4
Metodologia de solução do problema

A direção é mais importante que a velocidade.
ROBERTO SCARINGELLA

Este capítulo apresenta uma visão geral da modelagem e método de solução proposto para resolver o PPOLM. A estratégia de solução é baseada no uso da metaheurística ACO.

4.1.
Escolha dos algoritmos

Os seguintes fatores foram os principais motivadores para a escolha da metaheurística ACO para a construção do plano de trabalho das locomotivas de manobra no PPOLM :

- ✓ A facilidade de implementação e manutenção: Além da implementação muito simples, novas restrições e características do problema são adicionadas com mínima revisão de código. Estas características se tornaram evidentes durante a fase de desenho e codificação da aplicação;
- ✓ A oportunidade de explorar o estado da arte em tendências e ferramentas: o último *workshop* bianual sobre ACO e *Swarm Intelligence*, realizado em Bruxelas, na Bélgica em 2006, contou com 50 trabalhos apresentados a partir de uma seleção que aproveitou 42% do total de trabalhos submetidos. Além disso, o prêmio concedido pelo *INFORMS-RASIG Group* ao trabalho em Sabino (2002), o qual é um dos precursores deste, mostrou o interesse por esta nova abordagem, tanto pela comunidade

acadêmica como pelos profissionais da área de planejamento ferroviário;

- ✓ Os bons resultados em pesquisas anteriores: ACO tem se destacado como uma técnica eficiente e flexível para resolver problemas de otimização combinatória do tipo do PPOLM (Reimann, 2002; Gambardella et al., 2003).

O algoritmo de Dijkstra foi a base para a implementação da lógica de determinação da melhor rota dos comboios pelo pátio. Esta escolha foi feita considerando os critérios de flexibilidade, escalabilidade, desempenho e complexidade de implementação deste algoritmo comparado a outros métodos de solução de problemas de determinação de caminho mínimo. A avaliação destes critérios foi feita baseada nos modelos de desempenho apresentados em Foster (2006) para projeto e implementação de programas eficientes.

4.2.
Construção do plano de trabalho das locomotivas de manobras

O algoritmo principal desenvolvido para solução do problema de programação das locomotivas foi denominado YoYo - mnemônico de *Yard Operation Yeld Optimization*. O algoritmo YoYo foi construído tomando como base o algoritmo *competANTS*, apresentado em Reimann (2002), que é uma implementação de otimização com colônia de formigas onde há duas colônias com regras de prioridade diferentes. Uma colônia, chamada de colônia *Empty Move (EM)*, busca soluções com baixo custo variável e para isso procura executar as manobras de modo a minimizar o deslocamento total das locomotivas de manobras. A outra colônia de formigas, chamada de colônia *Waiting Time (WT)* busca soluções com baixo custo fixo, procurando assim reduzir a quantidade de locomotivas de manobras necessárias.

As características do problema é que vão definir qual das duas colônias vai conseguir chegar à melhor solução. Por exemplo, se as janelas de tempo são curtas, então a colônia WT não vai ter muitas chances de atingir sua meta que é a maximizar a utilização das locomotivas de manobras através da redução do tempo de espera no ponto de coleta no caso da locomotiva chegar antes do horário mínimo permitido. Por outro lado, se as janelas de tempo são longas a colônia WT tende a obter melhores resultados. Considerando que tanto a redução do tempo de espera quanto a redução dos movimentos vazios minimiza o custo total, a idéia é explorar o ponto forte de ambas as colônias. Em primeiro lugar, o número de formigas de cada colônia não é constante. Depois de cada iteração, parte das formigas da colônia cuja média das soluções encontradas em cada iteração é menor migram para a outra colônia, dando assim maior capacidade computacional à outra colônia na próxima iteração. Além disso, algumas formigas utilizam não só o feromônio de suas próprias colônias, mas também o feromônio da outra colônia para influenciar suas decisões sobre como prosseguir na construção de seus caminhos. As Formigas que utilizam somente o feromônio de suas colônias são chamadas formigas *nativas* e as formigas que consideram também o feromônio da outra colônia são chamadas formigas *espiãs*. O balanceamento entre o número de formigas nativas e espiãs de uma colônia é definido em função da melhor solução encontrada por cada colônia na iteração anterior.

Formigas das duas colônias constroem seus caminhos da mesma forma: Iniciando no tempo `t=0`, uma locomotiva é escolhida e manobras são atribuídas sequencialmente a esta locomotiva. A cada nova manobra atribuída, o tempo é atualizado de acordo com o tempo necessário para realização da manobra. Novas manobras são atribuídas até o fim do horizonte de planejamento ou até que, em função das restrições do problema, não haja mais nenhuma atribuição possível para esta locomotiva. Neste ponto, outra locomotiva de manobras é alocada para compor a solução, o tempo volta ao valor zero e o processo de atribuição de manobras a locomotivas continua. Este procedimento é repetido até que todas as manobras estejam atribuídas ou até que haja mais nenhuma locomotiva que possa ser colocada em uso. Para decidir qual manobra ou qual locomotiva será adicionada ao seu caminho as formigas utilizam a atratividade η e a quantidade de feromônio τ armazenada anteriormente por outras formigas ao longo do caminho.

Em suma, uma formiga constrói o seu caminho partindo de uma solução inicial contendo apenas uma locomotiva e acrescentando elementos, que podem ser manobras ou locomotivas, visando formar soluções parciais mais completas até chegar a uma solução do PPOLM, ou seja, um plano de trabalho que envolva todas as manobras a serem executadas no horizonte de planejamento dado.

O diagrama da Figura 11 mostra uma solução de um PPOLM. Os quadrados representam locomotivas e os círculos representam manobras. Os segmentos horizontais que unem cada elemento ao seu sucessor representam passos, ou seja, mudanças de estado que correspondem à adição de itens a uma solução parcial, formando assim o caminho da formiga.

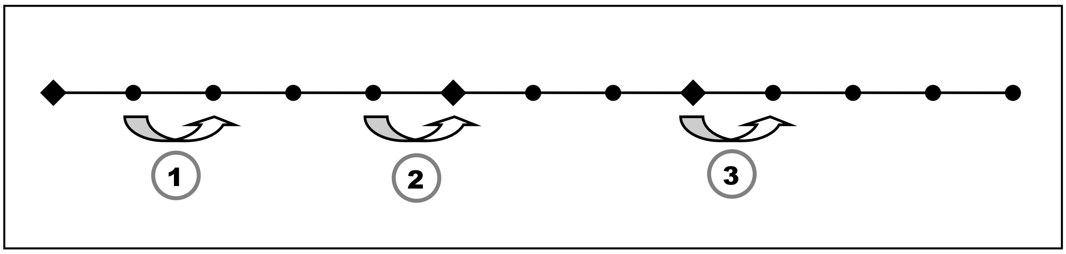

Figura 11: Caminho de uma formiga com três passos destacados

Há três passos destacados na Figura 11, assinalados com os números 1, 2 e 3, indicando os três tipos possíveis de mudança de estado, conforme abaixo:

1. Nova Manobra (NOMA): Se a última operação feita foi a adição de uma manobra no plano de trabalho de uma locomotiva e ainda é possível adicionar mais manobras ao plano de trabalho desta mesma locomotiva, o item adicionado será então mais uma manobra para mesma locomotiva;
2. Nova Locomotiva (NOLO): Se a última operação feita foi a adição de uma manobra ao plano de trabalho de uma locomotiva, mas não é possível adicionar mais manobras ao plano de trabalho desta mesma locomotiva, o item adicionado será uma nova locomotiva;
3. Primeira manobra (PRIMA): Se a última operação foi a adição de uma locomotiva, o item adicionado será a primeira manobra desta locomotiva.

Como em qualquer algoritmo de otimização baseado em colônias de formigas, a mudança de estado, ou seja, o acréscimo de um novo item no caminho da formiga depende da atratividade η e da quantidade de feromônio τ existente nas arestas do grafo *G(V,A)* que ligam o nó *i* atual a cada um dos nós candidatos a ser o próximo nó daquele caminho. Assim, as formigas selecionam a próxima locomotiva, ou a próxima manobra, baseadas na atratividade, que guia o processo construtivo para a minimização da função objetivo, e na quantidade de feromônio, o qual fornece informações históricas sobre a qualidade das soluções obtidas nas iterações anteriores.

O algoritmo CompetAnts é apresentada na Figura 12. Primeiramente são definidas as soluções iniciais para as duas colônias. Depois disso, é determinado o número de formigas nativas e espiãs de cada colônia. Em seguida, é feita uma chamada ao algoritmo *ACO* (apresentado na Figura 10) para cada tipo de colônia. Finalmente é apresentado o resultado do procedimento.

```
Algoritmo CompetAnts
   Leitura dos parâmetros;
   Initialização do sistema;
   Execução da primeira iteração
     Chamada ACO para a colônia EM;
     Chamada ACO para a colônia WT;
   Para i de 2 até num_max_iterações
     Adaptação do número de formigas das colônias baseado no
     custo médio;
     Determinação do número de espiões baseado na melhor
     solução;
     Execução da otimização
        Chamada ACO para a colônia EM;
        Chamada ACO para a colônia WT;
   Fim Para
   Gravação do resultado;
```

Figura 12: Algoritmo CompetAnts, conforme Reimann (2002)

4.3. Cálculo da atratividade

A atratividade η é uma informação básica para a definição da fórmula que guia o processo incremental de construção do caminho das formigas. Cada colônia utiliza uma fórmula de atratividade diferente para as mudanças de estado.

A fórmula seguinte é usada para o cálculo da atratividade para a colônia WT. Ela procura expressar a produtividade das locomotivas de manobras, considerando improdutivo o tempo de espera, bem como o tempo de deslocamento escoteira da locomotiva de manobras:

$$\eta_{ij}^{WT}(t) = \begin{cases} e^{-4K\cdot[DTWE_j + 2\cdot UT_j(i,t)]} & \text{se } (i,j) \text{ é um passo viável do tipo NOMA ou PRIMA,} \\ 1 & \text{se } (i,j) \text{ é um passo viável do tipo NOLO,} \\ 0 & \text{nos demais casos} \end{cases} \tag{19}$$

onde $DTWE_j$ é o instante final da janela de entrega e $UT_j(i,t)$ é o tempo improdutivo estimado com a adição da manobra *j* ao caminho da formiga.

Caso o próximo elemento a ser adicionado no caminho seja uma manobra, o valor de DTWEj é exatamente o valor fornecido como dado de entrada associado àquela manobra e o valor de $UT_j(i,t)$ é dado pela seguinte fórmula:

$$UT_j(i,t) = WT_j(t) + LT(i,j) \tag{20}$$

onde $WT_j(t)$ é a diferença entre o horário do início da janela de coleta e o horário estimado de chegada da locomotiva de manobras à linha de coleta e $LT(i,j)$ é o tempo estimado de deslocamento da locomotiva de manobras escoteira até a linha de coleta dos vagões. Vale notar que o valor de $WT_j(t)$ só é positivo se a locomotiva de manobras chegar à linha de coleta antes do início da janela de coleta. Neste caso, $WT_j(t)$ representa o tempo improdutivo que a locomotiva de manobras tem que esperar até o horário mínimo permitido de início da manobra.

Os valores relacionados às janelas de tempo $DTWE_j$ e $WT_j(t)$ privilegiam o encadeamento de manobras que estejam próximas umas das outras em relação ao tempo. O valor de $LT(i,j)$ faz com que a atratividade assuma valores menores quanto maior é o tempo improdutivo de deslocamento das locomotivas na condição escoteiras.

Os fatores 2 e 4 são exatamente os mesmos usados no algoritmo CompetAnts e foram definidos em um estudo preliminar para ajuste de parâmetros. A constante *K* foi introduzida durante os testes computacionais com valores reais coletados na rotina de um pátio ferroviário e deve ser escolhido convenientemente para evitar que o valor de η_{ij}^{WT} assuma um valor muito próximo de zero.

O cálculo da atratividade para a colônia EM utiliza a seguinte fórmula:

$$\eta_{ij}^{EM} = \begin{cases} e^{-16\Upsilon(r_i^-, r_j^+)} & \text{se } (i,j) \text{ é um passo viável do tipo NOMA,} \\ e^{-16\Upsilon(e^+, r_j^+)} & \text{se } (i,j) \text{ é um passo viável do tipo PRIMA,} \\ 1 & \text{se } (i,j) \text{ é um passo viável do tipo NOLO,} \\ 0 & \text{nos demais casos} \end{cases} \tag{21}$$

Nesta fórmula, seguindo o mesmo princípio do algoritmo CompetAnts, o valor da atratividade é influenciado somente pela distância percorrida pela locomotiva de manobra no modo escoteira até o ponto de coleta dos vagões da manobra j, de modo que as formigas tendem a construir caminhos com menor distância total percorrida. O fator *-16* é exatamente o mesmo usado no algoritmo CompetAnts, a função Υ é a mesma da fórmula (9), da página 36 e as linhas r_i^- e r_j^+ são as linhas de entrega da manobra anterior e a linha de coleta da próxima manobra respectivamente, sendo, naturalmente, a linha r_i^- substituída pela linha e^+ no caso da mudança de estado do tipo PRIMA.

Tanto no caso do custo variável estático quanto no caso do custo variável dinâmico, a distância e conseqüentemente o custo de deslocamento entre a linha de coleta e a linha de entrega de uma manobra, fixado o instante de início deste deslocamento, é constante. Assim, somente os deslocamentos na condição escoteira podem ser minimizados para cumprir o objetivo da população EM, e deste modo, a fórmula (21), utilizada no algoritmo CompetAnts, é apropriada para o PPOLM.

É importante observar que, para as colônias WT e EM, a escolha da nova locomotiva a ser alocada nas mudanças de estado do tipo NOLO depende somente da quantidade de feromônio.

4.4.
Regras de decisão

As regras de decisão são as mesmas apresentadas em Reimann (2002). Uma formiga nativa k_n decide qual o próximo item a ser adicionado ao seu caminho baseado na distribuição de probabilidade expressa por:

$$p_{ij}^{k_n}(t) = \begin{cases} \dfrac{[\tau_{ij}]^{\alpha}[\eta_{ij}(t)]^{\beta}}{\sum_{h \in N}[\tau_{ij}]^{\alpha}[\eta_{ij}(t)]^{\beta}} & \text{se } j \in N \\ 0 & \text{caso contrário} \end{cases} \tag{22}$$

onde N é o conjunto de nós que, ao serem adicionados ao caminho da formiga depois do nó i, não violam nenhuma das restrições do problema, α e β são os parâmetros que determinam a influência relativa do feromônio e da atratividade, $\eta_{ij}(t)$ é o valor da atratividade, calculado usando a fórmula (19) ou (21), de acordo com a colônia a que pertence a formiga nativa k_n e τ_{ij} é a quantidade de feromônio depositada no arco *(i, j)*.

A regra de decisão utilizada pelas formigas nativas é a regra clássica do algoritmo *Ant Systems* conforme Dorigo & Stützle (2004). Uma formiga espiã k_f utiliza uma regra de decisão segundo uma distribuição de probabilidade baseada na média ponderada entre o feromônio de sua própria colônia $\tau_{i,j}^{n}$ e o feromônio da outra colônia $\tau_{i,j}^{f}$ dada por:

$$p_{ij}^{k_f}(t) = \begin{cases} \dfrac{[\chi \cdot \tau_{ij}^{n} + (1-\chi) \cdot \tau_{ij}^{f}]^{\alpha}[\eta_{ij}(t)]^{\beta}}{\sum_{h \in N}[\chi \cdot \tau_{ih}^{n} + (1-\chi) \cdot \tau_{ih}^{f}]^{\alpha}[\eta_{ih}(t)]^{\beta}} & \text{se } j \in N \\ 0 & \text{caso contrário.} \end{cases} \tag{23}$$

onde N, α e β são os mesmos parâmetros da fórmula (22) e $0 \leq \chi \leq 1$ representa a importância do feromônio da própria colônia em relação ao feromônio da outra colônia.

Tanto as formigas nativas quanto as espiãs consideram, ao construírem seus caminhos, todas as restrições apresentadas nas fórmulas de (2) a (6), páginas 36 e 33, e mais a restrição de precedência da fórmula (14), página 41.

No caso NOLO, como a atratividade é sempre 1, somente o feromônio influencia na escolha da próxima locomotiva a ser alocada.

4.5. Regras de atualização de feromônio

Foram testadas duas regras de atualização de feromônio. A primeira, chamada de CME, usa o mesmo método proposto em Reimann (2002) e a segunda, chamada RNK, segue a regra intitulada *rank based ant system* proposta em Bullnheimer et al. (1999).

4.5.1. Regra CME

Depois que todas as formigas tiveram a oportunidade de construir a sua solução, os passos que compõem o caminho percorrido pelas formigas que obtiveram as Λ melhores soluções são consideradas para atualização da quantidade de feromônio.

Em primeiro lugar, a quantidade de feromônio destes arcos é reduzida. Esta metáfora da evaporação do feromônio ocorre apenas uma vez para cada arco, mesmo que esta faça parte de mais de um dos Λ caminhos. Em seguida é depositada em cada arco uma quantidade de feromônio proporcional à qualidade da solução que utilizou aquele arco. A regra de evaporação e depósito de feromônio é definida pela fórmula:

$$\tau = \rho \cdot \tau_{ij} + \sum_{\lambda=1}^{\Lambda} \Delta\tau_{ij}^{\lambda}, \forall (i,j) \in J, \text{ e } \; 0 \le \rho \le 1 \qquad (24)$$

onde ρ é o parâmetro que define persistência do feromônio. O primeiro termo desta soma especifica a evaporação e o segundo termo é um somatório que define a quantidade $\Delta\tau_{ij}^{\lambda}$ de feromônio a ser adicionado em cada arco de um dos Λ melhores caminhos.

Apenas as formigas que obtiveram as Λ melhores soluções da iteração depositam feromônio e a evaporação só ocorre nos caminhos percorridos por estas formigas. A fórmula que define a quantidade de feromônio a ser depositada pela formiga classificada na posição λ entre as Λ melhores é a seguinte:

$$\Delta\tau_{ij}^{\lambda} = \begin{cases} 1 - \dfrac{\lambda - 1}{\Lambda} & \text{se } 0 \leq \lambda \leq \Lambda \\ 0 & \text{caso contrário} \end{cases} \tag{25}$$

Nesta fórmula, proposta em Reimann (2002), o valor da função objetivo apurado em iterações anteriores não é usado para o cálculo de $\Delta\tau_{ij}^{\lambda}$ e o número Λ é igual ao número total de formigas dividido por 16.

Este método de atualização da quantidade de feromônio é o mesmo do algoritmo CompetAnts e será chamado neste trabalho de CME.

4.5.2.Regra RNK

Além do método original de atualização de feromônio usado no algoritmo CompetAnts, foi testada uma variação chamada *rank-based ant system*, referenciada neste trabalho como RNK, onde a formiga que obteve a melhor solução até então deposita a maior quantidade de feromônio em cada iteração. De acordo com Dorigo & Stützle (2004), o *rank based ant system* tem desempenho um pouco melhor que o *elitist ant system* e significativamente melhor que o *ant system* original. Esta posição de destaque do método RNK em relação a outros métodos de atualização da quantidade de feromônio na trilha das formigas despertou o interesse em considerar a regra RNK como alternativa à regra CME para a solução do PPOLM.

No RNK, antes de realizar a atualização da quantidade de feromônio, as formigas são classificadas em ordem não decrescente de custo da solução e a quantidade de feromônio que uma formiga deposita depende da posição λ da formiga nesta classificação. Em cada iteração apenas as primeiras *(ω-1)* formigas e a formiga que obteve o melhor resultado até então podem depositar feromônio em seus caminhos. A formiga que obteve o melhor resultado, considerando todas as iterações ocorridas até então e incluindo a iteração atual dá a maior contribuição, depositando feromônio com peso ω. A λ-ésima melhor formiga da iteração atual contribui com peso *max{0,ω-λ}*. Sendo assim, a atualização do feromônio é feita baseada na seguinte fórmula:

$$\tau_{ij} = \rho \cdot \tau_{ij} + \sum_{\lambda=1}^{\omega-1} (\omega - \lambda) \cdot \Delta\tau_{ij}^{\lambda} + \omega \cdot \Delta\tau_{ij}^{bs} \tag{17}$$

onde $\Delta\tau_{ij}^{\lambda} = 1/C^{\lambda}$, $\Delta\tau_{ij}^{bs} = 1/C^{bs}$, C^{bs} é o custo da melhor solução obtida até então e C^{λ} é o custo da solução da λ-ésima melhor formiga na iteração atual. Todos os custos são calculados através da fórmula (16).

Uma diferença importante entre as duas regras de atualização de feromônio propostas neste trabalho é que na regra RNK usa-se o custo C^{bs} para calcular a quantidade de feromônio a ser acrescentada nos arcos dos melhores caminhos. Como a formiga que obteve a melhor solução até então não necessariamente é da iteração atual, este processo faz com que informação sobre a qualidade das soluções anteriores seja passada para iterações seguintes, o que não ocorre na regra CME, pois a mesma utiliza informações sobre a qualidade das soluções obtidas somente na iteração atual. Testes computacionais apresentados em Sabino et al. (2006) sugerem que o método RNK conduz a melhores resultados.

4.6. Detalhes sobre a implementação do algoritmo YoYo

Este item apresenta as principais rotinas utilizadas na implementação do algoritmo YoYo, desenvolve uma estimativa de sua ordem de complexidade e apresenta a estrutura de dados mais importante do algoritmo, a qual foi utilizada para armazenamento das informações sobre feromônio.

4.6.1. Lógica das principais rotinas

Este item apresenta o pseudocódigo das principais rotinas utilizadas na implementação do algoritmo YoYo. A rotina principal *ACO* usa, em sua lógica, a rotina *RegraDecisão*, que por sua vez usa as rotinas *SortearResposta* e *Viável*.

4.6.1.1. Rotina *ACO*

O pseudocódigo da rotina ACO utilizada no algoritmo YoYo é apresentado na Figura 13. O número total de formigas da colônia, o número de formigas espiãs e o tipo de colônia (i.e. EM ou WT) são informados como parâmetros de entrada. Inicialmente, são definidas seis matrizes de feromônio (apresentadas na Figura 17) e é reservado espaço de memória para as estruturas de dados que armazenam valores de custo para a função objetivo, e os respectivos caminhos encontrados pelas melhores formigas. A variável que controla o tipo de item a ser adicionado no caminho das formigas é inicializada com o valor PRIMA, pois todo caminho começa por uma locomotiva e o primeiro item a ser identificado é então a primeira manobra daquela locomotiva.

Seguem-se as iterações de construção de caminhos. Todas as *m* formigas têm uma chance de construir seu caminho iniciando por cada locomotiva da frota dada. Após definir se a formiga é nativa ou espiã, a construção do caminho prossegue com a chamada da rotina *RegraDecisão* para definir qual será o próximo item do caminho e, em seguida proceder os ajustes apropriados no tempo e acrescentar o novo item na estrutura de dados que armazena os elementos do caminho. Ao final da repetição de construção do caminho, caso se tenha obtido uma solução viável (i.e., se o caminho contém as linhas lógicas de coleta e entrega de todas as ordens de serviço do conjunto R), calcula-se o valor da função objetivo e, caso este valor esteja entre os melhores, atualiza-se a lista dos melhores caminhos. Depois que todas as formigas tiveram a chance de construir seus caminhos, segue-se a atualização da matriz de feromônio.

```
ACO (num. formigas, num. espiãs, tipo de colônia)
   Passo 1: Inicialização
01   Alocar memória e inicializar matrizes de feromônio;
02   Ler os valores de η;
03   Definir valor da solução com o pior valor possível;
04   Alocar memória e inicializar solução com caminho vazio;
05   Ler número de melhores formigas a ser considerado;
06   Alocar memória para armazenar valores das melhores
     soluções e caminho das melhores formigas;
07   Inicializar item = PRIMA;
   Passo 2: Construção
08   Para cada uma das |E| locomotivas repetir {
09      Inicializar caminho com a locomotiva v;
10      Zerar tempo;
11      Para cada uma das m formigas repetir {
12            Definir se a formiga vai ser nativa ou
                    espiã baseado no número especificado de
                    formigas espiãs;
13            Enquanto o caminho da formiga não   contém
                    todas as manobras repetir {
14                  Chamar rotina RegraDecisão (item, resposta)
15                  Se item é NOLO {
16                        Se resposta é OK {
17                              Zerar tempo;
18                              item = PRIMA;
19                              Adicionar locomotiva ao
                                caminho;
                        }
20                        Se não, interromper repetição;
                  }
21                  Se item é PRIMA {
22                        Se resposta é OK {
23                              Atualizar tempo;
24                              item = NOMA;
25                              Adicionar manobra ao caminho;
                        }
26                        Se não, interromper repetição;
                  }
27                  Se item é NOMA {
28                        Se resposta é OK {
29                              Atualizar tempo;
30                              item = NOMA;
31                              Adicionar manobra ao caminho;
                        }
32                        Se não, item = NOLO;
                  }
            } /* fim da repetição para construção do caminho
33            Se caminho tem todas as manobras {
34                  Computar valor da função objetivo (custo);
35                  Atualizar lista dos melhores caminhos;
36                  Atualizar média do custo;
            }
      } /* fim da repetição para cada formiga
37      Se alguma formiga encontrou solução então atualizar a
            matriz de feromônio usando a fórmula (18);
   } /* fim da repetição para cada locomotiva
```

Figura 13: Rotina ACO usada no algoritmo YoYo

4.6.1.2. Rotina RegraDecisão

A rotina *RegraDecisão*, conforme apresentado na Figura 14, processa a mudança de estado em cada passo da construção do caminho da formiga, ou seja, identifica qual a próxima locomotiva ou a próxima manobra a ser adicionada ao caminho. Para tanto, é especificado o tipo de *item* que se quer adicionar e espera-se receber uma *resposta* identificando o item ou, em caso de inviabilidade, a informação de que nenhum item pôde ser identificado.

```
RegraDecisão (item, resposta)
     Alocar e inicializar lista de itens viáveis;
     Se item é NOLO {
        Para cada locomotiva v repetir {
             Se v ainda não foi usada neste caminho {
                   Definir v como viável;
                   Calcular e armazenar probabilidade de escolha
                   desta locomotiva utilizando a fórmula (22), caso
                         a formiga seja nativa ou utilizando a
                         fórmula (23), caso contrário;
             }
        }
        Se for encontrada alguma locomotiva viável {
             SortearResposta (locomotivas viáveis, resposta);
             Retornar resposta;
        }
     }
     Se item é PRIMA ou NOMA {
        Para cada manobra v repetir {
             Se Viável (v,item) {
                   Adicionar v à lista de itens viáveis;
                   Calcular e armazenar probabilidade de escolha
                   desta manobra utilizando a fórmula (22), caso
                   a formiga seja nativa ou utilizando a
                   fórmula (23), caso contrário;
             }
        }
        Se for encontrada alguma manobra viável {
             SortearResposta (manobras viáveis, resposta);
             Retornar resposta;
        }
     }
```

Figura 14: Rotina *RegraDecisão*, usada na rotina ACO

4.6.1.3. Rotina SortearResposta

Esta rotina implementa o método de seleção por roleta, muito utilizado na seleção de cromossomas em algoritmos genéticos, como em Obitko (1998). Primeiro gera-se um número aleatório entre 0 e 1. Depois, a lista de itens viáveis é percorrida, somando-se a probabilidade de escolha de cada item até que esta soma ultrapasse o número aleatório gerado. Quando isto ocorre, o item da lista que estava selecionado é o escolhido. O pseudocódigo da rotina *SortearResposta* está apresentando na Figura 15.

```
SortearResposta (lista de itens viáveis, resposta)
01 Zerar soma acumulada;
03 Gerar um número aleatório rnd, tal que 0 ≤ rnd ≤ 1;
04   Para cada item viável e enquanto soma < rnd      repetir {
05     Selecionar item viável;
06     Acumular em soma a probabilidade de escolha    do item
           viável selecionado;
     }
07   resposta = item selecionado;
08   Retornar resposta;
   }
09 Se não, resposta é [não OK];
```

Figura 15: Rotina *SortearResposta*, usada rotina *RegraDecisão*

4.6.1.4. Rotina *Viável*

A rotina *Viável* verifica se a inclusão da manobra no caminho sendo construído pela formiga pode ser feita atendendo todas as restrições dadas. Esta rotina retorna um valor falso ou verdadeiro para a rotina *RegraDecisão* de acordo com o resultado da verificação feita. Esta rotina tem uma lógica linear e apenas valida, seqüencialmente, cada restrição operacional do PPOLM, conforme apresentado na Figura 16.

```
Viável ()
   01 Se esta manobra já consta no caminho atual, retorne falso;
   02 Se a locomotiva que vai executar esta manobra não pode
      chegar à linha de coleta antes do final da janela de tempo
      de coleta, retorne falso;
   03 Se o desacoplamento da locomotiva não pode ocorrer durante
      a janela de tempo de entrega, retorne falso;
   04 Se a capacidade de tração da locomotiva não é suficiente
      para mover o conjunto de vagões desta manobra, retorne
      falso;
   05 Se esta manobra tem outra manobra predecessora que ainda
      não está com a coleta já realizada, retorne falso;
   06 Caso contrário, retorne verdadeiro.
```

Figura 16: Rotina *Viável*, usada rotina *RegraDecisão*

4.6.2. Estimativa da ordem de complexidade

Para estimar a ordem de complexidade do algoritmo *CompetAnts* aplicado no contexto da solução do PPOLM, basta observar que no algoritmo ACO da

Figura 13 da página 70, em cada iteração, iniciando por cada uma das |E| locomotivas disponíveis no pátio, *m* formigas tentam construir uma solução que contém, no máximo, $(|E|+2n)$ passos, onde *n* é o número de ordens de serviço. Em cada passo na construção do caminho, cada formiga deve escolher uma opção de mudança de estado em um conjunto com menos de *n* opções. Temos então a expressão $|E|.m.(|E|+2n).n$ como primeira estimativa para a complexidade de tempo do algoritmo proposto. Note que a complexidade do algoritmo proposto não depende do número |E| de locomotivas, já que este número é limitado pela quantidade de ordens de serviço, pois, no pior caso, cada ordem de serviço é executada por uma locomotiva diferente. Outro detalhe importante é que todas as demais operações do algoritmo *CompetAnts*, tais como a atualização e depósito de feromônio são $O(n^2)$. Finalmente, deve-se considerar as duas possibilidades para cálculo do custo variável. No caso do custo variável estático, tem-se uma tabela com os valores das distâncias entre cada par de nós do grafo G´ armazenado em uma matriz de adjacências de modo que o acesso é direto e não influencia na complexidade do algoritmo. No caso do custo variável dinâmico, é necessário refazer o cálculo para cada uma das possibilidades de mudança de estado. Este cálculo é feito com o algoritmo de Dijkstra, que no pior caso tem ordem de complexidade $O(n^2)$ (Black, 2006). Como o custo variável dinâmico é calculado para cada uma das possibilidades de mudança de estado, tem-se um novo fator $O(n^2)$ a ser considerado na complexidade final. Utilizando então a notação O e considerando o que foi exposto acima, segue-se que a complexidade final de cada iteração do algoritmo é $O(m.n^2)$ no caso de se utilizar o custo variável estático e $O(m.n^4)$ no caso se utilizar o custo variável dinâmico. Desta forma, nota-se que o algoritmo proposto é capaz de encontrar uma boa solução para o PPOLM com ordem de complexidade polinomial e que o uso do custo variável dinâmico aumenta o tempo de execução do algoritmo, em relação ao caso de uso do custo variável estático, na proporção estimada do quadrado do número de ordens de serviço do PPOLM.

4.6.3. Estrutura de dados para armazenar as informações de feromônio

A quantidade de feromônio depositada em cada passo da construção do caminho da formiga é armazenada em três matrizes de feromônio. Considerando que há p= |R| manobras a serem executadas e q= |E| locomotivas disponíveis no pátio, são utilizadas três matrizes, uma para cada um das três possibilidades indicadas na Figura 11: Uma matriz *(p x p)* armazena a quantidade de feromônio associado às possíveis decisões de adição de mais uma manobra para a mesma locomotiva, uma matriz *(p x q)* armazena o feromônio associado às possíveis decisões de adicionar uma das q locomotivas depois de executar uma das *p* manobras e uma matriz *(q x p)* armazena a quantidade de feromônio associada à decisão de adição da primeira manobra de uma locomotiva. Como é característica do PPOLM a existência de algumas poucas locomotivas no pátio para atender a dezenas de manobras, as três matrizes citadas acima são armazenas em uma única estrutura matricial, onde são inutilizadas *(q x q)* elementos.

Como há duas colônias de formiga, cada uma com suas três matrizes de feromônio, toda a informação de feromônio é então armazenada em uma única matriz *M (p+q, p+q, 2)*, conforme indicado na Figura 17, onde o espaço de memória não utilizado está indicado pela matriz quadrada *(qxq)* hachurada.

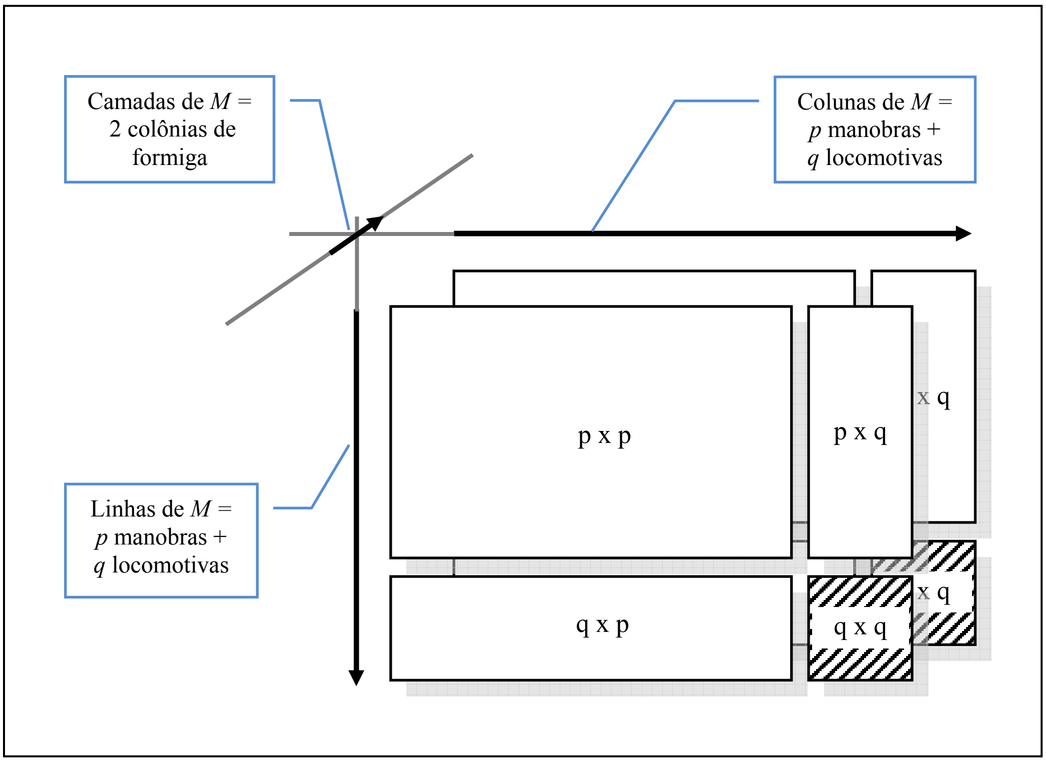

Figura 17: Matriz M (p+q, p+q, 2) de feromônios

4.7. Determinação da rota de cada passo

Se o custo variável a ser utilizado na função objetivo é estático, a rota de cada passo do caminho de coleta e entrega depende somente do leiaute do pátio. Desta forma, o caminho mínimo em G`, e o valor da distância percorrida neste caminho para cada passo pode ser calculada uma vez e fornecida como dado de entrada para o programa que implementa o algoritmo YoYo. Se o custo variável é dinâmico, a determinação da rota mínima deve ser feita durante a construção do caminho, pois a mesma varia com o tempo. A estrutura de dados para armazenar o estado de ocupação das linhas do pátio e a rotina para determinação da rota mínima e o cálculo do custo variável dinâmico estão detalhadas nos dois itens que se seguem.

4.7.1. A modelagem da alocação das linhas do pátio

Para identificar se há conflito de alocação de linhas quando as locomotivas executam as manobras seguindo o plano definido pelo programa é fundamental estabelecer um mecanismo de controle de alocação das linhas, de modo que dois elementos (i.e. comboio, locomotiva escoteira ou conjunto de vagões) não possam ocupar a mesma linha ao mesmo tempo.

Espaço e tempo são as variáveis de controle consideradas no processo e a modelagem utilizada parte das seguintes premissas:

- A velocidade de circulação ao longo de uma linha é um valor constante, utilizado para todas as manobras. Para estimar este valor, pode-se considerar o comprimento médio das linhas do pátio em questão, as fórmulas para cálculo da curva de velocidade (Pachl, 2002) que consideram a oscilação da velocidade em função da aceleração e frenagem da locomotiva e as fórmulas de Davis, que consideram a resistência ao movimento.
- Todo comboio cabe completamente dentro da linha de origem e dentro da linha de destino. A garantia de capacidade das linhas de origem e destino é um dos objetivos da etapa de identificação das

manobras, descrita no item 2.2.2. Vale observar que linhas de circulação podem ser menores que o comprimento do comboio.

- Linhas de circulação não podem ter vagões estacionados, ou seja, a menos de ocupação por um comboio aguardando a liberação da linha adiante, as linhas que estão entre a origem e o destino dos percursos das locomotivas escoteiras e dos comboios estão sempre desocupadas.

As duas últimas premissas acima tornam desnecessário considerar a alocação das linhas pelos conjuntos de vagões estacionados antes e depois da realização da manobra, restando somente a necessidade de controle de alocação das linhas pelos comboios em deslocamento.

A modelagem das linhas do pátio e seu estado de alocação foi feita da seguinte forma: Divide-se a duração do horizonte de planejamento Δh em n intervalos $\{i_0, i_1, i_2,...,i_{n-1}\}$ de igual duração t_0 que são denominados *intervalos de alocação de linha*, ou simplesmente *intervalos*, como serão referenciado neste trabalho. Eventualmente, quando não existe $n \in \mathbb{N} \mid n \cdot t_0 = \Delta h$, convenciona-se que o intervalo i_{n-1} é, excepcionalmente, o único intervalo com duração $t_0' = \Delta h - t_0(n-1) < t_0$.

O valor do intervalo t_0 deve ser definido levando em conta a velocidade média de circulação das locomotivas de manobra no pátio e o tamanho das linhas do pátio, de modo a representar uma duração conveniente para fins de controle de alocação de linhas do pátio, conforme será mostrado adiante.

A fórmula proposta para estimativa inicial do valor do intervalo é a seguinte:

$$t_0 = \frac{\min\{l_{v'}, v' \in G'\}}{2s_a} \tag{26}$$

onde $\min\{l(v), v \in G'\}$ é o tamanho da menor linha do pátio e s_a é a velocidade média de circulação das locomotivas de manobra no pátio.

A idéia da fórmula (26) é definir um tempo t_0 de modo que para percorrer uma linha de pátio, uma locomotiva média, em condições normais, gaste pelo menos dois intervalos. Esta fórmula pode ser utilizada para se estimar o valor inicial de t_0 e a partir daí, podem ser feitos os ajustes necessários pela equipe de planejamento de pátio.

Como será mostrado nos parágrafos que se seguem, o intervalo t_0 possibilita a transformação do tempo em uma variável discreta, simplificando o controle de alocação das linhas.

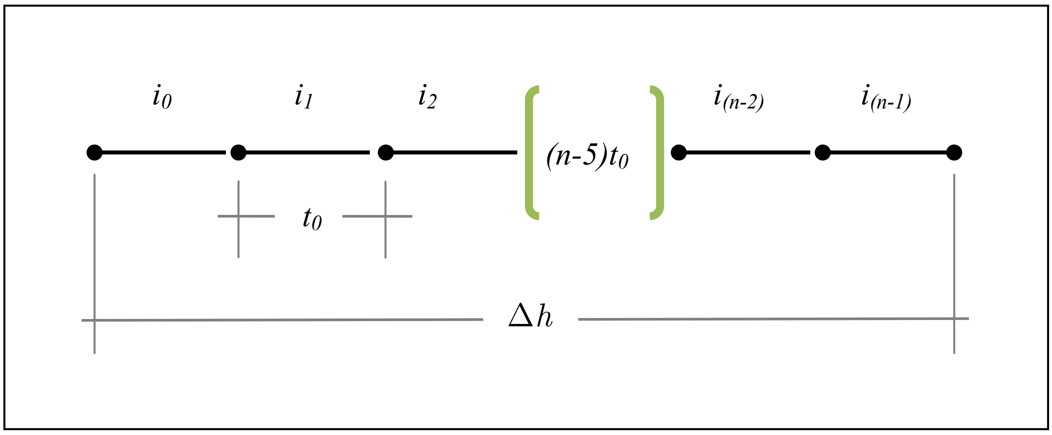

Figura 18: Horizonte de planejamento dividido em intervalos

A Figura 18 mostra o intervalo Δh dividido em n intervalos de mesma duração t_0. Apenas os primeiros três intervalos e os últimos dois intervalos são mostrados explicitamente. Os demais estão omitidos para simplificação do desenho. Os círculos pretos nas extremidades dos intervalos indicam se esta extremidade pertence ou não ao intervalo. Assim por exemplo, $i_2 = \{t \mid 2t_0 \le t < 3t_0\}$ e $i_{n-1} = \{t \mid (n-1)t_0 \le t \le nt_0\}$.

Diz-se que uma linha de pátio i está ocupada durante o intervalo i_k se existe um comboio, uma locomotiva ou um conjunto de vagões na linha i em algum instante do intervalo i_k.

O principal objetivo do intervalo é tornar o tempo uma variável discreta de modo que seja possível mapeá-lo através de colunas de uma matriz. Desta forma, o modelo de dados definido para controle da alocação das linhas se baseia numa matriz e funciona da seguinte forma: Define-se uma matriz $Z(n,l)$ onde n é o número de intervalos em Δh e l é o número de linhas do pátio. Desta forma, é possível associar um intervalo a cada coluna da matriz Z e uma linha de pátio a cada linha da matriz Z. Os elementos da matriz só podem assumir os valores 0 ou 1. O valor 1 no elemento $z(i,j)$ da matriz Z indica que a linha i se encontra ocupada durante o intervalo *j* e o valor 0 indica que a linha *i* se encontra livre durante o intervalo *j*.

4.7.2. O controle da alocação das linhas

O controle de alocação das linhas ocorre paralelamente à construção do caminho das formigas. Inicialmente todos os elementos da matriz *Z* contêm o valor zero. Em seguida, ainda como parte do processo de inicialização, atribui-se o estado de ocupada durante todo o horizonte de planejamento para as linhas onde as locomotivas estão localizadas inicialmente. A cada novo passo acrescentado no caminho de coleta e entrega, calcula-se os intervalos em que cada linha estará alocada no percurso e atualiza-se a matriz *Z* atribuindo-se o valor um aos intervalos em que as linhas estão ocupadas e o valor zero aos intervalos em que as linhas estão desocupadas.

O cálculo dos tempos de deslocamento se baseia nos seguintes valores obtidos diretamente ou calculados a partir dos dados de entrada: tempo de deslocamento da locomotiva na condição escoteira, tempo de espera, tempo de transporte e tempo de serviço. É claro que o grau de precisão das informações fornecidas pelo usuário (e.g. velocidade média das locomotivas, tempo de serviço e comprimento das linhas do pátio) influencia diretamente a precisão dos cálculos para controle de alocação das linhas.

As figuras e explicações que se seguem são usadas para mostrar as premissas e o método utilizado para o cálculo dos tempos de alocação dos intervalos. A Figura 19, na página 79, mostra as três primeiras linhas, as quatro primeiras colunas e as quatro últimas colunas de uma matriz *Z* de alocação de pátio. As linhas A, B e C do pátio estão associadas às primeiras linhas da matriz e os *n-1* intervalos que compõem o horizonte de planejamento estão associados às respectivas colunas da matriz.

Matriz de alocação de linhas do pátio	i_0	i_1	i_2	i_3	...	i_{n-4}	i_{n-3}	i_{n-2}	i_{n-1}
Linha A	1	0	0	0	...	0	0	0	0
Linha B	0	0	0	1	...	1	1	1	1
Linha C	0	1	1	0	...	0	0	0	0
...	...								

Figura 19: Matriz *Z* de alocação de pátio após movimento do comboio κ

Chamemos de κ o comboio composto de uma locomotiva e três vagões da Figura 20. Se κ parte da linha A, passa pela linha C e depois fica estacionado até o final do horizonte de planejamento na linha B, os valores da matriz mostram como fica a representação correspondente à alocação das linhas do pátio para representar este deslocamento de κ.

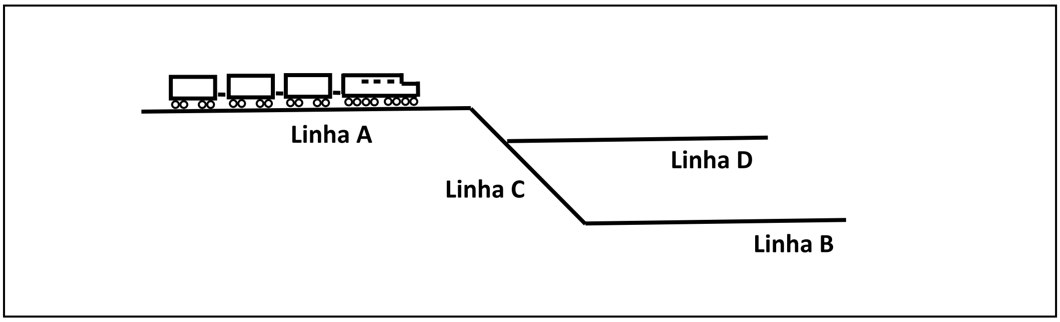

Figura 20: Comboio estacionado em A, pronto para se deslocar até B.

O comboio κ ocupa a linha A somente durante o primeiro intervalo, depois passa pela linha C durante os intervalos i_1 e i_2. Ao final do intervalo i_2 a linha C está desocupada e o comboio κ já se encontra completamente na linha B, onde fica estacionado até o fim do intervalo i_{n-1}, ou seja, até o final do horizonte de planejamento.

Como a unidade de tempo de alocação é medida em intervalos, se em algum instante *t*, contido num intervalo i_k, uma linha se encontra ocupada, então a linha é considerada ocupada durante todo o intervalo i_k.

Os tempos de ocupação dos intervalos são calculados considerando o comprimento de cada linha por onde passa o comboio e o comprimento do comboio. Uma linha por onde passa um comboio é considerada ocupada desde o momento em que a frente do comboio atinge uma de suas extremidades até o momento em que o final do comboio deixa a outra extremidade. A linha onde inicialmente está localizado um comboio é considerada ocupada até que o comboio deixe completamente a linha.

A linha de destino de um comboio é considerada ocupada até o fim do horizonte de planejamento ou até que uma manobra executada posteriormente retire o comboio de lá. Além disso, se a linha de destino não estiver livre desde o momento em que o início do comboio a atingir até o momento em que o centro do comboio estiver alinhado com o seu centro, então a linha não será considerada desocupada.

A Figura 21 mostra a vista lateral do comboio κ, localizado na linha A, pronto para se deslocar até a linha B. A linha A aparece representada em preto e a linha B em cinza. O ponto central de cada linha está assinalado com um pequeno triângulo da cor da linha. Na figura estão indicados o comprimento l_0 do comboio e os comprimentos das linhas A e B, que são l_a e l_b respectivamente. O ponto de referência de cada linha é considerado o meio da linha. O meio do comboio é o ponto de referência do comboio. Assim, o meio do comboio está alinhado inicialmente com o meio da linha A e ficará, ao final do deslocamento, alinhado com o meio da linha B.

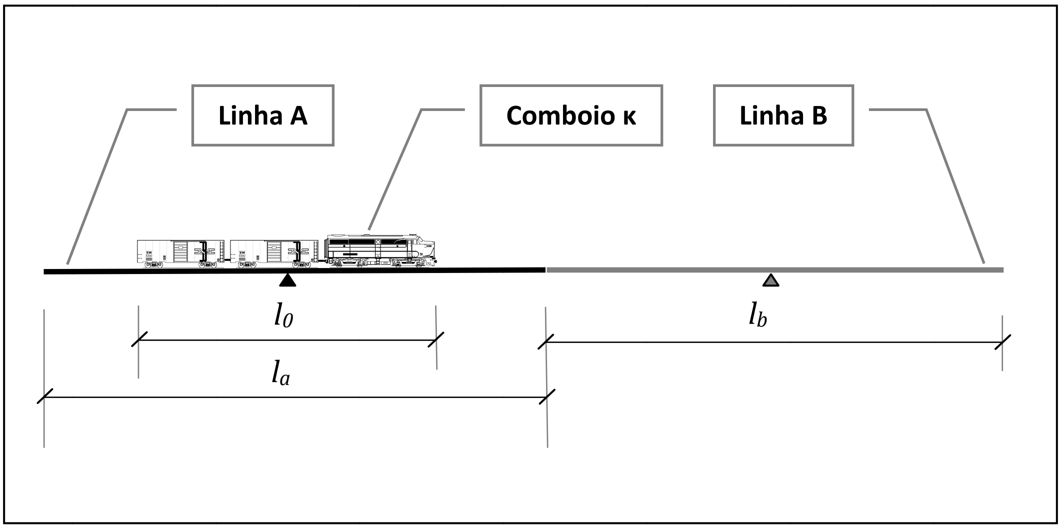

Figura 21: Dimensões envolvidas no cálculo do tempo de alocação das linhas

As seguintes proposições ilustram os critérios de cálculo de espaço durante o deslocamento do comboio. Os respectivos tempos de deslocamento são então obtidos como resultado da divisão do espaço percorrido pela velocidade de deslocamento, a qual é deduzida a partir dos dados de entrada.

- O comboio atinge a linha B após percorrer um espaço $l = (l_a - l_0)/2$ e, a partir deste momento, a linha B é considerada ocupada;
- O comboio só libera a linha A após o seu último vagão deixar a linha A, ou seja, após percorrer o espaço $l' = (l_a - l_0)/2 + l_0 = l_a + l_0/2$;
- O comboio chega à linha B depois de percorrer o espaço $l'' = (l_a + l_b)/2$.

Diante da modelagem utilizada nota-se que o controle de alocação de linhas está baseado na divisão do horizonte de planejamento em unidades discretas de tempo, chamadas de intervalo, e no mapeamento destes intervalos e das linhas do pátio em uma matriz de alocação de linhas. Vale notar que o valor definido para a duração do intervalo t_0 influencia na quantidade de memória necessária para a matriz de alocação de linhas. Intervalos muito curtos implicam em uma matriz com muitas colunas e numa demanda maior por recursos computacionais para processar o controle de alocação de linhas. Intervalos muito longos, por outro lado, reduziriam a precisão do controle de alocação de linhas do pátio, comprometendo assim a qualidade das estimativas feitas durante a elaboração da solução.