

ROTEIRIZAÇÃO DINÂMICA DE VEÍCULOS EM ATIVIDADES DE PRESTAÇÃO DE SERVIÇOS NO SETOR ELÉTRICO

Leonardo K. Mariano da Rocha

leokurtz@gmail.com, UFRGS

Resumo

Neste trabalho é proposta uma metodologia de roteirização de veículos que atenda a atividade de manutenção corretiva da rede para o setor elétrico. Para tal, foi adaptado um algoritmo elaborado para uma empresa de ambiente com características semelhantes ao da empresa usada como modelo. O algoritmo utiliza de uma heurística de inserção, e modifica as rotas em tempo real. Os experimentos realizados indicam que o tratamento dinâmico traz vantagens em nível de serviço.

Palavras-chave:

Roteirização dinâmica de veículos; Heurística; Setor elétrico.

1. Introdução

A economia brasileira, que nos últimos anos tem-se mantido alheia a crises externas, cresce de forma rápida e contínua. Este crescimento vem acompanhado por uma melhor distribuição de renda, o que, acompanhado por fatores como a globalização e a popularização de tecnologias voltadas para o conforto, resulta na crescente qualidade de vida do brasileiro. O aumento populacional causa também um aumento potencial da demanda por serviços básicos (coleta de lixo, distribuição de correspondências, eletricidade, saneamento básico, polícia, bombeiros, dentre outros), o que pode resultar em congestão do sistema, afetando o nível de serviço (MENDOÇA; MORABITO, 2000). Com o grau de exigência dos consumidores cada vez maior, um nível de serviço elevado passa ser prioritário no reparo da rede elétrica, uma vez que grande parte do aumento da qualidade de vida é possibilitada pela eletricidade. Assim a interrupção do serviço precisa ser tratada como uma emergência, sob o risco de sofrer perdas não só financeiras como também elevado prejuízo na confiabilidade. Em áreas de concessão com grande parte dos consumidores localizados no meio rural o tempo gasto no transporte passa a ser tão significativo quanto o tempo de reparo propriamente dito, uma vez que os deslocamentos são maiores do que os realizados no meio urbano.

A impossibilidade de alocar um veículo para cada ordem cria a necessidade de designar várias ordens de serviço para cada veículo. Uma vez que a produtividade dos trabalhadores das concessionárias de energia elétrica é diretamente relacionada à eficiência do transporte, o processo de encontrar os melhores trajetos que um veículo deve fazer passa a ser

fundamental para minimizar os deslocamentos. Esse processo é conhecido na literatura como roteirização (BALLOU, 1999). Grande parte das soluções para problemas de roteirização são estáticos (SALVESBERGH; SOL, 1995), situação onde demanda e localizações dos clientes são conhecidas no momento do planejamento do trajeto. Em problemas dinâmicos, as rotas são reprogramadas após a saída dos veículos do depósito, devido a entrada de novas informações.

O tratamento dinâmico é adequado para solucionar problemas emergenciais, como situações de falta de energia elétrica. Entretanto, a maior parte dos estudos de roteirização em serviços emergenciais encontrados é voltada para sistemas médicos (GENDREAU; LAPORTE; SEMET, 1997; 2001; IANNONI; MORABITO, 2006; 2008), e no caso dos serviços de manutenção da rede de energia elétrica o custo passa a ter uma importância ainda maior, dado que geralmente a eficiência não é uma questão de vida ou morte, fazendo-se necessário estabelecer um ponto ótimo na relação entre nível de serviço e custo logístico.

Segundo Kagan, Oliveira e Robba (2005) a qualidade do serviço de uma empresa de distribuição de energia elétrica pode ser entendida como continuidade de fornecimento, e neste cenário os indicadores de continuidade, que medem os períodos com falta de serviço as atividades de manutenção (geradas por falhas do sistema ou programadas), são a principal forma de medir a qualidade do serviço. Com a reestruturação do setor elétrico brasileiro (SEB) e introdução da análise comparativa de desempenho (*yardstick competition*) na resolução nº024 de 2000 da agência nacional de energia elétrica (ANEEL), o atendimento das metas estabelecidas vem sendo um desafio crescente para as empresas de fornecimento de energia elétrica, que encontram dificuldades para adaptarem-se a nova realidade. O interesse principal deste artigo é propor uma metodologia de roteirização de veículos que atenda o setor elétrico.

Este artigo é composto por seis seções. Na seção 2 descreve-se o problema de roteirização dinâmico e os prazos de atendimento do setor elétrico. Na seção 3 é apresentado o caso estudado em detalhe e a metodologia empregada na pesquisa. Na seção 4 serão apresentados os resultados obtidos pela aplicação da metodologia e finalmente, na seção 5 serão apresentadas as principais conclusões do artigo e perspectivas para pesquisas futuras.

2. Referencial Teórico

Problemas de roteirização de veículos (PRVs) podem ser classificados tanto em dependência do tempo (estáticos ou dinâmicos) quanto em disponibilidade de informações (determinístico ou estocástico). Um problema é dito estático e determinístico se todos os

dados são conhecidos antes dos veículos partirem dos seus pontos de origem e as rotas não são replanejadas ao longo de sua execução. No PRV dinâmico e estocástico as rotas são modificadas durante a execução das mesmas, pois nem todas as informações são conhecidas durante o planejamento inicial, e informações conhecidas podem mudar após a construção das rotas. O grau de dinamismo de um problema pode variar de fraco, onde mais de 80% dos clientes são conhecidos no momento do planejamento das rotas, até forte, no caso de serviços puramente emergenciais, como bombeiros ou ambulâncias (GHIANI et al., 2003; LARSEN, 2000).

Quanto à abordagem de resolução para problemas estocásticos e/ou dinâmicos, Larsen (2000) classifica os problemas em *a priori* e otimização em tempo real. No primeiro caso é determinada uma solução baseada em dados probabilísticos sobre um futuro incerto, tais como tempos de viagens e demanda de clientes. Correções são aplicadas conforme a confirmação ou não dos dados considerados, por exemplo, se a demanda de um ou mais clientes não se confirma, os pontos de demanda nula são ignorados, e a rota continua normalmente.

Métodos de otimização em tempo real, por outro lado, utilizam somente dados reais, modificando as rotas com os veículos em trânsito, devido à chegada de novos pedidos ou alteração das solicitações existentes. Resolver estes problemas com soluções matemáticas exatas só é possível para casos simples, pois o aumento do número de clientes e/ou veículos aumenta a complexidade do problema exponencialmente, o que resulta em tempos computacionais demasiadamente elevados. Mais detalhes sobre a abordagem matemática exata pode ser encontrada em Wagner (1986). Como a maioria dos casos práticos são de tempo polinomial não determinístico e difíceis (NP hard), a abordagem heurística é mais utilizada. Dentre os métodos heurísticos para resolver o problema em questão pode-se citar a busca local (FLEISCHMANN; GNUTZMANN; SANDVOß, 2004), as heurísticas de inserção (PUREZA; LAPORTE, 2008), a busca tabu ((GENDREAU; LAPORTE; SEMET, 1997; 2001) e os algoritmos genéticos (IANNONI; MORABITO, 2006; 2008)).

A Figura 1 mostra um exemplo de replanejamento de rotas num problema dinâmico. Na situação **a** temos as rotas planejadas antes do início do expediente. A situação **b** simula a chegada de um novo pedido ao sistema duas horas depois. É interessante observar que os veículos não se encontram mais na base, portanto já cumpriram parte de suas rotas e a situação **b** apresenta um maior aproveitamento dos deslocamentos.

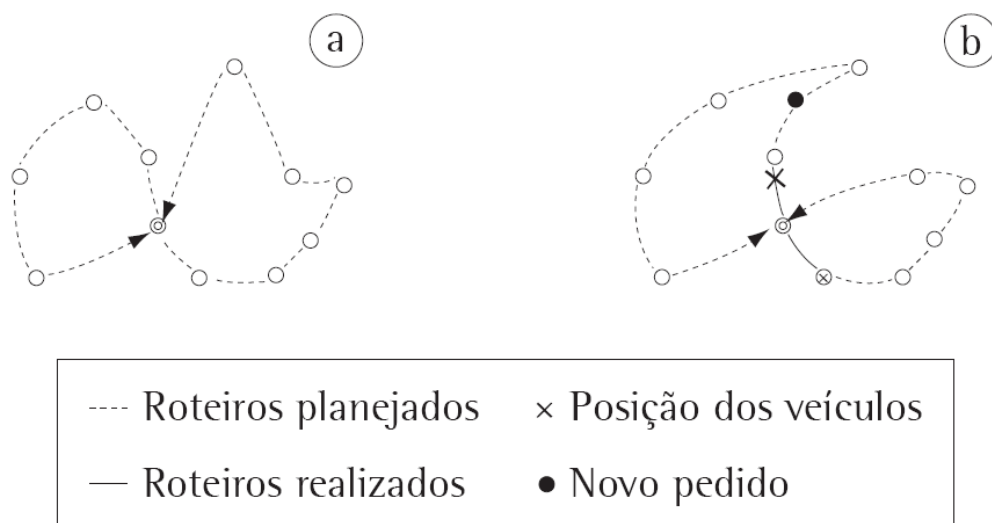


Figura 1 - Planejamento dinâmico de rotas: a) rotas planejadas em $t = 7h45min$; b) chegada de um novo pedido ao sistema em $t = 9h45min$ e replanejamento das rotas (Pureza e Lazarin, 2010).

Tratar este problema como um modelo dinâmico se torna viável graças a tecnologias essenciais, como um rádio para informar a posição do veículo a central, ou idealmente um sistema GPS. O aparelho celular também pode servir como alternativa para substituir o rádio em algumas situações. Os aparelhos GPS se tornaram mais acessíveis nos últimos dez anos e possibilitam uma informação muito mais precisa, fornecendo a localização exata do veículo para o sistema, se utilizado em conjunto com um sistema de informações geográficas (SIG). Os SIGs hoje mapeiam grande parte do território e incluem informações detalhadas sobre as vias (LARSEN; MADSEN; SOLOMON, 2008; PSARAFTIS, 1995).

2.1. O algoritmo DINÂMICO de Pureza e Lazarin

Pureza e Laporte (2008) testaram estratégias de espera e armazenamento através da aplicação de uma heurística construtiva-desconstrutiva e Lazarin (2008) adaptou um dos algoritmos testados (chamado de WEF dinâmico) para o caso de uma empresa de bebidas que prestava serviços de manutenção. Os passos do módulo de despacho são apresentados no Quadro 1.

1. Inicialização: Faça $t = 0$, $P = \emptyset$, $PP = \emptyset$, e $PP^* = \emptyset$.
2. Enquanto $t \leq l_0$, (limitante superior da janela de tempo do depósito):
 - 2.1. Faça $t =$ instante de ocorrência do próximo evento e $r =$ número de rotas na solução corrente.
 - 2.2. Se $t = 0$ e $r = 0$ (o evento corresponde à chegada de um ou mais pedidos), inclua estes pedidos em P e vá para o passo 2.5.
Caso contrário, se $t > 0$ e $r > 0$, faça $S_0 = S$ (armazene a solução corrente), e avalie o status de cada localidade da solução S como se segue:
 - a) Permanentemente designadas a uma rota e posição na rota: localidades que já foram servidas, ou que foram fixadas deliberadamente.
 - b) Permanentemente designadas a uma rota: localidades que precisam ser mantidas na rota atual para manutenção da factibilidade temporal, especificadamente, para garantir a chegada ao depósito até o fim de sua janela de tempo.
 - c) Não permanentemente designadas: localidades que não satisfazem os casos (a) e (b).
 - 2.3. Desconstrua a solução corrente S (obtendo uma nova solução S), pela remoção das localidades não permanentemente designadas. Inclua os pedidos das localidades retiradas em P .
 - 2.4. Se o evento corresponde à chegada de um novo pedido p , faça, $P = P + \{p\}$.
 - 2.5. Planejamento de rotas: Construa rotas com os pedidos em P , utilizando a heurística de inserção descrita no Quadro 2. Seja S o conjunto de rotas resultantes.
 - 2.6. Se algum pedido $p_f \in PP$ não tiver sido reinserido em S ou se algum pedido $p_f \in PP^*$ tiver sido reinserido em S com violação do prazo, faça $S = S_0$ e fixe p_f em sua posição na rota. Faça $P = P - p_f$, e retorne ao passo 2.3.
 - 2.7. Se o evento é a chegada de um ou mais pedidos, inclua cada novo pedido p inserido em S no conjunto PP ($PP = PP + \{p\}$). Se p foi planejado dentro do prazo do atendimento, inclua-o no conjunto PP ($PP^* = PP^* + \{p\}$).

Quadro 1. Módulo de despacho diário para o PRV dinâmico (PUREZA; LAZARIN, 2010).

Os conjuntos do Quadro são descritos a seguir. O conjunto P armazena pedidos conhecidos que não fazem parte da solução corrente, PP registra os pedidos planejados ao longo da execução do módulo, e PP^* armazena o subconjunto de pedidos em PP planejados dentro do prazo. A heurística de inserção no passo 2.5 do Quadro 1 é baseada numa das heurísticas de Solomon (1987) para o problema de roteirização de veículos com janelas de tempo (PRVJT) que é apresentada no quadro 2 (PUREZA; LAZARIN, 2010).

1. Inicialização: Faça $NI = P$ e $l = \emptyset$.
2. Se $r = 0$ ou não houver posições factíveis para inserção dos pedidos NI na solução corrente S , inicie uma rota utilizando o critério de inicialização para seleção do pedido semente (p_s). Faça $r = r + 1$, $l = l \cup p_s$ e $NI = NI - p_s$. Caso haja caminhos mais rápidos entre o depósito e a localidade de p_s através de localidades conhecidas, inclua estes caminhos em r . Caso nenhum pedido possa iniciar rotas, vá para o passo 6.
3. Para cada pedido $p \in NI$, obtenha a melhor posição de inserção factível de p nas rotas correntes segundo o critério de inserção. Caso haja caminhos mais rápidos entre a localidade de p e demais localidades conhecidas, adote estes caminhos e atualize a programação das rotas quanto pertinente.
4. Se no passo anterior tiver sido obtida pelo menos uma posição factível de inserção para pelo menos um pedido, selecione o pedido que melhor atende o critério de inserção (pedido p^*) e o insira na rota e posição associada. Seja $l = l \cup p^*$ e $NI = NI - p^*$. Caso contrário, se nenhum $p \in NI$ apresentar posições factíveis de inserção e se $r < F$ (tamanho da frota), vá para o passo 2.
5. Se $NI \neq \emptyset$, vá para o passo 3.
6. Retorne a solução S e $P = NI$ ao programa principal.

Quadro 2. Passos da heurística de inserção para o PRV dinâmico.

O conjunto **NI**, do Quadro 2, contém os pedidos que não fazem parte da solução corrente **S** (conjunto **P** do módulo de despacho no Quadro 1), e o conjunto **I** (do Quadro 2) armazena o subconjunto de pedidos em **P** planejados na execução da heurística. O critério de inicialização utilizado no passo 2 seleciona o pedido com prazo de atendimento mais curto em **NI** e o critério de inserção utilizado nos passos 3 e 4 é dado pela Equação 1.

$$Max g_{kij} = d_{dk} - (0,5(d_{ik} + d_{kj} - d_{ij}) + 0,5(s_{1e} - s_{1o})) \quad (1)$$

Onde:

- d_{dk} , d_{ik} , d_{kj} e d_{ij} : distância entre as localidades o depósito e k, i e k, i e j e i e j , respectivamente.
- s_{1e} : nível médio de serviço com a inclusão de k entre i e j .
- s_{1o} : nível médio de serviço na solução atual.

As localidades d_{dk} , d_{ik} , d_{kj} e d_{ij} são as distâncias entre os pontos. Essas distâncias são ligações possíveis, e medir essa extensão efetiva que o veículo percorre pode ser feito considerando os caminhos mais usados ou mais prováveis entre os pontos. Nesse sentido a extensão de cada segmento (arco) da rede. Contudo pode-se utilizar a distância euclidiana. Segundo Novaes (1989) na maioria das aplicações de transporte a distância euclidiana possa ser utilizada desde que ela represente uma distância efetiva. Essa distância representa o percurso viário mais curto, ou mais utilizado, dependendo do caso real de aplicação (urbano ou regional) e esta distância efetiva é diferente da euclidiana que representa a distância mais curta (linha reta) entre dois pontos. Para calibrar essa distância multiplicamos a distância euclidiana por coeficientes, que devem representar a sinuosidade das ligações, as condições de pavimento, condições de tráfego etc.

3. Descrição do Cenário

O problema de roteirização de veículos aqui tratado é baseado em uma situação prática de uma empresa de pequeno porte do setor de distribuição de energia elétrica, localizada no interior do Estado do Rio Grande do Sul. Além dos serviços de comercialização e distribuição, a empresa estudada presta serviços de manutenção corretiva e preventiva da rede. Existem três equipes, cada uma com um tipo de veículo e serviço: a equipe comercial, com motos e carros; a equipe de obras, com caminhões e a equipe de atendimento com pick-ups. Em particular, a equipe de atendimento, foco deste trabalho, é descrita como se segue.

Ao longo do dia, clientes ligam para o centro de operação e distribuição (COD), tendo seu pedido registrado por atendentes, em protocolos de solicitação de serviço. Esses protocolos descrevem as especificações do pedido, como a localização, tipo de serviço e a

descrição da solicitação. A ANEEL estabelece prazos variados para cada tipo de serviço a partir da colocação do pedido, estes prazos são detalhados na seção 3.1.

A equipe é composta por oito funcionários e tem a sua disposição quatro veículos, sendo uma dupla de funcionários designada para cada veículo. Os veículos se encontram em duas garagens, uma localizada na própria empresa e outra em um município próximo. Duas duplas são responsáveis por atender a área de cada município, porém a distribuição das ordens considera também o balanceamento da carga de trabalho. Os clientes estão distribuídos na região de atendimento de forma mista; partes da região apresentam concentração de muitos clientes, como é o caso da área central de cidades. Em outras partes, observam-se poucos clientes geograficamente dispersos uns dos outros, como é o caso da área rural. Assim, pode-se dizer que a distribuição dos pontos de demanda é do tipo aleatória/agrupada (random/clustered).

A sequência de atendimento é elaborada manualmente pelo atendente do COD de 1 a 3 horas antes do início da execução. O critério utilizado pode ser entendido como uma ponderação entre distância e tempo da localidade anterior da sequência. Assim, o primeiro cliente a ser atendido é o que está mais próximo (em espaço e tempo) da empresa, o segundo cliente é o que se encontra mais próximo do primeiro cliente, e assim sucessivamente. Esse critério é temporariamente desconsiderado em caso de atendimento imediato de clientes com prazos de serviço próximos do limite. Além disso, em casos de interrupção de fornecimento a equipe mais próxima é enviada imediatamente.

O turno de trabalho é variável, com duas possibilidades: semana normal (de segunda a sexta das 7:45 até as 12:00 e das 13:30 até as 18:00); e semana especial (de terça a sexta, das 12:00 até as 17:00 e das 18:00 até as 21:48 e sábado das 8:45 até as 11:45 e das 12:45 até as 18:33). Cada equipe costuma atender de 6 a 8 ordens num dia normal. Caso tenham atendido todas as ordens antes do término do expediente, telefonam para os assistentes a fim de verificar se há algum pedido a ser servido. Caso não haja, os funcionários tratam de atividades como limpeza do veículo ou do pátio da empresa. Após o fim do expediente, interrupções no fornecimento são anotadas no COD e passados a uma das equipes em semana especial para atendimento imediato.

3.1. Prazos de atendimento do setor elétrico

A ANEEL (Agência Nacional de Energia Elétrica) estabeleceu no módulo 8 do PRODIST (anexo na resolução ANEEL nº395/2010) metas para os indicadores individuais de continuidade do fornecimento. O DIC (Duração de Interrupção Individual por Unidade

Consumidora ou por Ponto de Conexão) indica o tempo que uma unidade ficou sem energia, o FIC (Frequência de Interrupção Individual por Unidade Consumidora ou por Ponto de Conexão) indica o número de ocorrências destas interrupções e o DMIC (Duração Máxima de Interrupção Contínua por Unidade Consumidora ou por Ponto de Conexão) indica a duração da maior interrupção de energia em uma unidade. Os indicadores são calculados para períodos de observação mensais, trimestrais e anuais.

Essas metas são revisadas a cada ciclo de revisão tarifária através da análise comparativa de desempenho (*yardstick competition*), que agrega as empresas de características semelhantes em clusters, e o valor médio em determinadas variáveis é utilizado como benchmark para as firmas do *cluster* (PESSANHA; SOUZA; LAURENCED, 2006). Em Barbosa, Carvalho e Lopes (2004) tem-se uma explicação detalhada da metodologia de penalidades por violações dos padrões nos indicadores de continuidade coletivos (DEC e FEC), e em 2003, a soma destas multas totalizaram R\$ 35,3 milhões. Entretanto para problemas de roteirização é mais prático e igualmente eficiente realizar o controle através dos indicadores de continuidade individuais.

Descrição	Prazo de Atendimento
Prazo máximo de vistoria de unidade consumidora, localizada em área urbana.	3 dias úteis
Prazo máximo de vistoria de unidade consumidora, localizada em área rural.	5 dias úteis
Prazo máximo de vistoria de unidade consumidora do grupo B, localizada em área urbana, a partir da data da aprovação das instalações.	2 dias úteis
Prazo máximo de vistoria de unidade consumidora do grupo B, localizada em área urbana, a partir da data da aprovação das instalações.	5 dias úteis
Prazo máximo de vistoria de unidade consumidora do grupo A, a partir da data da aprovação das instalações.	7 dias úteis
Prazo máximo de atendimento de solicitações de aferições dos medidores e demais equipamentos de medição.	30 dias úteis*
Prazo máximo para religação, sem ônus para o consumidor, quando constatada a suspensão indevida do fornecimento.	4 horas
Prazo máximo de atendimento a pedidos de religação para unidade consumidora localizada em área urbana, quando cessado o motivo da suspensão.	24 horas
Prazo máximo de atendimento a pedidos de religação para unidade consumidora localizada em área rural, quando cessado o motivo da suspensão.	48 horas

Prazo máximo de atendimento a pedidos de religação de urgência em área urbana, quando cessado o motivo da suspensão.	4 horas
Prazo máximo de atendimento a pedidos de religação de urgência em área rural, quando cessado o motivo da suspensão.	8 horas

Tabela 1 - Metas de atendimento (Adaptado de resolução ANEEL nº414/2010).

* A distribuidora pode agendar com o consumidor no momento da solicitação ou informar, com antecedência mínima de 3 (três) dias úteis, a data fixada e o horário previsto para a realização da aferição, de modo a possibilitar o seu acompanhamento pelo consumidor.

3.2. Formalização do problema

O problema acima pode ser descrito como um PRV dinâmico e estocástico com um único depósito e tamanho de frota limitado e homogêneo. Apesar de existirem duas garagens os veículos sempre retornam ao seu depósito de origem, então os problemas podem ser resolvidos separadamente. O grau de dinamismo do problema é moderado, visto que mais de 50% das ordens não são conhecidas no planejamento das rotas. Restrições de janela de tempo se aplicam sobre os depósitos, devido à jornada de trabalho, e aos serviços de aferição de medidores, que são marcados antecipadamente com os clientes. A carga requisitada pode ser considerada zero, dado que se trata de uma prestação de serviço. O planejamento das rotas é feito dia-a-dia.

Os pedidos chegam ao sistema em $t_p \geq 0$ em um dia d_p , e é caracterizado pelas coordenadas geográficas da localidade k do cliente e pelo tempo de serviço estimado st_p . Uma janela de tempo $[e_o, l_o]$ está associada ao depósito e corresponde a duração máxima de todas as atividades de cada dia, e para pedidos de aferição de medidores se aplica uma janela $[e_p, l_p]$. Cada motorista é informado de sua próxima visita assim que o serviço na localidade atual é concluído, e em seguida parte. Por simplicidade, os veículos que já saíram de uma localidade para outro atendimento não podem ser redesignados a um destino diferente (política de compromisso). A duração da rota inclui a somatória dos tempos de viagem e de serviço.

Devido às várias restrições impostas, o atendimento de todos os pedidos dentro do prazo não é garantido. O objetivo é determinar um conjunto de rotas de maneira hierárquica minimize: (1) número de pedidos não atendidos; (2) número de pedidos com prazo de atendimento violado; (3) o tempo médio decorrido entre a colocação dos pedidos e início de seus atendimentos (maximização do nível médio de serviço); e (4) a distância total percorrida. O atendimento de interrupções de fornecimento é prioritário, o que é explicável, dado que este tipo de serviço é tratado como emergência, maximizar o nível médio de serviço e minimizar o número de pedidos com prazo de atendimento violado também refletem o interesse da

empresa na geração de uma imagem de confiança e base segura de mercado. A melhoria nas distâncias percorridas reduz os custos variáveis.

4. Procedimentos Metodológicos

Um algoritmo baseado no algoritmo WEF dinâmico foi implementado em PERL, e os experimentos realizados em um computador Intel Core 2 Duo T5800 @2,00 GHz e 4,00 GB de RAM. A implementação foi realizada em quatro etapas:

1. Inicialmente a programação foi feita em PERL, e foram realizadas as adaptações necessárias para que o algoritmo se adeque ao problema e a linguagem utilizada (anexo 1).
2. Foram coletados os dados necessários para a utilização do algoritmo, como tempos de atendimento, velocidades médias e distâncias.
3. Os resultados obtidos no programa e os obtidos na prática, em um dia de trabalho, foram comparados em termos de (i) número de pedidos atendidos;(ii) número de pedidos não-atendidos; (iii) nível médio de serviço por pedido atendido; (iv) distância total percorrida.

O método de pesquisa empregado é uma pesquisa-ação de natureza aplicada com procedimento quantitativo. A pesquisa é de natureza aplicada, pois é voltada a encontrar soluções práticas para um problema real, diferentemente das pesquisas de natureza básica, que são motivadas por curiosidade intelectual. O procedimento é quantitativo, uma vez que serão tomadas medidas numéricas.

4.1. O algoritmo WEF modificado

A discussão a seguir é uma descrição do procedimento utilizado e adaptações para elaboração das rotas, com ênfase nas diferenças para o algoritmo apresentado na seção 2.1. A Figura 2 apresenta o módulo de despacho implementado, modificado a partir do apresentado no Quadro 1. No passo 1, a variável que recebe os instantes de ocorrência de eventos (t) é inicializada, juntamente com o conjunto P, que armazena pedidos conhecidos que não fazem parte da solução corrente. Os conjuntos solução S1, S2, S1prox e S2prox são todos inicializados com o elemento 0 (o depósito) incluso. Isso ocorre devido ao fato de que todas as rotas começam e terminam no depósito. S1 e S2 recebem a parte da solução já realizada e S1prox e S2prox recebem a parte da solução a ser realizada.

O passo 2 consiste das operações de processamento de eventos, e inserção de pedidos durante a jornada de trabalho ($t \leq l_0$). Se existirem novos pedidos no sistema estes são

inclusos em P (passo 2.2) e aciona o procedimento de inserção de pedidos (passo 2.4), que insere todos os pedidos em $S1prox$ ou $S2prox$, dependendo de qual carro ira atender ao pedido. O passo 2.4 é detalhado na seção 4.2. Os passos 2.1 e 2.3 garantem o avanço do tempo caso não ocorra nenhum evento, sendo o passo 2.3 utilizado no início da jornada caso não existam pedidos no sistema ($P = \emptyset$), e o passo 2.1 avança o tempo ao longo do dia. Após cada planejamento de rotas os pedidos atendidos são removidos das soluções futuras, $S1prox$ e $S2prox$, e incluídos nas soluções realizadas, $S1$ e $S2$ (passo 2.5). Antes do próximo evento o conjunto P é esvaziado, dado que cada pedido $p \in P$ foi incluído em alguma solução.

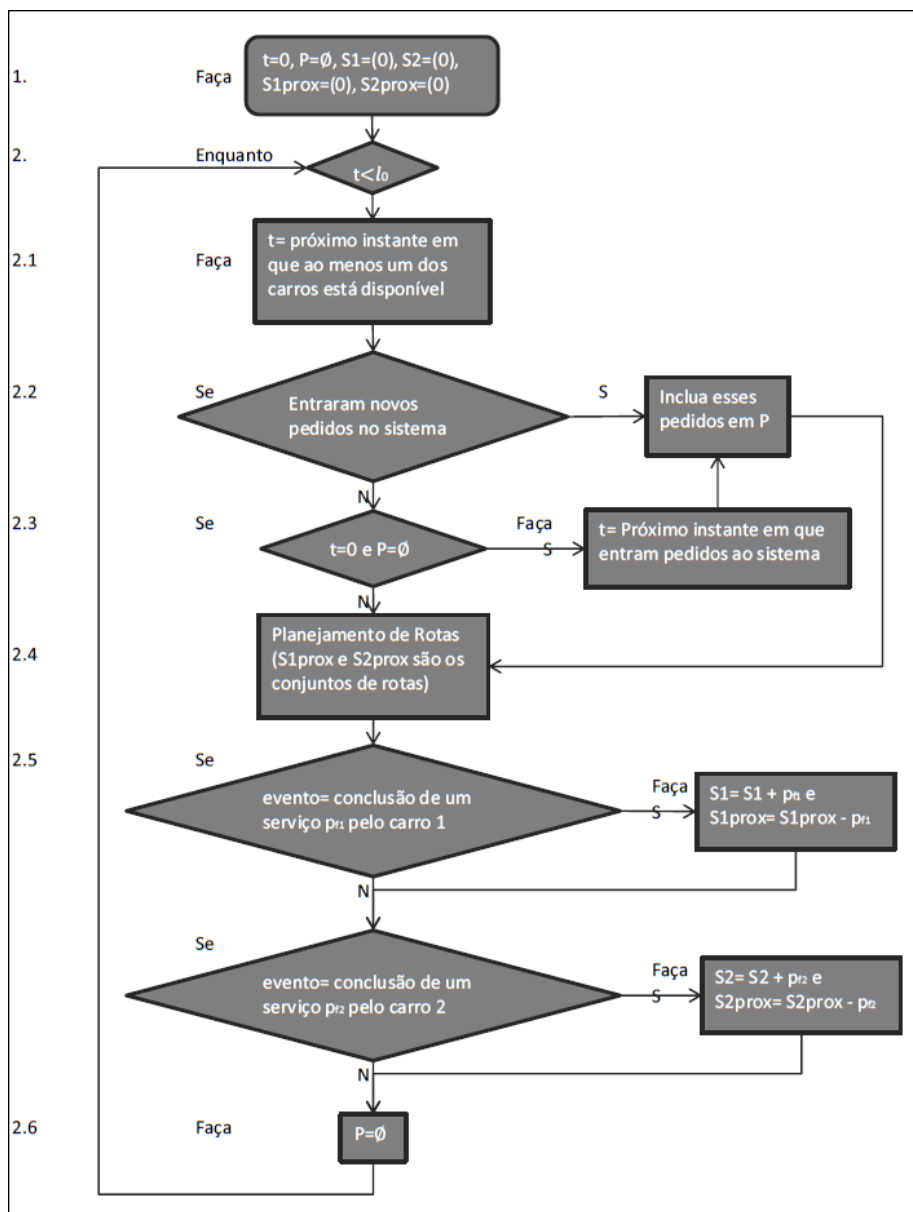


Figura 2. Passos do módulo de despacho implementado.

As principais modificações na heurística utilizada estão no fato de que as rotas não são desconstruídas e reconstruídas como na heurística apresentada na seção 2.1, fazendo apenas a

inserção dos pedidos dinâmicos nas rotas existentes. Com isso é possível manter o compromisso com os clientes sem comprometer o balanceamento de serviço entre as rotas, isto é, a distribuição dos pedidos entre os carros. No algoritmo WEF original era possível que todas as inserções ocorressem na mesma rota. A Figura 3 apresenta a heurística de inserção adaptada.

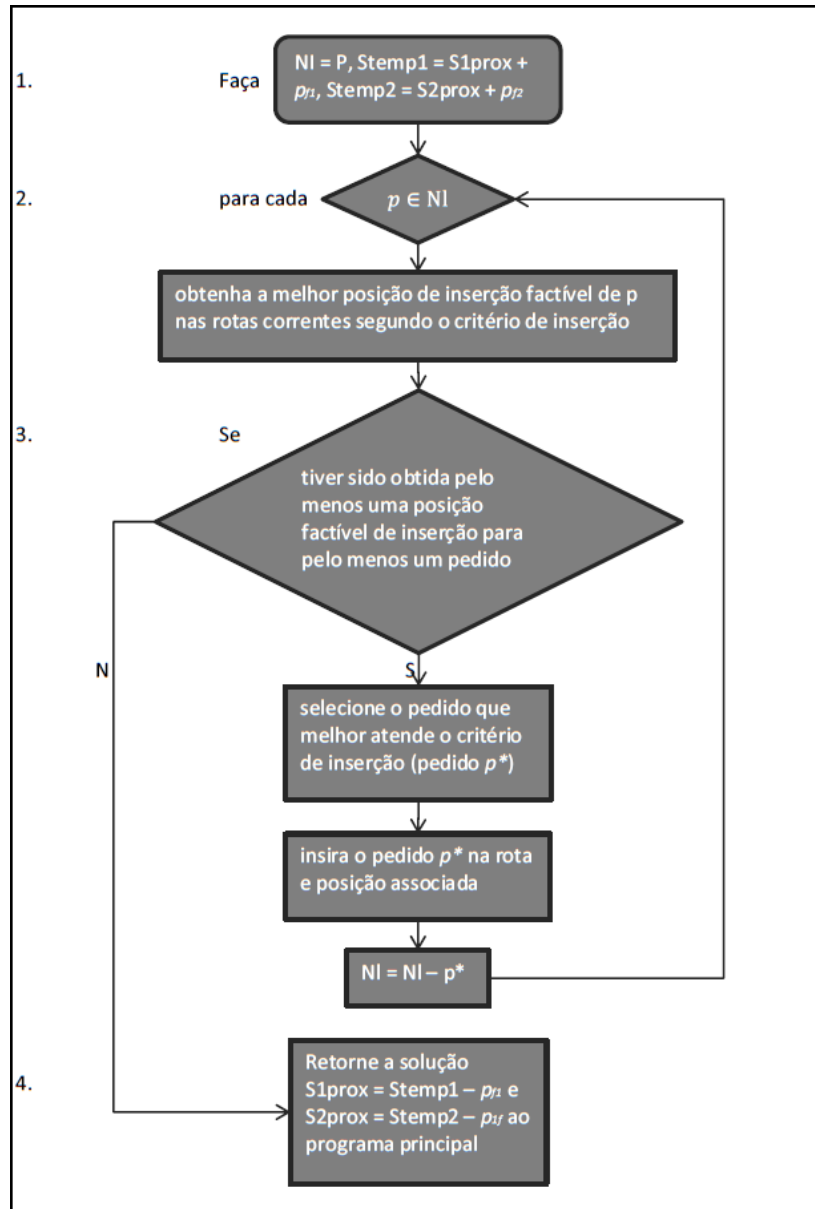


Figura 3. Passos da heurística de inserção para o PRV dinâmico.

No passo 1, os conjuntos **NI**, **Stemp1** e **Stemp2** são inicializados. O conjunto **NI** contém os pedidos que não fazem parte da solução corrente **S** (o conjunto **P** do módulo de despacho na Figura 2), e **Stemp1** e **Stemp2** armazenam os pedidos a serem atendidos já programados (**S1prox** e **S2prox**) e o pedido que contém a localização onde cada um dos carros será liberado (p_{r1} e p_{r2}). Nos passos 2 e 3 é feita a inserção de cada pedido em uma das

rotas. Caso nenhum pedido possa ser inserido nas rotas o procedimento é finalizado (passo 4). O critério de inserção utilizado nos passos 2 e 3 busca maximizar ganhos com a inclusão da localidade k do pedido p não roteado entre duas localidades i e j em rotas parciais. Esse critério é semelhante ao utilizado por Pureza e Lazarin (2010) no caso em que o número de rotas é igual ou superior ao tamanho da frota menos dois ($F-2$), o que no caso com dois veículos é sempre verdadeiro. Especificamente, as localidades i e j são selecionadas tal que (Equação 2):

$$Max g_{kij} = 0,5(d_{ik} + d_{kj} - d_{ij}) + 0,5(s_{lc} - s_{lo}) \quad (2)$$

Onde:

- d_{ik} , d_{kj} e d_{ij} : distância entre as localidades i e k , i e j e i e j , respectivamente.
- s_{lc} : nível médio de serviço com a inclusão de k entre i e j .
- s_{lo} : nível médio de serviço na solução atual.

5. Coleta de dados e experimentos computacionais

Os dados utilizados nos experimentos foram obtidos por meio de documentos gerados após a conclusão de cada serviço (comprovantes de conclusão de serviço). Esses dados representam um dia típico de atendimento aos clientes da empresa. As informações de localização dos pedidos, encontrada na ficha no formato de endereço foram convertidas em coordenadas cartesianas (x e y). Os tempos de deslocamento foram calculados a partir das informações de hora de partida e hora de chegada ao local de atendimento, enquanto o tempo de serviço é dado pela diferença entre a hora da conclusão do atendimento e a hora de chegada ao local. Por critério de simplicidade a velocidade foi considerada constante (igual a 60 km/h) e foram utilizadas distâncias euclidianas. Para manter os tempos de deslocamento fidedignos, as distâncias foram multiplicadas por coeficientes (vide seção 2.1), o que modificou as coordenadas de alguns pedidos. Tal fato é condizente com a realidade visto que os pontos de deslocamento mais lento refletem estradas em más condições. A tabela 2 apresenta as instâncias que foram utilizadas.

n	x	Y	ready time	due date	t
0	30	35	0	615	0
1	44	30	66	156	16
2	30	25	94	184	10
3	31	25	88	178	10
4	21	42	210	975	45
5	21	11	447	537	60
6	50	33	705	795	18
7	44	49	348	975	20
8	39	48	423	975	20
9	39	28	255	975	27
10	41	24	0	975	15
11	31	33	0	975	15
12	15	89	71	975	57
13	4	15	85	975	25
14	29	23	232	975	14
15	50	45	238	975	21
16	45	42	516	975	13

Tabela 2 – Instâncias.

A instância 0 corresponde ao depósito, e inicializa todas as soluções os valores de *ready time* e *due date* nesta linha correspondem ao instante de inicialização do sistema e final da jornada de trabalho para este dia. Nas demais linhas *ready time* é o instante da chegada do pedido (em minutos) e *due date* é o prazo máximo de atendimento para aquele pedido. Para os pedidos que tem prazo de entrega maior do que um dia foi adotado um valor de *due date* de 975 (equivalente a meia-noite, se o depósito inicializa as 7:45). A última coluna apresenta os dados de tempo de serviço (*t*), em minutos.

Para os experimentos computacionais foram utilizadas as informações de cada pedido atendido em um dia real (instâncias 1 a 9 da Tabela 2) da empresa para comparação do nível de serviço praticado com o simulado no algoritmo. Em seguida foi realizado um segundo experimento utilizando de todos os pedidos que entraram no sistema neste mesmo dia, inclusive os que só foram atendidos posteriormente (instâncias 1 a 16 da Tabela 2). Essa segunda etapa foi realizada para avaliar o potencial do algoritmo de aumentar a capacidade de atendimento. Os tempos computacionais praticados foram inferiores a 5 segundos. A Tabela 2 apresenta os resultados obtidos na situação real e nas duas simulações realizadas (Dinâmico-1 e Dinâmico-2). As duas primeiras colunas identificam a situação. Também são apresentados os dados de nível médio de serviço e distância total percorrida. Na última coluna são apresentadas as rotas utilizadas, sendo 0 (o depósito) o início e final de todas as rotas. Para atender ao pedido 6 foi realizada uma simulação com limitante inferior do depósito as 18:00

($e_0 = 615$) e limitante superior as 21:48 ($l_0 = 843$), referente ao atendimento da equipe de plantão.

Situação	Pedidos Atendidos	Nível médio de serviço (minutos)	Distância (Km)	Rota Carro 1	Rota Carro 2
Real	9	130,3	196,0	0-1-2-3-4-5-0	0-7-8-9-0-6-0
Dinâmico-1	9	84,7	181,1	0-3-2-7-8-0	0-4-9-5-0-6-0
Dinâmico-2	16	87,1	518,8	0-10-1-13-2-4-15-9-7-16-0	0-11-12-3-14-8-5-0-6-0

Tabela 2 – Resultados acumulados.

As vantagens da adoção do algoritmo dinâmico são evidentes com ganhos expressivos em todos os critérios avaliados. Na primeira simulação (Dinâmico-1) obteve-se uma melhora de 35,0% em relação a situação real, sendo gasto apenas 65,0% do tempo de atendimento praticado, além disso foi possível reduzir a distância percorrida em 7,6%. Na segunda simulação (Dinâmico-2) foi possível atender todos os pedidos no sistema, uma melhora de 77,8% em pedidos atendidos, mantendo um nível de atendimento muito próximo ao obtido na primeira simulação (2,8% inferior) e melhora de 33,1% em relação a situação real. O aumento da distância percorrida era esperado, em virtude do maior número de atendimentos e distribuição espacial dos pedidos, mas o custo relacionado é facilmente compensado pela economia em multas e ganho no índice de satisfação dos consumidores.

6. Conclusões e considerações

Neste trabalho foi proposta e implementada uma metodologia de roteirização de veículos que atenda o setor elétrico. O algoritmo implementado foi adaptado do algoritmo WEF dinâmico de Lazarin (2008). Os experimentos realizados para um dia típico de serviço da empresa indicam que o tratamento dinâmico traz diversas vantagens. Foi possível gastar quase dois terços do tempo de atendimento médio por cliente, em relação às rotas feitas de forma manual, e foi possível atender a quase 80% a mais de pedidos. Devido às características dinâmicas do setor, as vantagens aos consumidores e concessionárias seriam facilmente refletidas em qualidade de vida e resultado econômico se a aplicação do modelo dinâmico for implantada. O algoritmo, pela sua rapidez de execução, deve servir de apoio ao atendente que elabora as rotas.

Para pesquisas futuras, sugere-se uma nova adaptação do algoritmo para um cenário estocástico. O tratamento estocástico também possibilitaria incorporar quebra de veículos, evento aleatório não considerado no modelo atual. Outra modificação que poderia ser

agregada ao algoritmo é relacionada a veículos com turnos diferenciados, o que não foi implementado neste experimento.

Referências Bibliográficas

- BALLOU, R.H. *Gerenciamento da cadeia de suprimentos: planejamento, organização e logística empresarial*. 4. ed. Porto Alegre: Bookman, 2001.
- BARBOSA, A. S.; CARVALHO, P. L.; LOPES, P. H. S. Procedimento para aplicação de penalidade por violação dos padrões dos indicadores de continuidade DEC e FEC. XVI. *Seminário Brasileiro sobre Qualidade da Energia Elétrica*, Belém, 2005.
- BRASIL. Agência nacional de energia elétrica (ANEEL), Resolução nº024, de 27 de janeiro de 2000. Disponível em: <<http://www.aneel.gov.br/cedoc/res2000024.pdf>>. Acesso em 25 nov. 2010.
- BRASIL. Agência nacional de energia elétrica (ANEEL), Resolução nº395, de 15 de dezembro de 2009. Disponível em: <<http://www.aneel.gov.br/cedoc/ren2009395.pdf>>. Acesso em 25 nov. 2010.
- BRASIL. Agência nacional de energia elétrica (ANEEL), Resolução nº414, de 9 de setembro de 2010. Disponível em: <<http://www.aneel.gov.br/cedoc/ren2010414.pdf>>. Acesso em 25 nov. 2010.
- GENDREAU, M.; LAPORTE, G.; SEMET, F. A dynamic model and parallel tabu search heuristic for real-time ambulance relocation. *Parallel Computing*, v. 27, n.12, p. 1641-1653, 2001.
- GENDREAU, M.; LAPORTE, G.; SEMET, F. Solving an ambulance location model by tabu search. *Location Science*, v. 5, n.2, p. 75-88, 1997.
- GHIANI, G. et al. Real-time vehicle routing: solution concepts, algorithms and parallel computing strategies. *European Journal of Operational Research*, v. 151, n. 1, p.1-11, 2003.
- IANNONI, A. P.; MORABITO, R. Otimização da localização das bases de ambulâncias e do dimensionamento das suas regiões de cobertura em rodovias. *Produção*, v. 18, n. 1, p. 47-63, 2008.
- IANNONI, A. P.; MORABITO, R. Modelo Hipercubo integrado a um algoritmo genético para análise de sistemas médicos emergenciais em rodovias. *Gestão & Produção*, v. 13, n. 1, p.93-104, 2006.
- FLEISCHMANN, B.; GNUTZMANN, S.; SANDVOß, E. Dynamic vehicle routing based on online traffic information. *Transportation Science*, v. 38, n. 4, p. 420-433, 2004.
- KAGAN, N.; OLIVEIRA, C.C.B.; ROBBA, E.J. *Introdução aos sistemas de distribuição de energia elétrica*. São Paulo: Editora Edgard Blücher, 2005.
- LARSEN, A. *The dynamic vehicle routing*. 2000. 208 f. Thesis (Phd) - Department of Mathematical Modeling - IMM, Technical University of Denmark - DTU, Denmark, 2000.
- LARSEN, A.; MADSEN, O. B. G.; SOLOMON, M.M. Recent Developments in Dynamic Vehicle Routing Systems, *OPERATIONS Research/Computer Science Interfaces Series The Vehicle Routing Problem: Latest Advances and New Challenges Part I*, 2008. Disponível em: <<http://www.springerlink.com/content/x841419553436m66/fulltext.pdf>>. Acesso em 25 nov. 2010.

- LAZARIN, D. F. *Roteamento dinâmico de veículos: análise do impacto em atividades de prestação de serviço*. 2008. 67 f. Dissertação (Mestrado) - UFSCAR, São Carlos, 2008.
- NOVAES, A.G. *Sistemas logísticos: Transporte, Armazenagem e Distribuição Física de Produtos*. São Paulo: Editora Edgard Blücher, 1989.
- MENDOÇA, F. C.; MORABITO, R. Aplicação do modelo hipercubo para análise de um sistema médico-emergencial em rodovia. *Gestão & Produção*, v. 7, n. 1, p. 73-91, 2000.
- PESSANHA, J. F. M.; SOUZA, R. C.; LAURENCED, L. C. Um modelo de análise envoltória de dados para o estabelecimento de metas de continuidade do fornecimento de energia elétrica. *Pesquisa Operacional*, v. 27, n. 1, p. 51-83, 2007.
- PSRAFTIS, H. N. Dynamic vehicle routing: status and prospects. *Annals of Operations Research*, v. 61, n. 1, p. 143-164, 1995.
- PUREZA, V.; LAPORTE, G. Waiting and buffering strategies for the dynamic pickup and delivery problem with time windows. *Information Systems and Operational Research*, v. 46, n. 3, p. 165-175, 2008.
- PUREZA, V.; LAZARIN, D. F. Um estudo de impactos do roteamento dinâmico de veículos em atividades de prestação de serviço. *Produção*, 2010. Disponível em: http://www.scielo.br/scielo.php?pid=S0103-65132010005000042&script=sci_arttext. Acesso em 25 nov. 2010.
- SAVELSBERGH, M.W.P.; SOL, M. The general pickup and delivery problem. *Transportation Research*, v.29, p.17-29, 1995.
- SOLOMON, M. M. Algorithms for the vehicle routing and scheduling problem with time window constraints. *Operations Research*, v. 35, n. 2, p. 254-265, 1987.
- WAGNER, H.M. *Pesquisa Operacional*. 2.ed. Rio de Janeiro: Prentice-Hall do Brasil, 1986.

DYNAMIC VEHICLE ROUTING ON MAINTENANCE SERVICE ACTIVITIES FOR THE ELECTRIC POWER INDUSTRY

Abstract

This paper is a methodology proposition for vehicle routing that is adequate for the maintenance service of an electric power company. The algorithm is powered by an insertion heuristic that modifies routes in real time. The results indicate gains in terms of service level.

Keywords:

Dynamic Vehicle Routing; Heuristics; electric power industry.

ANEXO 1

```
#####
```

```
#SUBROTINAS
```

```
sub proximo_evento #recebe o tempo atual, percorre o array de eventos e devolve o mínimo tempo que é maior do que o atual
```

```
{
    ($t_atual) = @_ ;
#    print $t_atual . " ";
    $t_min = 999999;
    foreach $tempo(@readytime) {
        if (($tempo > $t_atual) && ($tempo < $t_min)) {
            $t_min = $tempo;
        }
    }
#    print $t_min;
    return $t_min;
}
```

```
sub define_r #a partir dos intervalos de tempo, define r
```

```
{
    ($t_atual) = @_ ;
    if ($t_atual < 465) {
        return 0;
    }
    else {
        if ($t_atual < 720) {
            return 0;
        }
    }
}
```

```
sub distancia #calcula a distancia entre dois pontos , recebe 4 coordenadas
```

```
{
    ($ax,$ay,$bx,$by) = @_ ;
    $dist = ((($ax - $bx)**2 + ($ay - $by)**2)**(1/2));
    return $dist;
}
```

```

sub distancia_ped #calcula a distancia entre dois pedidos
{
    my $ped1; my $ped2;
    ($ped1,$ped2) = @_;
    return &distancia($coordx[$ped1],$coordy[$ped1],$coordx[$ped2],$coordy[$ped2])
}

```

```

sub criterio_inicializacao
{
    my $t_min = 999999999;
    my $i = 99;
    foreach $indice(@NL) {
        if ($duedate[$indice] < $t_min) {
            $t_min = $duedate[$indice];
            $indice_inic = $indice;
#            print $t_min; print " "; print $indice; print " ";
        }
    }
    return $indice_inic;
}

```

```

sub deletar_p_do_NL
{
    ($valor) = @_;
    my $size = scalar @NL;
    $nao_achou = 1;
    my $i = 0;
    while (($i < $size)) {
        if ($NL[$i] == $valor) {
            splice(@NL,$i,1);
            $nao_achou = 0;
        }
        $i++;
    }
}

```

```

sub deletar_p_de_P

```

```

{
    ($valor) = @_;
    my $size = scalar @P;
    my $i = 0;
    while (($i < $size)) {
        if ($P[$i] == $valor) {
            splice(@P,$i,1);
        }
        $i++;
    }
}

```

sub service_level

```

{
    ($t_temp,@Stemp) = @_; #recebe o tempo atual para a variavel que fará a contagem;
    my $soma_nivel = 0;
    my $i;
    my $nsol = scalar @Stemp;
    for ($i=0;$i<$nsol-2;$i++) {

        my $pedidi = $Stemp[$i];
        my $pedidj = $Stemp[$i+1];
        my $duracao_atendimento = &distancia_ped($pedidi,$pedidj)/$velocidade +
        $servicetime[$pedidj];

        $t_temp = $t_temp + $duracao_atendimento;
        # print $duedate[$pedidj];print "\n";
        if ($t_temp > $duedate[$pedidj]) { #verifica se há atraso no atendimento e,
        se sim, atribui uma punicao
            $soma_nivel = $soma_nivel + 2*($t_temp - $readytime[$pedidj]);
        }
        else {
            $soma_nivel = $soma_nivel + 1*($t_temp - $readytime[$pedidj])
        }
        # print "a";
    }
    $t_temp = $t_temp + &distancia_ped($Stemp[$nsol-2],$Stemp[$nsol-1]) / $velocidade;
    # print "total: $t_temp\n";
}

```

```

#           print "soma:$soma_nivel.";
           if ( $t_temp > $tempo_maximo ) {
               $nivel_medio = "estouro"; }
           else {   if ($nsol < 3) { $nivel_medio = 0; }
               else {   $nivel_medio = $soma_nivel / ($nsol - 2); } }
           return $nivel_medio;
}

```

sub Valor

```

{
    ($tvlr,$pedi,$pedk,$pedj,$s10,@Stempvlr) = @_;
    $slc = &service_level($tvlr,@Stempvlr);
    $dik = &distancia_ped($pedi,$pedk);
    $dij = &distancia_ped($pedi,$pedj);
    $dkj = &distancia_ped($pedk,$pedj);
    if ($slc eq "estouro")
        { $g = 99999; }
    else { $g = 0.5 * ($dik + $dkj - $dij) + 0.5 * ($slc - $s10);}
#   print "slc: " . $slc . " s10: " . $s10 . "\n";
#   print "dik: $dik , dij: $dij , dkj: $dkj\n";
#   print "valor: $g , slc: $slc , s10: $s10\n";
    return $g;
}

```

sub ultimo_elemento

```

{
    @array = @_;
    $elemento = $array[scalar(@array)-1];
    return $elemento;
}

```

sub alocar_carros

```

{
    if (($C1 == -1) && (scalar(@S1prox) > 1)) {
        $prox_pedido = shift(@S1prox);
        $ped_anterior = &ultimo_elemento(@S1);
    }
}

```

```

        $C1 = $t + &distancia_ped($ped_anterior,$prox_pedido)/$velocidade +
$servicetime[$prox_pedido];
        $endtime[$prox_pedido] = $C1;
        $carro[$prox_pedido] = 1;
        push(@S1,$prox_pedido);
        &deletar_p_de_P($prox_pedido);
    }

    if (($C2 == -1) && (scalar(@S2prox) > 1) ) {
        $prox_pedido = shift(@S2prox);
        $ped_anterior = &ultimo_elemento(@S2);
        $C2 = $t + &distancia_ped($ped_anterior,$prox_pedido)/$velocidade +
$servicetime[$prox_pedido];
        $endtime[$prox_pedido] = $C2;
        $carro[$prox_pedido] = 2;
        push(@S2,$prox_pedido);
        &deletar_p_de_P($prox_pedido);
    }
}

```

sub heuristica

```

{
    @Stemp1 = @S1prox;
    unshift(@Stemp1,($S1[scalar(@S1)-1])); #pega o ultimo elemento da soluçao já completa.

    @Stemp2 = @S2prox;
    unshift(@Stemp2,($S2[scalar(@S2)-1])); #pega o ultimo elemento da soluçao já completa.

    @NL = @P;

#    print "\nP: "; print @P; print "\nNL: "; print @NL;

    if ($t < $C1) { #ajusta o tempo de início de acordo com o carro estar ocupado ou não.
        $tempoc1 = $C1;
    }
}

```

```

} else { $tempoc1 = $t; }
if ($t < $C2) {
    $tempoc2 = $C2;
} else { $tempoc2 = $t; }

while ((scalar @NL) > 0) {          #####ENQUANTO AINDA HOVER PEDIDOS PARA INSERIR
#
    $nivel_serv = &nivel_servico;

    $nsol1 = scalar @Stemp1; #n guarda quantos pedidos possui na soluçã atual
    $nsol2 = scalar @Stemp2; #n guarda quantos pedidos possui na soluçã atual
#
    if (($nsol1 < 3) || ($nsol2 < 3)) {
#
        $ps = &criterio_inicializacao;
#
        deletar_p_do_NL($ps);
#
    }

    $Min = 10000;
    $pos_k;
    $k_maximo;
    $carro;
#
    print "aaaa\n";

    $sl0c1 = &service_level($tempoc1,@Stemp1);
    $sl0c2 = &service_level($tempoc2,@Stemp2);
#
    print "[sl0 = " . $sl0 . "]\n";
    foreach $pedidok (@NL) { ##### PARA CADA PEDIDO PARA INSERIR

        for ($i=0;$i<$nsol1-1;$i++) {          ##### TESTA INSERIR NA
ROTA DO CARRO 1

            @StempSLC = @Stemp1;
            splice(@StempSLC,$i+1,0,$pedidok);
            $pedidoi = $StempSLC[$i];
            $pedidoj = $StempSLC[$i+2];
#
            print @StempSLC; print "Carro 1 i: "; print $i; print "\n";
            $Gkij = &Valor($tempoc1,$pedidoi,$pedidok,$pedidoj,@StempSLC);
            if ($Gkij < $Min) {

```

```

        $Min = $Gkij;
        $pos_k = $i + 1; #guarda a posicao do k
        $k_maximo = $pedidok;
        $carro = 1;
    }
}

for ($i=0;$i<$nsol2-1;$i++) { ##### TESTA INSERIR NA ROTA DO
CARRO 2

        @StempSLC = @Stemp2;
        splice(@StempSLC,$i+1,0,$pedidok);
        $pedidoi = $Stemp2[$i];
        $pedidoj = $Stemp2[$i+2];
#        print @StempSLC; print "Carro2 i: "; print $i; print "\n";
        $Gkij = &Valor($Stempoc2,$pedidoi,$pedidok,$pedidoj,@StempSLC);
        if ($Gkij < $Min) {
            $Min = $Gkij;
            $pos_k = $i + 1; #guarda a posicao do k
            $k_maximo = $pedidok;
            $carro = 2;
        }
    }
}
### VAI INSERIR O QUE MELHOR SE ENCAIXAR
#        print "-----Carro: $carro Max: " . $Max . " pos_k: " . $pos_k . " " . $k_maximo . "\n";
        if ($carro == 1) {
            splice(@Stemp1,$pos_k,0,$k_maximo);
        } else { splice(@Stemp2,$pos_k,0,$k_maximo); }

#        print "["; print @Stemp; print "]";
        &deletar_p_do_NL($k_maximo);
#        print "[SLS: " ; print @Stemp1; print " "; print @Stemp2; print "]\n";
    }
#        print "[" . @Stemp . "];
        shift(@Stemp1);
        @S1prox = @Stemp1 ;
        shift(@Stemp2);
        @S2prox = @Stemp2 ;

```



```
#          print "[[[[SLS: " ; print @Stemp1; print " "; print @Stemp2; print "]]]]\n";
}
```

```
sub inicializa_P
{
    my $size = scalar @coordx;
    for ($i=0;$i<$size;$i++)
    {
        push(@P,$i);
    }
}
```

```
sub atualiza_P
{
    ($t_atual, $t_anterior) = @_ ;
#    print $t_atual . " " . $t_anterior . "\n";
    my $size = scalar @coordx;
    for ($i=1;$i<$size;$i++) {
        if (($readytime[$i] <= $t_atual) && ($readytime[$i] > $t_anterior)) {
            push (@P,$i);
#            print "$i\n";
        }
    }
}
```

```
#####
```

```
#IMPORTAR INSTANCIAS DE ARQUIVO TEXTO
```

```
open(DAT, "instancias.txt");
```

```
open (LOG, '>solucao.txt');
```

```
while ($record = <DAT>) {
    @linha = split ('\t',$record);
    push(@coordx,$linha[0]);
    push(@coordy,$linha[1]);
}
```

```

push(@readytime,$linha[2]);
push(@duedate,$linha[3]);
push(@servicetime,$linha[4]);
push(@endtime,"-1");
push(@carro,"0");
}
#####
##### MAIN #####
#####

# print "#####\n";

$velocidade = 1;
$t = $readytime[0];
@P = ();
@S1 = (0);
@S2 = (0);
@S1prox = (0);
@S2prox = (0);
$tempo_maximo = $duedate[0];
$C1 = -1; #C1 e C2 guardam o tempo que o carro ficará livre. Caso seja -1, ele encontra sem nenhum pedido
alocado
$C2 = -1;
$t_ant = -1;

while ($t < $tempo_maximo) {
    if ($t >= $C1) {          #se o tempo é maior ou igual ao tempo que o pedido é atendido, torna o
carro disponível.
        $C1 = -1;
    }
    if ($t >= $C2) {
        $C2 = -1;
    }
    if (($C1 == -1) || ($C2 == -1)) { #se um dos carros está disponível

        &atualiza_P($t,$t_ant);
        $t_ant = $t;
#        foreach (@P) { print "$_\n"; }
        if (scalar (@P) == 0) { #se P for vazio, incrementa o tempo em uma unidade. esse
laço será executado até que entre algum pedido ou acabe a janela de tempo

```

```

                $t = $t + 1;
            }
            else {
#                print "entrou";
                &heuristica;
                &alocar_carros;
                @P = ();
            }
        }
    else { #caso nenhum dos carros esteja disponível, avança o tempo até o evento onde um deles se
libera
        if ($C1 < $C2) { $t = $C1; }
        else { $t = $C2; }
#        print "entrou";
    }
}

```

ESCREVE O LOG DA EXECUCAO

```

print LOG "Pedido\tCarro\tConclusao\n";
for($i = 1; $i < (scalar(@coordx));$i++) {
    print LOG "$i\t$carro[$i]\t$sendtime[$i]\n";
}

# print LOG "\n\nS1prox: ";
# print LOG @S1prox;
# print LOG "\n\nS2prox: ";
# print LOG @S2prox;

print LOG "\n\nS1: \n";
$distancia1 = 0;
for($i = 1; $i < (scalar(@S1));$i++) {
    $distancia1 = $distancia1 + &distancia_ped($S1[$i-1],$S1[$i]);
    print LOG "$S1[$i]\n";
}

print LOG "Distancia total: $distancia1\n";

```

```
# print LOG @S1;
print LOG "\n\nS2: \n";
$distancia = 0;
for($i = 0; $i < (scalar(@S2));$i++) {
    $distancia2 = $distancia2 + &distancia_ped($S1[$i-1],$S1[$i]);
    print LOG "$S2[$i]\n";
}

print LOG "Distancia total: $distancia2\n";

# print " " . $t;

# }

# print $x;

# while ($t < $tz)
# {
#     $t = proximo_evento($t);
#
#
# }
}
```