MODELAGEM E OTIMIZAÇÃO DO PROBLEMA DAS PONTES ROLANTES USANDO REDES DE PETRI E ESTRATÉGIAS EVOLUTIVAS

Clarimundo Machado Moraes Júnior[1]

[1]Departamento de Ciências da Computação, Universidade Estadual de Montes Claros, Av. Dr. Rui Braga s/nº, Campus Universitário, Vila Mauricéia, CEP 39401-089, Minas Gerais
cmm1jr@yahoo.com.br

Resumo – Em sistemas industriais que envolvem normalmente a automação do transporte da matéria-prima para os diversos tratamentos, um dos desafios é o escalonamento das tarefas sem que haja conflitos. Quando os recursos utilizados para o transporte da matéria-prima são solicitados ao mesmo tempo por mais de um agente, deve-se tomar um maior cuidado na política de controle desses recursos. Uma técnica que tem sido sugerida para contribuir na obtenção de um escalonamento admissível dos recursos é modelar o sistema através das redes de Petri que através de suas propriedades permitem acompanhar a evolução dele. Uma vez que se possui a modelagem do sistema, é particularmente interessante obter o escalonamento otimizado das alocações dos recursos e para isto sugere-se nesse trabalho o uso das estratégias evolutivas. Este artigo propõe ainda como estudo de caso o problema das pontes rolantes.

Palavras-chave – Escalonamento, Otimização, Redes de Petri, Estratégias Evolutivas

1. Introdução

Os sistemas flexíveis de produção modernos possuem características como paralelismo de tarefas, alocação de recursos, políticas Kanbans, políticas cíclicas e sincronização que fazem deles grandes desafios aos pesquisadores na área de otimização.

O primeiro obstáculo é modelar esses sistemas considerando suas especificidades e restrições, culminando num modelo propício ao processo de otimização. Modelar tais sistemas requer um cuidado especial no que tange à formalização de todas as suas restrições físicas. Deixar de tratar alguma restrição não impede que haja otimização das tarefas mas, certamente trará uma sequência inconsistente delas.

Uma das ferramentas indicadas para modelar sistemas com alto grau de flexibilidade, são as redes de Petri (PN). As PN são ferramentas matemáticas e gráficas que permitem a obtenção de modelos visuais, compactos e dinâmicos dos sistemas garantindo propriedades estruturais fundamentais (vivacidade, reiniciabilidade, limitabilidade e invariantes) [1]. Um grande diferencial das PN é que elas permitem modelar, analisar, controlar e supervisionar o sistema usando o mesmo padrão de modelagem.

Pelo fato das PN serem gráficas, modelarem o sistema seguindo a estrutura física dele e de permitirem acompanhar a dinâmica dele, faz com que elas sejam interessantes para visualização e entendimento de todo o funcionamento dos seus processos.

Há uma grande variedade de PN e cada uma com particularidades que se adaptam a sistemas específicos, veja [1]. Seguem algumas delas: PN interpretadas, PN associadas ao tempo, PN estocásticas e PN de alto nível.

A modelagem por equações aliada a uma PN para auxiliar o entendimento do sistema trará uma maior confiabilidade e segurança ao especialista.

A otimização de processos envolve técnicas que vão desde metodologias determinísticas às chamadas técnicas probabilísticas. Dentro desse segundo grupo existem as estratégias evolutivas (EE) que possuem o mesmo princípio básico dos algoritmos genéticos, porém segundo [2], permitem a evolução de seus parâmetros à medida que o processamento continua.

O tratamento dos indivíduos nas estratégias evolutivas é vetorial, sendo que cada um deles é composto de um vetor de variáveis, de um vetor de desvios padrões e de um vetor de ângulos de rotação. Há 3 operações básicas nas EE: seleção, recombinação e mutação. O princípio básico das EE é gerar uma população de indivíduos pais inicialmente, selecionar os pais, recombiná-los gerando os filhos, mutar os parâmetros e, finalmente, mutar os filhos gerados usando os novos parâmetros segundo [3]. Será proposto neste trabalho o uso das EE para otimizar o problema das Pontes Rolantes [4].

O problema das Pontes Rolantes é essencialmente um problema que envolve alocação de recursos e paralelismo. Na otimização o objetivo será obter o menor tempo total, dado um lote de ordens de serviços (OS). Isto é importante devido ao fato das pontes só poderem ser utilizadas por uma única ordem de serviço em cada instante e pelo fato de que a movimentação delas depende dos seus posicionamentos. Serão consideradas 3 Pontes Rolantes (PRA, PRB e PRC) e 4 locais de tratamento (convertedor 1 e 2, forno panela, desgaseificação, lingotamento).

2. O Sistema

O sistema proposto em [5] considera um conjunto de 498 OS, sendo que cada uma é tratada com um formato específico. Cada OS aloca um panela e no final gera um certo tipo de aço. O tipo de aço determina uma sequência específica de trechos que devem ser seguidos. Em termos de recursos serão considerados 4 tipos: a origem definida por dois convertedores (fornos que transformam a matéria prima em aço), tratamento 1 definido pelo forno panela (forno utilizado para melhorar a composição do aço), tratamento 2 definido pela desgaseificação a vácuo (retira os gases provenientes dos aquecimentos) e o fim definido pelos lingotamentos convencional e contínuo (processo de solidificação do aço). Na figura 1 é mostrada a estrutura física do sistema, onde são detalhados os tempos de transferência necessários em cada trecho.

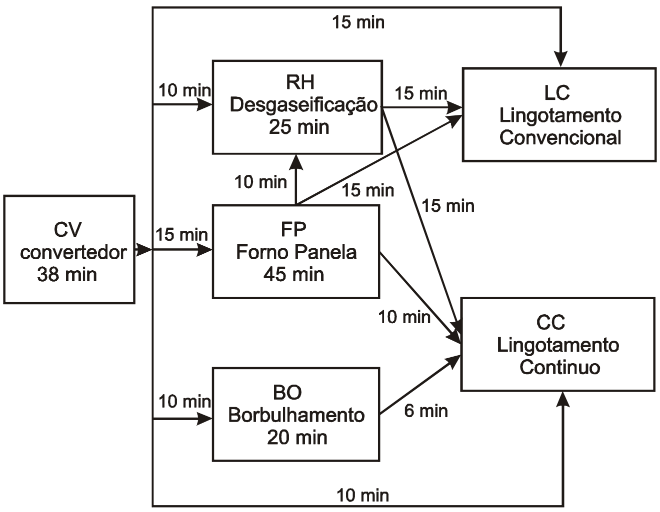


Figura 1 – Estrutura do Sistema

Apesar de aparecer o tratamento de borbulhamento no sistema, ele não exige o uso de Pontes Rolantes para transportar as panelas de chegada e saída nele.

O sistema considera que as 3 Pontes Rolantes possuem trechos de deslocamentos bem específicos. Considere que o CV é o local de início de uma OS, que RH e FP é o local intermediário de uma OS e LC ou CC é o fim de uma OS. A PRC só pode ser usada do local intermediário para o fim de uma OS, a PRB e PRA podem ser usadas em qualquer localidade, conforme mostrado na figura 2. As Pontes Rolantes nunca se cruzam e os tempos de deslocamentos delas quando estão vazias é a metade do tempo gasto se estivessem transportando uma panela.

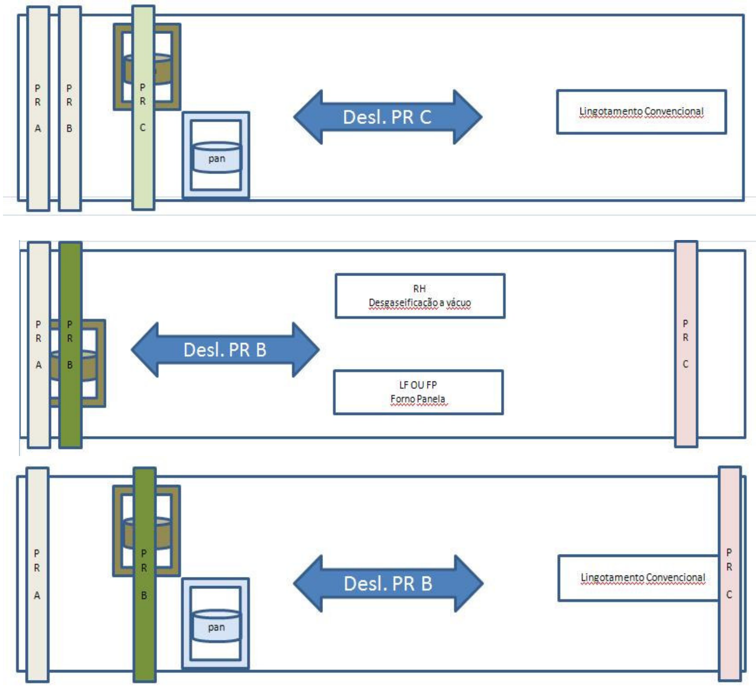


Figura 2 – Deslocamento das Pontes Rolantes

3. A Modelagem

Uma vez especificado o sistema o próximo passo é modelá-lo considerando todas as suas restrições. Por se tratar de um problema essencialmente de execuções paralelas e de alocação de recursos, pode ser particularmente interessante modelá-lo usando PN. Como já foi dito há diversos tipos de PN, porém, escolheu-se o tipo t-temporizada [6] por oferecer recursos de modelagem que tratem as restrições do sistema proposto. Não houve um estudo para se saber se outro tipo de PN é mais indicado para o estudo de caso deste trabalho.

Na figura 3 é mostrado a modelagem referente a uma OS que utiliza os trechos: CV – FP – CC. Pode-se verificar que o modelo prevê a restrição de alocação dos recursos PRB e PRC. Além disso os lugares K1, K2, K3 e K4 modelam a restrição de que em cada trecho é possível o movimento de apenas uma Ponte Rolante por vez.

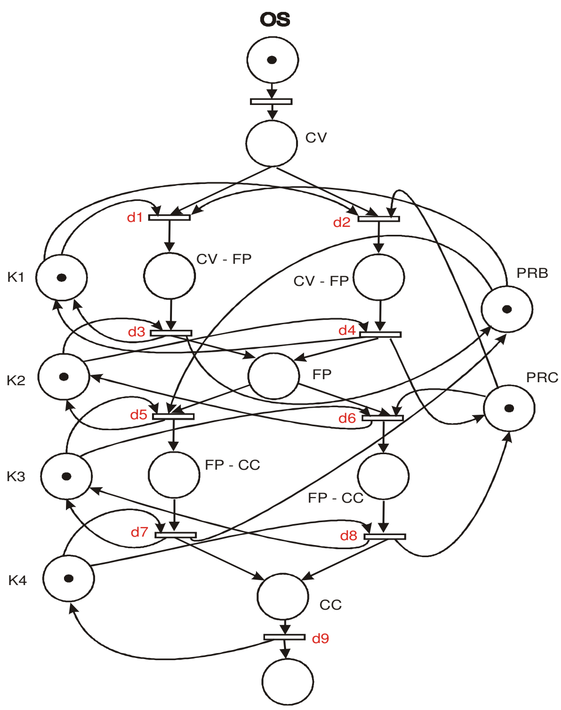


Figura 3 – Modelo de PN t-temporizada

No modelo de PN utilizado as temporizações indicam a duração temporal entre um estado e outro. Por exemplo, a temporização d1 indicará o tempo gasto entre o início do estado CV, fim do estado CV e início do estado CV-FP . Observe que na primeira transição não há temporização e isso indica que uma vez identificada a OS, imediatamente o estado CV será iniciado e, conseqüentemente, a ficha passará para esse estado no modelo. Lembre que a presença de uma ficha num estado indica que ele está ocorrendo no momento atual.

Outro ponto relevante no modelo é o fato de que a seqüência de temporizações d1, d3, d5, d7 e d9 indica a OS utilizando a PRB para transporte e a seqüencia de temporizações d2, d4, d6, d8 e d9 indica a OS utilizando a PRC para transporte.

Para o trabalho proposto, a modelagem de PN gerada no sistema auxilia no seu entendimento, mas não permite otimizá-lo, e para isso deve-se modelá-lo através de equações. Vale ressaltar que a metodologia apresentada vale para todo sistema que seja a eventos discretos.

3.1 Equacionamento do Sistema

A equação (1) do sistema tem como foco o tempo total gasto por todas as OS, considerando o fato de que as OS podem ter suas execuções realizadas em paralelo umas com as outras, integral ou parcialmente. Desta forma tem-se:

$$T_{total} = \left(\bigcup_k T_k \right) - \left(\bigcap_k T_k \right) \tag{1}$$

$$\text{com , } T_k = \sum_i (t_{trans\ ki}) + \frac{(N-k+1)}{N} * t_{esp\ ki} + \sum_j (t_{trat\ kj})$$

Sendo que:

- $t_{trans\ ki}$ é o tempo de transporte no i-ésimo trecho da k-ésima OS
- $t_{trat\ kj}$ é o tempo de tratamento no j-ésimo recurso da k-ésima OS
- $t_{esp\ ki}$ é o tempo de espera no i-ésimo trecho da
- k-ésima OS
- $i \in \{ 1,2,3 \}$
- $j \in \{ 1,2,3, \dots ,11 \}$
- $k \in \{ 1,2,3, \dots ,10 \}$
- $\cup$ é a união dos intervalos de tempo das OS
- $\cup$ é a interseção dos intervalos de tempo das OS
- $N = 10$ (lote de OS)

Vale ressaltar que T_{total} representa um intervalo de tempo total gasto por todas as OS e que a subtração apresentada sempre dará positiva uma vez que a união dos intervalos de tempo é maior que a interseção dos mesmos.

3.2 Restrições do Sistema

São consideradas as seguintes restrições no sistema:

1. $\sum_p Pa_{tlp} \leq 1$, no tempo t a panela l só pode ser transportada por uma ponte rolante.
2. $t \geq 0$, tempo positivo.
3. $\sum_i Pr_{tpi} \leq 1$, no tempo t a ponte p só pode ser usada em um único trecho.
4. $Pa_{tlp} \in \{ 0,1 \}$
5. $Pr_{tpi} \in \{ 0,1 \}$

3.3 Equação de Otimização do Sistema

A otimização do sistema é um problema de minimização do tempo total de processamento das OS cujo equacionamento é mostrado em (2):

$$Min \quad T_{total}$$

$$s.a. \qquad (2)$$

$$\sum_p Pa_{tlp} \leq 1 \qquad \sum_i Pr_{tpi} \leq 1 \qquad t \geq 0$$

com, $l \in \{1,2,3, \ldots , 28\}$; $p \in \{1,2,3\}$; $i \in \{1,2,3\}$

4. Estratégias Evolutivas

O princípio básico das EE é evoluir os parâmetros probabilísticos junto com a população de indivíduos, diferentemente dos algoritmos genéticos que mantém as probabilidades *pc* e *pm* fixos em todas as gerações.

A estrutura de um indivíduo é vetorial e compõe-se de um vetor de variáveis, um vetor de ângulos de rotação e um vetor de desvios padrões conforme equação (3):

$$\boldsymbol{a} = (\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\sigma}) \qquad (3)$$

Assim como nos algoritmos genéticos, há essencialmente 3 operações genéticas nas EE: a seleção, a mutação e a recombinação, conforme [3]. São considerados dois métodos de seleção: seleção (μ , λ) e seleção $(\mu + \lambda)$. Na primeira seleção escolhe-se os melhores indivíduos da população de filhos e na segunda seleção escolhe-se os melhores indivíduos da população de pais e filhos.

A mutação ocorre tanto para os indivíduos quanto para os parâmetros ($\boldsymbol{\alpha}$ e $\boldsymbol{\sigma}$). Na verdade o que se faz é mutar os parâmetros, mutar as variáveis usando os respectivos parâmetros já mutados e assim gerar novos indivíduos na forma mostrada em (4):

$$\sigma'_i = \sigma_i * exp(\tau' * N(0,1) + \tau * N_i(0,1))$$

$$\alpha'_j = \alpha_j + \beta * N_j(0,1) \qquad (4)$$

$$x' = x + T^T(\alpha') * S^T * (\sigma') * N(0,I)$$

com, $\forall\, i \in \{1,\ldots,n\}$ e $\forall\, j \in \{1,\ldots, n*(n-1) / 2\}$

Vale ressaltar que:

- S : matriz diagonal com valores positivos dos desvios padrões;
- T : matriz ortogonal que contém o produto de $l(l-1)/2$ matrizes rotacionais R_{ij} com ângulos $\alpha_k \in (0,2\pi]$;
- $\beta = 0.0873$ rad;

Uma matriz rotacional R_{ij} é uma matriz identidade com as quatro entradas : $r_{ii} = r_{jj} = \cos \alpha$ e $r_{ij} = - r_{ji} = - \text{sen}\ \tilde{\alpha}$. A recombinação é na verdade uma forma de obter material genético de mais de um pai gerando assim filhos recombinados. Neste trabalho será considerada a recombinação panmítica onde um pai é escolhido arbitrariamente e mantido fixo enquanto, para cada componente de seus vetores, o segundo pai é escolhido arbitrariamente em toda a população. Esta técnica permite que num caso extremo um filho tenha parte do material genético de todos os pais da população. Na figura 4 é mostrado o algoritmo das EE.

```
Algoritmo Estratégias Evolutivas (μ+λ) ou (μ,λ)
inicialize μ indivíduos do tipo: a = (x, α, σ)
avalie todos os indivíduos
t ← 1
enquanto teste de convergência não é satisfeito faça
    repita λ vezes,
        selecione (ρ > 2) pais aleatoriamente
        recombine os indivíduos para gerar um filho
        mute  o filhe gerado
        avalie o filho
    fim repita
    t ← t + 1
    selecione os μ melhores indivíduos de λ ou μ+λ
fim enquanto
fim procedimento
```

Figura 4 – Algoritmo das Estratégias Evolutivas

5. Implementação

Na implementação considerou-se o sistema mostrado na figura 1, de forma que há 4 locais de tratamento onde o uso das Pontes Rolantes para fazer o transporte de um local ao outro se faz necessário. Assim ficou definida a existência de 4 trechos bidirecionais:

1. CV – FP
2. FP – RH
3. FP – LC (CC) ou RH – LC (CC)
4. CV – LC (CC)

Em termos de estruturas de dados há duas que tratam especificamente do escalonamento temporal das Pontes Rolantes. A primeira estrutura de dados é mostrada na figura 5. Como há 4 possibilidades de deslocamento das Pontes Rolantes, então, gerou-se uma matriz composta de 4 colunas associadas, respectivamente, aos 4 trechos citados anteriormente.

CV – FP					CV – LC (CC)			
esp	tran	ti	tf	------	esp	tran	ti	tf
Coluna 1					**Coluna 4**			

Figura 5 – Estrutura de uma linha da matriz de escalonamento

O campo **esp** representa o tempo de espera que possa acontecer antes que uma dada Ponte Rolante seja usada para deslocar ao referido trecho. O campo **tran** é o tempo de transporte associado ao trecho. O campo **ti** é o tempo (data) de início do tratamento e **tf** é o tempo (data) de fim do referido tratamento. Observe que os dois primeiros campos são durações de tempo e os dois últimos campos são datas. O início da primeira ordem de serviço ocorre na data 0.

A segunda estrutura de dados é mostrada na figura 6. Nela representa-se o local e a data de disponibilidade de cada Ponte Rolante durante o escalonamento temporal. Entende-se por local neste caso, o local de início de uma OS (CV), o local intermediário (FP ou RH) e o local de fim de uma OS (LC ou CC). Vale ressaltar que, a cada finalização de uma OS, há um status desta estrutura cujos valores indicam apenas a disponibilização corrente das Pontes Rolantes. Desta forma esta variável não é matricial e sim simples. A idéia é que a cada nova OS, saiba-se o posicionamento de cada Ponte e a partir de quando cada uma passou a estar liberada na referida posição.

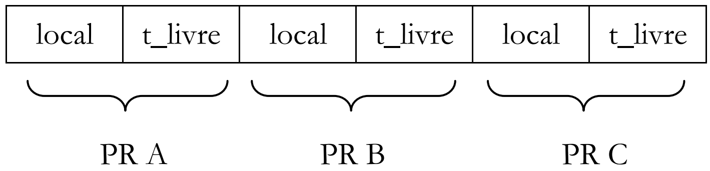


Figura 6 – Estrutura da variável de disponibilização das Pontes

Foi usada a representação real para os indivíduos. Cada indivíduo é composto pelo escalonamento das Pontes Rolantes relacionado a um lote de 10 OS. Cada OS é completada considerando-se uma combinação dos 4 trechos citados anteriormente. Desta forma são 40 indicações, veja figura 7, de uso das Pontes Rolantes nos transportes de cada trecho considerado.

2	3	0	3	1	-----	3
1	2	3	4	5		40

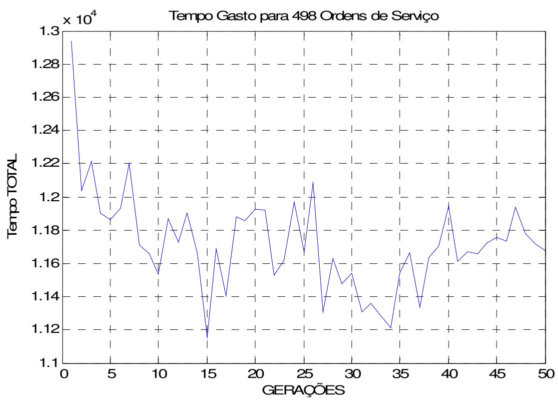
Figura 7 – Exemplo de um indivíduo

O valor 0 é dado quando um trecho numa dada OS não existir, daí atribui-se o valor zero indicando a inexistência de Ponte Rolante. As Pontes Rolantes A, B e C recebem os valores 1, 2 e 3, respectivamente. Conforme já foi dito, além dessas 40 variáveis cada indivíduo constitui-se de 40 valores de desvios padrões e de 780 valores (1 $*$ (1 - 1) / 2) de ângulos de rotação. Desta forma o comprimento de cada indivíduo vale 40 + 40 + 780 = 860.

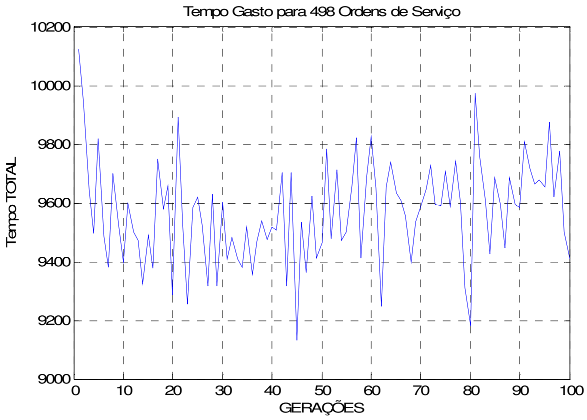
6. Resultados Obtidos

Os gráficos apresentados são de Número de gerações x tempo de execução das OS (min). Em termos de plataforma computacional usou-se uma máquina core 2DUO, com 1Gb de memória RAM, Windows XP, cache 2Mb, barramento de 533 Mhz.

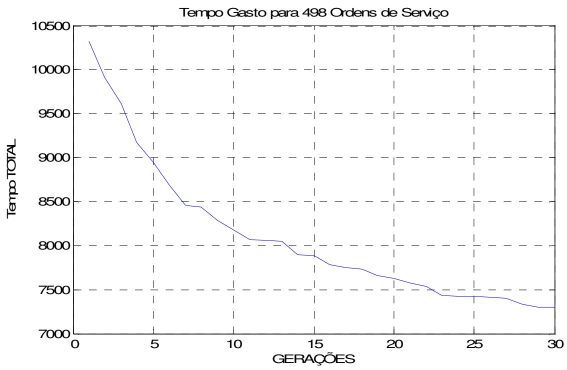
Simulação 1:
SELEÇÃO (μ , λ) RECOMBINAÇÃO: panmítica
Núm.máx. de gerações e indivíduos = 50 e 10 respectivamente
Tempo otimizado (498 OS) = 11677 min
Tempo de processamento = 1,467 horas



Simulação 2:
SELEÇÃO(μ , λ) RECOMBINAÇÃO: panmítica
Núm.máx. de gerações e indivíduos = 100 e 80 respectivamente
Tempo otimizado (498 OS) = 9408 min
Tempo de processamento = 22,73 horas



Simulação 3:
SELEÇÃO($\mu + \lambda$)   RECOMBINAÇÃO: panmítica
Núm.máx. de gerações e indivíduos = 30 e 50 respectivamente
Tempo otimizado (498 OS) = 7303 min
Tempo de processamento = 4,39 horas



6. Conclusões

A modelagem do problema usando uma PN t-temporizada sugere uma abordagem interessante, principalmente pelo fato de que no próprio modelo consegue-se visualizar a dinâmica do problema acompanhando-se a evolução das fichas. Uma proposta é criar um escalonador de eventos que utilize uma PN t-temporizada para gerar escalonamentos viáveis (sem violações de restrições) e otimizar esses escalonamentos gerados pela PN usando EE. A idéia é, posteriormente, evoluir a PN seguindo uma temporização que permite uma evolução consistente e otimizada. Esse acompanhamento da evolução é particularmente interessante quando se está no nível de supervisão em tempo real do sistema. Nas PN consegue-se modelagem, análise e controle.

Se for realizada a modelagem completa do sistema proposto, com todos os tipos de trechos e alocações de Pontes Rolantes, a modelagem ficará bem complexa. Propõe-se o uso de uma PN Colorida associada a temporizações onde se consegue compactar o modelo global através do uso de cores nas fichas e, conseqüentemente, pode-se agregar partes do modelo em que há similaridade estrutural.

Comparando as simulações 1 e 2 verifica-se que aumentar mais o número de indivíduos do que o número de gerações surte um melhor efeito na otimização proposta.

Na seleção (μ , λ) devido ao fato de se esquecer soluções de gerações passadas, verificou-se que de uma geração para outra a solução otimizada ora melhora ora piora.

Como o referido problema possui uma fitness (função de aptidão) que não varia (sem ruídos), a seleção ($\mu + \lambda$) se mostrou mais eficiente. Nela as melhores soluções de cada geração são repassadas para as próximas gerações. Observa-se que a convergência inicial é mais rápida do que na seleção (μ , λ) e valores mais otimizados são obtidos na ($\mu + \lambda$).

A simulação 3 proposta com o número máximo de 30 gerações mostra que com apenas 30 gerações o tempo total da OS já se encontra abaixo dos encontrados nas simulações 1 e 2 que utilizaram um número maior de gerações.

Um outro ponto a ser considerado é que a seleção ($\mu + \lambda$) é melhor adaptada em problemas que não são multimodais, que é o caso do problema proposto.

Apesar de convergir para soluções ótimas, as EE se mostram relativamente "pesadas" em termos de processamento se comparadas a outras técnicas quando o problema envolve um número elevado de variáveis. No problema proposto cada indivíduo é representado por uma sequência de valores reais de 40 variáveis seguidas de 40 valores de desvios padrões e de 780 valores de ângulos rotacionais totalizando um comprimento de 860 valores reais. Pensando que um número razoável de indivíduos mostrado nas simulações é de 50, então, a população do referido problema é uma matriz (50 x 860).

Apesar do relativo peso no processamento, para problemas de escalonamentos de tarefas onde se tem períodos de tempo suficientes para se obter a otimização do mesmo, esta não é uma característica que vá comprometer o funcionamento real do sistema.

Uma sugestão é utilizar um algoritmo genético para o problema das Pontes Rolantes e comparar seus resultados com os resultados obtidos nas EE.

Referências:

[1] J. Cardoso, R. Valette, Redes de Petri, UFSC, 1ª Ed, 1997.

[2] L. S. Coelho, A. A. R. Coelho, Algoritmos Evolutivos em Identificação e Controle de Processos : uma visão integrada e perspectivas. SBA Controle & Automação, vol 10, nº 01, 1999.

[3] L. S. Coelho, F. V. Zuben, Estratégias Evolutivas – (EE's). Tópico 10, IA707 , DCA/FEEC/Unicamp, Campinas, 2002. ftp://ftp.dca.fee.unicamp.br/pub/docs/vonzuben/ia707_02/topico10_02.pdf

[4] A. Tamasauskas, Metodologia do Projeto Básico de Equipamento de Manuseio e Transporte de Cargas - Ponte Rolante - Aplicação Não-Siderúrgica, Dissertação de Mestrado, Departamento de Engenharia Mecânica, Escola Politécnica da USP, 2000.

[5] J. A. Vasconcelos, Computação Evolucionária, 2º semestre, CPDEE/UFMG, Belo Horizonte, 2008. http://www.cpdee.ufmg.br/~joao/CE/DefinicaoTrabalhoFinal/PonteRolante/ ProblemaPonteRolante.xls

[6] M. Marton, Um Analisador Automático de Redes de Petri Temporizadas, Dissertação de Mestrado, DTIFEE/IUNICAMP, 107pp,1989.