

OTIMIZAÇÃO DE SEQUENCIAMENTO DE VEÍCULOS NUMA LINHA DE MONTAGEM UTILIZANDO ALGORITMOS GENÉTICOS

FÁBIO C.M ALINOWSKI

RENAULT do Brasil Automóveis S.A.
BR277, km73, Estrada da Roseira
80090-900 – São José do Pinhais -PR
E-mails: .familinowski@uol.com.br

FLÁVIO NEVES JR., HEITOR S.L OPES

CPGEI, CEFET -PR
Av. Sete de Setembro, 3165, 80230 -901, Curitiba, PR, BRASIL
E-mails: {neves, hslopes}@cpgei.cefetpr.br

Resumo— Neste artigo é descrito o uso de algoritmos genéticos para o problema de otimização de uma linha de montagem de veículos. O problema é descrito conforme os pedidos de produção e as capacidades da linha de produção. Procura-se a ordem na qual os veículos de diferentes tipos de opção serão produzidos, considerando as restrições operacionais das estações de trabalho e a relação de cada opção. Foram realizadas quatro simulações, nas quais o AG demonstrou excelente performance, comparado com uma ferramenta de otimização industrial.

Abstract— In this work it is described the use of Genetic Algorithms for the car assembly line scheduling optimization problem. The problem is defined in terms of the variables related to the assembly line and requested production. It can be summarized as finding the order in which vehicles of different types with different optional accessories are assembled, respecting the operational restrictions of the workstations imposed by each option type. Four simulations were done in which the GA approach has demonstrated excellent performance, even when compared with an industrial optimization system.

Keywords— Genetic Algorithms Scheduling, Production Planning .

1 Introdução

Atualmente, as montadoras de veículos devem estar preparadas para montar veículos personalizados tão rapidamente quanto possível e com o menor custo. Isto ocorre devido à crescente diversidade de tipos de veículos, onde o cliente tem liberdade de escolher dentre um vasto gama de opções e acessórios que lhe sejam mais adequados em termos de conforto e de custo.

Para reduzir custos na linha de montagem, os veículos em montados devem ter um seqüenciamento otimizado. Assim como ocorre em todas as linhas de montagem, os recursos relacionados aos processos tecnológicos são restritos (tempo, mão-de-obra, estações de trabalho, etc.), e não é interessante aumentar os custos de produção contratando mais pessoal ou investindo em equipamentos suplementares. Estas restrições impõem limites severos ao sistema de produção para que a qualidade final seja garantida.

Um algoritmo genético –AG (Goldberg, 1989) é um método de busca de otimização que tem sido largamente utilizado em problemas de Engenharia e da Ciência da Computação (Lopes, 1999). Neste trabalho, um AG foi utilizado para a otimização da seqüência na qual os veículos são colocados na linha de montagem, considerando o tipo de veículo e os seus opcionais. Para um problema semelhante (Warwick & Tsang, 1996), AG se mostrou eficiente e isto motivou a sua utilização no presente

trabalho, embora existam muitas diferenças nas duas aplicações.

2 Metodologia

2.1 Descrição do Algoritmo

O primeiro passo na aplicação da metodologia é a completa definição do problema em termos das relações básicas entre as variáveis pertinentes à linha de montagem de veículos. O estado atual da fábrica e a quantidade de veículos solicitada à produção devem ser considerados. Um plano diário de produção determina o número de tipos de veículos a serem produzidos (com base na quantidade faturada às concessionárias). O tipo de veículo é determinado pelos seus acessórios opcionais. Usando estas informações, através da equação 1 se pode computar o número de veículos que podem ser produzidos.

$$N = \sum_{j=1}^k pr[j] \quad [1]$$

onde:

k = tipo de veículo (0...7) produzido nesta fábrica;
 $pr[j]$ = demanda de produção do veículo tipo j ;
 N = número total de veículos a serem produzidos.

Para associar os possíveis opcionais aos diferentes tipos de veículos, um agrupamento $O[m,j]$ é usado, onde m representa o opcional e j o tipo de veículo. Por exemplo, $O[m,j]=1$ significa que o veículo tipo j tem o opcional m , caso contrário, $O[m,j]=0$ a quele tipo de veículo não possui o opcional.

Em seguida, para cada opcional, suas restrições correspondentes são definidas. Restrições são representadas pelos valores das variáveis $pm:qm$ que significam que para um dado opcional m , pm veículos podem ser montados com este opcional, respeitando um intervalo mínimo de outros qm veículos sem o opcional. Usando tais parâmetros, as seguintes relações podem ser definidas:

1. $Onum$: o número total de um dado opcional solicitado no plano de produção diário:

$$Onum(m) = \sum_{j=1}^k (pr[j] * O[m, j]) \quad [2]$$

2. $Omax$: o número máximo de cada opcional possível no plano de produção, levando em consideração suas respectivas restrições:

$$Omax(m) = \left(\frac{pm}{qm} \right) * N \quad [3]$$

3. Um : o grau de utilização dos recursos para a estação de trabalho, para cada opcional:

$$Um(m) = \frac{Onum(m)}{Omax(m)} \quad [4]$$

Uma condição necessária, porém não suficiente, para que um determinado seqüência de produção seja factível é que todas as restrições de capacidade para cada opcional sejam satisfeitas, resultando em um valor médio para o grau de utilização dos recursos $\mu < 1$, isto é:

$$\mu = \frac{\sum_{m=1}^n Um(m)}{n} \quad [5]$$

onde:

n = número de opcionais;

$Um(m)$ = nível de utilização dos recursos para o opcional m .

Para um plano de produção diário típico usando dados reais, a tabela 1 mostra os opcionais, as restrições e os parâmetros anteriormente definidos para a linha de montagem. Nesta tabela, os tipos de opcionais são: (1) freios ABS, (2) banco sem couro, (3) motor Diesel, (4) ar condicionado, (5) conjunto elétrico, (6) motor 1.0 16V e (7) *airbag*.

Tabela 1 – Exemplo de um plano de produção diário típico.

Opcionais	Tipos de veículos							Restrições do processo				
	0	1	2	3	4	5	6	7	$p:q$	$Omax$	$Onum$	Um
1	0	1	0	0	0	0	1	0	1:3	27	16	0.60
2	0	0	1	0	0	0	0	0	1:5	20	10	0.50
3	0	0	0	1	0	0	0	0	1:4	27	13	0.48
4	0	1	0	0	0	1	1	0	2:5	53	20	0.37
5	0	0	1	0	1	0	0	0	1:2	40	27	0.67
6	0	0	0	0	0	0	1	0	1:4	27	11	0.41
7	0	0	0	0	0	0	0	1	1:1	80	20	0.25
Produção requerida	0	5	1	1	1	4	1	2	$\mu=0.5$			
			0	3	7		1	0				

2.2 Função de penalidade

O passo seguinte necessário para a aplicação da metodologia é a definição de uma função de penalidade (P). Esta função penaliza um veículo que viola alguma das restrições do processo. Esta violação ocorre quando um dado veículo da seqüência requer o mesmo opcional (m) sem respeitar o intervalo requerido (qm), excedendo, assim, a capacidade da estação de trabalho (pm). Assim, para cada opção (m), um valor correspondente de penalidade para o veículo na i -ésima posição é calculado através da equação 6:

$$custo(i, m) = P \left(m, \left[O[m, S_i] * \sum_{j=i+1}^{i+(qm-1) \leq N} O[m, S_j] \right] \right) \quad [6]$$

onde:

i = posição do veículo na seqüência de montagem ($i=1 \dots 80$);

S_i = o i -ésimo tipo de veículo no plano de produção;

$O[m, S_i] = 1$ quando o tipo de veículo (S_i) requer o opcional (m), e 0 caso contrário.

Se um intervalo de proximidade menor para cada opcional for definido (menor do que qm), é possível melhorar o posicionamento na seqüência dos veículos penalizados na linha de montagem. Isto pode ser feito mantendo ao menos um veículo entre os veículos com os mesmos opcionais. Para cada veículo que não respeite este intervalo mínimo, uma penalidade adicional $F[m]$ aumentará o valor de penalização para seus opcionais (equação 7), de acordo com a equação 7.

$$Tcusto(i, m) = custo(i, m) + F[m] \quad [7]$$

Assim sendo, o custo total do plano de produção dos (N) veículos a serem produzidos pode ser definido completamente pela equação 8:

$$Scusto = \sum_{i=1}^N \sum_{m=1}^n Tcusto(i, m) \quad (8)$$

3 Aplicação do algoritmo genético

3.1 Codificação

O comprimento do cromossomo é definido de acordo com o número de veículos a serem produzidos (diariamente) na linha de montagem. Para uma situação real, este número é de 290 veículos. Os genes no cromossomo são representados como inteiros, no intervalo [0..7], indicando o tipo de veículo. A posição do gene no cromossomo representa a ordem na qual o veículo correspondente deverá ser produzido, isto é, o primeiro gene representa o primeiro veículo, o segundo gene, o segundo e assim sucessivamente até o 290^{ésimo} gene, o qual será o último veículo a ser produzido no dia.

3.2 Método de seleção

Nos experimentos computacionais realizados, o método de seleção por torneio (*tournament selection*) foi utilizado em associação com uma estratégia elitista. O método de seleção por torneio é decidido de maneira melhor do que o método clássico da "roleta" (proporcional ao *fitness*) dado que o mesmo proporciona uma pressão seletiva menor evitando, assim, uma convergência prematura. Utilizando o elitismo, a melhor solução de uma geração é sempre copiada para a próxima geração garantindo um valor estável ou monotonicamente crescente da função de *fitness* (do melhor indivíduo a cada geração) até a convergência.

3.3 Função objetivo

A função objetivo é determinada pelo número máximo de penalidades que podem estar contidas em uma determinada sequência de veículos a ser produzida. Para o presente problema, o cálculo desta função é feito com a equação 9:

$$P_{max} = N * m * (q_{max} - 1) + \sum_{j=1}^k |pr[j]_S - pr[j]_P| \quad [9]$$

onde:

N = número total de veículos a serem produzidos;

m = número total de opções;

k = tipo de veículo (0...7) produzido na fábrica;

q_{max} = valor máximo de opções para cada restrição;

$pr[j]$ = número de veículos do tipo j solicitados (S) e produzidos (P).

3.4 Função de fitness e operadores genéticos

A função de *fitness* é o que o algoritmo genético tenta otimizar (por definição, maximizar) e inclui, além da função objetivo (seção 3.3), ajustes de escala/normalizações, aplicação de penalidades, e outros ajustes que se façam necessários para avaliar o indivíduo decodificado como uma possível solução para o problema do mundo real. Dado que a função objetivo, definida pela equação 9, exibe valores maiores para planos de produção piores, a função de *fitness* é, então, definida como sendo a penalidade máxima estimada subtraída de P_{max} . Em outras palavras, a função de *fitness* simplesmente avalia quantos veículos da sequência planejada são penalizados. Desta maneira, se nenhum veículo codificado no cromossomo violar as restrições do processo, o valor ótimo obtido corresponde à máxima penalidade estimada. Não se faz necessário normalizar as penalidades, dado que todas as variáveis consideradas estão na mesma escala e unidade de medida (número de veículos).

Os dois operadores genéticos clássicos, *crossover* (recombinação) e mutação foram utilizados. O *crossover* mais simples, de um ponto, foi aplicado com probabilidade ($p_c=1$) para preservar a estrutura do cromossomo, evitando a quebra de blocos construtivos significativos e induzindo, assim, a busca local. A mutação simples de um bit foi aplicada com probabilidade ($p_m=0.07$ por bit), permitindo ao algoritmo manter uma diversidade populacional adequada ao longo das gerações, para explorar mais eficientemente o espaço de busca (busca global).

4 Simulações e resultados

Quatro simulações computacionais foram feitas para avaliar o desempenho do sistema de AG em diferentes situações. O AG desenvolvido foi baseado na versão 3.2 do software de domínio público denominado GALOPPS (Goodman, 1986).

Na primeira simulação o algoritmo foi testado numa situação extrema, onde todos os veículos do plano de produção tinham algum opcional que representasse restrições à linha de montagem, além do que o nível de utilização dos recursos foi estipulado como um valor alto ($\mu=0.7$). Como era de se esperar, foi observado que o algoritmo tendeu a favorecer veículos com menor número de restrições do que aqueles com maior número. Este comportamento diminuiu a sobrecarga das estações de trabalho.

Na segunda simulação foi incluído no plano de produção um tipo de veículo que não apresentava nenhuma restrição à linha de montagem (modelo básico), reduzindo o nível de utilização dos recursos para $\mu=0.5$. Para este problema, a solução ótima foi obtida, isto é, todas as restrições foram respeitadas.

Para a terceira simulação, o número de veículos a serem produzidos foi significativamente aumentado. Este experimento objetivou simular a seqüência ideal com razão 2:3 (veja seção 2.1). Isto é, para cada dois veículos com restrições, três outros devem ser produzidos sem restrições. Como resultado, o algoritmo conseguiu uma seqüência de veículos sem violar nenhuma restrição, mas a quantidade de veículos produzida foi menor do que a solicitada.

Na última simulação, o AG foi comparado com um sistema de planejamento industrial que é atualmente utilizado na fábrica, denominado OLGA. Até o momento, não há informações disponíveis a respeito da metodologia de otimização utilizada pelo OLGA. Dados reais de um dia normal de produção foram utilizados, onde o plano de produção apresentava diversas restrições.

Ambos os sistemas (OLGA e AG) foram comparados de acordo com sua habilidade em obter uma seqüência de produção com o menor número possível de violações. A tabela 2 mostra detalhes do plano de produção solicitado para esta simulação. Outros parâmetros do sistema foram: $N=290$, $k=8$, $m=7$, $\mu=0.33$. Para esta tabela, os opcionais foram: (1) freios ABS, (2) motor Diesel MN, (3) motor Diesel M2, (4) "Clio" tipo L65, (5) banco sem couro, (6) "Clio" básico, e (7) "Scénic" básico.

Tabela 2 – Plano de produção para um dia normal, onde 24 penalidades foram registradas.

opcionais	Tipos de veículos							Restrições do processo				
	0	1	2	3	4	5	6	7	p:q	Omax	Onum	Um
1	1	0	0	0	0	1	0	0	1:3	97	77	0.80
2	0	1	0	0	0	0	0	0	1:6	48	5	0.10
3	0	0	1	0	0	0	0	0	1:5	58	29	0.50
4	1	0	0	0	1	1	0	0	1:2	145	46	0.32
5	0	1	0	1	0	0	0	0	1:6	48	4	0.10
6	0	0	0	0	0	0	1	0	1:1	290	116	0.40
7	0	0	0	0	0	0	0	1	1:1	290	33	0.11
Produção requerida	6	5	4	18	35	1	11	33	$\mu=0.33$			
	8						1	6				

Para a quarta simulação, o AG convergiu em aproximadamente 300 gerações, encontrando uma solução melhor do que a que foi obtida pelo sistema OLGA. A tabela 3 apresenta a comparação entre o AG e o OLGA.

5 Conclusões

Foi observado que a habilidade do AG em encontrar soluções satisfatórias (planos de produção para a linha de montagem) sem penalidades diminuiu à medida que o nível de utilização dos recursos da linha de montagem (μ) aumenta. Em geral, a habilidade do AG em encontrar boas soluções não está necessariamente restrita pelo tamanho do espaço de busca (número

de veículos do plano de produção). Tal habilidade, de fato, é limitada somente pela complexidade das interações entre os opcionais contidos nos diversos tipos de veículos e pelas restrições do processo para cada opção. Além disso, uma seqüência importante do aumento do espaço de busca é o crescimento do esforço computacional (tempo de processamento) necessário para processar o algoritmo, que é um tipo de função exponencial do número de veículos (N).

Tabela 3 – Resultados comparativos entre os dois sistemas.

parâmetro	OLGA	GA
Penalidades para ABS	16	18
Penalidades para Motor Diesel tipo MN	0	0
Penalidades para Motor Diesel tipo M2	3	0
Penalidades para "Clio" L65	0	0
Penalidades para Banco sem couro	0	0
Penalidades para "Clio" básico	5	4
Penalidades para "Scénic" básico	0	0
Número total de penalidades	24	22

Baseado nos resultados preliminares aqui apresentados, foi possível verificar que o comportamento do AG foi bastante satisfatório em ambas as situações (com alto e baixo nível de restrições), atingindo um desempenho excelente. Uma vantagem importante do AG em relação ao sistema OLGA é que, no caso da inclusão de um veículo que cause penalidade no plano de produção, este veículo não será posicionado imediatamente após o outro veículo como opção opcional, o que evita a sobrecarga da estação de trabalho. Considerando os resultados obtidos, uma vez que a otimização do plano de produção não necessitar ser processada em tempo real (pois é feita uma vez a dia), a substituição do sistema atual pelo sistema de otimização baseado em AG poderá ser considerada num futuro próximo.

Referências Bibliográficas

- GOLDBERG, D.E., (1989) Genetic Algorithms in Search, Optimization and Machine Learning. Reading: Addison - Wesley.
- GOODMAN E.D. (1996) Genetic Algorithm Optimized for Portability and Parallelism System. East Lansing, Michigan State University.
- LOPES, H.S. (1999) Algoritmos genéticos em projetos de engenharia: aplicações e perspectivas futuras. In: Anais do SBAI'99 - IV. Simpósio Brasileiro de Automação Inteligente. São Paulo, 08 - 10/ setembro/1999, p.64 -74.
- WARWICK T., T SANG E.P.K., (1996) Tackling Car Sequencing Problems Using Genetic Algorithm, Evolutionary Computation, v.3, n.3, p.267 -298.