capacidade de extração máxima.

O tempo de carregamento é o tempo que uma máquina gasta para carregar um caminhão, esse tempo pode ser influenciado por muitos fatores, como a dificuldade momentânea de extrair o material (uma rocha extensa, por exemplo), pela habilidade do operador em manobrar próximo a máquina, fatores climáticos (chuva) etc. Portanto o tempo de carregamento não é fixo, mas podemos utilizar um tempo médio mais comum a mina sem provocar erros ao sistema, mas o ideal é esse fator ser ajustado por um *feedback* da máquina de extração. A capacidade extraída é a quantidade de material da frente, nessa

The screenshot shows a software application window titled 'Cadastro de dados - versão 0.2.0.6'. At the top, there are several buttons: '+ FRENTE', '+ INTER', '+ DESTINO', '+ CM', '+ DMT', '+ PATH', and 'Gerar Arquivos'. Below these buttons is a table with columns: Tag, Veloc. Media, Capacidade, Frente, DMT Atual, and Rota. The table contains data for CM1, CM2, CM3, and PILHA 510915. A scroll bar is visible on the right side of the table. Below the table, there is another section with columns: Frente, Cap. Extração/Process., Prioridade, T Carreg., Cap. Extraída/Process., and Tipo. It lists various entries like 1080/M1, 1085/M2, 1140/M1, and several intercessões (INTERCESSÃOFO01 to FO05). The 'Frente' column for these entries is empty.

Figura 23 – Software de cadastro

Fonte: Acervo do Autor.

hora.

This screenshot shows a dialog box for creating a 'Frente' (front). The fields include: 'Nome:' (Name) set to '1140/M1', 'Capacidade extração (ton)' (Extraction capacity (ton)) set to '900', 'Prioridade (%)' (Priority (%)) set to '60', 'Tempo de carregamento (s)' (Loading time (s)) set to '65', and 'Capacidade extraída (ton)' (Extracted capacity (ton)) set to '0'. On the right, under 'Tipo:' (Type), there are three radio buttons: 'Origem' (Origin) which is selected, 'Destino' (Destination), and 'Intercessão' (Interception). At the bottom are 'Cancelar' (Cancel) and 'Salvar' (Save) buttons.

Figura 24 – Software de cadastro - Frentes

Fonte: Acervo do Autor.

Para cadastrar um intercessão só é necessário a tag. A tela dos pontos de bascula tem os seguintes parâmetros: Nome, capacidade de processamento, tempo de basculamento e capacidade processada.

O nome é a tag do equipamento (britador) onde o material será depositado. A capaci-

dade de processamento é a quantidade (em toneladas) de minério que o britador consegue processar por hora. O tempo de basculamento é tempo de manobra até depositar o material no britador.

Normalmente esse tempo fica entre 40 e 120 segundos, o tempo de fila não deve ser considerado nesse parâmetro. A capacidade processada é a quantidade de material que já foi, ou está sendo processada nessa hora. O excesso de material no britador ocasiona a quebra do mesmo, e a falta de minério para a industria.

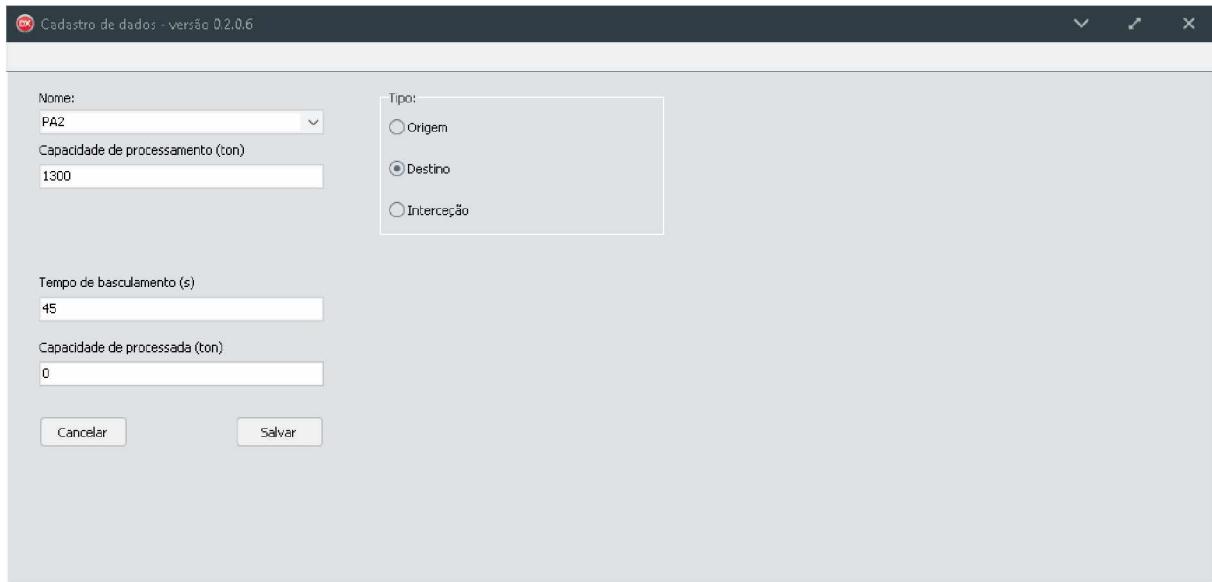


Figura 25 – Software de cadastro - pontos de báscula

Fonte: Acervo do Autor.

A tela de castro de caminhões tem os parâmetros: Nome, velocidade média, capacidade, frente, DMT restante e rota. O nome é a tag, conforme mencionado anteriormente.

A velocidade média é a média da velocidade do caminhão durante o ciclo. Ela pode ser caculada pelo GPS (que foi utilizado no trabalho) ou pelo velocímetro do caminhão, sendo que se a velocidade vier do velocímetro, é necessário que o sistema embarcado esteja conectado ao barramento *Controller Area Network* (CAN) do caminhão.

A capacidade do caminhão está relacionada ao tamanho da caçamba e seus limites nominais são fornecidos pelo fabricante. A frente, ponto de carregamento inicial, assim como a rota, são parâmetros definidos por especialistas, com base em históricos da mina. Mesmo que a rota seja fixa durante o cadastro, isso não impede que seja alterada ao longo da operação.

O cadastro da DMT é uma operação que exige precisão, pois ele será a base para o cálculo da distância entre as rotas. A Figura 27 mostra uma tabela que relaciona os nós com todas as intercessões. A célula formada pela junção da linha com a coluna deve ser preenchida com o valor da distância entre os dois pontos. Os elementos da tabela que

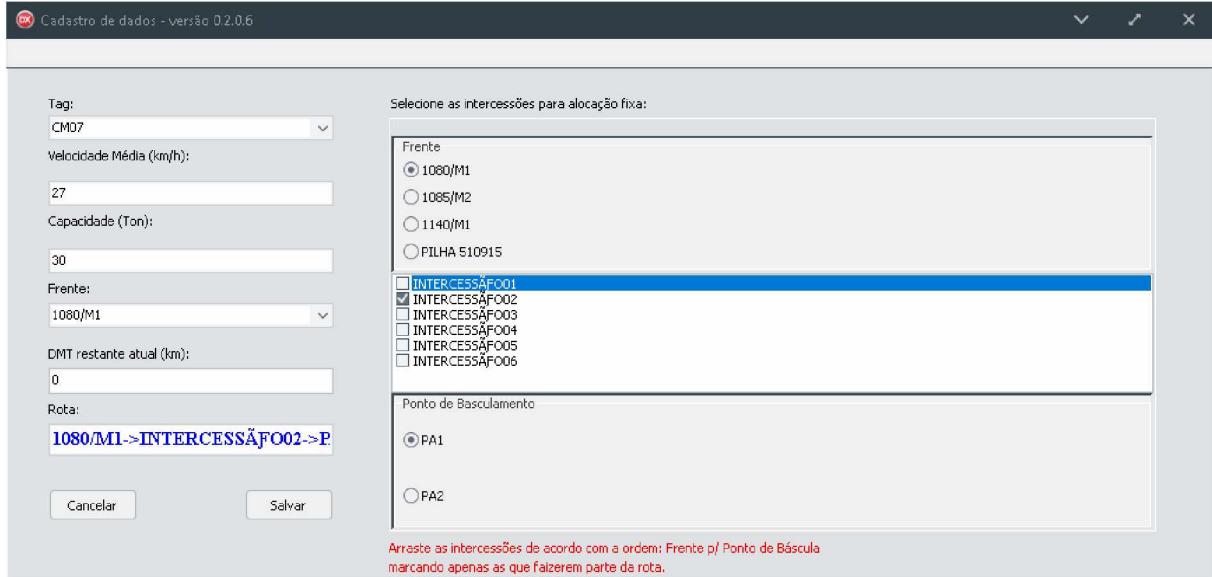


Figura 26 – Software de cadastro

Fonte: Acervo do Autor.

possuem os mesmos nomes nas linhas e colunas (mesmas *tags*) e os que não tem conexão entre si são sempre nulos.

[KM]	INTERCESSÃFO01	INTERCESSÃFO02	INTERCESSÃFO03	INTERCESSÃFO04	INTERCESSÃFO05	INTERCESSÃFO06
1080/M1	0	1,236	1,827	0,945	0	0,896
1085/M2	0	0	1,301	1,404	1,896	0
1140/M1	0,794	0,685	1,834	1,266	0	2,145
PILHA 510915	0	0	1,089	1,639	1,834	0
INTERCESSÃFO01	0	0	0	0	0	0
INTERCESSÃFO02	0	0	0	0	0	0
INTERCESSÃFO03	0	0	0	0	0	0
INTERCESSÃFO04	0	0	0	0	0	0
INTERCESSÃFO05	0	0	0	0	0	0
INTERCESSÃFO06	0	0	0	0	0	0
PA1	1,236	0,487	0,544	1,47	2,136	1,356
PA2	0,578	1,896	1,311	1,695	0,786	0,896

Figura 27 – Software de cadastro

Fonte: Acervo do Autor.

A Figura 28 contém as informações dos caminhos que possuem conexão entre si. Os elementos da tabela são preenchidos com “1” para os casos em que existe conexão e “0” para os casos em que não há ligação.

ORIGEM [col]													
D	[P]	1080/M1	1085/M2	1140/M1	PILHA 510915	INTERCESSÂFO01	INTERCESSÂFO02	INTERCESSÂFO03	INTERCESSÂFO04	INTERCESSÂFO05	INTERCESSÂFO06	PA1	PA2
E	1080/M1	0	0	0	0	0	0	0	0	0	0	0	0
S	1085/M2	0	0	0	0	0	0	0	0	0	0	0	0
T	1140/M1	0	0	0	0	0	0	0	0	0	0	0	0
I	PILHA 510915	0	0	0	0	0	0	0	0	0	0	0	0
N	INTERCESSÂFO01	0	0	0	0	0	0	0	0	0	0	0	0
O	INTERCESSÂFO02	0	0	0	0	0	0	0	0	0	0	0	0
[lin]	INTERCESSÂFO03	0	0	0	0	0	0	0	0	0	0	0	0
	INTERCESSÂFO04	0	0	0	0	0	0	0	0	0	0	0	0
	INTERCESSÂFO05	0	0	0	0	0	0	0	0	0	0	0	0
	INTERCESSÂFO06	0	0	0	0	0	0	0	0	0	0	0	0
	PA1	0	0	0	0	0	0	0	0	0	0	0	0
	PA2	0	0	0	0	0	0	0	0	0	0	0	0

Figura 28 – Software de cadastro

Fonte: Acervo do Autor.

4.4.1 Aplicativo Android

Foi desenvolvido um aplicativo para o tablet. Apesar do Google fornecer as ferramentas de programação nativa do Android gratuitamente, a IDE da Embarcadeiro, Delphi XE10.2, foi escolhida para o desenvolvimento do app.

Essa IDE tem um *framework* denominado Firemonkey que possibilita criar apenas um código e compilar para as plataformas de Android, Ios, Linux, Windows(86x/64x) e Mac, sem que seja necessário realizar alterações no código.

Mesmo que a ferramenta da Embarcadeiro seja paga, a portabilidade do código compensa o custo benefício, pois outras plataforma de hardware podem ser utilizadas como IHM.

O firemonkey suporta a linguagem de programação C++, no entanto a utilizada no app do trabalho foi a pascal, devido ao grau de experiência do autor.

A tela inicial é a área onde o usuário entra com as informações fundamentais no sistema. A chapa é uma número de identificação único que representa cada operador, geralmente fornecido pela empresa, ao efetuar o login o sistema associa o operador ao caminhão, assim toda a produção do caminhão durante a jornada de trabalho será creditada ao operador, e pode ser informada aos gestores na forma de relatório (Figura 29).

A senha é um mecanismo de validação da chapa, se acontecer de um operador digitar a chapa erroneamente, a senha evita que esse entre no sistema, além disso, impede que pessoas não autorizadas entrem no sistema, conforme mostrado na Figura 30.

Durante o período de trabalho a IHM permanece na tela de notificação. Sempre que o servidor envia uma mensagem ou a telemetria envia uma nova rota, aparece uma



Figura 29 – Tela inicial

Fonte: Acervo do Autor.

mensagem na tela e um aviso sonoro é ativado, para alertar o operador. Então, o operador deve realizar um clique na tela para confirmar o recebimento da mensagem, como mostra a Figura 31, "Siga pra PA1, Rota 1080/M1''.

A extração de minérios, em grande parte, ocorre por 24 horas no dia. Por isso, foi adicionado ao software o recurso de brilho automático. Esse recurso ajusta o brilho da tela. O brilho com valor elevado facilita a leitura durante o dia, mas a noite incomoda quem está dirigindo. Assim, todas as telas coloridas tornam-se pretas, ficando branco apenas as mensagens que são apagadas após a confirmação do operador, como mostrado na Figura 32.

4.5 Simulação

Simulação é o processo de criação de um modelo dinâmico real com o objetivo de compreender o comportamento do sistema ou de avaliar várias estratégias (dentro dos limites impostos por um critério ou conjunto de critérios) para a operação do sistema Ingalls (2011).

A simulação é um recurso usado em outros trabalhos nessa área como o Tan e Takakuwa (2016), que utiliza a plataforma de simulação Arena para simular o despacho de caminhões em uma mina. A aplicação prática da solução proposta só é possível após a validação do sistema e negociação com a empresa cliente, todo esse processo leva tempo para ser aprovado, logo a simulação é uma abordagem comum para análise da solução

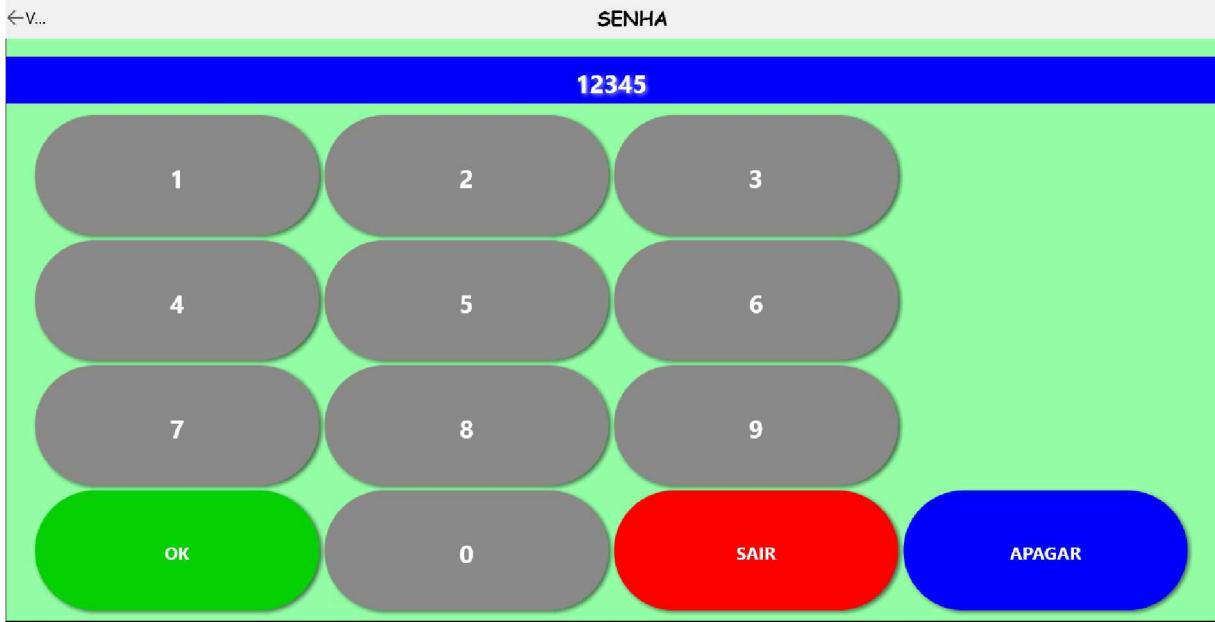


Figura 30 – Senha de proteção

Fonte: Acervo do Autor.

proposta.

A validação do hardware também faz parte do trabalho, por isso, o simulador deve ter uma interface de comunicação com o kit de hardware embarcado (IHM e telemetria). Por motivos de economia de custos de desenvolvimento, utilizou-se apenas um kit de hardware embarcado, os demais são simulados, assim como a mina e o servidor.

Existem alguns software de simuladores no mercado que atenderiam, como o *Arena Simulation*. No entanto, o desenvolvimento de um simulador facilita a customização e adéqua o software ao hardware.

No sistema real de uma mineradora os caminhões inciam sua jornada descarregados, porque a contagem de viagens apenas é incrementada após o basculamento, ou seja, o operador que não bascular antes de terminar seu turno, perderá a ultima viagem e ela será creditada ao seu substituto do próximo turno.

O algoritmo do simulador considera que inicialmente todos os caminhões estejam descarregados e tenham uma rota predefinida pelo engenheiro de mina responsável. O operador continuará dirigindo enquanto o sistema verifica se ele está em uma área de cerca eletrônica, conforme a Figura 33, em caso negativo o operador prosseguirá em seu trajeto.

Ao chegar na área da interseção, uma rotina de verificação da condição do caminhão é iniciada (Figura 34), pois as cercas eletrônicas podem estar presentes em frentes de extração, pontos de alimentação e interseções. Cada tipo de local tem suas rotinas específicas.

De acordo com o fluxograma, como o caminhão está vazio, ele segue para o ponto “B”. Em seguida executa o AG para buscar pela melhor rota (Figura 35). O resultado pode

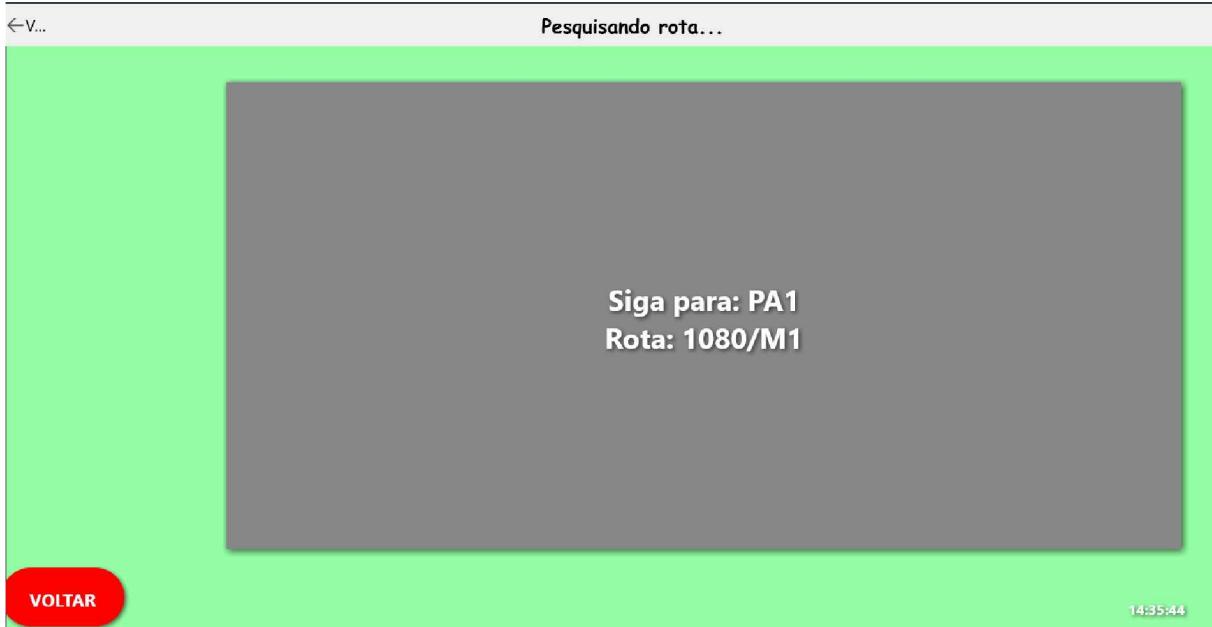


Figura 31 – Notificação de alteração de rota

Fonte: Acervo do Autor.

ser diferente da rota anterior, portanto, nesse caso, uma notificação alerta o operador, indicando uma nova rota preferencial disponível. Por outro lado, se a rota for a mesma, não há motivos para notificar o operador, ele permanecerá em sua trajetória atual e o sistema retorná para a verificação de cercas eletrônicas.

Continuando sua rota, o operador entrará na cerca eletrônica da frente de extração e o sistema confirma que o local é do tipo “frente”. O operador deverá informar a condição da escavadeira que está trabalhando nesse local, se essa estiver disponível, ele deverá manobrar para prosseguir com o carregamento, todavia, pode acontecer da máquina estar ocupado carregando outro caminhão que chegou anteriormente. Nessa condição, esse operador deve esperar em fila, até chegar sua vez de carregar.

Ao fim do carregamento o sistema executa o AG buscando uma nova rota. Assim como na interseção, o operador só será notificado se a nova rota for diferente da sua anterior. Posteriormente o operador entrá na região do ponto de alimentação e verificará se o equipamento está disponível ou se ele precisará aguardar em fila.

Após basculamento do caminhão, o PA contabiliza a quantidade de material depositado à produção total e uma viagem é creditada ao operador. Em seguida, o AG busca uma nova rota e fecha o ciclo.

Visualmente os caminhos são representados no formato de grafo, com objetivo de simplificar e facilitar o entendimento do processo. No lado esquerdo é representado as frentes de extração de minério e a direita os pontos de alimentação. No centro os pontos representam bifurcações no trajeto (Figura 36).

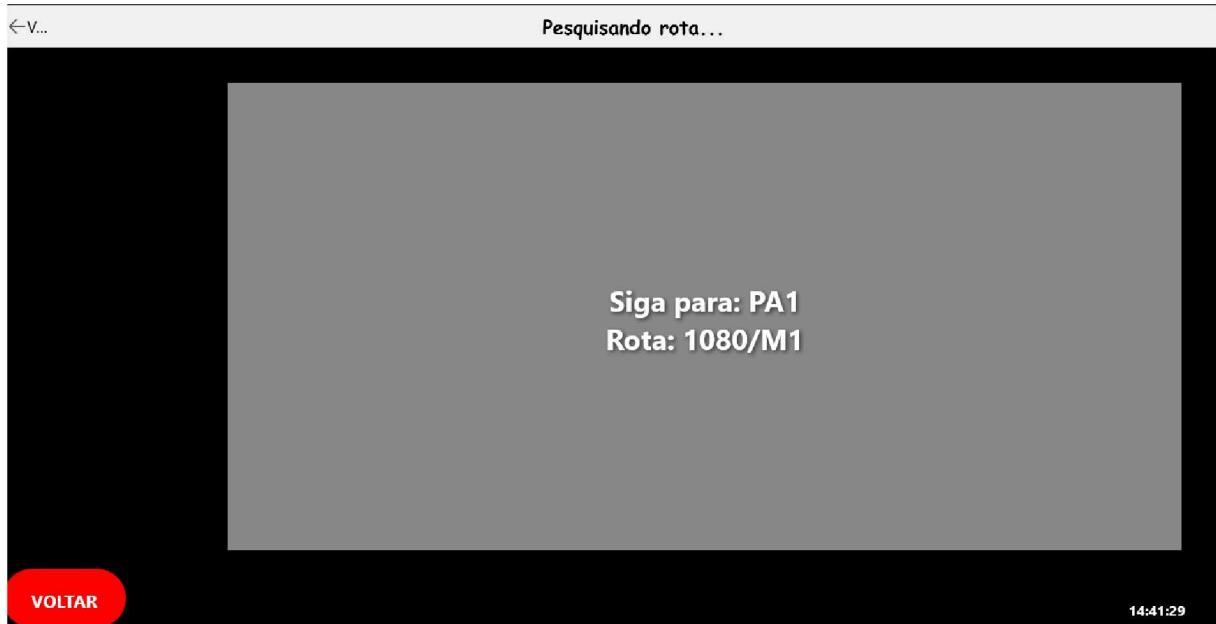


Figura 32 – Tela de notificação em modo noturno

Fonte: Acervo do Autor.

A simulação foi criada com base em dados reais de uma mineradora local e informações de especialistas em mineração, com o objetivo de tornar a simulação o mais próximo da realidade.

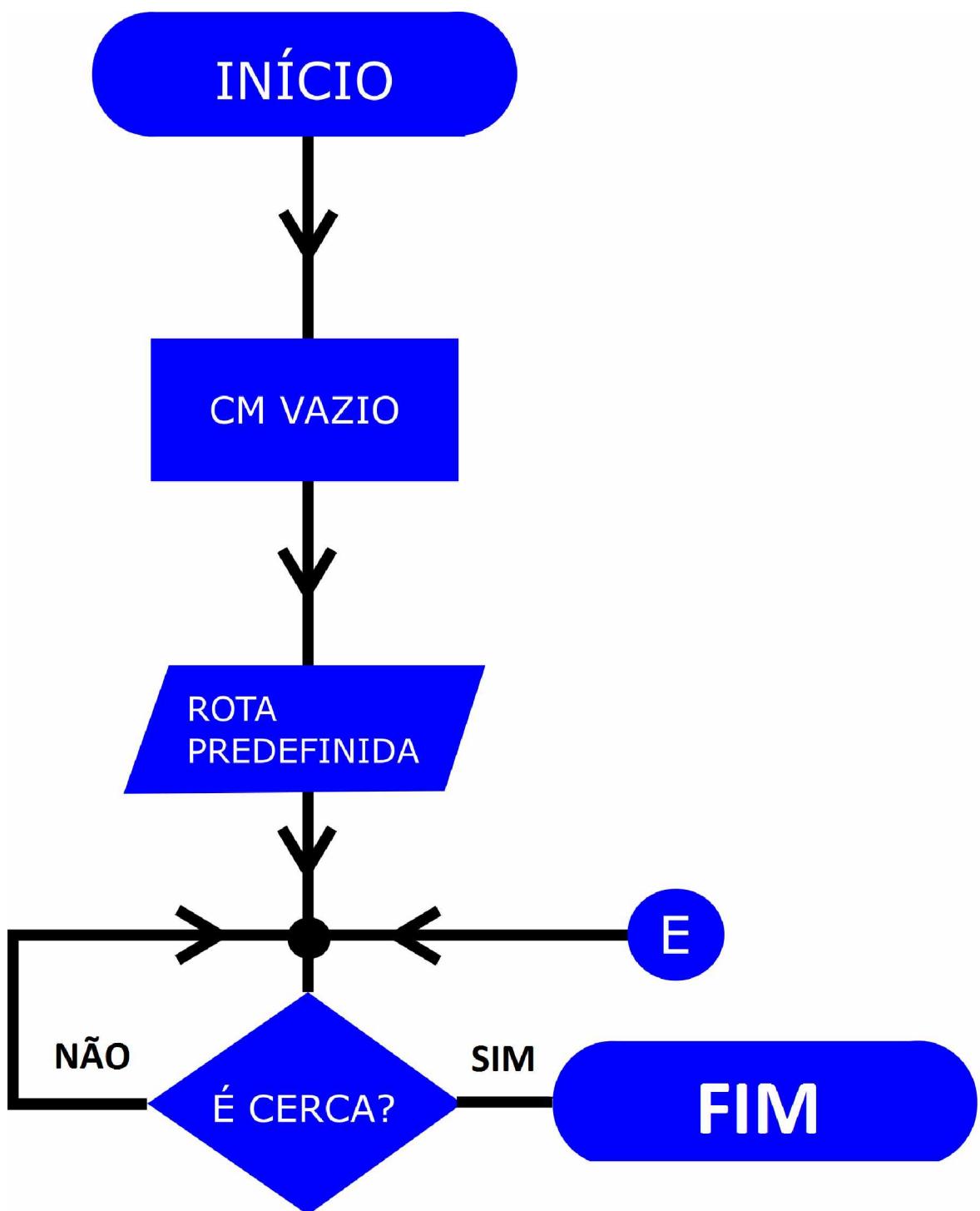


Figura 33 – Fluxograma da rotina inicial

Fonte: Acervo do Autor.

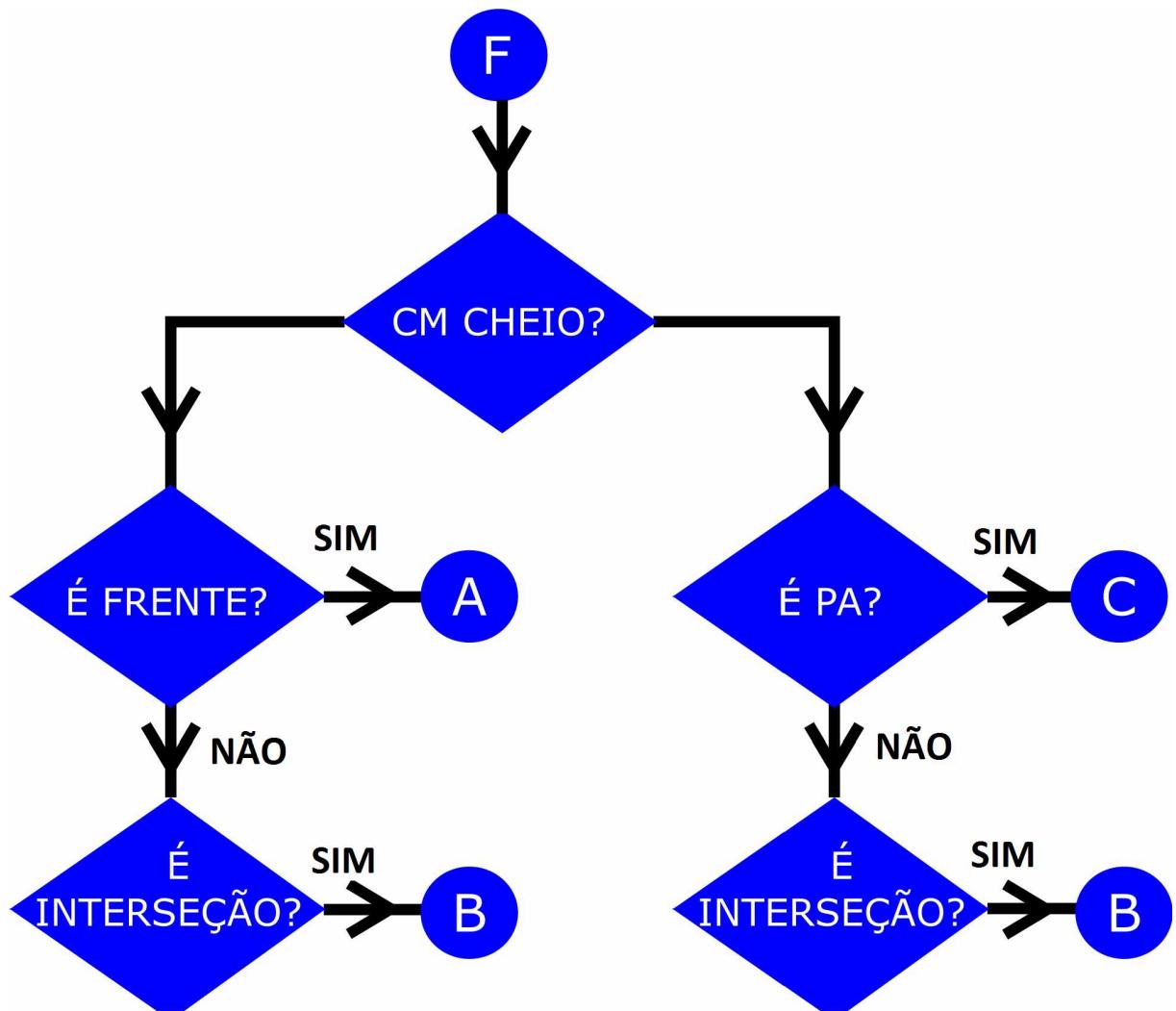


Figura 34 – Fluxograma da rotina cerca eletrônica

Fonte: Acervo do Autor.

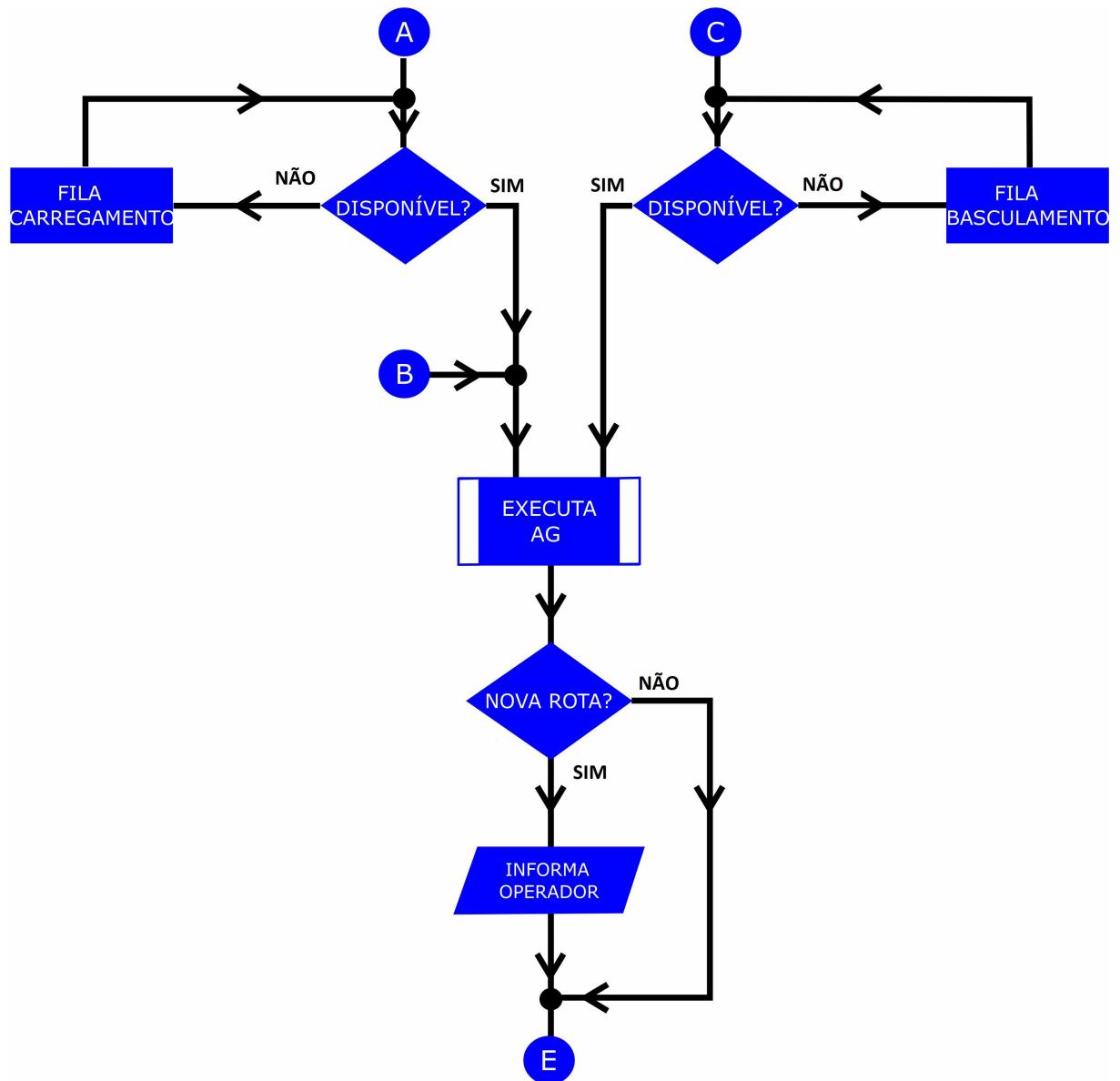


Figura 35 – Fluxograma da rotina AG

Fonte: Acervo do Autor.

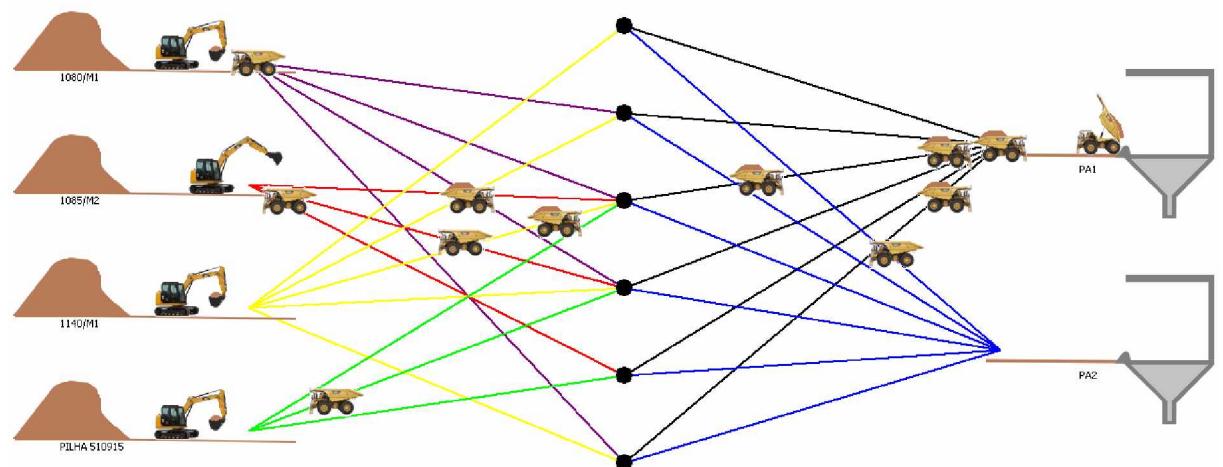


Figura 36 – Simulador desenvolvido para a validação

Fonte: Acervo do Autor.

CAPÍTULO 5

Resultados

A alocação dinâmica de rotas de vários veículos é um problema complexo porque vários caminhos levam ao mesmo destino e diversos caminhos devem ser traçados ao mesmo tempo de modo a atender aos vários veículos.

5.1 Alocacão Fixa e Alocacão Dinâmica

Para avaliar a eficiência do AG no sentido de evitar a formação de filas, traçar as melhores rotas e, consequentemente, elevar a produção, um senário com 9 caminhões, 4 frentes de minério e 2 pontos de basculamento foram configurados no software de simulação. Três simulações foram realizadas durante 6 horas cada, onde foram priorizados cada fator separadamente.

Os resultados são comparados a uma simulação de mesmo período com alocação fixa, mantendo-se durante todo o período de simulação as mesmas rotas definidas no início.

5.1.1 Simulação Específica - Redução de distância, ociosidade e fila

Foram realizadas três simulações. Na primeira simulação utilizou-se a função objetivo do AG apenas para otimizar a distância de transporte. Na segunda simulação a função objetivo foi configurada para reduzir a ociosidade das máquinas. Na terceira simulação a meta foi reduzir o tempo de fila.

A segunda coluna da Tabela 3 mostra a busca de rotas com foco na redução da distância percorrida. Os resultados mostram um aumento no tempo de fila, isso porque a rota mais curta é a mais viável nesse caso. O algoritmo direcionou todos os caminhões para a mesma rota, isso provocou uma perda de produção em fila superior aos outros métodos.

A terceira coluna da Tabela 3 exibe a busca de rotas com foco na redução da ociosidade das máquinas de carregamento. O AG deu preferêncià às frentes de minério que estão

Tabela 3 – A atraso em fila com objetivos separados

TEMPO EM FILA (segundos)			
TAG	DISTÂNCIA DE TRANSPORTE	OCIOSIDADE DE MÁQUINA	ATRASO EM FILA
CM1	7390	560	50
CM2	6370	520	40
CM3	6900	290	40
CM4	6190	450	40
CM5	5970	210	80
CM6	7080	210	40
CM7	4880	730	10
CM8	7140	300	10
CM9	6360	540	40
TOTAL	58280 (16h)	3810 (1h)	350 (5min)

a mais tempo sem operar, isso provocou inicialmente o rodízio de caminhões entre as frentes, mas após atingir o equilíbrio o algoritmo força os caminhões ao “comportamento de manada”, eles passam a atender as frentes em grupos, o que consequentemente aumenta o tempo em fila.

A quarta coluna da Tabela 3 é relativo a busca de rotas com foco na redução do tempo em fila. Nesse caso, o tempo em fila foi inferior aos outros, todavia, o tempo não foi nulo. Isso ocorre porque não há como não evitar a formação de filas. Nessa situação, o AG tem que encontrar um meio de evitar a formação de filas.

Tabela 4 – Produção por caminhão com objetivos separados

PRODUÇÃO POR CAMINHÃO (toneladas)			
TAG	DISTÂNCIA DE TRANSPORTE	OCIOSIDADE DE MÁQUINA	ATRASO EM FILA
CM1	840	750	720
CM2	780	690	720
CM3	810	630	720
CM4	780	600	600
CM5	810	720	690
CM6	810	720	750
CM7	810	660	630
CM8	780	630	690
CM9	810	690	720
TOTAL	7230	6090	6240

A Tabela 4 mostra a produção obtida em cada frente. A produção que visou a busca de rotas com foco na redução da distância, foi maior que as demais. Isso ocorreu devido a formação de uma fila de tamanho constante, na mesma frente, torna o carregamento contínuo. No entanto, isso torna as outras máquinas ociosas.

Para os dois outros métodos a produção são de valores próximos, com uma pequena redução quando a produção teve como objetivo minimizar a ociosidade das máquinas.

5.1.2 Simulação Completa - Alocação Fixa e Dinâmica

Na simulação completa, foram realizadas duas simulações com os mesmos parâmetros referentes ao número de caminhões, frentes de minério e pontos de basculamento utilizados na Simulação Específica.

A primeira simulação foi realizada utilizando-se de alocação fixa e a segunda com o uso do AG proposto configurado para selecionar a melhor rota, visando redução de fila, ociosidade das máquinas e redução da distância de transporte. Em seguida temos a comparação dos resultados encontrados:

Tabela 5 – Tempo perdido em fila

TEMPO E FILA (S)		
TAG	FIXA	DINÂMICA
CM1	120	130
CM2	0	60
CM3	80	250
CM4	0	50
CM5	310	190
CM6	310	170
CM7	240	20
CM8	10	10
CM9	30	240
TOTAL	1100 (18min 20s)	1120 (18min 40s)

Tabela 6 – Produção dos caminhões: Alocação fixa vs dinâmica (multiobjetivo)

PRODUÇÃO POR CAMINHÃO (TON)		
TAG	FIXA	DINÂMICA
CM1	930	1020
CM2	630	510
CM3	510	1020
CM4	450	900
CM5	750	930
CM6	660	960
CM7	1080	840
CM8	480	990
CM9	750	900
TOTAL	6240	8070

A Tabela 6 apresenta os resultados do ponto de vista da produção. Os resultados mostram que apesar do tempo em fila (tabela 5) ser aproximadamente o mesmo, com

relação a alocação fixa, houve um ganho na produção de 30%. Esse ganho está relacionado a otimização das distâncias das rotas, ao rodízio dos caminhões entre as máquinas de carregamento e a utilização de rotas que levam a filas menores. Todos esses fatores juntos aumentam o número de viagens por caminhão e, consequentemente, aumentam a produção, comprovando a eficácia do AG proposto elevando a produção de uma mina.

A Tabela 5 apresenta a comparação dos tempos em fila para alocação fixa e dinâmica. Observa-se que quase não há diferença nos tempos de fila com relação a alocação fixa, a diferença é menor que 1.8%.

Uma terceira análise consiste em simular no mesmo cenário, diferentes quantidades de caminhões, variando de 1 a 9, e obtendo a produção total.

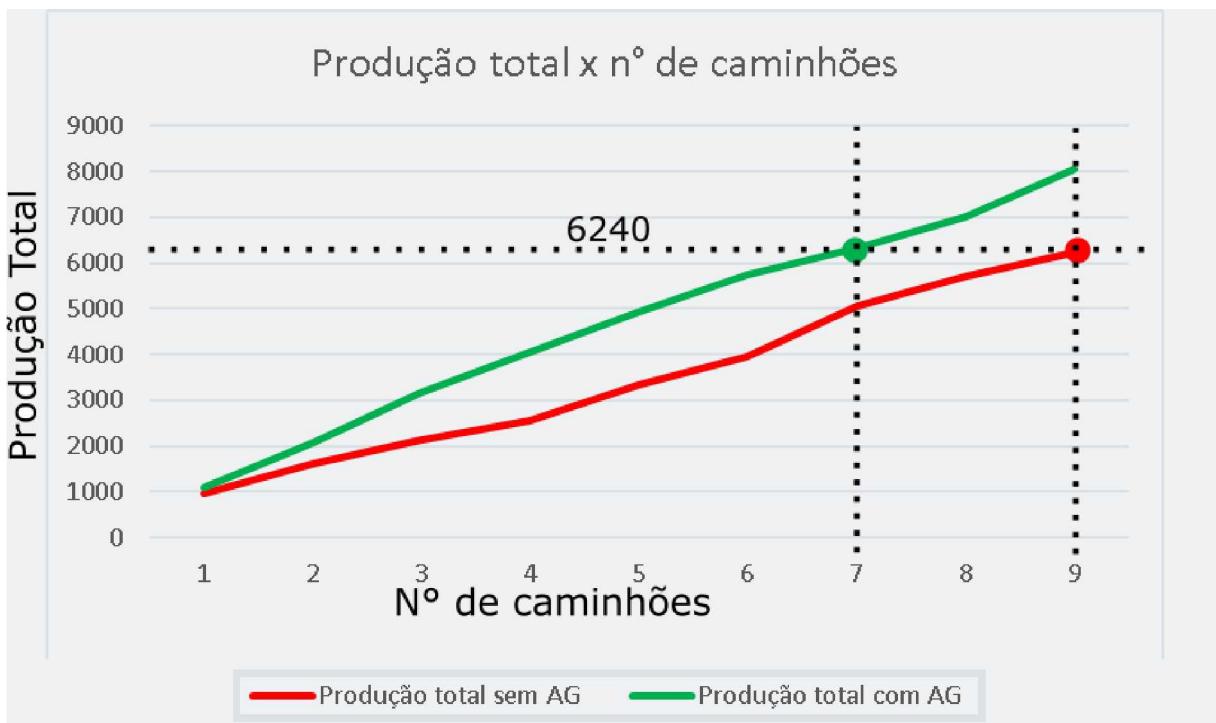


Figura 37 – Gráfico de Produção Total x N° de caminhões

Fonte: Acervo do Autor.

A Figura 37 mostra o gráfico da produção por número de caminhões. Observando os valores nota-se que com 7 caminhões e a alocação dinâmica por AG, produz a mesma quantidade que 9 caminhões por alocação fixa.

5.2 Desempenho: Análise do Algoritmo Embarcado

A validação do hardware foi realizada medindo-se o tempo de execução do AG, no computador embarcado e comparando o resultado obtido com o tempo de execução do algoritmo no servidor utilizado. A medição foi obtida por meio da função *clock()* da bibli-

oteca *Timer.h* Manteve-se os parâmetros de 9 caminhões, 4 frentes de minério e 2 pontos de basculamento, comuns em uma mina de pequeno porte. Ademais, uma população de 30 indivíduos e 60 gerações foram utilizadas. Os tempos encontrados consistem de uma média de 100 repetições.

Tabela 7 – Tempo de execução do AG

Placa	Sistema Operacional	Clock (GHz)	Tempo Total (ms)
Beagle Bone Black RevC	Linux Debian (ARM)	1	37
Notebook I7 8GB	Windows 10 Pro (x64)	2.4	12

A Tabela 7 mostra que apesar do notebook ter um hardware com poder de processamento aproximadamente 60 vezes (130.000 MIPS/2000 MIPS) superior à Beagle Bone, os resultados obtidos são próximos. Isso ocorre porque o Notebook executa muitos outros processos em *background* no sistema operacional Windows 10.

O sistema embardado, por outro lado, é dedicado a tarefa de executar o AG proposto. Por isso, é possível conseguir uma solução mais simples e econômica. Além disso, foi contatado que o tempo gasto para executar o algoritmo proposto é menor que o tempo de reação do operador do caminhão. Isso significa que o tempo de processamento não vai atrapalhar a tomada de decisão do motorista com mensagens atrasadas, atendendo um dos requisitos principais para a instalação em mineradoras.

Tabela 8 – Custo dos materiais utilizados

Placa	Custo (R\$)
Beagle Bone Black RevC	400
Notebook I7 8GB	2700
Tablet Dell Venue 7	800
Módulo Wifi ESP8266	40

A Tabela 8 mostra o custo dos equipamentos envolvidos no trabalho.

CAPÍTULO 6

Discussão e Conclusões

Este trabalho consiste na implementação de um AG que utilizou dados de uma mina real, para encontrar rotas otimizadas em um ambiente de mineração. Os resultados comprovaram que o uso do AG proposto configurado para otimizar, filas, distância e melhor alocação de caminhões para atender as escavadeiras utilizadas na extração e carregamento do minério aumentou em 30% a produção da mina. O minério de ferro processado custa em torno de U\$50 a tonelada, enquanto o nióbio custa U\$45 mil a tonelada, portanto isso representará uma grande economia para as empresas mineradoras.

A comunicação wifi consegue manter uma proteção segura com a criptografia WPA2, mas para melhora a segurança do sistema deve ser criado uma criptografia adicional nos pacotes enviados via rede.

O uso da solução proposta reduz a quantidade de caminhões necessários para manter a mesma produção de alocação fixa, isso significa que a empresa minerado pode reduz sua frota e cortar os custos envolvendo os caminhões, como consumo de combustível e manutenção.

O algoritmo pode ser configurado para atender a situações cujo objetivo é a redução de filas, otimização da alocação de caminhões ou redução das rotas, simplesmente com a troca da função objetivo.

O hardware atendeu as expectativas e comprovou executar o algoritmo dentro da janela de tempo necessário para que o motorista possa obter a informação e redirecionar o caminhão para a rota informada.

Na prática, o motorista tem total controle sobre o caminhão, podendo escolher seguir ou não a rota sugerida pelo software, o que pode influenciar nos resultados obtidos. Nessas condições a empresa deve treinar os operadores, demonstrando a importância de seguir as sugestões do software.

A solução apresentada será adaptada a equipamentos específicos para operar em campo, isto é, que possuem proteções de alimentação, choque físico, queda, calor e outros requisitos necessários para operarem em um ambiente de mineração.

Referências

- AHN, C. W.; RAMAKRISHNA, R. S. A genetic algorithm for shortest path routing problem and the sizing of populations. **IEEE Transactions on Evolutionary Computation**, v. 6, n. 6, p. 566–579, Dec 2002. ISSN 1089-778X.
- ALAM, C. N. et al. Implementation of haversine formula for counting event visitor in the radius based on android application. In: **2016 4th International Conference on Cyber and IT Service Management**. [S.l.: s.n.], 2016. p. 1–6.
- ALVARENGA, G. B. Despacho ótimo de caminhões numa mineração de ferro utilizando algoritmo genético com processamento paralelo. **Belo Horizonte, MG, Brasil: Master's thesis**, 1997.
- BONATES, E.; LIZOTTE, Y. A computer simulation model to evaluate the effect of dispatching. **Reclamation and Environment**, p. 99–104, 02 1988.
- CHAKRABORTY, B. Simultaneous search for multiple routes using genetic algorithm. In: **2004 IEEE International Conference on Computational Intelligence for Measurement Systems and Applications, 2004. CIMSA**. [S.l.: s.n.], 2004. p. 77–80.
- COSTA, F.; SOUZA, M.; PINTO, L. Um modelo de alocação dinâmica de caminhões visando ao atendimento de metas de produção e qualidade. In: **Anais do III Congresso Brasileiro de Mina a Céu Aberto e III Congresso Brasileiro de Mina Subterrânea**. [S.l.: s.n.], 2004.
- COX, W. et al. A genetic algorithm for truck dispatching in mining. **GCAI 2017. 3rd Global Conference on Artificial Intelligence**, 2017.
- DARWIN, C. **On the Origin of Species by Means of Natural Selection**. London: Murray, 1859. Or the Preservation of Favored Races in the Struggle for Life.
- GOLDBERG, D. E. Genetic algorithms in search, optimization, and machine learning, 1989. **Reading: Addison-Wesley**, 1989.
- HARTMAN, H. L.; MUTMANSKY, J. M. **Introductory mining engineering**. [S.l.]: John Wiley & Sons, 2002.
- HE, M.-X. et al. The genetic algorithm for truck dispatching problems in surface mine. **Information Technology Journal**, v. 9, n. 4, p. 710–714, 2010.

- HEINEN, M. R.; OSÓRIO, F. S. Algoritmos genéticos aplicados ao problema de roteamento de veículos. **HÍFEN**, v. 30, n. 58, 2006.
- HJORRING, C.; HOLT, J. New optimality cuts for a single?vehicle stochastic routing problem. **Annals of Operations Research**, v. 86, n. 0, p. 569–584, Jan 1999. ISSN 1572-9338. Disponível em: <<https://doi.org/10.1023/A:1018995927636>>.
- HOLLAND, J. H. **Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence**. [S.l.]: The MIT Press, 1992. ISBN 0262581116,9780262581110.
- INAGAKI, J.; HASEYAMA, M.; KITAJIMA, H. A genetic algorithm for determining multiple routes and its applications. In: **Circuits and Systems, 1999. ISCAS '99. Proceedings of the 1999 IEEE International Symposium on**. [S.l.: s.n.], 1999. v. 6, p. 137–140 vol.6.
- INGALLS, R. G. Introduction to simulation. In: **Proceedings of the 2011 Winter Simulation Conference (WSC)**. [S.l.: s.n.], 2011. p. 1374–1388. ISSN 0891-7736.
- INOUE, Y. **Exploration method of various routes with genetic algorithm**. Tese (Doutorado) — Master's Thesis, Information System Engineering, Kochi Institute of Technology, 2001.(in Japanese), 2001.
- KANOH, H.; NAKAMURA, T. Knowledge based genetic algorithm for dynamic route selection. In: **Knowledge-Based Intelligent Engineering Systems and Allied Technologies, 2000. Proceedings. Fourth International Conference on**. [S.l.: s.n.], 2000. v. 2, p. 616–619 vol.2.
- KURNIAWAN, R.; SULISTIYO, M. D.; WULANDARI, G. S. Genetic algorithm for capacitated vehicle routing problem with considering traffic density. In: **2015 International Conference on Information Technology Systems and Innovation (ICITSI)**. [S.l.: s.n.], 2015. p. 1–6.
- LI, Y.; HE, R.; GUO, Y. Faster genetic algorithm for network paths. In: **The 6th International Symposium on Operations Research and Its Applications,(ISORA'06). August**. [S.l.: s.n.], 2006. p. 8–2.
- MARIN, T. **Impacto da variabilidade operacional na execução do plano de lavra**. Tese (Doutorado) — Universidade de São Paulo, 2009.
- MELO, W. B. d. et al. Otimização de despacho de veículos utilizando programação linear. **Brazil Automation ISA**, 2013.
- MITCHELL, M. **An introduction to genetic algorithms**. [S.l.: s.n.], 1998.
- NICOLAE, M.; LUCACI, L.; MOISE, I. Embedding android devices in automation systems. In: **2013 IEEE 19th International Symposium for Design and Technology in Electronic Packaging (SIITME)**. [S.l.: s.n.], 2013. p. 215–218.
- SCHULZE, M. A. Linear programming for optimization. 1998.
- SOUZA, M. et al. A hybrid heuristic algorithm for the open-pit-mining operational planning problem. **European Journal of Operational Research**, v. 207, n. 2, p. 1041 – 1051, 2010. ISSN 0377-2217. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0377221710003875>>.

- TAN, Y.; TAKAKUWA, S. A practical simulation approach for an effective truck dispatching system of open pit mines using vba. In: **2016 Winter Simulation Conference (WSC)**. [S.l.: s.n.], 2016. p. 2394–2405.
- TANAKA, M. et al. Multiroute planning for aircraft using genetic algorithms. **Systems and Computers in Japan**, Wiley Subscription Services, Inc., A Wiley Company, v. 35, n. 2, p. 39–48, 2004. ISSN 1520-684X. Disponível em: <<http://dx.doi.org/10.1002/scj-1.10428>>.
- WARDLAW, R.; BHAKTIKUL, K. Comparison of genetic algorithm and linear programming approaches for lateral canal scheduling. v. 130, 08 2004.
- WHITE, J.; OLSON, J. Computer-based dispatching in mines with concurrent operating objectives. **Min. Eng. (Littleton, Colo.); (United States)**, v. 38:11, Nov 1986.