

SISTEMA DE CONTROLE INTELIGENTE DE LINHAS FERROVIÁRIAS

LUIZ C. MAIA JR, GREGORY CUVELIE, REINALDO A. C. BIANCHI

*Faculdade de Engenharia Industrial – UNIFEI – Departamento de Engenharia Elétrica
Av. Humberto de A. C. Branco, 3972, CEP 09850-901
São Bernardo do Campo, SP, Brasil*

*E-mails: luiz.maia@transport.alstom.com, gregory.cuvelie@transport.alstom.com,
rbianchi@fei.edu.br*

Resumo — Este artigo descreve o projeto e a implementação de um sistema de controle de trens urbanos que realiza a revisão e a suavização dos intervalos entre partidas das composições. Este sistema foi implementado em software e adapta o sistema de controle dos trens às novas condições de tráfego cada vez que algum evento fora do comum ocorrer dentro da via férrea. O sistema é baseado em técnicas de Programação Genética, e é composto pelo módulo de simulação, que implementa o controle da movimentação dos trens dentro da via, e por uma interface homem-máquina, que permite observar a movimentação dos trens na via e a interação do usuário com a simulação. Os testes realizados indicaram que o sistema é eficiente, sendo capaz de se adaptar à maioria dos eventos de interrupção ou de atraso de composições.

Abstract — This article describes the project and implementation of a control system for urban trains, which is able to do revisions and smoothing of the intervals between departures of the compositions. This system was implemented in software and adapts the control system of the trains to the new traffic conditions each time an odd event happens on the railway. The system is based on Genetic Programming techniques, and is composed of a simulation module, that implements the control of the trains movements in the railway, and of a man-machine interface which allows to observe the movement of the trains in the railway and to interact with the simulation. The system was exhaustively tested and the experimental results had indicated that it is efficient, being capable of adapting to the majority of the interruption events or the delay of compositions.

Keywords — Genetic Algorithms, Artificial Intelligence, Control Applications, Train Control.

1 Introdução

Este artigo descreve o projeto e a implementação de um sistema de controle de trens urbanos que realiza a revisão e a suavização dos intervalos entre partidas das composições. Este sistema foi implementado em software, e adapta o sistema de controle dos trens às novas condições de tráfego cada vez que algum evento fora do comum ocorrer dentro da via férrea.

O sistema é baseado em técnicas de Programação Genética, e é composto pelo módulo de simulação, e por uma interface homem-máquina.

O Simulador implementa o controle da movimentação dos trens dentro da via. Ele utiliza técnicas de Programação Genética (Koza, 1992) para evoluir uma população de programas, que são testados dentro de um ambiente simulado, e cujos melhores são selecionados para controlar os trens.

A Interface Homem Máquina (IHM) permite observar a movimentação dos trens na via e interagir com a simulação da programação genética de tal forma que facilite a compreensão dos acontecimentos. Ela foi desenvolvida utilizando o “FrontVue”, uma ferramenta desenvolvida especialmente para interfaces gráficas que facilita a configuração dos pontos de exibição. É um cliente com um protocolo de comunicação padronizado que recebe comandos de um servidor OPC de controle de vias e executa esses comandos. É baseado em VBA e serve para simplificar a interface gráfica de um programa de controle de vias.

Na próxima seção são apresentadas de forma sucinta, características básicas da Programação Genética. A seção 3 descreve o sistema de controle

de linhas ferroviárias e a seção 4 Sistema de Controle Inteligente de Linhas Ferroviárias. Os resultados são apresentados na seção 5 e a última seção apresenta as conclusões deste trabalho.

2 Programação Genética

De acordo com a Teoria da Evolução de Darwin, os organismos mais adaptados ao mundo em que vivem são os que terão maiores chances de sobrevivência. Os organismos que existem hoje são consequência da evolução de outros organismos inferiores que se extinguíram e que sem eles, provavelmente, não existiriam. Cada criatura nessa cadeia é o produto de uma série de “acidentes” que têm acontecido continuamente sobre a pressão seletiva do ambiente.

Programação Genética (Koza, 1992) é uma forma de computação baseada em simulação de evolução de indivíduos. Nela é aproveitado o fato de que a evolução na natureza é criativa, pois produz muitas vezes resultados inesperados, impensáveis e não-lineares, diferente do modo de programação usual. Algumas características importantes da Programação Genética são descritas a seguir.

Em PG o indivíduo que deve ser evoluído é um programa de computador definido como uma árvore de funções e terminais, que constituem suas características e define o comportamento no ambiente para o qual foi desenvolvido. Cada função é um ramo da árvore e cada terminal uma folha do indivíduo. Funções podem ser condições, sensores, operações aritméticas ou lógicas etc. e são usadas para captar informações sobre o ambiente e, convenientemente, para melhorar o desempenho de cada indivíduo. Já os

terminais geralmente representam as ações que os indivíduos criados podem executar. Uma população contém um determinado número de indivíduos que é renovada a cada iteração, criando uma nova geração.

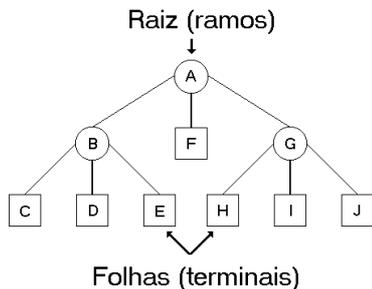


Figura 1. Exemplo de um indivíduo

Fitness é um parâmetro que mede o “encaixe” da espécie em seu ambiente, sendo usado para descartar, a cada iteração, os piores indivíduos. Existem várias definições para *fitness*, sendo aqui utilizada a seguinte: “Quanto maior o *fitness* do indivíduo maior sua probabilidade de sobrevivência e adaptação ao ambiente”. Com este dado calculado, os indivíduos podem ser organizados de forma descendente, sendo então eliminados os piores e mantidos os melhores para serem reproduzidos ou cruzados.

A criação de uma nova geração pode ser feita de três maneiras básicas distintas: a Reprodução, onde os melhores indivíduos da geração atual são simplesmente copiados para a nova geração, com o objetivo de não se descartar os melhores indivíduos; o Cruzamento, onde são escolhidos dois indivíduos (pais) bem adaptados, dos quais são criados dois descendentes que vão estar na próxima geração – o objetivo é um novo indivíduo seja criado usando o que há de melhor de uma geração, com alguma variação; finalmente, na Mutação um indivíduo é selecionado, depois um ponto de mutação, e a partir dele é criado um novo ramo – o objetivo é melhorar um indivíduo, mas com isso ele pode perder desempenho e ser eliminado.

Abaixo é mostrado apenas um exemplo de cruzamento pois a mutação não foi utilizada neste desenvolvimento.

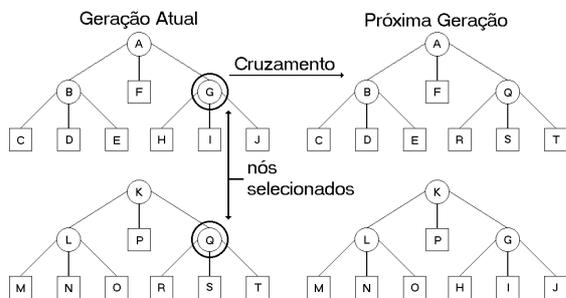


Figura 2. Exemplo de cruzamento

3 Linhas Ferroviárias

O domínio de linhas ferroviárias envolve todos os dispositivos e métodos de uso desses dispositivos com o objetivo de obter uma movimentação segura. A seguir são apresentadas as definições mais importantes.

3.1 Circuito de via (CDV)

Trecho da via metroferroviária em que existem sensores para a detecção de ocupação/desocupação de trens. Um CDV é considerado ocupado quando um trem entra nele, mesmo que não o esteja ocupando por completo. Um CDV pode ser maior que um trem e vice-versa. E um trem pode ocupar apenas um ou mais CDV's ao mesmo tempo.

Tabela 1. Estados de um CDV

| Livre | Requisitado | Ocupado |
|-------|-------------|---------|
| | | |

3.2 Máquina de chave (AMV)

Dispositivo instalado ao longo da linha nos desvios ou interseções entre linhas, e que permite a passagem de um trem de ou para um tronco principal a um ramal principal ou secundário ou uma outra linha. Uma chave pode estar na posição normal, reversa, em movimentação ou sem correspondência.



Figura 3. Máquina de chave na posição normal

3.3 Estação

Instalações prediais projetadas para o embarque e desembarque de passageiros.



Figura 4. Estação JTI

3.4 Regulação de trens

Função responsável por supervisionar e controlar a movimentação dos trens de forma a minimizar atrasos, atender à demanda de passageiros em um dado instante de tempo e verificar o cumprimento da tabela horária ou do intervalo programado.

3.5 Sentido de tráfego

Sentido no qual se movem os trens dentro de um trecho delimitado por duas regiões de máquina de chave em uma dada via.

3.6 Sinaleiro

Dispositivo instalado ao lado da via e que permite a visualização das condições da via à sua frente.

Tabela 2. Estados de um Sinaleiro.

| Aberto | Fechado |
|---|---|
|  |  |

3.7 Rota

A rota é um conjunto um ou mais de CDVs em seqüência na via, que formam o caminho do trem. A rota pode ser formada só por CDVs ou pode ter uma máquina de chave (que consideramos dois CDVs que podem mudar de posição). Geralmente compreende o intervalo entre máquinas de chave ou estações, delimitada por sinaleiros. Todas as rotas da via são pré-definidas, ou pela posição dos sinaleiros, ou pelo das estações. Assim, uma rota sempre termina em um sinaleiro ou depois de uma estação.

3.8 Zona de manobra

Espaço onde fica o estacionamento dos trens e/ou onde eles fazem as manobras de inversão de cabine.

3.9 Tempo de parada

Intervalo de tempo em que o trem fica parado em uma estação. Esse intervalo depende se a estação é ou não um terminal e do fluxo de passageiros que ela possui.

3.10 Requisição de rota

Para que o trem possa percorrer uma rota à sua frente ele deverá requisitá-la. Na requisição é verificado se a rota está desocupada ou ocupada. Se estiver desocupada a requisição será confirmada e o trem passa a ser dono da rota. Caso esteja ocupada, o trem entra em uma fila de requisição, sendo que esta fila contém os números dos trens que pretendem requisitá-la a dando preferência à ao primeiro trem e ao trem mais veloz. Um trem deve requisitar sempre o máximo de rotas possíveis para evitar que depois ele não consiga terminar uma viagem e tenha que parar.

3.11 Tempo de percurso

É o tempo necessário para que um trem saia de um terminal e chegue até a outra extremidade ou até uma estação intermediária, sendo que cada estação em cada via tem o seu tempo de percurso.

3.12 Tratamento de dados do campo

Com os eventos gerados pela movimentação dos trens que são detectadas através de aparelhos específicos distribuídos ao longo da via, essas informações são concentradas em equipamentos que fazem as primeiras consistências para que haja o máximo de segurança, ou seja, não executa ou envia para o servidor dados que ponham em risco o funcionamento dos trens, esse servidor trata os dados e executa ou não as ações corretivas necessárias.

4 Sistema de Controle Inteligente de Linhas Ferroviárias

Utilizando-se das técnicas de Programação Genética foi desenvolvido um sistema de controle de trens que consiste em um simulador de via ferroviária e um programa que simula um operador, ou seja, tenta substituí-lo na tomada de decisões dentro das situações de erro que ele conhece. O sistema implementado possui as seguintes características:

4.1 Parâmetros

- População: 500;
- Cruzamento: 70%;
- Reprodução: 30%;
- Número de vezes que cada indivíduo é testado para determinar fitness: 1;
- Número de vezes que a árvore é executada a cada teste: 20;
- Limite no número de pontos da árvore (evita estouro de memória): 1200;
- Tempo entre cada comando que o indivíduo manda para o simulador: 5 (ms).

4.2 Funções

Os números à frente de cada função são como elas foram chamadas dentro da árvore:

- DOIS (11): executa duas ramificações;
- TRES (12): executa três ramificações;
- FR_DEVAGAR (13): verifica se trem a frente está mais lento;
- ATRASADO (14): verifica se trem está atrasado;
- ADIANTADO (15): verifica se trem está adiantado;

4.3 Terminais

Os números à frente de cada terminal são como eles foram chamados dentro da árvore:

- TRM_ACELERAR (0): acelera o trem sendo controlado até um limite máximo;
- TRM_FREAR (1): freia o trem sendo controlado até um limite mínimo;
- TRM_PARAR (2): para o trem sendo controlado;
- TRM_DESVIAR (3): faz com que o trem desvia pela via paralela;
- TRM_DIMINUIR_TP (4): diminui o tempo de parada da estação até um limite mínimo;
- TRM_AUMENTAR_TP (5): aumenta o tempo de parada da estação até um limite máximo;
- TRM_CRIAR (6): cria um trem na zona de manobra de VDF;
- TRM_REMOVER (7): remove um trem na zona de manobra de VDF;
- TRM_POSTERIOR (8): faz com que o indivíduo passe a controlar o trem que está a frente;
- TRM_ANTERIOR (9) faz com que o indivíduo passe a controlar o trem que está atrás;
- TRM_PARAR_EST (10): define se o trem sendo controlado deve ou não parar nas plataformas.

4.4 Fitness

O *fitness* foi definido como uma penalização aplicada a uma pontuação inicial igual para todos os indivíduos (30.000 pontos). A cada comando executado o indivíduo é penalizado se ocorreu algum dos seguintes casos e subtraindo-se os valores informados em cada caso:

- Trens atrasados ou muito adiantados - 50;
- Trens parados por muito tempo na estação ou em qualquer ponto - 100;
- Muito tempo com velocidade acima da média - 50;
- Fazer com que o trem acelere muito/freie muito ou execute isso várias vezes seguidas - 50;
- Desviar de um trem sem necessidade (e.g. desviar de trem que esteja com velocidade maior que a sua e indo na mesma direção.) - 100.

Sendo assim, o *fitness* final é o que sobrou dos 30.000 pontos iniciais subtraindo-se os erros que aconteceram durante sua execução.

5 Resultados

O gráfico evolutivo mostrado a seguir mostra o desenvolvimento de uma população por 29 gerações.

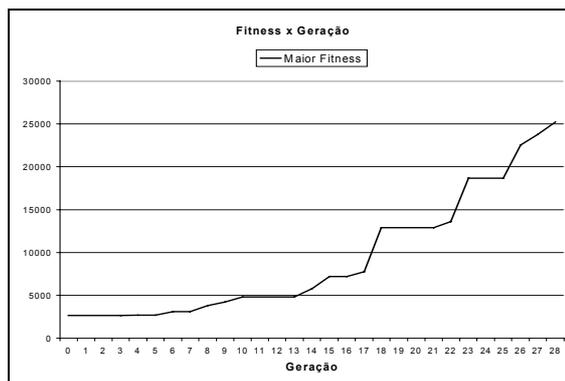


Figura 5. Gráfico Evolutivo de 29 gerações.

Pode-se perceber que no início existe uma certa estabilidade, mas a partir da décima geração o ritmo de evolução parece manter-se quase constante.

A tabela 3 mostra o exemplo de um indivíduo com dezesseis pontos de complexidade na forma como ele é salvo em arquivo (atravessamento *in-order* da árvore do indivíduo). Os números da tabela a seguir são os mesmos descritos na seção anterior e representam uma função ou um terminal.

Tabela 3. Exemplo de um indivíduo

12 13 10 15 13 14 11 6 3 0 10 13 0 4 4 1

A seguir são mostrados as formas de interação com o programa. A tela de simulação de erros (figura 6) envia comandos diretamente para a via, depois de recebidos e executados os comandos o simulador detecta as mudanças ocorridas e se houver algum indivíduo em teste, o indivíduo executa as ações corretivas que ele aprendeu durante sua evolução para corrigir o fluxo de trens na via.

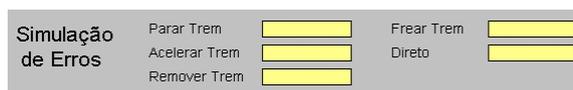


Figura 6. Tela com erros que podem ser simulados

O sinóptico da via (figura 7) é a representação gráfica do trajeto executado pelos trens e possui as indicações necessárias para localização de cada trem de acordo com um número associado. Ele segue as definições apresentadas anteriormente para representar ocupação, requisição e etc.

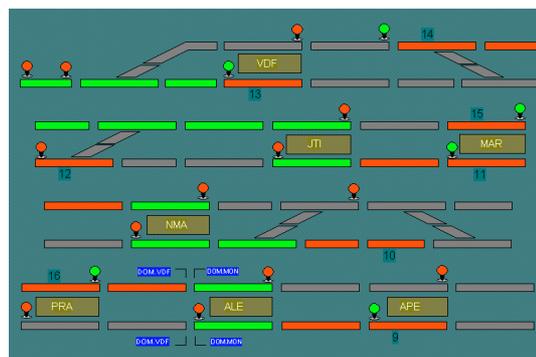


Figura 7. Sinóptico da via com trens se movimentando

Esta tela do sinóptico pode ser dividida em quatro segmentos de dois pares de vias unidas por máquinas de chave, mostradas de cima para baixo. A via mostrada na parte de cima do par é a Norte, que normalmente comporta movimentos da esquerda para a direita (excetuando-se casos de desvio), enquanto a via Sul comporta movimentos da direita para a esquerda. Juntando estes trechos lado a lado obtém-se metade de uma via. Dessa forma é fácil compreender que o trem de número '15' está metade no último CDV do segundo segmento da via norte e metade no primeiro CDV do terceiro, tendo a via liberada por mais um CDV.

O simulador possui todos os objetos da via instanciados na memória, permitindo o controle de todos eles. Isto não impede que o simulador de erros tenha acesso e possa modificar o estado desses objetos.

Os comando possíveis do simulador são:

- Criar: cria um trem na via fazendo manobra de saída na estação VDF;
- Iniciar: distribui 16 trens na via para criar uma situação de teste ou evolução de indivíduos;
- Reset: apaga todos os trens da via;
- Treinar: cria uma população de indivíduos para ser evoluída;
- Testar: carrega o melhor indivíduo da última evolução para ser testado durante uma simulação.



Figura 8. Interface do simulador com comandos da PG

Finalmente, a figura 9 (próxima página) apresenta quinze imagens de uma seqüência de movimentação durante uma das simulações executadas.

Antes de se entender as figuras é importante saber que um trem inicialmente se encontra parado em apenas um CDV (geralmente o último de uma rota) e movimenta-se da seguinte forma:

1. Requisita a próxima rota e espera parado até que esta seja liberada;
2. Ao ter a via liberada, ocupa o próximo CDV da via, mantendo sempre dois CDVs ocupados enquanto se encontra em movimento.
3. O número do trem está sempre na parte de trás, no último CDV que ele está ocupando.
4. Existem duas vias, a norte (superior) e a sul (inferior).

Alguns pontos importantes sobre a movimentação dos trens podem ser observados:

1. Inicialmente, o trem de número '1' está parado na estação VDF no topo esquerdo da figura (figura 9-a). Este trem está realizando uma manobra de retorno após ter terminado uma viagem pela via sul. Esta manobra se inicia na figura 9-a e termina na figura 9-i, podendo-se notar a mudança no estado da máquina de chave pela qual o trem passa. Nas figuras 9-i e 9-j o trem '1' se encontra parado na via norte da estação VDF. Na figura 9-j ele requisita duas rotas em seqüência, visto que a próxima rota (definição no item 3.7) entre a estação VDF e a estação JTI é muito curta (só um CDV). Na figura 9-k o trem recebe a permissão (o sinaleiro muda de estado) e reinicia sua movimentação pela via norte.
2. trem de número '3' que na 'figura 9-b' está partindo da estação NMA, requisita duas rotas ao mesmo tempo. O mesmo trem '3' faz a mesma coisa quando chega em ALE pela via. É importante notar que requisitar duas rotas ao mesmo tempo significa estender ao máximo a movimentação antes de precisar parar, o que otimiza a movimentação.
3. Finalmente, pode-se observar os diferentes tamanhos de rotas que podem existir e suas variações pelo posicionamento das máquinas de chave. O fato de um trem ocupar apenas um CDV de uma rota faz com que ela não esteja disponível para nenhum outro trem, ou seja, travada.

6 Conclusão

A utilização do software de controle baseado em Programação Genética para gerenciar Trens Urbanos permitiu um aumento do desempenho na movimentação dos trens. A comparação foi feita com os sistemas utilizados atualmente, que usam um sistema de regulação que sabe lidar apenas com situações previstas durante o seu desenvolvimento. Nestes sistemas, uma ocorrência fora do comum deve ser solucionada por um operador, o que gera um pequeno atraso na tomada de decisão e alocação das rotas. O sistema proposto implementa a mesma segurança com a melhora da eficiência na resolução destes casos. Com isso são ganhos de alguns segundos (o até, em casos extremos, alguns minutos) a cada ocorrência especial.

A simulação da via foi simplificada por envolver muitos parâmetros, mas isso não impede que estes sejam adicionados posteriormente ao simulador, melhorando a simulação do ambiente e dos erros possíveis, tornando mais rápido o desenvolvimento dos indivíduos.

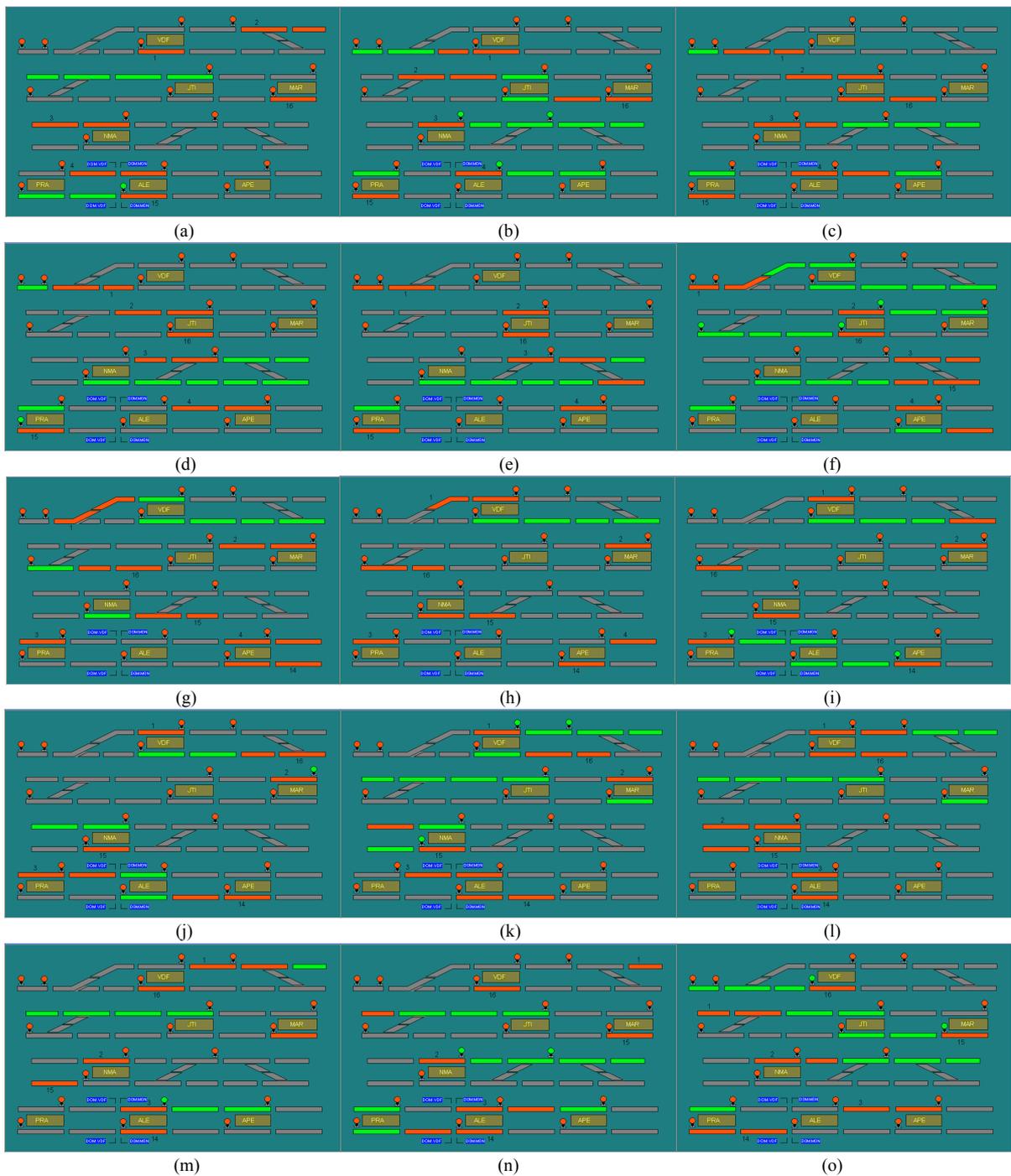


Figura 9. Exemplo de movimentações na via.

Da forma como foram evoluídos os indivíduos, todos os erros que foram simulados tiveram resposta coerente, pois foram previstos na função de *fitness*.

Como continuação desse trabalho estão previstos: inclusão de mais parâmetros no simulador, permitindo simular com maior exatidão o funcionamento real de uma via; a inclusão de novas funções e terminais nas árvores dos indivíduos para lidar com essas novas situações; e futuramente o controle total e a simulação completa da via.

Referências Bibliográficas

- KOZA, J. **Genetic programming: on the programming of computers by means of natural selection**. Cambridge, MA, MIT Press, 1992
- FrontVue Graphics Animation Engine provided by ARC Informatique, all rights reserved, Basic Scripting Engine provided by Cypress Software Inc., all rights reserved.