

ALGORITMOS GENÉTICOS NO PROBLEMA DE BALANCEAMENTO DE LINHAS DE MONTAGEM: UMA APLICAÇÃO REAL NUMA LINHA AUTOMOTIVA

SOLIVAN A. VALENTE, HEITOR S. LOPES, LÚCIA VALÉRIA R. DE ARRUDA

Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial

Centro Federal de Educação Tecnológica do Paraná - CPGEI

Av. 7 de setembro, 3165 – 80230-901 – Curitiba – PR – Brasil

E-mails: solivan@uol.com.br, lopeshs@yahoo.com, arruda@cpgei.cefetpr.br

Resumo— Este artigo relata a aplicação de Algoritmos Genéticos (AGs) num problema real de balanceamento de linhas de montagem, especificamente na indústria automotiva. O problema é modelado e um AG padrão é aplicado. A solução encontrada via AG é 28.5% melhor do que a solução atual adotada na linha de montagem real, e esta minimização de tempo representa uma significativa redução de custos. Este resultado indica que a utilização de AGs em problemas reais encontrados na indústria pode ser muito promissora.

Abstract— This paper reports the use of Genetic Algorithms (GAs) to solve the assembly line balancing problem in a real-world application: a car assembly facility. The problem is modeled and a standard GA is applied. The line layout solution found by GA reduces by 28.5% the total assembly time of the current line layout, which implies in a significant reduction of costs. This result suggests that the use of GAs in real-world industrial problems can be very promising.

Keywords— Genetic Algorithms, Optimization, Manufacturing, Production Line, Car Assembling.

1 Introdução

Um Algoritmo Genético (AG) é um método de busca e otimização bastante conhecido que se baseia no princípio da evolução das espécies de Darwin e em alguns fundamentos da genética (Goldberg, 1989). Do ponto de vista classificatório, o AG é uma técnica de Computação Evolucionária, a qual também engloba a Programação Genética, os Sistemas Classificadores, as Estratégias Evolutivas e a Programação Evolucionária.

A principal aplicação dos AGs é a otimização. Muitos problemas de engenharia são problemas de otimização e, por isso, durante as últimas décadas os AGs têm sido aplicados com sucesso em diversos problemas de engenharia. Na otimização o objetivo é encontrar um conjunto tal de valores para as variáveis do problema que minimize uma função de custo ou que maximize uma função de ganho. Existem diversos métodos de otimização, numéricos e algébricos, que procuram por este conjunto de valores no espaço de busca multidimensional do problema em questão. Entretanto, em problemas que apresentam não-linearidade, ruído, descontinuidade ou espaços de busca extremamente grandes estes métodos podem se tornar computacionalmente proibitivos ou mesmo não serem aplicáveis. Neste cenário de problemas multimodais que são intratáveis por métodos clássicos, os AGs mostram-se como uma alternativa simples e eficiente.

2 Algoritmos Genéticos e o Balanceamento da Linha de Montagem

Sem dúvida, um forte argumento para a utilização de AGs no problema clássico de balanceamento de uma linha de montagem industrial é a natureza combinatorial deste tipo de problema. Mesmo o balanceamento de uma linha simples pode tornar-se uma tarefa bastante difícil e demorada para um algoritmo determinístico, dada a amplitude do espaço de busca.

Neste trabalho, parte de uma planta de montagem de automóveis (um trecho da área de montagem final) é otimizada. Todos os dados contidos neste trabalho são reais, mas o nome da empresa foi propositalmente omitido. O trecho da linha de montagem considerado é composto de 10 estações de trabalho. Em cada uma delas, os operadores podem realizar tarefas em ambos os lados da linha, num total de 13 atividades a serem feitas no processo de montagem final do veículo.

Uma breve análise da topologia da linha (ver Fig. 1) mostra que o número total de configurações possíveis é dado pelo arranjo $A_{2n,p}$, isto é, $(2n!)/(2n-p)!$, onde “n” é o número de estações de trabalho e “p” é o número de atividades a serem realizadas. O trecho da linha em análise tem $n=10$ e $p=13$, o que resulta em cerca de $4,83 \cdot 10^{14}$ diferentes configurações possíveis. Se expandirmos o horizonte de análise para toda a área de montagem final desta mesma planta, as 66 atividades distribuídas nas 49 estações de trabalho aumentam o número de configurações possíveis para cerca de $3,58 \cdot 10^{118}$. Este espaço de busca é intratável por qualquer método de busca

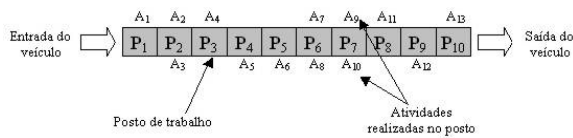


Figura 1. Representação da linha de produção sob análise.

seqüencial, em um tempo razoável, por mais recursos computacionais de que se disponha.

A otimização do tempo de trabalho em uma unidade industrial tem conseqüências óbvias. O aumento de produtividade pode refletir-se no faturamento, permitindo à empresa uma recuperação mais rápida de seus investimentos iniciais, além de melhorar a capacidade de atendimento à demanda do mercado. Neste artigo é mostrado como o problema de balanceamento de uma linha de montagem pode ser modelado e resolvido com sucesso por um AG padrão.

Alguns trabalhos já foram desenvolvidos com a utilização de AGs na otimização de problemas combinatoriais na indústria, incluindo variações dos problemas clássicos de *job shop* e *flow shop*, bem como outros problemas de maior complexidade. Veja, por exemplo, Chan & Hu (2000), Knosala & Wal (2001), Kopfer (1996), Li & Love (1998), Li et al (1998), Nakano (1991), Reeves (1995), e Warwick e Tsang (1996). A solução do problema de balanceamento de uma linha de montagem foi particularmente tratada por Anderson & Ferris (1994). Nesse trabalho, os autores propõem a codificação das variáveis e a matriz de precedências que, com algumas modificações, foram utilizadas no presente trabalho. Não foi adotada, entretanto, a implementação paralela do AG contida no referido artigo.

3 O Problema de Otimização da Linha de Montagem

O trecho da linha de montagem sob análise é mostrado de forma esquemática na Fig. 1. Da esquerda para a direita, os veículos sendo montados entram na linha e passam pelos dez postos de trabalho (P₁ a P₁₀). Em cada um deles pode haver atividades (A_i) em um ou em ambos os lados da linha. A linha não pode ser completamente compacta, isto é, não pode haver atividades em ambos os lados da linha em todos os postos de trabalho, devido às restrições de espaço causadas pelas máquinas utilizadas no processo de montagem, além da presença da logística, onde as peças são disponibilizadas aos operadores em prateleiras e caixas.

Cada posto de trabalho tem comprimento fixo e o veículo move-se continuamente, sem parar em momento algum do processo. O tempo total de trabalho em um dos postos é dado pelo tempo da atividade de maior duração deste posto, uma vez que ambas iniciam-se no mesmo instante. Assim, o tempo total

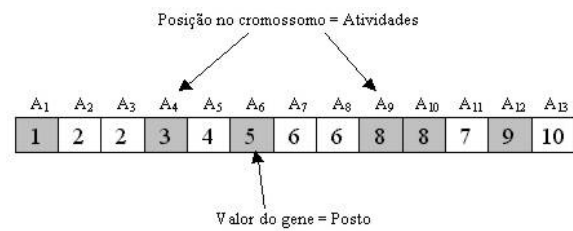


Figura 2. Codificação do Cromossomo.

de trabalho realizado no posto P₂, por exemplo, é dado pela maior das durações entre as atividades A₂ e A₃.

O tempo total de trabalho em todo o trecho é dado pela soma dos tempos em cada um dos postos. Na linha real, ainda não otimizada, este tempo é atualmente de 21min22s (ou 21,37min).

4 Solução Usando Algoritmos Genéticos

4.1 Codificação das Variáveis

A Fig. 2 mostra a codificação do cromossomo adotada para a representação das soluções candidatas. O cromossomo é composto de 13 números inteiros, um para cada atividade a ser realizada na linha, todos entre 1 e 10. O conteúdo de cada “gene” identifica um dos 10 postos de trabalho, enquanto que a posição que o mesmo ocupa no cromossomo mapeia as atividades realizadas neste posto. No exemplo da Fig. 1, as atividades A₂ e A₃ são realizadas no posto P₂, a atividade A₆ no posto P₅, e assim para os demais.

Apesar de tratar-se de um problema de natureza combinatorial, os operadores especiais de *crossover* citados em Goldberg (1989), tais como PMX, OX e CX não se aplicam, uma vez que a ordem de precedência entre as atividades não obedece diretamente a numeração das mesmas. Por exemplo, a atividade A₁₁ deve ser feita antes da atividade A₉, apesar de a mesma aparecer em uma posição mais à direita do cromossomo. Desta forma, a informação de posição relativa de um “gene” em relação a outro não é relevante. Esta é uma característica especial desta abordagem em relação às demais implementações encontradas na literatura.

4.2 Dados Relevantes

Para o tratamento do problema são necessárias duas informações fundamentais: o tempo necessário para a execução de cada atividade e a ordem de precedência entre elas. Os dados mostrados nas Tabelas 1 e 2 são da linha de produção real e representam, respectivamente, os tempos de execução de cada atividade e as precedências entre as atividades.

Na Tabela 1, o Tempo de Trabalho Efetivo – TTE é o tempo necessário para o operador realizar uma atividade, o Tempo de Operações Variáveis –

Tabela 1. Tempos de Execução das Atividades.

Atividade	Tempo (minutos)		
	Tempo de Trabalho Efetivo (TTE)	Tempo de Operações Variáveis (TOV)	Tempo Total da Atividade (TTA)
A ₁	0,91	0,24	1,15
A ₂	1,84	0,53	2,37
A ₃	1,64	0,43	2,07
A ₄	0,99	0,31	1,30
A ₅	2,14	0,56	2,70
A ₆	1,89	0,63	2,52
A ₇	1,78	0,28	2,06
A ₈	1,46	0,56	2,02
A ₉	1,72	0,52	2,24
A ₁₀	1,58	0,37	1,95
A ₁₁	1,81	0,46	2,27
A ₁₂	1,70	0,52	2,22
A ₁₃	2,17	0,37	2,54

TOV é o tempo necessário para operações associadas (por exemplo, o tempo necessário ao operador para deslocar-se e pegar uma peça na prateleira), e o Tempo Total da Atividade – TTA é a soma de TTE e TOV. Note-se que a parte fracionária dos tempos é dada em centésimos de minutos de forma a facilitar as operações aritméticas, uma vez que as conversões são evitadas.

A Tabela 2 mostra as relações de precedência entre as atividades. Para um dado par de atividades, A_i e A_j, há precedência entre A_i e A_j se houver um “1” na intersecção da linha de A_i com a coluna de A_j. Caso contrário, não há precedência.

Por exemplo, considere as atividades A₉ e A₁₁ (conforme citado no item 4.1). A Tabela 2 mostra que há precedência entre estas atividades (observe o “1” na linha 9, coluna 11), significando que a atividade A₁₁ deve ser feita antes da atividade A₉. A recíproca é óbvia, e a posição da tabela com coluna 9 e linha 11 deve conter o número “0”.

O formato de tabela para a representação da ordem de precedência entre as atividades tem algumas vantagens: permite uma visualização rápida por inspeção, permite a correção rápida de qualquer precedência que venha a ser modificada em função de alterações no processo, e permite um processamento rápido pelo AG no momento de se determinar as violações para o cálculo de penalidades.

4.3 Funções Objetivo e de Fitness

Conforme já dito, o tempo total de trabalho em um posto é dado pelo tempo da atividade de maior duração deste posto, uma vez que ambas iniciam-se no mesmo instante. Se há apenas uma atividade atribuída ao posto, o tempo do posto é igual à duração

Tabela 2. Precedências Entre Pares de Atividades.

	A ₁	A ₂	A ₃	A ₄	A ₅	A ₆	A ₇	A ₈	A ₉	A ₁₀	A ₁₁	A ₁₂	A ₁₃
A ₁	0	0	0	0	0	0	0	0	0	0	0	0	0
A ₂	1	0	0	0	0	0	0	0	0	0	0	0	0
A ₃	1	0	0	0	0	0	0	0	0	0	0	0	0
A ₄	1	1	1	0	0	0	0	0	0	0	0	0	0
A ₅	1	0	0	0	0	0	0	0	0	0	0	0	0
A ₆	1	0	0	0	1	0	0	0	0	0	0	0	0
A ₇	1	0	1	0	0	0	0	0	0	0	0	0	0
A ₈	1	0	0	0	1	0	0	0	0	0	0	0	0
A ₉	1	0	1	0	1	0	1	1	0	0	1	0	0
A ₁₀	1	0	1	0	1	0	1	1	0	0	1	0	0
A ₁₁	0	0	0	0	0	0	0	0	0	0	0	0	0
A ₁₂	1	0	1	0	1	0	1	1	0	0	0	0	0
A ₁₃	1	0	1	0	1	0	1	1	0	0	0	1	0

desta atividade. Postos sem atividade têm tempo nulo (0).

A função objetivo é calculada como o somatório dos tempos trabalhados em todos os postos, dado pela equação (1), em que $TTA[A_{esq}(P_i)]$ é o Tempo Total da Atividade atribuída ao lado esquerdo do i -ésimo posto e $TTA[A_{dir}(P_i)]$ é o seu equivalente para o lado direito.

$$F_{obj} = \sum_{i=1}^{10} \max\{TTA[A_{esq}(P_i)], TTA[A_{dir}(P_i)]\} \quad (1)$$

A função de *fitness* (a ser maximizada) é calculada pela equação (2) levando-se em consideração as penalidades (analisadas na próxima seção). O seu valor é então normalizado, isto é, forçado a permanecer no intervalo [0,1]:

$$F_{fitness} = 1 - \frac{F_{obj} + C_p \cdot Penalty}{STT + 72 \cdot Tp_{max}} \quad (2)$$

C_p é o coeficiente de aplicação de penalidade (fixado em 2,0), 72 é número máximo possível de violações¹ e STT é a somatória dos tempos de execução de todas as atividades da linha, dada pela equação (3).

$$STT = \sum_{i=1}^{13} TTA(A_i) \quad (3)$$

4.4 Restrições e Penalidades

Durante o processo “evolutivo”, nem todos os indivíduos representam soluções factíveis. Entretanto, estudos como o citado em Anderson & Ferris (1994) mostram que é interessante manter-se alguns destes indivíduos “não adaptados” na população para con-

¹ A partir da expressão $11*2+9*2+7*2+5*2+3*2+1*2=72$, obtida pela análise do pior caso, em que as 2 atividades atribuídas ao primeiro posto violam todas as precedências em relação às outras 11 atividades, as 2 atividades atribuídas ao segundo posto violam todas as precedências em relação às 9 atividades restantes, e assim por diante.

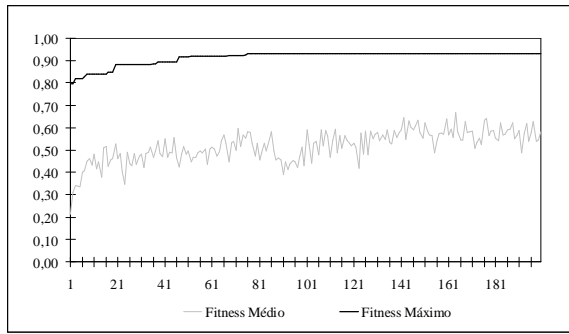


Figura 3. Curvas de Evolução do Fitness.

servar a diversidade genética da mesma. Assim, as chamadas “soluções candidatas” que não representam soluções adequadas ao problema de otimização são “penalizadas”, ao invés de serem simplesmente eliminadas. As restrições que devem ser impostas às soluções candidatas são:

- a) Existência de, no máximo, duas atividades por posto de trabalho; esta restrição é devida à característica da linha, que só pode ter postos à sua esquerda e à sua direita. (obs.: há postos onde algumas atividades são executadas sob ou sobre o veículo; mesmo assim, para fins de planejamento, as atividades são sempre atribuídas às laterais da linha onde encontram-se as operações);
- b) As operações e atividades deve obedecer as precedências mostradas na Tabela 2.

Quando estas restrições não são totalmente satisfeitas a solução candidata sofre penalidades, as quais se refletem como um decremento no valor do *fitness* deste indivíduo. Este decremento é função do grau de violações às restrições acima e obedece aos seguintes critérios:

- a) Qualquer indivíduo que apresente mais de 2 atividades por posto de trabalho tem penalidade máxima, e seu *fitness* é igualado a 0; como esta restrição é uma limitação física, não há interesse em que se perpetuem indivíduos com estas características; e
- b) Indivíduos que respeitem a primeira restrição mas que violem a precedência entre as atividades, são punidos (em tempo) proporcionalmente ao número de violações, segundo a equação (4):

$$Penalidade = V.Tp_{m\acute{a}x} \quad (4)$$

Onde V indica o número de violações às precedências e $Tp_{m\acute{a}x}$ o tempo de duração da atividade mais longa da linha.

4.5 Parâmetros do AG

Após vários experimentos com diferentes combinações de parâmetros para o AG, os melhores resultados foram obtidos com os parâmetros descritos a seguir. Os dois operadores genéticos clássicos foram utilizados: *crossover* de ponto (com probabilidade de 80%) e mutação simples de bits (com

probabilidade de 4% por bit). O elitismo foi usado através das gerações de modo conservar o melhor indivíduo encontrado em uma geração na geração seguinte. Uma vez que o elitismo pode causar uma maior pressão seletiva, foi utilizado um método de seleção mais eficiente do que a roleta tradicional: o *stochastic universal sampling*. A população era composta de 100 indivíduos com *gap* entre as gerações de 1, o que significa que a cada geração toda uma nova população era criada, sem a permanência de indivíduos em mais de uma geração consecutiva. O critério de parada do processo evolutivo foi simplesmente o de limitação de tempo, através da fixação do número máximo de gerações em 200.

O desenvolvimento do sistema baseou-se em uma versão anterior do pacote de software chamado *Galopps (Genetic Algorithm Optimized for Portability and Parallelism)*, desenvolvido por Goodman (1996) e escrito em ANSI C.

5 Resultados

A evolução do *fitness* é mostrada na Fig. 3. Nela, o eixo “x” representa as gerações, o eixo “y” o valor de *fitness* (normalizado entre 0 e 1), o traço superior o *fitness* do melhor indivíduo e o traço inferior o *fitness* médio da população. Cada execução do AG consumiu cerca de 0,85s de processamento num Pentium™ III 500MHz em Windows™ 98.

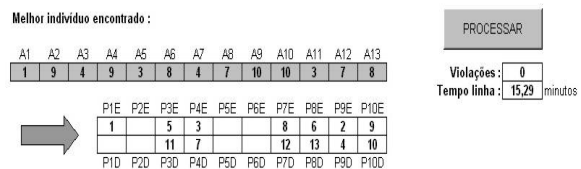


Figura 4. Planilha de Análise do Melhor Indivíduo.

Para a avaliação dos melhores indivíduos encontrados em cada execução do AG, foi construída uma planilha em Microsoft Excel™, com a finalidade de calcular o tempo total de trabalho das configurações de linha representadas por estes indivíduos. A planilha também analisa o mapeamento do cromossomo e elabora uma pequena representação esquemática dos postos na linha. Parte desta planilha é mostrada na Fig. 4.

A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13
1	9	4	9	3	8	4	7	10	10	3	7	8

Figura 5. Melhor Indivíduo Encontrado.

Após diversas execuções do AG, o melhor indivíduo encontrado foi o representado na Fig. 5. Este indivíduo representa a solução para a configuração da linha esquematizada na Fig. 6. O melhor indivíduo encontrado leva a um tempo total de trabalho na linha de 15,29min, o que representa uma diminuição

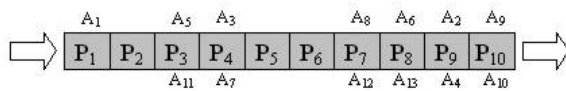


Figura 6. Melhor Solução Encontrada (*Layout* da Linha).

de 28,5% em relação ao tempo da configuração atual, que é de 21,37min.

6 Conclusões

Este trabalho procurou demonstrar que os AGs podem ser muito úteis na busca de melhores alternativas de *layout* para uma linha de montagem. Se tal análise tivesse sido feita durante a fase de projeto da planta em questão, os resultados seriam ainda mais expressivos, com a sugestão de melhorias nos arranjos de todas as linhas de produção, otimizando o tempo global de trabalho.

Levando-se em consideração a dimensão do espaço de busca e a natureza combinatorial deste tipo de problema, o AG mostrou-se bastante rápido e eficiente. Uma comparação mais detalhada com outras metodologias clássicas para o problema de balanceamento de uma linha de montagem, a ser feita futuramente, investigará o quão competitivo um AG, de fato, pode ser.

Por simplificação, o presente trabalho não levou em consideração questões de logística, tais como a disposição das prateleiras de peças. Também não foram considerados os custos decorrentes das modificações de *layout* da linha atual. Entretanto, a solução encontrada é factível e mostra que os AGs podem ser usados como uma ferramenta poderosa para a organização fabril, com significativos ganhos de tempo e, possivelmente, de produtividade e de lucro para as empresas.

Referências Bibliográficas

- Anderson, E.J. & Ferris, M.C. (1994). Genetic algorithms for combinatorial optimization: the assembly line balancing problem. *ORSA Journal of Computing*, vol. 6, no. 2, pp. 161-173.
- Chan, W.T. & Hu, H. (2000). Precast production scheduling with genetic algorithms. *Proc. of 2000 Congress on Evolutionary Computation*, vol. 2, pp. 1087-1094.
- Goldberg, D.E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley.
- Goodman, E.D. (1996). An Introduction to Galopps – The Genetic Algorithm Optimized for Portability and Parallelism System. *Technical Report #96-07-01*, Michigan State University, East Lansing.

- Knosala, R. & Wal, T. (2001). A production scheduling problem using genetic algorithms. *Journal of Materials Processing Technology*, vol. 109, no. 1-2, pp. 90-95.
- Kopfer, H. (1996). *Evolutionary search and the job shop: investigations on genetic algorithms for production scheduling*. Heidelberg: Physica-Verlag.
- Li, H. & Love, P. (1998) Site-level facilities layout using genetic algorithms. *Journal of Computing in Civil Engineering*, vol. 12, no. 4, pp. 227-231.
- Li, Y. & Ip, W.H. & Wang, D.W. (1998) Genetic algorithm approach to earliness and tardiness production scheduling and planning problem. *International Journal of Production Economics*, vol. 54, no. 1, pp. 65-76.
- Nakano, R. (1991). Conventional genetic algorithm for job shop problems. *Proc. of the 4th International Conference on Genetic Algorithms*, pp. 474-479.
- Reeves, C.R. (1995). A genetic algorithm for flowshop sequencing. *Computers in Operations Research*, vol. 22, no. 1, pp. 5-13.
- Warwick, T. & Tsang, E.P.K. (1996). Tackling car sequencing problems using a generic genetic algorithm. *Evolutionary Computation*, vol. 3, no 3, pp. 267-298.