

UMA ABORDAGEM AO PROBLEMA DE SEQUENCIAMENTO EM UMA MÁQUINA COM PENALIDADES POR ANTECIPAÇÃO E ATRASO DA PRODUÇÃO POR MEIO DE ALGORITMOS EVOLUTIVOS

Renato da Silva Ramos

Discente do curso de Ciência da Computação do UNIFOR–MG

e-mail: renattormz@gmail.com

Fernando Bernardes de Oliveira

Professor Assistente da Universidade Federal de Ouro Preto (UFOP)

e-mail: fbo.fernando@gmail.com

RESUMO

O problema de sequenciamento em uma máquina com penalidades por antecipação e atraso da produção (PSUMAA), objeto de estudo deste artigo, baseado em um trabalho monográfico, faz parte dos problemas de programação da produção. Esse tipo de problema é amplamente aplicável em empresas que trabalham com produção sob encomenda e que sempre necessitam se adaptar às atividades executadas durante o processo produtivo das necessidades dos clientes. Para o problema abordado, considerou-se como restrição as datas para entrega da produção e tempo de preparação da máquina dependente da sequência. É proposto o desenvolvimento de um algoritmo evolutivo com características híbridas para solucionar este problema. Os resultados obtidos pelo algoritmo proposto são comparados com o que foi atingido pelo algoritmo genético clássico, bem como com a literatura. Para o algoritmo evolutivo híbrido, foram implementados três operadores de cruzamento (LOX, OX de 1 ponto e OX de 2 pontos), sendo que o operador OX de 2 pontos apresentou melhor desempenho em relação aos demais. Em comparação ao algoritmo genético clássico, houve considerável melhora nos resultados encontrados pelo algoritmo evolutivo híbrido desenvolvido, atingindo melhora superior a 74% para alguns casos de teste. Em relação à literatura, os resultados obtidos apresentaram melhora superior a 25% para os casos de teste com 20 e 25 tarefas, considerando 10.000 gerações para o algoritmo evolutivo.

Palavras-chave: PSUMAA. Algoritmos Genéticos. Problemas de Sequenciamento. Problemas de programação da produção.

AN APPROACH ABOUT THE PROBLEM OF SEQUENCE OF A MACHINE WITH PENALTIES FROM EARLINESS AND TARDINESS OF PRODUCTION THROUGHT EVOLUTIONARY ALGORITHMS

ABSTRACT

The machine scheduling problem for minimizing earliness and tardiness penalties (PSUMAA), object of survey of this article, based on a monographic, is a part of the problems of production scheduling. This kind of problems is widely applicable in companies which work with production under order and that always need to adapt to the activities performed during the productive process to the needs of the clients. For the problems in subject, it was considered the dates for the delivery of production and time of preparation of the machine depending on the sequence. It is proposed the development of an algorithm evolutionary with hybrid features to resolve this problem. The results obtained by the proposed algorithm are compared with the one that was achieved by classic genetic algorithm, as well as the literature. For the hybrid evolutionary algorithm, three operators were of crossover (LOX, OX if 1 point and OX of 2 points), being that the operator OX of 2 points presented a better performance compared the others. In comparison with the classical genetic algorithm, there was a considerable improvement in the results found by the evolutionary algorithm hybrid developed, achieving a better improvement to 74% for some cases of test. As for the literature, the results obtained presented a better improvement to 25%, for the cases of test with 20 and 25 tasks, considering 10.000 generations for the evolutionary algorithm.

Keywords: Single machine scheduling for minimizing earliness and tardiness penalties. Genetic algorithm. Scheduling problems. Production problems.

1 INTRODUÇÃO

A redução dos custos com o processo produtivo é um dos grandes objetivos das empresas que trabalham com produção de bens sob encomenda, a qual implica na análise de diversos fatores que envolvem a cadeia produtiva das empresas, como os Problemas de Programação da Produção (PPP). Esses problemas consistem em organizar os recursos disponíveis de uma forma que sejam executadas todas as tarefas necessárias com o menor custo possível. O fato desses problemas serem de difícil resolução em sua otimalidade se deve à sua complexidade, já que eles são considerados problemas da classe NP-Difícil. (GOMES JÚNIOR, 2007). Isso implica na aplicação de cada vez mais métodos heurísticos para sua resolução, os quais, apesar de não garantir encontrar a melhor solução, podem se aproximar e muito das soluções ótimas (VON ZUBEN, 2009), o que traz para as empresas alguns benefícios como: redução considerável dos custos e também auxílio nas tomadas de decisão.

(BUSTAMANTE, 2007).

Este artigo aborda o problema de sequenciamento em uma única máquina com penalidades por antecipação e atraso da produção (PSUMAA). Para essa classe de problemas de sequenciamento, considera-se um tempo de *setup* ou tempo de preparação da máquina dependente da sequência de execução das tarefas e uma janela de entrega. Para a solução do problema, foi proposto o desenvolvimento de um algoritmo evolutivo em conjunto com métodos de busca local para a busca de soluções viáveis para o PSUMAA.

2 PROBLEMAS DE PROGRAMAÇÃO DA PRODUÇÃO

A crescente globalização e o aumento da competitividade no mercado mundial fazem com que a preocupação das empresas com a qualidade dos serviços prestados seja sempre considerada na tomada de decisão. A parcela do mercado que trabalha com produção sobre demanda encontra nos PPP um grande desafio, como o de conseguir organizar todos os recursos disponíveis de uma forma que se possa atender toda a demanda de requisições. (LÓPEZ, BARCIA, EYADA, 1995; GOMES JÚNIOR et al, 2007).

A programação da produção consiste em alocar uma quantidade de recursos que possam processar um conjunto de tarefas e assim obter um produto final. O problema consiste em encontrar uma programação ou sequenciamento que possa otimizar todo o processo de produção. O conceito de programação da produção é aplicável a uma série de atividades industriais, como de manufatura e de construção (LÓPEZ, BARCIA, EYADA, 1995), indústrias metalúrgicas, têxteis e de tinta. (GOMES JÚNIOR, 2007).

Os principais elementos que compõem os PPP são:

- (i) Recursos: são os bens disponíveis para a execução do processo, como máquinas, matéria-prima, entre outros.
- (ii) Tarefas: trabalho que demanda de alguma quantidade de tempo e recurso para ser executado.
- (iii) *Job*: sequência de fabricação de um determinado produto que é composta por uma ou mais tarefas. Para alguns problemas, o conceito de *job* coincide com o conceito de tarefa, como no PSUMAA, por exemplo. (GOMES JÚNIOR, 2007).

A maioria dos métodos utilizados para resolução dos PPP pode ser classificada como procedimentos heurísticos, os quais buscam uma solução boa, não havendo garantias de que seja a solução ótima. A utilização de métodos heurísticos para encontrar solução para estes problemas vem crescendo e muito com relação aos métodos exatos. Isso se deve, entre outros fatores, ao menor consumo de tempo e poder computacional. E, na maioria dos casos, encontrar uma solução boa, em um tempo aceitável, pode ser considerado melhor que encontrar uma solução ótima que irá demandar muito mais tempo e poder de processamento.

Outro fator que é levado em consideração no momento de decidir qual modelo de implementação utilizar para a solução destes problemas é o número de variáveis e restrições a se analisar. Algumas destas restrições são comuns a maioria dos problemas, como:

- (i) Cada tarefa pode ser executada por apenas uma máquina, a cada instante de tempo.
- (ii) Cada máquina pode executar apenas uma tarefa em um instante de tempo.
- (iii) Após ser iniciado o processamento de uma tarefa, não pode haver preempção, ou seja, a tarefa já não pode mais ser interrompida até a finalização do processo.

Alguns outros fatores são específicos para algumas situações que são tratadas neste trabalho e serão apresentadas nas próximas seções, como tempo de preparação da máquina e janelas de entrega.

2.1 Problema de sequenciamento em uma máquina

Nesta classe de PPP, uma única máquina irá realizar o processamento de uma quantidade n de tarefas. Em sistemas produtivos, nos quais as máquinas são dependentes umas das outras, representam este tipo de problema. (GOMES JÚNIOR, 2007). Considera-se que, para este problema, todas as tarefas estão disponíveis para processamento no tempo 0 e que após o início de seu processamento a preempção (interrupção) não seja aceita. (SOUZA, 2009). Poderia se aplicar esse modelo de problema em uma indústria que produz tintas, na qual as sequências de produção das tintas seriam os *jobs* que devem ser produzidas em uma única máquina, mas que podem corresponder a tempos de processamentos diferentes dependendo da cor e qualidade do produto desejado.

Para o problema de sequenciamento em uma máquina com penalidades por

antecipação e atraso da produção (PSUMAA), com janelas de entrega e tempo de preparação dependente da sequência, que é o problema abordado por este trabalho, além das características apresentadas anteriormente, considera-se que cada tarefa possui:

P_i : tempo de processamento da tarefa;

E : data de início do período (janela) de entrega da tarefa;

T : data final do período (janela) de entrega da tarefa;

S_{ij} : tempo de preparação da máquina.

Considerando que a execução das tarefas i e j seja sequencial, a máquina pode necessitar de um tempo após terminar de processar a tarefa i , para que o processo de execução da tarefa j ou a própria máquina não seja prejudicada. Esse tempo pode ser utilizado para inspeção ou limpeza da máquina, ajuste de ferramentas entre outros fatores e é chamado de tempo de preparação da máquina ou tempo de *setup*. Em alguns casos o tempo de preparação da máquina é desconsiderado ou incluído no tempo de processamento das tarefas, o que pode afetar a qualidade das soluções encontradas para o problema. Para os casos, em que o tempo de preparação é explícito como no PSUMAA, eles podem ser dependentes ou não da sequência de execução. Para o caso do tempo de preparação da máquina ser dependente da sequência de produção (TPMDSP), o tempo de preparação irá depender tanto da tarefa a ser executada quanto da última tarefa que foi processada. E no caso do tempo de preparação ser independente da sequência, é considerado o mesmo tempo para todas as sequências de tarefas. (GOMES JÚNIOR, 2007).

Outros pontos que devem ser considerados são descritos abaixo:

- (i) É permitido tempo ocioso entre a execução das tarefas;
- (ii) A máquina sempre está pronta para produção no tempo 0, ou seja, não há necessidade de tempo inicial de preparação;
- (iii) Cada tarefa i da sequência apresenta, associada a si, um custo unitário de antecipação α e um custo unitário de atraso β ;
- (iv) Para tarefas finalizadas fora da janela de entrega definida em $[E, T]$ haverá um custo por antecipação ou atraso. Antecipação caso a tarefa seja finalizada antes de E , ou por atraso, caso seja finalizada após T .

O objetivo principal é minimizar a soma ponderada dos custos com a antecipação e com o atraso da entrega das tarefas. O somatório a ser minimizado é representado na Equação 1.

Equação 1 - Função de minimização

$$z = \sum_{i=1}^n (\alpha_i e_i + \beta_i t_i)$$

O custo com a antecipação da tarefa é definido como α_i , β_i representa o custo com o atraso na entrega, e_i e t_i os tempos totais de antecipação e atraso do *job* respectivamente, i representa o índice da tarefa que está sendo avaliada e n representa a quantidade de tarefas que aquela sequência apresenta.

2.2. Trabalhos Relacionados

Diversos trabalhos envolvendo o PSUMAA podem ser encontrados na literatura. Dentre esses, alguns podem ser destacados por sua relevância e modo com que o problema foi abordado. Gomes Júnior et al (2007) aplicaram para o mesmo problema uma solução baseada em GRASP para geração das soluções iniciais, *Iterated Local Search* (ILS) e VND, para a busca local e refinamento das soluções respectivamente. Os autores encontraram soluções boas para problemas de pequenas dimensões (8 a 12 *jobs*). Para problemas de dimensões maiores (15 a 75 *jobs*), as soluções encontradas apresentaram um aumento no tempo de processamento e uma pequena variação na qualidade com outros resultados encontrados na literatura.

Ribeiro et al (2009) propuseram um algoritmo genético adaptativo, utilizando também a metaheurística GRASP para geração das soluções iniciais do AG. As soluções encontradas com essa proposta apresentaram pouca variação em problemas com dimensões de até 30 *jobs*, em relação a outras soluções encontradas na literatura. E para problemas entre 40 e 75 *jobs*, a solução encontrada resultou em soluções melhores que as existentes na literatura. Os resultados daqueles autores foram comparados aos de Gomes Júnior et al. (2007).

Com base em alguns dos trabalhos já desenvolvidos na tentativa de encontrar soluções

para o PSUMAA, conclui-se que a aplicação de técnicas heurísticas como estratégia para solucionar esse problema apresenta um nível de satisfação maior que os métodos matemáticos, não apenas para este problema, mas também para outros problemas de otimização. Os resultados encontrados por essas técnicas produzem soluções viáveis e com qualidade, além do custo e tempo de processamento ser bem menor.

3 ALGORITMO EVOLUTIVO HÍBRIDO

A partir do Algoritmo Genético (AG) clássico, foi proposto um algoritmo evolutivo, com a utilização da técnica de busca local visando melhorar as soluções inicialmente encontradas pelo AG clássico. Abaixo são descritas as etapas do desenvolvimento do algoritmo evolutivo, bem como a inclusão das heurísticas de busca local, aplicadas com o objetivo de melhorar as soluções encontradas inicialmente pelo algoritmo evolutivo.

A representação dos cromossomos foi implementada utilizando-se de vetores de n posições. Cada posição desse vetor representa uma tarefa (gene do cromossomo) a ser processada na máquina. A cada uma dessas tarefas, estão associadas as características especificadas na Seção 2.1.

Os dados das tarefas que deverão ser sequenciadas são recuperados de um arquivo de entrada, cedidos por Souza (2011) e os quais foram base para o trabalho de Gomes Júnior et al (2007). Estes arquivos contêm todas as informações referentes às tarefas, necessárias para execução do AG, como: identificador, tempo de processamento, datas para entrega, custos e tempos de preparação.

Após a recuperação das informações contidas no arquivo de entrada, a população inicial do algoritmo evolutivo é gerada do seguinte modo: (i) são gerados uma quantidade x de indivíduos de acordo com o tamanho da população. (ii) Estes indivíduos, então, são armazenados em um vetor que representa toda a população do algoritmo. (iii) Cada indivíduo da população então sofre trocas randômicas na sua seqüência de tarefas, tornando-se assim as seqüências de execução aleatórias.

A função de avaliação (ou função-objetivo) é definida pela Equação 1, descrita na Seção 2.1. Essa equação tem como critério de otimização a minimização dos custos de

antecipação e atraso da sequência de execução das tarefas. Ela é utilizada para atribuir a cada indivíduo seu custo de execução.

A seleção dos indivíduos é realizada considerando-se o valor de *fitness* (aptidão) dos indivíduos. A aptidão é calculada segundo o valor de custo de cada indivíduo, utilizando-se *ranking* linear. Segundo Lacerda (2009), essa técnica consta em ordenar os indivíduos segundo a função objetivo. Após isso, é atribuído a cada indivíduo um valor de aptidão, segundo a sua posição dentro da população. Para o melhor indivíduo, é atribuído o maior valor possível, e para o pior indivíduo, é atribuído o menor valor. O valor de adaptabilidade de cada indivíduo é dado pela Equação 2.

Equação 2 - Função de *ranking* linear

$$f(x_i) = \text{Min} + (\text{Max} - \text{Min}) \frac{n-i}{n-1}$$

Sendo i o índice do indivíduo, n o tamanho total da população, Min representa o valor mínimo de aptidão que um indivíduo pode possuir e Max representa maior valor que pode ser atribuído à aptidão de um indivíduo.

Foi implementado o método de seleção por roleta, o qual é o mais utilizado neste modelo de algoritmo e já apresentou bons resultados na resolução deste tipo de problema.

Os operadores de cruzamento utilizados foram o *crossover* OX (*Order Operator*) de dois pontos, o *crossover* OX de um ponto e o *crossover* LOX (*Linear Order Crossover*) ou cruzamento de ordem linear. Sugere-se a leitura de Ramos (2010) para maiores informações acerca desses operadores.

A operação de mutação é realizada trocando duas posições do cromossomo. Assim, garante-se que não serão geradas sequências inválidas para o problema. Para cada filho gerado, a partir do operador genético de cruzamento, é sorteada uma probabilidade de se executar o operador de mutação.

Para compor a nova população, foi utilizado o método de substituição de estado estacionário. Neste método, são selecionados não apenas os novos indivíduos da população, mas também aqueles indivíduos da população atual que irão morrer. (STEMMER, 2005). Para proceder desse modo, os filhos gerados pelo processo evolutivo são ordenados do melhor para

o pior segundo a função objetivo do problema e os indivíduos da geração atual são ordenados de forma inversa. Então, para todo indivíduo da geração atual que se encontrar um filho na mesma posição e com custo menor, é realizada a substituição. Dessa maneira, ocorrerá elitismo apenas se houver indivíduos na geração atual que tenha custo menor que os filhos gerados. Assim, para cada geração do algoritmo evolutivo o percentual de substituição dos indivíduos poderá variar, de acordo com a qualidade das novas soluções geradas.

Foi aplicado ao algoritmo evolutivo um método de busca local, baseado na heurística de Descida Randômica, no qual são realizados movimentos aleatórios na solução inicial e a nova solução é aceita apenas se ela possuir um custo menor que a solução inicial. Esse processo é repetido uma quantidade fixa de iterações sem melhora na solução.

O algoritmo de busca local é aplicado sobre um percentual m da população, o qual é fixo para todas as gerações do algoritmo. A solução à qual será aplicado o método de busca local é selecionada aleatoriamente dentro da população corrente. Os movimentos são feitos realizando a troca aleatória de dois *jobs* da sequência e são divididos em níveis. A cada nível, são executadas x trocas sobre a solução inicial, e ao se encontrar um ótimo local melhor que o ótimo local corrente o algoritmo é reinicializado explorando a vizinhança desta nova solução. O algoritmo é finalizado após um número y de movimentos sem melhora na solução obtida até então. Como este método não explora toda a estrutura de vizinhança, não há como garantir que a nova solução encontrada pela busca local seja um ótimo global.

4 RESULTADOS E DISCUSSÃO

Para melhor analisar o desempenho do algoritmo evolutivo híbrido, aplicaram-se testes sobre sequências de 8, 10, 20, 25, 30, 40 e 50 tarefas, utilizando problemas-teste cedidos por Souza (2011). Todas as informações inerentes a essas tarefas, descritas na Seção 2.1 e necessárias para a execução do algoritmo evolutivo, são recuperadas desses problemas-teste.

Os testes foram efetuados sobre populações de 100 indivíduos, com probabilidade de cruzamento e mutação de 80,0% e 5,0% respectivamente. A busca local é executada a cada cinco gerações criadas pelo AG, sobre um percentual de 20,0% da população, definindo-se um número máximo de 20 iterações sem melhora para finalizar a execução da busca local.

Para finalizar o processo de execução do AG, considera-se o número de gerações e são selecionados os cinco melhores indivíduos da última população gerada para fazer parte do resultado final.

Foram executados os testes finais para cada um dos problemas com 100, 1.000 e 10.000 gerações, aplicados a cada operador de cruzamento: OX de 1 ponto, OX de 2 ponto e LOX para o algoritmo evolutivo híbrido. Para o AG clássico, considerou-se apenas um total de 300 gerações em cada instância do problema. Como heurística de refinamento, utilizou-se o método de Descida Randômica. Assim, foram realizados 63 casos de teste. Para cada problema-teste será apresentado, de acordo com o experimento, a diferença com relação ao AG clássico e também com relação aos resultados da literatura. Neste caso, a literatura em questão referenciada tem como base o trabalho de Souza (2011), juntamente com os problemas-testes cedidos. A diferença foi calculada conforme a Equação 3.

Equação 3 - Função de cálculo da diferença entre os resultados

$$D = \left(\frac{RO - RL}{RL} \right) \cdot 100$$

Na Equação 3, **D** representa o percentual de diferença entre os resultados utilizados, **RO** define o melhor custo obtido pelo algoritmo evolutivo híbrido e **RL** o melhor custo obtido pelo algoritmo genético clássico e pela literatura. Caso a equação apresente um resultado negativo, indica melhora em relação à referência.

Levando-se em consideração a quantidade elevada de casos de teste, os resultados obtidos serão avaliados separadamente, segundo o operador de cruzamento aplicado. Ao final, será feita uma análise de desempenho geral do algoritmo evolutivo.

De acordo com os resultados apresentados na TAB. 1, as soluções encontradas pelo algoritmo evolutivo híbrido são mais satisfatórias para o PSUMAA que os resultados obtidos pelo AG clássico para a maioria dos casos de testes aplicados, apresentando variações de até 64,91% para problemas com sequências de 50 tarefas e 10.000 gerações.

Além disso, quanto maior o espaço de busca do problema, mais efetiva é a utilização de uma técnica de busca local. Para problemas nos quais o espaço de busca é menor, como o PSUMAA aplicado a sequências de 8 e 10 tarefas, o algoritmo evolutivo híbrido converge

para as mesmas soluções encontradas pelo AG clássico. Enquanto para sequências de tarefas maiores (20, 25, 30, 40 e 50 tarefas), as quais apresentam um espaço de busca maior, a aplicação de uma heurística de busca local melhora consideravelmente as soluções encontradas em relação ao AG clássico.

Comparando-se os melhores resultados encontrados pelo algoritmo evolutivo com os resultados presentes na literatura, encontrou-se melhores soluções para o PSUMAA para sequências de 20 e 25 tarefas, com uma melhora de 14,95% para sequências de 25 tarefas e 8,04% para a sequência de 20 tarefas. Nesses casos, os melhores resultados foram obtidos quando se utilizou 10.000 gerações. Para sequências de 8, 10 e 40 tarefas, o algoritmo evolutivo híbrido ficou próximo aos resultados da literatura em 9,33%, enquanto para as sequências de 30 e 50 tarefas, a variação entre os resultados chegou até 61,01%.

Tabela 1 - Resultados - *Crossover* OX de 1 ponto.

<i>Jobs</i>	Gerações	Algoritmo Evolutivo	AG Clássico	Diferença (%)	Literatura	Diferença (%)
8	100	5.388	5.388	0,00	4.928	9,33
	1.000	5.388	5.388	0,00	4.928	9,33
	10.000	5.388	5.388	0,00	4.928	9,33
10	100	14.618	14.618	0,00	14.411	1,44
	1.000	14.618	14.618	0,00	14.411	1,44
	10.000	14.618	14.618	0,00	14.411	1,44
20	100	26.913	32.767	-17,87	20.164	33,47
	1.000	24.637	32.767	-24,81	20.164	22,18
	10.000	18.542	32.767	-43,41	20.164	-8,04
25	100	46.608	48.801	-4,49	37.313	24,91
	1.000	33.228	48.801	-31,91	37.313	-10,95
	10.000	31.736	48.801	-34,97	37.313	-14,95
30	100	103.729	156.394	-33,67	43.673	137,51
	1.000	74.557	156.394	-52,33	43.673	70,72
	10.000	70.318	156.394	-55,04	43.673	61,01
40	100	188.455	206.473	-8,73	77.005	144,73
	1.000	81.479	206.473	-60,54	77.005	5,81
	10.000	81.479	206.473	-60,54	77.005	5,81
50	100	364.547	393.227	-7,29	118.220	208,36
	1.000	195.685	393.227	-50,24	118.220	65,53
	10.000	137.974	393.227	-64,91	118.220	16,71

De modo análogo aos resultados encontrados pelo operador de cruzamento OX de 1 ponto, os resultados apresentados na TAB. 2 mostram que, para as sequências de 8 e 10 tarefas, o algoritmo evolutivo híbrido converge para as mesmas soluções que o AG clássico,

mesmo com a modificação do operador de cruzamento.

O operador de cruzamento OX de 2 pontos apresenta uma ligeira melhora nos resultados encontrados pelo operador de cruzamento OX de 1 ponto para sequências de 25 e 40 tarefas, apresentando variações totais com relação ao AG clássico de 41,21% para sequências de 25 tarefas e 74,9% para sequências de 40 tarefas. Para os outros casos de testes aplicados (sequências de 20, 30 e 50 tarefas), a variação entre os resultados encontrados pelo *crossover* OX de 2 pontos diminui. Com relação ao outro operador de cruzamento apresentado, as variações foram de 29,18%, 46,18% e 60,77% respectivamente. Nesses casos, os melhores resultados foram obtidos quando se utilizou 10.000 gerações. Para as sequências de 8 e 10 tarefas, o algoritmo evolutivo híbrido continuou sem modificação nos resultados.

Tabela 2 - Resultados - Crossover OX de 2 pontos.

Jobs	Gerações	Algoritmo Evolutivo	AG Clássico	Diferença (%)	Literatura	Diferença (%)
8	100	5.388	5.388	0,00	4.928	9,33
	1.000	5.388	5.388	0,00	4.928	9,33
	10.000	5.388	5.388	0,00	4.928	9,33
10	100	14.618	14.618	0,00	14.411	1,44
	1.000	14.618	14.618	0,00	14.411	1,44
	10.000	14.618	14.618	0,00	14.411	1,44
20	100	24.673	28.189	-12,47	20.164	22,36
	1.000	21.161	28.189	-24,93	20.164	4,94
	10.000	19.963	28.189	-29,18	20.164	-1,00
25	100	41.737	47.400	-11,95	37.313	11,86
	1.000	30.601	47.400	-35,44	37.313	-17,99
	10.000	27.865	47.400	-41,21	37.313	-25,32
30	100	106.686	109.688	-2,74	43.673	144,28
	1.000	75.561	109.688	-31,11	43.673	73,02
	10.000	59.029	109.688	-46,18	43.673	35,16
40	100	192.923	301.675	-36,05	77.005	150,53
	1.000	80.125	301.675	-73,44	77.005	4,05
	10.000	75.721	301.675	-74,90	77.005	-1,67
50	100	537.848	452.230	18,93	118.220	354,96
	1.000	202.847	452.230	-55,15	118.220	71,58
	10.000	177.416	452.230	-60,77	118.220	50,07

Em relação aos resultados encontrados na literatura, o algoritmo evolutivo desenvolvido, utilizando operador de cruzamento OX de 2 pontos, encontrou melhores soluções para os casos de teste utilizando sequências de 20, 25 e 40 tarefas, com diminuição nos custos encontrados de 1%, 25,32% e 1,67% respectivamente. Nesses casos, os melhores

resultados foram obtidos quando se utilizou 10.000 gerações. Para os testes com sequências de 8, 10, 30 e 50 tarefas, os resultados obtidos foram maiores em até 50,07% para sequências de 50 tarefas, para as sequências de 8, 10 e 30 tarefas, as variações foram de 9,33%, 1,44% e 35,16% no aumento dos custos finais do algoritmo.

Como na utilização dos outros dois operadores de cruzamento estudados, o operador LOX também não obteve modificação nas soluções encontradas pelo AG clássico para sequências contendo 8 e 10 tarefas, como mostrado na TAB. 3. Para os outros casos de teste, com o aumento do espaço de busca, o algoritmo evolutivo apresentou melhoras de 11,29% para sequências de 20 tarefas e 1.000 gerações, 41,21% para sequências de 25 tarefas e 10.000 gerações, 38,97% para sequências de 30 tarefas e 1.000 gerações; e 31,24% e 29,13% para sequências de 40 e 50 tarefas respectivamente, 10.000 gerações. Em relação aos resultados presentes na literatura o algoritmo evolutivo híbrido, com a utilização do operador de cruzamento LOX, apresentou melhora apenas para o caso de teste com sequências de 25 tarefas, com redução de até 25,32% nos resultados encontrados, com 1.000 gerações e 10.000 gerações. Houve um aumento nos custos das sequências encontradas para todos os outros casos de teste em que se aplicou o operador de cruzamento LOX. As maiores variações encontradas foram para sequências de 30 e 40 tarefas, em que o aumento nos custos encontrados foi de 33,28% e 24,41% respectivamente.

Tabela 3 - Resultados - *Crossover* LOX.

<i>Jobs</i>	Gerações	Algoritmo Evolutivo	AG Clássico	Diferença (%)	Literatura	Diferença (%)
8	100	5.388	5.388	0,00	4.928	9,33
	1.000	5.388	5.388	0,00	4.928	9,33
	10.000	5.388	5.388	0,00	4.928	9,33
10	100	14.618	14.618	0,00	14.411	1,44
	1.000	14.618	14.618	0,00	14.411	1,44
	10.000	14.618	14.618	0,00	14.411	1,44
20	100	22.132	23.250	-4,81	20.164	9,76
	1.000	20.625	23.250	-11,29	20.164	2,29
	10.000	23.113	23.250	-0,59	20.164	14,63
25	100	41.737	47.400	-11,95	37.313	11,86
	1.000	30.601	47.400	-35,44	37.313	-17,99
	10.000	27.865	47.400	-41,21	37.313	-25,32
30	100	70.820	95.374	-25,74	43.673	62,16
	1.000	61.377	95.374	-35,65	43.673	40,54
	10.000	58.209	95.374	-38,97	43.673	33,28
40	100	120.134	139.326	-13,77	77.005	56,01

<i>Jobs</i>	Gerações	Algoritmo Evolutivo	AG Clássico	Diferença (%)	Literatura	Diferença (%)
	1.000	96.736	139.326	-30,57	77.005	25,62
	10.000	95.801	139.326	-31,24	77.005	24,41
50	100	286.837	197.546	45,20	118.220	142,63
	1.000	150.254	197.546	-23,94	118.220	27,10
	10.000	139.997	197.546	-29,13	118.220	18,42

O algoritmo evolutivo híbrido apresentou melhoras significativas em relação ao AG clássico, com valores superiores a 74% para alguns casos de testes. Em relação à literatura em questão, apesar de em alguns casos de teste os resultados obtidos estarem abaixo dos resultados presentes na literatura, para alguns casos de teste observou-se melhora superior a 25%, nos quais foram utilizadas 10.000 gerações. A TAB. 4 mostra uma comparação de desempenho entre os operadores de cruzamento, em que o OX de 2 pontos obteve melhores.

Tabela 4 - Comparação de desempenho dos operadores de cruzamento.

<i>Jobs</i>	Crossover OX 1 ponto		Crossover LOX		Crossover OX de 2 pontos	
	AG Clássico	Literatura	AG Clássico	Literatura	AG Clássico	Literatura
8	0,00	9,33	0,00	9,33	0,00	9,33
	0,00	9,33	0,00	9,33	0,00	9,33
	0,00	9,33	0,00	9,33	0,00	9,33
10	0,00	1,44	0,00	1,44	0,00	1,44
	0,00	1,44	0,00	1,44	0,00	1,44
	0,00	1,44	0,00	1,44	0,00	1,44
20	-17,87	33,47	-4,81	9,76	-12,47	22,36
	-24,81	22,18	-11,29	2,29	-24,93	4,94
	-43,41	-8,04	-0,59	14,63	-29,18	-1,00
25	-4,49	24,91	-11,95	11,86	-11,95	11,86
	-31,91	-10,95	-35,44	-17,99	-35,44	-17,99
	-34,97	-14,95	-41,21	-25,32	-41,21	-25,32
30	-33,67	137,51	-25,74	62,16	-2,74	144,28
	-52,33	70,72	-35,65	40,54	-31,11	73,02
	-55,04	61,01	-38,97	33,28	-46,18	35,16
40	-8,73	144,73	-13,77	56,01	-36,05	150,53
	-60,54	5,81	-30,57	25,62	-73,44	4,05
	-60,54	5,81	-31,24	24,41	-74,90	-1,67
50	-7,29	208,36	45,20	142,63	-18,93	354,96
	-50,24	65,53	-23,94	27,10	-55,15	71,58
	-64,91	16,71	-29,31	18,42	-60,77	50,07

Esse operador apresentou melhoras em relação à literatura para sequências de 20, 25 e 40 tarefas, nos casos em que são utilizadas até 10.000 gerações para o algoritmo evolutivo. Além disso, observa-se que, com o aumento do número de gerações, o algoritmo evolutivo

híbrido tende a melhorar os resultados. Isso indica que o algoritmo ainda não atingiu um ponto de iterações sem melhora, podendo ainda reduzir as diferenças entre os referenciais adotados. A literatura em questão determina a melhor data para se iniciar o processamento de cada *job*, de acordo com a janela de entrega do mesmo. Mesmo neste trabalho não sendo utilizado o ADDOIP, o algoritmo evolutivo híbrido conseguiu reduzir significativamente os resultados em relação à literatura, como descrito acima.

5 CONCLUSÃO

O problema de sequenciamento em uma máquina com penalidades por antecipação e atraso da produção (PSUMAA) abordado por este trabalho considera janelas de entregas para entrega das tarefas e tempo de preparação da máquina dependente da sequência. Esta classe de problemas de sequenciamento é amplamente aplicável em empresas que trabalham com produção sob encomenda, nas quais se deseja que a produção seja entregue exatamente na data esperada, sem aumento de encargos com a manutenção de estoques, no caso da produção antecipar o seu término, ou dos custos que podem ser gerados pelo atraso na entrega da produção para o cliente, como multas contratuais por exemplo.

A utilização de métodos heurísticos aplicados à resolução desse modelo de problemas se torna cada vez mais frequente, devido à dificuldade em se resolver esse problema por meio de métodos exatos e pela qualidade nas soluções geradas pelos métodos heurísticos. Tais métodos, apesar de não garantirem que a solução obtida seja ótima, conseguem encontrar soluções viáveis e com qualidade, quando aplicados a problemas de otimização. Neste trabalho, foi proposto o desenvolvimento de um algoritmo genético em conjunto com a utilização de técnicas de busca local, para verificar a efetividade da utilização de um algoritmo evolutivo híbrido aplicado ao PSUMAA.

A utilização do algoritmo evolutivo híbrido se mostrou eficiente perante a análise dos resultados encontrados por este modelo de método heurístico, em comparação com um algoritmo genético clássico. Os resultados encontrados apresentam melhora de até 74,0% em alguns casos de teste, quando utilizando o algoritmo evolutivo proposto em relação aos resultados encontrados pelo algoritmo genético clássico.

Comparando os resultados encontrados com a literatura, o algoritmo proposto comporta-se de maneira satisfatória, com resultados em alguns casos de teste superiores a 25%, mesmo não sendo utilizado o algoritmo ADDOIP. Isso indica que a implementação desse algoritmo, associado às soluções do algoritmo evolutivo híbrido, pode trazer resultados ainda melhores.

REFERÊNCIAS

- BUSTAMANTE, L. **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.** 2007. Dissertação (Mestrado em Engenharia da Computação) –Universidade Federal de Minas Gerais, Belo Horizonte, 2007.
- GOMES JÚNIOR, A. C. **Problema de sequenciamento em uma máquina com penalidades por antecipação e atraso: modelagem e resolução.** 2007. Dissertação (Mestrado em Engenharia da Computação)-Universidade Federal de Minas Gerais, Belo Horizonte, 2007.
- GOMES JUNIOR, A. C. et al. Métodos heurísticos baseados em GRASP, ILS E VND aplicados à resolução do problema de sequenciamento em uma máquina com penalidades por antecipação e atraso da produção. In: SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL, 2007, Fortaleza. **Anais...** Fortaleza: SBPO, 2007.
- LACERDA, E. **Algoritmos genéticos: aspectos práticos.** 2009. Disponível em: <http://www.dca.ufrn.br/~estefane/metaheuristicas/ag_pratica.pdf>. Acesso em: 11 mar. 2010.
- LÓPEZ, O. C.; BARCIA, R. M.; EYADA, O. **Problema de programação da produção um esquema de classificação.** 1995. Disponível em <<http://www.scielo.br/pdf/prod/v5n2/v5n2a03.pdf>>. Acesso em: 12 mar. 2011.
- RAMOS, R. S. **Algoritmo evolutivo híbrido aplicado ao problema de sequenciamento em uma máquina com penalidades por antecipação e atraso da produção.** 2010. Trabalho de Conclusão de Curso (Graduação em Ciência da Computação)-Centro Universitário de Formiga-UNIFOR-MG, Formiga, 2010.
- RIBEIRO, F. F.; SOUZA, M. J. F.; SOUZA, S. R. Um algoritmo genético adaptativo para o problema de sequenciamento em uma máquina com penalidades por antecipação e atraso da produção. In: ENCONTRO NACIONAL DE INTELIGÊNCIA ARTIFICIAL, 2009, Bento Gonçalves. **Anais...** Bento Gonçalves, 2009.

ROCHA, P. L. **Um problema de sequenciamento em máquinas paralelas não-relacionadas com tempos de preparação dependentes de máquina e da sequência:** modelos e algoritmo exato. 2006. Dissertação (Mestrado em Ciência da Computação)- Universidade Federal de Minas Gerais, Belo Horizonte, 2006.

SOUZA, M. J. F. **Otimização combinatória.** Disponível em: <<http://www.decom.ufop.br/prof/marcone/Disciplinas/OtimizacaoCombinatoria/OtimizacaoCombinatoria.pdf>>. Acesso em: 12 mar. 2009.

SOUZA, M. J. F. **Página pessoal do Prof. Marcone Jamilson Freitas Souza.** Disponível em: <http://www.iceb.ufop.br/decom/prof/marcone/>. Acesso em: 12 mar. 2011.

STEMMER, M. R. **Algoritmos genéticos.** 2005. Disponível em: <http://www.das.ufsc.br/~camponog/Disciplinas/DAS-5341/stemmer/Algoritmos_Geneticos.pdf>. Acesso em: 12 mar. 2011.

VON ZUBEN, F. J. **Computação evolutiva:** uma abordagem pragmática. 2009. Disponível em: <ftp://ftp.dca.fee.unicamp.br/pub/docs/vonzuben/ia013_1s07/tutorialEC.pdf>. Acesso em: 15 abr. 2010.