

IMPLEMENTAÇÃO DE ALGORÍTMOS GENÉTICOS NO BALANCEAMENTO INTERATIVO DE LINHAS DE MONTAGEM

Éldman de Oliveira Nunes

Instituto de Computação – UFF, R Passo da Pátria 156, 24210-240, Niterói, RJ
enunes@ic.uff.br

Aura Conci

Instituto de Computação – UFF, R Passo da Pátria 156, 24210-240, Niterói, RJ
aconci@ic.uff.br

Resumo

O emprego de métodos convencionais para solução de problemas de otimização combinatória NP-difíceis normalmente torna-se inviável em virtude do enorme esforço computacional exigido para sua solução. Algoritmos genéticos (AG's) são metaheurísticas adequadas para solução de problemas desta natureza. Esse trabalho descreve um estudo da aplicação de um AG para o problema de otimização de um trecho da linha de montagem de automóveis. Ao algoritmo genético apresentado no artigo original foram incorporadas modificações visando melhorar o seu desempenho. Os resultados computacionais demonstraram que as modificações propostas melhoram de forma considerável os resultados originais do artigo referenciado.

Palavras-chave: Metaheurísticas, algoritmos genéticos, linhas de montagem

Abstract

The use of conventional methods to find optimum or close solutions for problems of optimization NP-hard (nondeterministic polynomial) normally becomes impracticable due to the enormous computational effort required for its solution. Genetic Algorithms (GA's) are metaheuristics based on the mechanisms of natural selection and the genetics for solution of problems of this nature. An interface for GA is implemented in [2] based on the proposition presented in [1], objectively better alternatives of layout for a automotive assembly line. The solution encountered was 28.5% better than the current solution adopted in the real line. This paper presents diverse modifications in the GA described in [1]: genetic parameters you configured, heuristical generation of the initial population and new operators of crossing, mutation and local searches. These modifications allowed to find out a configuration of the line that could reduce the time in 30,3% of the current configuration representing a significant cost reduction. A comparative analysis between the genetics operators of both implementations is made and its argued results.

Keywords: Genetic algorithms, assembly lines, optimization.

1. INTRODUÇÃO

A natureza combinatória do problema de otimização de uma linha de montagem aliado ao seu amplo espaço de busca (espaço de possíveis soluções) são fortes razões para o uso de metaheurísticas como Algoritmos Genéticos (AG's). Um AG é empregado em [1] na busca de melhores alternativas de *layout* para uma linha de montagem automotiva. A solução encontrada foi 28,5% melhor do que a solução atual adotada numa linha de montagem real, e esta redução no tempo de montagem da produção representa uma significativa redução de custos. Este trabalho apresenta diversas modificações no AG descrito em [1]: parâmetros genéticos configuráveis, geração heurística da população inicial e novos operadores de

cruzamento, mutação e busca local. Estas modificações permitiram encontrar uma configuração da linha que reduzisse em 30,3% do tempo da configuração atual. Uma análise comparativa entre os operadores genéticos propostos e os descritos em [1] é feita e seus resultados discutidos.

2. DESCRIÇÃO DO PROBLEMA

O trecho da linha de montagem considerado, ilustrado na figura 1, 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.

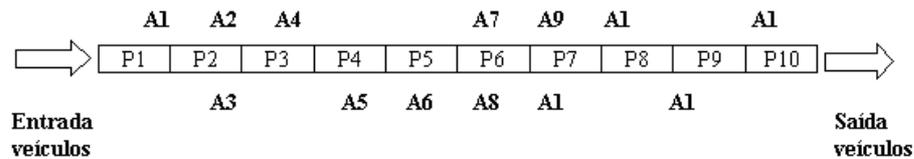


Figura 1. Trecho da linha de montagem

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. Não pode haver atividades em ambos os lados da linha em todos os postos de trabalho, devido às restrições de espaço (suprimentos e equipamentos). 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 se iniciam no mesmo instante. Assim, o tempo total 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).

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, tabelas 1 e 2, respectivamente.

Precedência entre pares de atividades													
	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13
A1	0	0	0	0	0	0	0	0	0	0	0	0	0
A2	1	0	0	0	0	0	0	0	0	0	0	0	0
A3	1	0	0	0	0	0	0	0	0	0	0	0	0
A4	1	1	1	0	0	0	0	0	0	0	0	0	0
A5	1	0	0	0	0	0	0	0	0	0	0	0	0
A6	1	0	0	0	1	0	0	0	0	0	0	0	0
A7	1	0	1	0	0	0	0	0	0	0	0	0	0
A8	1	0	0	0	1	0	0	0	0	0	0	0	0
A9	1	0	1	0	1	0	1	1	0	0	1	0	0
A10	1	0	1	0	1	0	1	1	0	0	1	0	0
A11	0	0	0	0	0	0	0	0	0	0	0	0	0
A12	1	0	1	0	1	0	1	1	0	0	0	0	0
A13	1	0	1	0	1	0	1	1	0	0	0	1	0

Tabela 1. Precedência entre pares de atividades

Tempos de execução das atividades			
Atividade	Tempo (minutos)		
	Tempo de Trabalho Efetivo (TTE)	Tempo de Operações Variáveis (TOV)	Tempo Total da Atividade (TTA)
A1	0,91	0,24	1,15
A2	1,84	0,53	2,37
A3	1,64	0,43	2,07
A4	0,99	0,31	1,30
A5	2,14	0,56	2,70
A6	1,89	0,63	2,52
A7	1,78	0,28	2,06
A8	1,46	0,56	2,02
A9	1,72	0,52	2,24
A10	1,58	0,37	1,95
A11	1,81	0,46	2,27
A12	1,70	0,52	2,22
A13	2,17	0,37	2,54

Tabela 2. Tempo de execução das atividades

A otimização do tempo de trabalho em uma unidade industrial tem conseqüências óbvias. O aumento de produtividade pode refletir-se no faturamento e na melhora da capacidade de atendimento à demanda do mercado. Num problema como este, o número total de configurações possíveis é dado pelo arranjo $A_{2n,p}$, isto é, $(2n!)/(2np)!$, 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 $4,83 \times 10^{14}$ 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 $3,58 \times 10^{118}$. Este espaço de busca é intratável por qualquer método de busca seqüencial, em um tempo razoável, por mais recursos computacionais de que se disponha. Neste cenário, os AGs mostram-se como uma alternativa simples e eficiente.

3. DESCRIÇÃO DO AG ORIGINAL

Uma solução para o problema apresentado na seção anterior é desenvolvida em [1] e descrita a seguir. O cromossomo, ilustrado na figura 2, 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. As atividades A2 e A3 são realizadas no posto P2, a atividade A6 no posto P5, e assim para os demais.

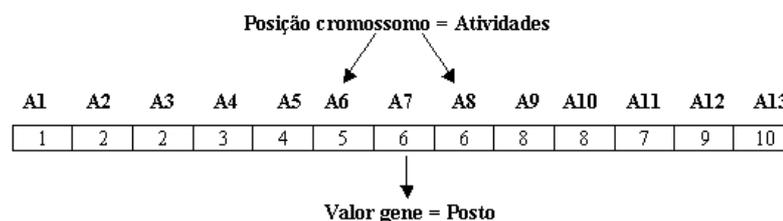


Figura 2. Codificação do cromossomo da figura 1.

3.1 PARÂMETROS DO AG

- *População inicial* : Composta de 100 indivíduos.
- *Avaliação dos indivíduos*: Será descrito mais adiante.
- *Reprodução*: Os dois operadores genéticos clássicos utilizados foram o **crossover** de um ponto (com probabilidade de 80%) e **mutação simples** de bits (com probabilidade de 4% por bit).
- *Seleção*: 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 roleta (stochastic universal sampling).
- *Sobrevivência* : 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.
- *Critério de parada*: o de limitação de tempo, através da fixação do número máximo de gerações em 200.

3.2 AVALIAÇÃO DOS INDIVÍDUOS

O tempo total de trabalho em um posto é dado pelo tempo da atividade de maior duração deste posto, uma vez que ambas se iniciam no mesmo instante. Se há apenas uma atividade atribuída ao posto, o tempo do posto é igual à duração desta atividade. Postos sem atividade têm tempo nulo (0). A função de **fitness** a ser maximizada, equação 1, é calculada pela equação a seguir levando-se em consideração as penalidades. O seu valor é então normalizado, isto é, forçado a permanecer no intervalo [0,1].

$$1. \quad \underline{F}_{fitness} = 1 - (F_{obj} + C_p \cdot Penalty) / STT + 72 . \quad (1)$$

A função objetivo, equação 2, é calculada como o somatório dos tempos trabalhados em todos os postos, 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.

$$2. \quad F_{Obj} = \sum_{(i=1 \rightarrow 10)} \max \{ TTA [A_{esq} (P_i)], TTA [A_{dir} (P_i)] \} \quad (2)$$

$$2.1. \quad C_p \text{ (Coeficiente de aplicação de penalidade)} = 2,0$$

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.
- b) A ordem das atividades deve obedecer as precedências mostradas na tabela de precedências.

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;

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 3.

$$\text{Penalidade} = V \cdot T_{p \max} \quad (3)$$

onde V = número de violações às precedências, e $T_{p \max}$ = tempo de duração da atividade mais longa da linha.

O número máximo possível de violações é calculado a partir da expressão $(11 \cdot 2 + 9 \cdot 2 + 7 \cdot 2 + 5 \cdot 2 + 3 \cdot 2 + 1 \cdot 2) = 72$, 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.

2.1. Somatória dos tempos de execução de todas as atividades da linha (STT), é dada pela equação 4.

$$3. \quad \text{SST} = \sum_{(i=1 \rightarrow 13)} \text{TTA}(A_i) \quad (4)$$

3.3 RESULTADOS:

Cada execução do AG consumiu cerca de 0,85s de processamento num Pentium™ III 500MHz em Windows™ 98. O melhor indivíduo encontrado e o seu correspondente layout da linha de montagem, figura 3 e 4, leva a um tempo total de trabalho na linha de **15,29min**, o que representa uma diminuição de 28,5% em relação ao tempo da configuração atual, que é de 21,37min. 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.

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 3. Melhor cromossomo encontrado.

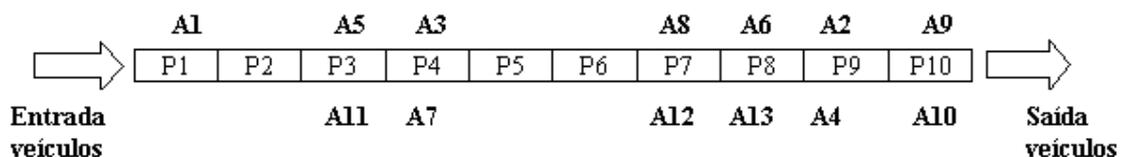


Figura 4. Melhor solução encontrada (layout da linha)

4. DESCRIÇÃO DO AG PROPOSTO

O Algoritmo Genético descrito na seção 3 foi implementado utilizando o ambiente de desenvolvimento Borland Delphi Enterprise, versão 7.0 e linguagem de programação Object Pascal. A figura 5 retrata a interface do programa.

O AGBALIM (Algoritmo Genético para Balanceamento de Linha de Montagens) permite as seguintes configurações:

- 1) *Número de gerações* : de 1 a 300;
- 2) *Critério de parada*: pelo número de gerações ou por estabilização do fitness (encerra após k gerações sem modificação do fitness, sendo $10 \leq k \leq 100$);
- 3) *Forma de geração e o tamanho da população inicial*: a população inicial pode ser gerada aleatoriamente ou através de uma forma heurística. O tamanho da população pode variar de 1 a 200;
- 4) *Tipo de Seleção*: Roleta (stochastic universal sampling);
- 5) *Tipo de operador genético de Cruzamento*: cruzamento de um ponto ou através do operador especial de cruzamento proposto, com taxa de ocorrência variável entre 50% e 100%;
- 6) *Tipo de operador genético de Mutação*: mutação simples ou operador especial de mutação proposto, com taxa de ocorrência variável entre 1% e 10%;
- 7) *Opção de Busca Local*: com taxa de ocorrência variável entre 1% e 10%;
- 8) *Definição dos tempos das atividades*;
- 9) *Definição da precedência entre as atividades*;

As seguintes informações podem ser obtidas com AGBALIM:

- 1) *Gráfico do fitness do melhor indivíduo em cada geração*;
- 2) *Gráfico do fitness médio da população em cada geração*;
- 3) *Relatório contendo*: parâmetros de configuração do AG, número da geração, codificação do cromossomo, valor da função objetivo, valor da função de avaliação (fitness), número de violações, valor da função de avaliação média da população a cada geração, tempo de execução do AG;
- 4) *Codificação do melhor cromossomo*;
- 5) *Lay out da linha para o melhor cromossomo*;
- 6) *Tempo da linha do melhor cromossomo*;
- 7) *Número de violações do melhor cromossomo*;
- 8) *Informações sobre o artigo*.

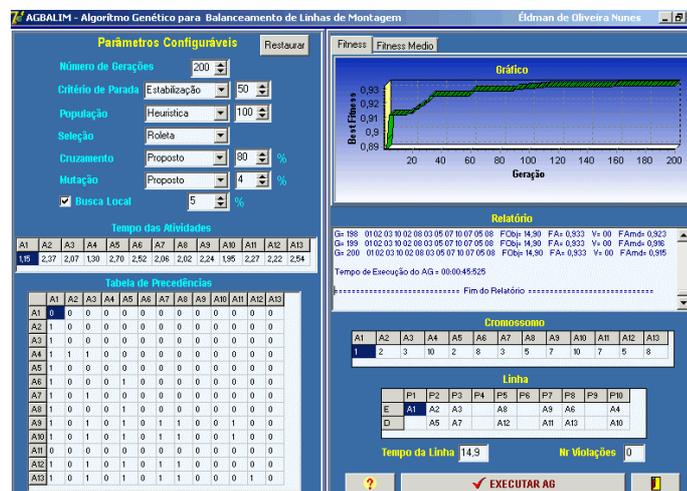


Figura 5. Layout da interface do programa proposto

4.1 INICIALIZAÇÃO HEURÍSTICA

Em adição a forma aleatória da geração da população inicial descrita no artigo em estudo, o algoritmo AGBALIM permite a geração da população através da execução de um AG simples com os seguintes parâmetros:

- Número de gerações = 5;
- Tamanho da População = 10;
- Cruzamento de um ponto;
- Mutação simples.
- Taxa de cruzamento e mutação definido pelo usuário;

4.2 OPERADOR DE CRUZAMENTO

Além do cruzamento de um ponto descrito no artigo em estudo, foi implementado um novo operador genético de cruzamento. A figura 6 apresenta o fluxograma deste novo operador. Inicialmente um número aleatório, representado o ponto de corte, é gerado. Em seguida, a troca de material genético entre os pais para formar os filhos é realizada como no cruzamento de um ponto. Após isso, os filhos gerados são avaliados. Se ambos os filhos forem válidos (decisão 1), isto é, possuírem fitness diferente de zero (lembrando que o fitness igualado a zero é penalidade máxima para a solução que possui mais de duas atividades em um mesmo posto) estes serão candidatos a inserção na nova população. Se pelo menos um dos filhos não for válido, a decisão 2 é realizada, verificando se um contador é menor que 20. Se o contador for maior do que 20, os dois pais serão candidatos a inserção na nova população. Senão, a decisão 3 é verificada para saber se pelo menos um dos filhos é válido. Se um dos filhos for válido, o filho válido com seu respectivo pai é candidato a ser inserido na nova população. Caso contrário, um novo ponto de corte aleatório é gerado e o processo se repete. Desta forma, o novo operador genético de mutação evita a inserção de filhos inválidos na nova população.

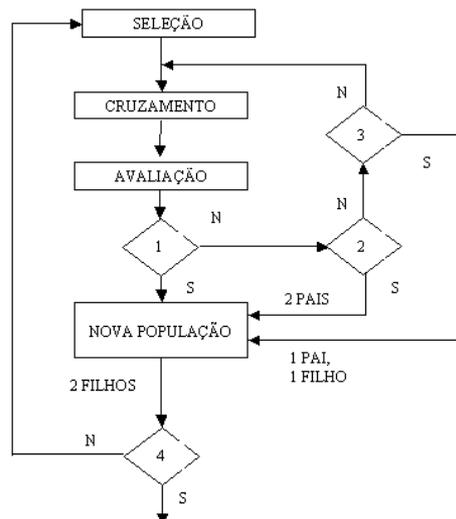


Figura 6. Fluxograma do operador de cruzamento proposto

4.2 OPERADOR DE MUTAÇÃO

Além da mutação simples descrita no artigo em estudo, foi implementado um novo operador genético de mutação. A figura 7 ilustra o processo realizado por este novo operador. Inicialmente dois números aleatórios são gerados. Ambos os números determinam a posição

no cromossomo onde ocorrerá a mutação. Em seguida, os valores de ambos os genes nos pontos de mutação são trocados dentro do mesmo cromossomo. Esta operação é realizada para os dois filhos. Este processo de mutação permite inserir diversidade genética sem ocasionar o surgimento de soluções não factíveis.

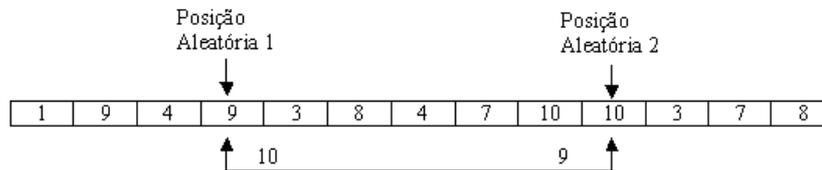


Figura 7. Filho gerado pela mutação proposta

4.3 BUSCA LOCAL

A última modificação ao algoritmo genético do artigo em estudo foi a implementação de um módulo de busca local. Caso o usuário tenha optado pela realização da busca, os pares de cromossomos oriundos da seleção, cruzamento e/ou mutação serão submetidos ao seguinte processo (ver figura 8):

- 1) O primeiro gene de cada cromossomo será fixado (pivô);
- 2) O valor do pivô será trocado com o valor do segundo gene em cada cromossomo;
- 3) Os novos cromossomos serão avaliados;
- 4) Armazenar o melhor e o segundo melhor cromossomo;
- 5) Repetir 2) e 3) mantendo o pivô e variando o gene de troca do terceiro ao último gene;
- 6) Fixar o segundo gene de cada cromossomo como pivô;
- 7) Repetir o processo de 2) a 5);
- 8) Fixar o próximo gene como pivô (repetir até o penúltimo gene);
- 9) Retornar o melhor e o segundo melhor cromossomos encontrados.

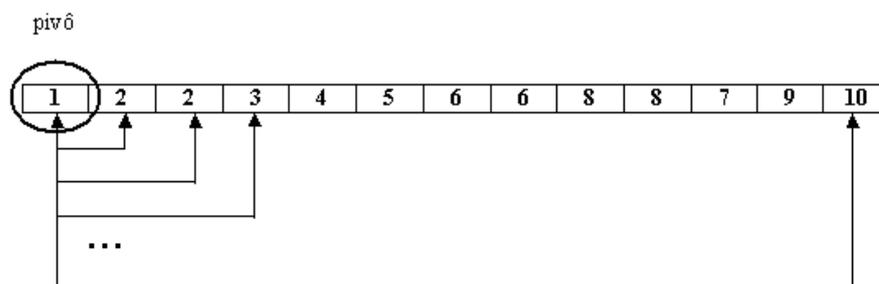


Figura 8. Busca local proposta

Para o primeiro gene de um cromossomo serão realizadas 12 trocas, para o segundo 11 trocas e assim por diante. No total serão realizadas 78 trocas.

5. TESTES E RESULTADOS

Diversos testes para verificação do desempenho do AG original e do AG modificado foram realizados. Estes testes foram realizados num Pentium III 750 MHz, 128 MB RAM em Windows 2000 e serão descritos a seguir. Inicialmente, foram realizadas 10 (dez) gerações aleatórias da população inicial e 10 (dez) gerações heurísticas (AG simples descrito na seção 4.2). O gráfico 1 apresenta o resultado encontrado.

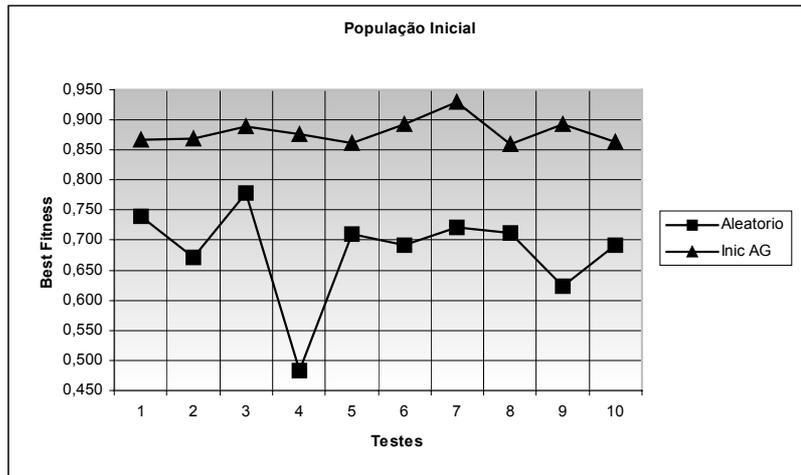


Gráfico 1. Testando a inicialização heurística (fitness do melhor indivíduo da 1ª geração).

Pode-se verificar claramente que o fitness dos 10 (dez) melhores indivíduos (o melhor de cada uma das dez populações iniciais) geradas pelo AGsimples ficou acima de 0,850 enquanto que a maioria dos melhores indivíduos gerados aleatoriamente apresentaram *fitness* abaixo de 0,750. A geração aleatória da população inicial se mostrou pouco eficiente, pois gerou cromossomos com baixo valor de *fitness* e pouca diversidade na população uma vez que muitos indivíduos foram descartados por serem irrealizáveis. A geração por heurística apresentou resultados melhores sem perder diversidade genética.

Novos casos de testes foram realizados agora verificando o melhor cromossomo encontrado após 200 gerações, com população de tamanho 100, cruzamento de um ponto, taxa de cruzamento de 80%, mutação simples, taxa de mutação de 4%. Foram realizados dez testes com geração aleatória da população inicial e dez testes com geração heurística. O gráfico 2 apresenta o resultado encontrado. É possível verificar que o tempo da linha (Fobj) foi menor na grande maioria dos casos para o AG que executou sobre uma população inicial gerada de forma heurística. Para todos os resultados, tanto com AG original quanto com AG modificado, o número de violações foi igual a zero.

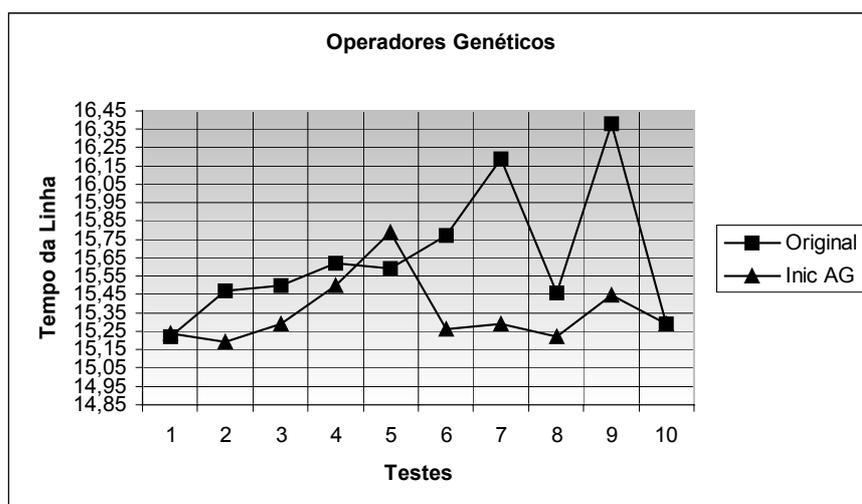


Gráfico 2. Testando a inicialização heurística (melhor indivíduo encontrado).

Para testar o novo operador genético de cruzamento, novos casos de testes foram realizados, agora verificando o melhor cromossomo encontrado após 200 gerações, com

população de tamanho 100, população inicial aleatória, mutação simples e taxa de mutação de 4%. Foram realizados dez testes com cruzamento de um ponto e taxa de cruzamento de 80% e dez testes com o operador de cruzamento proposto e taxa de cruzamento de 80%. O gráfico 3 apresenta o resultado encontrado. Para todos os resultados, tanto com AG original quanto com AG modificado, o número de violações foi igual a zero. Aparentemente este operador não resultou em melhora significativa agindo isoladamente. É conveniente lembrar que este operador atuou sobre uma população inicial gerada aleatoriamente contendo provavelmente vários elementos infactíveis e pouca diversidade genética. Tal situação pode ter comprometido seu desempenho.

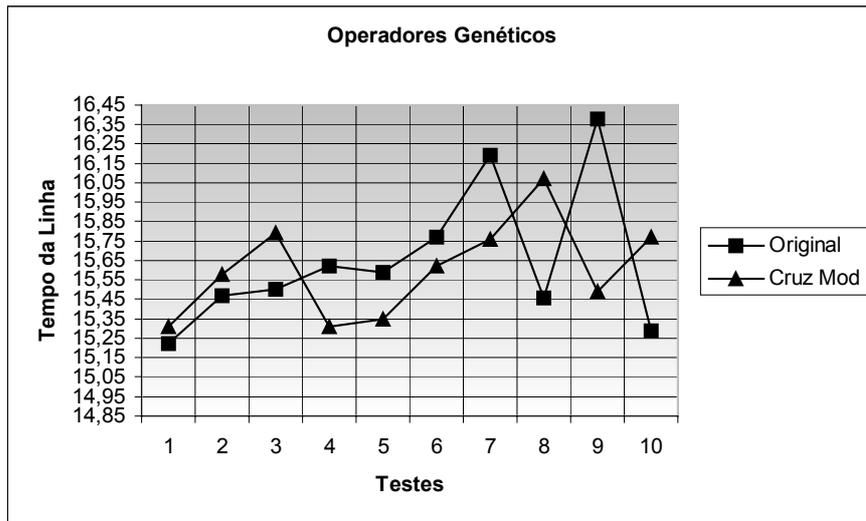


Gráfico 3. Testando o cruzamento proposto (melhor indivíduo encontrado).

Para testar o novo operador genético de mutação, novos casos de testes foram realizados, agora verificando o melhor cromossomo encontrado após 200 gerações, com população de tamanho 100, população inicial aleatória, cruzamento de um ponto e taxa de cruzamento de 80%. Foram realizados dez testes com mutação simples e taxa de mutação de 4% e dez testes com o operador de mutação proposto e taxa de mutação de 4%. O gráfico 4 apresenta o resultado encontrado. Para todos os resultados, tanto com AG original quanto com AG modificado, o número de violações foi igual a zero. Percebe-se claramente uma sensível superioridade dos resultados obtidos com o uso do operador de mutação proposto.

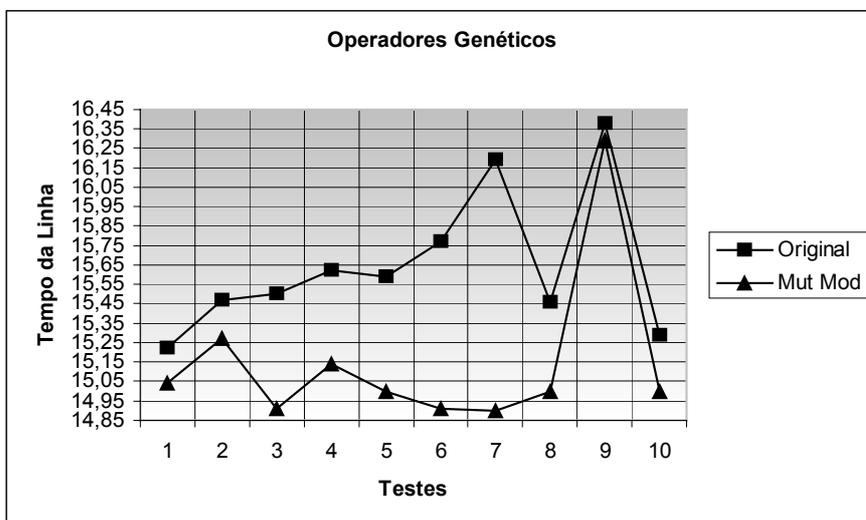


Gráfico 4. Testando a mutação proposta (melhor indivíduo encontrado).

Para testar o novo operador genético de busca local, novos casos de testes foram realizados, agora verificando o melhor cromossomo encontrado após 200 gerações, com população de tamanho 100, população inicial aleatória, cruzamento de um ponto, taxa de cruzamento de 80 %, mutação simples e taxa de mutação de 4%. Foram realizados dez testes sem busca local e dez testes com busca local e taxa de busca de 5%. O gráfico 5 apresenta o resultado encontrado. Para todos os resultados, tanto com AG original quanto com AG modificado, o número de violações foi igual a zero. Pode-se verificar que os melhores indivíduos encontrados empregando a busca local apresentaram uma forma de minimizar a Fobj (tempo da linha) bem maior do que os encontrados sem o seu emprego.

Para testar os novos operadores genéticos em conjunto, novos casos de testes foram realizados, agora verificando o melhor cromossomo encontrado após 200 gerações, com população de tamanho 100. Dez testes foram realizados com o AG original (população inicial aleatória, cruzamento de um ponto, taxa de cruzamento de 80%, mutação simples e taxa de mutação de 4%). E dez testes com o AG modificado (inicialização heurística, cruzamento proposto, taxa de cruzamento proposto de 80%, mutação proposta e taxa de mutação proposta de 4%, busca local, taxa de busca local de 5%). O gráfico 6 apresenta o resultado encontrado. Para todos os resultados, tanto com AG original quanto com AG modificado, o número de violações foi igual a zero. Pode-se verificar que em todos os casos os resultados do AG modificado foram bem melhores do que o AG original (minimizam a função objetivo).

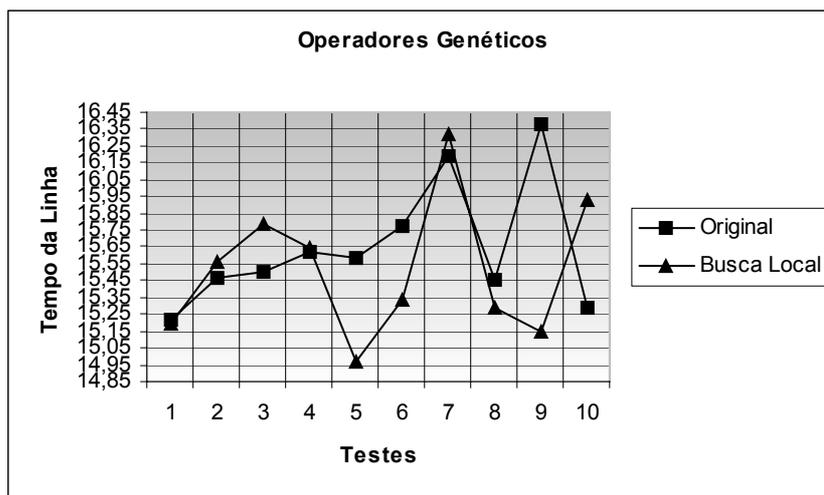


Gráfico 5. Testando a busca local proposta (melhor indivíduo encontrado).

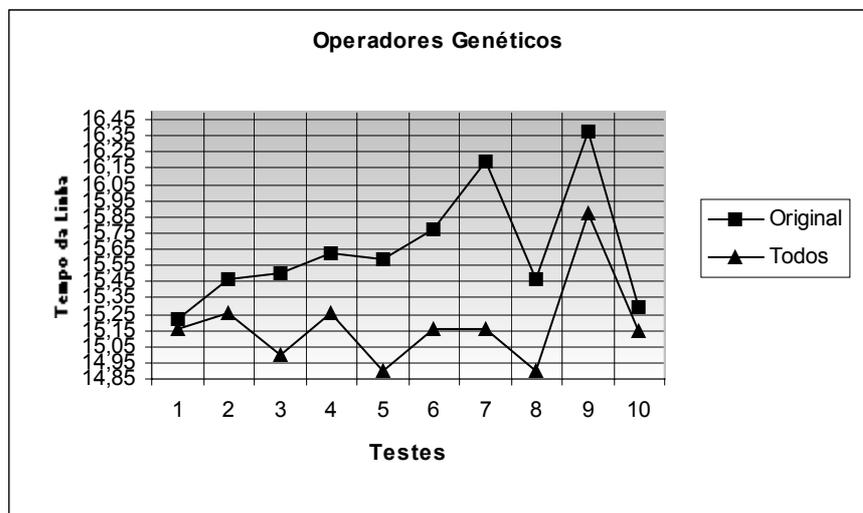


Gráfico 6. Testando todos os operadores propostos (melhor indivíduo encontrado).

Para testar o problema da diversidade genética, novos casos de testes foram realizados, agora verificando o melhor cromossomo encontrado após **50 gerações**, com **população de tamanho 50**. Dez testes foram realizados com o AG original (população inicial aleatória, cruzamento de um ponto, taxa de cruzamento de 80%, mutação simples e taxa de mutação de 4%). E dez testes com o AG modificado (inicialização heurística, cruzamento proposto, taxa de cruzamento proposto de 80%, mutação proposta e taxa de mutação proposta de 4%, busca local, taxa de busca local de 5%). O gráfico 7 apresenta o resultado encontrado.

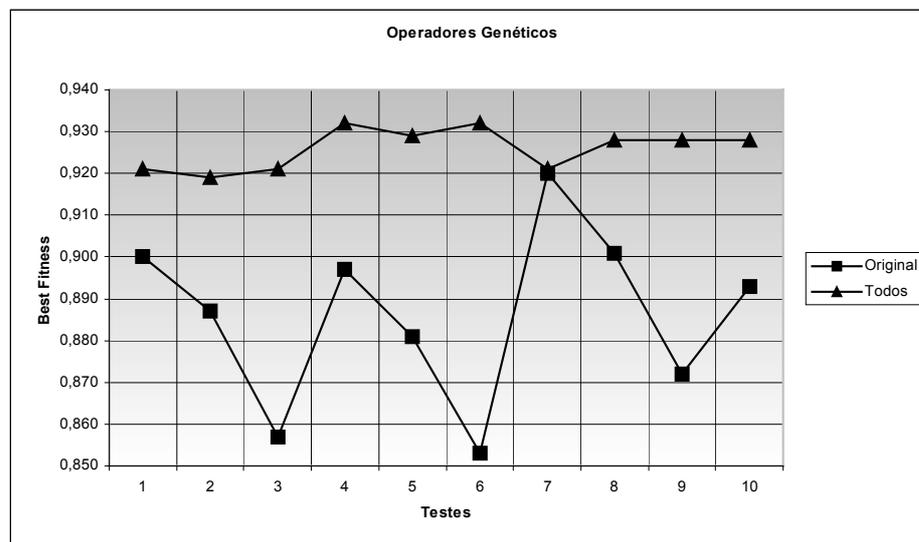


Gráfico 7. Testando todos os operadores propostos (melhor indivíduo encontrado).

Percebe-se claramente que a eficiência e a regularidade do AG original foi grandemente afetada pela diminuição da população e do número de gerações enquanto o AG modificado manteve sua regularidade e bom desempenho.

6. CONCLUSÃO

Analisando-se os resultados dos vários casos de testes realizados verificou-se que a estratégia de geração aleatória da população inicial resultou em desempenho desfavorável para AG. Isto se deve ao fato de gerar soluções não factíveis (cromossomos com mais de duas atividades em um mesmo posto). O grande número de elementos irrealizáveis numa população acabava por reduzir sua diversidade e provocava rápida convergência para ótimos locais. Para solucionar este problema uma inicialização heurística foi implementada apresentando excelentes resultados. Além disso, os operadores de cruzamento de um ponto e mutação simples também poderiam gerar filhos irrealizáveis. Assim, um novo operador de cruzamento e mutação foi implementado de modo a não gerar indivíduos que desrespeitem a restrição de limitação de atividades nos postos. A inserção de diversidade genética sem gerar filhos irrealizáveis permitiu melhorar bastante os resultados do AG. O operador de busca local permitiu varrer uma extensa possibilidade de combinações de um cromossomo sem perda de tempo na geração e avaliação de elementos irrealizáveis. Com estes novos operadores trabalhando em conjunto e com a opção de critério de parada por estabilização do fitness foi possível obter resultados excelentes com um número bem menor de gerações e tamanho menor de população, reduzindo o tempo de execução do algoritmo. Através do emprego conjunto destes operadores foi possível obter soluções melhores do que aquelas encontradas no artigo original em estudo. O gráfico 8 apresenta a evolução do AG modificado para o melhor cromossomo encontrado. O melhor cromossomo encontrado pelo AGBALIM leva um tempo total de trabalho na linha de **14,90 min**, o que representa uma diminuição de **30,3%**

em relação ao tempo da configuração atual, que é de **21,37 min**. O melhor resultado do artigo em estudo apresentou um tempo total de trabalho na linha de **15,29 min** o que representa uma diminuição de **28,5%** em relação ao tempo da configuração atual.

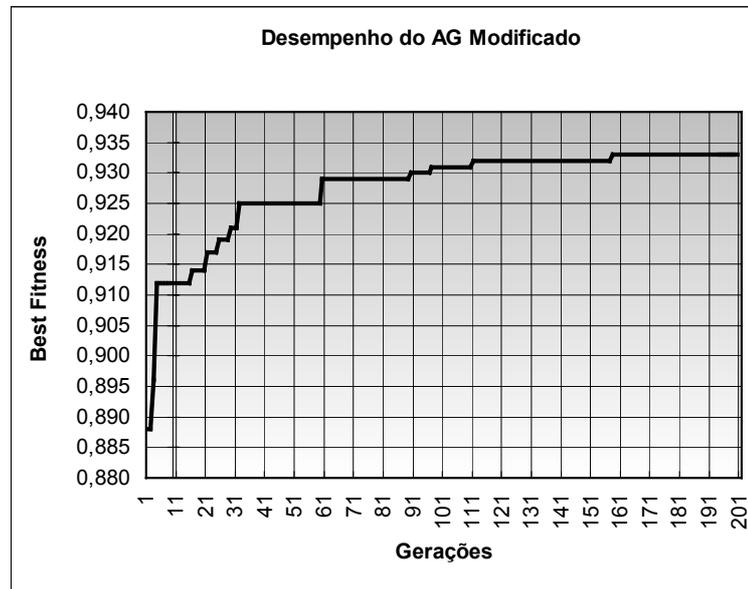


Gráfico 8. Valor do Fitness em cada uma das gerações do AGBALIM

Os resultados encontrados com o algoritmo proposto foram superiores aos apresentados no artigo em estudo comprovando a boa integração entre as várias estratégias escolhidas e desenvolvidas para formarem o corpo do algoritmo genético modificado. O algoritmo AGBALIM demonstrou ser capaz de gerar melhores soluções com menor número de gerações, menor tamanho de população e conseqüente, menor tempo de processamento. Para trabalhos futuros estuda-se a possibilidade do usuário realizar a configuração dos parâmetros do AGsimples para construção da população inicial, bem como a possibilidade de configuração do tamanho da linha de montagem.

7. REFERÊNCIAS BIBLIOGRÁFICAS

[1] Valente, S.A., Lopes, H.S., Arruda, L.V.R. *Algoritmos Genéticos no Problema de Balanceamento de Linhas de Montagem: Uma Aplicação Real numa Linha Automotiva*, disponível em www.delet.ufrgs.br/vsbai/vsbai/artigos/1069.pdf

[2] Nunes, E.O., Conci, A., *Otimização de Linhas de Montagem por Algoritmos Genéticos: Análise, Aperfeiçoamento e Implementação*, painel apresentado no XXXV SBPO, organizado pela SOBRAPO/UFRN - Natal, de 4 a 7 de novembro, 2003.