Sequenciamento de produção em um processo de impressão Digital utilizando algoritmo NSGA II

Felipe R. P. Carneiro, Eduardo G. Carrano

*Programa de Pós-Graduação em Engenharia Elétrica - Universidade Federal de Minas Gerais
Av. Antônio Carlos 6627, 31270-901
Belo Horizonte, MG, Brasil
(e-mail: feliperoncalli@gmail.com; e-mail: egcarrano@gmail.com)*

Abstract: This work addresses the scheduling of tasks in unrelated parallel machines under a multiobjective perspective, more precisely the scheduling of printing jobs in a print shop. The goal is to plan the weekly jobs among the digital printers in order to minimize *makespan* and cost. The considered problem is NP-hard, which makes it unviable to employ exact methods for solving instances with the required dimension. The solution strategy proposed combines a NSGA II algorithm and a local search operator.

Resumo: Este trabalho aborda o problema de sequenciamento de tarefas em máquinas paralelas não relacionadas sob uma perspectiva multiobjetivo, mais precisamente o problema de sequenciamento de tarefas de impressão. O objetivo é planejar o trabalho semanal de impressão em máquinas digitais com o intuito de minimizar tempo e custo. Este problema é do tipo NP-difícil, o que exclui a possibilidade de se aplicar técnicas exatas para instâncias do mesmo porte das consideradas. A estratégia de solução proposta combina um algoritmo NSGA II e um operador de busca local.

Keywords: Optimization; Multiobjective; Pareto; Print; Digital; NSGAI; Hypervolume; Makespan; Cost;

Palavras-chaves: Otimização; Multiobjetivo; Pareto; Impressão; Digital; NSGAI; Hipervolume; Makespan; Custo;

1. INTRODUÇÃO

O sequenciamento pode ser definido como um processo de ordenação e seleção do recurso mais interessante para a execução de uma determinada tarefa, levando em consideração restrições pertinentes (Pinedo, 2008). Este problema tem como objetivos usuais a minimização do tempo total de conclusão (TTC) de todas as tarefas; a minimização do custo associado a execução destas tarefas, além de combinações destes (Morton & Pentico, 1993; Pinedo, 2008).

No aspecto teórico, problemas MPNR são da classe NP-difícil (Mokotoff, 2001), o que provavelmente justifica o grande número de abordagens propostas para sua solução, em especial tendo em conta a minimização do TTC das tarefas. (Mokotoff, 2001; Liaw et al., 2003; Pinedo, 2008; Lin et al., 2011).

Este artigo apresenta um problema de planejamento e controle da produção de uma gráfica. Este problema lida com o sequenciamento eficiente das tarefas a serem executados por cada uma das impressoras. A programação da produção visa minimizar tempo e custo para execução das tarefas.

Uma das principais atividades da programação da produção é o sequenciamento de tarefas, também conhecido como *scheduling*. O sequenciamento de tarefas tem como objetivo

alocar um conjunto de n tarefas, em um conjunto de m máquinas disponíveis, determinando a ordem de processamento com que as tarefas são executadas por cada máquina. Os critérios de otimização normalmente considerados estão relacionados principalmente à utilização eficiente de recursos, redução do tempo combinado com os clientes, diminuição dos custos com a produção, estocagem e entregas dos produtos, entre outros critérios (Lustosa; Mesquita; Quelhas & Oliveira; 2011).

Neste contexto a principal contribuição deste trabalho se dá na realização de um fluxo completo de pesquisa operacional voltado ao processo de impressão digital, proporcionando ao decisor um conjunto de soluções.

Na seção 2 deste trabalho revisamos alguns temas pertinentes à solução do problema proposto; na sessão 3 apresentamos a descrição mais ampla do problema; na sessão 4 é exposto o algoritmo de solução; na sessão 5 é descrita a metodologia aplicada; na sessão 6 são apresentados os resultados que, por sua vez, são analisados na sessão 7; por fim, algumas conclusões são apresentados na sessão 8.

2. DEFINIÇÃO E FORMULAÇÃO DO PROBLEMA

O problema considerado consiste em determinar o sequenciamento de n produtos (*jobs*) a serem impressos em m impressoras não necessariamente idênticas, de forma a minimizar o *makespan* e o custo relacionado ao processo de impressão.

É importante ressaltar que as máquinas que compõem o parque gráfico em questão são máquinas digitais, que possuem capacidade de impressão de todas as cores em uma única passagem da folha.

A minimização do *makespan* consiste na minimização do tempo total de conclusão de todas as tarefas, enquanto que o custo consiste na multiplicação do tempo gasto para se completar a tarefa pelo valor/hora atrelado à máquina que a executou. O valor/hora distinto é pré-estabelecido para cada máquina m , com base em seu custo. As tarefas podem ser executadas por qualquer uma das m máquinas disponíveis, porém o tempo de processamento de cada tarefa varia de máquina para máquina. Por fim, as tarefas não podem ser repartidas, ou seja, cada tarefa é executada inteiramente por uma única máquina.

No caso real considerado tem-se 4 máquinas que trabalham ininterruptamente por 12 horas /dia. A janela de tempo utilizada em um planejamento é de 5 dias (semana útil), totalizando 60 horas de trabalho. Os dois conjuntos de dados reais considerados contêm 80 tarefas cada.

Os parâmetros do problema são: conjunto de *Jobs* (N), conjunto de máquinas (M), custo/hora de cada máquina (c_m) e tempo de processamento do *job* i na máquina m (p_{im}). (Bretas; 2013).

As variáveis do problema consistem na data de início de cada *job* (t_i). A matriz binária α_{im} determina se a tarefa i é processada pela máquina m ($\alpha_{im}=1$) ou não ($\alpha_{im}=0$). A variável Z representa o *makespan* (tempo máximo para completar o último *job*). A constante G é um valor inteiro de valor muito elevado. Com base nesses parâmetros, foi considerado o seguinte o modelo de otimização multiobjetivo:

$$\min Z \quad (1)$$

$$\min f_2 = \left(\sum_{i \in N} \sum_{m \in M} (c_m * p_{im} * \alpha_{im}) \right) \quad (2)$$

Restrições

$$\sum_{m \in M} \alpha_{im} = 1 \quad \forall i \in N \quad (3)$$

$$Z - t_i + (1 - \alpha_{im}) * G \geq p_{im} \quad \forall i \in N, m \in M \quad (4)$$

$$\alpha_{im} \in \{0,1\} \quad \forall i \in N, m \in M \quad (5)$$

$$t_i \geq 0 \quad \forall i \in N \quad (6)$$

$$c_m \geq 0 \quad \forall m \in M \quad (7)$$

$$Z \geq 0 \quad (8)$$

A função objetivo (1) representa a minimização do *makespan*; a função objetivo (2) visa a minimização do custo; as restrições (3) garantem a unicidade na execução da tarefa por apenas uma máquina, (4) asseguram que o *makespan* é maior ou igual ao tempo de execução da última tarefa. As demais restrições demonstram o domínio das variáveis do modelo.

3. REVISÃO BIBLIOGRÁFICA

Problemas do tipo máquinas paralelas não relacionadas (MPNR) buscam a definição da sequência de tarefas mais indicada para a alocação em cada máquina com intuito de otimizar uma ou mais funções objetivo (López et al., 1995; Pfund et al., 2004; Pinedo, 2008; Lin et al., 2011).

Nos casos mono-objetivo, talvez a abordagem mais usual seja a que considera a minimização do tempo total de conclusão da última tarefa como função objetivo. Como exemplo, os trabalhos de Fanjul-Peyro e Ruiz (2010; 2011) propõem metaheurísticas de busca iterativa local e redução do tamanho do problema original para problemas com a característica de tarefas com tempo de processamento distribuídos uniformemente. As heurísticas apresentadas atingiram um desempenho interessante quando comparados a outras proposições, chegando a gerar resultados próximos às soluções ótimas. Lin et al. (2011) propuseram uma solução de dois estágios, baseada na combinação de relaxação de programação linear inteira com procedimentos de reatribuição e intercâmbio. O método apresentou excelentes resultados para problemas com poucas tarefas e uma aproximação satisfatória para conjuntos com muitas de tarefas.

De forma semelhante Gairing et al. (2005) propuseram uma abordagem combinatória de enumeração parcial inspirada em programação linear inteira combinando com *Branch-and-cut*. Grigoriev et al. (2005) propôs uma implementação em duas partes, sendo a primeira uma relaxação da programação linear inteira e a segunda uma heurística de arredondamento. Srivastava (1998) utilizou heurísticas baseadas em *Tabu Search* para encontrar o desvio percentual em relação a solução ótima.

Para o mesmo problema, Chang e Chen (2011) propõem um Algoritmo Genético utilizando intercâmbio adjacente e não adjacente como alternativa para aumentar a velocidade de convergência. Por outro lado, Ying et al. (2010) utilizaram um *Simulated Annealing* restrito, que elimina movimentações de busca ineficientes. Rabadi et al. (2006) realizou uma comparação entre metaheurística de busca prioritária aleatória (MHBPA) com heurística de particionamento (HP). Os trabalhos apresentados acima obtiveram as soluções ótimas em instâncias pequenas.

O estudo realizado por Arnaout et al. (2008; 2010) utiliza um algoritmo baseado em *Ant Colony Optimization* (ACO) contendo dois estágios. Quando comparado com MHBPA, HP e *tabu search*, o ACO proposto apresentou desempenho superior. Problemas MPNR com duas ou mais funções objetivo têm recebido forte atenção recentemente. A formulação mais usual considera a minimização do tempo total ponderado de conclusão (*makespan*) e a minimização dos tempos de atraso referentes às entregas programadas como critérios de otimização. Por se tratar de um problema NP-difícil, diversos heurísticas e metaheurísticas vêm sido propostas para gerar soluções satisfatórias em tempo aceitável (López et al., 1995; Pfund et al., 2004; Pinedo, 2008).

4. ALGORITMO DE SOLUÇÃO

Fig. 1 Etapas do algoritmo implementado

Neste estudo o indivíduo é representado por um conjunto de vetores, onde cada um desses vetores representa uma máquina e seus componentes representam a sequência das tarefas que são executadas por esta máquina. Desta forma a estrutura é escalável, permitindo a inserção de novas máquinas e tarefas.

O algoritmo desenvolvido para solucionar o problema em questão foi baseado no algoritmo NSGA II. Na sequência são detalhadas as etapas que foram customizadas a partir do fluxo padrão do algoritmo original (Bandyopadhyay & Bhattacharya, 2013):

4.1 Cruzamento

O cruzamento consiste em escolher dois indivíduos que venceram na etapa do torneio binário. Após definem-se dois pontos de corte para que seja efetuada a troca de bloco de tarefas entre os indivíduos. O próximo passo consiste em utilizar um método baseado no PMX (Goldberg; Ling, 1985) para realizar a adequação e reorganização das tarefas dentre os indivíduos selecionados.

4.2 Mutação

Foram utilizadas duas abordagens, detalhadas a seguir.

4.2.1 Mutação Slide

Neste operador de mutação, seleciona-se uma máquina (m) e na lista de tarefas desta máquina seleciona-se uma posição (i) nessa máquina. Por fim, seleciona-se uma tarefa (t), que será removida da sua posição atual e inserida na posição i .

Após operador, as tarefas contidas na máquina m a partir da posição i são deslocadas uma posição para frente. Por sua vez, as tarefas contidas após t em sua máquina de origem são deslocadas uma posição para trás.

Este operador é capaz de alterar tanto tarefas na mesma máquina quanto mover tarefas entre máquinas, alterando os tamanhos das permutações.

4.2.2 Mutação Troca

Neste operador, seleciona-se uma máquina ($m1$) uma tarefa dentro da máquina $m1$ ($t1$), uma máquina ($m2$) e uma tarefa

dentro da máquina $m2$ ($t2$). Após a definição destas quatro informações, é realizada a troca, com a tarefa $t1$ sendo inserida em $m2$, na posição da tarefa $t2$ e; $t2$ sendo inserida em $m1$, na posição da tarefa $t1$.

Este operador também permite alterar tarefas em uma mesma máquina e em máquinas distintas, mas os tamanhos das sequências nunca são alterados.

4.3 Busca Local (quando necessário)

Para este trabalho foi implementado um operador de busca local baseado nas mesmas funções de vizinhança empregadas na mutação. A seguir é detalhada esta busca.

Primeiramente seleciona-se as soluções onde será executada a busca local. São selecionados os indivíduos com maior distância de multidão, visando priorizar a diversidade das soluções escolhidas.

No caso da vizinhança de slide, são geradas duas listas aleatórias, sendo uma com a posição das tarefas e uma com a posição de encaixe das tarefas. Após a geração destas listas, insere-se a tarefa em primeiro lugar na lista de tarefas na primeira posição da lista de encaixes. O processo se repete (combinando tarefas e encaixes) até que a primeira melhoria (*first improvement*) seja identificada ou o número máximo de avaliações sem melhoria seja atingido.

No caso da vizinhança de troca, também são geradas duas listas, mas ambas contendo as tarefas do problema. As trocas entre duas tarefas são realizadas com base nestas listas, uma a uma, até que um dos critérios de parada seja atendido (os critérios são os mesmos da vizinhança de slide).

A busca local é executada sem que um determinado número de avaliações ocorre sem melhora no melhor indivíduo considerando apenas o objetivo *makespan*. Esse número é definido pelo usuário.

5. METODOLOGIA

Para a execução deste estudo foram utilizados dois conjuntos de dados reais, contendo o detalhamento dos trabalhos a serem executados em um período de uma semana. Ambos os conjuntos possuem 80 tarefas de impressão. Cada conjunto possui valor/hora de cada máquina e o tempo que cada máquina leva para executar cada tarefa.

Conforme mencionado anteriormente, foi utilizada uma implementação do algoritmo genético NSGAII para alocar as tarefas nas máquinas e identificar os melhores sequenciamentos. De posse dessas informações foram realizados alguns testes com as bases disponíveis para estimar os parâmetros do algoritmo proposto. Nesse sentido foram feitos experimentos utilizando as duas bases fornecidas, variando os parâmetros de taxa de mutação, probabilidade de cruzamento, tamanho da população e número de gerações. Os intervalos testados foram:

- probabilidade de cruzamento: de 80% a 100%;

- probabilidade de mutação: de 10% a 40%;
- tamanho da população: de 50 a 100 indivíduos;
- número de gerações: de 1000 a 30000.

Após analisar os resultados obtidos identificou-se como mais robusta a parametrização: probabilidade de cruzamento 90%, probabilidade de mutação 30%, tamanho da população 100 e número de gerações 20000. A estimação dos parâmetros ocorreu de forma empírica.

Após a calibração dos parâmetros do algoritmo, iniciou-se os experimentos com as duas bases disponibilizadas. Este experimento foi construído com um total de seis rodadas de execução para cada amostra de dados, cada rodada composta por 30 execuções, totalizando 180 execuções por conjunto de dados. Após o término das execuções para cada um dos conjuntos de dados, foram gerados como saída uma aproximação final da fronteira Pareto para cada conjunto.

6. RESULTADOS

Após as 180 execuções em cada uma das bases analisadas, obteve-se os resultados sumarizados na Tabela 1:

Tabela 1 - Resultados obtidos nas amostras

	Melhor Makespan	Makespan Médio	Pior Makespan	Melhor Custo	Custo Médio	Pior Custo
1	45	61.28	103	106200	110324.08	116200
2	47	61.87	91	111300	114781.2	119900

Nas figuras 2 e 3 são apresentadas as aproximações das fronteiras Pareto obtidas em cada um dos conjuntos avaliados. A fronteira final está destacada em vermelho.

Fig. 2 Fronteira Pareto do 1ª conjunto de dados

Fig. 3 Fronteira Pareto do 2ª conjunto de dados

Abaixo segue a representação de uma solução. Tal solução corresponde ao conjunto de dados 2, na imagem acima a solução é destacada por um retângulo na cor preta onde o *makespan* é 47 e o custo 119900. Abaixo são apresentados a sequência de tarefas a serem executadas em cada uma das máquinas, seus *makespan* e seus custos.

SEQUÊNCIA																							TOTAL
Máquina1	61	14	16	36	24	30	18	52	70	38	47	72	41	0	32	78	56	60	62	67	3	2	
Máquina2	34	44	76	6	20	11	10	33	12	7	5	73	77	53	74	50	63	35				1	
Máquina3	66	45	27	71	39	46	75	8	40	13	79	26	37	51	31	4	29	69	25	58	64	2	
Máquina4	19	59	57	1	28	68	22	17	49	42	23	9	54	15	2	55	21	48	65	43		20	

MAKESPAN													TOTAL
Máquina1	3	3	3	2	1	3	2	2	3	2	3	1	43
Máquina2	2	3	2	3	3	2	5	1	3	5	2	1	47
Máquina3	3	2	1	3	1	3	1	4	4	2	1	2	46
Máquina4	3	3	3	2	2	2	2	3	2	2	2	2	45

CUSTO																			TOTAL
1500	1500	1500	1000	500	1500	1000	1000	1500	1000	1500	500	1500	2000	500	1000	1000	500	1000	23500
1200	1800	1200	1800	1800	1200	3000	600	1800	3000	1200	600	1200	3000	1200	1200	1800	600		28200
2100	1400	700	2100	700	2100	700	2800	2800	1400	700	1400	1400	2100	2100	1400	700	1400	700	32200
2400	2400	2400	1600	1600	1600	1600	2400	1600	1600	1600	2400	1600	1600	1600	2400	1600	1600	800	36000
																			119900

Fig. 4 representação completa da solução, solução destacada na figura 3

Utilizou-se a métrica de hipervolume para avaliar a convergência das soluções. O hipervolume é uma métrica que avalia a distribuição do conjunto de soluções em relação ao espaço de objetivos. Em um problema com dois objetivos, ele é área coberta por um conjunto de pontos em relação a um ponto de referência (Zitzler; Thiele, 1998).

Como os extremos das funções não eram conhecidos, foram utilizados para o vetor de referência os piores valores de função objetivo dentre todas as aproximações obtidas para cada instância. Com base na Tabela 5 é possível notar que as diferenças entre as melhores e as piores execuções são muito pequenas nas duas instâncias, o que indica que o algoritmo está convergindo de forma consistente para a mesma região do espaço de objetivos.

Tabela 2 – Valores de hipervolume por amostra

Conjunto	Maior	Valor Médio	Menor
1	0,725132	0,718515	0,702154
2	0,735115	0,7315	0,727321

7. CONCLUSÃO

O algoritmo proposto foi capaz de realizar o planejamento da execução de forma consistente, com baixa variabilidade entre execuções. Os resultados obtidos indicam ser possível aumentar o número de jobs semanais com certa tranquilidade. Por exemplo, no conjunto 2 obteve-se um melhor *makespan* de 47, cerca de 78.33% do tempo total disponível. Comparando-se a solução sugerida com o planejamento efetivamente executado foi possível promover uma melhora no *makespan* de 81% (de 58 horas para 47 horas) e um pequeno aumento de 0,4% no custo (de 119900 para 120400).

Com as informações explicitadas acima, pode-se definir que o objetivo *makespan* é mais sensível que o objetivo custo, ou seja, as maiores variações entre soluções ocorrem no contexto do *makespan*.

AGRADECIMENTOS

Agradecemos ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Minas pelo suporte ao trabalho desenvolvido.

REFERÊNCIAS

- Arnaout, J. P.; Rabadi, G. & Musa, R. (2010) A two-stage ant colony optimization algorithm to minimize the makespan on unrelated parallel machines with sequencedependent setup times. *Journal of Intelligent Manufacturing* 21, 693-701
- Arnaout, J. P.; Musa, R. & Rabadi, G. (2008) Ant colony optimization algorithm to parallel machine scheduling problem with setups. 4th IEEE Conference on Automation Science and Engineering, 578-582
- Bandyopadhyay, S.; Bhattacharya, R. (2013) Solving multi-objective parallel machine scheduling problem by a modified NSGA-II, *Applied Mathematical Modelling*, vol. 37, no. 10-11, pp. 6718-6729
- Bretas, A. M. C. (2013) Modelagem do sequenciamento no processo de lingotamento contínuo por métodos exatos e heurísticos. Dissertação (Mestrado em Engenharia de Produção) apresentado à Universidade Federal de Minas Gerais, Minas Gerais
- Chang, P. C. & Chen, S. H. (2011) Integrating dominance properties with genetic algorithms for parallel machine scheduling problems with setup times. *Applied Soft Computing* 11, 1263-1274
- Fanjul-Peyro, L. & Ruiz, R. (2010) Iterated greedy local search methods for unrelated parallel machine scheduling. *European Journal of Operational Research* 207, 55-69

Gairing, M.; Monien, B. & Woclaw, A. (2005) A faster combinatorial approximation algorithm for scheduling unrelated parallel machines. *ICALP 2005, LNCS 3580*, 828-839

Grigoriev, A.; Sviridenko, M. & Uetz, M. (2005) Unrelated parallel machine scheduling with resource dependent processing times. *IPCO 2005, LNCS 3509*, 182-195

Liaw, C. F.; Lin, Y. K.; Cheng, C. Y. & Chen, M. (2003) Scheduling unrelated parallel machines to minimize total weighted tardiness. *Computers & Operations Research* 30, 1777-1789

López, O. C.; Barcia, R. M. & Eyada, O. (1995) Problema de programação da produção: um esquema de classificação. *Revista Produção* 5, 145-168

Lin, Y. K.; Pfund, M. E. & Fowler, J. W. (2011) Heuristics for minimizing regular performance measures in unrelated parallel machine scheduling problems. *Computers & Operations Research* 38, 901-916

Lustosa, L.; Mesquita, M. A.; Quelhas, O. & Oliveira, R. (2011) Planejamento e Controle da Produção (Pcp), Elsevier Brasil.

Mokotoff, E. (2001) Parallel machine scheduling problems: a survey. *Asia-Pacific Journal of Operational Research* 18, 193-242

Morton, T. E. & Pentico, D. W. (1993) Heuristic scheduling systems: with applications to production systems and project management. New York: John Wiley & Sons.

Pfund, M.; Fowler, J. W. & Gupta, J. N. D. (2004) A survey of algorithms for single and multi-objective unrelated parallel-machine deterministic scheduling problems. *Journal of the Chinese Institute of Industrial Engineers*, Vol. 21, No 3, 230-241

Pinedo, M. L. Scheduling: (2008) Theory, Algorithms, and Systems. New York: Springer.

Srivastava, B. (1998) An effective heuristic for minimising makespan on unrelated parallel machines. *Journal of the Operational Research Society* 49, 886-894

Ying, K. C.; Lee, Z. J. & Lin, S. W. (2010) Makespan minimization for scheduling unrelated parallel machines with setup times. *Journal of Intelligent Manufacturing*, Publicação online.

Zitzler, E.; Thiele, L. (1998) Multiobjective optimization using evolutionary algorithms - a comparative case study. In: EIBEN, A. et al. (Ed.). *Parallel Problem Solving from Nature - PPSN V*. Springer Berlin / Heidelberg, (Lecture Notes in Computer Science, v. 1498). p. 292-301. ISBN 978-3-540-65078-2.