

3 Revisão da literatura

Pensar é mais interessante que saber e menos interessante que observar.

JOHANN WOLFGANG GOETHE

O objetivo deste capítulo é apresentar as referências mais importantes relacionadas aos três principais tópicos cobertos neste trabalho que são o planejamento operacional de pátios ferroviários, o problema de coleta e entrega e a otimização com colônia de formigas.

Neste trabalho, o PPOLM é modelado como um problema de coleta e entrega com vários veículos, restrição de janelas de tempo, capacidade e precedência, composto com um problema de encontrar o caminho mais curto entre dois vértices de um grafo onde o peso é dependente de espaço e tempo. O problema de coleta e entrega é resolvido com uma metaheurística de otimização com colônia de formigas envolvendo duas colônias e o caminho mínimo é encontrado com o tradicional algoritmo de Dijkstra.

3.1. Planejamento operacional de pátios ferroviários

A principal motivação deste trabalho é a redução dos custos operacionais de pátios ferroviários. Neste contexto, espaço, tempo e custo são variáveis interdependentes e constituem o foco principal da análise. Por exemplo, espaços mais curtos percorridos pelas locomotivas de manobras implicam em menor tempo para percorrê-lo e conseqüentemente menor consumo de combustível e menor tempo de alocação dos recursos humanos envolvidos (e.g. tripulação, técnicos de operação ferroviária), reduzindo assim o custo operacional do pátio.

Daganzo et al. (1983) destaca os dois componentes mais importantes do ciclo médio de vagões: O tempo em que o vagão fica na ferrovia, que equivale, em média, a 14% do ciclo médio de vagão e o tempo em que o vagão fica em pátios, que corresponde a 62% do ciclo médio de vagão. Trabalhos mais recentes como Dyke (2006) e Dirnberger & Barkan (2007) mostram que esta proporção de distribuição dos componentes do ciclo de vagão não mudou muito durante as últimas décadas. Isto significa que investimentos na melhoria operacional dos pátios, visando reduzir o tempo de permanência dos vagões nos mesmos implicam em um grande potencial de redução do ciclo de vagão e, conseqüentemente, em uma perspectiva de melhoria considerável na eficiência do sistema de transporte ferroviário como um todo. Com efeito, segundo Bektas et al. (2007), uma das iniciativas mais importantes para aumentar a produtividade e a competitividade das ferrovias é a gestão eficiente dos pátios ferroviários, fazendo com que os vagões sejam processados o mais rápido possível, de modo a propiciar uma utilização eficiente dos ativos e atender os horários de conexão.

Embora o investimento em melhorias operacionais de pátios sugira ganhos potenciais, existem poucos trabalhos publicados nesta área. De acordo com Lübbecke (2001), o trabalho pioneiro Charnes & Miller (1956) era o único publicado sobre programação de trabalho para locomotivas de manobras em pátios ferroviários em um intervalo de 45 anos até aquela data. Reis Jr. (2003), estuda algoritmos de *branch and bound* paralelos para planejamento de manobras em pátios e, finalmente, encerrando a lista das referências conhecidas sobre o assunto, aparece Sabino (2004) que é a pesquisa precursora deste trabalho.

Reis Jr. (2003) é um trabalho de interesse peculiar para esta pesquisa já que o mesmo usa como referência o mesmo ambiente industrial. Nesse trabalho, não são apresentadas as características detalhadas do ambiente estudado, nem o método utilizado para coleta dos dados reais, nem os dados de entrada usados nos testes computacionais e nem as soluções encontradas para cada instância testada. Isto torna impossível usá-lo como referência para uma comparação dos problemas e dos métodos adotados para solução. No entanto, aparentemente, esse trabalho estuda um problema similar ao PPOLM e, sem um tratamento formal do mesmo, o modela como um problema de escalonamento de tarefas, caracterizando-o como uma busca pelo melhor caminho em um grafo e um problema de programação de

manobras. O objetivo principal foi apresentar uma solução paralela para tais problemas e a conclusão foi que, embora a técnica escolhida seja eficiente, ela é sensível a problemas internos como gargalos de comunicação e anomalias. Do ponto de vista de aplicação prática, não foi apresentado o resultado de testes comparativos usando como referência dados reais ou mesmo a simulação destes. A conclusão naquela época foi que, embora o algoritmo proposto tenha sido considerado aplicável em qualquer pátio, o mesmo não considerava alguns aspectos importantes tais como linhas férreas interditadas, rotas conflitantes e lotes de vagão de diferentes tamanhos, já que estes itens foram relaxados para se permitir um primeiro passo na solução dos problemas citados.

Kraft (1998) e Daganzo et al. (1983) tratam de outros problemas relacionados ao planejamento operacional de pátios sem, no entanto, referenciar o PPOLM especificamente. Kraft (1998) apresenta um método de controle operacional e programação de carregamento e Daganzo et al. (1983) propõe um procedimento para melhorar a eficiência das operações de classificações em pátios ferroviários. Estes dois trabalhos foram as principais referências durante os estudos que resultaram na decomposição do planejamento operacional de pátios ferroviários em três etapas e forneceram os fundamentos para a modelagem do caminho das locomotivas de manobras.

A maioria (e, de fato, a quase totalidade) dos estudos sobre planejamento operacional de ferrovias aborda a movimentação de cargas pela ferrovia, a programação de trabalho com as locomotivas de viagem, a atribuição de blocos a trens e a definição da programação de trens. Trabalhos como Ahuja & Jha (2004) e Ahuja et al. (2004), Crainic et al. (1984), Crainic & Roy (1988), Keaton (1989), Marin & Salmeron (1996a, 1996b), Martinelli & Hualiang (1996), Crainic & Laporte (1997), Gualda & Murgel (2000) e Crainic (2002), embora não tratem especificamente das operações de pátio e sim do planejamento operacional e tático de todo o sistema ferroviário, forneceram uma boa base sobre a terminologia e a modelagem contribuindo para que a abordagem contida neste trabalho considerasse o planejamento de pátios no contexto do sistema ferroviário como todo.

Ahuja et al. (2002), Kuo & Nicholls (2007) e Vaidyanathan et al. (2008), embora tratem da programação de trabalho de locomotivas visando a redução do

custo operacional, consideram locomotivas que viajam pela ferrovia para transportar carga e não locomotivas que executam manobras em pátios.

Também na área de simulação não foi possível identificar trabalhos que tratassem de planos de trabalho de locomotivas de manobra. A referência que mais se aproxima deste assunto é Franzese et al. (2003) que apresenta um simulador da Estrada de Ferro Vitória a Minas, da qual um dos pátios terminais é exatamente o pátio de referência para a aplicação prática do algoritmo apresentado nesta tese. Este simulador, que utiliza uma interface gráfica com o usuário desenvolvida em Microsoft Excel®, se baseia no *software* ARENA®, da *Rockwell Automation* para desenvolver um estudo cujo objetivo principal foi verificar se trens com 320 vagões aumentariam a capacidade de transporte do sistema ferroviário. Mais uma vez, o foco deste trabalho está na ferrovia e não no pátio.

Diante do número limitado de referências bibliográficas sobre o problema específico do qual trata este trabalho, muitas informações importantes foram obtidas através de entrevistas com profissionais da área.

3.2. Problema de coleta e entrega

De acordo com o modelo proposto neste trabalho e considerando a modelagem apresentada em Parragh et al. (2006), o problema que mais se aproxima do PPOLM é o problema de coleta e entrega clássico. Mais especificamente, pode-se dizer que o PPOLM é um *problema de coleta e entrega com janelas de tempo*, como em Dumas et al. (1991), devido à especificação de um intervalo de tempo para coleta e para entrega dos vagões, *com múltiplos veículos*, como em Zhou et al. (2006), porque a solução pode utilizar mais de uma locomotiva de manobra, *com frota heterogênea*, como em Lee et al. (2007), pois as locomotivas de manobra podem ter capacidades de tração diferente, *com carga única*, ou *full truck load*, como em Jin & Muriel (2005), já que uma locomotiva não pode coletar vagões adicionais sem que antes tenha entregue os vagões que esteja transportando, *com precedências*, como em Sigurd et al. (2000), pois há uma ordem pré-definida para realizar certas manobras, e, finalmente, *com múltiplos objetivos*, como em Lu & Dessouky (2004), já que deseja-se reduzir tanto custo fixo, relacionado à quantidade de locomotivas necessárias para a solução, quanto o custo variável, relacionado ao deslocamento das locomotivas

para realização das manobras planejadas. Em suma o PPOLM pode ser considerado um problema de coleta e entrega com janelas de tempo, múltiplos veículos, frota heterogênea, carga única, precedências e múltiplos objetivos.

O problema de coleta com janelas de tempo tem inúmeras aplicações práticas referenciadas na literatura. Toth & Vigo (2002) citam alguns exemplos clássicos como o transporte de deficientes físicos e idosos, transporte aéreo e naval de cargas e tropas militares, serviços urbanos de coleta e entrega de encomendas e planejamento de rotas de ônibus escolares. Motivados pela diversidade de trabalhos disponível na literatura, alguns pesquisadores chegaram a desenvolver levantamentos bibliográficos sobre o problema de coleta e entrega, dentre os quais os seguintes foram referências importantes para este trabalho:

- Savelsbergh & Sol (1995), que desenvolve uma análise sistemática cobrindo todas as variações conhecidas até então do problema de coleta e entrega. Para cada versão é apresentada a formulação e a modelagem, bem como os métodos conhecidos para a solução;
- Mitrovic-Minic (1998), que apresenta um levantamento sobre as variações e métodos disponíveis para solução com foco no problema de coleta e entrega com janelas de tempo, incluindo a formulação matemática utilizada para o desenho de soluções exatas;
- Parragh et al. (2006), que é o levantamento mais recente de modelos para tratamento do problema de coleta e entrega e, em especial, faz referência ao caso onde há itens transportados entre locais de coleta e entrega (e não entre depósitos) e suas variações, desenvolvendo uma classificação com respectivas formulações matemáticas e métodos de solução conhecidos para cada caso.

Algoritmos para solução de problemas de coleta e entrega com janelas de tempo e outras restrições são geralmente divididos em três classes:

- Os algoritmos exatos, aplicados a modelos baseados em programação inteira, que resolvem o problema usando métodos com geração de colunas, *branch-and-bound* ou *branch-and-cut* como em Dumas et al. (1991), Baldacci et al. (2004), Cordeau (2006), Dell'Amico et al. (2006) e Ropke et al. (2007);

- As heurísticas que constroem, por decomposição ou inserção, um conjunto de rotas viáveis a partir do zero como em Lu & Dessouky (2006) e Dell'Amico et al. (2007) ou procuram melhorar uma solução inicial através da aplicação de um algoritmo de busca local como em Nanry & Barnes (2000) e Cordeau & Laporte (2003).
- As metaheurísticas, tais como *Ant Colony Optimization* (ACO), *Evolutionary Computation* (EC), *Simulated Annealing* (SA) e *Tabu Search* (TS), que segundo Dorigo & Stützgen (2004), são métodos heurísticos de propósito geral, desenvolvidos para guiar uma heurística subordinada (e.g. uma heurística de construção ou de busca local) em direção a regiões do espaço de busca que contenham soluções de alta qualidade. Este é o caso de Li & Lim (2003), Barcus (2004) e Tarantilis et al. (2005). Blum & Roli (2003) apresentam e comparam as metaheurísticas mais importantes disponíveis na literatura até aquela época.

O problema de coleta e entrega com janela de tempo é NP-difícil (Mitrovic-Minic, 1998). Com efeito, algoritmos exatos têm requerido um consumo considerável de recursos computacionais para encontrar a solução ótima de problemas mais simples e com instâncias de tamanho modesto. Este é o caso de Toth & Vigo (1997), que trata de um problema de coleta e entrega sem janelas de tempo com uma técnica baseada em limites inferiores obtidos pela aplicação de Relaxação Lagrangeana, combinada com o uso de planos de cortes, algoritmo *branch-and-bound*, procedimentos de redução e critérios de dominância, para resolver instâncias com 100 pontos de coleta e entrega, consumindo 1h40min de tempo de CPU de um processador Pentium® de 60MHz. Outro exemplo mais recente é Lu & Dessouky (2004) que, no melhor caso, consegue resolver um problema de coleta e entrega com múltiplos veículos e múltiplos objetivos, mas sem janelas de tempo com, no máximo, 5 veículos e 25 pontos de coleta, requerendo, para tanto, três horas de tempo de CPU em um servidor *Sunfire® 4800*. Um caso prático do PPOLM tem cerca de 120 pontos de coleta e entrega, e se deseja resolvê-lo em alguns minutos usando um computador de baixo custo. Diante deste cenário, fica claro o motivo pelo qual é comum o uso de heurísticas ou metaheurísticas para buscar soluções de problemas da vida real, já que deste

modo é possível encontrar boas soluções com tempo de CPU razoável e, portanto, tempo de execução praticável.

Mesmo diante da diversidade de modelos, métodos e resultados reportados na literatura, a avaliação do desempenho relativo de algoritmos para resolver problemas de coleta e entrega com janelas de tempo é praticamente impossível, já que até o momento não há conjuntos conhecidos de instâncias de referência (i.e. *benchmarks*) para o caso genérico deste problema. Toth & Vigo (2002) explicam que isto ocorre principalmente devido à diversidade de variações dos problemas estudados, já que a maioria dos trabalhos se baseia em aplicações práticas, as quais induzem modelagens específicas para a representação das restrições e da função objetivo. Assim, os métodos são testados com dados simulados da situação real em estudo e os resultados são comparados com os obtidos com as soluções manuais em uso.

3.3. Otimização com Colônia de Formiga

Otimização com Colônia de Formiga ou *Ant Colony Optimization (ACO)* como são mais conhecidos na literatura, é uma classe de algoritmos que geram soluções candidatas para um problema de otimização através de um mecanismo construtivo onde a escolha do componente da solução a ser adicionado em cada passo se baseia em um balanceamento probabilístico que considera trilhas de feromônio artificial e informações heurísticas sobre o problema em questão (Dorigo & Stützle, 2004).

Dentre os diversos algoritmos ACO disponíveis atualmente na literatura, o primeiro a ser proposto foi o algoritmo *Ant System (AS)*, apresentado nas versões originais denominadas *ant-density* (Dorigo et al., 1991), *ant-quantity* (Colomi et al., 1992) e *ant-cycle* (Dorigo, 1992). Atualmente ACO se caracteriza como uma ferramenta de otimização poderosa e versátil, com um número crescente de publicações e aplicações em diversas áreas da pesquisa operacional, gestão e tecnologia (Gutjahr, 2007).

Problemas de otimização combinatória tratados com ACO são normalmente codificados através de um grafo de construção completo $G(V,A)$, onde os nós de V são componentes da solução e os arcos de A são conexões entre os

componentes. Construir uma solução significa encontrar um caminho viável em G . Por exemplo, no problema do caixeiro viajante, os nós correspondem aos clientes, os arcos correspondem às ruas que conectam os clientes e uma solução viável é um caminho hamiltoniano no grafo (Bianchi, 2006).

Os parágrafos seguintes apresentam uma visão geral do algoritmo *Ant System* (AS) baseado na descrição apresentada em Dorigo et al. (2006).

Uma *formiga* é um agente computacional simples, que constrói iterativamente uma solução para o problema de otimização em questão. Usualmente, nos algoritmos ACO, cada formiga artificial parte de uma solução vazia e adiciona componentes à sua solução parcial até conseguir uma solução candidata completa. Soluções parciais para o problema são chamadas *estados*. Assim, uma formiga *move* de um estado i para um estado j correspondente a uma solução parcial mais completa para o problema. Em cada *passo* s , uma formiga a determina o conjunto de expansões viáveis para o seu estado atual e escolhe uma destas expansões, movendo-se para um novo estado, baseada na distribuição de probabilidade que se segue.

A probabilidade $p_{i,j}^a$ de uma formiga se mover de um estado i para um estado j depende da combinação de dois valores:

1. A atratividade η do movimento, calculada antes do processo de construção da solução e, portanto, considerada uma indicação prévia do grau de atração do movimento;
2. O nível de feromônio τ do movimento, que indica o quão eficiente foi fazer aquele movimento no passado. Este valor é considerado uma indicação posterior da atratividade do referido movimento, já que é calculado durante o processo de construção da solução.

A Figura 9 ilustra as diversas possibilidades de passo de uma formiga partir do estado i . Se o estado j ainda não foi visitado, ou seja, se ainda não pertence ao caminho da formiga, a probabilidade de escolha deste estado é proporcional ao nível de feromônio τ e à atratividade η associado ao passo (i,j) .

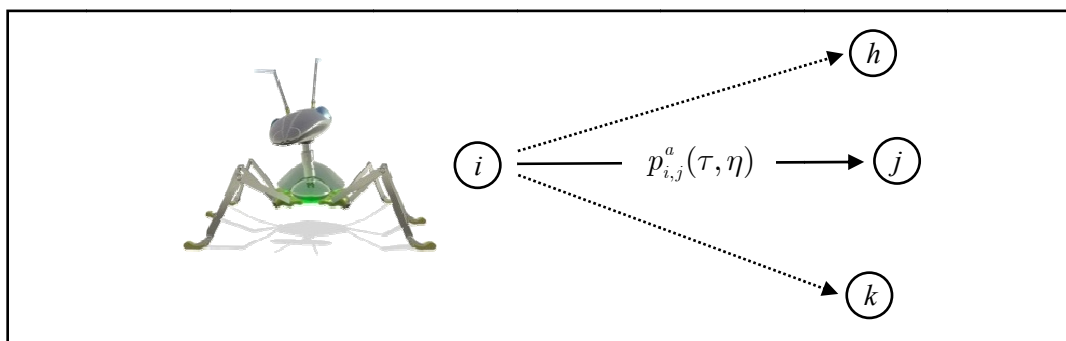


Figura 9: Movimentos possíveis de uma formiga a partir do estado i

A fórmula que define a distribuição de probabilidade em cada passo utiliza o conjunto $inviáveis_a$ que define as soluções inviáveis para a formiga a , de acordo com condições que dependem da definição do problema. As probabilidades são computadas como segue: $p_{i,j}^a$ é igual a zero para todos os passos inviáveis (isto é, os que estão contidos no conjunto $inviáveis_a$), caso contrário, a probabilidade é calculada pela seguinte fórmula:

$$p_{ij}^a = \frac{\tau_{ij}^\alpha \cdot \eta_{ij}^\beta}{\sum_{(iv) \notin inviáveis_a} \tau_{iv}^\alpha \cdot \eta_{iv}^\beta} \quad (17)$$

onde τ_{ij} é a quantidade de feromônio associado ao movimento de i para j , η_{ij} é a atratividade associada a este movimento e os parâmetros α e β definem a importância do feromônio em relação à atratividade. A atratividade normalmente é definida como sendo o inverso do custo associado ao movimento de i para j . Para o problema clássico do caixeiro viajante, por exemplo, onde o custo é a distância entre dois nós, a atratividade é normalmente o inverso da distância.

A quantidade de feromônio deixada no caminho é atualizada em cada iteração. Inicialmente, todos os caminhos têm sua quantidade de feromônio reduzida, simulado assim o fenômeno da evaporação do feromônio, tal como ocorre na natureza. Em seguida, os caminhos que fazem parte da solução encontrada por alguma formiga tem sua quantidade de feromônio incrementada. Os demais caminhos não têm acréscimo de sua quantidade de feromônio. Em cada iteração t do algoritmo, depois que todas as formigas completaram sua solução a quantidade de feromônio é atualizada de acordo com a seguinte fórmula:

$$\tau_{ij}(t) = \rho \cdot \tau_{ij}(t-1) + \sum_{k=1}^m \Delta\tau_{ij}^k \quad (18)$$

onde ρ é um valor definido pelo usuário que representa a persistência do feromônio ao longo das iterações, m é o número de formigas e $\Delta\tau_{ij}^k$ representa a quantidade de feromônio deixada pela formiga k na aresta (i,j) .

A contribuição de cada formiga para o depósito de feromônio em cada passo de seu caminho é proporcional à qualidade da solução obtida por aquela formiga: Quanto melhor for a qualidade da solução, maior é a quantidade de feromônio depositado. Se o caminho da formiga k não inclui a aresta (i,j) então o valor de $\Delta\tau_{ij}^k$ é zero.

```

ACO
Passo 1: Inicialização
  Definir valor inicial do feromônio;
  Definir solução com o pior valor possível;
  Inicializar caminho com vazio;

Passo 2: Construção
  Para cada formiga  $k$  repetir {
    Calcular os valores de  $\eta$ ;
    Escolher o estado para movimentação usando a
    probabilidade dada pela fórmula (17);
    Adicionar o movimento escolhido ao conjunto inviáveisk;
  } Até que a formiga  $k$  obtenha uma solução completa
    ou desista;

    [melhorar a solução com busca local] Opcional

  Se (valor da nova solução é melhor que solução) {
    Atualizar solução com o novo valor;
    Atualizar caminho associado ao novo valor;
  }

Passo 3: Atualização do feromônio
  Para cada movimento do caminho da solução {
    Calcular  $\Delta\tau_{ij}^k$ 
    Atualizar a matriz de feromônios usando a fórmula (18)
  }

Passo 4: Condição de término
  Se (condição de término não satisfeita) vá para o passo2;

```

Figura 10: Algoritmo ACO genérico, baseado em Maniezzo & Carbonaro (1999)

A Figura 10 apresenta um procedimento ACO genérico em pseudo-código. Primeiramente a matriz de feromônios é inicializada. Depois cada formiga tem a

chance de construir a sua própria solução completa ou desistir. A desistência ocorre caso a formiga chegue a um estado sem possibilidade de movimento e não estiver ainda com uma solução completa. Depois disso, o rastro de feromônio ao longo dos caminhos percorridos é atualizado e finalmente a condição de término é testada (e.g. depois de decorrido um determinado tempo de execução) para definir se uma nova iteração será necessária. Um exemplo de condição de término seria Em alguns casos, usa-se um algoritmo de busca local para aperfeiçoar cada solução encontrada.

Embora a convergência do algoritmo ACO tenha sido provada (Dorigo & Blum, 2005), não há estudos disponíveis na literatura sobre a velocidade de convergência. Sendo assim, não há outra forma de medir o desempenho do algoritmo a não ser a execução de extensos testes experimentais (Dorigo & Stützle, 2004).

Dorigo et al. (2006), apresenta a família de algoritmos ACO como um conjunto de métodos para resolver vários problemas de otimização combinatória, quer sejam estáticos ou dinâmicos. A Tabela 1 mostra algumas aplicações bem sucedidas da metaheurística ACO para resolver problemas clássicos de otimização, e os respectivos anos de publicação.

Lista não exaustiva de aplicações do algoritmo ACO agrupados por tipo de problema			
TIPO DE PROBLEMA	NOME DO PROBLEMA	AUTORES	ANO
ROTEAMENTO	CAIXEIRO VIAJANTE	DORIGO ET AL	1991, 1996
		DORIGO E GAMBARDELLA	1997
		STÜTZLE E HOOS	1997, 2000
	ROTEAMENTO DE VEÍCULOS	GAMBARDELLA ET AL	1999
		REIMANN ET AL	2004
ORDENAÇÃO SEQUENCIAL	GAMBARDELLA E DORIGO	2000	
ATRIBUIÇÃO	ATRIBUIÇÃO QUADRÁTICA	STÜTZLE E HOOS	2000
		MANIEZZO	1999
	TABELA DE CURSOS	SOCHA ET AL	2002, 2003
	COLORAÇÃO DE GRAFOS	COSTA E HERTZ	1997
AGENDAMENTO	PROGRAMAÇÃO DE PROJETOS	MERKLE ET AL	2002
	ATRASSO TOTAL PONDERADO	DEN BESTEN ET AL	2000
		MERKLE E MIDDENDORF	2000
PROGRAMAÇÃO DA PRODUÇÃO	BLUM	2005	
SUBCONJUNTO	RECOBRIMENTO DE CONJUNTOS	LESSING ET AL	2004
	ÁRVORES DE CARDINALIDADE i	BLUM E BLESÁ	2005
	MOCHILA MÚLTIPLO	LEGUIZAMÓN E MICHALEWICZ	1999
	CLIQUE MÁXIMO	FENET E SOLNON	2003
OUTROS	SATISFAÇÃO DE RESTRIÇÕES	SOLNON	2000, 2002
	REGRAS DE CLASSIFICAÇÃO	PARPINELLI ET AL	2002
		MARRTENS ET AL	2006
	REDES BAYESIANAS	CAMPOS ET AL	2002
	ENROLAMENTO DE PROTEÍNAS	SHMYGELSKA E HOOS	2005
	DOCKING PROTEÍNA-LIGANTE	KORB ET AL	2006

Tabela 1: Aplicações de ACO, baseada em Dorigo, Birattari e Stützle, 2006

Dorigo & Blum (2005) é uma boa referência teórica sobre a metaheurística ACO. Este trabalho apresenta uma revisão dos resultados sobre convergência do algoritmo, discute as relações entre ACO e outros métodos de aproximação e identifica algumas questões em aberto como motivação para pesquisas futuras.

As referências mais importantes sobre ACO para este trabalho foram as que apresentam casos de solução de problemas de otimização multi-objetivo em sistemas de transporte. Neste contexto se destacam Reimann (2002), que apresenta um algoritmo ACO com duas colônias de formigas para resolver um problema de transporte com caminhões de carga similar ao PPOLM, Sabino (2004) que é a pesquisa precursora deste trabalho e que apresenta um protótipo de algoritmo para resolver o PPOLM e Gambardella et al. (2003), que apresenta duas ferramentas para assistir o planejamento de roteiro em várias fases de um processo de entrega de encomendas.