

2019	Minimizing the makespan on two identical parallel machines with mold constraints	China
<b>Autores:</b> Tsuiping Chung, Jatinder N.D. Gupta, Haidan Zhao, Frank Werner		
<b>Problema:</b> Sequenciamento de tarefas em duas máquinas paralelas idênticas utilizando moldes para a fabricação de <i>chips</i> eletrônicos, onde cada molde possui somente uma unidade disponível e várias tarefas podem utilizar o mesmo molde em tempos diferenciados, de forma que o mesmo molde não pode ser utilizado em duas máquinas ao mesmo tempo e o <i>makespan</i> seja o menor possível;		
<b>Metodologia:</b> Foram utilizados 3 métodos heurísticos para a definição das soluções, sendo que todos são baseados na heurística conhecida como <i>Longest Processing Time</i> (LPT) definidos, no artigo, apenas como TLPT, DMLPT e CTD. Também foi desenvolvido um modelo matemático que representa o problema proposto e utilizado o CPLEX para definição do <i>lower bound</i> das instâncias, para que a eficiência dos métodos utilizados nos experimentos fosse calculada.		
<b>Instâncias e benchmark:</b> Foram geradas 580 instâncias de forma aleatória pelos próprios autores, sendo 260 de problemas pequenos $n \in \{6, 9, 12, 15\}$ e 320 de problemas maiores $n \in \{20, 50, 100, 150\}$ e número de moldes $q \in \{4, 6, 8, 10\}$ para os dois casos. Embora o objetivo seja a avaliação dos métodos heurísticos propostos, foram utilizadas as técnicas metaheurísticas <i>Discrete Particle Swarm Optimal</i> (DPSO) e <i>Variable Neighborhood Search</i> (VNS) para <i>benchmark</i> .		
<b>Experimentos e Conclusões:</b> Como o objetivo é a avaliação dos métodos construtivos propostos, o estudo realizou experimentos determinando o tempo limite de 10s para a execução dos métodos heurísticos propostos. Além disso, para instâncias maiores, foram utilizadas, também, as metaheurísticas DPSO e VNS, com soluções iniciais aleatórias e soluções calculadas pelos métodos TLPT e DMLPT, isso porquê, foram os métodos construtivos que apresentaram menor quantidade de resultados ótimos. O fato de que metaheurísticas costumeiramente apresentam melhores resultados com menores tempos de resposta ao serem utilizadas soluções iniciais construtivas, fizeram com que as soluções apresentadas pelos algoritmos TLPT e DMLPT também fossem utilizadas como soluções iniciais para comparação. A avaliação foi realizada calculando o <i>gap</i> dos resultados obtidos em relação ao <i>Lower Bound</i> do problema, de forma que o algoritmo definido como CTD apresentou os melhores resultados tanto para instâncias pequenas quanto para as grandes, sendo que nessas últimas, o algoritmo apresentou os menores <i>gaps</i> , além de apresentar tempos de resposta em média 5% menores do que os das metaheurísticas. Ainda foi demonstrado que o nível de melhora apresentado pelas metaheurísticas para os problemas com instâncias grandes, foram, em média, praticamente idênticas.		
<b>Observações:</b> *Bounds for multiprocessor scheduling with resource constraints - Garey and Graham - 1975. Primeiros a tratarem do PSTMP.		



UNIVERSIDADE FEDERAL  
DE OURO PRETO

Departamento de Engenharia de Produção  
Mestrado em Engenharia de Produção



2017	Scheduling identical parallel machines with tooling constraints	Brasil
<b>Autores:</b> Andreza Cristina Beezão, Jean-François Cordeau, Gilbert Laporte, Horacio Hideki Yanasse		
<b>Problema:</b> Estuda o sequenciamento de tarefas em $m$ máquinas idênticas paralelas com restrições na utilização de ferramentas ( <i>Identical Parallel Machine with Tooling Constraints</i> - IPMTC), onde cada máquina é apta a processar qualquer uma das tarefas e cada tarefa deverá ser processada apenas uma vez. Interrupções não são permitidas, não existe relação de precedência entre os trabalhos, para a execução das tarefas existe um grupo de ferramentas que deverão ser alocados em <i>slots</i> que possuem uma capacidade de armazenamento pré-determinada. O objetivo geral é a redução do <i>makespan</i> .		
<b>Metodologia:</b> Foi utilizada a combinação de uma heurística para soluções iniciais do Problema de Sequenciamento de Tarefas e Troca de Ferramentas em uma máquina simples como método de inicialização, em seguida foram utilizadas duas versões da metaheurística <i>Adaptive Large Search Neighborhood</i> (ALNS) para refinamento, uma com a solução inicial proposta (ALNS-B) e a outra com solução inicial aleatória (ALNS-R). Também foram propostos dois modelos matemáticos $M_1$ e $M_2$ .		
<b>Instâncias e benchmark:</b> Foram usadas 2880 instâncias divididas em 2 grupos definidos como IPMTC-I e IPMTC-II, essas instâncias foram geradas pelos próprios autores, obedecendo as regras de distribuição uniforme de dois artigos: "Um algoritmo enumerativo baseado em ordenamento parcial para resolução do problema de minimização de trocas de ferramentas" que deu origem às instâncias pequenas e " <i>Minimizing the number of tool switches on a flexible machine</i> " que gerou as instâncias maiores. Os resultados foram comparados com os do artigo " <i>Heuristic procedures for the parallel machine problem with tool switches</i> " que utilizaram os algoritmos CLIP1 e CLIP2.		
<b>Experimentos e Conclusões:</b> Todos os métodos foram executados nos diversos experimentos do artigo, utilizando como método de avaliação o cálculo dos <i>gaps</i> dos resultados obtidos e comparados aos obtidos na literatura, de forma que o algoritmo ALNS-R apresentou os melhores resultados em todas as instâncias testadas, pequenas ou grandes com média de 2% para o grupo IPMTC-I e até 1,2% para o grupo IPMTC-II.		



# UNIVERSIDADE FEDERAL DE OURO PRETO

Departamento de Engenharia de Produção  
Mestrado em Engenharia de Produção



2016	Parallel machine with tool loading	Turquia
<b>Autores:</b> Selin Özpeynirci, Burak Gökgür, Brahim Hnich		
<b>Problema:</b> Estuda o sequenciamento de tarefas em máquinas idênticas paralelas com restrições na utilização de ferramentas, de forma que três características são consideradas: execução da tarefa, carregamento das ferramentas e o próprio sequenciamento. São usadas máquinas paralelas não relacionadas e tarefas que demandam várias ferramentas, as quais deverão ser carregadas nas máquinas com antecedência, de forma que cada ferramenta usada em uma determinada tarefa, consome uma quantidade limitada de recursos. Não é permitido preempção e cada tarefa pode ser executada em qualquer máquina, porém, o tempo de execução das tarefas pode alterar de uma máquina para outra.		
<b>Metodologia:</b> Foram usados dois modelos matemáticos: Modelo de Programação Inteira Mista 1 e Modelo de Programação Inteira Mista 2, onde um se diferencia do outro pela quantidade de variáveis de decisão e restrições - menor quantidade no segundo modelo. A execução desses modelos no CPLEX se mostrou insuficiente para encontrar soluções ótimas para instâncias com número de tarefas maiores ou iguais a 15 durante o prazo de 1h, desta forma, foi desenvolvida uma heurística gulosa que produzisse uma solução inicial e a metaheurística <i>Tabu Search</i> foi utilizada para refinamento.		
<b>Instâncias e benchmark:</b> Foram geradas 120 instâncias pequenas, conjunto de ferramentas e ferramentas utilizadas nas tarefas. Foi realizada a comparação entre os resultados obtidos pelos modelos propostos (MIP1 e MIP2) e os resultados da busca tabu. Foi calculado também o valor dos <i>gaps</i> em relação às soluções ótimas calculadas pelo CPLEX, ou pelo <i>Upper Bound</i> quando o mesmo não foi capaz de calcular soluções ótimas.		
<b>Experimentos e Conclusões:</b> Os experimentos foram realizados com limite de tempo de 1h, de forma que algumas soluções ótimas foram obtidas, porém, o tempo se mostrou insuficiente para encontrar soluções ótimas em problema pouco maiores, evidenciando a necessidade do aumento do tempo disponibilizado ou a utilização de relaxações lagrangeanas, para se encontrar soluções ótimas para essas instâncias. Os modelos propostos não apresentaram resultados ótimos em todas instâncias testadas, de forma que para instâncias com 15 tarefas, não foi possível encontrar uma solução ótima sequer, mesmo utilizando a busca tabu. Desta forma, foi proposto a inclusão de uma restrição ao modelo MIP2 que representa a quebra de simetria das ferramentas a serem utilizadas nas tarefas, ou seja, se um tipo de ferramenta for utilizado por duas tarefas e essa ferramenta possuir 2 cópias, não faz diferença qual cópia será utilizada em cada tarefa. Assim, o novo MIP2 encontrou 10 soluções ótimas para todas as instâncias, inclusive para as instâncias com 15 tarefas foi possível localizar 2 soluções ótimas, de forma a apresentar uma performance superior às dos outros experimentos. Embora a busca tabu não tenha encontrado soluções ótimas para instâncias com número de tarefas iguais a 15, concluiu-se pela robustez do desempenho do algoritmo que o mesmo será capaz de produzir soluções viáveis em tempos hábeis, pois o tempo médio de cálculo das soluções foi bastante inferior.		



2015	Uniform parallel machine scheduling with resource consumption constraint	Taiwan
<b>Autores:</b> Wei-Chang Yeh, Mei-Chi Chuang, Wen-Chiung Lee		
<b>Problema:</b> Propoem três algoritmos para resolver problemas de sequenciamento em máquinas paralelas uniformes com o objetivo de reduzir o makespan com a restrição de que o consumo de recursos não pode exceder à quantidade máxima de cada tipo. O artigo não cita se os recursos são ou não são renováveis.		
<b>Metodologia:</b> São usados três algoritmos: Genetic Algorithm (GA), Swarm Particle Optimization Algorithm (SPO) e um Simplified Swarm Optimization Algorithm (SSO).		
<b>Instâncias e benchmark:</b> Foram realizados 3 níveis de testes:  <ol style="list-style-type: none"><li>1. No primeiro caso, foram criadas 100 instâncias com 3 máquinas (8 ou 10 tarefas) e 5 máquinas (10 ou 12 tarefas);</li><li>2. No segundo, os números de máquinas usadas foram 5 ou 10 e o número de tarefas variou de acordo com o número de máquinas - o número de instâncias não foi citado;</li><li>3. Finalmente, o número de máquinas permaneceu 5 ou 10, porém o número de tarefas variou com <math>n=50, 100, 200, 500</math> e <math>1000</math> - novamente o número de instâncias não foi citado.</li></ol>		
<b>Experimentos e Conclusões:</b> Os resultados foram tratados caso a caso em relação aos experimentos e comparados uns com os outros:  <ol style="list-style-type: none"><li>1. Os melhores resultados foram obtidos pelo algoritmo SSO e os piores resultados foram do GA;</li><li>2. Novamente, os melhores resultados foram obtidos pelo algoritmo SSO quando o número de máquinas era 5, porém, quando o número de máquinas aumentou para 10, o GA passou a apresentar os melhores resultados;</li><li>3. Novamente o GA apresentou os melhores resultados, entretanto, o SSO apresenta uma convergência mais rápida, embora para instâncias menores.</li></ol>		



UNIVERSIDADE FEDERAL  
DE OURO PRETO

Departamento de Engenharia de Produção  
Mestrado em Engenharia de Produção



2014	<b>Block Models for Scheduling Jobs on Two Parallel Machines with a Single Server</b>	Irã
<b>Autores:</b> Keramat Hasani, Svetlana A. Kravchenko and Frank Werner		
<b>Problema:</b> Propõe o estudo de sequenciamento de tarefas em duas máquinas paralelas e idênticas, cada tarefa deverá ser alocada em uma máquina e o tempo de preparação deverá ser levado em conta.		
<b>Metodologia:</b> Foram propostos dois modelos de programação linear inteira BM1 e BM2 baseados na ideia de decomposição de uma sequência em blocos, que consiste em ordenar todas as tarefas e caso o tempo de setup mais o de processamento de uma determinada tarefa seja maior do que a anterior ou posterior, a(s) menor(es) será(ão) agrupada(s) à tarefa de maior tempo de execução, definida como tarefa de primeiro nível, de forma que esse bloco formado será considerado como uma única tarefa. O modelo definido como BM2, possui uma restrição à mais, onde o número de tarefas de segundo nível não pode superar um valor pré-definido. Os modelos foram processados pelo CPLEX.		
<b>Instâncias e benchmark:</b> Foram geradas um total de 540 instâncias, utilizando o mesmo padrão de distribuição uniforme de dois outros artigos para <i>benchmark</i> , de forma que para instâncias pequenas ( $n \in \{8, 20, 50, 100\}$ ) já existiam na literatura e instâncias maiores ( $n \in \{200, 250\}$ ) foram geradas pelos autores.		
<b>Experimentos e Conclusões:</b> Para as instâncias com $n \in \{8, 20, 50\}$ foi determinado o tempo limite de execução em $(300/8)n$ utilizando somente o modelo BM1. Para as instâncias com $n \in \{100, 200, 250\}$ (que não haviam sido desenvolvidos na literatura) o tempo limite foi de 1h e o modelo utilizado foi o BM2. Os resultados do modelo BM1 apresentou média de <i>gap</i> de 1,01. Enquanto que para o modelo BM2, foram apresentados resultados com <i>gaps</i> máximos de 1,1 e em alguns casos o valor ótimo foi atingido.		

2014	<b>Simulated annealing and genetic algorithms for the two-machine scheduling problem with a single server</b>	Irã
<b>Autores:</b> Keramat Hasani, Svetlana A. Kravchenko and Frank Werner		
<b>Problema:</b> Estuda o problema de sequenciamento em duas máquinas paralelas com um único servidor, onde o mesmo é responsável por preparar cada máquina para receber uma determinada tarefa, com o objetivo de reduzir o <i>makespan</i> .		
<b>Metodologia:</b> Foram utilizados dois algoritmos: o <i>Genetic Algorithm</i> e <i>Simulated Annealing</i> . Não foi proposto nenhum modelo no artigo, entretanto foi utilizado um <i>lower bound</i> que foi calculado aritmeticamente pelos autores.		
<b>Instâncias e benchmark:</b> Foram criadas 560 instâncias pelos próprios autores, utilizando o mesmo padrão de distribuição de outro artigo - <i>Block Models for Scheduling Jobs on Two Parallel Machines with a Single Server</i> que utilizou duas formulações de programação linear inteira, que se baseiam em decompor uma dada sequência em blocos.		
<b>Experimentos e Conclusões:</b> No geral, os dois algoritmos apresentaram <i>gaps</i> próximos do <i>lower bound</i> . Para $n \geq 700$ os resultados obtidos pelo GA apresentaram uma quantidade maior de resultados que se aproximavam do <i>lower bound</i> com uma média de 2% de desvio, porém, no geral, o SA apresentou uma porcentagem de desvio em relação ao <i>Lower Bound</i> de 0,5%. De qualquer forma, ambos os algoritmos apresentaram resultados melhores do que quaisquer outros apresentados pela literatura sobre o tipo de problema estudado.		



UNIVERSIDADE FEDERAL  
DE OURO PRETO

Departamento de Engenharia de Produção  
Mestrado em Engenharia de Produção



2015	Minimizing the makespan for the two-machine scheduling problem with a single server: Two algorithms for very large instances	Irã
<b>Autores:</b> Keramat Hasani, Svetlana A. Kravchenko and Frank Werner		
<b>Problema:</b> Estuda o problema de sequenciamento em duas máquinas paralelas com um único servidor, onde o mesmo é responsável por preparar cada máquina para receber uma determinada tarefa, com o objetivo de reduzir o <i>makespan</i> .		
<b>Metodologia:</b> Foram utilizados dois algoritmos para calcular as sequências e os seus respectivos makespans. Como existem duas formas de tratar o problema, onde o <i>Min-Idle</i> tenta reduzir o tempo entre o final de uma tarefa e o início de outra e o <i>Min-Loadgap</i> tenta reduzir o tempo gasto entre o início de uma tarefa e o início de outra tarefa. Novamente, neste estudo, foi determinado um <i>lower bound</i> que já havia sido utilizado no artigo que lhe deu origem, no qual foram utilizados os algoritmos GA e SA.		
<b>Instâncias e benchmark:</b> Foram utilizadas as 560 instâncias geradas pelos próprios autores em outro artigo - <i>Simulated Annealing and Genetic Algorithms for the Two-Machine Scheduling Problem with a Single Server</i> , que foi utilizado para realizar o <i>benchmark</i> , além de mais 700 instâncias com $n \in \{1500, 2000, 3000, 4000, 5000, 6000, 7000, 8000, 9000, 10000\}$ geradas para o artigo atual. Os algoritmos SA e GA utilizados no artigo anterior, também foram aplicados às novas instâncias.		
<b>Experimentos e Conclusões:</b> Os métodos propostos apresentaram resultados piores para número de tarefas $n \leq 400$ em relação ao GA e SA, utilizados no artigo anterior, embora o tempo médio de cálculo das soluções tenha sido superior aos algoritmos propostos anteriormente, enquanto que para $n \geq 500$ os resultados obtidos pelos métodos <i>Min-Idle</i> e <i>Min-Loadgap</i> apresentaram resultados com <i>gaps</i> mais próximos do <i>lower bound</i> - em média, 14,3% alcançaram o resultado ótimo, além de se mostrar bastante rápido, pois no pior caso, com $n = 10000$ tem média de 371,3 segundos para alcançar o ótimo. Para as instâncias consideradas verdadeiramente grandes $n \in \{1500, 2000, 3000, 4000, 5000, 6000, 7000, 8000, 9000, 10000\}$ foi possível alcançar o <i>lower bound</i> em até 31,1%, na média.		





UNIVERSIDADE FEDERAL  
DE OURO PRETO

Departamento de Engenharia de Produção  
Mestrado em Engenharia de Produção



2013	<b>Parallel machine scheduling with additional resources: Notation, classification, models and solution methods</b>	Turquia
------	---	---------

**Autores:** Emrah B. Edis, Ceyda Oguz, Irem Ozkarahan

**Problema:** Estuda o problema de sequenciamento em máquinas paralelas com recursos adicionais, onde a execução das tarefas depende do consumo de recursos finitos. Este artigo foi selecionado por conter citações sobre o problema tratado como tema de dissertação. (Pág. 5/453, 12-13/460-461)

**Metodologia:** O artigo trata da revisão bibliográfica do tema, apresentando discussões sobre os pontos fortes e fracos da literatura.

**Instâncias e benchmark:** Por se tratar de uma revisão literária, instâncias não são discutidas e o benchmark não realiza comparações entre resultados e sim entre os modelos matemáticos utilizados.

**Experimentos e Conclusões:** O estudo apresentou textos sobre o tema e as ligações entre si, apresentou também os pontos fortes da literatura e finalmente, alguns modelos de programação inteira para o tema.

2011	<b>Parallel machine scheduling problem to minimize the makespan with resource dependent processing times</b>	China
------	--	-------

**Autores:** Kai Li, Ye Shi, Shan-lin Yang, Ba-yi Cheng

**Problema:** Estuda o problema de sequenciamento de tarefas em máquinas paralelas que possuem o tempo de processamento dependente do recurso utilizado no processo, no qual a função de consumo de recurso é considerada com uma função linear decrescente e o objetivo é reduzir o *makespan*.

**Metodologia:** Para o cálculo das soluções foi utilizado o algoritmo *Simulated Annealing*, para se determinar as soluções iniciais utilizadas para execução do SA, foi utilizado o método *Modified Longest Processing Time* (MLPT). Para efeito de avaliação dos resultados foram calculados os *gaps* em relação ao *lower bound*, que foi calculado aritmeticamente.

**Instâncias e benchmark:** Foram realizados testes em 10000 instâncias geradas randomicamente pelos próprios autores, utilizando como parâmetro de avaliação a porcentagem de desvio dos resultados em relação ao *lower bound*.

**Experimentos e Conclusões:** Foram realizadas comparações entre os resultado da heurística MLPT para solução inicial e o do SA em relação aos *gaps*. Os experimentos avaliaram que o algoritmo SA é bastante preciso, pois a média dos *gaps* em todas as soluções foi de apenas 0,000201%, além do fato de o algoritmo ser capaz de resolver problemas de até 1000 tarefas em 0,875s.



UNIVERSIDADE FEDERAL  
DE OURO PRETO

Departamento de Engenharia de Produção  
Mestrado em Engenharia de Produção



2010	Heuristic procedures for the parallel machine problem with tool switches	USA
<b>Autores:</b> Y. Fathi e K. W. Barnette		
<b>Problema:</b> Estuda o problema de sequenciamento de tarefas, que consistem em processar um conjunto de peças com um dado tempo de processamento e requisitos de ferramentas em $m$ máquinas paralelas e idênticas, objetivando reduzir o <i>makespan</i> .		
<b>Metodologia:</b> Foram utilizados 3 procedimentos para solucionar o problema proposto: 1 - Baseado em uma busca local de primeira melhora, sendo que para este procedimento foram propostos dois métodos de solução definidos como <i>Composed Local Improvement Procedure</i> ( $CLIP_1$ e $CLIP_2$ ); 2 - Utilizado o método construtivo conhecido como <i>Longest Processing Time</i> (LPT), e o método definido como <i>Random List Processing Procedure</i> (RLPP) que se baseia no próprio LPT; 3 - O terceiro procedimento se baseia em um método guloso utilizado no Problema do Caxeiro Viajante que se assemelha ao Problema de Sequenciamento de Tarefas. Definido como <i>Constructive Heuristic</i> (CSTR), sem uma definição clara sobre a abreviação utilizada.		
<b>Instâncias e benchmark:</b> As 38 instâncias foram geradas pelos autores e separadas em 3 conjuntos de dados, definidos como: Instâncias Estruturadas, Instâncias Crescentes e Instâncias Randômicas. Dois artigos foram utilizados para definição dos parâmetros de geração das instâncias: <i>Minimizing the number of tool switches on a flexible machine</i> e <i>The tool switching problem</i> .		
<b>Experimentos e Conclusões:</b> Para cada instância testada, o procedimento de busca local de melhora foi rodada com 12 diferentes soluções iniciais e os melhores resultados obtidos foram reportados, sendo que dessas, 10 foram geradas aleatoriamente, 1 utilizando o LPT e mais 1 utilizando a heurística gulosa de vários reinícios para o problema de uma única máquina.		

2010	A pairwise interchange algorithm for parallel machine scheduling	França
<b>Autores:</b> S. M. T. Fatemi Ghomi e F. Jolai Ghazvini		
<b>Problema:</b> Estuda o sequenciamento de $n$ tarefas em $m$ máquinas paralelas e idênticas sem preempção com chegadas simultâneas, objetivando reduzir o <i>makespan</i> .		
<b>Metodologia:</b> É proposto um algoritmo de intercâmbio em pares (PIA - <i>Pairwise Interchange Algorithm</i> ) que consiste em definir uma solução inicial e, caso a mesma não seja igual ao <i>lower bound</i> , realiza trocas entre as duas máquinas, as que tiverem maior e menor <i>makespans</i> , observando a necessidade de haver pelo menos uma tarefa na máquina com maior <i>makespan</i> que tenha maior tempo de execução do que pelo menos uma das tarefas na máquina com menor <i>makespan</i> , caso contrário, uma nova solução inicial será tratada. Foi definido um <i>lower bound</i> aritmeticamente e os métodos <i>Longest Processing Time</i> (LPT) e Multifit também foram utilizados.		
<b>Instâncias e benchmark:</b> Foram geradas 80 instâncias pelos próprios autores e a comparação de resultados foi realizada entre os algoritmos LPT e Multifit. Cada uma das instâncias foram testadas com 4 tipos de soluções iniciais diferentes, onde um deles é o próprio LPT, o <i>Shortest Processing Time</i> (SPT), e duas combinações entre os dois, o SPT-LPT e o LPT-SPT.		
<b>Experimentos e Conclusões:</b> O método proposto atingiu o <i>lower bound</i> em maior número de instâncias do que os algoritmos avaliados, de forma que, no geral, com as heurísticas construtivas propostas como soluções iniciais, o <i>lower bound</i> foi atingido em 72 das 80 instâncias e 8 apresentaram <i>gaps</i> menores do que 1.		





UNIVERSIDADE FEDERAL  
DE OURO PRETO

Departamento de Engenharia de Produção  
Mestrado em Engenharia de Produção



2010	A LISTFIT heuristic for minimizing makespan on identical parallel machines	USA
<b>Autores:</b> J. N. D. Gupta e J. Ruiz-Torres		
<b>Problema:</b> Estuda o sequenciamento de tarefas em máquinas paralelas e idênticas.		
<b>Metodologia:</b> Foi proposto um novo algoritmo denominado por LISTFIT que é uma combinação de 2 outros algoritmos conhecidos: embalagem de lixo ( <i>bin-packing</i> ) e lista de sequenciamento ( <i>list scheduling</i> ). Basicamente, o algoritmo utiliza o <i>bin-packing</i> , dividindo as tarefas em dois grupos e cria iterativamente listas que são atribuídas à máquina menos carregada. As tarefas de cada uma das listas é alterada a cada iteração, assim como os métodos de ordenação das mesmas, ora pelo método <i>Shortest Processing Time</i> (SPT), ora pelo <i>Longest Processing Time</i> (LPT).		
<b>Instâncias e benchmark:</b> As instâncias foram geradas pelos autores e divididas em 4 grupos: E1, E2, E3 e E4. Foram passadas informações somente sobre os grupos E2 (2000 instâncias) e E3 (4800 instâncias). O objetivo do estudo é apresentar o desempenho do algoritmo LISTFIT proposto, desta forma os experimentos foram rodados também com os algoritmos LPT, MULTIFIT e COMBINE para <i>benchmark</i> .		
<b>Experimentos e Conclusões:</b> O algoritmo proposto alcançou maior quantidade de soluções ótimas do que qualquer uma das outras heurísticas propostas em todas as instâncias testadas. O principal objetivo do artigo é apresentar os níveis de melhoras que o LISTFIT impõe sobre o COMBINE: para as instâncias E1 não houve melhora; para E2 a média do percentual de melhora variou entre 3 e 4% para algumas instâncias; para E3 a média foi de 1,6%; e para E4 a média foi de aproximadamente 6,4%.		

2009	Parallel-machine scheduling problems with sequence-dependent setup times using an ACO, SA and VNS hybrid algorithm	Irã
<b>Autores:</b> J. Behnamian, M. Zandieh e S. M. T. Fatemi Ghomi		
<b>Problema:</b> Estuda o problema de sequenciamento de tarefas em máquinas paralelas e idênticas, onde o tempo de preparação das máquinas para a execução de uma tarefa é levado em consideração.		
<b>Metodologia:</b> O estudo propõe um método híbrido, definido como HMH, de três outros algoritmos conhecidos: <i>Ant Colony Optimization</i> (ACO), <i>Variable Neighborhood Search</i> (VNS) e <i>Simulated Annealing</i> (SA), que basicamente funciona do seguinte modo: A pesquisa começa com alguma solução inicial baseada no ACO e movimentos são executados iterativamente entre as soluções vizinhas. Na iteração, uma solução aleatória é selecionada da vizinhança e será aceita por uma probabilidade definida. No entanto, se a solução não for aceita, a estrutura da vizinhança será alterada da mesma maneira que na técnica VNS. Foi utilizado o seguinte critério de parada, para todas as instâncias: $(m^2 \times (n + 1)/2)$ .		
<b>Instâncias e benchmark:</b> Foram geradas 252 instâncias pelos próprios autores divididas em 6 grupos: 2 pequenos, 2 médios e 2 grandes. Os resultados do método foram comparados com a colônia de formigas utilizando busca local variável (ACO-VNS), com o recozimento simulado também utilizando utilizando busca local variável (SA-VNS) e à busca local variável (VNS).		
<b>Experimentos e Conclusões:</b> Os resultados apresentados pelos experimentos demonstraram que o HMH obteve <i>gaps</i> menores na maioria das instâncias, sendo que no grupo de instâncias definidas como de tamanho médio, os resultados apresentados foram os melhores em todos os problemas. A média geral dos <i>gaps</i> foi de 1,969%.		

2008	An improved delayed-start LPT algorithm for a partition problem on two identical parallel machines	USA
------	--	-----

**Autores:** Christos Koulamas E George J. Kyparisis

**Problema:** Estuda o problema de sequenciamento de tarefas em duas máquinas paralelas e idênticas.

**Metodologia:** O artigo propõe um método baseado no *Longest Processing Times* (LPT), definido como *off-line delayed-start* LPT e denominado por Algoritmo A, que consiste em: se  $n \leq 4$ , usa a regra LPT normalmente; se  $n \geq 5$ , seleciona as 5 primeiras tarefas com maior tempo de execução e as divide entre as duas máquinas e o restante das tarefas, caso haja, será sequenciado de acordo com a regra LPT normalmente.

**Instâncias e benchmark:** O artigo não trata de experimentos computacionais. Realiza cálculos aritméticos do funcionamento do método proposto, desta forma, não existem instâncias e nem foi realizado *benchmark*.

**Experimentos e Conclusões:** Os estudos demonstraram que o método proposto pode ser mais preciso para grupos de tarefas maiores demandando um certo esforço computacional, porém, se a quantidade de tarefas subir em exagero, o esforço pode se tornar proibitivo.

2008	A modified LPT algorithm for the two uniform parallel machine makespan minimization problem	USA
------	---	-----

**Autores:** Christos Koulamas E George J. Kyparisis

**Problema:** Estuda o problema de sequenciamento de tarefas em duas máquinas paralelas e idênticas.

**Metodologia:** O artigo propõe um método baseado no *Longest Processing Times* (LPT), definido como *modified* LPT e denominado por Algoritmo MLPT, que consiste em: se  $n \leq 2$ , usa a regra LPT normalmente; se  $n \geq 3$ , seleciona as 3 primeiras tarefas com maior tempo de execução e as divide entre as duas máquinas e o restante das tarefas, caso haja, será sequenciado de acordo com a regra LPT normalmente.

**Instâncias e benchmark:** O artigo não trata de experimentos computacionais. Realiza cálculos aritméticos do funcionamento do método proposto, desta forma, não existem instâncias e nem foi realizado *benchmark*.

**Experimentos e Conclusões:** Os estudos demonstraram que o método proposto pode ser mais preciso para grupos de tarefas maiores demandando um certo esforço computacional, porém, se a quantidade de tarefas subir em exagero, o esforço pode se tornar proibitivo.



UNIVERSIDADE FEDERAL  
DE OURO PRETO

Departamento de Engenharia de Produção  
Mestrado em Engenharia de Produção



2008	A Heuristic Algorithm for the Scheduling Problem of Parallel Machines with Mold Constraints	Taiwan
------	---	--------

**Autores:** Tzung-Pei Hong, Pei-Chen Sun e Shin-Dai Li

**Problema:** Estuda o problema de sequenciamento de tarefas em máquinas paralelas, onde o tempo de preparo entre duas tarefas deverá ser levado em consideração, caso o molde utilizado pelas duas tarefas sejam diferentes. São definidas  $n$  tarefas, com  $m$  máquinas e  $l$  moldes, onde a quantidade de cada tipo de molde pode ser diferente. As datas de entrega das tarefas também deverão ser levadas em consideração.

**Metodologia:** É proposto um modelo matemático que representa o problema descrito e um algoritmo que consiste basicamente em dividir as tarefas por grupos que utilizam o mesmo molde e sequenciá-los de acordo com a regra da data de vencimento mais cedo *Earliest Due Date* (EDD) e distribuí-los nas máquinas disponíveis.

**Instâncias e benchmark:** Foram usados dados reais de uma empresa de produção de tubos de aço, que utiliza 10 máquinas paralelas não relacionadas. O período estudado apresentava, no dia inicial do rastreo, 332 tarefas não realizadas e 95 foram tomadas com exemplo. Até o final do período de realização dos cálculos, novas tarefas foram incluídas chegando ao número de 578 tarefas não executadas. As definições de qual tarefa a ser executada era realizada pela empresa de forma manual.

**Experimentos e Conclusões:** Não foram apresentados os resultados nem pelo método proposto e nem a programação manual realizada pela empresa, porém, nas discussões de resultados foi afirmado que o método trouxe grandes melhoras no sequenciamento apresentado em relação àquele tratado pela empresa.

20xx	Nome	??????
------	------	--------

**Autores:**

**Problema:**

**Metodologia:**

**Instâncias e benchmark:**

**Experimentos e Conclusões:**