

Lucas Teles Agostinho
Rodrigo Mendonça da Paixão

Algoritmos Genéticos Aplicados no Problema de Roteirização de Veículos com Janela de Tempo

São Paulo – Brasil

2017

Lucas Teles Agostinho
Rodrigo Mendonça da Paixão

Algoritmos Genéticos Aplicados no Problema de Roteirização de Veículos com Janela de Tempo

Centro Universitário Senac
Bacharelado em Ciência da Computação

Orientador: Eduardo Heredia

São Paulo – Brasil
2017

Lista de abreviaturas e siglas

AG	Algoritmos Genéticos
API	Application Programming Interface
IA	Inteligência Artificial
CPU	Central Processing Unit
PCV	Problema do Caixeiro Viajante
PRV	Problema de Roteirização de Veículos
PRVJT	Problema de Roteirização de Veículos com Janela de Tempo
TS	Têmpera Simulada

Sumário

1	INTRODUÇÃO	5
1.1	Motivação	5
1.2	Objetivos	7
1.2.1	Objetivos Específicos	7
1.3	Método de trabalho	7
1.4	Organização do trabalho	7
2	REVISÃO DE LITERATURA	9
2.1	Roteamento de Veículos com Janelas de Tempo	9
2.1.1	Formulação matemática	9
2.1.2	Complexidade	11
2.1.3	Heurísticas	12
2.1.3.1	Construção de rotas	12
2.1.3.2	Aprimoramento de rotas	13
2.1.3.3	Estruturas de vizinhança	14
2.1.4	Meta-heurísticas	15
2.1.4.1	Busca Tabu (Tabu Search)	16
2.1.4.2	Têmpera Simulada (Simulated Annealing)	16
2.1.4.3	Busca Local com Múltiplos Pontos Iniciais (Multi-Start Local Search)	16
2.1.4.4	Algoritmos Genéticos / Evolutivos	16
2.2	Algoritmos genéticos	18
2.2.1	Funcionamento	18
2.2.2	Inicialização	19
2.2.3	Avaliação	19
2.2.4	Seleção	19
2.2.5	Cruzamento	20
2.2.6	Mutação	20
2.2.7	Finalização	20
2.2.8	Aplicações	22
3	METODOLOGIA	25
4	IMPLEMENTAÇÃO	30
4.1	Tecnologias	30
4.2	Estrutura do Projeto	30
4.2.1	Calculador de Rotas - CalcRoute	30

4.2.2	Gerador de Dados - CalcRoute.RouteGenerate	30
4.2.3	API de Calculo de Rotas - CalcRoute.API	31
4.2.4	Interface para comunicação com a API	31
5	CONCLUSÃO	34
5.1	Limitações	34
5.2	Roteiros dos Testes	35
5.3	Comparativo	45
5.4	Trabalhos futuros	46
	REFERÊNCIAS	47

1 Introdução

No meio empresarial é comum pensar em logística, essa é a área que gerencia os recursos, matérias-primas, componentes, equipamentos, serviços, informações necessárias para execução e controle das atividades da empresa. Ela tem como foco orquestrar esses itens de forma a encontrar melhores condições de operação.

Uma atividade da logística que envolve grande valor de negócio é a distribuição eficiente de mercadorias ou serviços. Os custos de transporte representam uma parcela significativa do preço de muitos produtos. (DIAS, 2010)

1.1 Motivação

Com o crescimento populacional, a descentralização dos pontos de venda e o aumento da variedade de produtos tem provocado o crescimento da complexidade da rede de distribuição de bens e serviços. O valor da logística se manifesta em termos de tempo e lugar, ou seja, disponibilidade. Sendo assim eles não têm valor, a menos que estejam em poder dos clientes quando (tempo) e onde (lugar) eles pretendem consumi-los (TSUDA, 2007).

As decisões de transporte são parte fundamental da estratégia e planejamento logístico, com destaque para o problema de roteirização veículos (PRV), isso porque o transporte pode representar de um a dois terços dos custos logísticos totais (BALLOU, 1973; RODRIGUES, 2007).

O PRV é definido de forma que a partir de um conjunto de rotas que será percorrido por uma quantidade N veículos onde cada rota começa e termina em um depósito, e todos os endereços serão visitados somente uma vez.

A identificação da ordem dos destinos, quando há um número elevado de endereços, se torna complexa por se tratar de um problema combinatório, onde é preciso avaliar todas as combinações para encontrar uma rota de menor tempo e distância (KARP, 1975).

Sendo assim PRV pode exigir um alto esforço computacional, pertencendo a classe dos problemas NP-difíceis, não pode ser solucionado em tempo polinomial, sendo uma forma de combinação da solução do problema do Caixeiro Viajante e do Problema da Mochila (OLIVEIRA, 2005b), por isso torna-se importante uma boa escolha do método a ser usado para sua solução.

Existem variações do problema PRV que adicionam mais complexidade para se adequar aos problemas reais de logística, uma delas é o problema de roteamento de veículos

com janela de tempo, o PRVJT, assim como o PRV também pertence a classe NP-difíceis, nele deve-se considerar um intervalo de tempo para o atendimento dos consumidores nos locais das entregas a serem realizadas, por exemplo, não poderia realizar uma entrega para uma cliente em um horário em que não poderia receber ou para empresas que só funcionem em horário comercial.

Como exemplo de aplicações podemos citar:

- Entrega postal;
- Entrega em domicílio de produtos comprados nas lojas de varejo ou pela internet;
- Distribuição de produtos dos centros de distribuição (CD) de atacadistas para lojas do varejo;
- Escolha de rotas para ônibus escolares ou de empresas;

Para se aproximar de uma situação mais real, deve-se levar em consideração que o trânsito das grandes cidades muda constantemente, e o tempo de percorrer uma certa distância dependendo do dia e horário da semana também muda, assim como acidentes, obras em vias e etc, tornando o trânsito uma variável importante para o cálculo da rota de entrega. Tendo destinos com horários de funcionamento delimitados, pode não existir uma rota que satisfaça as restrições de horário, tornando impossível de ser encontrado uma rota que passe por todos os destinos com apenas um entregador, somando o problema de se identificar a quantidade de entregadores necessária para realizar todas as entregas respeitando todas as restrições de horários partindo do depósito.

Os atuais resultados encontrados na literatura referentes ao PRV e PRVJT comprovam que os algoritmos exatos restringem-se à resolução de problemas-teste com tamanho reduzido e janelas de tempo apertadas. Embora hoje possamos resolver problemas com um tamanho que seja ligeiramente maior que os de alguns anos atrás, o crescimento da capacidade dos computadores e da eficiência dos algoritmos está muito distante da curva exponencial representada por este problema. Pode-se dizer que os métodos exatos não são uma alternativa viável para situações onde a um número maior de consumidores, como ocorre na maioria dos casos reais ([CHABRIER, 2006](#)).

Por isso a utilização de meta-heurísticas tais como algoritmos genéticos, podem conseguir resultados satisfatórios em menor tempo que as soluções exatas. ([BRÄYSY; DULLAERT; GENDREAU, 2004](#))

1.2 Objetivos

Este trabalho tem como objetivo desenvolver uma solução computacional para o PRVJT utilizando algoritmos genéticos.

Ao encontrar rotas com muitos destinos de forma a não conseguir realizar todas as entregas, estas serão separadas em rotas menores, cada rota deve ser percorrida por um entregador diferente, e todas tem como endereço inicial e final o depósito.

Também sendo possível determinar o número máximo de entregadores, horário de saída do depósito e horário máximo para realizar todas as entregas, caso a rota não seja possível com esse número limite entregados até o horário limite, o usuário será sinalizado.

O trânsito é considerado como alterador de tempo entre os endereços, fazendo com que a solução mude dependendo do dia da semana e horário. Todas as rotas são organizadas considerando o trânsito médio.

Esta solução irá visar a minimização da quantidade de entregadores necessários, distância total percorrida e tempo para realizar o percurso.

1.2.1 Objetivos Específicos

- Realizar a integração com o Google Maps, considerando o trânsito utilizando o tempo médio entre os endereços.
- Implementar uma solução de algoritmos genéticos para PRVJT
- Criação de uma interface web para definição dos destinos, indicação do depósito, exibição em tabelas das rotas calculadas e exibição de cada rota em um mapa interativo do Google Maps.

1.3 Método de trabalho

Diferentes instancias do problema serão criadas para a simulação computacional, com estes testes já integrados programa, podendo ser escolhido e executado de uma maneira simples. Os endereços são reais, escolhidos em diferentes pontos no mapa e horários de abertura e fechamento são os indicados no Google Maps para cada endereço. Utilizando essa ambiente controlado situações impossíveis devem ser rejeitas.

1.4 Organização do trabalho

Este trabalho é dividido em 4 capítulos. O primeiro capítulo faz uma introdução geral do problema, com a descrição dos objetivos e a motivação para a resolução do

problema proposto.

O segundo capítulo trata do problema de forma separada, mostrando o que existe na literatura para uma possível solução. Também explica de forma mais detalhada o funcionamento das heurísticas e aplicações dos algoritmos genéticos para problemas semelhantes.

O terceiro capítulo é a proposta apresentada para a criação deste trabalho.

O quarto capítulo detalha implementação do programa e métodos utilizados para o seu funcionamento.

O quinto capítulo exhibe os testes executados, resultados encontrados e futuras melhorias que podem ser adicionadas ao projeto.

2 Revisão de Literatura

Nesse capítulo é feita uma revisão no estado da arte dos algoritmos de roteamento de veículos a aplicação de algoritmos genéticos para mesma finalidade.

2.1 Roteamento de Veículos com Janelas de Tempo

O PRVJT é amplamente estudado na literatura de pesquisa operacional. Tendo pelo menos duas frentes de soluções, as exatas e as baseadas em heurísticas.

2.1.1 Formulação matemática

O PRVJT pode ser definido a partir um grafo completo orientado $G = (V, A)$ em que $V = 0, \dots, n + 1$ é um conjunto de vértices e $A = (i, j) | i, j \in V$ é o conjunto de arcos. Cada arco (i, j) é associado a um tempo t_{ij} e um custo de travessia c_{ij} .

É necessário uma definição precisa do termo *custo de travessia*. Em casos práticos pode se considerar diversos fatores, tais como distancia, tempo, desgaste do veículo ao percorrer determinado caminho, entre outros fatores. Porém, quando se trata de problemas teóricos envolvendo janelas de tempo, é comum converter todas as medidas relevantes em unidades de tempo para fins de padronização e também para facilitar a comparação entre diferentes métodos. Por isso, adota-se aqui a mesma definição de custo que a maioria dos trabalhos teóricos da literatura, considerando que o custo de viagem consiste na distância convertida em unidades de tempo. (YVES ROCHAT, 1995)

Podemos descrever o problema como sendo um conjunto K de veículos com capacidade Q , eles devem atender n clientes, representados pelos vértices $1, \dots, n$. Considera-se que $N = V \setminus \{0, n + 1\}$ representa o conjunto de clientes. Os veículos devem partir do depósito e, após visitar todos os clientes, devem retornar ao mesmo local de onde partiram. Por conveniência, o depósito é representado por dois vértices, o vértice 0, que representa a origem, e o vértice $n + 1$ que representa o destino. A cada cliente i , uma demanda q_i é associada, esta deve ser atendida por um único veículo. E todos os vértices possuem uma janela de tempo $[e_i, l_i]$, o serviço no vértice i deve ser iniciado dentro desse intervalo. Caso ocorra que a chegada ao cliente i aconteça antes do horário previsto e_i , ele deve esperar a abertura da janela. O veículo não poderá chegar a i depois do instante l_i , pois isso faria violar a restrição de tempo do problema. Esse tipo de restrição é conhecido na literatura como janela de tempo rígida. A cada vértice é também associado um tempo de serviço, denotado por s_i . O objetivo é encontrar uma solução s de custo mínimo, de forma

a minimizar a soma de todos os custos de viagem $\sum_{(i,j) \in s} c_{ij}$ que são associados aos arcos (i, j) presentes nas rotas que compõem essa solução.

A formulação matemática do PRVJT, é apresentada pelas expressões:

$$\min \sum_{k \in K} \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ijk} \quad (2.1)$$

Sujeito as seguintes restrições:

$$\sum_{k \in K} \sum_{j \in V} x_{ijk} = 1, \forall i \in N \quad (2.2)$$

$$\sum_{j \in V} x_{0jk} = 1, \forall k \in K \quad (2.3)$$

$$\sum_{i \in V} x_{ijk} - \sum_{i \in V} x_{jik} = 0, \forall k \in K, \forall j \in N \quad (2.4)$$

$$\sum_{i \in V} x_{i(n+1)k} = 1, \forall k \in K \quad (2.5)$$

$$\sum_{i \in N} q_i \sum_{j \in V} x_{ijk} \leq Q, \forall k \in K \quad (2.6)$$

$$b_{ik} + s_i + t_{ij} - (1 - x_{ijk})M_{ij} \leq b_{jk}, \forall k \in K, \forall (i, j) \in A \quad (2.7)$$

$$e_i \leq b_{ik} \leq l_i, \forall k \in K, \forall i \in V \quad (2.8)$$

$$x_{ijk} \in \{0, 1\}, \forall k \in K, \forall (i, j) \in A \quad (2.9)$$

A variável binária x_{ijk} assume valor 1 se o veículo k passa pelo arco (i, j) e 0, caso contrário.

A função objetivo 2.1 expressa o custo total a ser minimizado. As restrições 2.2 asseguram que somente um veículo k sai de cada cliente i . As restrições 2.3, 2.4, 2.5 garantem a continuidade do caminho a ser percorrido pelo veículo k , ou seja, cada veículo parte do depósito, visita os clientes e em seguida retorna ao depósito. As restrições 2.6 fazem com que cada veículo k somente possa atender a um conjunto de clientes cuja demanda total não ultrapasse sua capacidade Q . As restrições 2.7, 2.8 asseguram a viabilidade das rotas no que diz respeito as restrições de janelas de tempo, em que b_{ik} representa o tempo em que o veículo k começa a atender o cliente i e M_{ij} são constantes

de valor suficientemente grande. As restrições 2.9 definem o domínio das variáveis de decisão. (CORDEAU et al.,)

2.1.2 Complexidade

Encontrar a solução ótima do PRVJT implica em obter simultaneamente a solução de vários problemas NP-difíceis, dentre os quais citam-se o *Problema do Caixeiro Viajante* (PCV) e o *Problema da Mochila*. Sendo assim, tal tarefa é também NP-difícil. Além disso, encontrar uma simples solução viável para o PRVJT dispondo de um conjunto limitado de veículos é NP-difícil no sentido forte (KOHL, 1995). Porém, uma solução inicial viável é trivial caso o número de veículos seja ilimitado, bastando atender cada consumidor com um veículo.

Os atuais resultados encontrados na literatura referentes ao PRVJT comprovam que os algoritmos exatos restringem-se à resolução de problemas-teste com tamanho reduzido e janelas de tempo apertadas. Embora hoje podemos resolver problemas com um tamanho que seja ligeiramente maior que o de alguns anos atrás, o crescimento da capacidade dos computadores e da eficiência dos algoritmos esta muito distante da curva exponencial representada por este problema. Pode-se dizer que os métodos exatos não são uma alternativa viável para situações onde a um número maior de consumidores, como ocorre na maioria dos casos reais. (CHABRIER, 2006)

Abordagens heurísticas e algoritmos aproximativos também tem sido utilizadas na resolução do PRVJT. As Heurísticas buscam obter uma solução em tempo hábil. Este fato torna as estratégias heurísticas muito poderosas se comparadas com abordagens exatas, que focam exclusivamente na obtenção da solução ótima. Uma boa heurística deve ser capaz de encontrar soluções próximas da ótima, em tempo bem inferior ao necessário pelos métodos exatos. A qualidade da solução não deve variar demasiadamente ao aplicá-la em diferentes ou ao mesmo problemas-teste. Até 2006, 45 do total de 56 problemas de Solomon tiveram uma solução ótima. Alguns casos foram gastos mais que cinco horas de processamento na resolução de algumas instancias, enquanto em outras puderam ser resolvidas em menos de um minuto. (JEPSEN; SPOORENDONK, 2006)

Os métodos aproximativos vem ao encontro destas características. Um método aproximativo é uma heurística com garantia de qualidade no resultado. A melhor solução encontrada por um algoritmo de aproximação esta sempre a uma distancia percentual previamente definida da solução ótima desconhecida. A "distancia do ótimo" é particular de cada algoritmo, podendo até não ser muito relevante em termos práticos. Um exemplo bem conhecido é o algoritmo PRIM, para árvore geradora mínima, que é capaz de oferecer uma solução viável para o PCV, que é no máximo duas vezes o ótimo em distancia total percorrida (ALVARENGA, 2005).

Dado essa complexidade, resolver esse problema utilizando de abordagens puramente exatas é uma tarefa extremamente árdua, demandando tempo computacional muito elevado. Por isso é motivado o desenvolvimento de novos algoritmos heurísticos com tempos mais reduzidos para a solução do PRVJT, mesmo que esses não garantam uma solução ótima.

2.1.3 Heurísticas

Heurísticas são procedimentos de busca que visam a obtenção de soluções com uma qualidade satisfatória em um tempo computacional aceitável. Porém tais procedimentos não garantem encontrar a solução ótima nem são capazes de mensurar o quão próxima a solução obtida está da ótima. Será enumerado as ideias centrais de algumas heurísticas construtivas e de refinamento disponíveis na literatura.

2.1.3.1 Construção de rotas

Um dos trabalhos mais antigos sobre heurística para construção de rotas para PRVJT proposto por Baker ([BAKER; SCHAFFER, 1986](#)) em 1989. Foi criada a partir da ideia da heurística das economias de Clarke e Wright ([CLARKE; WRIGHT, 1964](#)) que foi proposta para criação de soluções para o PRV. O algoritmo funciona primeiramente criando uma rota partindo do depósito para cada cliente i , para em seguida, executar várias iterações, em cada interação, o algoritmo calcula quais duas rotas que podem ser combinadas de forma a gerar a maior economia possível.

Outra heurística proposta por ([LANDEGHEM, 1988](#)) também baseada na heurística das economias é uma heurística de dois critérios, nesta as janelas de tempo são utilizadas para mensurar o quanto uma ligação entre dois clientes é boa em termos de tempo.

De forma semelhante ([SOLOMON, 1987](#)) desenvolveu um algoritmo baseado na ideia na heurística das economias para resolução do PRVJT. Devido a existência de janelas de tempo, deve-se considerar também a orientação da rota. Também deve-se checar as violações de janelas de tempo quando mais de uma rota é combinada. Ela de forma igual a heurística das Economias original possuem complexidade $O^2 \log n^2$. Nesta heurística toda rota é inicializada encontrando o cliente mais próximo ao depósito que ainda não pertença a nenhuma rota. A cada iteração subsequente o cliente mais próximo ao último adicionado à rota é considerado para inserção ao final da rota que está sendo gerado. Quando a busca falha, uma nova rota é inicializada.

As heurísticas de ([LANDEGHEM, 1988](#)) e ([SOLOMON, 1987](#)) de forma geral conseguem encontrar uma solução rapidamente. Porém as soluções que suas heurísticas encontram são geralmente de baixa qualidade. Geralmente pouco a mais de 10% do ótimo ([EL-SHERBENY, 2010](#)).

Criar uma rota por vez traz uma desvantagem, usualmente as rotas geradas por

ultimo são de baixa qualidade, uma vez que os clientes sem rota tendem estar distantes geograficamente (EL-SHERBENY, 2010).

É possível encontrar uma tentativa de solução deste problema de inserção no trabalho de Rousseau (POTVIN; ROUSSEAU, 1993) por meio de construção simultânea de varias rotas. A inicialização das rotas é feita usando a heurística de inserção de Solomon. Em cada rota o cliente mais distante do depósito é selecionado como semente. A partir desse ponto, computa-se a melhor inserção viável para cada cliente que ainda não foi visitado. Este método é melhor que a heurística de Solomon, porem as soluções geradas continuam distantes das ótimas.

Antes e Derigs (ANTES; DERIGS, 1997) evoluem as ideias clássicas de inserção. No seu trabalho, todo cliente sem rota designada recebe um custo de inserção de cada uma das rotas. A definição desse custo é semelhante ao adotado nas heurísticas de Solomon. Cada cliente sem rota envia uma proposta a rota com melhor oferta, cada rota aceita a melhor proposta dos clientes com menor número de alternativas. Vale observar que mais clientes podem ser inseridos em cada iteração. Se houver alguma violação nas rotas, um certo numero de veículos é removido e o processo é reiniciado.

Os resultados do trabalhos de Antes e Derigs (ANTES; DERIGS, 1997) são comparados aqueles apresentados por Potvin e Rousseau (POTVIN; ROUSSEAU, 1993). Segundo os autores, construir rotas paralelamente produz soluções de maior qualidade que construir rotas uma a uma.

2.1.3.2 Aprimoramento de rotas

Quase todas as heurísticas de melhoria de rotas tem a noção de vizinhança. A vizinhança de uma solução S é um conjunto de soluções $N(s)$ que podem ser geradas pela aplicação de uma única alteração denominada *movimento* na solução S .

Checar algumas ou todas as soluções de uma vizinhança pode revelar soluções melhores em relação a uma determinada função objetivo. Esta ideia pode ser repetida partindo-se da melhor solução obtida até o momento. Se em algum momento nenhuma solução melhor for encontrada em uma vizinhança, um ótimo local foi obtido. Trata-se definitivamente de um ótimo local, porem este pode eventualmente ser um ótimo global. A este algoritmo da-se o nome de Hill Climbing. (BRÄYSY; GENDREAU, 2005). Na próxima seção serão introduzidas varias estruturas de vizinhança empregadas na literatura para melhorar soluções do PRVJT. Em seguida serão descritos alguns dos algoritmos que as utilizam.

2.1.3.3 Estruturas de vizinhança

Uma estrutura de vizinhança mais utilizada em roteamento é a k -opt, onde k arcos são removidos e substituídos por outros k arcos. Um ótimo local obtido utilizando-se a vizinhança k -opt é dita solução k -optimal. Normalmente, k é no máximo 3.

Para todas as possíveis trocas 2 -opt e algumas das permutações da vizinhança 3 -opt, parte da rota é invertida. Isto comumente acarreta violações nas janelas de tempo. No trabalho de (POTVIN; ROBILLARD, 1995) são apresentadas duas variantes, a 2 -opt* e a Or -opt, que mantêm a direção da rota.

Na vizinhança Or -opt, um conjunto contíguo de até 3 clientes é realocado para outra posição na mesma rota. Uma vez que nessa vizinhança três arcos são trocados por outros três, é fácil observar que ela é um subconjunto da vizinhança 3 -opt. Desta forma, o tamanho da vizinhança é reduzido de $O(n^3)$ para $O(n^2)$. De forma geral, o tamanho da vizinhança k -opt é da ordem de $O(n^k)$. A vizinhança 2 -opt* consiste na troca de um segmento de uma rota por um segmento de outra rota. Esses operadores de vizinhança são muitas vezes denotados na literatura por *crossover* ou simplesmente *cross*.

Movimentos da vizinhança *exchange* alteram diferentes rotas através da troca simultânea de dois clientes. A vizinhança k -node, proposta no trabalho de (CHRISTOFIDES; BEASLEY, 1984) tem sido adaptada por alguns autores para que este leve em consideração aspectos referentes às janelas de tempo. Nesta estrutura, cada cliente i é considerado e os conjuntos $M1$ e $M2$ são identificados. Em $M1$ são alocados o cliente i e seu sucessor j . O conjunto $M2$ é formado pelos clientes mais próximos aos clientes i e j que não estejam na mesma rota que i e j (encontrados pela minimização do custo de inserção considerando distância euclidiana). A vizinhança é então definida pela remoção dos elementos desses conjuntos e posterior inserção em qualquer outra possível localização. Como trata-se de uma vizinhança de dimensões muito elevadas, apenas os k candidatos mais promissores são considerados.

Outra vizinhança explorada na literatura é a λ - interchange desenvolvida em (OSMAN, 1993), originalmente para o PRV. Trata-se de uma generalização do operador *relocate*. Nesta estrutura, um subconjunto de clientes de uma mesma rota é trocado por outro conjunto de outra rota. O mecanismo de geração λ - interchange pode ser descrito como segue. Dada uma solução para o problema, representada pelo conjunto de rotas $S = \{r_1, \dots, r_p, \dots, r_q, \dots, r_k\}$, um λ - interchange entre um par de rotas (r_p, r_q) consiste na troca dos clientes $S_1 \cup r_p$ de tamanho $|S_1| \leq \lambda$ por outro subconjunto $S_2 \cup r_q$ de tamanho $|S_2| \leq \lambda$ para gerar novas rotas $r_p^* = (r_p - S_1) \cup S_2$, $r_q^* = (r_q - S_2) \cup S_1$ e uma nova solução $S' = \{r_1, \dots, r_p^*, \dots, r_q^*, \dots, r_k\}$. A vizinhança $N_\lambda(S)$ de uma dada solução S é o conjunto de todos os vizinhos S' gerados para um dado valor de λ .

A vizinhança denotada por *shift-sequence* é proposta em (SCHULZE; FAHLE, 1999).

Nesta, um cliente é movido de uma rota para outra checando-se todas as possibilidades de inserção. Caso uma inserção possa se tornar viável pela remoção de outro cliente j , este é removido e inserido em outra rota. Este procedimento é repetido até que a viabilidade seja restabelecida.

A Tabela 1 apresenta uma breve descrição das estruturas de vizinhança comumente utilizadas na literatura por algoritmos de busca local para a resolução do PRVJT. Observa-se que algumas dessas são também utilizadas neste trabalho

Tabela 1 – Estruturas de vizinhança para o PRVJT

Vizinhança	Descrição
<i>Relocate</i>	Move um cliente de uma rota para outra.
<i>Exchange</i>	Troca dois clientes entre duas rotas
$2 - opt^*$	Troca um segmento de uma rota por um segmento de outra rota.
<i>Or-opt</i>	Um segmento contínuo de clientes é movido de uma posição em uma rota para outra posição da mesma rota.
<i>k-node</i>	Os clientes i , seu sucessor j e os dois clientes mais próximos que não estão na mesma rota são removidos. Tenta-se então inserir os quatro vértices em todas as possíveis localizações. Como trata-se de uma vizinhança de dimensões muito elevadas, apenas os k candidatos mais promissores são considerados.
$\lambda - interchange$	Um subconjunto S_1 de clientes de tamanho $ S_1 \leq \lambda$ de uma rota é trocado por um subconjunto S_2 de tamanho $ S_2 \leq \lambda$ de outra rota.
<i>Shift-sequence</i>	Um cliente é movido de uma rota para outra checando-se todas as possibilidades de inserção. Caso uma inserção se torne viável pela remoção de um consumidor j , este é removido e inserido em alguma outra rota. Este procedimento é repetido até que a viabilidade seja restabelecida.

2.1.4 Meta-heurísticas

Meta-heurísticas são procedimentos destinados a encontrar uma boa solução, eventualmente a ótima, consistindo na aplicação, em cada passo, de uma heurística subordinada,

a qual tem que ser modelada para cada problema específico (RIBEIRO, 1996) Contrariamente às heurísticas convencionais, as metaheurísticas são de caráter geral e providas de mecanismos para tentar escapar de ótimos locais. (SOUZA, M. J. F., 2011)

2.1.4.1 Busca Tabu (Tabu Search)

Essa meta-heurística foi introduzida por (GLOVER, 1986), mas foi (GARCIA; POTVIN; ROUSSEAU, 1994) propôs primeira aplicação para o problema de roteamento e programação de veículos com janela de tempo. O conceito básico da busca tabu (BT) é explorar o espaço solução, a cada iteração, movendo de uma dada solução para outra que pertença à sua vizinhança. Diferentemente dos métodos clássicos de descida, aceita-se soluções piores, o que pode gerar ciclos. Para evitar a ciclagem, as soluções já avaliadas são marcadas como proibidas e incluídas em uma lista tabu.

2.1.4.2 Têmpera Simulada (Simulated Annealing)

A têmpera simulada (TS) é uma técnica de relaxação estocástica baseada no processo térmico utilizado na metalurgia para obtenção de estados de baixa energia num sólido. No algoritmo da TS, uma nova solução x_t é aceita sempre que $f(x_t) < f(x)$, onde x é a solução corrente. Para fugir dos mínimos locais, soluções com $f(x_t) \geq f(x)$ também são aceitas com uma probabilidade $e^{\delta/T}$, onde $\delta = f(x_t) - f(x)$ e T é um parâmetro (chamado temperatura) que varia ao longo das iterações, partindo de um número grande e terminando próximo de zero. A queda na temperatura ocorre gradativamente e costuma ser feita através da regra $T_k = \alpha T_{(k-1)}$ para $0 \leq \alpha \leq 1$. (CHIANG; RUSSELL, 1996)

2.1.4.3 Busca Local com Múltiplos Pontos Iniciais (Multi-Start Local Search)

Este método de busca local envolve a geração de um conjunto de soluções iniciais, seguida da aplicação de um procedimento de refinamento a cada solução gerada. As diferentes soluções iniciais permitem uma diversificação do espaço de busca, o que evita ótimos locais. A heurística de inserção mais barata é usada para criar a solução inicial, que é refinada por uma extensão da heurística de cadeia de ejeções, com o objetivo de reduzir o número de veículos. Finalmente, uma modificação da troca cruzada é empregada para reduzir a distância total percorrida em cada solução. (BRÄYSY; HASLE; DULLAERT, 2004)

2.1.4.4 Algoritmos Genéticos / Evolutivos

Algoritmos genéticos e evolutivos são metaheurística com boas aplicações no problema de PRV e PRVJT, (HOMBERGER, 2005) propuseram duas estratégias evolucionárias diferentes. As duas estratégias utilizam uma aproximação estocástica baseada na heurística das economias, ou seja, os clientes pertencentes a lista das economias são

escolhidos aleatoriamente para constituir a rota. A função objetivo pondera o número de rotas, a distância total e um critério que determina a facilidade de eliminação da menor rota. A mutação é feita pelas heurísticas de refinamento *Or-opt*, $2 - opt^*$ e $\lambda - Interchange$. No primeiro algoritmo, o cruzamento não é executado. No segundo, o cruzamento é feito através de um procedimento uniforme.

Foi desenvolvido por (JUNG; MOON, 2002) um algoritmo genético híbrido, no qual a função objetivo é baseada na distância. O algoritmo começa com a aplicação da heurística de inserção de (SOLOMON, 1987) para a determinação de uma solução inicial. O primeiro cliente de cada rota é escolhido de forma aleatória entre o cliente mais distante do depósito, o cliente com o menor instante final da janela de tempo e um cliente também determinado aleatoriamente.

Em seu algoritmo a seleção é feita pelo torneio. Para o cruzamento, o grafo que contém o depósito, os clientes e as arestas utilizadas para formar as rotas de cada veículo são mapeados e a escolha dos pontos de corte é feita por meio de curvas ou figuras geométricas de diferentes tipos. A figura 1 exemplifica o procedimento de cruzamento. As regiões formadas pela sobreposição das figuras geométricas definem conjuntos de nós. Cada conjunto pertence a um único pai. Primeiramente, definimos as rotas dentro desses conjuntos. Como essa divisão de nós é arbitrária, restarão várias rotas desconexas, de modo que é preciso utilizar algoritmos de reparação para reconstruir uma solução factível. Essa reconstrução é feita seguindo a regra do vizinho mais próximo.

Na mutação, são feitas mudanças de nós entre, no máximo, 3 rotas. As heurísticas de refinamento *or-opt*, *relocate* são aplicadas ao final da iteração para melhorar a solução.

Algoritmos evolucionários para o problema com janela de tempo foram analisados e comparados por Braysy (BRÄYSY; DULLAERT; GENDREAU, 2004).

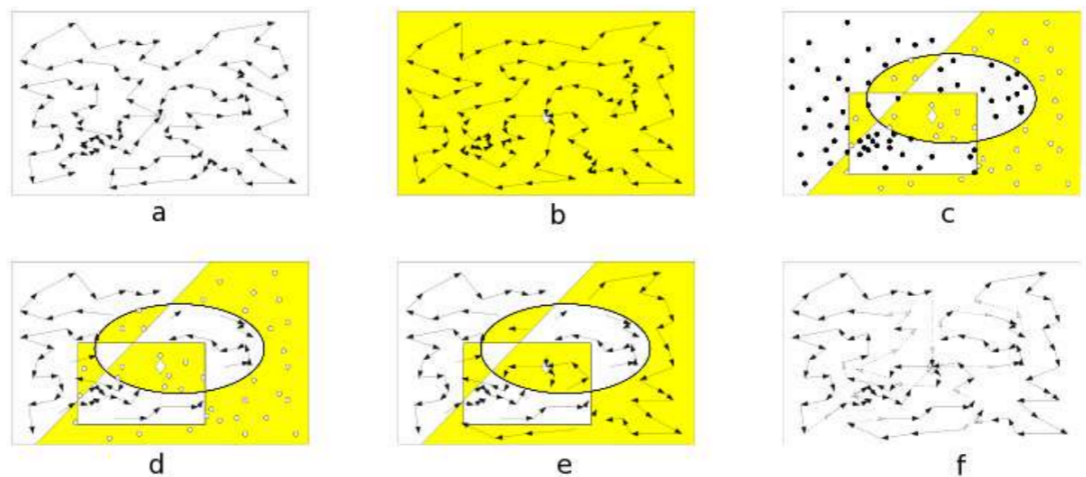


Figura 1 – O crossover utilizado por Jung e Moon. As Figuras a e b representam, respectivamente, os pais 1 e 2. A Figura c mostra a divisão dos clientes com base em figuras geométricas. A Figura d mostra as ligações feitas nas regiões referentes ao primeiro pai, enquanto a Figura e mostra as rotas internas à região referente ao segundo pai. Finalmente, a Figura f mostra as rotas após a aplicação do algoritmo de reparação.

2.2 Algoritmos genéticos

AG é uma técnica amplamente utilizada de IA, que utilizam conceitos provenientes do princípio de seleção natural para abordar uma ampla série de problemas, geralmente de adaptação. (LUCAS, 2002)

2.2.1 Funcionamento

Inspirado na maneira como a seleção natural explica o processo de evolução das espécies, Holland (HOLLAND, 1975) decompôs o funcionamento dos AG em sete etapas, essas são *inicialização*, *avaliação*, *seleção*, *cruzamento*, *mutação*, *atualização* e *finalização* conforme a Figura 2.

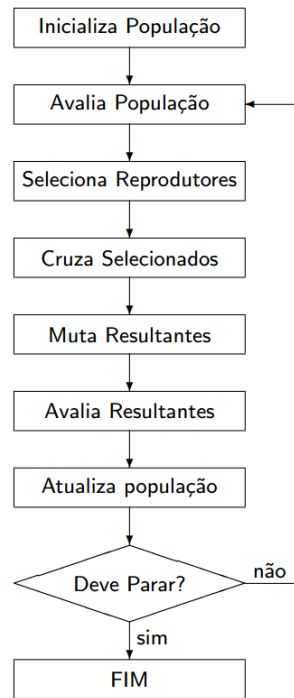


Figura 2 – Estrutura de um AG (LUCAS, 2002)

2.2.2 Inicialização

Criar uma população de possíveis respostas para um problema. É comum fazer uso de funções aleatórias para gerar os indivíduos, sendo este um recurso simples que visa fornecer maior diversidade.

2.2.3 Avaliação

Avalia-se a aptidão das soluções, os indivíduos da população, então é feita uma análise para que se estabeleça quão bem elas respondem ao problema proposto. A função de avaliação também pode ser chamada de função objetivo. Ela pode variar de acordo com problema, Calcular com exatidão completa o grau de adaptação dos indivíduos pode ser uma tarefa complexa em muitos casos, e se levarmos em conta que esta operação é repetida varias vezes ao longo do processo de evolução, seu custo pode ser consideravelmente alto. Em tais situações é comum o uso de funções não determinísticas, que não avaliam a totalidade das características do indivíduo, operando apenas sobre uma amostragem destas.

2.2.4 Seleção

Ela é a responsável pela perpetuação de boas características na espécie. Neste estágio que os indivíduos são escolhidos para posterior cruzamento, fazendo uso do grau de adaptação de cada um é realizado um sorteio, onde os indivíduos com maior grau de adaptação tem maior probabilidade de se reproduzirem. O grau adaptação é calculado a

partir da função de avaliação para cada indivíduo, determina o quão apto ele está para reprodução relativo a sua população.

Selection Roulette Wheel: Faz a soma de todos os valores da função de aptidão da população, depois calcula a porcentagem de cada indivíduo referente ao total e guarda em um vetor. Então é gerado um valor X aleatório entre 0 a 1, e multiplicado pelo valor total dos pesos. Para selecionar o indivíduo é feito um loop no vetor de pesos e seus valores somados até que seja menor ou igual ao valor X aleatório calculado antes, a posição atual no vetor deste valor, será a utilizada para selecionar o indivíduo no vetor de indivíduos. Desta forma aumentando a possibilidade de selecionar um indivíduo com melhor aptidão.

2.2.5 Cruzamento

Após alguns indivíduos serem selecionados, normalmente pares, estes são usados como base para novos descendentes, combinando as suas características para gerar um novo genoma.

2.2.6 Mutação

Características dos indivíduos resultantes do processo de reprodução são alteradas, acrescentando assim variedade a população. A mutação opera sobre os indivíduos resultantes do processo de cruzamento e com uma probabilidade pré-determinada efetua algum tipo de alteração em sua estrutura. A importância desta operação é o fato de que uma vez bem escolhido seu modo de atuar, é garantido que diversas alternativas serão exploradas.

2.2.7 Finalização

É testado se as condições de encerramento da evolução foram atingidas, retornando para a etapa de avaliação em caso negativo e encerrando a execução em caso positivo.

Os critérios para a parada podem ser vários, desde o número de gerações criadas até o grau de convergência da população atual.

Toda base dos AG se fundamenta nos indivíduos, eles são a unidade básica em qual o algoritmo se baseia, sua função é codificar as possíveis soluções do problema a ser tratado e partir de sua manipulação no processo evolutivo, a partir daí que são encontradas as respostas.

Esses indivíduos precisam de uma representação, essa será o principal responsável pelo desempenho do programa. É comum chamar de *genoma* ou *cromossomo* para se referir ao indivíduo. Por essa definição podemos resumir um indivíduo pelos genes que possui, ou seja seu *genótipo*.

Apesar de toda representação por parte do algoritmo ser baseada única e exclusivamente em seu genótipo, toda avaliação é baseada em seu fenótipo, o conjunto de características observáveis no objeto resultante do processo de decodificação dos genes do indivíduo, ver Tabela 2.

Tabela 2 – Exemplos de genótipos e fenótipos correspondentes em alguns tipos de problemas (LUCAS, 2002)

Problema	Genótipo	Fenótipo
Otimização numérica	0010101001110101	10869
Caixeiro viajante	CGDEHABF	Comece pela cidade C, depois passe pelas cidades G, D, E, H, A, B e termine em F
Regras de aprendizado para agentes	$C_1R_4C_2R_6C_4R_1$	Se condição 1 (C_1) execute regra 4 (R_4), se (C_2) execute (R_6), se (C_4) execute (R_1)

Para cada indivíduo é calculado o seu grau de adaptação, a partir de uma função objetivo, comumente denotada como na formula 2.10.

$$f_O(x) \quad (2.10)$$

Que vai representar o quão bem a resposta apresentada pelo individuo soluciona o problema proposto.

Também é calculado o grau de adaptação do indivíduo relativo aos outros membros da população a qual ele pertence, esse é chamado de grau de aptidão, para um indivíduo x temos seu grau de aptidão denotado pela fórmula 2.11.

$$f_A(x) = \frac{f_O(x)}{\sum_{i=1}^n f_O(i)} \quad (2.11)$$

Sendo n o tamanho da população.

A dinâmica populacional é a responsável pela evolução, ao propagar características desejáveis a gerações subsequentes no processo de cruzamento, enquanto novas são testadas no processo de mutação.

Algumas definições importantes relativo as populações de um AG são:

Geração: É o número de vezes em que a população passou pelo processo de seleção, reprodução, mutação e atualização.

Média de adaptação: É a taxa média que os indivíduos se adaptaram ao problema, é definida pela formula 2.12.

$$M_A = \frac{\sum_{i=1}^n f_o(i)}{n} \quad (2.12)$$

Grau de convergência: define o qual próxima esta a media de adaptação desta população relativo as anteriores. O objetivo dos AG é fazer a população convergir para uma valor de adaptação ótimo. Um estado negativo que pode ocorrer relativo a esta medida é a *convergência prematura*, a mesma ocorre quando a população converge em uma média de adaptação sub-ótima, e dela não consegue sair por causa de sua baixa diversidade.

Diversidade: Mede o grau de variação entre os genótipos da população. Ela é fundamental para o tamanho da busca. Sua queda esta fortemente ligada ao fenômeno de *Convergência prematura*.

Elite: São os indivíduos mais bem adaptados da população. Uma técnica comum nos AG é p *elitismo*, onde são selecionados k melhores indivíduos que serão mantidos a cada geração.

2.2.8 Aplicações

Existem vários aplicações para os algoritmo genéticos, por serem uma inteligência artificial não supervisionada, de rápido aprendizado e podendo ser paralelizado.

O modelo m-PRC(Problema de Rotas de Cobertura multi-veículo) é uma aplicação de algoritmos genéticos para construção de rotas em uma região mapeada, encontrando uma boa distribuição de viaturas para patrulhamento urbano, que pode ser utilizado por departamentos segurança, como a policia, guardas municipais ou segurança privada citeWashington. O Modelo é definido como um grafo não direcionado 2.13.

$$G = (V \cup W, E) \quad (2.13)$$

Onde 2.14:

$$V \cup W \quad (2.14)$$

Compõem o conjunto de vértices e E o conjunto de arestas, ou seja, o subgrafo induzido por E e um grafo completo cujo conjunto de nós é V. V são todos os vértices que podem ser visitados e é composto pelo subconjunto T, que são os vértices que devem ser visitados por algum veiculo. W é um conjunto de vértices onde todos os M veículos devem passar. M é o numero de rotas de veículos que começam no vértice base V_0 .

O m-PRC atribui o conjunto de m rotas de veículos com as restrições: todas as m rotas de veículos começam e terminam na base V_0 , Tem exatamente m rotas, cada vértice de V pertence a no máximo uma rota, cada vértice de T pertence a exatamente uma rota, com exceção a base, cada vértice de W deve ter uma rota que passa por ele e em uma distancia C de um vértice V visitado, O modulo da diferença entre o número de vértices de diferentes rotas não pode exceder um determinado valor R . A Figura 3 mostra o grafo da relação de V com W .

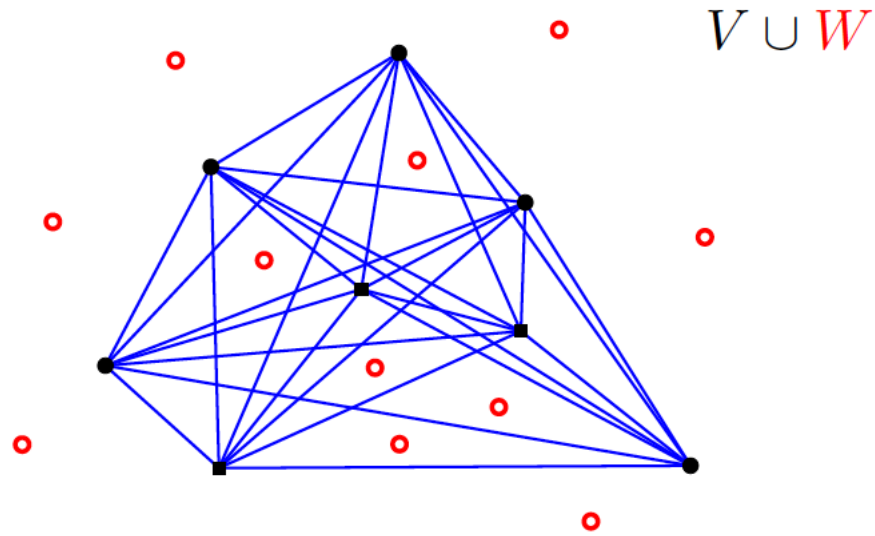


Figura 3 – Exemplo de grafo não direcionado para $V \cup W$. (OLIVEIRA, 2009)

Para utilizar o algoritmos genéticos com o modelo m-PRC, o trabalho propõem dois modelos. O AGS (Algoritmo genético sequencial), que utiliza heurísticas GENIUS e 2-opt balanceada para ajustes finais para tentar melhor a solução; O AGH(Algoritmos genéticos H-1-PRC), que utiliza heurísticas H-1-PRC-MOD e 2-opt balanceada em todo o processo de resolução.

A conclusão de (OLIVEIRA, 2009) é que a utilização de algoritmos genéticos para a resolução de uma adaptação do problema de rotas de cobertura de veículos como bastante relevantes e de fácil manipulação. O modelo AGS resolve o problema de forma rápida e tem uma fácil implementação dentro dos critérios de comparação adotadas. O modelo AGH é mais lento e não conseguiu encontrar a solução para alguns exemplos.

Homberger, Jorg, Gehring e Hermann propuseram duas estratégias evolucionárias diferentes. As duas estratégias utilizam uma aproximação estocástica baseada na heurística das economias, ou seja, os clientes pertencentes a lista das economias são escolhidos aleatoriamente para constituir a rota. A função objetivo pondera o número de rotas, a distância total e um critério que determina a facilidade de eliminação da menor rota. A mutação é feita pelas heurísticas de refinamento Or-opt, 2-opt e λ -Interchange. No primeiro algoritmo, o cruzamento não é executado. No segundo, o cruzamento é feito através de um procedimento uniforme. (??)

O trabalho de Sabir Ribas na universidade federal de fluminense utilizou uma abordagem híbrida para resolver o Problema de Roteamento de Veículos, usando Algoritmos Genéticos com a metaheurística Iterated Local Search e o método Variable Neighborhood Descent. Chama de IILS-SP, desenvolvido em C++. O algoritmo foi submetido a 56 problemas teste de 100 clientes, para cada problema, executado 5 vezes em intervalos de 10min. Teve resultados melhores do que os encontrados pelo autor em sua pesquisa na literatura. ([RIBAS, 2011](#)).

Humberto, Germano e Guilherme também tentaram resolver o PRVJT utilizando GA, para a criação da população inicial utilizar a heurística Push-Forward Insertion Heuristic. Utilizando 56 problemas de teste e 100 clientes, chamadas instâncias de Solomon de 1987, mesma base de teste utilizadas em sua revisão de literatura para comparação de resultados, utiliza a mesma metodologia de teste do Sabir Ribas. ([OLIVEIRA, 2005a](#))

No trabalho de Glaydston da UniAracruz e Luiz Antonio da Instituto Nacional de Pesquisas Espaciais utilizaram algoritmos genéticos para resolver Problemas de Roteamento de Veículos Dinâmico com Janelas de Tempo, em casos reais, podem existir mudanças que podem afetar as rotas, pensando nisso, o algoritmo recalcula a rota de todos os veículos. O GA deve um resultado equivalente ou superior em alguns casos se comparados com as heurísticas. ([LORENA,](#))

3 Metodologia

O software desenvolvido tem como objetivo demonstrar uma solução para o problema de PRVJT, descobrindo caminho viáveis entres diferentes endereços, onde é possível realizar as entregas em um período pre determinado, também identificar roteiros que são impossíveis de serem realizados a tempo.

Para adquirir informações próximas as reais de distancia e locomoção entre endereços esta sendo utilizado a API do Google Maps([Google Inc., 2018](#)).

Por se tratar de entregas de pequenos porte, os testes foram criados com endereços dentro das proximidades da cidade de São Paulo, uma grande metrópoles com um dos maiores índices de transito ([TOMTOM INTERNATIONAL BV, 2017](#)), por esse motivo a menor distância de rota pode não ser a melhor escolha para certos horários do dia, podendo uma rota com maior distância que evita transito ser uma melhor escolha.

Para preparar o calculo da rota, deve-se levar em consideração que todos os entregadores partem de uma única origem chamada deposito, a quantidade de entregadores é predefinida, o sistema ira devolver rotas para atender todos os pontos visando diminuir a quantidade necessária de entregadores e distancia percorrida sem desrespeitar as restrições de janelas de tempo.

Cada entrega tem um tempo médio de descarga, produtos pequenos podem demorar minutos, muitos produtos demoram mais para a retirar do veiculo e produtos grande podem precisar ser levamos com mais cuidado.

Cada endereço tem um horário de abertura e fechamento para realizar entregas, por exemplo, um super mercado recebe produtos na madrugada, já que receber em seu horário normal de abertura provavelmente irá atrapalhar as compras dos clientes, então esse horário deve ser considerado como restrição de tempo.

Por causa das restrições de tempo, quando um entregador chega a um endereço deve ser avaliado com penalidade caso o mesmo chegue antes ou depois da janela de tempo, se chegar antes o mesmo deve esperar até o horário de abertura do endereço, somar o tempo de descarga e sair para o próximo endereço, caso chegue depois é considerado que não é uma solução de rota viável para as restrições definidas.

É importante avaliar situações onde não é possível realizar as entregas, essas situação podem depender de distancia, por exemplo, caso o deposito esteja longe a ponto de um entregador não ter como chegar no endereço dentro de sua janela de tempo, ou caso mesmo sendo próximos ocorrer uma mudança no transito tornando impossível realizar a entrega á tempo, ou ainda pelo fato de ser definido um numero finito de entregadores, que

podem não ser suficientes para realizar todas as entregas do roteiro.

Depois que uma entrega é feita, uma nova rota deve ser recalculada, agora como ponto inicial o endereço atual, todo o processo será feito para verificar se o trânsito ou possíveis atrasos não afetaram a ordem dos próximos destinos, depois do recálculo o entregador deve seguir o próximo destino indicado.

Se caso não for mais possível entregar no horário por motivos de piora de trânsito ou um grande tempo de atraso para descarregar, um alerta será emitido indicando que todos os destinos não podem ser visitados a tempo.

Cada entregador tem seu próprio recálculo de rota, sendo que se um entregador concluir todos os destinos, os outros continuam pedindo novas rotas até que todos terminem suas entregas.

O fluxograma a baixo demonstra o fluxo da solução.

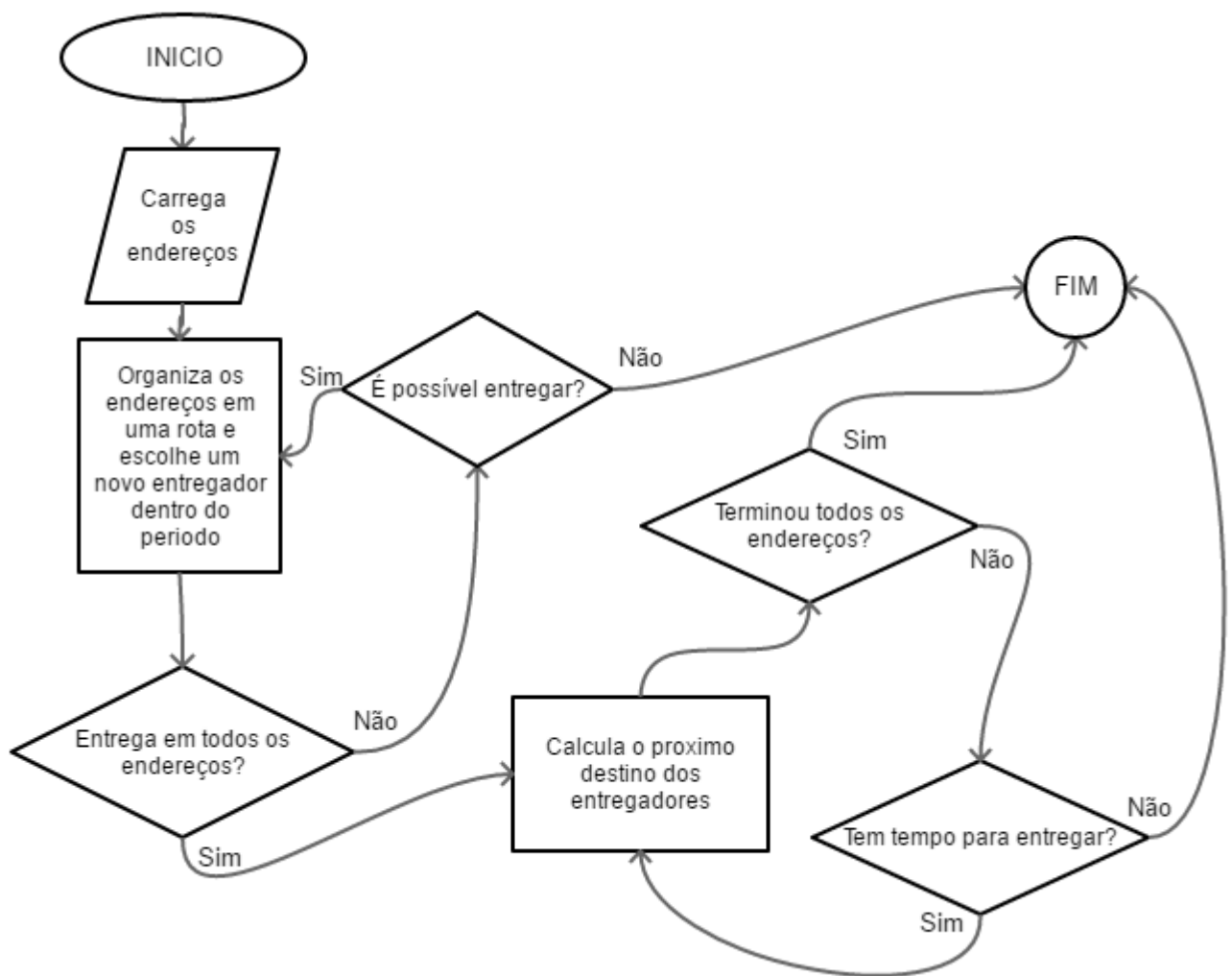


Figura 4 – Fluxograma macro do funcionamento do software.

Sempre que é necessário calcular a rota, o módulo de GA é chamado. Considerando

que um indivíduo é uma solução completa, gera uma população de varias soluções, onde a ordem da rota é aleatória somente mantendo o deposito fixo como primeiro endereço para criar a população inicial.

Para cada rota da população de indivíduos a seleção determina dois para a realização do cruzamento, onde endereços das duas rotas são trocados de forma a criar duas novas rotas mantendo o deposito sempre como inicial. A mutação é executada individual em cada rota, mudando de posição um ou mais endereços da rota, sempre mantendo o deposito como ponto inicial.

Depois que todas as rotas dos indivíduos da população foram modificados, agora é hora de verificar quais são os melhores, o que define isso é a função de aptidão, todos os parâmetros da rota são agrupados em um único numero e a rota que tem o menor número é a melhor rota da população. O valor de aptidão é definido com a soma da distância entre todos os endereços da rota, mais o tempo de cada um dos trajetos com o tempo de espera e descarga.

A função de aptidão do GA considera para uma frota homogênea de m veículos o horário de saída como parâmetros inicial, com isso, utiliza o tempo dado pelo Google Maps entre os pontos e soma ao horário verificando se está dentro da janela de tempo do destino. Se o horário calculo for menor que o de abertura, é somado o tempo restante de espera e uma penalidade por chegar antes. Se o horário for maior que o tempo de fechamento, é somado o tempo restante de espera até a abertura no próximo dia mais uma penalidade pela diferença. O Valor de aptidão final é a soma da distancia em metros do percurso passando por todos os pontos, com o tempo total em minutos com as penalidades de chegar antes ou depois das janelas de tempo.

A função de aptidão é definida como:

$$v(s) = \sum_{k=1}^m d_k + \sum_{i=1}^n \alpha_i \text{Max}(0, tc_i - l_i) + \beta_i te_i, \quad s \in S$$

Figura 5 – Formula de avaliação de solução de rota (*GENDREAU et al., 1999*)

Onde para cada ponto de entrega i temos um e_i e um l_i que representam, respectivamente, a abertura e o fechamento da janela de tempo do ponto i , O tc_i é o tempo de chegada do veículo no ponto i , te_i calculado como $e_i - tc_i$.

dk é a distância total percorrida na rota k para todo $k=1,...,m$, α_i é um coeficiente de penalidade associado à chegada do veículo no vértice i após o fechamento da janela de tempo.

O GA foi configurado com o numero de gerações em 200, tamanho da população em 1000, melhores indivíduos por geração em 20, probabilidade de cruzamento em 70% e

de Mutação em 0,5%.

Os testes foram configurados com um numero de 6 roteiros diferentes, endereços dentro da cidade de São Paulo e cidades próximas. Utilizando 4 tipos de mutação, sendo DisplacementMutation, InsertionMutation, InversionMutation e SwapMutation e um tipo de cruzamento, o SubRouteInsertionCrossover.

Considerando o transito médio enviado pelo Google Maps, para poder demonstrar o impacto do transito nos caminhos calculados. Tudo será rodado 10 vezes e será retirada uma média do valor de aptidão, por que, cada vez que roda o GA a resposta da solução pode mudar, por ele não ser determinístico.

E uma base pré-definidas rotas, para prevenir possíveis problemas sera ignorado o transito atual. Ja que o mesmo altera dependendo das condições do clima ou horário do dia. Então utilizando o Google Maps, um cache inicial foi preparado e o software utiliza simulando uma buscar ao Google Maps, com isso, a informação é obtida mais rapidamente e sempre fixa para garantir a resposta pré-determinada do teste.

O fluxograma a baixo demonstra o funcionamento do software utilizando o GA.

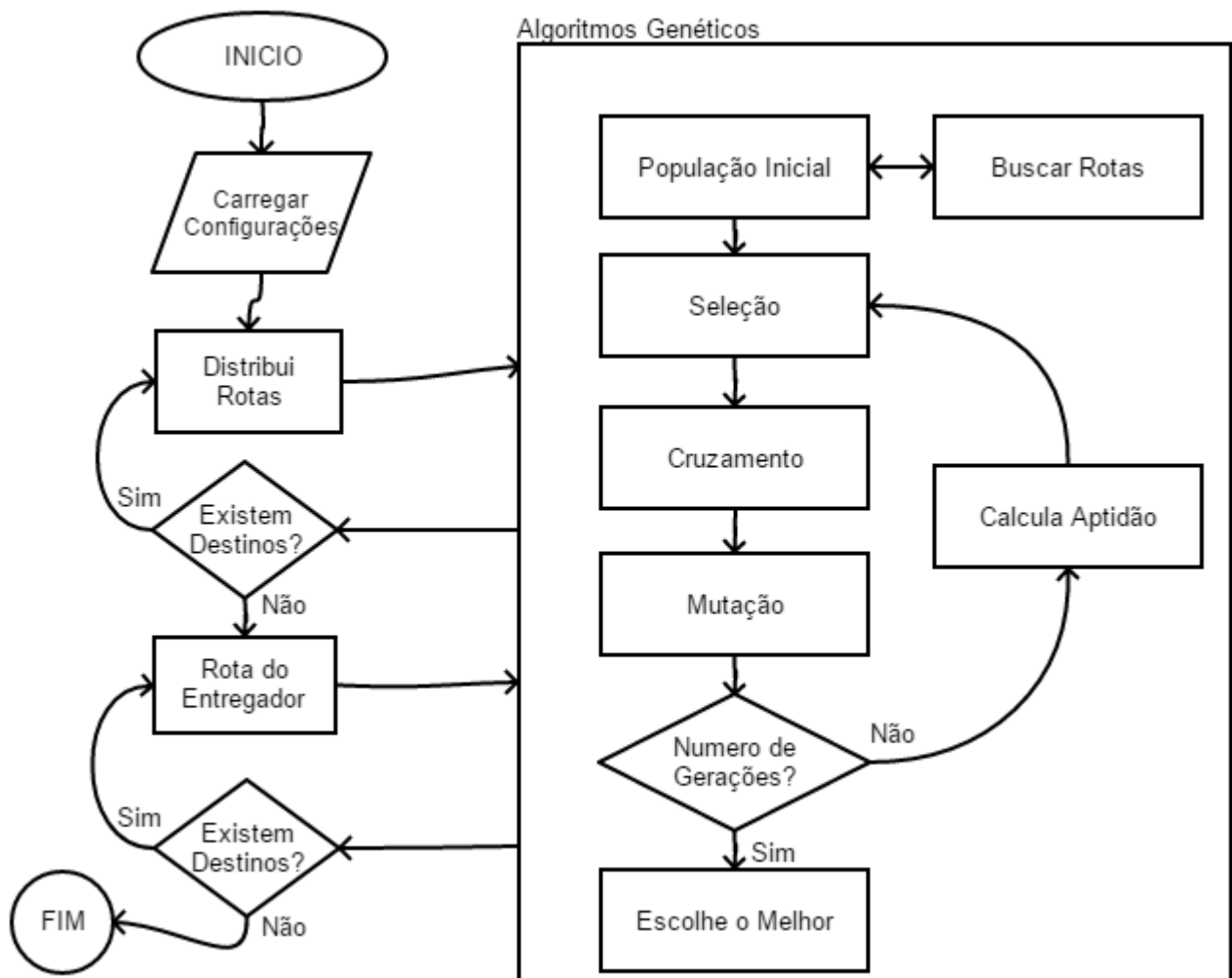


Figura 6 – Fluxograma macro da integração com o GA.

4 Implementação

Nesse capítulo será apresentado mais aprofundadamente as tecnologias, ferramentas e métodos que foram utilizados para a implementação do algoritmo genético para busca de rota com janela de tempo.

4.1 Tecnologias

O software foi desenvolvido na linguagem C# com Visual Studio 2017 com o Framework .Net Core 2.2 compatível com Windows, Linux e MacOS para o Back-End e ReactJs para a interface de utilização. Toda comunicação da interface com o calculador de rotas é por WebAPI Rest, onde são trocados dados em formato de Json.

4.2 Estrutura do Projeto

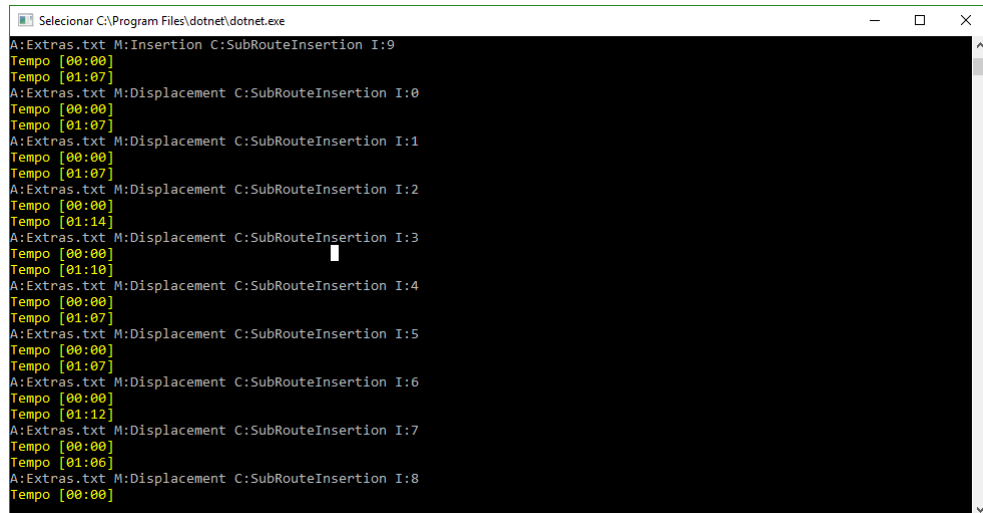
Para facilitar o entendimento inicial do projeto, é importante ter uma orientação de onde está cada parte de seu comportamento. A Software é dividido em 4 projetos principais, sendo um projeto feito em ReactJs e 3 projetos em C#, onde eles são:

4.2.1 Calculador de Rotas - CalcRoute

Projeto principal que tem toda a logica de preparação das rotas, recebendo parâmetros e utilizando o GA para gerar a resposta. A classe PRVJTFinder que é responsável por essa tarefa, podendo carregar as configurações por arquivo ou recendo os parâmetros diretamente. Os endereços podem vir sem as coordenadas, por que existe uma preparação dos endereços antes de começar o calculo das rotas. Os projetos CalcRoute.GeneticAlgorithm e CalcRoute.Routes são auxiliares para o calculo da rota.

4.2.2 Gerador de Dados - CalcRoute.RouteGenerate

Para ajudar na identificação dos melhores parâmetros a ser utilizado, esse projeto é configurado para utilizar todos os cruzamentos e mutação do GA, com isso, diferentes configurações de população, numero de gerações, taxa de cruzamento e taxa de mutação, podem ser testados e os resultados são agrupados em um arquivo CSV com o valor do melhor valor de aptidão encontrado para cada parâmetro para facilitar a analise dos dados.



```

Selecionar C:\Program Files\dotnet\dotnet.exe
A:Extras.txt M:Insertion C:SubRouteInsertion I:9
Tempo [00:00]
Tempo [01:07]
A:Extras.txt M:Displacement C:SubRouteInsertion I:0
Tempo [00:00]
Tempo [01:07]
A:Extras.txt M:Displacement C:SubRouteInsertion I:1
Tempo [00:00]
Tempo [01:07]
A:Extras.txt M:Displacement C:SubRouteInsertion I:2
Tempo [00:00]
Tempo [01:14]
A:Extras.txt M:Displacement C:SubRouteInsertion I:3
Tempo [00:00]
Tempo [01:10]
A:Extras.txt M:Displacement C:SubRouteInsertion I:4
Tempo [00:00]
Tempo [01:07]
A:Extras.txt M:Displacement C:SubRouteInsertion I:5
Tempo [00:00]
Tempo [01:07]
A:Extras.txt M:Displacement C:SubRouteInsertion I:6
Tempo [00:00]
Tempo [01:12]
A:Extras.txt M:Displacement C:SubRouteInsertion I:7
Tempo [00:00]
Tempo [01:06]
A:Extras.txt M:Displacement C:SubRouteInsertion I:8
Tempo [00:00]

```

Figura 7 – Tela da geração de dados de comparação

4.2.3 API de Calculo de Rotas - CalcRoute.API

Projeto de comunicação com a interface usando POST, recebendo parâmetros da interface e organiza para serem calculados no CalcRoute. O retorno é organizado e os campos formatados para melhor apresentação na tabela da interface.

4.2.4 Interface para comunicação com a API

A interface é utilizada para interagir com o calculador de rotas de uma forma mais simples, podendo procurar por endereços, adicionar a lista, definir o depósito, os horários de cada endereço, número de entregadores e tempo de descarga de cada endereço. Para melhor visualização dos testes, é possível encontrar todos os roteiros de teste para fácil execução e verifica os resultados.

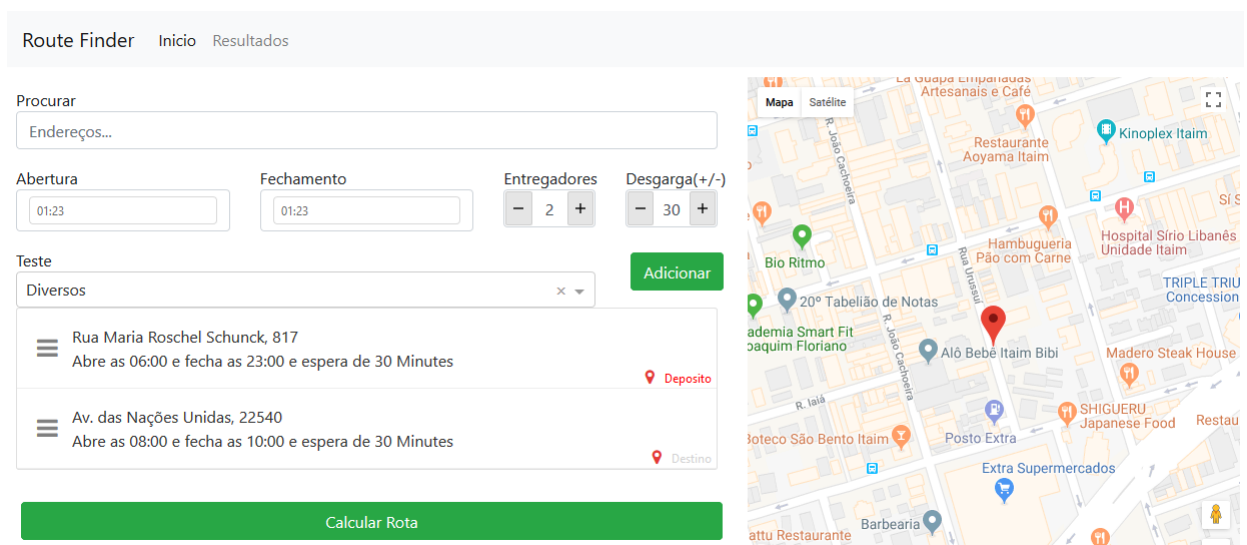


Figura 8 – Aparência da interface, configuração das rotas

- No campo procurar é preciso digitar o endereço para ser localizado no Google Maps, uma lista de possíveis escolhas aparece a baixo assim que começa a digitar, encontra o endereço e seleione.
- O mapa exibe o endereço selecionado.
- Abertura é o horário que que é permitida a entrada no local, no caso do deposito é o horário que o entregador sai para realizar as entregas.
- Fechamento é o horário limite para realizar a entrega, no caso do deposito, é o horário limite para terminar todas as entregas, horário que o deposito fecha.
- Entregadores é o numero de entregados disponível para realizar as entregas.
- Descarga o tempo médio para descarregar no endereço de entrega, no caso do deposito esse campo não é utilizado.
- Adicionar Agrupa os campos do endereço, abertura, fechamento e descarga para a lista.
- Lista de endereços reúne todos os endereços escolhidos para enviar para o calculador de rotas, o primeiro da lista é sempre o deposito
- Calcular Rota faz a chama da Api de calculo e envia a lista.

Entregador 1							
Endereço Saída	Endereço Chegada	KM	Saída	Percurso	Espera	Entrada	Descarga
Rua Maria Roschel S...	Av. das Nações Unid...	21,472	06:00:00	00:55:01	01:04:59	08:00:00	00:30:00
Av. das Nações Unid...	Rua Urussuí, 271 - It...	16,221	08:30:00	00:29:17	03:00:43	12:00:00	01:30:00
Rua Urussuí, 271 - It...	Av. Paulista - Bela Vis...	5,718	13:30:00	00:18:53	00:00:00	13:48:53	00:30:00
Av. Paulista - Bela Vis...	Rua Augusta - Cons...	1,798	14:18:53	00:06:51	00:00:00	14:25:44	00:20:00

Entregador 2							
Endereço Saída	Endereço Chegada	KM	Saída	Percurso	Espera	Entrada	Descarga
Rua Maria Roschel S...	Av. Engenheiro Eusé...	22,206	06:00:00	00:55:48	05:04:12	12:00:00	01:00:00
Av. Engenheiro Eusé...	Rua Vergueiro - Vila ...	15,647	13:00:00	00:35:19	00:00:00	13:35:19	00:10:00
Rua Vergueiro - Vila ...	Praça da Sé - Centro,...	8,067	13:45:19	00:21:36	00:00:00	14:06:55	00:30:00
Praça da Sé - Centro,...	Catavento Cultural e ...	1,920	14:36:55	00:07:07	00:00:00	14:44:02	00:30:00

Entregador:

1

Figura 9 – Aparência da interface para exibição dos resultados

Cada entregador é exibido de forma separada, com sua própria tabela de endereço e recalculo para o próximo destino. A lista acima do mapa, é possível escolher qual rota será exibida no mapa, trocando entre os entregadores. A tabela tem 8 colunas com informações de cada percurso, essas colunas são:

- Endereço Saída: É o endereço que o entregador vai sair, na primeira vez será do depósito. Todos entregadores saem do mesmo endereço de depósito.
- Endereço Chegada: Endereço que o entregador chegará depois que sair do endereço de saída.
- KM: É a distancia em Quilômetros de cada percurso indicada pelo Google Maps.
- Saída: Horário que o entregador sairá para realizar a entrega.
- Percurso: Tempo de locomoção para chegar até o destino de entrega.
- Espera: Se chegar no destino antes do horário de abertura, esse é o tempo que o entregador ficará esperando.
- Entrada: Horário que o entregador conseguiu entrar para começar a realizar a descarga da entrega.
- Descarga: Tempo Médio para descarregar toda a encomenda.

Depois de finalizar a entrega o campo saída deve ser preenchido com o horário que está tudo pronto para fazer a próxima entrega e clicar no botão Próxima Rota, uma nova chamada para a Api será feita somente com o entregador pedido, esse processo pode ser repetido até que todas os endereços sejam visitados.

5 Conclusão

Nesse capítulo é apresentado como os testes foram organizados, as limitações do software, os resultados encontrados e futuras melhorias.

5.1 Limitações

Na criação do software os testes nos iniciais já possível identificar possíveis situações que o software não tem como dar uma resposta, ou seja, suas limitações. Identificamos 3 possíveis limitações nos testes iniciais:

Não é possível entregar a tempo: Levando em consideração o horário de saída e o horário limite para realizar todas as entregas, é possível que um percurso entre os endereços tem seu tempo de trajeto mais demorado que o tempo disponível para a realização de todas as entregas, com isso seria impossível entregar, mesmo com mais entregadores, com isso, o software não consegue definir uma rota por considerar a velocidade média das vias por onde ele passará.

Limite de Entregadores: Para pedir a definição de uma rota é preciso indicar quantos entregadores estão disponível para realizar as entregas, em algumas situações é possível que mesmo dividindo para todos os entregadores, mais entregadores seriam precisos para chegar a tempo em todos os endereços, com isso, o software não consegue definir uma rota.

Tempo limite de entrega excedido: Se a rota não tiver nenhum percurso muito demorado e também for possível determinar a divisão da rota principal entre o numero de entregadores, podemos encontrar outro problema, sempre que um entregador chega a um destino, é possível sofre atrasos na descarga, um maior tempo de espera ou até o transito piorando por causa de um acidente em uma via principal, por exemplo, isso muda o tempo dos próximos percursos, podendo elevar muito o tempo do trajeto. É possível encontrar a situação que seria preciso mais entregadores para finalizar a entrega, que não é mais possível por que o entregador já está em transito, também pode encontrar um percurso completamente parado, com isso, o software não consegue definir uma rota para finalizar as entregas.

5.2 Roteiros dos Testes

Os roteiros foram escolhidos de forma arbitrária com endereços dentro ou próximos da cidade de São Paulo. As listas de endereços e as rotas encontradas podem ser vista a baixo:

O teste da tabela abaixo 3 começa o roteiro em Senac Nacões Unidas, endereço Av. Eng. Eusébio Stevaux, 823 - Santo Amaro, horário de saída 09:00, horário de volta 20:00, com 3 entregadores disponíveis e um tempo de espera médio em cada ponto 20 minutos.

Tabela 3 – Senac

#	Nome	Endereço	Aber.	Fech.
1	Senac Largo Treze	R. Dr. Antônio Bento, 393 - Santo Amaro	09:00	12:00
2	Senac Taboão da Serra	Rua Salvador Branco de Andrade, 182 - Jardim Sao Miguel	09:00	18:00
3	Senac Jabaquara	Av. do Café, 298 - Jabaquara	09:00	11:00
4	Senac Osasco	R. Dante Batiston, 248 - Centro	12:00	15:00
5	Senac Santana	R. Voluntários da Pátria, 3167 - Santana	10:00	19:00
6	Senac Tatuapé	R. Cel. Luís Americano, 130 - Tatuapé	15:00	19:00
7	Senac Vila Prudente	Rua do Orfanato, 316 - Vila Prudente	09:00	17:00
8	Senac - Campos do Jordão	Av. Frei Orestes Girardi, 3549 - Capivari, Campos do Jordão - SP	09:00	17:00

O calculo de rotas dividiu o roteiro para 2 entregadores, mesmo com varias tentativas para conseguir um melhor resultado, não foi possível encontrar uma rota para apenas um entregador, isso é pelo fato que, um dos endereço do roteiro fica em outra cidade, levando muito tempo para ir e voltar da entrega chegando no horário limite de volta para o deposito. Baixo um dos resultados encontrados para desse roteiro.

Entregador 1							
Endereço Saída	Endereço Chegada	KM	Saída	Percurso	Espera	Entrada	Descarga
Av. Eng. Eusébio Stev...	Rua Salvador Branco...	17,220	09:00:00	00:35:01	00:00:00	09:35:01	00:20:00
Rua Salvador Branco...	Av. Eng. Eusébio Stev...	14,511	09:55:01	00:32:19	00:00:00	10:27:20	00:00:00

Figura 10 – Resultado do Entregador 1

Entregador 2							
Endereço Saída	Endereço Chegada	KM	Saída	Percurso	Espera	Entrada	Descarga
Av. Eng. Eusébio Stev...	R. Dr. Antônio Bento,...	3,945	09:00:00	00:13:26	00:00:00	09:13:26	00:20:00
R. Dr. Antônio Bento,...	Av. do Café, 298 - Ja...	9,806	09:33:26	00:26:04	00:00:00	09:59:30	00:20:00
Av. do Café, 298 - Ja...	Rua do Orfanato, 31...	11,269	10:19:30	00:25:28	00:00:00	10:44:58	00:20:00
Rua do Orfanato, 31...	R. Voluntários da Pát...	13,848	11:04:58	00:32:58	00:00:00	11:37:56	00:20:00
R. Voluntários da Pát...	R. Dante Batiston, 24...	21,935	11:57:56	00:37:08	00:00:00	12:35:04	00:20:00
R. Dante Batiston, 24...	Av. Frei Orestes Girar...	197,...	12:55:04	02:37:17	00:00:00	15:32:21	00:20:00
Av. Frei Orestes Girar...	R. Cel. Luís American...	178,...	15:52:21	02:22:37	00:00:00	18:14:58	00:20:00
R. Cel. Luís American...	Av. Eng. Eusébio Stev...	24,827	18:34:58	00:43:15	00:00:00	19:18:13	00:00:00

Figura 11 – Resultado do Entregador 2

O teste da Tabela 4 começa o roteiro no Extra Morumbi, endereço Av. das Nações Unidas, 16741 - Santo Amaro, horário de saída 06:00, horário de volta 22:00:00, com 3 entregadores disponíveis e um tempo de espera médio em cada ponto 60 minutos.

Tabela 4 – Extra

Nome	Endereço	Aber.	Fech.
Extra Hipermercado	R. João Batista de Oliveira, 47 - Centro, Taboão da Serra - SP	09:00	22:00
Extra João Dias	Av. Guido Caloi, 25 - Jardim São Luís, São Paulo - SP	09:00	22:00
Extra Aeroporto	Avenida Washignton Luís, 5859 - Jd. Aeroporto, São Paulo - SP	09:00	22:00
Extra - Itaim Bibi	R. João Cachoeira, 899 - Itaim Bibi, São Paulo - SP	06:00	22:00
Extra - Ricardo Jafet	Av. Dr. Ricardo Jafet, 1501 - Vila Mariana, São Paulo - SP	09:00	22:00
Extra	R. Nossa Sra. das Mercês, 29 - Vila das Merces, São Paulo - SP	09:00	22:00
Extra Hipermercado	Av. Brigadeiro Luís Antônio, 2013 - Bela Vista, São Paulo - SP	09:00	22:00
Extra	Rua Três Rios, 282 - Bom Retiro, São Paulo - SP	09:00	22:00
Extra Hiper Guarapiranga	Av. Guarapiranga, 752 - Socorro, São Paulo - SP	09:00	22:00
Extra - Jardim Angela	Estrada Velha do M'Boi Mirim, 4374 - Jardim Angela, São Paulo - SP	09:00	22:00
Extra Hiper	Av. Sen. Teotônio Vilela, 2926 - Jardim Iporanga, São Paulo - SP	09:00	22:00

O calculo da rota encontrou um rota que pode ser feita por apenas um entregador, de forma que, não foi preciso utilizar todos os disponíveis. Os resultados completos estão nas tabelas abaixo.

Entregador 1							
Endereço Saída	Endereço Chegada	KM	Saída	Percurso	Espera	Entrada	Descarga
Av. das Nações Unid...	R. João Cachoeira, 8...	8,670	06:00:00	00:14:20	00:00:00	06:14:20	01:00:00
R. João Cachoeira, 8...	Av. Sen. Teotônio Vil...	23,386	07:14:20	00:41:57	01:03:43	09:00:00	01:00:00
Av. Sen. Teotônio Vil...	Av. Guarapiranga, 75...	8,158	10:00:00	00:23:45	00:00:00	10:23:45	01:00:00
Av. Guarapiranga, 75...	Estrada Velha do M'...	5,828	11:23:45	00:16:09	00:00:00	11:39:54	01:00:00
Estrada Velha do M'...	Av. Guido Caloi, 25 - ...	7,211	12:39:54	00:20:04	00:00:00	12:59:58	01:00:00
Av. Guido Caloi, 25 - ...	R. João Batista de Oli...	6,786	13:59:58	00:17:54	00:00:00	14:17:52	01:00:00
R. João Batista de Oli...	Av. Brigadeiro Luís A...	14,267	15:17:52	00:43:17	00:00:00	16:01:09	01:00:00
Av. Brigadeiro Luís A...	Rua Três Rios, 282 - ...	5,715	17:01:09	00:18:30	00:00:00	17:19:39	01:00:00
Rua Três Rios, 282 - ...	Av. Dr. Ricardo Jafet, ...	9,804	18:19:39	00:24:52	00:00:00	18:44:31	01:00:00
Av. Dr. Ricardo Jafet, ...	R. Nossa Sra. das Me...	4,939	19:44:31	00:13:09	00:00:00	19:57:40	01:00:00
R. Nossa Sra. das Me...	Avenida Washignton...	9,564	20:57:40	00:16:29	00:00:00	21:14:09	01:00:00
Avenida Washignton...	Av. das Nações Unid...	9,585	22:14:09	00:14:35	00:00:00	22:28:44	00:00:00

Figura 12 – Resultado do Entregador 1

O teste da Tabela 5 começa o roteiro no McDonald's Jardim Paulista, endereço R. Pamplona, 734 - Jardim Paulista, São Paulo - SP, horário de saída 09:00, horário de volta 23:00, com 3 entregadores disponíveis e um tempo de espera médio em cada ponto 40 minutos.

Tabela 5 – MacDonal'd's

Nome	Endereço	Aber.	Fech.
McDonald's Augusta	R. Augusta, 1856 - Cerqueira César, São Paulo - SP	08:00	23:00
McDonald's Brigadeiro	Av. Brigadeiro Luís Antônio, 3477/3481 - Jardim Paulista, São Paulo - SP	09:00	19:00
McDonald's José Maria	Av. José Maria Whitaker, 81 - Jardim Paulista, São Paulo - SP	09:00	21:00
McDonald's Nações Unidas	Av. das Nações Unidas, 12555 - Pinheiros, São Paulo - SP	06:00	21:00
McDonald's Eliseu de Almeida	Av. Eliseu de Almeida, 2700 - Jardim Peri, São Paulo - SP	09:00	18:00
McDonald's Vital Brasil	Av. Vital Brasil, 1256 - Butantã, São Paulo - SP	09:00	18:00
McDonald's Henrique Schaumann	Rua Henrique Schaumann, 80/124 - Cerqueira César, São Paulo - SP	09:00	23:00
McDonald's Santo Antônio	Av. Roque Petroni Júnior, 1089 - Chácara Santo Antônio (Zona Sul), São Paulo - SP	09:00	23:00

O calculo da rota dividiu o roteiro para apenas um entregador, de forma que, não foi preciso utilizar todos os 3 disponíveis. O resultado completo podem ser visto na tabela abaixo.

Entregador 1							
Endereço Saída	Endereço Chegada	KM	Saída	Percurso	Espera	Entrada	Descarga
R. Pamplona, 734 - J...	Av. Brigadeiro Luís A...	1,708	09:00:00	00:06:47	00:00:00	09:06:47	00:40:00
Av. Brigadeiro Luís A...	Av. José Maria Whita...	3,808	09:46:47	00:11:44	00:00:00	09:58:31	00:40:00
Av. José Maria Whita...	Av. Roque Petroni Jú...	11,152	10:38:31	00:26:36	00:00:00	11:05:07	00:40:00
Av. Roque Petroni Jú...	Av. Eliseu de Almeid...	8,470	11:45:07	00:23:13	00:00:00	12:08:20	00:40:00
Av. Eliseu de Almeid...	Av. Vital Brasil, 1256 ...	4,839	12:48:20	00:12:04	00:00:00	13:00:24	00:40:00
Av. Vital Brasil, 1256 ...	Av. das Nações Unid...	2,915	13:40:24	00:12:07	00:00:00	13:52:31	00:40:00
Av. das Nações Unid...	Rua Henrique Schau...	3,919	14:32:31	00:12:44	00:00:00	14:45:15	00:40:00
Rua Henrique Schau...	R. Augusta, 1856 - C...	2,886	15:25:15	00:11:21	00:00:00	15:36:36	00:40:00
R. Augusta, 1856 - C...	R. Pamplona, 734 - J...	1,174	16:16:36	00:05:00	00:00:00	16:21:36	00:00:00

Figura 13 – Resultado do Entregador 1

O teste da Tabela 6 começa o roteiro no Uninove Vergueiro, endereço Rua Vergueiro, 235/249 - Liberdade, São Paulo - SP, horário de saída 08:00, horário de volta 23:00:00, com 5 entregadores disponíveis e um tempo de espera médio em cada ponto 90 minutos.

Tabela 6 – Uninove

Nome	Endereço	Aber.	Fech.
Uninove Osasco	R. Dante Batiston, 87 - Centro, Osasco - SP	08:00	23:00
Uninove Santo Amaro	R. Amador Bueno - Santo Amaro, São Paulo - SP	08:00	23:00
Uninove São Bernardo do Campo	Av. Dom Jaime de Barros Câmara, 90 - Planalto, São Bernardo do Campo - SP	08:00	23:00
Uninove Vila Prudente	Av. Professor Luiz Ignácio Anhaia Mello, 1363 - Vila Prudente, São Paulo - SP	08:00	23:00
Uninove Vila Maria Baixa	R. Itauna, 74 - Vila Maria Baixa, São Paulo - SP	08:00	23:00
Uninove Barra Funda	Av. Dr. Adolpho Pinto, 109 - Barra Funda, São Paulo - SP	08:00	23:00
Uninove Mauá	R. Álvares Machado, 48 - Vila Bocaina, Mauá - SP	08:00	23:00
Uninove Santo André	R. Princesa Isabel - Vila Guiomar, Santo André - SP	08:00	23:00

O calculo da rota dividiu o roteiro para 2 dos 5 entregadores disponíveis. O resultado completo podem ser visto nas tabelas abaixo.

Entregador 1							
Endereço Saída	Endereço Chegada	KM	Saída	Percurso	Espera	Entrada	Descarga
Rua Vergueiro, 235/2...	Av. Dr. Adolpho Pint...	7,880	08:00:00	00:25:50	00:00:00	08:25:50	01:30:00
Av. Dr. Adolpho Pint...	R. Dante Batiston, 87...	18,307	09:55:50	00:29:50	00:00:00	10:25:40	01:30:00
R. Dante Batiston, 87...	R. Amador Bueno - S...	23,527	11:55:40	00:27:27	00:00:00	12:23:07	01:30:00
R. Amador Bueno - S...	Rua Vergueiro, 235/2...	16,771	13:53:07	00:35:59	00:00:00	14:29:06	00:00:00

Figura 14 – Resultado do Entregador 1

Entregador 2							
Endereço Saída	Endereço Chegada	KM	Saída	Percurso	Espera	Entrada	Descarga
Rua Vergueiro, 235/2...	R. Itauna, 74 - Vila M...	11,186	08:00:00	00:24:18	00:00:00	08:24:18	01:30:00
R. Itauna, 74 - Vila M...	Av. Professor Luiz Ig...	10,179	09:54:18	00:20:32	00:00:00	10:14:50	01:30:00
Av. Professor Luiz Ig...	R. Álvares Machado, ...	21,753	11:44:50	00:42:31	00:00:00	12:27:21	01:30:00
R. Álvares Machado, ...	R. Princesa Isabel - V...	11,591	13:57:21	00:23:58	00:00:00	14:21:19	01:30:00
R. Princesa Isabel - V...	Av. Dom Jaime de Ba...	9,448	15:51:19	00:11:19	00:00:00	16:02:38	01:30:00
Av. Dom Jaime de Ba...	Rua Vergueiro, 235/2...	21,509	17:32:38	00:44:55	00:00:00	18:17:33	00:00:00

Figura 15 – Resultado do Entregador 2

O teste da Tabela 7 simula a situação de 1 turista hospedado no Hotel Ibis São Paulo Paulista no endereço Av. Paulista, 2355 - Bela Vista, São Paulo - SP, em seu plano é conheço os endereços, permanecendo 30 minutos em cada, saindo do hotel 09:00 e voltando 23:00.

Tabela 7 – Pontos Turísticos de São Paulo

Nome	Endereço	Aber.	Fech.
Catedral Metropolitana de São Paulo	Praça da Sé - Sé, São Paulo - SP	00:00	23:59
Pte. Estaiada	Av. Jorn. Roberto Marinho, 85 - Cidade Monções, São Paulo - SP	00:00	23:59
Museu Catavento	Pq. Dom Pedro II - Av. Mercúrio, s/n - Brás, São Paulo - SP	09:00	16:00
Aquário de São Paulo	R. Huet Bacelar, 407 - Ipiranga, São Paulo - SP	09:00	17:00
Museu do Ipiranga	Parque da Independência - s/n - Ipiranga, São Paulo - SP	09:00	17:00
Parque Ibirapuera	Av. Pedro Álvares Cabral - Vila Mariana, São Paulo - SP	05:00	23:59
Zoológico De Sao Paulo	Av. Miguel Estefno, 4241 - Vila Santo Estefano, São Paulo - SP	09:00	19:00
Jardim Botânico de São Paulo	Av. Miguel Estefno, 3031 - Vila Água Funda, São Paulo - SP	09:00	17:00
Pateo do Collegio	Pç. Pateo do Collegio, 2 - Centro, São Paulo - SP	09:00	16:30
Parque Estadual Alberto Löfgren	R. do Horto, 931 - Horto Florestal, São Paulo - SP	06:00	18:00
Parque Estadual do Jaraguá	R. Antônio Cardoso Nogueira, 539 - Vila Chica Luisa, São Paulo - SP	07:00	17:00
Autódromo de Interlagos	Av. Sen. Teotônio Vilela, 261 - Interlagos, São Paulo - SP	07:00	17:00
Anhembi Sambadrome	Av. Olavo Fontoura, 1209 - Santana, São Paulo - SP	07:00	17:00

Mesmo com varias tentativas, o calculador de rotas não encontrou um rota onde seria possível com apenas uma pessoa em um dia, visitar todos os endereços do roteiro. Alterando para um tempo de permanência menor de 30 minutos, depois de varias tentaria, foi possível obter um resultado que pode ser visto a baixo.

Entregador 1							
Endereço Saída	Endereço Chegada	KM	Saída	Percurso	Espera	Entrada	Descarga
Av. Paulista, 2355 - B...	Pç. Pateo do Colégi...	4,329	09:00:00	00:16:00	00:00:00	09:16:00	00:30:00
Pç. Pateo do Colégi...	Pq. Dom Pedro II - A...	2,217	09:46:00	00:09:46	00:00:00	09:55:46	00:30:00
Pq. Dom Pedro II - A...	Av. Olavo Fontoura, ...	5,237	10:25:46	00:12:11	00:00:00	10:37:57	00:30:00
Av. Olavo Fontoura, ...	R. Antônio Cardoso ...	18,571	11:07:57	00:26:32	00:00:00	11:34:29	00:30:00
R. Antônio Cardoso ...	Av. Jorn. Roberto Ma...	23,481	12:04:29	00:35:12	00:00:00	12:39:41	00:30:00
Av. Jorn. Roberto Ma...	Av. Sen. Teotônio Vil...	14,911	13:09:41	00:29:39	00:00:00	13:39:20	00:30:00
Av. Sen. Teotônio Vil...	Av. Miguel Estefno, 4...	18,010	14:09:20	00:35:46	00:00:00	14:45:06	00:30:00
Av. Miguel Estefno, 4...	Av. Miguel Estefno, 3...	1,444	15:15:06	00:02:25	00:00:00	15:17:31	00:30:00
Av. Miguel Estefno, 3...	R. Huet Bacelar, 407 ...	5,810	15:47:31	00:13:00	00:00:00	16:00:31	00:30:00
R. Huet Bacelar, 407 ...	Parque da Independ...	3,091	16:30:31	00:08:01	00:00:00	16:38:32	00:30:00
Parque da Independ...	R. do Horto, 931 - H...	15,974	17:08:32	00:41:16	00:00:00	17:49:48	00:30:00
R. do Horto, 931 - H...	Praça da Sé - Sé, São...	13,829	18:19:48	00:39:30	00:00:00	18:59:18	00:30:00
Praça da Sé - Sé, São...	Av. Pedro Álvares Ca...	7,596	19:29:18	00:13:26	00:00:00	19:42:44	00:30:00
Av. Pedro Álvares Ca...	Av. Paulista, 2355 - B...	4,775	20:12:44	00:16:31	00:00:00	20:29:15	00:00:00

Figura 16 – Resultado do Turista

Mesmo com o foco em entregas, o software poderia ser utilizado para outras funções, mesmo parecendo simples encontrar um roteiro que de para ver todos os pontos de interesse, se torna um trabalho complicado com você tem horários e tempo de percurso. O software por utilizar GA para encontrar a rota, teve dificuldade para encontrar uma rota que seria possível para apenas uma pessoa passar por todos esses lugares.

5.3 Comparativo

O software se mostrou eficiente para entrar uma rota que minimiza o numero de entregadores e respeitando os horários de cada endereço. Roteiros que, a primeira vista, parecem impossíveis de ser entregue, rotas foram encontradas e organizadas. O Google Maps ajuda na definição da rota por entregar o transito médio da rota fazendo com que o caminho escolhido pelo software fique mais próximo de uma situação real.

O GA ajuda na escolhas das rotas por exemplo tentar minimizar o tempo, a distância e o numero de entregadores, encontrando padrões difíceis de ser vistos, porém, para encontrar uma solução próxima do ótimo é preciso perder performance para calcular a resposta, aumentando o numero de gerações e população para encontrar a rota.

Utilizando um numero baixo de destinos o processo de recálculo de rotas para cada vez que chegar em um destino, se mostra útil para identificar mudanças no transito e ainda chegar no horário proposto. Em casos que é preciso calcular para muitos destinos, é mais recomendado somente utilizar para uma divisão de tarefas ou pré-analise do roteiro, por tornar a analise muito demora, o tamanho da população e numero de gerações precisa ser mais altos para melhor precisão dos resultados.

Para comparativo foi utilizado os diferentes tipos de mutação e cruzamentos, de forma, a identificar uma melhor combinação para ser utilizada no software, 10 vezes para cada combinação utilizando a média do valor do valor de aptidão. A tabela abaixo estão os resultados encontrados:

Tabela 8 – Comparação dos Cruzamentos e Mutações

Extras.txt	EM	DIVM	DM	IM	IVM	SM
OBX	109560	109560	109560	109560	109560	109560
PBX	109560	109560	109560	109560	109560	109560
MacDo-nalts.txt	EM	DIVM	DM	IM	IVM	SM
OBX	43320	43320	43320	43320	43320	43320
PBX	43320	43320	43320	43320	43320	43320
Senacs.txt	EM	DIVM	DM	IM	IVM	SM
OBX	311876	311876	311876	311876	311876	311876
PBX	311876	311876	311876	311876	311876	311876
Unino-ves.txt	EM	DIVM	DM	IM	IVM	SM
OBX	190656	190656	190656	190656	190656	190656
PBX	190656	190656	190656	190656	190656	190656
Turisti-cos.txt	EM	DIVM	DM	IM	IVM	SM
OBX	84573	84573	84573	84573	84573	84573
PBX	84573	84573	84573	84573	84573	84573

Como mostrado na tabela 8, a mutação ou cruzamento escolhida não influencia no resultado, apresentando o mesmo valor de aptidão para todas as combinações em nos diferentes roteiros.

5.4 Trabalhos futuros

Devido a média de tempo para se encontrar as rotas, avaliar a possível paralelização da rotina de algoritmos genéticos para tornar o projeto viável para uso em uma aplicação web. Avaliar a aplicação de heurísticas λ – *interchange*, *k-node*, *Or-opt* no calculo do *Fitness*. Utilizar uma base de dados maior para os testes, baseada em casos reais de logística. Criar uma aplicação web para utilização em um caso real de entregas.

Referências

ALVARENGA, G. B. Um algoritmo híbrido para os problemas de roteamento de veículos estático e dinâmico com janela de tempo. Universidade Federal de Minas Gerais, 2005. Disponível em: <http://www.bibliotecadigital.ufmg.br/dspace/bitstream/handle/1843/RVMR-6EAKH8/guilherme_bastos.pdf?sequence=1>. Citado na página 11.

ANTES, J.; DERIGS, U. A new parallel tour construction algorithm for the vehicle routing problem with time windows. 12 1997. Citado na página 13.

BAKER, E. K.; SCHAFFER, J. R. Solution improvement heuristics for the vehicle routing and scheduling problem with time window constraints. *American Journal of Mathematical and Management Sciences*, v. 6, n. 3-4, p. 261–300, 1986. Disponível em: <<http://dx.doi.org/10.1080/01966324.1986.10737197>>. Citado na página 12.

BALLOU, R. H. Book. *Business logistics management [by] Ronald H. Ballou*. [S.l.]: Prentice-Hall Englewood Cliffs, N.J, 1973. xiii, 514 p. p. ISBN 0131048023. Citado na página 5.

BRÄYSY, O.; DULLAERT, W.; GENDREAU, M. Evolutionary algorithms for the vehicle routing problem with time windows. *Journal of Heuristics*, v. 10, n. 6, p. 587–611, Dec 2004. ISSN 1572-9397. Disponível em: <<https://doi.org/10.1007/s10732-005-5431-6>>. Citado 2 vezes nas páginas 6 e 17.

BRäYSY, O.; GENDREAU, M. Vehicle routing problem with time windows, part i: Route construction and local search algorithms. *Transportation Science*, v. 39, n. 1, p. 104–118, 2005. Disponível em: <<https://pubsonline.informs.org/doi/abs/10.1287/trsc.1030.0056>>. Citado na página 13.

BRäYSY, O.; HASLE, G.; DULLAERT, W. A multi-start local search algorithm for the vehicle routing problem with time windows. *European Journal of Operational Research*, Elsevier, v. 159, n. 3, p. 586–605, 12 2004. ISSN 0377-2217. Citado na página 16.

CHABRIER, A. Vehicle routing problem with elementary shortest path based column generation. *Comput. Oper. Res.*, Elsevier Science Ltd., Oxford, UK, UK, v. 33, n. 10, p. 2972–2990, out. 2006. ISSN 0305-0548. Disponível em: <<http://dx.doi.org/10.1016/j.cor.2005.02.029>>. Citado 2 vezes nas páginas 6 e 11.

CHIANG, W.-C.; RUSSELL, R. A. Simulated annealing metaheuristics for the vehicle routing problem with time windows. *Annals of Operations Research*, v. 63, n. 1, p. 3–27, Feb 1996. ISSN 1572-9338. Disponível em: <<https://doi.org/10.1007/BF02601637>>. Citado na página 16.

CHRISTOFIDES, N.; BEASLEY, J. E. The period routing problem. *Networks*, Wiley Subscription Services, Inc., A Wiley Company, v. 14, n. 2, p. 237–256, 1984. ISSN 1097-0037. Disponível em: <<http://dx.doi.org/10.1002/net.3230140205>>. Citado na página 14.

- CLARKE, G.; WRIGHT, J. W. Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, v. 12, n. 4, p. 568–581, 1964. Disponível em: <<https://doi.org/10.1287/opre.12.4.568>>. Citado na página 12.
- CORDEAU, J.-F. et al. 7. vrp with time windows. In: _____. *The Vehicle Routing Problem*. [s.n.]. p. 157–193. Disponível em: <<http://epubs.siam.org/doi/abs/10.1137/1.9780898718515.ch7>>. Citado na página 11.
- DIAS, M. A. P. *Administração de materiais: uma abordagem logística*. [S.l.: s.n.], 2010. Citado na página 5.
- EL-SHERBENY, N. A. Vehicle routing with time windows: An overview of exact, heuristic and metaheuristic methods. *Journal of King Saud University - Science*, v. 22, n. 3, p. 123 – 131, 2010. ISSN 1018-3647. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1018364710000297>>. Citado 2 vezes nas páginas 12 e 13.
- GARCIA, B.-L.; POTVIN, J.-Y.; ROUSSEAU, J.-M. A parallel implementation of the tabu search heuristic for vehicle routing problems with time window constraints. *Computers & Operations Research*, v. 21, n. 9, p. 1025 – 1033, 1994. ISSN 0305-0548. Disponível em: <<http://www.sciencedirect.com/science/article/pii/0305054894900736>>. Citado na página 16.
- GENDREAU, M. et al. Parallel tabu search for real-time vehicle routing and dispatching. *Transportation Science*, v. 33, n. 4, p. 381–390, 1999. Disponível em: <<https://doi.org/10.1287/trsc.33.4.381>>. Citado na página 27.
- GLOVER, F. Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research*, v. 13, n. 5, p. 533 – 549, 1986. ISSN 0305-0548. Applications of Integer Programming. Disponível em: <<http://www.sciencedirect.com/science/article/pii/0305054886900481>>. Citado na página 16.
- Google Inc. *Google Distance Matrix API*. 2018. [Online; accessed Maio 01, 2018]. Disponível em: <<https://developers.google.com/maps/documentation/distance-matrix/?hl=pt-br>>. Citado na página 25.
- HOLLAND, J. *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI, USA: University of Michigan Press, 1975. Disponível em: <<http://books.google.com/books?id=YE5RAAAAMAAJ>>. Citado na página 18.
- HOMBERGER. A two-phase hybrid metaheuristic for the vehicle routing problem with time windows. *European Journal of Operational Research*, v. 162, n. 1, p. 220–238, 2005. Disponível em: <<https://EconPapers.repec.org/RePEc:eee:ejores:v:162:y:2005:i:1:p:220-238>>. Citado na página 16.
- JEPSEN, M.; SPOORENDONK, S. A non-robust branch-and-cut-and-price algorithm for the vehicle routing problem with time windows. 01 2006. Citado na página 11.
- JUNG, S.; MOON, B.-R. A hybrid genetic algorithm for the vehicle routing problem with time windows. In: *Proceedings of the 4th Annual Conference on Genetic and Evolutionary Computation*. San Francisco, CA, USA: Morgan Kaufmann

- Publishers Inc., 2002. (GECCO'02), p. 1309–1316. ISBN 1-55860-878-8. Disponível em: <<http://dl.acm.org/citation.cfm?id=2955491.2955733>>. Citado na página 17.
- KARP, R. M. On the computational complexity of combinatorial problems. 1975. Citado na página 5.
- KOHL, N. Exact methods for time constrained routing and related scheduling problems. Richard Petersens Plads, Building 321, 2800 Kgs. Lyngby, p. 234, 1995. Disponível em: <<http://www2.imm.dtu.dk/pubdb/p.php?2100>>. Citado na página 11.
- LANDEGHEM, H. V. A bi-criteria heuristic for the vehicle routing problem with time windows. *European Journal of Operational Research*, v. 36, n. 2, p. 217 – 226, 1988. ISSN 0377-2217. Disponível em: <<http://www.sciencedirect.com/science/article/pii/0377221788904286>>. Citado na página 12.
- LORENA, G. M. R. e L. A. N. Roteamento de veículos dinâmico usando algoritmos genéticos. Citado na página 24.
- LUCAS, D. C. Algoritmos genéticos: uma introdução. Universidade Federal do Rio Grande do Sul, 2002. Disponível em: <<http://www.inf.ufrgs.br/~alvares/INF01048IA/ApostilaAlgoritmosGeneticos.pdf>>. Citado 3 vezes nas páginas 18, 19 e 21.
- OLIVEIRA, G. C. V. e G. B. A. Humberto César Brandão de. Uma abordagem evolucionária para o problema de roteamento de veículos com janela de tempo. 2005. Citado na página 24.
- OLIVEIRA, H. C. B. de. Algoritmo evolutivo no tratamento do problema de roteamento de veículos com janela de tempo. 2005. Citado na página 5.
- OLIVEIRA, W. A. de. Algoritmo genético para o problema de rotas de cobertura multiveículo. 2009. Citado na página 23.
- OSMAN, I. H. Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals of Operations Research*, v. 41, n. 4, p. 421–451, Dec 1993. ISSN 1572-9338. Disponível em: <<https://doi.org/10.1007/BF02023004>>. Citado na página 14.
- POTVIN, J.-Y.; ROBILLARD, C. Clustering for vehicle routing with a competitive neural network. *Neurocomputing*, v. 8, n. 2, p. 125 – 139, 1995. ISSN 0925-2312. Optimization and Combinatorics, Part II. Disponível em: <<http://www.sciencedirect.com/science/article/pii/092523129400012H>>. Citado na página 14.
- POTVIN, J.-Y.; ROUSSEAU, J.-M. A parallel route building algorithm for the vehicle routing and scheduling problem with time windows. *European Journal of Operational Research*, v. 66, n. 3, p. 331 – 340, 1993. ISSN 0377-2217. Disponível em: <<http://www.sciencedirect.com/science/article/pii/0377221793902218>>. Citado na página 13.
- RIBAS, S. Um algoritmo híbrido para o problema de roteamento de veículos com janelas de tempo. 2011. Citado na página 24.
- RIBEIRO, C. C. Metaheuristics and applications. In: _____. *Advanced School on Artificial Intelligence*. [S.l.: s.n.], 1996. Citado na página 16.

- RODRIGUES, P. R. A. *Introdução aos sistemas de transporte do Brasil e à logística internacional*. [S.l.: s.n.], 2007. Citado na página 5.
- SCHULZE, J.; FAHLE, T. A parallel algorithm for the vehicle routing problem with time window constraints. *Annals of Operations Research*, v. 86, n. 0, p. 585–607, Jan 1999. ISSN 1572-9338. Disponível em: <<https://doi.org/10.1023/A:1018948011707>>. Citado na página 14.
- SOLOMON, M. M. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, v. 35, n. 2, p. 254–265, 1987. Disponível em: <<https://doi.org/10.1287/opre.35.2.254>>. Citado 2 vezes nas páginas 12 e 17.
- SOUZA, M. J. F. 2011. Disponível em: <<http://www.decom.ufop.br/marcone/Disciplinas/InteligenciaComputacional/InteligenciaComputacional.htm>>. Acesso em: 01/12/2017. Citado na página 16.
- TOMTOM INTERNATIONAL BV. 2017. Disponível em: <https://www.tomtom.com/en_gb/trafficindex/list?citySize=LARGEcontinent=ALL&country=BR>. Acesso em: 12/11/2017. Citado na página 25.
- TSUDA, D. S. Modelo de roteirização de veículos em uma empresa importadora de produtos japoneses. 2007. Citado na página 5.
- YVES ROCHAT, E. Probabilistic diversification and intensification in local search for vehicle routing. 1995. Citado na página 9.