



GERENCIAMENTO DE MANOBRAS E DESCARREGAMENTO DE TRENS DE MINÉRIO DE FERRO EM PÁTIOS FERROVIÁRIOS USANDO ALGORITMO GENÉTICO

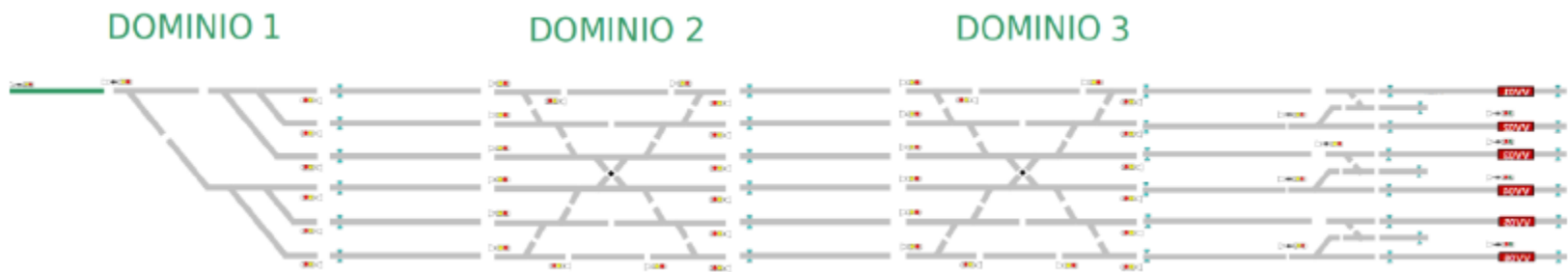
**Juan Xabier E. de Aquino Calles
Leandro Colombi Resendo
Rodrigo de Alvarenga Rosa**

1. Introdução

- Pátios ferroviários são fundamentais para a distribuição de recursos minerais extraídos;
- 70% de toda a frota de vagões de uma ferrovia encontram-se em um terminal ferroviário;
- Trens carregados de minério chegam no pátio de recepção – onde são descarregados, os vagões descarregados prosseguem para o pátio de classificação – onde podem ser enviados para limpeza ou manutenção, e finalmente para o pátio de formação – onde novos trens são formados e enviado às minas para serem carregados novamente.
- Este trabalho contribui para o processo de tomada de decisão no pátio de recepção.

1. Introdução

- Pátios ferroviários caracterizam-se pela disposição de vários segmentos de linha em paralelo.
- A troca entre segmentos de linha é feita com a ajuda de Aparelhos de Mudança de Via, ou AMV
- Nos pátios de recepção o descarregamento de trens é feito por viradores de vagões (à extrema direita na Figura).
- Um segmento de linha não pode ser ocupada simultaneamente por dois ou mais trens.



1.1 Objetivo

➤ OBJETIVO GERAL:

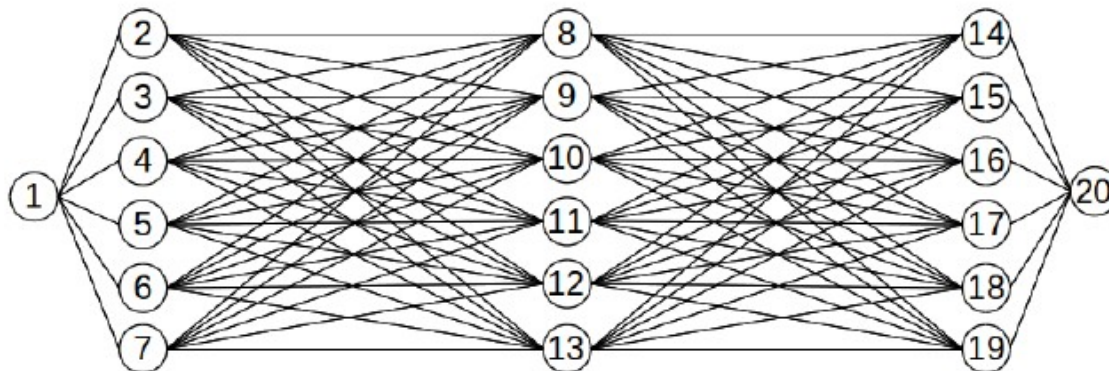
Auxiliar na tomada de decisão de virador e rota para trens que entram no pátio de recepção.

Proposta de um Algoritmo Genético e um modelo de Programação Linear Inteira.

2. Descrição do problema

Computacionalmente um pátio ferroviário pode ser representado como uma estrutura de grafo direcionado $G = (V, E)$

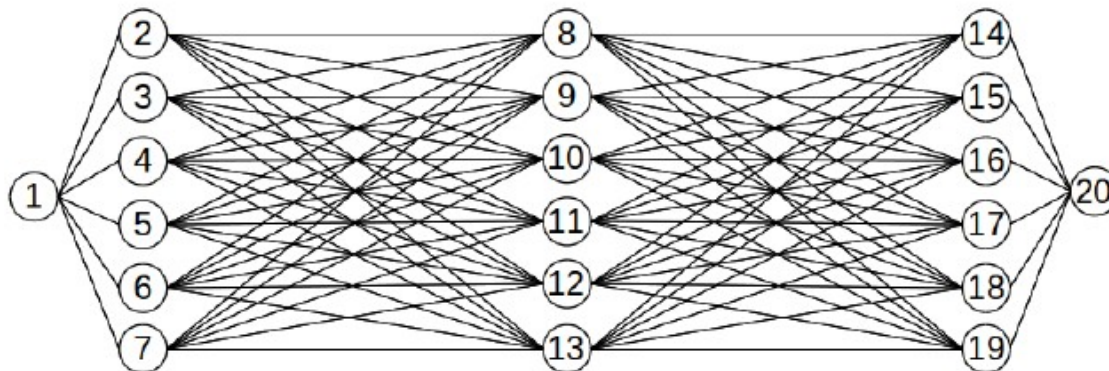
No grafo G do pátio, cada segmento de linha é representada por um vértice pertencente ao conjunto V e cada enlace do conjunto E representa uma sequência de AMV que faz a conexão entre segmentos de linha adjacentes. Viradores também fazem parte do conjunto V (à extrema direita na Figura).



2. Descrição do problema

Os custos representam o tempo de passagem de trens em cada elemento do grafo G .

Foi inserido um nó virtual no final do grafo representando o término de fluxo para o modelo ILP.



2. Descrição do problema

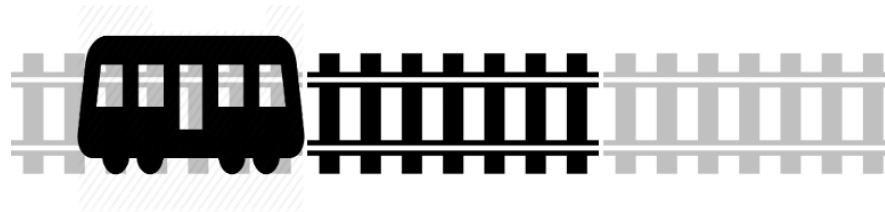
O caminho de um lote é representado por uma lista ordenada de vértices do grafo.

$$R = \{V_1, V_2, \dots, V_n\}$$

2. Descrição do problema

Sobre o tempo que um trem gasta num segmento de linha, tem-se os seguintes eventos:

- 1) Parte de frente de trem entra no segmento de linha. Segmento de linha ocupado.



- 2) Parte de trás do trem entra no segmento de linha. Segmento de linha anterior desocupado.



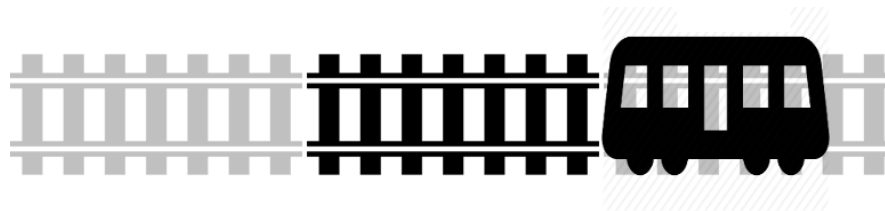
2. Descrição do problema

Sobre o tempo que um trem gasta num segmento de linha, tem-se os seguintes eventos:

- 3) Parte de frente do trem entra no segmento de linha seguinte. Próximo segmento de linha ocupado.



- 4) Parte de trás do trem entra no segmento de linha seguinte. Segmento de linha desocupado.



3. Modelo ILP

Parâmetros

N : Conjunto de nós, onde cada nó representa um segmento de linha, sendo $|N| = n$.

L : Conjunto de lotes que entrarão no pátio.

$E[i][j]$: Matriz de adjacência, representando a conexão entre os nós i e j . Sendo cada enlace representando um conjunto de AMVs.

$TN[i]$: Tempo gasto para uma extremidade do lote atravessar o nó i . Note que esse parâmetro identifica o comprimento do nó.

$TAMV[i][j]$: Tempo gasto para uma extremidade de um lote atravessar uma aresta ij referente a um conjunto de AMVs.

$B[l]$: Momento em que o lote l chega no pátio (primeiro nó).

$TL[l]$: Tempo que o lote l gasta para atravessar completamente um ponto fixo no pátio. Note que esse parâmetro indica o “comprimento” do lote.

3. Modelo ILP

Variáveis de decisão

$x_{i,j}^l$: Variável binária que vale 1 se o lote l se desloca do nó i até o nó j , usando a aresta ij e zero caso contrário.

b_i^l : O instante em que o lote l chega no nó i .

3. Modelo ILP

Função objetivo

A função objetivo representa a minimização do tempo que cada lote demora para atravessar a recepção do pátio ferroviário.

$$\text{Minimizar: } \sum_{l \in L} B^l \quad (1)$$

Sujeito a:

$$\sum_{j \in N} x_{1,j}^l = 1 \quad \forall l \in L \quad (2)$$

$$\sum_{i \in N} x_{i,n}^l = 1 \quad \forall l \in L \quad (3)$$

$$\sum_{j \in N} x_{i,j}^l - \sum_{j \in N} x_{j,i}^l = 0 \quad \forall l \in L; i \in \{2, \dots, n-1\} \quad (4)$$

$$b_1^l \geq B[l] \quad \forall l \in L \quad (5)$$

$$b_j^l \geq b_i^l + TN[i] + TAMV[i][j] - M(1 - x_{ij}^l) \quad \forall l \in L; \forall i, j \in N / j > i \quad (6)$$

$$b_i^l \geq b_i^m + TN[i] + TL[l] - M \sum_{j \in N} (2 - x_{ji}^l - x_{ji}^m) \quad \forall i \in N; \forall l, m \in L / B[l] > B[m] \quad (7)$$

4. Algoritmo genético

- O algoritmo genético (AG) é uma meta-heurística que imita o processo de seleção natural das espécies para gerar soluções de boa qualidade, usado amplamente para problemas de busca heurística e otimização combinatória.
- O AG realiza uma busca multidirecional mantendo, a cada iteração, uma população de soluções candidatas, chamadas de cromossomos (formados por genes), que trocam informações entre si para gerar novos indivíduos.
- A busca no espaço de soluções é realizada por meio de duas operações básicas, o cruzamento (ou crossover) e a mutação

4. Algoritmo genético

Representação de um cromossomo

Um cromossomo (solução) é representado como uma lista de rotas associadas a cada lote, ordenados por ordem de chegada. Cada rota representa um gene, e o último nó de cada rota corresponde a um virador.

Rotas

N_{11}	N_{12}	...	$N_{1(n-1)}$	N_{1n}
N_{21}	N_{22}	...	$N_{2(n-1)}$	N_{2n}
...				
$N_{(m-1)1}$	$N_{(m-1)2}$...	$N_{(m-1)(n-1)}$	$N_{(m-1)n}$
N_{m1}	N_{m2}	...	$N_{m(n-1)}$	N_{mn}

Viradores

4. Algoritmo genético

Geração inicial

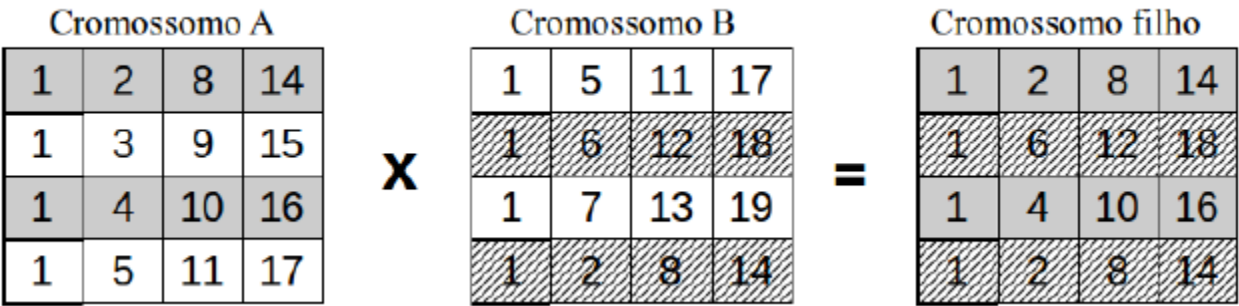
Determina os locais, dentro do espaço de busca, nos quais a heurística iniciará a busca por boas soluções.

- Solução inicial aleatória: cada rota é formada tomando aleatoriamente nós adjacentes.
- Solução inicial baseada em regras: para cada rota, primeiro é escolhido o virador mais “próximo”, baseado no próprio custo. Em seguida o menor caminho até o virador é traçado. Depois de gerada essa rota, o virador tem seu custo acrescentado, para representar uma fila de espera de atendimento.

1º Lote				2º Lote				3º Lote				4º Lote			
1	2	8	14	1	2	8	14	1	2	8	14	1	2	8	14
				1	3	9	15	1	3	9	15	1	3	9	15
								1	4	10	16	1	4	10	16
												1	5	11	17

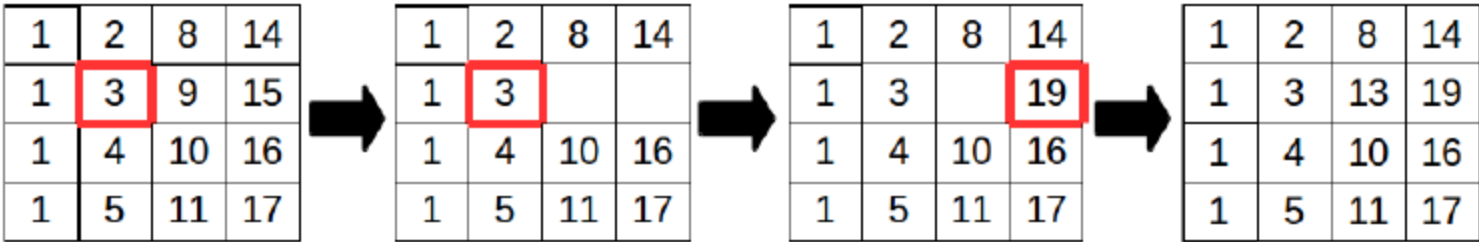
4. Algoritmo genético

Crossover



4. Algoritmo genético

Mutação



4. Algoritmo genético

Função fitness

A função fitness corresponde ao somatório do tempo gasto por cada trem em atravessar todo o pátio.

5. Experiências e resultados

Objetivos

- Testar a convergência;
- Calibrar o AG;
- E testar a sua eficiência.

A heurística foi implementada em Python 2.7, usando os pacotes Numpy e NetworkX.

As experiências foram realizadas usando um processador Intel Core i7-3.4GHz com oito núcleos e 8GB de memória RAM em um sistema operacional Ubuntu 14.04 LTS.

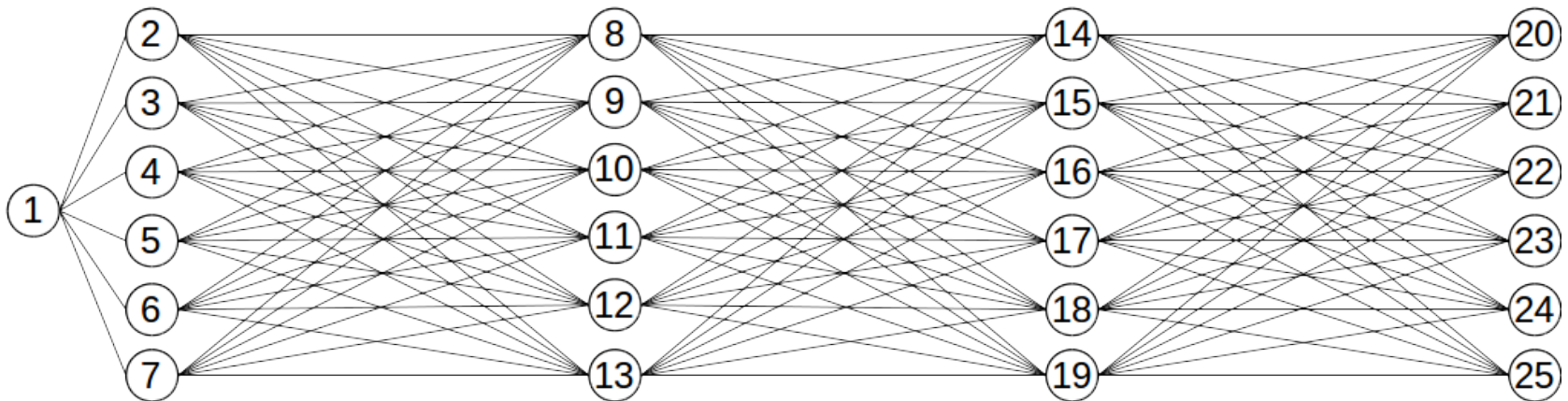
5. Experiências e resultados

Teste de convergência

Consistiu na execução incremental do algoritmo genético (*sol. Inicial*; *sol. Inicial + crossover*; e *sol. Inicial + crossover + mutação*), tomando duas instâncias de pátio ferroviário.

Ambas instâncias de layout com 25 nós e 114 arestas.

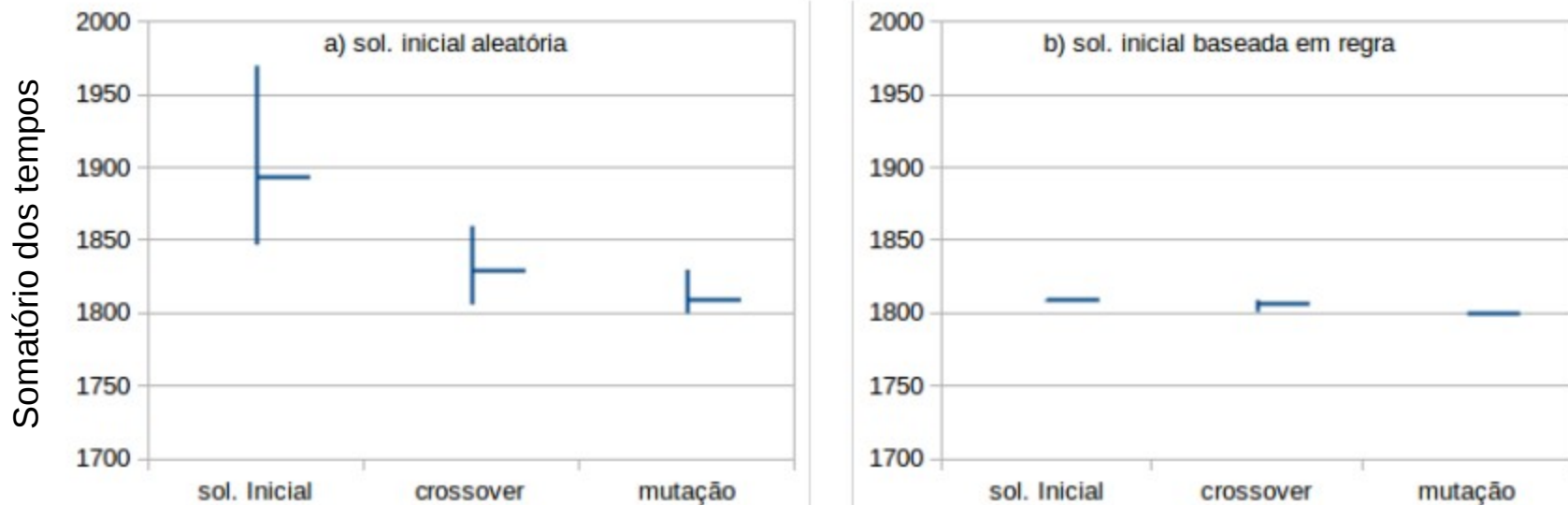
- Instância 1: Layout simétrico (todas as arestas com mesmo custo)
- Instância 2: Layout não simétrico (arestas de custos variados)



5. Experiências e resultados

Teste de convergência

Layout simétrico

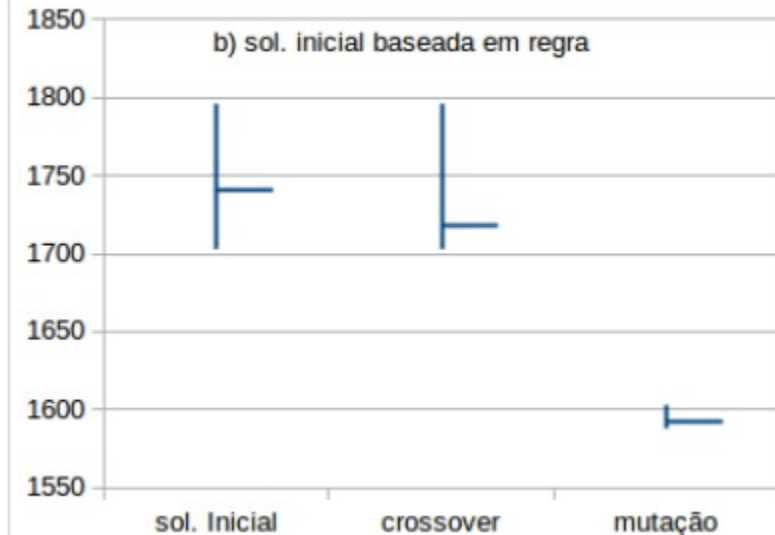
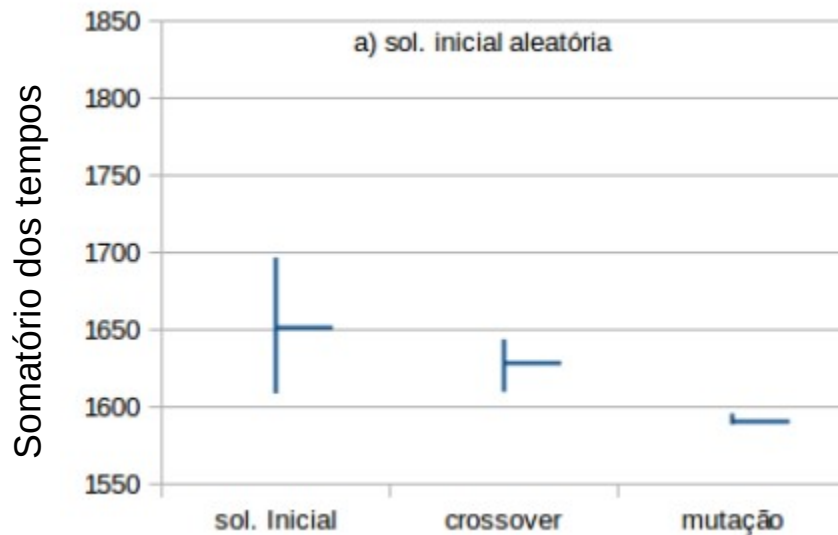


***Cada versão de heurística foi executada 10 vezes para cada layout**

5. Experiências e resultados

Teste de convergência

Layout não simétrico



5. Experiências e resultados

Calibração

Parâmetros:

- Quantidade de iterações: 50, 100, 250, 500 ou 1000
- Tamanho de geração (a quantidade máxima de cromossomos): 10, 30 ou 50
- Probabilidade de mutação: 1%, 5% ou 10%

Num total de 45 combinações. O experimento foi realizado num pátio de recepção de layout de 25 nós e 120 arestas. Cada uma das 45 combinações foi executada 10 vezes e o valor médio registrado.

Iterações	Tam. Geração	Mutação
250	50	10%
500	30	10%
500	50	10%
1000	30	5%
1000	30	10%
1000	50	5%

***As seis melhores combinações**

5. Experiências e resultados

Eficiência

A quantidade de lotes de trens foi experimentada em dez, cem e mil lotes.

O mesmo layout de 25 nós e 114 arestas foi usado.

O modelo ILP aproximadamente 4 horas para resolver para 10 lotes.

Num. Lotes	Tempo processamento
10	600 ms
100	6 s
1000	60 s

5. Conclusões

- Projetado um algoritmo genético para auxiliar na escolha de rotas e viradores para trens de minério em pátios de recepção.
- Cromossomos representam conjuntos de rotas com destino a viradores.
- Experimentos mostram que soluções baseadas em regra convergem muito rápido para pátios simétricos.
- Soluções iniciais aleatórias convergem regularmente tanto para layouts simétricos e não simétricos.

5. Conclusões

Trabalhos futuros:

- Aplicar o algoritmo genético proposto num pátio de recepção real.
- Estudar aplicabilidade em pátios de classificação e formação.

Agradecimentos



CNPq (469172/2014-2 e 307439/2016-0)