Proposta de uma abordagem híbrida de simulação e heurística para um problema de *flowshop* estocástico

Pedro Villarinho
Pontifícia Universidade Católica do Rio de Janeiro
R. Marquês de São Vicente, 225 - Gávea, Rio de Janeiro - RJ, 22451-900 - Departamento de Engenharia Industrial (DEI)
villarinhopedro@gmail.com

David Pinto
Pontifícia Universidade Católica do Rio de Janeiro
R. Marquês de São Vicente, 225 - Gávea, Rio de Janeiro - RJ, 22451-900 - Departamento de Engenharia Industrial (DEI)
szpdavid@gmail.com

Mauricio Lila
Pontifícia Universidade Católica do Rio de Janeiro
R. Marquês de São Vicente, 225 - Gávea, Rio de Janeiro - RJ, 22451-900 - Departamento de Engenharia Industrial (DEI)
mauriciolila@gmail.com

Javier Panadero
Universitat Oberta de Catalunya
Edifici B3, Parc Mediterrani de la Tecnologia, Avinguda Carl Friedrich Gauss, 5, 08860 Casteldefels, Barcelona, Espanha - Internet Interdisciplinary Institute (IN3)
jpanaderom@uoc.edu

Fernando Cyrino
Pontifícia Universidade Católica do Rio de Janeiro
R. Marquês de São Vicente, 225 - Gávea, Rio de Janeiro - RJ, 22451-900 - Departamento de Engenharia Industrial (DEI)
cyrino@puc-rio.br

Luciana Pessoa
Pontifícia Universidade Católica do Rio de Janeiro
R. Marquês de São Vicente, 225 - Gávea, Rio de Janeiro - RJ, 22451-900 - Departamento de Engenharia Industrial (DEI)
lucianapessoa@puc-rio.br

RESUMO

O artigo propõe uma nova abordagem para problema de *flowshop scheduling* com datas de entrega e ganhos cumulativos em um ambiente de digitalização de livros, a partir da consideração da incerteza presente no processo, representada por uma distribuição de probabilidade associada aos tempos de processamento das tarefas. Para a solução do problema, uma abordagem baseada em

simheuristic foi proposta, constituindo-se, neste caso, da hibridação da metaheurística *Iterated Local Search* e Simulação de Monte Carlo. Este trabalho, portanto, explora a existência de correlação entre as melhores soluções nos ambientes determinístico e estocástico, bem como o impacto na solução de um fator de dispersão na variância associada a uma distribuição de probabilidade. A partir dos experimentos realizados, observou-se que a configuração de tarefas que proporciona a melhor solução no cenário determinístico não necessariamente será a configuração de tarefas que proporcionará a melhor solução no cenário estocástico. Além disso, pode-se observar que o aumento do fator de dispersão acarreta em um aumento da discrepância entre resultados obtidos e também na redução dos resultados encontrados no cenário estocástico.

PALAVRAS CHAVE. ***Flowshop Scheduling*****, Heurística, Simulação.**

Tópicos: MH (Metaheurísticas), SIM(Simulação).

ABSTRACT

This article proposes a new approach to flowshop scheduling problem with delivery dates and cumulative payoff in a book digitization environment, based on the uncertainty present in the scanning process, which is represented by a probability distribution associated with the jobs processing times. For the resolution of this problem, a simheuristic approach was proposed, constituting in this case the hybridization of the metaheuristic Iterated Local Search with the Monte Carlo simulation. This article aimed to study the existence of a correlation between the best solutions in the deterministic environment and in the stochastic environment, as well as the impact on the solution of a dispersion factor in the variance of the probability distribution. From the experiments carried out, it was observed that the configuration of tasks that provides the best solution in the deterministic scenario will not necessarily be the configuration of tasks that provides the best solution in the stochastic scenario. Moreover, it can be observed that the increase in the dispersion factor leads to an increase in the discrepancy between the results obtained and also in the reduction of the results found in the stochastic scenario.

KEYWORDS. Flowshop Scheduling,Heuristic, Simulation

Paper topics: MH (Metaheuristics), SIM (Simulation).

https://proceedings.science/p/106899?lang=pt-br

1. Introdução

O problema de *flowshop scheduling* é um dos mais estudados no campo da Pesquisa Operacional, possuindo aplicações em diversas áreas [Pinedo, 2016], no qual busca-se otimizar uma ou mais medidas de desempenho por meio da permutação de uma sequência de atividades a serem processadas por um conjunto finito de máquinas. Segundo Gupta e Stafford Jr [2006], este problema é caracterizado por um fluxo contínuo e ininterrupto de tarefas através de máquinas em série.

Considerando que este é um problema NP-difícil [Lenstra et al., 1977], poucos métodos exatos foram desenvolvidos para solucioná-lo [Garey et al., 1976], fazendo com que a combinação entre metaheurísticas e outras metodologias venham ganhando popularidade devido a sua eficiência em resolver problemas de otimização combinatória [Ferreira, 2013]. Deve-se considerar ainda que as decisões sobre este problema são tomadas no mundo real, no qual seus componentes podem não apresentar comportamento determinístico, uma vez que a incerteza está frequentemente presente nos mais diversos problemas da vida real [Grasas et al., 2016]. Neste contexto, a abordagem *simheuristic*, desenvolvida pela hibridação de metaheurísticas com técnicas de simulação, tem se mostrado promissora, dado que permite a resolução de problemas de otimização estocástica geralmente presentes nas indústrias [Juan et al., 2018].

Neste trabalho, o problema de *flowshop scheduling* com datas de entrega e ganhos cumulativos, proposto por Pessoa e Andrade [2018], é estudado num contexto de incerteza, considerando-se tempos de processamento estocásticos. Assim, uma abordagem *simheuristic* é proposta como método de resolução pela hibridação da metaheurística *Iterated Local Search* [Lourenço, 1995] e simulação de Monte-Carlo. Conforme discutido em De Armas et al. [2017], a simulação de Monte Carlo é capaz de fornecer os valores esperados dos lucros estocásticos associados à uma solução determinística e guiar o processo de busca de soluções.

Este artigo está organizado da seguinte forma: na Seção 2 será apresentada a definição do problema. Na Seção 3, a abordagem híbrida *simheuristic* é detalhada, bem como os métodos que a compõem. Os experimentos computacionais são descritos na Seção 4, juntamente com seus resultados e análise. Por fim, na Seção 5 serão feitas as considerações finais e indicações de trabalhos futuros.

2. Definição do Problema

Esta Seção apresenta o problema de *flowshop scheduling* com datas de entrega e ganhos cumulativos, em sua versão original (determinística), tal como descrito por Pessoa e Andrade [2018]. Além disso, uma versão estocástica deste problema é proposta, de modo a melhor representar os cenários de incerteza mais próximos à realidade. Desse modo, as soluções para o problema determinístico serão avaliadas considerando a incerteza do ambiente estocástico.

2.1. Versão determinística

No problema de *flowshop scheduling* com datas de entrega e ganhos cumulativos, um conjunto de n tarefas $J = \{1, 2, \ldots, n\}$ deve ser processado, seguindo a mesma ordem, em um conjunto de m máquinas $M = \{1, 2, \ldots, m\}$. Cada tarefa $j \in J$ tem um tempo de processamento $p_{j\ell} > 0$ na máquina $\ell \in M$. Além disso, é atribuída uma data de disponibilidade r_j (*release date*), a partir da qual a tarefa pode ter seu processamento iniciado. A definição deste problema considera ainda um conjunto de K datas de entrega $\mathcal{D} = \{D_1, D_2, \ldots, D_K\}$. Estas datas definem o ganho obtido pela conclusão do processamento de cada tarefa de acordo com a função degrau $\mathcal{F}(C_j)$ descrita abaixo, onde C_j é o instante de conclusão (*completion time*) do processamento da tarefa j na última máquina.

$$\mathcal{F}(C_j) = \begin{cases} 0 & \text{se } D_K < C_j \\ 1 & \text{se } D_{K-1} < C_j \leq D_K \\ \vdots & \\ K & \text{se } 0 < C_j \leq D_1 \end{cases} \quad (1)$$

A resolução deste problema visa, portanto, obter uma solução que maximize $\sum_{j=1}^{n} \mathcal{F}(C_j)$, conforme descrito em Seddik et al. [2013] e Pessoa e Andrade [2018].

A Figura 1 ilustra a solução deste problema considerando uma instância com 2 datas de entrega, 4 tarefas, 2 máquinas e com as seguintes informações de uma instância:

- Datas de disponibilidade: r1 = 14, r2 = 2, r3 = 9, r4 = 7.
- Tempos de processamento: p1 = (5,1), p2 = (3,2), p3 = (3,6), p4 = (6,3).
- Datas de entrega: D1 = 10, D2 = 20.

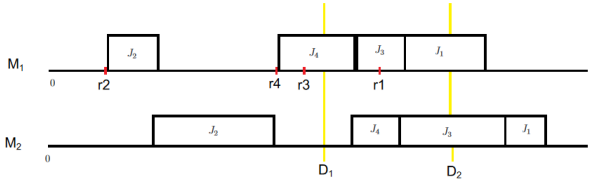


Figura 1: Ilustração de um *flowshop scheduling* com datas de entrega e datas de disponibilidade das tarefas.

Observa-se que a configuração de tarefas ilustrada na Figura 1, proporciona um benefício à empresa contratada no valor de 3. Isto ocorre, pois, de acordo com a função degrau definida anteriormente, a finalização da tarefa 2 na máquina 2 antes da primeira data de entrega gera um benefício igual a 2. A finalização da tarefa 4 na máquina 2 após a primeira data de entrega e antes da segunda data de entrega proporciona um benefício igual a 1. Por fim, a realização das tarefas 3 e 4 após a última data de entrega não gera qualquer benefício.

Este problema foi previamente abordado na literatura por Seddik et al. [2013] e Pessoa e Andrade [2018] no contexto de uma empresa contratada para realizar a digitalização de livros, cujo o objetivo é a maximização de um benefício associado a realização da atividade, considerando as restrições de disponibilidade de cada livros e datas de entrega. Seddik et al. [2013] desenvolveram o artigo sob o contexto de uma única máquina, enquanto que Pessoa e Andrade [2018] estenderam a abordagem para um ambiente de *flowshop* buscando uma melhor representação da configuração de sistemas em contextos reais.

2.2. Versão estocástica

No problema original, descrito por Pessoa e Andrade [2018], os tempos de processamento de cada tarefa são considerados determinísticos, isto é, cada máquina demora exatamente o tempo de processamento informado na execução de uma dada tarefas. Dessa forma, nenhuma incerteza foi considerada no processo, fazendo com que esta modelagem não consiga representar fielmente a realidade, uma vez que a realidade é dinâmica e incerta [Chica et al., 2017]. Com isso, uma

versão estocástica para o problema foi proposta neste artigo. Os tempos de processamento de cada máquina são representados por variáveis aleatórias associadas a uma determinada distribuição de probabilidade. Neste estudo, optou-se pela distribuição log-normal para modelar variáveis não negativas (tempo de processamento). De forma a definir a parametrização utilizada na distribuição de probabilidade em função do tempo de processamento p_{jl}, pode-se utilizar as equações (2) a (4). A distribuição log-normal possui a seguinte função de densidade de probabilidade dependente do tempo de processamento.

$$f(x;\mu_{jl},\sigma_{jl}^2) = \frac{1}{x\sigma\sqrt{2\pi}} exp\left\{-\frac{(lnx-\mu_{jl})^2}{\sigma_{jl}^2}\right\} \tag{2}$$

onde $x \in (0,+\infty)$, e

$$\mu_{jl} = lnE(p_{jl}) - \frac{1}{2}\left(1+\frac{V(p_{jl})}{E(p_{jl})^2}\right) \tag{3}$$

$$\sigma_{jl}^2 = ln\left(1+\frac{V(p_{jl})}{E(p_{jl})^2}\right) \tag{4}$$

Conforme discutido em Juan et al. [2014] e com o objetivo de estudar diferentes cenários de incerteza nesta abordagem, foi considerado que a variância dos tempos de processamento é uma combinação linear do valor esperado, ou seja, $V(p_{jl}) = kE(p_{jl})$.

Na estimação dos lucros estocásticos gerados pela permutação das tarefas neste ambiente, foi empregado o valor esperado dos lucros da simulação considerando a função degrau definida na Seção 2.1.

2.3. Problemas correlatos

A consideração da incerteza nos modelos é de grande importância, a fim de representar melhor a realidade. Na literatura, esta consideração pode ser observada em diversas indústrias e nos mais diversos campos de conhecimento. Com relação aos problemas de *flowshop scheduling*, os primeiros pesquisadores a abordarem o problema considerando incertezas foram Banerjee [1965] e Makino [1965]. Banerjee [1965] desenvolveu um modelo para minimizar o tempo total de atraso, considerando apenas uma máquina, na qual os tempos de processamento são oriundos de uma distribuição de probabilidade. Makino [1965] desenvolveu um modelo com o objetivo de minimizar o *makespan* considerando que os tempos de processamento seguem as distribuições de probabilidade exponencial e de *k-Erlang*. Cunningham e Dutta [1973] consideraram os tempos de processamento exponencialmente distribuídos, a fim de minimizar o tempo de processamento em duas máquinas. Weber [1982] considerou tempos de processamentos estocásticos para minimizar o tempo de fluxo esperado em máquinas em paralelo. Dodin [1996] buscou minimizar o *makespan* por meio de diferentes distribuições de probabilidade – Uniforme, Normal e Exponencial – representando os tempos de processamento. Wang et al. [2005] propuseram uma abordagem com o intuito de minimizar o *makespan*, representando o tempo de processamento por meio de uma distribuição uniforme.

3. Método Proposto

Ao longo dos anos, diversos estudos foram realizados buscando a criação de ferramentas e métodos para resolução de problemas considerando a incerteza. Dentre os estudos realizados, a abordagem *simheuristic* ganhou notoriedade, sendo considerada uma rica ferramenta na abordagem

de problemas de otimização sob incerteza [Chica et al., 2017]. Esta abordagem é composta pela hibridação entre as técnicas de simulação e metaheurísticas, e possui como objetivo a busca por soluções quase ótimas dentro de um espaço de soluções, a partir de uma metaheurística e pelo tratamento da incerteza presente nos processos por meio da técnica de simulação [Chica et al., 2017]. Com isso, soluções para o problema abordado são encontradas em tempos computacionais adequados considerando a incerteza presente.

Diversos campos de estudo tem se beneficiado da utilização dessa abordagem, dentre eles estão: Panadero et al. [2018], que propuseram uma abordagem de *simheuristics* buscando identificar o melhor portfólio para investimentos (Finanças); Hatami et al. [2018] propuseram uma abordagem utilizando *simheuristics* para o problema de *flowshop scheduling* com máquinas em paralelo com tempos de processamento estocásticos, também buscando a minimização do *makespan* (*Flowshop*). Gonzalez-Martin et al. [2018] propuseram uma abordagem de *simheuristics* para solucionar problemas de roteamento com base na Simulação de Monte Carlo e na heurística RandSHARP, esta previamente proposta no trabalho de González-Martín et al. [2012] (Roteamento de veículos).

Neste trabalho, a abordagem *simheuristic*, combina a heurística baseada *Iterated Local Search* (*ILS*) com simulação de Monte Carlo (SMC). A metaheurística *Iterated Local Search* foi inicialmente proposta por Lourenço [1995] para o problema de *jobshop scheduling*. Desde então, diversos problemas de *scheduling*, têm sido resolvidos com aplicação deste método, no qual uma solução localmente ótima é gerada inicialmente e utilizada como ponto de partida para explorar o espaço de soluções. Em seguida, o método iterativamente executa procedimentos de perturbação, busca local e avaliação até que um critério de parada seja satisfeito. O recente trabalho de Lourenço et al. [2019] apresenta tais aplicações bem como detalhes do método.

A heurística ILS utilizada neste trabalho foi desenvolvido por Pessoa e Andrade [2018] para a versão determinística do problema. O Algoritmo 1 descreve a geração da solução inicial. Na linha 1, o método construtivo guloso FF [Fernandez-Viagas e Framinan, 2015], iterativamente inclui uma tarefa na última posição da solução considerando tempos de inatividade (*idle times*) e momento da conclusão das atividades em cada máquina. No *loop* entre as linhas 2 e 6, um método VND (*Variable Neighborhood Descent*) é aplicado sobre a solução recém-gerada com o objetivo de encontrar uma solução aprimorante. O VND utilizado é constituído pelos métodos de busca local *insertion* (linha 5) e *interchange* (linha 6). No método *insertion* explora-se a vizinhança de uma solução tentando inserir uma tarefa em uma posição alternantiva. No método *interchange* por sua vez, considera a troca de posição de duas tarefas, uma com a outra. Ambos os métodos utilizam a estratégia de busca *first improvement*, ou seja, encerram uma busca na vizinhança quando uma primeira solução aprimorante é encontrada. O Algoritmo 1 se completa com as linhas 7 e 8. Na linha 7 uma pequena simulação de Monte Carlo é realizada, a fim de estimar o lucro estocástico e é salva na linha 8 como sendo, até o momento, a melhor solução estocástica.

Algorithm 1: ILS: Geração da Solução Inicial

```
1 Solução Base ← FF(Instância)
2 do
3 |   c ← Lucro(Solução Base)
4 |   BuscaLocal.Insertion(Solução Base)
5 |   BuscaLocal.Interchange(Solução Base)
6 while Lucro (Solução Base) > c;
7 Simulação (Solução Base)
8 Melhor Solução ← Simulação(Solução Base)
```

Em seguida à geração da solução inicial, o algoritmo ILS executa sua fase iterativa, conforme descrito pelo Algoritmo 2, que inclui ainda as etapas de simulação de Monte Carlo.

Algorithm 2: *ILS: Fase Iterativa com Simulação de Monte Carlo*

```
1  Δ ← 0
2  Δ' ← 0
3  Credito ← 0
4  while Condição de Parada = FALSE do
5      Nova Solução ← Solução Base
6      Pertubação (Nova Solução)
7      do
8          c ← Lucro (Nova Solução)
9          BuscaLocal.Insertion (Nova Solução)
10         BuscaLocal.Interchange (Nova Solução)
11     while Lucro(Nova Solução) > c;
12     Δ ← Lucro (Nova Solução) − Lucro (Solução Base)
13     if Δ > 0 then
14         Simulação (Nova Solução)
15         Δ' ← Lucro Estocástico Esperado (Nova Solução) − Lucro Estocástico
            Esperado (Solução Base)
16         if Δ' > 0 then
17             Credito ← Δ
18             Solução Base ← Nova Solução
19             if Lucro Estocástico Esperado (Nova Solução) > Lucro Estocástico
                Esperado (Melhor Solução) then
20                 Melhor Solução ← Nova Solução
21             end
22         end
23     end
24     else if Δ < 0 and Δ >= Credito then
25         Credito ← 0
26         Simulação (Nova Solução)
27         Solução Base ← Nova Solução
28     end
29 end
```

A abordagem proposta tem inicio a partir de uma perturbação e uma busca local, as quais são aplicadas na solução de base visando melhorar a solução corrente (linhas 6 a 11). Se a nova solução encontrada após a etapa inicial for melhor no ambiente determinístico, ou seja, caso o lucro determinístico da nova solução seja melhor do que o lucro determinístico da solução de base atual, então uma simulação curta é executada, a fim de comparar os lucros esperados em um ambiente estocástico (linhas 12 a 14). Se a nova solução for melhor no ambiente estocástico, ou seja, caso o valor esperado do lucro estocástico da nova solução executada na simulação da segunda etapa seja melhor do que o valor esperado do lucro estocástico da solução de base atual, então o crédito (Δ) é atualizado para a diferença entre a nova solução e a solução base no ambiente determinístico. Além disso, a solução base atual é atualizada para a nova solução (linhas 15 a 18). Se a nova solução no ambiente estocástico for melhor do que a melhor solução já encontrada no ambiente estocástico, ou

seja, caso o valor esperado do lucro estocástico da nova solução seja melhor do que o valor esperado do lucro estocástico da melhor solução encontrada , então a melhor solução é atualizada para a nova solução (linhas 19 e 20). Caso a nova solução não seja melhor do que a solução atual no ambiente determinístico e caso a diferença entre a nova solução e a solução base no ambiente determinístico seja maior ou igual ao crédito atual, então uma simulação curta é executada e a solução base é atualizada (linhas 24 a 27).

O método *simheuristic*, tal como descrito em Juan et al. [2014], contempla ainda uma execução longa da SMC. As três melhores soluções determinísticas são armazenadas e, posteriormente, são realizadas suas respectivas simulações com o propósito de analisar o seu comportamento no cenário estocástico.

4. Experimento

4.1. Instâncias e Ambiente Computacional

Para execução dos experimentos computacionais, 20 instâncias foram analisadas. Cada uma possui as informações de número de tarefas, número de máquinas, tempos de processamento nas respectivas máquinas, datas de disponibilidade de cada tarefa e data de entrega. Pode-se observar na leitura da instância o número de datas de entrega e número de tarefas, na instância *k10n100a0.7r0.7*, por exemplo, nota-se que é composta por 10 datas de entrega (K10), 100 tarefas (n100), e possui parâmetros referentes às datas de disponibilidade e entrega das tarefas. As instâncias utilizadas neste experimento podem ser encontradas em: Pessoa e Andrade [2018], `https://data.mendeley.com/datasets/m2wcd42pvy/1`

O programa principal foi escrito em C++, executado em uma máquina com processador Intel i7-3370 de 3.40GHz, 16GB de memória RAM, e sistema operacional Windows 7 Profissional.

4.2. Configuração do Experimento

A partir do desenvolvimento do método proposto, um experimento foi executado considerando três fatores de dispersão na distribuição log-normal (10, 25, 50). Foram executadas 5 rodadas independentes do método variando-se a semente do gerador de números aleatórios. Na busca iterativa por soluções aprimorantes, foi utilizado o critério de parada de 60 segundos, na execução do ILS. Neste experimento buscou-se analisar se o comportamento da permutação de tarefas nos cenários determinístico e estocástico. Para tanto, as três melhores soluções determinísticas foram armazenadas e, posteriormente, para cada uma das soluções, uma simulação foi realizada variando-se, de forma estocástica, os tempos de processamento de cada uma das tarefas. Desde modo, é possível analisar se a configuração de tarefas capaz de gerar o maior lucro no cenário determinístico também é capaz de gerar o maior lucro no cenário estocástico. Com isso, uma pequena simulação e uma longa simulação foram realizadas compostas por 200 e 1000 execuções, respectivamente.

4.3. Resultados e Análise

A Tabela 1 ilustra os resultados obtidos a partir do método proposto. As três soluções determinísticas, estão representadas nas colunas Lucro Det1, Lucro Det2, Lucro Det3. Os lucros obtidos com as soluções nos cenários estocásticos são apresentados nas colunas Lucro Est.1,Lucro Est.2, Lucro Est.3. Observa-se que a melhor solução no cenário determinístico, nem sempre leva ao maior lucro quando exposta ao cenário estocástico. Isto pode ser observado, por exemplo, na instância *k10n100a0.7r0.7*, cuja solução determinística de maior lucro (393) levou ao segundo maior lucro quando analisada no cenário estocástico (363,205). A segunda melhor solução determinística (389) por sua vez, quando analisada em um ambiente estocástico conseguiu gerar o maior lucro (365,281). Entretanto, outro comportamento nos resultados foi observado em algumas instâncias, como por exemplo, a instância *k1n100a0.5r0.3*, onde a configuração de tarefas que gera melhor solução determinística (63) também foi capaz de gerar a melhor solução no cenário estocástico (59,512).

Instância	Fator de Dispersão	Lucro Det.1	Lucro Est.1	Lucro Det.2	Lucro Est.2	Lucro Det.3	Lucro Est.3
k1n100a0.1r0.5	10	21	18,495	20	17,841	19	16,742
	25	21	17,526	20	16,933	19	15,701
	50	21	16,736	20	16,167	19	14,973
k1n100a0.5r0.3	10	63	59,512	62	59,465	61	57,536
	25	63	57,782	62	57,755	61	55,893
	50	63	56,089	62	56,010	61	54,193
k2n100a0.1r0.3	10	30	26,100	29	25,198	28	26,317
	25	30	24,626	29	23,740	28	24,820
	50	30	23,668	29	22,893	28	23,786
k2n100a1.0r1.0	10	141	135,124	139	135,119	107	101,789
	25	141	130,903	139	130,937	107	98,831
	50	141	126,016	139	126,071	107	95,517
k2n100a0.5r0.3	10	98	92,683	97	92,716	95	87,910
	25	98	89,971	97	89,987	95	83,879
	50	98	87,087	97	87,107	95	80,039
k2n100a1.0r0.3	10	150	146,587	149	147,187	144	140,207
	25	150	144,277	149	144,954	144	136,880
	50	150	141,269	149	142,118	144	132,785
k3n100a1.0r0.5	10	201	194,099	199	193,597	188	179,123
	25	201	188,855	199	188,667	188	172,903
	50	201	182,787	199	182,735	188	166,455
k3n100a0.5r1.0	10	119	108,775	117	108,170	113	102,864
	25	119	103,143	117	102,713	113	98,017
	50	119	98,120	117	97,628	113	93,211
k3n100a0.1r1.0	10	36	29.972	35	29,189	34	29,464
	25	36	28,000	35	27,528	34	27,548
	50	36	26,890	35	26,481	34	26,367
k3n100a1.0r1.0	10	183	172,876	181	172,847	145	139,563
	25	183	166,259	181	166,215	145	135,132
	50	183	159,630	181	159,559	145	130,256
k5n100a1.0r1.0	10	285	268,754	280	269,743	276	262,832
	25	285	260,498	280	261,932	276	255,162
	50	285	252,123	280	253,553	276	246,714
k5n100a0.1r0.1	10	54	46,742	53	45,920	52	44,892
	25	54	44,177	53	43,398	52	42,562
	50	54	42,639	53	41,915	52	41,109
k5n100a0.7r0.7	10	231	219,000	229	219,652	216	201,451
	25	231	210,736	229	211,315	216	193,868
	50	231	202,308	229	202,747	216	186,331
k5n100a0.1r0.7	10	62	52,319	61	51,814	60	51,070
	25	62	48,914	61	48,455	60	47,911
	50	62	46,868	61	46,459	60	45,912
k7n100a0.3r0.1	10	168	149,890	165	149,046	162	143,447
	25	168	141,736	165	140,943	162	135,969
	50	168	134,577	165	134,184	162	129,254
k7n100a0.1r1.0	10	73	61,834	71	60,545	70	62,897
	25	73	58,665	71	57,532	70	60,125

Instância	Fator de Dispersão	Lucro Det.1	Lucro Est.1	Lucro Det.2	Lucro Est.2	Lucro Det.3	Lucro Est.3
	50	73	57,005	71	55,905	70	58,503
k10n100a0.7r0.7	10	393	363,205	389	365,281	378	348,581
	25	393	346,511	389	348,935	378	334,037
	50	393	330,872	389	333,591	378	320,122
k10n100a0.1r0.5	10	121	102,556	120	103,469	117	99,397
	25	121	96,913	120	97,648	117	93,776
	50	121	93,411	120	94,197	117	90,539
k10n100a0.5r0.1	10	364	340,082	362	340,679	355	329,572
	25	364	326,451	362	326,945	355	317,654
	50	364	313,806	362	313,944	355	306,067
k10n100a0.5r1.0	10	318	288,564	315	289,285	305	278,306
	25	318	274,323	315	274,937	305	265,300
	50	318	261,449	315	261,824	305	253,004

Tabela 1: Resultados obtidos a partir do método proposto

Outra análise foi realizada buscando avaliar o impacto do fator de dispersão nos resultados encontrados no cenário estocástico. A Figura 2 ilustra, para a instância *k10n100a0.7r0.7*, que o aumento no fator de dispersão leva a um aumento na discrepância entre resultados obtidos no cenário estocástico e na redução dos lucros médios. Este comportamento era esperado, uma vez que com o aumento do fator de dispersão impacta diretamente no aumento da variância, conforme definido na Seção 2.2. Resultados semelhantes foram observados para demais instâncias conforme exposto na Tabela 1.

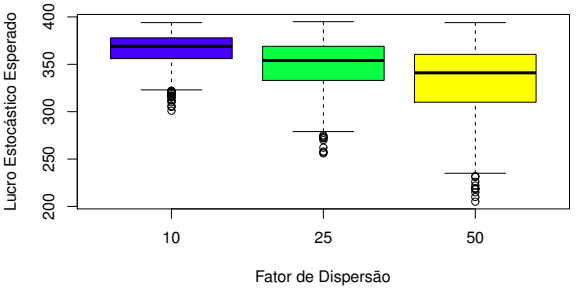


Figura 2: Comportamento na instância *k10n100a0.7r0.7* dos fatores de dispersão na melhor solução no cenário estocástico

5. Considerações Finais e Trabalhos Futuros

Neste trabalho, o problema de *flowshop scheduling* com datas de entrega e ganhos cumulativos foi estudado considerando um ambiente estocástico, onde os tempos de processamento das tarefas foram redefinidos seguindo uma distribuição log-normal. Para sua resolução, foi proposto um método híbrido pela combinação de uma heurística baseada em *Iterated Local Search* e simulação de Monte Carlo.

A partir da execução e análise dos experimentos observou-se que, para algumas instâncias, o comportamento no cenário determinístico se repete no cenário estocástico, isto é, o sequenciamento de tarefas responsável pela maximização do lucro no cenário determinístico se mantém no cenário estocástico. Entretanto, para outras, este comportamento não é observado, fazendo com que uma outra sequência de tarefas seja melhor ao considerar a incerteza no processo. Além disso, foi observado que o fator de dispersão acarreta em uma redução dos lucros estocásticos esperados bem como um aumento na discrepância entre os resultados, o que já era esperado devido à formulação da variância utilizada nesta abordagem e exposta na Seção 2.2.

Para futuros trabalhos, sugere-se analisar o comportamento de outras distribuições de probabilidade na representação dos tempos de processamento, bem como a hibridação com outras metaheurísticas.

Agredecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

Referências

Banerjee, B. (1965). Single facility sequencing with random execution times. *Operations research*, 13(3):358–364.

Chica, M., Juan Pérez, A. A., Cordon, O., e Kelton, D. (2017). Why simheuristics? benefits, limitations, and best practices when combining metaheuristics with simulation.

Cunningham, A. A. e Dutta, S. K. (1973). Scheduling jobs, with exponentially distributed processing times, on two machines of a flow shop. *Naval Research Logistics Quarterly*, 20(1):69–81.

De Armas, J., Juan, A. A., Marquès, J. M., e Pedroso, J. P. (2017). Solving the deterministic and stochastic uncapacitated facility location problem: from a heuristic to a simheuristic. *Journal of the Operational Research Society*, 68(10):1161–1176.

Dodin, B. (1996). Determining the optimal sequences and the distributional properties of their completion times in stochastic flow shops. *Computers & Operations Research*, 23(9):829–843.

Fernandez-Viagas, V. e Framinan, J. M. (2015). A new set of high-performing heuristics to minimise flowtime in permutation flowshops. *Computers & Operations Research*, 53:68–80.

Ferreira, J. S. (2013). Multimethodology in metaheuristics. *Journal of the Operational Research Society*, 64(6):873–883.

Garey, M. R., Johnson, D. S., e Sethi, R. (1976). The complexity of flowshop and jobshop scheduling. *Mathematics of operations research*, 1(2):117–129.

González-Martín, S., Juan, A. A., Riera, D., Castellà, Q., Muñoz, R., e Pérez, A. (2012). Development and assessment of the sharp and randsharp algorithms for the arc routing problem. *AI Communications*, 25(2):173–189.

Gonzalez-Martin, S., Juan, A. A., Riera, D., Elizondo, M. G., e Ramos, J. J. (2018). A simheuristic algorithm for solving the arc routing problem with stochastic demands. *Journal of Simulation*, 12(1):53–66.

Grasas, A., Juan, A. A., e Lourenço, H. R. (2016). Simils: a simulation-based extension of the iterated local search metaheuristic for stochastic combinatorial optimization. *Journal of Simulation*, 10(1):69–77.

Gupta, J. N. e Stafford Jr, E. F. (2006). Flowshop scheduling research after five decades. *European Journal of Operational Research*, 169(3):699–711.

Hatami, S., Calvet, L., Fernández-Viagas, V., Framiñán, J. M., e Juan, A. A. (2018). A simheuristic algorithm to set up starting times in the stochastic parallel flowshop problem. *Simulation Modelling Practice and Theory*, 86:55–71.

Juan, A. A., Barrios, B. B., Vallada, E., Riera, D., e Jorba, J. (2014). A simheuristic algorithm for solving the permutation flow shop problem with stochastic processing times. *Simulation Modelling Practice and Theory*, 46:101–117.

Juan, A. A., Kelton, W. D., Currie, C. S., e Faulin, J. (2018). Simheuristics applications: Dealing with uncertainty in logistics, transportation, and other supply chain areas. In *Proceedings of the 2018 Winter Simulation Conference*.

Lenstra, J. K., Kan, A. R., e Brucker, P. (1977). Complexity of machine scheduling problems. In *Annals of discrete mathematics*, volume 1, p. 343–362. Elsevier.

Lourenço, H. R., Martin, O. C., e Stützle, T. (2019). Iterated local search: Framework and applications. In *Handbook of Metaheuristics*, p. 129–168. Springer.

Lourenço, H. R. (1995). Job-shop scheduling: Computational study of local search and large-step optimization methods. *European Journal of Operational Research*, 83(2):347–364.

Makino, T. (1965). On a scheduling problem. *Operations Research Society of Japan*, 8:32–44.

Panadero, J., Doering, J., Kizys, R., Juan, A. A., e Fito, A. (2018). A variable neighborhood search simheuristic for project portfolio selection under uncertainty. *Journal of Heuristics*, p. 1–23.

Pessoa, L. S. e Andrade, C. E. (2018). Heuristics for a flowshop scheduling problem with stepwise job objective function. *European Journal of Operational Research*, 266(3):950–962.

Pinedo, M. (2016). *Scheduling: theory algorithms and systems*. Springer.

Seddik, Y., Gonzales, C., e Kedad-Sidhoum, S. (2013). Single machine scheduling with delivery dates and cumulative payoffs. *Journal of Scheduling*, 16(3):313–329.

Wang, L., Zhang, L., e Zheng, D.-Z. (2005). A class of hypothesis-test-based genetic algorithms for flow shop scheduling with stochastic processing time. *The International Journal of Advanced Manufacturing Technology*, 25(11-12):1157–1163.

Weber, R. R. (1982). Scheduling jobs with stochastic processing requirements on parallel machines to minimize makespan or flowtime. *Journal of Applied Probability*, 19(1):167–182.

https://proceedings.science/p/106899?lang=pt-br