

Aplicação do problema de open shop em uma linha de inspeção automotiva

Ludimila Aparecida Louzada¹

Ruvier Rodrigues Pereira³

Pedro Henrique Pedrosa de Melo²

Resumo: O presente artigo aborda o problema de sequenciamento ou *scheduling* em sistemas de produção. A problemática é situada numa linha de inspeção final automotiva, onde a produção é caracterizada como sendo do tipo *open shop* com restrição do tipo *no-wait*, no qual é preestabelecido uma sequência dos modelos de carros nos tipos de inspeções, com o intuito de reduzir o *makespan*. Para tanto, são abordadas as principais definições acerca dos sistemas e sequenciamento de produção, enfatizando os ambientes *open shop*. A busca pela solução ótima do problema é feita utilizando método analítico, por meio de tabelas e aplica-se também uma meta-heurística conhecida como método de Colônia de Formigas, apresentando sua definição e as principais características. Como resultado apresenta-se uma solução ótima para o problema, de forma que a meta-heurística empregada e o método analítico evidenciaram um ótimo desempenho no objetivo específico desse trabalho.

Palavras-chave: Open shop. Sequenciamento. Colônia de Formigas.

Introdução

Atualmente, as indústrias encontram-se em um mercado competitivo e inovador, tendo como principal objetivo a otimização de um ou mais processos. Desta

1 Universidade Federal de Goiás – UFG. Regional Catalão, Programa de Pós-Graduação em Modelagem e Otimização. Contato: ludimila_ap7@hotmail.com.

2 Universidade Federal de Goiás – UFG. Regional Catalão, Programa de Pós-Graduação em Modelagem e Otimização. Contato: eng.pedromelo@hotmail.com. Bolsista FAPEG.

3 Universidade Federal de Goiás – UFG. Regional Catalão, Programa de Pós-Graduação em Modelagem e Otimização. Contato: ruvierrodrigues@hotmail.com. Bolsista FAPEG.

forma, o Planejamento e Controle de Produção (PCP) desempenha um papel fundamental nas empresas, principalmente na otimização de processos juntamente com o sequenciamento de produção (*scheduling*), proporcionando maiores lucros, melhores aproveitamentos do sistema, redução de tempo, além de reduzir custos na produção. Assim, o sequenciamento abrange todas etapas do processo produtivo, desde a articulação das matérias primas até o desempenho dos empregados.

Anteriormente, o sequenciamento era tratado especificamente para o processo de produção de itens dentro da indústria, porém com a busca pela qualidade e redução dos gastos com reparo e recall, as empresas estão cada vez mais investindo nos processos de inspeção final de seus produtos. A inspeção nas fábricas é um método utilizado para manter o controle da qualidade, envolvendo testes em cada produto ou em itens aleatórios por meio de técnicas estatísticas. A finalidade geral da inspeção é detectar problemas nos produtos antes que os mesmos deixem a fábrica em direção aos clientes (MARINO, 2006).

Os problemas de sequenciamento estão inseridos em diversos ambientes industriais, como é o caso do *job shop* (Jm), *flow shop* (Fm) e *open shop* (Om). De forma que, em ambientes *open shop* com número de máquinas (m) maior que dois, com restrição *no-wait*, o problema é considerado NP-hard (Sahni e Cho, 1979). Como consequência, é improvável que tais problemas podem ser resolvidos de forma eficiente, isto é, por meio de algoritmos polinomiais. A fim de encontrar uma solução “boa” dentro de um período de tempo aceitável pode-se aplicar as meta-heurísticas.

De acordo com Stutzel (1999), meta-heurística é um método que permite novas buscas em um dado intervalo de soluções além do ótimo local, buscando novas soluções. Esses métodos apresentam vantagens devido a capacidade de tratar de forma eficiente a função objetivo e as restrições do problema, que tendem a convergir para uma solução que é o ótimo global no espaço do problema, embora não garantam uma solução ótima. Esta propriedade de convergência parece ser importante, na medida em que permite a melhoria, de uma forma iterativa das soluções sub-ótimas.

O presente artigo busca minimizar o tempo de conclusão (C_{\max}) de uma linha de inspeção em uma indústria automobilística, no qual o ambiente é do tipo *open shop* com restrição *no-wait*, no qual haverá uma predefinição da ordem em que os modelos dos carros devem ser inspecionados, bem como não será permitido o estoque de carros dentro do ambiente fabril que tiverem uma sequência na inspeção.

Desta forma, pretende-se resolver analiticamente o problema, visto que serão considerados poucos tipos de inspeções e poucos modelos de carros para serem inspecionados, possibilitando assim aplicação desse método. Pretende-se também aplicar uma meta-heurística bio-inspirada conhecida como Colônia de Formiga

(ACF) a fim de se verificar a sua eficiência na resolução de problema de sequenciamento, contribuindo para que, em trabalhos futuros, se possa aplicar esse método em problemas reais com proporções extensas e com maior complexidade.

Portanto, esse trabalho apresenta um referencial teórico acerca do problema de sequenciamento de produção – com ênfase no *open shop*, abordando as principais características da meta-heurística ACF, bem como analisa e compara a solução analítica com a aplicação da meta-heurística para o problema proposto pela metodologia.

1 **Scheduling** ou Sequenciamento da produção

O *scheduling* ou sequenciamento da produção pode ser definido como a busca pela determinação de uma sequência para as tarefas com um dado recurso. Logo, pode se dizer que o *scheduling* trata de um processo decisório com a meta de otimizar, sendo neste caso, otimizar o processo produtivo (PINEDO, 2010). Segundo Talma (2007), os principais objetivos do problema de sequenciamento são: cumprimento dos prazos e datas para entrega, minimização do tempo para o fluxo dos estoques intermediários e busca pela maximização da capacidade produtiva. As combinações destes problemas também são bastante comuns.

Os problemas de sequenciamento são de mesma natureza de problemas de otimização, visto que, em suma, sua solução se resume a minimizar ou maximizar algumas características do processo produtivo. Devido aplicação em grandes escalas de produção, os empregos de técnicas computacionais se tornam uma alternativa de alta relevância, sendo já de grande utilização.

Alguns problemas de sequenciamento são mais comuns e abrangem diversos seguimentos fabris, como é o caso do *job shop* (Jm), *flow shop* (Fm) e *open shop* (Om). No primeiro cada tarefa possui sua própria sequência de processamento dentro do conjunto de máquinas. Já no *Flow Shop* todas as tarefas possuem uma sequência igual de produção no conjunto de máquinas. Por outro lado, no *open shop* ocorre um processamento sem especificação ou preestabelecimento das tarefas no conjunto de máquinas, logo as tarefas podem ser executadas em qualquer sequência (NAGANO et al. 2004).

1.1 Open Shop

O problema de sequenciamento de tarefas em ambientes open shop consiste em determinar em que sequência um conjunto de n tarefas devem ser processadas em um conjunto de m máquinas (PINEDO, 2010). No entanto, alguns destes tempos de processamento podem ser iguais a zero, visto que a tarefa pode ser ou

não executada em todas as máquinas. Não há restrições no que diz respeito ao encaminhamiento de cada tarefa através do ambiente da máquina.

Em outras palavras, no problema de *open shop*, há duas decisões a serem tomadas, a determinação das rotas de processamento dos postos de trabalho e a sequência de trabalho em cada máquina (PINEDO, 2010). Todos os trabalhos são independentes e estão disponíveis para serem processados no tempo zero. Todas as máquinas são continuamente disponíveis, sendo que cada máquina pode, no máximo, processar uma tarefa de cada vez. É importante ressaltar que o processo de um trabalho em uma máquina não pode ser interrompido.

Dentro do ambiente *open shop* podem ser consideradas vários tipos de restrições de processamento, como por exemplo o *no-wait*, que restringe para que não haja tempo de espera no processamento de uma máquina para a seguinte, se tratando de uma mesma tarefa. Vale ressaltar que, nesse tipo de restrição é permitido apenas espera no início do processamento da tarefa na primeira máquina da sequência determinada (BRANCO, 2006).

O problema *open shop* em comparação aos outros tipos de problemas de *scheduling* está em um crescente desenvolvimento, visto que é um problema complexo presente em vários tipos de ambientes de produção e possui uma ampla diversidade a ser estudada. Brasel *et al.* (1999) discutiu a teoria por trás das sequências irredutíveis do problema de *open shop*. Sequências irredutíveis são conjunto de sequências estruturalmente ótimas no sentido em que há, pelo menos, uma sequência ótima no conjunto para cada instância do tempo de processamento.

Muitos algoritmos exatos, heurísticos e meta heurísticos foram propostos para solucionar problemas de open shop (ANAND & PANNEERSELVAM, 2015; SILVIA & CORREIA, 2013). Os primeiros algoritmos exatos foram propostos por Brucker *et al.* (1997) que foram melhorados por Guéret *et al.* (2000) e Dorndorf *et al.* (2001), conseguindo resolver problemas com maiores números de máquinas e tarefas.

Os resultados mais promissores para o tema têm sido obtidos através da aplicação de meta heurísticas, com aplicação das seguintes técnicas: Algoritmo Genético (PRINS, 2000; KHURI & MIRYALA, 1999; LIAW, 2000), *Simulated Annealing* (ROSHANAIE *et al.*, 2010), Colônia de abelhas (HUANG & LIN, 2011) e Colônia de Formigas e *Beam Search* (BLUM, 2005).

2 Algoritmo de colônia de formigas

O Algoritmo de Colônia de formigas (ACF) é uma meta-heurística baseada no comportamento de uma população de formigas na busca de seu alimento. Esta busca é realizada através da exploração das trilhas com uma substância química depositada pelas formigas durante seu percurso, conhecida como feromônio. Esta

trilha auxilia a formiga a encontrar o alimento e a voltar ao formigueiro, além de ajudar as outras formigas a encontrarem o alimento. Este comportamento simples de seguir trilhas faz emergir um comportamento mais complexo de encontrar trilhas mais curtas entre os dois pontos.

Os primeiros algoritmos ACO surgiram nos anos 1990. O *Ant System* (AS) é o primeiro algoritmo que surgiu inspirado em colônia de formigas, apresentado por Marco Dorigo e colaboradores (DORIGO *et al.*, 1991). Nesse algoritmo, as formigas percorriam caminhos deixando no mesmo uma certa quantidade de feromônio. As formigas escolhiam os caminhos conforme a função probabilística abrangendo a distância deste caminho e a quantidade de feromônio presente no mesmo.

Fazendo analogia ao sistema biológico-social das formigas, no algoritmo, inicialmente, uma formiga escolhe aleatoriamente um dos caminhos disponíveis, quando a mesma encontra seu objetivo, uma função implementada anteriormente avalia a qualidade da solução encontrada. A partir desta avaliação pode ocorrer o depósito de feromônio, sendo que o mesmo é utilizado pelas formigas para fazer a marcação de caminhos. Os algoritmos, bem como as formigas, associam a maior quantidade de feromônio em uma direção com o fato daquela direção possivelmente ser melhor ou mais curta, concentrando naquela direção um maior empenho computacional (KOIDE, 2010).

O pseudo-código para o ACF foi realizado por Dorigo *et al.* (1991) é exposto a seguir:

Algoritmo de Colônia de Formiga

1. Inicialização dos parâmetros (N_{pop} , N_{iter} , d , ρ , α , β e as trilhas de feromônio $\Delta\tau_{i,j}$)
2. Inicialização da população (\vec{x})
3. Atualização da melhor formiga (\vec{x}^*)
4. Avaliação de cada indivíduo da população – Função objetivo ($F(\vec{x})$)
- While** (o critério de parada não for satisfeito) **do**
5. Atualização de cada formiga através da regra de transição de estado ($P_{i,j}^k$)
6. Avalie o valor da função objetivo
7. Armazene o valor da melhor solução
8. Atualize os valores de $\Delta\tau_{i,j}$ e $\tau_{i,j}$
- end**
9. Retorna a melhor solução encontrada.

Em que N_{pop} é o número de população de formigas, N_{iter} é o número de iterações, ρ é a taxa de evaporação, α é a importância relativa da trilha de feromônio, β é a importância relativa da heurística local e $\Delta\tau_{i,j}$ é a quantidade de feromônio que será depositada na aresta (i, j) que depende da função objetivo f .

A regra de transição de estado utilizada é dada pela Equação 1:

$$p_{i,j}^k(t) = \begin{cases} \frac{\tau_{i,j}^\alpha(t)\eta_{i,j}^\beta}{\sum_{j \in J^k} \tau_{i,j}^\alpha(t)\eta_{i,j}^\beta} & \text{se } j \in J^k \\ 0 & \text{caso contrário} \end{cases} \quad (1)$$

na qual, $\eta_{i,j}$ é a informação heurística de (i, j) .

Para aumentar o feromônio entre as tarefas de melhor solução e diminuir as outras, tem-se a Equação 2:

$$\tau_{i,j}(t+1) = (1 - \rho)\tau_{i,j}(t) + \rho\Delta\tau_{i,j}(t) \quad (2)$$

Em que, $\Delta\tau(i, j)$ é dado pela Equação 3:

$$\Delta\tau(i, j) = \begin{cases} \frac{1}{f(s)} & \text{se } (i, j) \in S \\ 0 & \text{caso contrário} \end{cases} \quad (3)$$

3 Metodologia

Esse artigo apresentará um problema que consiste em programar um conjunto n de modelos de veículos, definido como $J = \{1, 2, 3, \dots, n\}$, realizando uma série de k testes de inspeção, indicado por $M = \{1, 2, 3, \dots, k\}$ com o objetivo de minimizar o *makespan* (C_{\max}) total dessa inspeção. Os modelos de veículos estão disponíveis na data zero de programação e podem realizar todos os testes de inspeção em uma sequência predefinida. E com o intuito de diminuir o estoque entre as inspeções não será permitido durante o processamento, as chamadas “filas de esperas” de um mesmo modelo de veículo que seguir de uma inspeção para a seguinte. Vale ressaltar que será permitido o modelo de veículo esperar apenas na primeira inspeção da sequência determinada. Portanto, esse problema conta com a notação de três campos introduzida por Graham *et al.* (1979), sendo representada por $Om | nwt, prec | C_{\max}$.

Com o intuito de otimizar esse problema, foi utilizado o método analítico e para a aplicação computacional será utilizado o método de Colônia de Formigas baseado em funções de penalidades. Desta forma o problema é reescrito sem restrições, alterando a função objetivo de maneira que se alguma restrição for violada, esta é penalizada. Dessa maneira, a solução se mantém na região viável. A função objetiva reescrita é dada pela Equação 4.

$$\min \phi(x, p) = f(x) + rpP(x) \quad (4)$$

onde, $\phi(x, rp)$ é a função pseudo-objetiva, $f(x)$ é a função objetivo original, $P(x)$ é denominada função de penalidade e rp é um escalar associado a $P(x)$. Para penalizar a função de pseudo-objetivo será utilizado o Método da Função de Penalidade Exterior, nesse caso $P(x)$ é dado pela Equação 5.

$$P(x) = \sum_{j=1}^m (\max(0, g_j(x)))^2 + \sum_{k=1}^l (h_k(x))^2 \quad (5)$$

em que, m e l representam respectivamente o número de restrições de desigualdade e de igualdade.

4 Análise de resultados

Para a experimentação computacional e analítica foram considerados 4 tipos de inspeção (M1, M2, M3, M4) para serem realizados em 5 tipos de modelos de carros (J1, J2, J3, J4, J5) em uma linha de *open shop*, as operações serão realizadas conforme a Tabela 11.1.

Tabela 11.1 Ordem Precedência.

M1	J2	J4	J5	J3	J1
M2	J1	J2	J4	J5	J3
M3	J3	J1	J2	J4	J5
M4	J5	J3	J1	J2	J4

Está apresentado na Tabela 11.2 o tempo de processamento dos modelos de carros nos tipos de inspeções que serão realizadas.

Tabela 11.2 Tempo de processamento.

	J1	J2	J3	J4	J5
M1	4	5	2	3	2
M2	2	6	3	4	5
M3	3	2	8	5	3
M4	4	7	8	9	4

Os valores que foram considerados são fictícios, escolhido de forma aleatória em um intervalo $[0,10]$, com o intuito somente de apresentar um problema real.

Com esses dados calculou-se a solução através da Método Analítico e a aplicação no algoritmo de Colônia de Formigas.

4.1 Método Analítico

Com o Método Analítico a solução foi realizada através de tabelas. As tabelas foram preenchidas conforme a ordem de precedência, e quando não obedecem a restrição de *no-wait* na próxima sequência, os valores em destaque deverão ser atualizados para obedecer tal restrição.

Tabela 11.3 Primeira iteração.

	M1	M2	M3	M4
J1	–	2	–	–
J2	5	11	–	–
J3	–	–	8	–
J4	8	–	–	–
J5	–	–	–	4

Na Tabela 11.3 para obedecer a restrição *no-wait*, o J1 na M2 deve ser atrasado, visto que a M3 só estará disponível no tempo 8, logo o tempo do J2 na M1 e M2 serão alterados.

Tabela 11.4 Segunda iteração.

	M1	M2	M3	M4
J1	–	8	11	–
J2	8	14	–	–
J3	–	–	8	16
J4	11	–	–	–
J5	13	–	–	4

Na Tabela 11.4 o J4 na M1 deve ser atrasado, visto que a M2 estará disponível no tempo 14, logo o tempo do J5 na M1 sofrerá modificações.

Tabela 11.5 Terceira iteração.

	M1	M2	M3	M4
J1	–	8	11	–
J2	8	14	–	–
J3	–	–	8	16
J4	14	18		–
J5	16	–	–	4

Na Tabela 11.5 o J1 deverá ser atrasado devido a M4 está ocupada no tempo 11, assim ocorrerá várias alterações.

Tabela 11.6 Quarta iteração.

	M1	M2	M3	M4
J1	–	13	16	20
J2	13	19	21	–
J3	23	–	8	16
J4	19	23	–	–
J5	21	–	–	4

Na Tabela 11.6 para obedecer a restrição *no-wait*, o J5 na M1 deve ser atrasado, não sendo necessário nenhuma outra alteração na tabela.

Tabela 11.7 Quinta iteração.

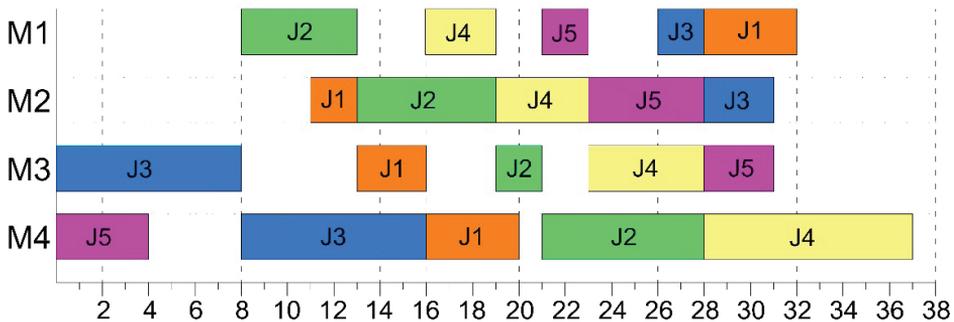
	M1	M2	M3	M4
J1	29	13	16	20
J2	13	19	21	28
J3	25	–	8	16
J4	19	23	28	–
J5	23	28	–	4

Na Tabela 11.7 somente o J3 na M1 deve ser atrasado.

Tabela 11.8 Sexta iteração.

	M1	M2	M3	M4
J1	32	13	16	20
J2	13	19	21	28
J3	28	31	8	16
J4	19	23	28	37
J5	23	28	31	4

Portanto, através da Tabela 11.8 é possível verificar que a tempo de conclusão máximo na linha de inspeção vai do modelo J4 na inspeção M4 com $C_{\max} = 37$ minutos. Para melhor visualizar a solução será apresentado o Gráfico de Gantt (Figura 11.1), que é uma importante ferramenta utilizada para representar a programação da produção de forma clara e objetiva.

**Figura 11.1** Gráfico de Gantt para a solução do problema analítico.

4.2 Algoritmo de Colônia de Formigas

Para um bom desempenho do algoritmo de Colônia de Formiga foram estudados os parâmetros do modelo, de forma que foram encontrados os valores das instâncias através do método de tentativa e erro até encontrar uma solução que respeitasse todas as restrições do modelo.

Como o algoritmo de Colônia de Formigas utiliza o comando *rand* que gera números aleatórios, foi útil que esse número gerado fosse sempre o mesmo para verificar quais restrições não estavam sendo respeitadas e para encontrar os parâmetros que resolvessem o problema. Para conseguir isso foi fixado a semente

(seed) do gerador de números aleatórios. Desta forma os parâmetros que proporcionaram um resultado satisfatório são apresentados na Tabela 11.9.

Tabela 11.9 Parâmetros utilizados.

Parâmetros	Valores
seed	0
N_{pop}	110
N_{int}	6500
ρ	0.5

Através desses parâmetros foi analisado a função objetivo, como pode ser visualizado na Figura 11.2, percebendo que a solução ótima foi encontrada após as 6000 iterações.

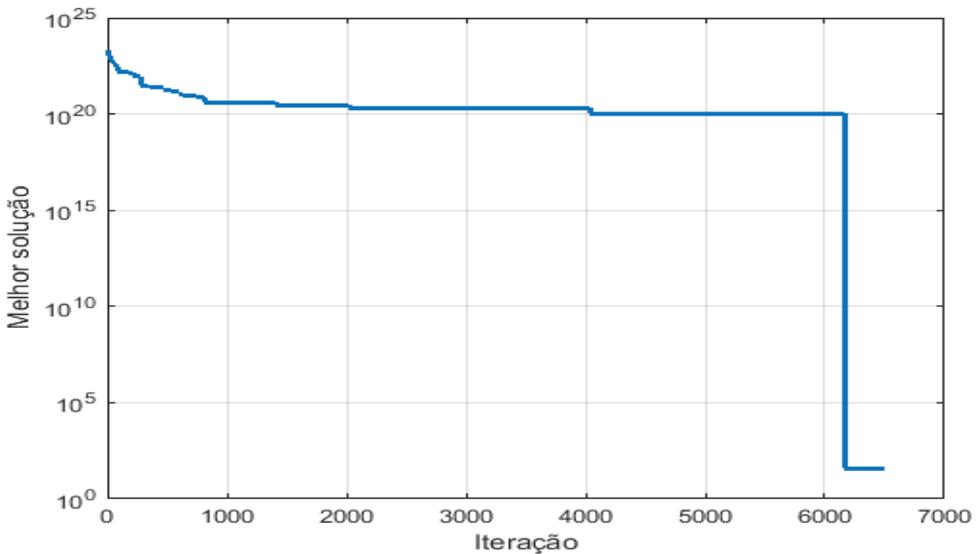


Figura 11.2 Comportamento da função objetivo no algoritmo Colônia de Formiga.

Desta forma, o algoritmo chegou na solução apresentada pelo gráfico de Gantt, conforme mostrado na Figura 11.3.

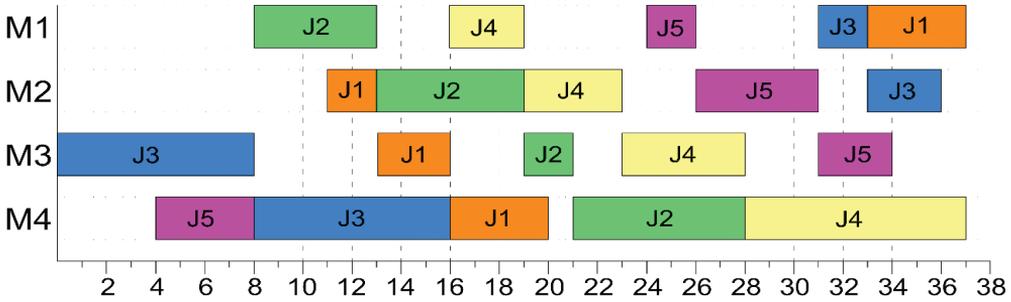


Figura 11.3 Gráfico de Gantt para a solução do problema pelo algoritmo Colônia de Formiga.

Analisando a Figura 11.3 é perceptível que o $C_{\max} = 37$ minutos, no qual todas as restrições foram atendidas e todas as inspeções foram realizadas com sucesso.

Todavia, esse método possui o parâmetro “seed” com alta sensibilidade, no qual uma alteração do valor acarreta em várias violações das restrições do problema, ou seja, não obedeceu a restrição de precedência ou a de *no-wait*, como pode ser visualizado na Tabela 11.10.

Tabela 11.10 Variação do parâmetro “seed”.

Semente	Função Objetivo	Restrições violadas
0	37	0
1	3E+20	3
2	6E+20	6
3	7E+20	7
4	3E+20	3
5	2E+20	2
6	8E+20	8
7	8E+20	8
8	3E+20	3
9	37	0

Através da Tabela 11.10 é visível que a mudança da semente proporcionou resultados diferentes, nos quais a maioria violou as restrições do problema, obtendo, assim, um valor muito alto da função objetivo devido as penalizações que foram aplicadas. Para melhor visualizar a sensibilidade do parâmetro *seed*, observe a figura 11.4.

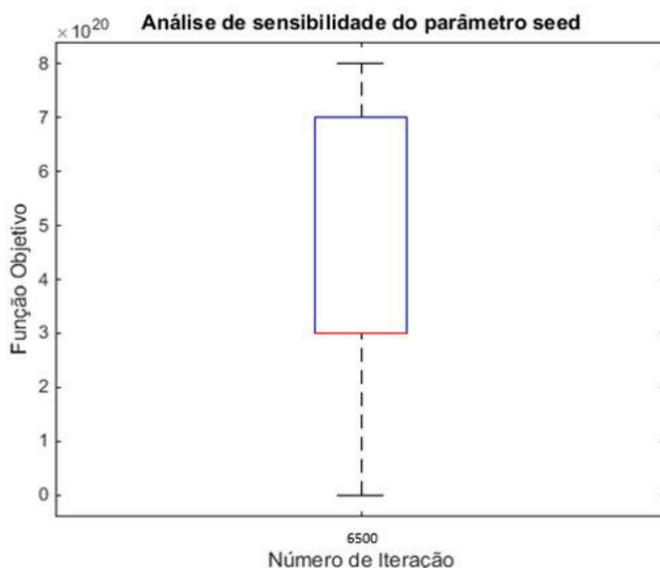


Figura 11.4 *Boxplot* para o algoritmo Colônia de Formigas.

Com base na Figura 11.4 é possível verificar que a média dos resultados obtidos são assimétricos negativos e que possui dois valores discrepantes da função objetivo, de $f = 8E20$ no limite superior e $f = 37$ no limite inferior.

4.3 Comparações

A solução analítica e a computacional apresentaram a solução ótima $C_{\max}^* = 37$ minutos, porém algumas tarefas tiveram o tempo final diferentes. Para analisar as diferenças entre os dois métodos segue a Tabela 11.11.

Tabela 11.11 Diferenças entre o Método Analítico e algoritmo de Colônia de Formigas.

	J1	J2	J3	J4	J5	C_{\max}	Tempo computacional (s)
Método Analítico (min)	32	28	31	37	31	37	–
Algoritmo ACF (min)	37	28	36	37	34	37	973,538
Desvio padrão (%)	2,5	0	2,5	0	1,5	0	–
Porcentagem de diferença relativa (%)	15,625	0	16,129	0	9,677	0	–

Desta forma, percebe-se que os dois métodos atingiram o objetivo e encontraram a solução ótima para o problema. Analisando o tempo final de cada tarefa individual, o método ACF teve eficiência inferior ao método analítico, porém cabe ressaltar que o algoritmo teve como objetivo somente a minimização do tempo de conclusão de todas as tarefas, evidenciando assim, que a meta-heurística necessitaria ser multi-objetiva para minimizar também o tempo individual de cada tarefa.

Conclusões

Nesse artigo, foi tratado o problema de *open shop* com restrição *no-wait* e precedências, com o intuito principal de minimizar o tempo de conclusão do processo de inspeção em alguns tipos de modelos de carros. Com o método analítico e o com o auxílio de software, aplicou-se o algoritmo de Colônia de Formigas proposto alcançando a solução ótima.

A meta-heurística empregada evidenciou um bom desempenho no objetivo de minimizar o C_{max} , mostrando-se bastante flexível, de forma que pode ser verificado uma grande viabilidade para problemas dinâmicos, mostrando necessário que para minimizar também o tempo de conclusão de cada tarefa individual, a função objetivo deve ser multi-objetivo.

Contudo, para problema maiores, com mais restrições e sem uma ordem predefinidas das tarefas é inviável realizar o método analítico. Desta forma, o método ACF mostrou-se apto para resolver problemas de maior complexidade, porém o mesmo terá um custo computacional alto.

Para trabalhos futuros, pretende-se aplicar o algoritmo de Colônia de Formigas, de maneira que seja minimizado o término das tarefas individuais, ou seja, propor uma função multi-objetivo. Pretende-se também aplicar o algoritmo de Colônia de Formigas em outros problemas na área de *scheduling* para melhor avaliar o seu desempenho nessa área específica.

Referências

- ANAND, E.; PANNEERSELVAM, R., Literature Review of Open Shop Scheduling Problems. **Intelligent Information Management**, 7, 33-52, (2015).
- BLUM, C., BEAM-ACO: Hybridizing Ant Colony Optimization with Beam Search: An Application to Open Shop Scheduling. **Computers and Operations Research**, 32, 1565-1591, (2005).
- BRANCO, F. J. C. Avaliação de métodos heréticos para problema no-wait flow shop com critério de minimização da duração total da programação

- da produção.** Dissertação (Mestrado) – Universidade de São Paulo, São Carlos, 2006.
- BRASEL, H., HARBORTH, M., TAUTENHAHN, T. and WILLENIUS, P., On the Set of Solutions of Open Shop Problem. *Annals of Operations Research*, 92, 241-263, (1999).
- BRUCKER, P., HURINK, J., JURISCH, B. and WOSTMANN, B., A Branch and Bound Algorithm for the Open Shop Problem. *Discrete Applied Mathematics*, 76, 43-59, (1997).
- DORIGO, M.; Maniezzo, V.; Colorni, A., Positive Feedback as a Search Strategy, Tech. Report, No. 91-016, *Politécnico di Milano*, Italy, 1991.
- DORNDORF, U.; PESCH, E.; PHAN-Huy, T. Solving the open shop scheduling problem. *Journal of Scheduling*, v. 4, p. 157–174, 2001.
- GRAHAM, R. L.; LAWLER, E. L.; LENSTRA, J.K.; RINNOOY KAN, A.H.G. Optimization and approximation in deterministic sequencing and scheduling: a survey. *Annals of Discrete Mathematics*, v. 5, p. 287-326, 1979.
- GUERET, C., JUSSIEN, N.; PRINS, C., Using Intelligent Backtracking to Improve Branch-and-Bound Methods: An Application to Open-Shop Problems. *European Journal of Operational Research*, 127, 344-354, 2000.
- HUANG, Y.M.; LIN, J.C., A New Bee Colony Optimization Algorithm with Idle-Time-Based Filtering Scheme for Open Shop Scheduling Problems. *Expert Systems with Applications*, 38, 5438-5447, 2011.
- KHURI, S. and MIRYALA, S.R. Genetic Algorithms for Solving Open Shop Scheduling Problems. *EPIA '99 Proceedings of 9th Portuguese Conference on Artificial Intelligence: Progress in Artificial Intelligence*, Évora, Portugal, 21-24 September 1999, 357-368, 1999.
- KOIDE, R. M. *Algoritmo de colônia de formigas aplicado à otimização de materiais compostos laminados*. 2010. 113 f. Dissertação (Mestrado em Engenharia) – Universidade Tecnológica Federal do Paraná, Curitiba, 2005.
- LIAW, C.F., A Hybrid Genetic Algorithm for the Open Shop Scheduling Problem. *European Journal of Operational Research*, 124, 28-42, 2000.

- MARINO, L. H. F. C. Gestão da qualidade e gestão do conhecimento: fatores-chave para produtividade e competitividade empresarial. **XIII SIMPEP**, Bauru, SP, Brasil. p. 1-9, 2006.
- NAGANO, M. S.; MOCCELLIN, J. V.; LORENA, L. A. N. Programação da produção flow shop permutacional com minimização do tempo médio de fluxo. **XXVII SBPO**. p. 23-16, 2004.
- PINEDO, L. M., **Scheduling Theory, Algorithms, and Systems**. New York: Springer, 2010.
- PRINS, C., Competitive Genetic Algorithms for the Open Shop Scheduling Problem. **Mathematical Methods of Operations Research**, 52, 389-411, 2000.
- ROSHANAIEI, V.; ESFEHANI, M.M.S.; ZANDIEH, M. Integrating Non-Preemptive Open Shop Scheduling with Sequence-Dependent Setup Times Using Advanced Metaheuristics. **Expert Systems with Applications**, 37, 259-266, 2010.
- Sahni, S., Cho, Y., 1979. Complexity of scheduling shops with no-wait in process. **Mathematics of Operations Research** 4, 448-457.
- SILVA, H. R. F.; CORREIA, C. S., **Algoritmo Heurístico Para Problema de Sequenciamento de Tarefas em Ambientes de Open-Shop**, 2013.
- STUTZEL, T., **Local Search Algorithms for Combinatorial Problems – Analysis, Algorithms and New Applications**. Germany, 1999.
- TALMA, N. S. **Planejamento de um sistema integrado para sequenciamento de produção em uma malharia**. Juiz de Fora: UFJF/Departamento de Engenharia de Produção, 2006. 40 p. Trabalho de conclusão de curso. Disponível em:< http://www.ufjf.br/ep/files/2014/07/2006_3_Nathalia.pdf>. Acesso em: 01 ago. 2016.