

UMA METODOLOGIA HÍBRIDA BASEADA EM COLÔNIA DE FORMIGAS, BUSCA TABU E RECONEXÃO POR CAMINHOS APLICADA AO PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM JANELAS DE TEMPO

Marcelo Caramuru Pimentel Fraga - caramuru@hotmail.com

Sérgio Ricardo de Souza - sergio@dppg.cefetmg.br

Alessandra Martins Coelho - ale_contatos@ig.com.br

Centro Federal de Educação Tecnológica de Minas Gerais - CEFET-MG

Av. Amazonas, 7675 - Nova Gameleira / Belo Horizonte - MG / 30510-000 - Brasil

Resumo. *Este artigo apresenta uma metodologia híbrida, baseada nas metaheurísticas de colônia de formigas e busca tabu e na técnica de reconexão por caminhos, para resolver o problema de roteamento de veículos com janela de tempo (PRVJT). O PRVJT é uma variação do problema de roteamento de veículos capacitado (PRVC) em que, além das limitações inerentes ao PRVC, o início dos serviços em cada consumidor deve ser iniciado em um intervalo de tempo chamado janela de tempo. A solução para este problema consiste, primeiro, em encontrar o número mínimo de veículos capaz de atender a todos os consumidores e, segundo, em determinar um conjunto de rotas que minimize a distância coberta pelos veículos. A metaheurística de colônia de formigas realimenta-se, em parte, de seus próprios resultados, conduzindo à convergência e à precisão dos resultados gerados. Já a metaheurística busca tabu depende fortemente de sua solução inicial e sua associação com colônia de formigas melhora os resultados de ambas. Periodicamente, além disso, a técnica de reconexão por caminhos é aplicada, para introduzir atributos bons das melhores soluções encontradas nas soluções presentes. A metodologia desenvolvida é validada a partir dos problemas-teste propostos por Solomon (1987).*

Palavras-Chave: *Colônia de Formiga; Busca Tabu, Reconexão por caminhos; Problema de Roteamento de Veículos; Metaheurística.*

1 INTRODUÇÃO

Este trabalho aborda o Problema de Roteamento de Veículos com Janela de Tempo (PRVJT), através de uma metodologia híbrida entre as metaheurísticas Colônia de Formigas e Busca Tabu, aliadas ainda à técnica de Reconexão por Caminhos (*Path Relinking*).

O PRVJT é uma variação do Problema de Roteamento de Veículos clássico e pode ser descrito como um problema no qual uma frota de veículos, inicialmente situada em um depósito, deve atender a um conjunto de consumidores que possuem diferentes demandas por produtos a serem distribuídos utilizando-se a frota. O atendimento a cada consumidor deve ser iniciado em um intervalo de tempo denominado janela de tempo. A solução para esse problema consiste em encontrar o número mínimo de veículos, capaz de atender a todos os consumidores, e um conjunto de rotas, que minimize a distância total percorrida pelos veículos. O PRVJT é classificado como um problema NP-Difícil, o que justifica o uso de métodos heurísticos para sua resolução. Entretanto, sofre das dificuldades inerentes à indefinição da qualidade da solução determinada,

relativas a estes métodos. Uma revisão de métodos heurísticos pode ser encontrada em Blum & Roli (2003).

Os algoritmos de Colônia de Formigas (ACO ou *Ant Colony Optimization*) foram desenvolvidos a partir da observação do comportamento de formigas reais na natureza. Ao longo dos anos, têm sido aplicados em problemas combinatoriais diversos. Uma revisão recente da metaheurística ACO é apresentada em Dorigo & Blum (2005).

A Busca Tabu (BT) é uma metaheurística robusta, mas depende fortemente da solução inicial para obter bons resultados. A partir de um ponto inicial mais avançado, a BT tem maiores chances de encontrar boas soluções que o algoritmo de Colônia de Formigas atuando sozinho, pois este último parte da construção até a solução final e, diante do grande número de soluções possíveis, pode ficar preso a ótimos locais. A integração da Busca Tabu à metaheurística Colônia de Formigas proporciona ainda outra vantagem, pois os resultados gerados irão influenciar as próximas fases de construção, contribuindo para a convergência e melhoria na qualidade dos resultados.

Reconexão por Caminhos é uma estratégia de intensificação da busca sobre resultados gerados, explorando trajetórias que conectam soluções elite, à procura de uma solução de melhor qualidade. A partir de uma solução-corrente e em direção a uma solução-guia, busca agrupar em uma única solução bons atributos encontrados em ambas soluções.

O foco deste trabalho é apresentar a metodologia Ant-TPR, desenvolvida a partir da junção entre as metaheurísticas ACO e Busca Tabu e associadas a técnica de Reconexão por Caminhos, de forma a produzir resultados competitivos com os melhores encontrados na literatura para a classe de problemas PRVJT. Ele está organizado como segue: a seção 2 a seguir introduz o problema de Roteamento de Veículos e sua variante considerada, qual seja, o problema de roteamento de veículos com janela de tempo (PRVJT). A seção 3 apresenta a metaheurística de colônia de formigas e discute sua formulação. A seguir, a seção 4 mostra as definições próprias da metaheurística Busca Tabu, enquanto que na seção 5 a técnica de intensificação Reconexão de caminhos é introduzida. A seção 6 representa o foco central desse trabalho, ao apresentar a metodologia híbrida Ant-PR, desenvolvida pela combinação das estruturas anteriormente apresentadas. A seção 7 mostra a aplicação da metodologia proposta aos problemas-teste introduzidos em Solomon (1987). Finalizando, a seção 8 mostra as conclusões a respeito do estudo realizado.

2 O PROBLEMA DE ROTEAMENTO DE VEÍCULOS

No Problema de Roteamento de Veículos Capacitados (PRVC) uma frota de veículos, localizada inicialmente em um depósito, deve atender a um conjunto de consumidores que possuem diferentes demandas por produtos a serem distribuídos pela frota. A única restrição presente no PRVC é a capacidade de carga dos veículos que, obviamente, não pode ser violada. As demandas são conhecidas *a priori* e não podem ser fracionadas em sua entrega, ou seja, cada consumidor não pode ser visitado mais de uma vez para entrega de produtos. Todos os veículos possuem a mesma capacidade de carga e suas rotas iniciam e terminam no depósito. O Problema de Roteamento de Veículos com Janelas de Tempo (PRVJT) é uma variação do PRVC na qual, além da restrição de capacidade, são adicionadas restrições relacionadas ao horário em que cada consumidor deve ser atendido. Para cada consumidor i , é associado um intervalo de tempo $[a_i, b_i]$, chamado de janela de tempo, que indica o horário em que deve ser iniciado o atendimento e, além disso, um tempo de serviço (s_i), que indica o período de tempo no qual o veículo deve aguardar a conclusão das tarefas. O PRVJT consiste em designar um conjunto de rotas que atenda a essas restrições, minimize o número total de veículos utilizado e o custo total da rota.

3 COLÔNIA DE FORMIGAS

Os algoritmos baseados em Colônia de Formigas (ou *Ant Colony Optimization* - ACO) têm origem nos trabalhos de Marco Dorigo (veja Dorigo et al. (1991)), inspirados no comportamento das formigas na natureza, que propôs o algoritmo *Ant System* (AS) para solucionar o Problema do Caixeiro Viajante (PCV) (veja, para uma revisão completa, Dorigo & Blum (2005)).

Também de acordo com Dorigo et al. (1991), as formigas são capazes de encontrar o caminho mais curto entre a fonte de alimento e o formigueiro, através da cooperação entre os indivíduos e comunicação indireta. O comportamento das formigas na natureza é descrito da seguinte forma: inicialmente, as formigas exploram aleatoriamente a área ao redor do formigueiro à procura de comida. Enquanto se deslocam, depositam sobre o solo uma substância volátil chamada feromônio, que as auxilia a encontrar o caminho de volta ao formigueiro. Desta forma, quando uma formiga estabelece uma trilha ou caminho entre a fonte de alimento e o formigueiro, o caminho percorrido ficará marcado por um rastro desta substância. As demais formigas à procura de alimento detectam a presença de feromônio no solo e tendem a escolher o caminho com a maior concentração do mesmo. As formigas que escolheram o caminho mais curto farão o percurso em menor tempo e o rastro de feromônio será reforçado com uma frequência maior que nos caminhos mais longos. Assim, os caminhos mais eficientes, ou seja, de menor distância percorrida, receberão maior quantidade de feromônio e tenderão a ser os mais escolhidos. Por ser uma substância volátil, a evaporação do feromônio evita que, com o tempo, um caminho que não esteja sendo mais utilizado continue a influenciar a decisão das formigas.

3.1 ANT COLONY SYSTEM

O algoritmo *Ant Colony System* (ACS) foi introduzido em Gambardella & Dorigo (1996), Dorigo & Gambardella (1997a) e Dorigo & Gambardella (1997b), de forma a melhorar o desempenho do algoritmo *Ant System*.

Um critério de seleção, chamado pseudo-randômico-proporcional, no qual o parâmetro q_0 controla a formação das soluções, se de forma probabilística ou de forma gulosa, conforme a expressão:

$$s = \begin{cases} \arg \max_{u \notin M} \left\{ [\tau_{(r,u)}]^\alpha \cdot [\eta_{(r,u)}]^\beta \right\}, & \text{se } q \leq q_0; \\ \text{expressão (2),} & \text{caso contrário.} \end{cases} \quad (1)$$

Um número aleatório q é gerado e comparado ao parâmetro q_0 , cada vez que a formiga necessita escolher o próximo nó para se mover. O algoritmo funciona de forma gulosa se o valor de q for menor que o parâmetro q_0 . Desta forma, a geração da solução valoriza o aprendizado gerado pela deposição do feromônio. Caso o valor de q seja maior que q_0 , a geração da solução será de feita de forma probabilística, segundo a expressão 2.

$$p_k(r, s) = \begin{cases} \frac{[\tau_{(r,s)}]^\alpha \cdot [\eta_{(r,s)}]^\beta}{\sum_{u \notin M_k} [\tau_{(r,u)}]^\alpha \cdot [\eta_{(r,u)}]^\beta}, & \text{se } s \notin M_k \\ 0, & \text{caso contrário} \end{cases} \quad (2)$$

Essa expressão é apresentada em Dorigo et al. (1991). Nela, tem-se que:

- $p_k(r, s)$: probabilidade da formiga k mover-se do nó r para o nó s ;
- $\tau_{(r,s)}$: quantidade de feromônio associado ao arco (r, s) ;
- $\eta_{(r,s)}$: atratividade do arco (r, s) ;
- U : conjunto de arcos não visitados;
- M_k : lista tabu da formiga k ;
- α e β : parâmetros de controle da influência de $\tau_{(r,s)}$ e $\eta_{(r,s)}$. Variam na faixa $[0, 1]$.

Na expressão 2, a cada iteração as formigas decidem, de acordo com uma medida probabilística, qual o caminho a seguir. A probabilidade de escolha de um caminho é proporcional ao rastro de feromônio e à atratividade do mesmo - que varia de acordo com o tipo e a modelagem do problema em que o algoritmo está sendo aplicado. Cada formiga tem uma memória que armazena os caminhos percorridos por ela e previne que um caminho seja visitado pela formiga mais de uma vez. Se a formiga já percorreu aquele caminho, a probabilidade de escolha é zero; caso contrário, é positiva. No roteamento de veículos, o parâmetro atratividade é chamado de visibilidade e é calculado como o inverso da distância entre os consumidores ou vértices do arco.

A atualização do feromônio no ACS é feita de duas formas diferentes: global e local. A atualização global é feita para privilegiar apenas os caminhos pertencentes à melhor solução global, ou seja, a cada iteração, após todas as formigas terem gerado suas soluções, apenas a formiga que gerou a melhor solução desde o início do algoritmo irá depositar feromônio. A atualização global do feromônio é feita pela expressão:

$$\tau_{(r,s)} = (1 - \rho) \cdot \tau_{(r,s)} + \rho / J_{\Psi}^{gb}, \quad \forall (r, s) \in \Psi^{gb} \quad (3)$$

na qual:

- $\tau_{(r,s)}$: quantidade de feromônio associado ao arco (r, s) ;
- ρ : taxa de evaporação, tal que $\rho \in [0, 1]$;
- J_{Ψ}^{gb} : menor distância percorrida, computada desde o início da aplicação do algoritmo;
- Ψ^{gb} : melhor solução computada desde o início da aplicação do algoritmo.

Segundo Ellabib et al. (2003), a fase de evaporação do algoritmo AS é substituída no ACS por uma atualização local do feromônio, como pode ser visto na expressão 4. Cada vez que uma formiga se move de um arco para outro, a quantidade de feromônio associado ao arco é reduzida. O resultado da atualização local é a modificação dinâmica da possibilidade de escolha do caminho pelas formigas. Logo, cada vez que um arco é percorrido por uma formiga, ele se torna ligeiramente menos desejável para as demais.

$$\tau_{(r,s)} = (1 - \rho) \cdot \tau_{(r,s)} + \rho \cdot \tau_0 \quad (4)$$

para

- $\tau_{(r,s)}$: quantidade de feromônio associado ao arco (r, s) ;
- ρ : taxa de evaporação, de modo que $\rho \in [0, 1]$;

- τ_0 : quantidade de feromônio inicial em todos os arcos.

De acordo com Gambardella et al. (1999), um bom valor inicial para τ_0 é dado por:

$$\tau_0 = \frac{1}{nJ_{\Psi}^h}$$

sendo J_{Ψ}^h a distância percorrida, calculada pela heurística do Vizinho mais Próximo, introduzida por Flood (1956), e n é o número de nós.

Um procedimento chamado *Daemon Actions* pode ser usado, opcionalmente, para um controle centralizado da tomada de decisões pelo ACS, em situações como ativar um procedimento de busca local ao final de cada iteração ou depositar uma quantidade extra de feromônio sobre a melhor solução. Esses são exemplos de decisões que não podem ser tomadas pelas formigas individualmente.

4 BUSCA TABU

A metaheurística Busca Tabu (BT) foi proposta, de forma independente, por dois autores, Glover (1986) e Hansen (1986), e é conhecida como uma das metaheurísticas mais robustas para a resolução de problemas combinatoriais.

A partir de uma solução inicial qualquer, a BT vasculha toda a vizinhança da mesma a procura pelo melhor vizinho. O melhor vizinho encontrado, mesmo que seja uma solução pior que a atual, se torna a nova solução corrente e o processo é repetido até que algum critério de parada seja atingido.

Um problema que surge ao aceitar movimentos de piora e escolher sempre o melhor vizinho é a ciclagem. Um exemplo de ciclagem ocorre quando um ótimo local é atingido, a BT moverá então para seu melhor vizinho, o qual possui uma solução de pior qualidade. Quando este melhor vizinho se tornar a solução corrente e a BT vasculhar a vizinhança a partir do mesmo, o novo melhor vizinho será novamente o ótimo local encontrado na iteração anterior.

O uso de uma estrutura de memória, conhecida como lista tabu, onde são armazenados as soluções visitadas, evita que a ciclagem ocorra ao proibir que a busca retorne às soluções que constam na lista. Entretanto, o tempo de processamento de todos os dados pertencentes à solução pode tornar o processo demasiado lento ou até mesmo o inviabilizar. Para tornar o processo menos custoso computacionalmente, a lista tabu, geralmente, é feita de forma a armazenar apenas os movimentos efetuados na exploração da vizinhança, reduzindo a quantidade de parâmetros a serem comparados e, conseqüentemente, o tempo computacional necessário. Uma lista tabu de movimentos armazena os movimentos proibidos (ou tabus) que, quando realizados, podem levar a uma solução já analisada anteriormente.

O número de iterações que um movimento estará proibido de ser realizado é definido pelo tamanho da lista tabu. Uma lista tabu muito grande pode restringir demais a busca, pois ao tornar tabu um grande número de movimentos, o procedimento termina de forma prematura por não existir movimentos factíveis a realizar. Por outro lado, uma lista tabu muito pequena aumenta as chances do processo ciclar, pois reduz o número de iterações em que o movimento é proibido de ser realizado.

Por isso, a BT deve manter um rigoroso controle sobre o tamanho da lista tabu e uma forma de se contornar estes problemas é a implementação de uma lista de tamanho variável. Também chamada de lista tabu dinâmica, ela tem seu tamanho alterado periodicamente no decorrer do processo, tentando evitar que a busca fique muito restrita ou cicle.

Apesar de todos os esforços para manter uma lista tabu de tamanho adequado, é possível que um movimento considerado tabu leve a uma solução não analisada ainda. Neste caso,

uma função de aspiração pode tornar possível a aceitação de um movimento pertencente a lista tabu, desde que leve a uma solução desconhecida e com bons atributos. O critério de aspiração mais comum na Busca Tabu é baseado na função objetivo, ou seja, o movimento tabu é aceito somente se levar a uma solução melhor do que a melhor solução existente até o momento.

Para uma referência mais detalhada sobre Busca Tabu, consultar Glover & Laguna (1997). Aplicações recentes dessa metaheurística ao PRVJT podem ser vistas em Fraga et al. (2005) e Costa (2005).

5 RECONEXÃO POR CAMINHOS

A técnica de Reconexão por Caminhos (*Path Relinking*) foi introduzida por Glover (1996) como uma estratégia de intensificação dos resultados, através da exploração de trajetórias que conectam soluções elite (melhores soluções geradas).

A idéia principal do método é unir os bons atributos pertencentes as melhores soluções em uma única solução. Ele funciona gerando e explorando caminhos no espaço de soluções, partindo de uma solução corrente até uma solução elite, através de movimentos que introduzem atributos da solução elite na solução corrente. Reconexão por Caminhos pode ser usada como um processo de intensificação a cada ótimo local encontrado e, de acordo com Glover & Laguna (1997), essa é a forma mais eficiente de utilização do método.

A técnica é aplicada em pares de soluções $(s_1; s_2)$, sendo s_1 a solução corrente, obtida após um procedimento de busca local e s_2 uma solução escolhida aleatoriamente de um limitado conjunto de soluções elite, gerado durante a construção das soluções. Cada solução obtida ao final de uma busca local é considerada como uma candidata a ser inserida no conjunto elite, desde que ela seja melhor que a solução de pior qualidade do conjunto e apresente um percentual mínimo de diferença em relação a cada solução do conjunto elite. Se o conjunto estiver vazio, a solução é simplesmente inserida no conjunto. Se o conjunto elite já possui o número máximo de soluções e a solução corrente é candidata a ser inserida neste conjunto, então esta substitui a solução de pior qualidade.

O método de Reconexão por Caminhos inicia-se computando a diferença de simetria, dada por $\Delta(s_1; s_2)$, entre duas soluções s_1 e s_2 , resultando em um conjunto de movimentos que deve ser aplicado a uma delas, dita solução inicial, para "transformar-se" na outra, dita solução guia. A partir da solução inicial, o melhor movimento ainda não executado de $\Delta(s_1; s_2)$ é aplicado à solução corrente, até que a solução guia seja atingida. A melhor solução encontrada ao longo dessa trajetória é considerada como candidata à inserção no conjunto elite e a melhor solução já encontrada é atualizada. Uma aplicação da técnica de Reconexão por Caminhos ao PRVJT pode ser vista em Ho & Gendreau (2006).

6 METODOLOGIA

A metodologia Ant-TPR proposta consiste em, inicialmente, definir os movimentos de exploração da vizinhança e tipo de refinamento a serem utilizados. Em seguida, é definida uma função de avaliação, para controle da qualidade das soluções geradas. Por fim, a descrição geral do algoritmo desenvolvido é realizada. Para maiores detalhes sobre a metodologia, consultar Fraga (2006).

Um movimento de realocação consiste em retirar um consumidor de uma rota qualquer e inseri-lo em uma outra posição, que pode ser tanto dentro da mesma rota quanto em uma outra rota distinta. Na fase de refinamento foi utilizada a técnica de busca local descida com primeira melhor.

A função de avaliação utilizada é lexicográfica, ou seja, formada por um conjunto de objetivos distintos, classificados segundo um critério de prioridades. Nesta metodologia, dois

objetivos são considerados, na seguinte ordem de prioridade: primeiro, deseja-se minimizar o número de veículos utilizados e segundo, minimizar a distância total percorrida. Cada nova solução gerada é comparada com a anterior, avaliando-se primeiramente o critério de maior prioridade e, somente caso a avaliação com relação a este critério produza um resultado melhor ou igual à solução anterior, o próximo objetivo é avaliado.

A Reconexão por Caminhos foi implementada da seguinte forma. São inseridos, gradativamente, arcos da solução guia s_2 em uma solução corrente s_1 . Em seguida, procura-se, na solução s_1 , a partir do ponto de inserção, os consumidores pertencentes ao arco incluído e os remove. Por fim, uma heurística de busca local é aplicada, somente a partir do ponto em que o arco foi inserido.

No algoritmo criado, a cada iteração, uma lista restrita de consumidores candidatos (LRCC) é gerada, contendo todos os consumidores (com exceção do depósito) da instância a ser solucionada. Além disso, uma lista tabu para nascimento de novas formigas (LTPNNF) e uma lista tabu geral das formigas (LTGF) (que representa a memória da formiga) são geradas, contendo, inicialmente, apenas o depósito. Um vetor de rotas geral (VRG) é criado e, inicialmente, não contém nenhuma rota. Em seguida, uma formiga é atribuída a um consumidor qualquer, escolhido aleatoriamente dentre os consumidores da LRCC, de modo que o consumidor selecionado nunca pertencerá a LTPNNF. A LTPNNF evita que, em uma futura iteração, uma formiga seja colocada ou em uma posição que não permita a formação de novas rotas ou em uma posição já colocada anteriormente.

Um vetor de rotas para a formiga é então iniciado e, caso o consumidor escolhido já pertença a alguma rota, o vetor rota da formiga (VRF) receberá, desse modo, a rota do consumidor. A formiga escolhe, dentre os consumidores da LRCC, para qual consumidor se mover. Cada formiga possui sua própria lista tabu (LTF) que, inicialmente, é uma cópia da LTGF e para onde serão movidos os consumidores analisados. A função da LTGF é evitar que as formigas visitem consumidores que estejam no meio de rotas, ou seja, que não estejam ou na primeira ou na última posição dos vetores de rotas das formigas. A união entre rotas geradas pelas formigas só é possível quando os consumidores envolvidos estão nas extremidades de cada rota. Quando o consumidor escolhido pela formiga já pertencer a uma rota, o algoritmo selecionará a rota do consumidor escolhido e tentará realizar a união.

A cada tentativa mal-sucedida de movimento da formiga (violação de qualquer uma das restrições existentes), a LTF receberá o consumidor envolvido, para que ela não retorne aos consumidores já analisados. Caso o movimento da formiga de um consumidor para outro respeite todas as restrições, a LTGF e o VRG são atualizados, e a LTF recebe novamente a LTGF. A formiga então, move-se para o último consumidor da rota gerada e continua seus movimentos, até que não existam mais movimentos possíveis, ou seja, até que todos os consumidores pertençam à LTF. A formiga, então, é eliminada; sua rota gerada fica armazenada no VRG; a LRCC e a LTPNNF são atualizadas de acordo com a LTGF; e uma nova formiga será gerada. Quando não existirem mais consumidores na LRCC, o custo da solução gerado pelo conjunto de formigas é analisado e a matriz de feromônio é atualizada.

Em lugar de gerar um conjunto fixo de soluções por iteração, são geradas novas soluções, até que surja uma solução melhor que a melhor solução da iteração anterior ou até um valor escolhido. A solução encontrada é refinada e, após certo número de iterações, o processo de Busca Tabu é acionado sempre que ocorra uma melhora no valor da função objetivo. Periodicamente, o processo de reconexão por caminhos é acionado tentando trazer bons atributos da melhor solução existente para a solução corrente.

Tabela 1: Ant-TPR vs. outras metodologias

Metod. / Grupo	R1	R2	C1	C2	RC1	RC2	Total
<i>TS – P</i> Potvin et al. (1996)	12,6 ^a	3,1	10	3	12,6	3,4	427 ^d
	1294,70 ^b	1185,90	861,00	602,50	1465,00	1476,10	64679 ^e
	639 ^c	722	435	431	586	662	32957 ^f
<i>TS – T</i> Taillard et al. (1997)	12,17	2,82	10	3	11,5	3,38	410
	1209,35	980,27	828,38	589,86	1389,22	1117,44	57953
	13774	20232	14630	16375	11264	11596	833390
<i>MACS</i> Gambardella et al. (1999)	12	2,73	10	3	11,63	3,25	407
	1217,73	967,75	828,38	589,86	1382,42	1129,19	57525
	1800	1800	1800	1800	1800	1800	100800
<i>GA</i> Berger et al. (2001)	11,92	2,73	10	3	11,5	3,25	405
	1221,10	975,43	828,48	589,93	1389,89	1159,37	57523
	-	-	-	-	-	-	-
<i>HGA</i> Chen et al. (2001)	13,20	5,00	10,10	3,25	13,50	5,00	478
	1227,00	980,00	861,00	619,00	1427,00	1223,00	59405
	-	-	-	-	-	-	84000
<i>IACS</i> Chen & Ting (2005)	12,83	3,09	10	3	12,5	3,75	432
	1203,56	932,23	828,76	589,86	1363,84	1079,81	56429
	425	437	239	363	403	370	21146
Ant-TPR (Geral)	12,25	2,82	10,00	3,00	12,13	3,38	416
	1211,61	968,77	828,46	589,86	1358,84	1119,93	57201
	2857,21	3142,06	847,80	606,76	2420,04	2635,30	121776

7 RESULTADOS COMPUTACIONAIS

O algoritmo Ant-TPR foi implementado em linguagem C++, com o compilador Builder 5.0 da Borland, e executado em um computador Pentium 4 2.4 GHz, com 512 MB de memória RAM, sob plataforma Windows XP Pro. Foram utilizadas, para teste, os problemas-teste introduzidos por Solomon (1987), que são utilizados como referência de desempenho no PRVJT. O algoritmo Ant-TPR desenvolvido foi avaliado em cinco intervalos de tempo pré-definidos: 100, 600, 1200 e 3600 segundos com a finalidade de conhecer o comportamento da metodologia em diversas situações. Para cada intervalo de tempo, foram feitas dez execuções por instância, cada uma delas partindo de uma semente diferente de números aleatórios.

Os melhores resultados produzidos pela metodologia Ant-TPR em cada instância podem ser vistos na tabela 1 (retirada de Chen & Ting (2005)), onde (*a*) é o valor médio do número de veículos da solução; (*b*) é a média da distância total percorrida; e (*c*) é a média do tempo computacional gasto, em segundos; (*d*) é a soma do número de veículos de todas as soluções; (*e*) é a distância total percorrida e (*f*) é o tempo computacional total. Os valores são referentes as melhores soluções encontradas durante as execuções.

Nas tabelas 2 e 3 são feitas comparações com a melhor implementação de colônia de formigas para o PRVJT (MACS) e a metodologia GA detentora dos melhores resultados para o problema. Nos diversos intervalos de tempo testados a metodologia Ant-TPR variou entre pouco menos de 3% a 5% dos melhores resultados, demonstrando a robustez do método.

Tabela 2: Ant-TPR vs. MACS (Gambardella et al. (1999))

Ant-TPR x MACS	Ant TPR Final	Ant TPR 100s	Ant TPR 600s	Ant TPR 1200s	Ant TPR 1800s	Ant TPR 3600s
Número de veículos	2,21%	4,42%	3,44%	2,95%	2,70%	3,19%
Distância percorrida	-0,56%	4,02%	1,44%	0,44%	0,02%	-1,00%

Tabela 3: Ant-TPR vs. GA (Berger et al. (2001))

Ant-TPR x GA	Ant TPR Final	Ant TPR 100s	Ant TPR 600s	Ant TPR 1200s	Ant TPR 1800s	Ant TPR 3600s
Número de veículos	2,72%	4,94%	3,95%	3,46%	3,21%	3,70%
Distância percorrida	-0,56%	4,02%	1,44%	0,45%	0,02%	-1,00%

8 CONCLUSÕES

Neste trabalho, foi apresentado um algoritmo híbrido denominado Ant-TPR para solucionar o Problema de Roteamento de Veículos com Janela de Tempo (PRVJT). O algoritmo desenvolvido é baseado nas metaheurísticas Ant Colony Optimization e Busca Tabu e associado ainda a técnica de Reconexão por Caminhos. Na metodologia desenvolvida, as soluções para o PRVJT são geradas pela metaheurística ACO, refinadas com busca local e, caso ocorram melhoras, também com Busca Tabu. Periodicamente, a técnica de Reconexão por Caminhos é acionada, para intensificar a busca sobre regiões promissoras. O método foi avaliado em 56 instâncias teste, produzindo alguns resultados melhores que os relatados na literatura. O resultado final obtido ficou a menos de 3% dos melhores encontrados na literatura, ressaltando a competitividade da metodologia na resolução do PRVJT.

REFERÊNCIAS

- Berger, J., Barkaoui, M., & Bräysy, O., 2001. A parallel hybrid genetic algorithm for the vehicle routing problem with time windows. Working paper, Defense Research Establishment Valcartier, Canada.
- Blum, C. & Roli, A., 2003. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys*, vol. 35, n. 3, pp. 268–308.
- Chen, C.-H. & Ting, C.-J., 2005. Algorithms for vehicle routing and scheduling problems with time windows constraints. *Journal of the Eastern Asia Society for Transportation Studies*, vol. 6, pp. 2822–2836.
- Chen, T., Lee, L., & Ke, O., 2001. Hybrid genetic algorithm in solving vehicle routing problem with window constraints. *Asia-Pacific Journal of Operational Research*, vol. 18, pp. 121–130.
- Costa, T. A., 2005. *Metaheurísticas Híbridas GRASP-Busca Tabu Aplicadas ao Problema de Roteamento de Veículos com Janelas de Tempo*. Dissertação de mestrado, Centro Federal de Educação Tecnológica de Minas Gerais, Belo Horizonte, MG.
- Dorigo, M. & Blum, C., 2005. Ant colony optimization theory: A survey. *Theoretical Computer Science*, vol. 344, pp. 243–278.
- Dorigo, M. & Gambardella, L. M., 1997a. Ant colonies for the traveling salesman problem. *BioSystems*, vol. 43, pp. 73–81.

- Dorigo, M. & Gambardella, L. M., 1997b. Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, vol. 1, pp. 53–66.
- Dorigo, M., Maniezzo, V., & Coloni, A., 1991. The ant system: An autocatalytic optimization process. Technical Report 91-016 Revised. Dipartimento di Elettronica, Politecnico di Milano, Italy.
- Ellabib, I., Basir, O. A., & Calamai, P., 2003. A new ant colony system updating strategy for vehicle routing problem with time windows. In *MIC2003: The Fifth Metaheuristics International Conference*, pp. 18–1 – 18–6.
- Flood, M. M., 1956. The traveling salesman problem. *Operations Research*, vol. 4, pp. 61–75.
- Fraga, M. C. P., Souza, S. R., Costa, T. A., & Souza, M. J. F., 2005. Algoritmos baseados em grasp e busca tabu para resolução do problema de roteamento de veículos com janela de tempo. In *VIII Encontro de Modelagem Computacional*. ISBN 85-904971-2-7.
- Fraga, M. C. P., 2006. *Uma metodologia híbrida Colônia de Formigas - Busca Tabu - Reconexão por Caminhos para resolução do Problema de Roteamento de Veículos com Janelas de Tempo*. Dissertação de mestrado, Centro Federal de Educação Tecnológica de Minas Gerais, Belo Horizonte, MG.
- Gambardella, L. M. & Dorigo, M., 1996. Solving symmetric and asymmetric tsps by ant colonies. In *ICEC96 Proceedings of the IEEE Conference on Evolutionary Computation*, pp. 622–627. IEEE Press.
- Gambardella, L. M., Taillard, É. D., & Agazzi, G., 1999. Macs-vrptw: A multiple ant colony system for vehicle routing problems with time windows. In D. Corne, M. D. & Glover, F., eds, *New Ideas in Optimization*, pp. 63–76. McGraw-Hill.
- Glover, F. & Laguna, M., 1997. *Tabu Search*. Kluwer Academic Publishers, Boston.
- Glover, F., 1986. Future paths for integer programming and links to artificial intelligence. vol. 5, pp. 553–549.
- Glover, F., 1996. Tabu search and adaptive memory programming - advances, applications and challenges. In Barr, R., Helgason, R., & Kennington, J., eds, *Interfaces in Computer Sciences and Operations Research*, pp. 1–75. Kluwer Academic Publishers.
- Hansen, P., 1986. The steepest ascent mildest descent heuristic for combinatorial programming. In *Proceedings of Congress on Numerical Methods in Combinatorial Optimization*, Capri, Itália.
- Ho, S. C. & Gendreau, M., 2006. Path relinking for the vehicle routing problem. *Journal of Heuristics*, vol. 12, pp. 55–72.
- Potvin, J., Kervahut, T., Garcia, B., & Rousseau, J., 1996. The vehicle routing problem with time windows part i: Tabu search. *INFORMS Journal on Computing*, vol. 8, pp. 157–164.
- Solomon, M. M., 1987. Algorithms for vehicle routing and scheduling problems with time windows constraints. *European Journal of Operational Research*, vol. 35, pp. 254–266.
- Taillard, É. D., Badeau, P., Gendreau, M., Guertin, F., & Potvin, J., 1997. A tabu search heuristic for the vehicle routing problem with soft time windows. *Transportation Science*, vol. 31, pp. 170–186.

Abstract. In this paper, it is presented an hybrid methodology called Ant-TPR to solve the Vehicle Routing Problem with Time Windows (VRPTW). The VRPTW is a variation from the Capacitated Vehicle Routing Problem (CVRP) in which, beyond the inherent restrictions to the CVRP, the attendance of each consumer must be initiated in a time interval called time window. The solution to this problem consists of, first, to find the minimum number of vehicles able to attend all the consumers and, second, to determine a set of routes that minimizes the covered distance of the vehicles. The Ant-TPR methodology combines two metaheuristics, Ant

Colony Optimization (ACO) and Tabu Search, with the Path Relinking intensification technique. The ACO metaheuristic feeds itself, in part, of its own results, leading to convergence and precision of the generated results. Tabu Search strongly depends of its initial solution so the association with ACO will improve results from both. Periodically, the Path Relinking technique is applied to insert good attributes from the best solutions found in the current solutions. The developed methodology was benchmarked in Solomon's VRPTW instances, generating some isolated better results than those found in literature and remaining less than 3% above the best existing methodology.

Keywords: *Ant Colony, Tabu Search, Path Relinking, Vehicle Routing, Metaheuristic.*