APLICANDO ALGORITMOS GENÉTICOS E ALGORITMOS CULTURAIS NO SEQUENCIAMENTO DE MÁQUINA COM PENALIDADES POR ANTECIPAÇÃO E ATRASO EM UMA INDÚSTRIA METALÚRGICA

Luis Renato dos Santos
Universidade Federal da Grande Dourados
http://www.ufgd.edu.br
luis.darueshi@gmail.com

Rafael de Araújo Dantas
Universidade Federal da Grande Dourados
http://www.ufgd.edu.br
rafiuskdantas@gmail.com

Honorivaldo Rodrigo A. Silva
Universidade Federal da Grande Dourados
http://www.ufgd.edu.br
honorivaldo@hotmail.com

Celso Gonçalves Camilo-Junior
Universidade Federal de Goiás
Instituto de Informática
http://www.ufg.br
celsocamilo@gmail.com

RESUMO

Um problema muito comum em indústrias de grande porte é a dificuldade de sequenciar uma quantidade elevada de tarefas, de modo que reduza custos e atenda os prazos determinados. Neste artigo apresenta-se um estudo de caso do Problema de Sequenciamento de Máquina Única com Penalização por Atraso e Antecipação em uma Indústria de Estruturas Metálicas. Por se tratar de um problema NP-completo, aplicou-se as metaheurísticas Algoritmo Genético e Algoritmo Cultural para resolvê-lo. Foram feitos testes para analisar o desempenho dos algoritmos e efetuar comparações. Analisando os resultados, o Algoritmo Genético teve melhor desempenho para o caso abordado, com uma economia de $ 2.312,08 a mais do que o AC.

PALAVRAS CHAVE. Algoritmos Genéticos. Algoritmos Culturais, Sequenciamento de Máquina Única. Área de classificação principal: Metaheurísticas.

ABSTRACT

A very common problem in large industries is the difficulty of sequencing a large amount of tasks in order to reduce costs and meet the deadlines. This article presents a case of the single-machine scheduling problem with earliness and tardiness penalties about an Industry of Steel Structures. Because it is an NP-hard problem, we applied the Genetic Algorithm and Cultural Algorithm to solve it. Tests were made to analyze the performance of the algorithms. Analyzing the results, the genetic algorithm performed best for the case dealt with an economy of $ 2,312.08 more than the AC.

KEYWORDS. Genetic Algorithms, Cultural Algorithms, Single-Machine Scheduling Problem . Main area: Metaheuristics.

1. Introdução

Um problema muito comum em muitas empresas é a dificuldade de aperfeiçoar sua linha de produção de modo a reduzir custos, atender os prazos e satisfazer o cliente [6]. A este problema se dá o nome de Problema de Programação da Produção (PPP).

O PPP basicamente consiste em selecionar quais tarefas farão parte do *job* em questão, determinar quando e onde serão processados esses *jobs*, alocá-los a maquina e determinar sua sequência de processamento. Bustamante [3] define uma tarefa como uma operação elementar e que necessita de tempo e recursos para ser processada. No problema de máquina única, o conceito de *job* pode ser igualado ao conceito de tarefas, ou seja, cada *job* representaria então uma operação elementar.

Os principais tipos PPP são: Máquina Única (PSUM), *Job Shop*, *Flow Shop* e *Flow Shop* Permutacional, *Open Shop* e Máquinas Paralelas.

- ***Máquina Única:*** O PSUM define que os jobs deverão ser processados em uma única máquina, sendo que esta máquina só poderá processar uma única tarefa por vez, como ilustra na Figura 1.

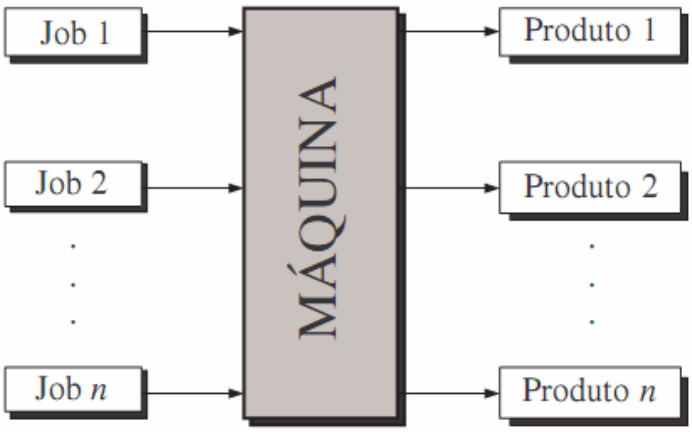

Figura 1. Exemplo do Problema de Máquina Única [9].

- ***Job Shop:*** O problema de *job shop* é definido como um conjunto de *n* tarefas que deve ser processado em um conjunto de *m* máquinas, sendo que cada *job* possuem sua própria sequência de fabricação.
- ***Flow Shop:*** O problema de *flow shop* é um caso especial do *job shop*, possui a mesma definição de que um *job* deve ser processado em um conjunto de *m* máquinas, porém, o *flow shop* define que todos os *jobs* possuem a mesma sequência de fabricação. Um caso especial é o *flow shop* permutacional, onde todas as máquinas executam todas as tarefas na mesma sequência.
- ***Open Shop:*** No problema de *open shop* não há uma sequência de processamento preestabelecida.
- **Máquinas Paralelas**: No problema de máquinas paralelas, as máquinas podem ser idênticas (possuem a mesma velocidade), ou podem ser diferentes (velocidades diferentes). Neste problema sempre há duas ou mais máquinas disponíveis que podem executar qualquer tarefa.

Entre os PSUM, o Problema de Sequenciamento em Máquina Única com Penalidades por Antecipação e Atraso (PSUMAA) é aplicado para melhor sequenciar as tarefas que são produzidas em apenas uma máquina e possuem penalidades por antecipação ou atraso da produção.

Dada a complexidade deste tipo de problema, NP-Completo [5], uma abordagem que tem obtido sucesso é a aplicação das metaheurísticas. Como exemplo, Wan e Yen [19] apresentam uma metodologia baseada na metaheurística Busca Tabu para resolver o PSUMAA com janelas de entrega distintas. Enquanto Hino et al. [7] utilizaram Busca Tabu e Algoritmo Genético para resolver o problema de PSUMAA com datas comuns de entrega, sem considerar tempos ociosos de máquina. Outros autores propõem, ainda, estratégias híbridas [10, 11, 15].

Assim, esse trabalho propõe a aplicação das metaheurísticas Algoritmos Genéticos e Algoritmos Culturais em problema real PSUMAA de uma indústria fabricante de estruturas metálicas.

Testes e comparações são feitas para identificar o melhor algoritmo para o contexto abordado, com o intuito de encontrar uma solução prática para a realidade da indústria.

Este trabalho está estruturado como se segue. Na Seção 2, descreve-se O Problema de Sequenciamento de Máquina Única com Penalidades por Antecipação e Atraso. Na seção 3, é apresentado o estudo de caso abordado no presente artigo. Já na Seção 4, encontra-se toda descrição da codificação do problema e como ele foi tratado. Nas Seções 5 e 6, são descritos os testes e os resultados, além da análise do problema proposto. Por fim, na Seção 7 é realizada a conclusão do artigo.

2. O Problema de Sequenciamento em Máquina Única com Penalidades por Antecipação e Atraso (PSUMAA)

O Problema de Sequenciamento de Máquina Única com Penalidades por Antecipação e Atraso é definido como uma máquina que processa um conjunto I de n *jobs* $(1,...,i,...,n)$, de modo que a cada instante a máquina executa no máximo um *job*. A possibilidade de preempção não é permitida, de modo que, ao iniciar um processamento, a execução não pode ser interrompida, devendo ser finalizada. Todos os *jobs* estão disponíveis no tempo 0, ou seja, trata-se de um sequenciamento estático[2], assim, qualquer *job* pode ser sequenciado no tempo 0.

A programação da Produção possui uma estrutura combinatorial complexa apresentada pela maioria dos problemas reais encontrados nos sistemas. Pois essa estrutura contém medidores de desempenho da solução e estes são classificados por Baker [1] como regulares e não regulares. Seja Z uma função objetivo qualquer, um medidor de desempenho é dito regular se e somente se Z objetivar a minimização e Z somente aumenta se pelo menos uma das datas de término das tarefas aumentarem.

Um critério de decisão regular muito utilizado em problemas PSUM é a minimização do *makespan*, que procura minimizar o instante de término da última operação processada. Este critério leva em conta a velocidade de produção, ou seja, a diferença do tempo entre o início e o término do processamento, porém ignora a consequência das tarefas que terminarem antes ou depois do prazo previsto, Skorin-Kapov *et al.* [18]. Levando em conta os custos consequentes de atraso ou antecipação da produção, pode-se dizer que tanto o atraso quanto a antecipação deveriam ser evitados.

Antecipando a produção, as empresas elevam os custos com estoque e armazenagem, apesar da segurança de não perderem ou extrapolarem os prazos. Já atrasando a produção, os custos com armazenagem e estoque são nulos, mas as empresas descumprem prazos e perdem créditos com clientes. Com o intuito de considerar o atraso e a antecipação, surge o PSUMAA.

Baker *et al.*[2], referenciado pelo trabalho de Bustamante [3], apresentou uma classificação dos PSUMAA. No caso dos problemas exigirem uma data desejada para o atendimento de uma demanda, a classificação foi feita em:

- **Problemas com Data Desejada Comum:** estes problemas possuem sua data de atendimento igual para todas as tarefas.
- **Problemas com Datas Desejadas Distintas:** possuem uma data diferente de atendimento para cada tarefa.
- **Problemas com Tolerância ou Janela de Tempo:** possuem uma espécie de janela de tempo para atendimento da tarefa, ou seja, as tarefas que forem processadas dentro desta janela de tempo não serão penalizadas por antecipação e atraso.

Já no caso dos problemas com medidores de desempenho não regulares, em relação às penalidades por antecipação e atraso, a classificação pode ser:

- **Problemas com Custos Iguais:** quando as penalidades por antecipação e atraso são de valores iguais e são avaliadas de acordo com todo o conjunto de tarefas.
- **Problemas com Custos Diferentes:** quando as penalidades por antecipação e atraso são de valores diferentes e são avaliados também de acordo com todo o conjunto de tarefas.
- **Problemas com Custos Dependentes:** quando as penalidades por antecipação e atraso dependem de cada tarefa e não de todo o conjunto, ou seja, a avaliação é feita de acordo com cada tarefa processada, assim assume-se valores diferentes de atraso e adiantamento para cada tarefa.

3. Estudo de Caso – O Problema Real

O presente trabalho trata de um problema real em uma empresa do setor metalúrgico, que produz estruturas metálicas para construção civil e produtos para serralherias. Como dito antes, o foco deste artigo é apresentar um problema de sequenciamento de produção em máquina única com penalidade por antecipação e atraso. Assim, optou-se pelo sequenciamento de uma máquina de corte da empresa.

A máquina de corte apresentada se caracteriza pelo corte de chapas grossas, geralmente com espessuras acima de 6 milímetros e executa o corte de maneira longitudinal. Seu sistema é baseado em programas com Controle Numérico Computadorizado (CNC), que interpreta os dados codificados em números e letras, e gera uma saída que controle os componentes da máquina, respeitando perfeitamente o que lhe foi programado Kalpakjian [12]. Sua estrutura é baseada em um pórtico com dupla motorização e com trilhos sobre os pedestais que permitem a movimentação por todo o tanque, local onde fica alocada a matéria-prima.

A matéria-prima, manipulada pela maquina de corte, resulta em componentes de estruturas metálicas, que depois de agrupados com outros componentes, às vezes oriundos de outros processos naturais de fabricação em outras máquinas, formarão uma estrutura completa.

Dado o objetivo de cumprir os prazos de entrega e acabar com os custos de estoque, devido aos términos adiantados das operações, percebe-se uma necessidade de uma melhor programação da produção.

4. Codificação do Problema

O Algoritmo Genético (AG) é uma técnica baseada na Teoria da Seleção Natural, de Darwin, utilizada em problemas de procura/descoberta. Possui uma forma de atingir os objetivos

(por evolução), mediante operações sobre uma estrutura codificadora (cromossomo), Marques [13].

Basicamente o AG inicia-se gerando uma população aleatória inicial de soluções; efetua o processo de seleção do individuo mais apto através da avaliação do *fitness*; realiza o cruzamento destes indivíduos; aplica mutação se necessário. Este processo é repetido até que atinja a condição de parada.

Já os Algoritmos Culturais (AC) foram propostos por Robert Reynolds como um complemento à metáfora evolutiva utilizada na Computação Evolutiva, metáfora essa que se concentra nos aspectos genéticos da evolução e na teoria da seleção natural proposta por Darwin. Em contrapartida, os ACs baseiam-se em teorias sociais e arqueológicas que modelam a evolução cultural dos povos [14].

O mecanismo de evolução dos ACs tem dois níveis: o nível micro-evolutivo e o nível macro-evolutivo. No nível micro-evolutivo os indivíduos são descritos em termos de características comportamentais, tem-se a modelagem do Espaço Populacional. No nível macro-evolutivo são armazenados modelos de individuo que armazenam conhecimento adquirido ao longo das gerações e ajudam a guiar o comportamento dos indivíduos em suas populações, o Espaço de Crenças. Considerando os vários tipos de espaço de crenças, esse trabalho utiliza o situacional [4].

O funcionamento do Algoritmo Cultural consiste em inicializar a População e o Espaço de Crenças de forma aleatória ou por meio de alguma técnica que gere indivíduos aceitáveis. Logo em seguida, um critério de término é estabelecido para a execução. É feito a avaliação dos indivíduos gerados inicialmente por uma função de desempenho *obj()* e, em seguida, a execução de seleção dos indivíduos mais aptos por uma função seleção *sel()*. Então é realizado o ajustamento do Espaço de Crenças através da Função de Aceitação *acpt ()*. Em seguida, novos indivíduos são gerados a partir da Função de Influência *infl ()* para a nova população. Esse processo cíclico é executado até o critério de término seja alcançado [17].

Os dois níveis evolutivos compartilham uma ligação, através da qual podem se comunicar (Protocolos de Comunicação), e ditam as regras sobre os indivíduos que podem contribuir com conhecimentos para o Espaço de Crenças (Função de Aceitação), além de estabelecer como o Espaço de Crenças vai influenciar novos os indivíduos (Função de Influência) [2; 4].

Na representação cromossômica de ambos os algoritmos para o problema abordado, cada gene representa uma tarefa a ser processada. A quantidade de genes foi fixada em 50, permitindo assim um sequenciamento de 50 tarefas por vez.

Considera-se ainda, que cada tarefa tem: um custo de atraso e antecipação, o tempo de execução, a data esperada para o término e a prioridade de execução dada pelo usuário.

De acordo com a Figura 2, pode-se verificar que na representação cromossômica a ordem importa, ou seja, a posição do vetor representa a ordem de execução da tarefa e o valor do gene determina a tarefa a ser executada.

1	2	3	4	5	6	7	8	9	10
2	3	4	5	1	10	9	6	8	7

Figura 2. Representação cromossômica.

O cromossomo representa um sequenciamento já pronto, ou seja, uma solução para o problema. Cada cromossomo possui sua avaliação (*fitness*), sendo que a melhor solução é a de maior *fitness*. O cálculo da avaliação é determinado pela função objetivo.

A função objetivo (FO) determina o problema a ser resolvido e, por isso, é o guia do AG durante o processo de otimização. Neste trabalho deve maximizar a FO que é baseada no atraso, adiantamento e preferências não atendidas das tarefas, como mostra a equação 1.

$$FO = E - (w1.Preferência)+(w2.Atraso)+(w3.Adiantamento) \quad (1)$$

Sendo E uma constante e w_i o peso que determina a importância de cada elemento i da função para o problema, sendo que $\sum w_i = 1$. Para o problema abordado, fixou-se E = 5, sendo assim, o maior valor possível para a FO é 4.

Para o cálculo de atraso e adiantamento, compara-se a data programada para término (Tt) com a data esperada para o término da tarefa (Tcorrente + Te), ver equação 2. A variável Tcorrente representa o tempo em que a tarefa atual começará sua execução. Nesta variável já está incluso o tempo de setup entre a tarefa anterior e a atual.

$$Diferença = (Tcorrente + Te) – Tt \quad (2)$$

$$Atraso = Diferença * Cat \quad (3)$$

$$Adiantamento = Diferença * Cad * -1 \quad (4)$$

Caso a diferença calculada, conforme a equação 2, for positiva então houve um atraso, calculado pela equação 3, onde multiplica-se o atraso em minutos pelo custo unitário monetário de atraso da tarefa. Neste caso o valor de custo do adiantamento é zerado. Quando a diferença for negativa, segue-se então a equação 4, que multiplica o valor da diferença pelo custo unitário monetário de adiantamento da tarefa. Contudo, por se tratar de um valor negativo, deve-se multiplicar também por -1.

Caso a tarefa cumpra rigorosamente seu prazo desejado de entrega, a diferença é zerada e o peso desta tarefa na FO é nulo. Lembrando que o problema proposto é de maximização da FO.

O elemento Preferência da FO é definido pelo intervalo de 1 a 5, sendo 1 para tarefas com maior nível de preferência e 5 para tarefas de menor preferência. A prioridade de uma tarefa quanto mais próxima de 1, maior o peso sobre sua posição, ou seja, quando selecionada diferentemente de 5 a posição da tarefa irá influenciar na avaliação de todo indivíduo. O cálculo do custo de preferência é demonstrado na equação 5, onde *maxPr* é denominado o valor máximo de preferência, *pos* é posição da tarefa na execução, *pr* a prioridade da atual tarefa e *tam* o tamanho do indivíduo.

$$CustoPr = ((1-(pr/maxPr))*pos)/tam \quad (5)$$

Existem muitos operadores de cruzamento para a representação inteira com importância da ordem. O utilizado neste artigo foi o operador OX (Figura 3), que escolhe dois *p* pontos de corte através de números randômicos. Os genes situados entre os dois pontos de cortes do cromossomo de um pai são copiados a um filho. O cromossomo filho é completado a partir do segundo ponto de corte com valores do segundo pai. Porém caso o valor a ser copiado do segundo pai já exista no filho este não é copiado. Os genes anteriores ao primeiro ponto, caso haja, só é preenchido após o preenchimento dos valores após o segundo ponto de corte. Na Figura 3 é demonstrado um cruzamento OX, segundo Silva [17]:

$$\left.\begin{array}{l} p1 = (1, 6, \mid 3, 2, 5, \mid 4) \\ p2 = (6, 2, \mid 1, 4, 3, \mid 5) \end{array}\right\} \Rightarrow \left\{\begin{array}{l} f1 = (1, 4, \mid 3, 2, 5, \mid 6) \\ f2 = (2, 5, \mid 1, 4, 3, \mid 6) \end{array}\right.$$

Figura 3. Representação do operador OX.

Como exemplo, copia-se os valores 3, 2 e 5 do *p*1 e insere-os nas mesmas posições do *f*1. Para preencher o restante do cromossomo é copiado os genes restantes a partir do segundo ponto de corte na mesma sequência apresentada em *p*2 para o *f*1, eliminando os valores repetidos. Repare que o próximo valor a ser copiado seria o valor 5 de *p*2, porém este já está incluído em *f*1. Então o próximo valor válido de *p*2 é o 6. Os valores de 2 e 3 também existentes em *f*1 não são copiados e seguindo a mesma ordem é copiado os valores 1 e 4.

Na mutação existem várias técnicas como a Mutação Baseada na Posição, Mutação Baseada na Ordem, Mutação *Scramble* e a Mutação de Modo. Neste problema utiliza-se a técnica da Mutação Baseada na Ordem, onde são escolhidas aleatoriamente duas posições que tem seus valores permutados, Ichihara *et al.* [8].

5. Descrição de Testes

Os testes são baseados no caso real, respeitando a programação padrão da empresa. Assim, os cenários de teste são semelhantes aos que já foram executados pela empresa.

Para tal, utilizou-se um conjunto 50 tarefas, onde cada tarefa se refere a cada execução da máquina, com valores de parâmetros diferentes. As tarefas, de acordo com o caso real, agrupam-se em lotes para delimitar o tempo esperado para término. O cenário utilizado detém 19 lotes abrigando todas as tarefas. A tabela 1 mostra, como exemplo, algumas tarefas.

Tabela 1. Amostra de conjunto de tarefas a ser sequenciadas

Tarefa	***Te***	***Tt***	***Cad***	***Cat***	**Prioridade**	**Setup**
L01_JOB1	42	2700	3	36	5	10
L02_JOB3	191	2300	1	23	5	10
L03_JOB6	187	1800	4	14	5	10
L04_JOB12	105	3000	1	32	5	10
L04_JOB13	23	3000	1	36	2	10
L5_JOB14	38	3600	2	35	5	10
L6_JOB16	260	3400	2	33	5	10
L7_JOB19	209	1500	2	29	5	10
L8_JOB20	199	2100	3	33	5	10
L9_JOB26	139	2900	3	15	5	10
L10_JOB27	170	4200	3	39	5	10
L11_JOB32	20	4800	3	12	1	10
L11_JOB33	71	4800	2	13	5	10
L12_JOB34	86	5300	1	23	5	10
L13_JOB36	130	4100	1	27	5	10
L14_JOB39	65	2800	3	38	5	10

L15_JOB43	260	4500	2	15	5	10
L16_JOB44	280	5000	2	40	5	10
L17_JOB46	290	4000	4	35	5	10
L18_JOB49	320	4300	2	37	5	10
L19_JOB50	140	4300	2	29	5	10

Na Tabela 1 os valores de **T*e*, T*t***, **C*ad*** e **C*at*** correspondem respectivamente ao tempo de execução em minutos, tempo esperado de término em minutos, custo monetário de adiantamento (entre 1 e 4) e ao custo monetário de atraso (entre 10 e 40).

Pode-se visualizar ainda na Tabela 1, que algumas tarefas a serem sequenciadas e executadas apresentam o tempo esperado de término igual. Este fato ocorre, pois estas tarefas correspondem a um mesmo lote cujo é responsável pela data de entrega.

Nota-se que os valores de setup das tarefas são iguais, pois no caso proposto não há distinção de tempo na troca de uma tarefa e outra, permanecendo somente um tempo fixo de manuseio das chapas e ajustes da máquina.

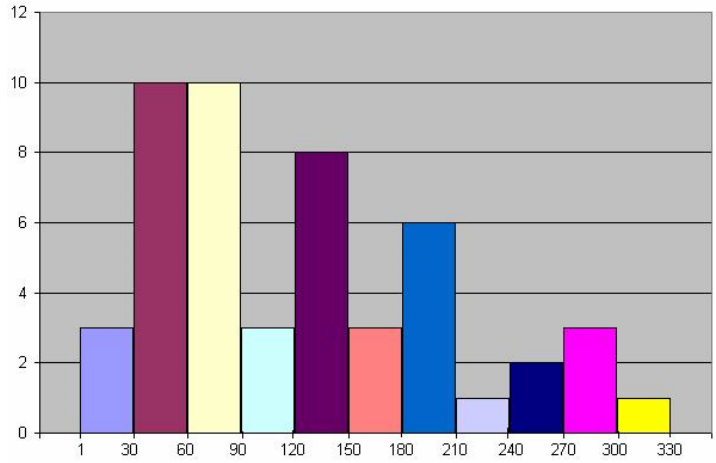

Figura 4. Distribuição de tarefas em intervalos de tempo de execução.

A figura 4 mostra a distribuição das tarefas em relação aos intervalos de tempo de execução. Para esta análise, foram estabelecidos 10 intervalos, sendo cada um com 30 minutos. Percebe-se que os intervalos de 31 a 60 minutos e 61 a 90 minutos são os com a maior quantidade de tarefas, sendo 10 tarefas cada. Além disso, pode-se dizer que a empresa tem a maioria das tarefas com até 210 minutos de execução.

Foram realizados testes preliminares para validação da aplicação e busca dos melhores valores para os parâmetros dos algoritmos. A tabela 2 apresenta os valores utilizados nos testes para cada parâmetro.

Tabela 2. Representação dos valores propostos no teste de desempenho da aplicação.

VALORES PROPOSTOS AOS PARÂMETROS DO AG	
Taxa Mutação	**Taxa Cruzamento**
0,01	0,60
0,005	0,75
0,001	0,85

Os testes revelaram que o melhor desempenho para a Taxa de Mutação é de 0.001 e para a Taxa de Cruzamento é 0,6.

Já no Algoritmo Cultural, os testes se aplicam também à Taxa de Mutação e Taxa de Cruzamento, porém deve-se calcular também a Taxa de Espaço de Crença. Nos testes realizados utilizam-se os parâmetros descritos na tabela 3.

Tabela 3. Representação dos valores propostos no teste de desempenho da aplicação.

VALORES PROPOSTOS AOS PARÂMETROS DO AC		
Taxa Mutação	**Taxa Cruzamento**	**Taxa de Espaço de Crença**
0,01	0.60	0.60
0,005	0.75	0.70
0,001	0.85	0.85

Após a execução dos testes pode-se observar que os resultados apontaram para os seguintes valores de parâmetros: 0.75 para a Taxa de Cruzamento, 0.01 para Taxa de Mutação e 0.85 para a Taxa de Espaço de Crença.

Outros parâmetros utilizados nas duas metaheurísticas são a Quantidade de Gerações, que para ambos é de 500, e o Tamanho de cada População, que são 100 indivíduos, onde cada indivíduo possui 50 genes.

6. Resultados e Análise

Aplicou-se a melhor configuração encontrada anteriormente para cada metaheurística no problema abordado. As tabelas 4 e 5 representam os resultados das aplicações. Os valores citados como QG correspondem à quantidade de gerações efetuadas para se chegar ao melhor resultado encontrado. Cada algoritmo foi executado 20 vezes em execuções independentes.

Tabela 4. Resultado do teste do AG com taxa de mutação de 0.001 e taxa de cruzamento de 0.75.

Teste do Algoritmo Genético		
Mut.: 0.001 Cruz.: 0.75	**Avaliação**	**QG**
Mínimo	3,9808	500
Máximo	3,9921	500
Média	3,9848	500

Tabela 5. Resultado do teste do AC com taxa de mutação de 0.01 e taxa de cruzamento de 0.75 e crença de 0.85.

Teste do Algoritmo Cultural		
Mut.: 0.01 Cruz.: 0.75 Crença: 0.85	**Avaliação**	**QG**
Mínimo	3, 9701	500
Máximo	3, 9788	500
Média	3, 9753	500

As tabelas 6 e 7 mostram a melhor solução encontrada para cada aplicação. Mostra-se também o custo monetário ($) total de cada solução. Como já foi dito, esse valor é calculado de acordo com o custo de tempo de atraso e adiantamento.

Tabela 6. Melhor solução encontrada pelo AG.

Avaliação	QG	Custo total de atraso (minutos)	Custo total de atraso ($)	Custo total de adiantamento (minutos)	Custo total de adiantamento ($)

3,9921	500	21.526	445,17	4.500	885,82
Custo Total ($)					**1.330,99**

Tabela 7. Melhor solução encontrada pelo AC.

Avaliação	**QG**	**Custo total de atraso (minutos)**	**Custo total de atraso ($)**	**Custo total de adiantamento (minutos)**	**Custo total de adiantamento ($)**
3,9788	500	31.102	642,30	14.237	3.000,77
Custo Total ($)					**3.643,07**

Considerando o valor de E usado e também maior valor possível para FO é igual a 4. Analisa-se os valores alcançados e considera-se o desempenho dos algoritmos muito bom, demonstrando a capacidade de ambos em auxiliar no processo de sequenciamento da indústria analisada, problema esse que apresenta-se como complexo.

Comparando os algoritmos utilizados, o Algoritmo Cultural não conseguiu obter melhores resultados do que o Algoritmo Genético para o problema proposto, mesmo com o mecanismo de busca local inerente ao AC. Analisando a diferença entre as avaliações 3,9921 (AG) e 3,9788 (AC) da melhor solução encontrada, percebe-se um delta baixo de 0,0133. No entanto, quando se analisa o impacto dessa diferença em valores monetários ($), percebe-se que é uma diferença considerável, pois o AG obteve uma economia de $ 2.312,08 a mais do que o AC.

Acredita-se que a instância analisada não favoreceu as características do AC, ou ainda, que o tipo de espaço de crença não foi o mais adequado, o que justificaria o desempenho inferior em relação ao AG.

7. Conclusão

Considerando a dificuldade encontrada pelas indústrias de fazer as suas programações da produção, alguns trabalhos têm contribuído para auxiliar a programação, ou mesmo, automatizá-la. Dos vários tipos de escalonamento de tarefas encontrados nas indústrias, um muito complexo é o escalonamento de máquina única com penalidades de atraso e antecipação (PSUMAA).

Por isso, esse trabalho propôs a aplicação do AG e o AC em um estudo de caso real de uma indústria metalúrgica, com o intuito de analisar o benefício que esses algoritmos poderiam trazer para a realidade abordada,

Considerando os resultados, entende-se que o desempenho das metaheurísticas foi bastante satisfatório, pois em um ótimo de 4, o AC obteve FO = 3, 9753 em média, e o AG obteve, em média, FO = 3,9848.

Comparando os algoritmos, percebe-se que o AG obteve um resultado superior, principalmente quando se analisa o custo total monetário, onde o AG economizou $ 2.312,08 a mais do que o AC.

Assim, considera-se que o objetivo de auxiliar a indústria foi atingido, pois houve uma minimização do adiantamento, diminuindo os custos pela manutenção de estoques, e de atraso, diminuindo o descumprimento de prazos e, consequentemente, a insatisfação dos clientes.

Como trabalho futuro, sugere-se a aplicação de testes para identificar a configuração de Espaço de Crença do Algoritmo Cultural que traga melhor desempenho para o algoritmo.

Referências Bibliográficas

1. Baker, K R.: Introduction to Sequencing and Scheduling. Ed. John Wiley e Sons (1974)
2. Baker, K. R., Scudder, G. D.: Sequecing with Earliness and Tardiness Penalties: A Review, Operations Research, v. 38, p. 22–36. (1990)
3. Bustamante, L. M.: Minimização do Custo de Antecipação e Atraso para o Problema de Sequenciamento de uma Máquina com Tempo de Preparação Dependente da Sequência: Aplicação em uma Usina Siderúrgica. Dissertação de mestrado, Programa de Pós Graduação em Engenharia de Produção, UFMG, Belo Horizonte (2006)
4. Cruz, A.V.A., P, M.A.C., Vellasco, M.M.B.R.: Algoritmos Evolutivos com Inspiração Quântica para Problemas com Representação Numérica. Rio de Janeiro, 109p. Tese (Doutorado em Sistemas de Apoio à Decisão)—Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro (2007)
5. Garey, M. R., Johnson, D.S. Computers and Intractability: A Guide to Theory of NP-Completeness. W. H. Freeman and Company, New York – NY. 1979.
6. Gomes, F.R.A.: Algoritmo Genético Aplicado aos Problemas de Sequenciamento Permutacional FlowShop sem e com restrição de espera. Dissertação, Universidade Federal do Ceará (2008)
7. Hino, C. M.; Ronconi, D.P.; Mendes, A. B. "Minimizing Earliness and Tardiness Penalties in a Single-Machine Problem with a Common Due Date". European Journal of Operational Research, v. 160, p. 190–201 (2005)
8. Ichihara, J. A., Gauthier, F. A. O., Barcia, R. M.: Um Algoritmo Genético para a Programação de Projetos Lineares Verticais. XVIII ENGEP (1998)
9. Junior, Aloísio de Castro Gomes, Problema de Sequenciamento em uma Máquina com Penalidades por Antecipação e Atraso: Modelagem e Resolução, Dissertação, Universidade Federal de Minas Gerais (2007)
10. Junior, A. C. G., Carvalho, C. R. V., Munhoz, P. L., Souza, M. J. F., Um Método Heurístico Híbrido para a Resolução do Problema de Sequenciamento em Uma Máquina com Penalidades por Antecipação e Atraso da Produção, ENEGEP (2007).
11. Junior, A. C. G., Carvalho, C. R. V., Munhoz, P. L., Souza, M. J. F., Métodos Heuríticos Baseados Em GRAPS, ILS e VND Aplicados À Resolução do Problema de Squenciamento de Uma Máquina com Penalidades por Antecipação e Atraso da Produção, SPOLM (2007).
12. Kalpakjian, S., Schmid, S. R.: Manufacturing Engineering and Technology, 4º edição, Prentice Hall (2001)
13. Marques, A.: Introdução aos algoritmos genéticos (2005)
14. Reynolds, R.G.: An introduction to cultural algorithm**.** In 3rd Annual Congerence on Evolutionary Programming (1994)
15. Ribeiro, F. F., Souza, M. J. F., Souza, S. R., Técnicas Heurísticas Evolutivas Adaptativas Aplicadas ao Problema de Sequenciamento em Uma Máquina Com Penalização por Antecipação e Atraso da Produção, SPOLM (2009).
16. Rodrigues N. M., Almeida C. P., Goncalves R. A.: Um Algoritmo Cultural para o Problema do Despacho Econômico e Ambiental. In: XIV Escola Regional de Informática da SBC, Guarapuava – Paraná (2007)
17. Silva, A. F.: Algoritmos genéticos: alguns experimentos com os operadores de cruzamento ("Crossover") para o problema do caixeiro viajante assimétrico. XXVI ENGEP (2006)

18. Skorin-kapov, J., Vakharia, A. J.: Scheduling a flow-line manufacturing cell: a tabu search approach, International Journal of Production Research, vol 31, no 7, page 1722 (1993)

19. Wan, G.; Yen, B. P. C. "Tabu Search for Single Machine Scheduling with Distinct Due Windows and Weighted Earliness/Tardiness Penalties". European Journal of Operational Research, v. 142, p. 129–146 (2002).