

ROTEIRIZAÇÃO DINÂMICA DE VEÍCULOS COM JANELAS DE TEMPO USANDO DE UM ALGORITMO DE COLÔNIAS DE FORMIGAS

Orivalde Soares da Silva Júnior

José Eugenio Leal

Departamento de Engenharia Industrial - PUC-Rio

RESUMO

Sistemas de colônias de formigas (*Ant Colony Systems* - ACS) são modelos matemáticos baseados no comportamento de formigas quando imersas em colônias de indivíduos semelhantes. Formigas são indivíduos simples, porém capazes de interagir entre si, obtendo muitos benefícios desta prática. Estes modelos são muito úteis na resolução de grandes problemas de otimização combinatória, geralmente muito complexos para serem resolvidos por métodos exatos de otimização e representam um incipiente e importante campo de estudos da pesquisa operacional. Neste trabalho serão descritos alguns algoritmos de colônias de formigas, utilizados em problemas de otimização combinatória/discreta. Particularmente, o foco do trabalho será na aplicação destes algoritmos no problema de roteirização dinâmica de veículos com janelas de tempo. Uma implementação do algoritmo no ambiente *Java* foi realizada e testada em problemas usados como *benchmarking* na literatura.

ABSTRACT

Ant Colony Systems are mathematical models based on the behavior of ants when immersed in colonies of likely individuals. Ants are simple individuals, however capable of interacting with each other, obtaining benefits from this practice. These models are very useful in solving large combinatorial optimization problems, usually too complex to be solved by exact optimization methods, and represent an important and incipient field of study in operations research. This work aims to describe some ant colony algorithms, used in combinatorial/discrete optimization problems. Particularly, the focus of this work will be in the application of these algorithms to the dynamic vehicle routing problem with time windows. The algorithm has been implemented in Java environment and tested in problems used as benchmarking in the literature.

1. INTRODUÇÃO.

No problema de roteirização de veículos com janelas de tempo (*Vehicle Routing Problem with Time Windows* - VRPTW), uma frota de veículos com capacidade limitada deve ser roteirizada para atender um conjunto de clientes através de uma sequência com custo mínimo (distância total percorrida ou o tempo total de viagem do veículo) e dentro dos intervalos previamente estipulados pelos clientes. Este problema é conhecido como um problema estático, pois todos os pedidos de serviço e dados dos clientes, tais como localização, demanda e janela de tempo, são conhecidos antes da construção das rotas e estes dados não se alteram.

O problema de roteirização dinâmica de veículos com janelas de tempo (*Dynamic Vehicle Routing Problem with Time Windows* - VRPTW) é uma nova classe de problemas emergiu nos últimos anos, principalmente pelas recentes avanços na comunicação e tecnologias de informação que permitem que os pedidos sejam processados em tempo real. Neste problema, todos ou uma parte dos pedidos de serviço ocorrem enquanto as rotas são executadas. Assim, as informações necessárias para a determinação de rotas eficientes são reveladas dinamicamente ao tomador de decisão, o qual deve planejar o atendimento durante a efetivação das rotas para inclusão de novos pedidos de serviço.

No problema tratado neste trabalho, parte dos pedidos de serviço é conhecida previamente, os quais são utilizados para roteirização inicial. No decorrer do dia, novos pedidos entram no sistema e são atribuídos aos veículos. Assim, o tomador de decisão se comunica periodicamente com motoristas para informar quais serão os próximos pedidos que deverão ser atendidos. Então os motoristas possuirão sempre o conhecimento parcial da sua rota, a qual é construída a cada parcela de tempo.

Considerou-se também que, os veículos que já estão em andamento não precisam retornar ao depósito para processar os novos pedidos. Esta formulação cobre três aplicações do problema de roteirização dinâmico. A primeira aplicação é chamada de sistema expresso (*feeder system*), mais conhecido com *dial-a-ride* e que consiste no transporte de passageiros com requisitos específicos (Gendreau e Potvin, 1998). A segunda trata o problema da coleta realizada pelo serviço de correios, no qual o veículo vazio parte do depósito e coleta as encomendas dos clientes (Gendreau et al., 1999). Na terceira aplicação, os veículos são carregados no depósito geralmente com apenas um tipo de produto e entregam uma parcela da sua carga para cada cliente.

Assim como em Gendreau et al. (1999), desenvolveu-se neste trabalho uma adaptação de um algoritmo estático para roteirização de veículos, considerando uma penalização caso ocorra a violação das restrições de janelas de tempo.

Ichoua et al. (2000) analisaram o efeito de alterar a rota do veículo no momento que surgem novos pedidos. Assim, o veículo pode antecipar o atendimento de um novo cliente antes de seguir para o próximo cliente que já estava definido em sua rota. Este problema pode ser denominado como preemptivo, mas neste trabalho tratou-se o problema não-preemptivo.

Em (Larsen, 2000 e Larsen et al., 2002) os problemas são classificados segundo seu nível de dinamismo (percentual de pedidos dinâmicos e disponibilidade de tempo para reagir aos mesmos) e principal objetivo do tomador de decisão (minimizar a distância percorrida vs. minimizar o tempo de resposta). Esta classificação também foi utilizada neste trabalho para realização de testes computacionais utilizando cinco diferentes graus de dinamismo.

As pesquisas mais recentes sobre o PDRVJT são para aplicações reais e conseqüentemente desenvolvidas diferentes formulações para este problema. Entre as técnicas utilizadas para solução deste problema, verifica-se que as heurísticas de inserção têm sido utilizadas por diversos autores, pois permitem que a inserção de um novo cliente seja na prática, executada com um tempo computacional reduzido (Larsen, 2000).

Propõe-se neste trabalho uma solução baseada no paradigma de Sistemas de Colônias de Formigas (*Ant Colony Systems - ACS*), que são modelos matemáticos baseados no comportamento de formigas quando imersas em colônias de indivíduos semelhantes. Estes algoritmos são muito úteis na resolução de grandes problemas de otimização combinatória, geralmente muito complexos para serem resolvidos por técnicas exatas de otimização.

Esta técnica foi inicialmente abordada por Dorigo *et al* (1991), que propuseram um modelo denominado *Ant System (AS)* para a resolução do problema do caixeiro viajante (*Traveling Salesman Problem - TSP*). Em 1996, Dorigo e Gambardella (1996) criaram o *Ant Colony System (ACS)* a partir do AS, o que representou um novo avanço no desempenho dos algoritmos. Em 1999, Bullnheimer *et al* (1999) aplicaram pela primeira ao Problema de Roteirização de Veículos (VRP). Em 1999, Gambardella *et al* (1999) propõe uma técnica denominada *Multiple Ant Colony System* aplicado ao Problema de Roteirização de Veículos com Janelas de Tempo (MACS-VRPTW). Em 2002, Montemanni et al. (2002), criaram um novo algoritmo para solução do Problema Dinâmico de Roteirização de Veículos baseado no ACS (ACS-DVRP). Este trabalho foi baseado principalmente nos algoritmos MACS-VRPTW e ACS-DVRP.

2. DESCRIÇÃO DO PROBLEMA

2.1 O Problema Estático de Roteirização de Veículos com Janelas de Tempo

O problema estático de roteirização de veículos é definido a seguir: n clientes devem ser atendidos por um único depósito. Cada cliente i solicita uma demanda d_i de produtos, um tempo de serviço s_i e uma janela de tempo $[b_i, e_i]$, sendo que b_i representa o horário de início do atendimento e e_i a hora limite para que um carro chegue ao cliente para ser atendido. Uma frota de veículos V , cada um com a capacidade cap está disponível para atender os clientes. Cada cliente possui um tempo de serviço necessário para atendê-lo. A solução final do problema é um conjunto de rotas.

O problema pode ser modelado matematicamente por um grafo completo $G(N,A)$, onde $N=\{0,1,...,n\}$ é um conjunto contendo todos os nós com o depósito sendo representado pelo nó (0) e os clientes $(1,...,n)$, e A é um conjunto contendo todas as arestas, cada uma contendo o tempo viagem T_{ij} associado.

O problema consiste basicamente em encontrar um conjunto de rotas de custo mínimo para veículos, partindo de um depósito (0), garantindo que todo cliente seja atendido por um e somente um veículo, respeitando as restrições de capacidade dos veículos e das janelas de tempo e retornando ao depósito (0). Além disso, deseja-se encontrar o número mínimo de veículos necessários para a realização da tarefa.

2.1 O Problema Dinâmico de Roteirização de Veículos com Janelas de Tempo

No problema dinâmico de roteirização de veículos, novos pedidos de serviço chegam quando a jornada de um dia de trabalho já iniciou e os veículos já saíram do depósito. Este problema pode ser visto como uma seqüência de problemas estáticos. Em termos matemáticos, este problema pode ser formalizado como uma seqüência de grafos assim como descreve o item 2.1. Cada grafo contém sempre os clientes que devem ser servidos durante o dia.

Os problemas dinâmicos podem ser classificados pelo nível de dinamismo, os quais podem ser fracamente, moderadamente ou fortemente dinâmicos. O grau de dinamismo dod assume um valor entre 0 e 1, o qual pode ser obtido através da expressão (1). Por exemplo, em um sistema totalmente estático, $dod = 0$ e caso seja totalmente dinâmico, $dod = 1$ (Larsen, 2000 e Larsen et al., 2002).

$$dod = \frac{\text{Número de ordens de serviço dinâmicas}}{\text{Número Total de ordens de serviço}} \quad (1)$$

3. O ANT SYSTEM (AS)

Neste algoritmo são realizadas simulações do ciclo de vida de cada formiga, o qual consiste na realização de um caminho a partir do seu ninho até uma fonte de alimento. Cada passo de uma formiga é definido por uma regra probabilística de decisão, pois deve ser escolhido um arco entre todos os outros existentes a partir da sua posição atual. Esta regra baseia-se nos valores da quantidade de feromônios acumulada pela passagem das formigas precedentes e do custo (distância) existente em cada arco. Para cada um destes valores, pode-se dar um peso diferente através dos parâmetros α e β , respectivamente. O cálculo da atratividade de cada arco pode ser descrito matematicamente por:

$$a_{ij}(t) = \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in N_i} [\tau_{il}(t)]^\alpha [\eta_{il}]^\beta} \quad (2)$$

A partir da expressão 1 é obtida a atratividade $a_{ij}(t)$ para uma formiga que se encontra em um nó i qualquer, e para todo nó j pertencente a N_i , onde N_i é o conjunto dos nós adjacentes ao nó i . A variável τ_{ij} representa a quantidade de feromônio acumulada no arco (i,j) , e a variável η_{ij} é igual ao inverso do comprimento do arco (i,j) . O parâmetro t , indica que os valores estão variando a cada iteração t . Uma vez calculados os valores de a de todos os arcos incidentes a i , define-se a probabilidade de uma formiga k escolher percorrer o arco ij como:

$$p_{ij}^k(t) = \frac{a_{ij}(t)}{\sum_{l \in N_i^k} a_{il}(t)} \quad (3)$$

Na expressão 2, N_i^k é o subconjunto de N_i que contém somente os nós que ainda não foram visitados pela formiga k , ou seja, os nós que não constam na memória da formiga.

Para uma formiga selecionar um arco entre todas as possíveis alternativas, calcula-se a probabilidade acumulada a partir das probabilidades de cada arco p_{ij} e sorteia-se um número aleatório entre 0 e 1. O arco escolhido será aquele em que o número sorteado possuir valor maior ou igual à $p_{ij-1}(\text{acumulado})$ e menor que $p_{ij}(\text{acumulado})$.

Após todas as k formigas completarem seus caminhos, a quantidade de feromônios de todos os nos arcos é atualizada. Esta atualização consiste na evaporação dos feromônios e na deposição de novos feromônios, a qual é representada pela seguinte equação:

$$\tau_{ij}(t) \leftarrow (1 - \rho)\tau_{ij}(t) + \sum_{k=1}^m \Delta\tau_{ij}^k(t) \quad (4)$$

Na expressão 3, o parâmetro ρ representa o nível de evaporação dos feromônios. O somatório representa todas as contribuições individuais de cada formiga k que tenha passado pelo arco (i,j) na iteração em questão; $\Delta\tau_{ij}^k$ é a quantidade de feromônios que a formiga k deposita no arco (i,j) , e é definida como o inverso do comprimento total do percurso L_k que a formiga k percorreu. Quanto mais formigas percorrerem por um arco, mais feromônios serão depositados no arco, e o tamanho destas deposições é proporcional à qualidade das soluções encontradas. Matematicamente, tem-se:

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{1}{L_k(t)}, & \text{se } (i,j) \in \text{percurso feito pela formiga } k \text{ na iteração } t \\ 0, & \text{caso contrário} \end{cases} \quad (5)$$

Esta é a idéia básica do AS, aplicado ao problema do caixeiro viajante. A seguir, apresenta-se uma outra meta-heurística que foi desenvolvida tendo como base o AS, conhecida como *Ant Colony System*.

4. O ANT COLONY SYSTEM (ACS)

O *Ant Colony System* consiste num melhoramento do *Ant System*. Uma importante diferença entre AS e o ACS é a atualização dos feromônios. No caso do AS, ela é feita de forma global e unificada, após o término de cada iteração. No ACS a atualização dos feromônios se dá

tanto de forma local após a ação de cada formiga, quanto de forma global após o fim de cada iteração, reforçando apenas o melhor caminho encontrado pelas formigas.

No ACS também realiza-se a alteração no tratamento da deposição da quantidade inicial de feromônios em cada arco (τ_0), a qual é definida como o inverso do comprimento da solução encontrada através do método do *Nearest Neighbor* ($1/L_{nn}$) (Santos e Leal, 2006). Adicionalmente, pode-se multiplicar este comprimento L_{nn} ($L=length$) pelo número de nós do problema, diminuindo ainda mais esta concentração inicial de feromônios ($1/n.L_{nn}$). Com isto, as informações mais antigas vão gradativamente assumindo um peso cada vez menor na decisão atual de cada formiga para que possam surgir melhores soluções.

Entretanto, a principal diferença entre os algoritmos consiste na regra de decisão das formigas. No ACS, esta regra de decisão trabalha simultaneamente com a regra probabilística (desbravadora) do AS e com a regra determinística (exploratória). Tem-se um novo parâmetro chamado $q0$ que varia de 0 a 1, e que define o poder de exploração de novos caminhos. Sorteia-se então um número aleatório q e se este número for maior que $q0$, a regra de decisão da formiga é probabilística, usando as expressões (1) e (2), mas com o expoente α da variável τ_{ij} igual a 1. Por outro lado, se q for menor que $q0$, a regra de decisão da formiga é determinística, baseada no conhecimento disponível sob a forma de depósitos de feromônios e distâncias. Assim, a regra de decisão da formiga será, para todo j pertencente a N_i^k . A expressão 5 define que a formiga escolherá o nó j que maximize o valor de a_{ij} .

$$p_{ij}^k(t) = \begin{cases} 1, & \text{se } j = \arg \max a_{ij}, \text{ onde } a_{ij} = \frac{[\tau_{ij}(t)]^\beta [\eta_{ij}]^\beta}{\sum_{l \in N_i} [\tau_{il}(t)]^\beta [\eta_{il}]^\beta} \\ 0, & \text{caso contrário} \end{cases} \quad (6)$$

O ACS foi a base para o desenvolvimento do MACS-VRPTW, o qual será visto na seção seguinte e que é aplicado ao problema de roteirização de veículos com janelas de tempo.

5. O MULTIPLE ANT COLONY SYSTEM (MACS)

Proposto por Gambardella *et al* (1999), o MACS-VRPTW é uma extensão do ACS para o problema de roteirização de veículos com janelas de tempo. Seu diferencial consiste no tratamento de múltiplos objetivos, os quais são minimizar a quantidade de veículos e a distância total percorrida. Neste caso, o primeiro objetivo possui prioridade sobre o segundo. Este modelo utiliza-se de duas colônias de formigas, onde cada uma visa a otimização de um dos objetivos do problema.

A primeira colônia, denominada ACS-VEI, tenta diminuir o número de veículos utilizados, enquanto a segunda colônia, ACS-TIME, tenta melhorar a solução encontrada por ACS-VEI. Ambas as colônias utilizam depósitos de feromônios independentes, mas elas se comunicam através do compartilhamento da melhor solução encontrada até o momento, operação esta que é gerenciada pelo MACS-VRPTW.

Inicialmente, obtém-se uma solução viável para o VRPTW através da heurística do vizinho mais próximo (*Nearest Neighbor*), a qual é utilizada como referência inicial para redução do tempo de busca pelo MACS-VRPTW. Então a solução é melhorada pelas duas colônias. Dentro de um processo iterativo, a solução inicial é melhorada primeiramente pelo algoritmo ACS-VEI, que tenta encontrar uma solução viável que possua menos veículos que a solução atual. Logo após, a solução encontrada é utilizada pelo algoritmo ACS-TIME, que visa

encontrar alguma solução viável que minimize o tempo total gasto pelo dado número de veículos encontrado pela primeira colônia. Terminada a execução deste segundo algoritmo, o ciclo é reiniciado e continua até que algum critério de término seja atendido. Por exemplo, este critério pode ser o número de iterações.

Conforme proposto por Gambardella et al (1999), o algoritmo deveria usar computação paralela, onde ambas as colônias estariam ativas e procurando por melhores soluções ao mesmo tempo. Visando a simplificação deste processo, conforme a proposta de Santos e Leal (2006) apresenta-se a seguir o algoritmo MACS-VRPTW trabalhando com ciclos (figura 1).

Algoritmo MACS-VRPTW

#quantidade_veiculo(s^*) retorna o número de veículos da solução s^*

$s^* \leftarrow$ Encontre uma solução inicial pela heurística *Nearest_Neighbor*

Enquanto não se cumpre a condição de parada faça:

$v \leftarrow$ #quantidade_veiculos(s^*)

$s^* \leftarrow$ Executa ACS-VEI($v - 1$)

$s^* \leftarrow$ Executa ACS-TIME(v)

Fim enquanto

Figura 1: Algoritmo do MACS-VRPTW

Inicialmente, obtém-se uma solução viável para o VRPTW através da heurística do vizinho mais próximo (*Nearest Neighbor*). Assim, a solução é melhorada pelas duas colônias (algoritmos). Primeiramente pelo algoritmo ACS-VEI, que tenta encontrar uma solução viável que possua menos veículos que a solução atual. Logo após, a solução encontrada é utilizada pelo algoritmo ACS-TIME, que visa encontrar alguma solução viável que minimize o tempo total gasto pelo dado número de veículos encontrado pela primeira colônia. Terminada a execução deste segundo algoritmo, o ciclo é reiniciado e continua até que algum critério de término seja atendido. Por exemplo, este critério pode ser o número de iterações.

Descreveu-se acima a idéia básica do MACS-VRPTW, porém, sua implementação possui muitas peculiaridades que precisarão ser vistas com mais detalhes, os quais podem ser obtidos nos trabalhos de Gambardella *et al.* (1999) e Santos e Leal (2006), que foram utilizados como base para implementação deste algoritmo.

6. O MULTIPLE ANT COLONY SYSTEM DINÂMICO (MACS-DVRPTW)

O MACS-DVRPTW é uma extensão do MACS-VRPTW para o problema dinâmico de roteirização de veículos com janelas de tempo. Seu diferencial consiste na possibilidade de atender novos clientes que requisitam o serviço durante a efetivação das rotas já atribuídas aos veículos, ou seja, depois que os veículos já saíram do depósito. Este modelo consiste na solução de uma seqüência de VRPTW estáticos, os quais são solucionados à medida que eles são requeridos.

6.1 Divisão do Dia de Trabalho em Faixas de Tempo

Assim como Kilby et al. (1998) e Montemanni et al. (2002), utilizou-se o conceito de dividir o dia de trabalho em faixas de tempo de tamanhos iguais. Portanto, para um dia de trabalho que possui um total T segundos é dividido em n_{ts} faixas de tempo, cada faixa de tempo T_{ts} possui T/n_{ts} segundos. Com o decorrer do tempo são recebidos novos pedidos e para cada faixa de tempo é resolvido um problema estático do VRPTW.

Todos os novos pedidos recebidos antes de um dado tempo limite T_{co} (*time of cutoff*) são aceitos e processados durante o dia de trabalho. Os pedidos recebidos depois do tempo limite T_{co} não são atendidos no dia atual e são atribuídos ao próximo dia de trabalho. Por exemplo, se o um dia de trabalho inicia às 7h e termina às 17h (total de 10 horas de trabalho) e a política da empresa consiste em aceitar novos pedidos até metade às 12h, então T_{co} será igual a 5. No primeiro problema estático são considerados apenas os pedidos não atendidos do dia anterior, os quais são utilizados para obter a primeira solução.

No final de cada faixa de tempo, analisa-se a melhor solução encontrada pelo MACS-VRPTW estático e atribuem os novos pedidos aos veículos. Serão atribuídos apenas os pedidos que possuam o tempo de processamento T_p (*Process Time*) dentro da faixa de tempo, ou seja, cada motorista deve conhecer apenas os seus próximos destinos que possuem o horário de chegada somado ao tempo de serviço maior que $T_{ts(i)}$ e menor que $T_{ts(i+1)}$, onde i representa o índice da faixa de tempo atual.

Um problema que surge desta consideração anterior é que ao atender o ultimo pedido, o veículo deverá aguardar o recebimento de seus próximos destinos. Para contornar este problema foi utilizada uma estratégia denominada tempo de compromisso avançado, T_{ac} (*advanced commitment time*) a qual consiste em comprometer as ordens com tempo de processamento iniciando nos próximos $T_{ts} + T_{ac}$ segundos, conforme ilustra a figura 2. Esta estratégia permite que seja comprometido ao menos o próximo destino do veículo.

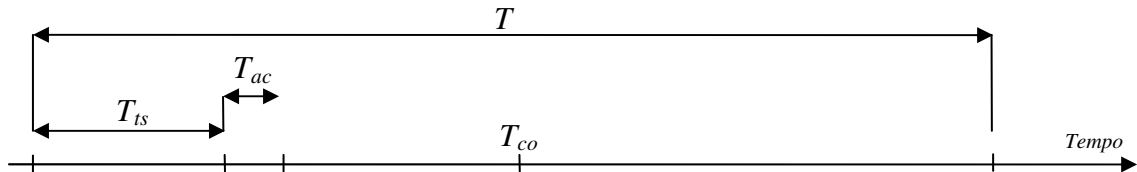


Figura 2: Conceito da divisão do dia em faixas de tempo

Uma exceção desta estratégia ocorre quando o veículo deve retornar ao depósito. Assim, o retorno ao depósito ocorre apenas em duas condições. A primeira ocorre quando todos os clientes do dia de trabalho já foram atendidos e a segunda quando a capacidade do veículo for totalmente utilizada. Quando nenhuma destas condições ocorrerem e o veículo não conhecer seu próximo destino, então ele deverá aguardar novas instruções no cliente atual. Isto ocorre porque as rotas podem ser replanejadas e novos pedidos podem ser atribuídos aos veículos.

Após o primeiro problema estático, no qual os veículos saem dos depósitos com sua capacidade totalmente disponível, cada problema passa a considerar como ponto inicial para suas rotas os últimos pedidos atendidos pelos veículos. São considerados: o momento em que o veículo finalizou o serviço no último pedido atendido e a capacidade residual de cada veículo. Para isto, o algoritmo do MACS-VRPTW teve que ser adaptado para considerar estes novos parâmetros.

A figura 3 ilustra as rotas de dois veículos obtidas a partir da execução de duas instâncias do problema estático. Os pontos representam o tempo de processamento T_p . Estes momentos são utilizados para definir os pedidos que serão comprometidos aos veículos. No primeiro estático, em $T_{ts(0)}$, os veículos iniciam suas rotas a partir do depósito com tempo atual e capacidade iguais a zero. Comprometem-se então os pedidos que estão dentro da faixa de tempo. Os pedidos que estão fora da faixa de tempo atual não são comprometidos com

nenhum veículo e serão utilizados novamente para criação das rotas do problema estático da faixa de tempo seguinte.

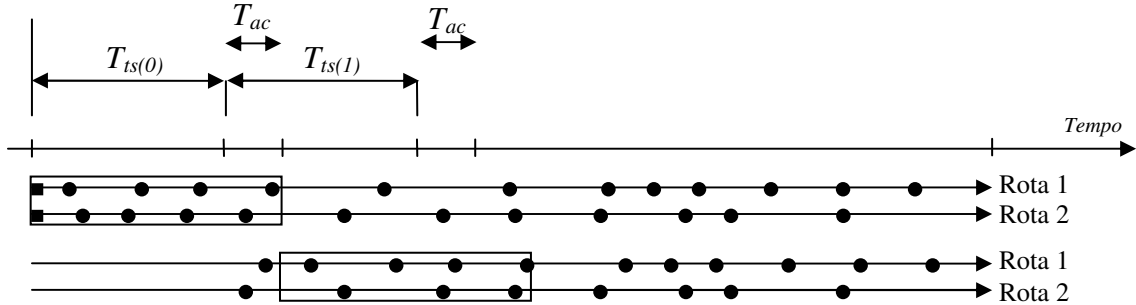


Figura 3: Funcionamento da estratégia de divisão do dia em faixas de tempo

No decorrer da faixa de tempo, novos pedidos são solicitados. Após o seu término, as rotas dos pedidos comprometidos são armazenadas e são atualizadas as posições dos veículos, os tempos atuais e as capacidades atuais. Com base nestas informações é resolvido o segundo problema estático de $T_{ts(1)}$. Comparando com a faixa de tempo anterior, nota-se que foram adicionados dois pedidos na Rota 1 e a Rota 2 manteve os mesmo pedidos definidos anteriormente. Este processo continua até que todos os clientes sejam atendidos.

Uma exceção que pode ocorrer diz respeito à inviabilidade do atendimento de todos os pedidos com o número de veículos atualmente utilizado. Neste caso, pode ser tomada a decisão de não atender algum pedido solicitado recentemente, mesmo que tenha sido solicitado antes do tempo limite T_{co} ou pode ser utilizado um veículo adicional partindo do depósito, caso o depósito possua um veículo disponível.

6.2 Estratégia de Conservação de Feromônios

Como visto anteriormente, o algoritmo de colônia de formigas utiliza a concentração de feromônio depositado pelas formigas para indicar uma boa solução do problema. Este paradigma utilizado no VRPTW estático produz um efeito colateral muito importante, o qual é utilizado como elemento chave no VRPTW dinâmico.

Após o término de cada faixa de tempo é resolvido um problema estático utilizando o MACS-VRPTW, o qual retorna duas matrizes de feromônios. A primeira matriz contém os feromônios resultantes do ACS-VEI e a segunda matriz contém os feromônios resultantes do ACS-TIME. Cada matriz de feromônios possuirá valores elevados nos elementos correspondentes a sequência de pares de clientes que são visitados na melhor solução do problema. No MACS-DVRP, estas matrizes são passadas para o problema estático correspondente à próxima faixa de tempo, uma vez que os problemas são potencialmente muito semelhantes. Esta operação impede que a otimização seja reiniciada toda vez a partir do zero e contribui muito para garantir o bom desempenho necessário no problema dinâmico.

Esta estratégia de transferência da informação dos feromônios é inspirada no trabalho de Guntsch e Middendorf (2002), no qual definem um novo parâmetro γ_r que regula a conservação de feromônios. Para cada par de clientes que aparecem tanto no antigo quanto no novo problema estático, a entrada correspondente na matriz feromônios é inicializada com uma concentração inicial definida pela expressão (7).

$$\tau_{ij} = (1 - \gamma_r) \tau_{ij}^{old} + \gamma_r \tau_0 \quad (7)$$

Onde τ_{ij}^{old} representa o valor do τ_{ij} do antigo problema estático. Sendo assim, a matriz de feromônios não é reiniciada toda vez, mas é reaproveitada para obter as próximas soluções. Para cada novo cliente é criada sua entrada correspondente na matriz de feromônios com a concentração de feromônio inicial τ_0 . Para definição do valor do parâmetro γ_r é importante considerar que as informações antigas possuirão menor importância para as novas soluções, para que as formigas possam explorar novos caminhos.

6.3 Um algoritmo para o MACS-DVRPTW

A estratégia de solução do DVRPTW é resumida com auxílio do algoritmo da figura 4. O conjunto *OrdensPendentes* contém inicialmente todas as ordens conhecidas do dia anterior. As variáveis *Tempo* e *TempoVirtual* são inicializadas com 0 e a posição de todos os veículos é fixado no depósito.

Algoritmo MACS-DVRPTW

Tempo \leftarrow 0

TempoVirtual \leftarrow 0

OrdensPendentes \leftarrow ordens conhecidas antes do início do dia

Enquanto (*OrdensPendentes* $\neq \{\}$ ou *Tempo* $< T_{co}$)

Se (*Tempo* > 0)

TempoVirtual = *Tempo* + T_{ac}

Fim se

ProblemaEstatico \leftarrow problema com *OrdensPendentes* e iniciando em *TempoVirtual*

 Executar MACS-VRPTW no *ProblemaEstatico*

OrdensComprometidas \leftarrow ordens com tempo de processamento \geq *Tempo* + T_{ts} + T_{ac}

 Entregar ordens de *OrdensComprometidas*

OrdensPendentes \leftarrow *OrdensPendentes* \ *OrdensComprometidas*

OrdensPendentes \leftarrow *OrdensPendentes* \cup {Ordens que apareceram nos últimos T_{ts} s}

Tempo \leftarrow *Tempo* + T_{ts}

 Atualiza as posições, capacidade e tempos de viagem dos veículos

 Atualiza a matriz de feromônios

Fim enquanto

Figura 4: Algoritmo do MACS-DVRPTW

Um processo iterativo é iniciado. Se o *Tempo* for maior que 0, então o *TempoVirtual* é atualizado para que sejam obtidas as próximas ordens que o veículo deve atender, conforme a estratégia do tempo de compromisso avançado (*advanced commitment time*). Um problema estático (*ProblemaEstatico*) é criado e resolvido com as ordens contidas em *OrdensPendentes* abrangendo a janela de tempo [*TempoVirtual*; *TempoVirtual* + T_{ts}] e algumas das ordens de serviço são comprometidas com os veículos e inseridas em *OrdensComprometidas*, conforme a estratégia de divisão do tempo (figura 3). *OrdensPendentes* é atualizado, removendo as ordens contidas em *OrdensComprometidas* e inserindo as novas ordens que apareceram nos últimos T_{ts} segundos. A variável *Tempo* é somada ao tamanho da faixa de tempo T_{ts} , as posições, capacidades e tempos de viagem dos veículos. Atualiza-se também a matriz de feromônios utilizando a estratégia de conservação de feromônios, conforme o item 6.2.

7. RESULTADOS COMPUTACIONAIS

O objetivo desta seção é mostrar o desempenho do MACS-DVRPTW quanto à qualidade da solução. Um dos problemas obtidos em Lackner (2004) foi usado ilustrar os testes com o modelo. Os algoritmos foram desenvolvidos na linguagem de programação Java e foram testados num computador com processador Intel Core Duo 1.66GHz e 1GB de RAM.

7.1 Definição dos parâmetros para o MACS-DVRPTW

Para aplicação do algoritmo MACS-DVRPTW aos problemas, foram definidos os parâmetros $\rho = 0,1$, $\beta = 1$, $q_0 = 0,9$, e os parâmetros a , b e c do *Nearest_Neighbor* iguais a 0,6, 0,2 e 0,2 respectivamente. Cada simulação das funções ACS-TIME e ACS-VEI foi feita com até 20 iterações com $k = 10$ formigas cada uma. De acordo com as sugestões de Montemanni et al. (2002), utilizou-se $\gamma_r = 0,3$, $n_{ts} = 25$ e de acordo com Kilby et al. (1998), $T_{co} = 0,5 * T$ e $T_{ac} = 0$.

7.2 Testes com uma Instância proposta por LACKNER

O conjunto de problemas propostos por Lackner (2004) são extensões de 56 problemas desenvolvidas por Solomon para o VRPTW, os quais foram adicionados os tempos em que cada ordem de serviço devem aparecer durante a jornada de trabalho de um dia. Estas extensões utilizaram os graus de dinamicidade iguais a 10, 30, 50, 70 e 90. A figura 5 ilustra os tempos em que aparecem as ordens dinâmicas de acordo com o *dod* para a instância R104.

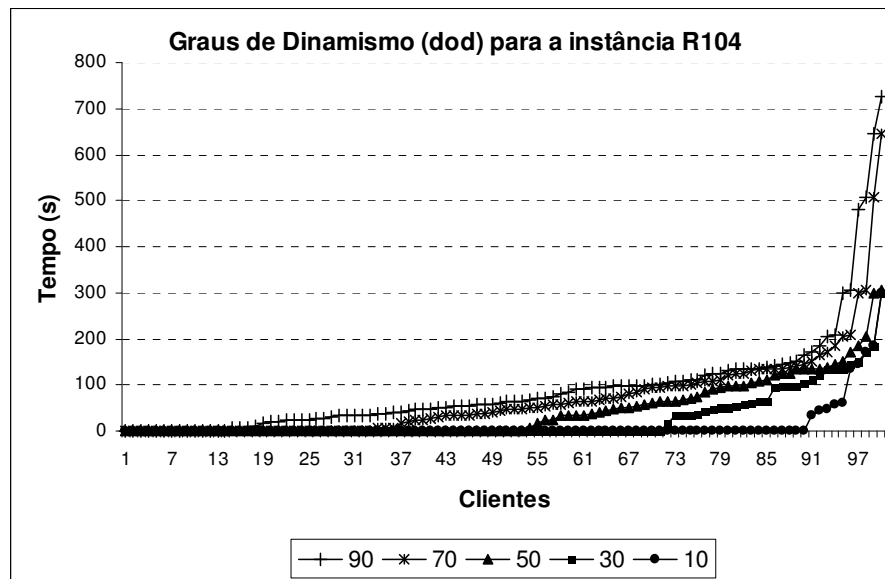


Figura 5: Tempos em que aparecem as ordens dinâmicas de acordo com o grau de dinamismo

Para efeitos ilustrativos, o problema R104 foi resolvido por Lackner (2004) utilizando o mesmo modelo proposto neste trabalho. A figura 6 apresenta as localizações dos clientes com ordens de serviço estáticas e dinâmicas, além do próprio depósito. Neste problema, 30% das ordens de serviço são dinâmicas ($dod=30$).

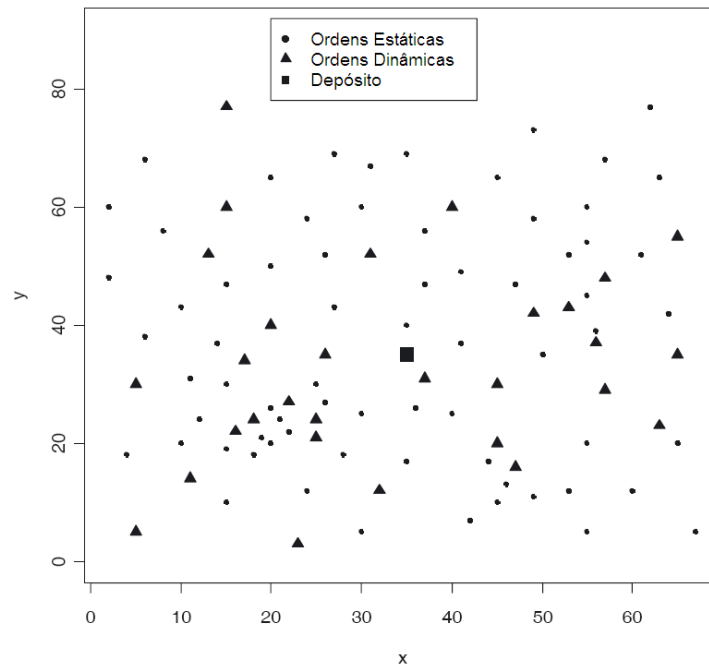


Figura 6: Ordens estáticas e dinâmicas com $dod=30$

A figura 7 apresenta as rotas obtidas para a instância R104 com 100 clientes. Neste problema foram necessários 9 veículos, que é o mesmo número de veículos obtido na melhor solução conhecida utilizando o algoritmo estático. Já para o tempo total (custo) utilizado pelos veículos aumentou de 1.007,24 para 1.256,32. Observa-se que foram atendidas 23 das 30 ordens dinâmicas do problema. Estas 7 ordens dinâmicas não atendidas deverão ser atendidas no próximo dia de trabalho, pois elas apareceram após o tempo limite (T_{co}).

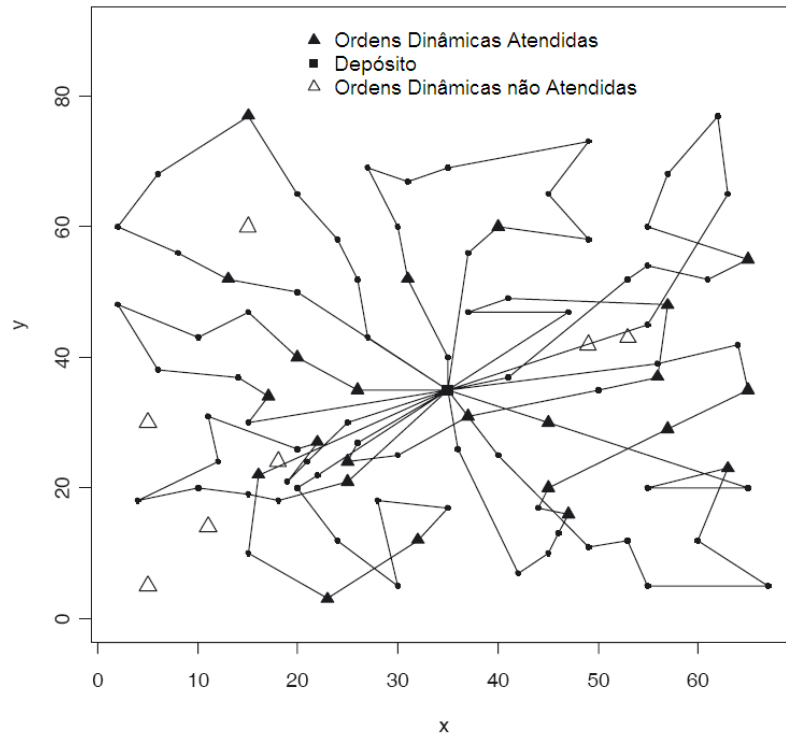


Figura 7: Ordens dinâmicas atendidas e não atendidas

8. CONCLUSÕES.

O objetivo deste trabalho foi apresentar um algoritmo para roteirização dinâmica de veículos com janelas de tempo (DVRPTW) utilizando sistema de colônias de formigas (ACS). Um algoritmo de solução foi proposto, como uma forma de se compreender mais profundamente o funcionamento da meta-heurística e principalmente da estratégia da divisão do dia de trabalho em faixas de tempo. Através de aplicações dos algoritmos a problemas propostos e de análise de seus resultados foi possível concluir que os modelos são realmente capazes de chegar a uma solução próxima da ótima, ou até mesmo a ela própria. A aplicação a uma instância clássica da literatura internacional mostra que o algoritmo permite a obtenção de soluções de boa qualidade.

9. REFERÊNCIAS BIBLIOGRÁFICAS

- Bullnheimer, B.; Hartl, R. F.; Strauss, C. (1999) An improved Ant System algorithm for the Vehicle Routing Problem. *Annals of Operations Research*, Vol 89, No. 0. pp. 319-328.
- Bullnheimer, B.; Hartl, R. F.; Strauss, C. (1999) Applying the Ant System to the Vehicle Routing Problem. In: 2nd International Conference on Metaheuristics MIC-97, Sophia-Antipolis, França, Meta-heuristics: Advances and Trends in Local Search Paradigms for Optimization, pp. 285-296.
- Dorigo, M.; Gambardella, L. M. (1997) Ant Colonies for the Travelling Salesman Problem. *Biosystems*, Vol. 43, No. 2. pp. 73-81.
- Dorigo, M.; Maniezzo, V.; Colormi (1991) A. Positive feedback as a search strategy. Milão: Dipartimento di Elettronica e Informatica, Politecnico di Milano, Itália, Relatório Técnico
- Gambardella, L. M.; Taillard, É.; Agazzi, G. (1999) MACS-VRPTW: A Multiple Ant Colony System for vehicle routing problems with time windows. In: *New Ideas in Optimization*. Londres: McGraw-Hill, pp. 63-76.
- Gendreau, M. ; Guertin, F. ; Potvin, J.Y. ; Taillard, E. (1999). Parallel tabu search for real-time vehicle routing and dispatching. *Transportation Science*, Vol. 33, No 4, pp. 381-390.
- Gendreau, M.; Potvin, J.Y. (1998) Dynamic vehicle routing and dispatching. in *Fleet management and logistic*. T.G. Crainic and G. Laporte editors, pp. 115-226.
- Guntsch, M.; Middendorf, M. (2001) Pheromone modification strategies for ant algorithms applied to dynamic TSP. In E.J.W. Boers et al., editor, *Application of evolutionary computing: Proceedings of EcoWorkshops*, Vol. Lecture Notes in Computer Science 2037, pp. 213-222.
- http://www.fernuniversitaet-hagen.de/WINF/inhalte/benchmark_data.htm Benchmark data sets for dynamic vehicle routing problems. Acesso em 28 de junho de 2009.
- Ichoua, S.; Gendreau, M.; Potvin J.Y. (2000) Diversion issues in real-time vehicle dispatching. *Transportation Science*, Vol. 34, No 4, pp. 426-438.
- Kilby, P.; Prosser, P.; Shaw, P. (1998) Dynamic VRPs: a study of scenarios. Technical Report APES-06-1998, University of Strathclyde, U.K.
- Lackner, A. (2004): “Dynamische Tourenplanung mit ausgewählten Metaheuristiken“, in *Göttinger Wirtschaftsinformatik*, Herausgeber: J. Biethahn, M. Schumann, Band 47.
- Larsen, A. (2000) The Dynamic Vehicle Routing Problem. PhD Thesis-Denmark Technical University, Lyngby.
- Larsen, A.; Madsen O.; Solomon M. (2002) Partially Dynamic Vehicle Routing-Models and Algorithms. *Journal of the Operational Research Society*, Vol. 53, pp. 637-646.
- Montemanni, R.; Gambardella, L. M.; Rizzoli, A. E.; Donati, A. V. (2002): “A new algorithm for a Dynamic Vehicle Routing Problem based on Ant Colony System”, in *Journal of Combinatorial Optimization*, Springer Netherlands, Vol. 10, No 4, pp. 327-343.
- Santos, R. L. (2006) Uma Aplicação de Algoritmos de Colônias de Formigas em Problemas de Roteirização de Veículos com Janelas de Tempo. Rio de Janeiro, 86p. Dissertação de Mestrado – Departamento de Engenharia Industrial, PUC-Rio.
- Solomon, M. M. (1987) Algorithms for the Vehicle Routing and Scheduling Problems with Time Windows Constraints. *Operations Research*, 35, 2, 254-2.