LI Simpósio Brasileiro de Pesquisa Operacional
Limeira–SP, 02 a 06 de setembro de 2019

ESTUDO DE MODELOS MATEMÁTICOS DE UM PROBLEMA DE SEQUENCIAMENTO DE TAREFAS COM FOCO NA CONSTRUÇÃO DE HEURÍSTICAS MATEMÁTICAS

Natália Antunes
Programa de Pós-Graduação em Engenharia Elétrica - Universidade Federal de Minas Gerais
Av. Antônio Carlos 6627, 31270-901, Belo Horizonte, MG, Brasil
natalia-antunes@ufmg.br

Eduardo Gontijo Carrano
Programa de Pós-Graduação em Engenharia Elétrica - Universidade Federal de Minas Gerais
Av. Antônio Carlos 6627, 31270-901, Belo Horizonte, MG, Brasil
egcarrano@ufmg.br

RESUMO

Este documento apresenta um estudo de modelos matemáticos para o problema de sequenciamento de tarefas em uma única máquina com data de entrega comum, onde o objetivo é minimizar penalidades decorrentes de atrasos e adiantamentos nas entregas das tarefas. O problema é NP-difícil, tendo sido tratado por diferentes heurísticas e metaheurísticas ao longo do tempo. Com a finalidade de desenvolver um método que combine metaheurísticas e algoritmos exatos, as chamadas heurísticas matemáticas, são validados modelos matemáticos para o problema de sequenciamento de tarefas. A validação do modelo é feita utilizando instâncias *benchmark* disponíveis na literatura.

PALAVRAS CHAVE. Sequenciamento, Data de entrega comum, Programação inteira mista.

PM- Programação Matemática

ABSTRACT

This document presents a study of mathematical models for the common due date single machine scheduling problem. The problem is NP-hard, which justifies the recent proposals of heuristics and metaheuristics for solving it. In order to build a method that combines metaheuristics and exact algorithms, the so-called matheuristics, mathematical models for the problem were validated. The evaluation of the models is performed using benchmark instances available in literature.

KEYWORDS. Scheduling. Common due date. Mixed integer programming.

MP - Mathematical Programming

https://proceedings.science/p/106936?lang=pt-br

1. Introdução

Produção *Just-in-time* enfatiza que adiantamentos e atrasos devem ser evitados. Adiantamentos podem resultar em custos com seguro, deterioração e armazenamento. Por sua vez, os atrasos podem causar insatisfação do cliente, penalidades contratuais, perdas de vendas e redução de reputação [Liao e Cheng, 2007]. Abordagens envolvendo atrasos e adiantamentos em problemas de sequenciamento de produção têm recebido bastante atenção nos últimos anos e diversos algoritmos foram implementados para resolução do problema *single machine common due-date scheduling* (SMCDD). Devido à sua natureza NP-difícil, heurísticas e metaheurísticas são as técnicas mais recorrentes, pois apresentam soluções com boa qualidade em tempo computacional razoável.

As heurísticas matemáticas, que são técnicas que combinam metaheurísticas e métodos de programação matemática, têm ganhado espaço principalmente devido à crescente disponibilidade de softwares de otimização, bibliotecas e frameworks para programação matemática. Aplicações de heurísticas matemáticas em problemas de *scheduling* são encontradas em [Ozkan et al., 2015], [Lin e Ying, 2016] e [Toffolo et al., 2016].

De acordo com [Talbi, 2016], a integração entre os métodos exatos e não exatos pode ocorrer em diferentes níveis, sendo a integração maior ou menor dependendo da configuração utilizada. As principais configurações utilizadas na integração dos dois métodos são:

- O algoritmo exato ou a metaheurística resolvendo um problema de natureza diferente do problema de otimização considerado. Um exemplo dessa configuração seria a utilização da metaheurística para geração dos limites superiores do algoritmo exato.
- A metaheurística incorporando um algoritmo exato ao longo da busca pela solução, ou vice versa.
- As duas famlias de algoritmos sendo usadas de forma independente, em uma etapa de pré-processamento ou pós-processamento. Nesta configuração, informações são trocadas entre as duas famílias de algoritmos de forma sequencial.
- As duas famílias de algoritmos resolvendo o problema de forma independente e trocando informações ao longo de processo de busca pela solução. Nesta configuração, informações são trocadas entre as duas famílias de algoritmos de forma paralela.

A metodologia abordada consiste em desenvolver uma heurística matemática que incorpore algoritmos exatos como parte da busca em uma metaheurística. A integração entre os dois métodos será realizada acrescentando vizinhanças exatas ao longo do processo de busca pela solução da metaheurística. Para que esta abordagem seja eficiente, o algoritmo exato deve ter um tempo de execução baixo.

O tempo computacional necessário para solucionar o problema SMCDD através de métodos exatos cresce exponencialmente à medida que o número de tarefas aumenta. Entretanto, existem estratégias para reduzir o tempo computacional do algoritmo exato quando o número de tarefas é grande. Nestas estratégias, utiliza-se a solução incumbente da metaheurística para reduzir o número de variáveis que serão otimizadas. Isto é feito fixando parte das variáveis da solução incumbente no modelo matemático e realizando a otimização considerando somente as variáveis livres.

As estratégias de fixação são classificadas em dois tipos: fixação forte e fixação fraca. Nas estratégias com fixação forte, um conjunto de variáveis da solução incumbente são fixadas no modelo matemático e o outro conjunto é otimizado. Nas estratégias de fixação fraca, uma restrição indica o máximo de variáveis da solução incumbente que podem mudar de valor por iteração, mas

não indica qual o conjunto de variáveis que manterão seu valor. Isso proporciona ao algoritmo uma flexibilidade para procurar as melhores variáveis para serem alteradas [Camargo et al., 2014].

Neste trabalho serão apresentados e validados modelos matemáticos para o problema SMCDD, estes modelos serão utilizados para construção de heurísticas matemáticas. Modelos matemáticos com características distintas são interessantes para elaboração de vizinhanças com algoritmos exatos, pois apresentam variáveis de decisão e tempos de execução distintos.

O artigo está organizado em 5 seções: nesta seção foi feita uma introdução sobre a ideia principal do artigo, destacando como será feita, futuramente, a integração entre o método exato e a metaheurística. Na segunda seção é feita a descrição do problema SMCDD e os detalhes sobre o problema são apresentados. Na terceira seção são apresentados os modelos matemáticos. Na quarta seção é feita a validação dos modelos matemáticos. Por fim, na quinta seção são apresentadas conclusões a respeito do trabalho.

2. Descrição do problema

O problema *single machine common due-date scheduling* tem a finalidade de determinar o sequenciamento de um conjunto de tarefas que apresentam uma data de entrega comum e devem ser processadas por uma única máquina. O objetivo é minimizar a soma das penalidades decorrentes de atrasos ou adiantamentos na entrega das tarefas. A data de entrega em comum, d, influencia na complexidade do problema e pode ser uma variável de decisão (cujo valor deve ser determinado na otimização) ou pode ser conhecida a priori. Neste trabalho, d é calculada como em [Biskup e Feldmann, 2001], onde foram propostas as instâncias *benchmark* consideradas para este problema. O valor de d na referência é um parâmetro de entrada e é restritiva, como segue:

$$d = round(h \sum_{i=1}^{N} p_i), \tag{1}$$

onde h é um escalar entre 0 e 1, p_i é o tempo de processamento da tarefa i e N é a quantidade de tarefas.

Neste problema, um conjunto de N tarefas devem ser processadas por uma única máquina a partir de um tempo inicial zero. Todas as tarefas estão disponíveis no tempo zero, são independentes e executadas em uma única operação da máquina. Além disso, não existe tempo ocioso entre as tarefas e não é permitido preempção (após iniciar o processamento da tarefa, não é permitido interrompê-lo). Sendo assim, somente uma tarefa pode ser concluída na data de entrega comum, sendo as outras tarefas finalizadas antes ou depois dela. Não é possível determinar previamente se uma tarefa será entregue antes ou depois da data de entrega comum, então calcula-se o adiantamento e o atraso para todas as tarefas. Se o tempo de conclusão da tarefa i, C_i, é menor ou igual a d, a tarefa está adiantada e o adiantamento é dado por $E_i = max\{d - C_i, 0\}$. Se o tempo de conclusão da tarefa i for maior que d, a tarefa está atrasada e o atraso é dado por $T_i = max\{C_i - d, 0\}$.

Além do tempo de processamento, p_i, as tarefas apresentam penalidades por adiantamento e atraso, α_i e β_i, respectivamente. Essas penalidades são definidas por unidade de tempo e também podem influenciar na complexidade do problema. Um estudo da influência de α_i e β_i no problema SMCDD pode ser encontrado em [Biskup e Feldmann, 2001].

O objetivo do problema é minimizar simultaneamente as penalidades por atraso e adiantamento, sendo a função objetivo dada por:

$$f(S) = \sum_{i=1}^{N} \alpha_i E_i + \sum_{i=1}^{N} \beta_i T_i \tag{2}$$

onde S representa um cronograma viável das tarefas. O cronograma S contém todas as informações necessárias para a produção das tarefas: a sequência π de produção, o tempo de início da primeira tarefa e a data de entrega em comum.

A sequência ótima apresenta algumas propriedades que podem auxiliar na resolução e formulação do problema. As propriedades da sequência ótima quando d é restritiva são:

- Propriedade 1: na sequência ótima não existe tempo ocioso entre o processamento de duas tarefas consecutivas.
- Propriedade 2: na sequência ótima, as tarefas são ordenadas em forma de V, onde as tarefas que estão adiantadas são sequenciadas em ordem decrescente de p_i/α_i e as tarefas que estão atrasadas são sequenciadas em ordem crescente de p_i/β_i.
- Propriedade 3: existe uma sequência ótima em que o processamento da primeira tarefa inicia no instante de tempo zero ou uma tarefa termina o seu processamento exatamente na data de entrega comum.

As demonstrações destas propriedades podem ser vistas em [Cheng e Kahlbacher, 1991], [Baker e Scudder, 1990] e [Hoogeveen e Van de Velde, 1991].

3. Modelagem matemática

Nesta seção serão descritos modelos matemáticos para o problema SMCDD. Estes modelos apresentam abordagens diferentes do problema, com partes das variáveis de decisão distintas. Isso pode ser interessante para a elaboração de vizinhaças exatas para as heurísticas matemáticas. As diferentes características das variáveis de decisão oferecem uma flexibilidade na fixação das variáveis no modelo matemático, permitindo que vizinhanças distintas sejam implementadas.

Além disso, diferentes modelos matemático podem apresentar tempos de execução distintos para uma mesma vizinhança. Isso ocorre devido às características do modelo. Sendo assim, é interessante que as vizinhanças sejam implementadas com os diferentes modelos, de forma que o modelo escolhido apresente o melhor desempenho em determinada vizinhança.

Conforme visto na seção de descrição do problema, algumas informações sobre as tarefas são disponibilizadas a priori, sendo os parâmetros de entrada do problema:

α_i	Penalidade por adiantamento da tarefa i
β_i	Penalidade por atraso da tarefa i
p_i	Tempo de processamento da tarefa i
d	Data de entrega em comum (*due-date*)

As características de cada um dos modelos serão apresentadas na sequência.

3.1. MIP 1

O modelo matemático de programação inteira mista (MIP, do inglês *mixed integer programming*) a seguir é apresentado em [Biskup e Feldmann, 2001] e, nesta formulação, o tempo de início de cada uma das tarefas é determinado com o auxílio de uma variável binária, x_{ij}, que assume valor 1 quando a tarefa i é processada antes da tarefa j (não necessariamente imediatamente) e igual a 0 caso contrário. Este modelo será citado ao longo do artigo como MIP 1. As variáveis de decisão do modelo são:

T_i	Atraso da tarefa i
E_i	Adiantamento da tarefa i
s_i	Instante de início da tarefa i
x_{ij}	Variável binária que indica a precedência ou não entre duas tarefas

As seguintes restrições garantem que o atraso e o adiantamento sejam determinados de forma condizente com o esperado no problema:

$$T_i - E_i = s_i + p_i - d \quad \forall i \in N \tag{3}$$

$$s_i + p_i \leq s_j + M(1 - x_{ij}) \quad i = 1, .., N-1; j = i+1, ..., N \tag{4}$$

$$s_j + p_j \leq s_i + M x_{ij} \quad i = 1, ..., N-1; j = i+1, ..., N \tag{5}$$

$$E_i, T_i, s_i \geq 0 \quad \forall i \in N \tag{6}$$

$$x_{ij} \in \{0, 1\} \quad i = 1, ..., N-1; j = i+1, ..., N \tag{7}$$

A restrição 3 determina o atraso e o adiantamento de cada uma das N tarefas. As restrições 4 e 5 determinam o tempo de início de cada uma das tarefas, estas restrições são disjuntivas e garantem que ou a tarefa i será processada antes da tarefa j ou a tarefa j será processada antes da tarefa i. Por fim, as restrições 6 e 7 são restrições de não negatividade e integralidade. Nesta formulação, o valor de M deve ser maior que a soma dos tempos de processamento de todas as tarefas.

3.2. MIP 2

O modelo MIP a seguir é baseado no modelo matemático posicional apresentado em [Keha et al., 2009] para diferentes problemas de agendamento de tarefas em uma única máquina. Algumas alterações foram feitas no modelo original para adequação ao objetivo do problema SMCDD. Nesta formulação, o tempo de conclusão de cada uma das tarefas é determinado com o auxílio de uma variável de decisão binária, x_{ij}, que equivale a 1 se a tarefa i está atribuída à posição j da sequência ou 0 caso contrário. Este modelo será citado ao longo do artigo como MIP 2. As variáveis de decisão do modelo são:

T_i	Atraso da tarefa i
E_i	Adiantamento da tarefa i
C_i	Tempo de conclusão da tarefa i
γ_j	Tempo de conclusão da tarefa na posição j
x_{ij}	Variável binária que determina a posição que a tarefa i está alocada

As seguintes restrições garantem que o atraso e o adiantamento sejam determinados de forma condizente com o esperado no problema:

$$T_i \geq C_i - d \quad \forall i \in N \tag{8}$$

$$E_i \geq d - C_i \quad \forall i \in N \tag{9}$$

$$\sum_{j \in N} x_{ij} = 1 \quad \forall i \in N \tag{10}$$

$$\sum_{i \in N} x_{ij} = 1 \quad \forall j \in N \tag{11}$$

$$\gamma_1 \geq \sum_{i \in N} p_i x_{ij} \tag{12}$$

$$\gamma_j \geq \gamma_{j-1} + \sum_{i \in N} p_i x_{ij} \quad j = 2, .., N \tag{13}$$

$$C_i \geq \gamma_j - M(1 - x_{ij}) \quad \forall i, j \in N \tag{14}$$

$$\sum_{j \in N} \gamma_j = \sum_{i \in N} C_i \tag{15}$$

$$E_i, T_i, C_i, \gamma_j \geq 0 \quad \forall i, j \in N \tag{16}$$

$$x_{ij} \in \{0, 1\} \quad \forall i, j \in N \tag{17}$$

As restrições 8 e 9 determinam o atraso e o adiantamento, respectivamente, de cada uma das N tarefas. As restrições 10 e 11 garantem que as tarefas sejam alocadas em apenas uma posição e que cada posição receba à atribuição de somente uma tarefa. As restrições 12 e 13 determinam o tempo de conclusão da tarefa na posição j. As restrições 14 e 15 determinam os tempos de conclusão de cada uma das tarefas para definição dos valores do atraso e do adiantamento. Por fim, as restrições 16 e 17 são restrições de não negatividade e integralidade. Nesta formulação, o valor de M também deve ser maior que a soma dos tempos de processamento de todas as tarefas.

4. Validação dos modelos matemáticos

A validação dos modelos matemáticos foi realizada utilizando as instâncias *benchmark* do problema SMCDD desenvolvidas por [Biskup e Feldmann, 2001]. O conjunto de instâncias apresenta quantidades de tarefas distintas, que variam em 10, 20, 50, 100, 200, 500 e 1000 tarefas. Cada conjunto contém 10 problemas e os valores de função objetivo são analisados para 4 valores de data de entrega comum.

Os modelos foram implementados utilizando o software Matlab 2016a e o software de otimização Gurobi 8.1.0. Os testes foram feitos em um PC Intel Core i5, com 4GB de memória RAM, em um ambiente Windows 10. O desempenho dos modelos matemáticos foi analisado em todas as instâncias de 10 e 20 tarefas para os diferentes valores de data de entrega em comum (utilizando os valores de h = 0,2, 0,4, 0,6 e 0,8 em $d = round(h \sum_{i=1}^{N} p_i)$).

As vizinhanças exatas da heurística matemática devem apresentar um tempo de execução baixo, pois serão introduzidas no processo de busca pela solução da metaheurística. Por esta razão, os testes serão realizados utilizando apenas os dois conjuntos com as menores quantidades de tarefas. À medida que o número de tarefas aumenta, o tempo de execução do algoritmo exato cresce exponencialmente e se torna impeditivo para a construção de vizinhanças exatas para a heurística matemática. Em instâncias cujo número de tarefas é elevado, serão utilizadas as estratégias de fixação de variáveis.

As instâncias com 10 tarefas convergiram em poucos segundos e, com a finalidade de determinar o tempo médio de cada problema, foram realizadas 30 execuções do algoritmo para cada um dos problemas. O tempo médio que cada um dos modelos demora para realizar a otimização pode ser visto na Tabela 1. É possível observar que os tempos médios para se obter a solução ótima são relativamente baixos para as instâncias com 10 tarefas. Para cada uma das instâncias, foi destacado o modelo que apresenta o menor tempo médio. Analisando os tempos médios dos dois modelos, é possível observar que:

- O MIP 1 apresentou tempos médios melhores em 19 instâncias, sendo que em 13 instâncias o problema foi solucionado para um valor de data de entrega comum com h=0,6 e h=0,8.
- O MIP 2 apresentou tempos médios melhores em 21 instâncias, sendo que em 14 instâncias o problema foi solucionado para um valor de data de entrega comum com h=0,2 e h=0,4.
- A instância influencia no desempenho do algoritmo, um exemplo é a instância sch10k6 que com o MIP 1 obteve tempos médios elevados, enquanto que com o MIP 2 obteve tempos médios relativamente pequenos.
- O maior valor de tempo médio obtido pelo MIP 1 foi 25,51s.
- O maior valor de tempo médio obtido pelo MIP 2 foi 13,11s.

As instâncias com 20 tarefas não convergem rapidamente e, por esta razão, o software de otimização foi limitado em 5 minutos. Em todas as instâncias, a otimização terminou quando o tempo limite foi atingido. Os valores de função objetivo obtidos pelos dois modelos podem ser observados na Tabela 2. Analisando os valores de função objetivo dos dois modelos, é possível observar que:

- O MIP 1 apresentou valores de função objetivo melhores em 15 instâncias, sendo que em 9 instâncias o problema foi resolvido para um valor de data de entrega comum com h=0,6 e h=0,8.
- O MIP 2 apresentou valores de função objetivo melhores em 25 instâncias, sendo que em 14 instâncias o problema foi resolvido para um valor de data de entrega comum com h=0,2 e h=0,4.

Além do valor de função objetivo, o valor do GAP (diferença entre a melhor solução encontrada até o momento e o melhor limitante indentificado) é importante quando se limita o tempo no teste. Logo, por meio deste parâmetro é possível saber se a otimização está próxima do fim. Os valores dos GAPs dos testes das instâncias com 20 tarefas podem ser observados na Tabela 3. Analisando os valores dos GAPs obtidos pelos dois modelos, é possível observar que:

- De uma forma geral, os GAPs são elevados. Isto indica que a otimização estava longe de terminar nos dois modelos.
- Os valores dos GAPs obtidos pelo MIP 1 foram melhores em 9 instâncias.
- Os valores dos GAPs obtidos pelo MIP 2 foram melhores em 31 instâncias.

	h=0,2		h=0,4		h=0,6		h=0,8	
Instâncias	MIP 1	MIP 2	MIP 1	MIP 2	MIP 1	MIP 2	MIP 1	MIP 2
sch10k1	3,50s	**2,85s**	2,71s	**1,54s**	**1,99s**	2,32s	**1,63s**	2,47s
sch10k2	1,08s	**0,84s**	**0,68s**	1,72s	**0,63s**	2,29s	**0,70s**	1,84s
sch10k3	4,78s	**1,58s**	3,23s	**1,66s**	2,53s	**1,85s**	2,99s	**1,80s**
sch10k4	8,66s	**2,81s**	**5,09s**	13,11s	**1,83s**	6,59s	**1,94s**	10,44s
sch10k5	**4,36s**	5,58s	2,32s	**2,21s**	**0,81s**	4,84s	**1,02s**	5,97s
sch10k6	11,21s	**2,73s**	25,51s	**2,06s**	6,23s	**2,23s**	5,24s	**2,12s**
sch10k7	**2,49s**	2,81s	3,13s	**2,51s**	**2,05s**	2,50s	**1,42s**	3,04s
sch10k8	**3,68s**	5,26s	2,69s	**2,60s**	1,89s	**1,40s**	**1,00s**	2,64s
sch10k9	2,96s	**1,95s**	2,27s	**1,24s**	1,34s	**1,30s**	1,25s	**1,20s**
sch10k10	**2,75s**	2,89s	3,24s	**2,33s**	**1,44s**	1,96s	**1,17s**	2,30s

Tabela 1: Tempo médio (em segundos) dos modelos MIP 1 e MIP 2 para as instâncias com 10 tarefas

	h=0,2		h=0,4		h=0,6		h=0,8	
Instâncias	MIP 1	MIP 2	MIP 1	MIP 2	MIP 1	MIP 2	MIP 1	MIP 2
sch20k1	**4545**	4558	**3070**	3186	**3005**	3081	3117	**2998**
sch20k2	8781	**8524**	5229	**4973**	3319	**3241**	3037	**2981**
sch20k3	6257	**6197**	3868	**3838**	3655	**3604**	3613	**3608**
sch20k4	9356	**9347**	**5183**	5297	3334	**3330**	**3051**	3052
sch20k5	4269	**4219**	**2522**	2589	**2234**	2266	**2188**	2201
sch20k6	6610	**6597**	3588	**3555**	3093	**3045**	3060	**3022**
sch20k7	10581	**10468**	6416	**6208**	**4143**	4204	3895	**3892**
sch20k8	**3968**	4096	2362	**2107**	1757	**1698**	**1638**	1656
sch20k9	3478	**3413**	**2080**	2081	**1973**	2017	**1984**	1996
sch20k10	5188	**5023**	3097	**2949**	2117	**2110**	**1995**	2002

Tabela 2: Valores de função objetivo obtidos pelos modelos MIP 1 e MIP 2 para as instâncias com 20 tarefas limitadas em 5 minutos no software Gurobi

	h=0,2		h=0,4		h=0,6		h=0,8	
Instâncias	MIP 1	MIP 2	MIP 1	MIP 2	MIP 1	MIP 2	MIP 1	MIP 2
sch20k1	76,65%	**54,40%**	**59,08%**	76,72%	**56,82%**	80,24%	**56,57%**	73,20%
sch20k2	89,84%	**46,22%**	82,93%	**60,70%**	72,78%	**50,90%**	69,99%	**45,26%**
sch20k3	82,60%	**63,93%**	73,07%	**72,30%**	71,87%	**68,20%**	**69,51%**	69,98%
sch20k4	88,80%	**66,25%**	80,84%	**64,80%**	70,84%	**53,48%**	66,27%	**56,90%**
sch20k5	75,65%	**52,50%**	64,86%	**54,92%**	**57,70%**	68,65%	**55,38%**	67,84%
sch20k6	83,92%	**74,19%**	**71,58%**	71,99%	63,72%	**59,01%**	**68,87%**	70,12%
sch20k7	88,23%	**68,24%**	83,20%	**73,85%**	72,14%	**59,33%**	71,21%	**55,04%**
sch20k8	82,10%	**55,05%**	73,09%	**69,21%**	65,25%	**59,56%**	62,58%	**56,76%**
sch20k9	80,33%	**43,24%**	68,61%	**58,57%**	**67,67%**	74,95%	68,19%	**68,08%**
sch20k10	81,06%	**53,15%**	74,98%	**65,85%**	61,43%	**52,65%**	57,92%	**55,75%**

Tabela 3: Valores dos GAPs obtidos pelos modelos MIP 1 e MIP 2 para as instâncias com 20 tarefas limitadas em 5 minutos no software Gurobi

5. Conclusões

Este trabalho teve como objetivo estudar modelos matemáticos do problema *single machine common due-date scheduling* para construção de heurísticas matemáticas. Dois modelos com características distintas foram estudados e seu desempenho foi analisado em instâncias *benchmark* disponíveis na literatura.

O MIP 2 apresentou um desempenho melhor que o MIP 1 na maioria dos testes realizados. Entretanto, para valores de data de entrega comum com h = 0,6 e h = 0,8, os resultados obtidos pelo MIP 1 são relevantes. Sendo assim, ambos modelos são interessantes para o desenvolvimento das heurísticas matemáticas.

Os modelos matemáticos serão empregados em vizinhanças exatas acrescentadas na busca da metaheurística. Tendo em vista que estas vizinhanças devem apresentar um baixo tempo computacional, o desempenho do algoritmo exato foi testado nos dois conjunto de instâncias com o menor número de tarefas. Observou-se que o tempo de execução do algoritmo exato nas instâncias com 10 tarefas é relativamente baixo. Sendo assim, os tempos obtidos nestas instâncias viabilizam a implementação de vizinhanças exatas na metaheurística.

Os testes com as instâncias de 20 tarefas mostram que à medida que o número de tarefas aumenta, a complexidade do problema também aumenta. A otimização nestas instâncias terminou quando o tempo limite de 5 minutos foi atingido e com valores de GAP elevados. Isto mostra que para a construção de vizinhanças exatas com um tempo computacional baixo, é necessário que o algoritmo exato realize a otimização com poucas tarefas. Problemas com 20 tarefas apresentariam um tempo computacional impeditivo.

Os próximos passos para composição de heurísticas matemáticas são o desenvolvimento de vizinhanças exatas com os modelos estudados e a integração com uma metaheurística.

Agradecimentos: Os autores agradecem aos seguintes órgãos de fomento e instituições pelo suporte financeiro destinado ao desenvolvimento deste trabalho: CNPq, FAPEMIG e PPGEE UFMG. Além destes, o presente trabalho foi realizado com o apoio financeiro da CAPES - Brasil.

Referências

Baker, K. R. e Scudder, G. D. (1990). Sequencing with earliness and tardiness penalties: A review. *Operations Research*, 38:22â36.

Biskup, D. e Feldmann, M. (2001). Benchmarks for scheduling on a single machine against restrictive and unrestrictive common due dates. *Computers and Operations Research*, 28:787–801.

Camargo, V., Toledo, F. M., e Almada-Lobo, B. (2014). Hops â hamming-oriented partition search for production planning in the spinning industry. *European Journal of Operational Research*, 234:266–277.

Cheng, T. C. E. e Kahlbacher, H. G. (1991). A proof for the longestâjobâfirst policy in oneâmachine scheduling. *Naval Research Logistics*, 38:715–720.

Hoogeveen, J. A. e Van de Velde, S. L. (1991). Scheduling around a small common due date. *Journal of Operational Research*, 55:237â242.

Keha, A. B., Khowala, K., e F., J. W. (2009). Mixed integer programming formulations for single machine scheduling problems. *Comput. Ind. Eng.*, 56:357–367.

Liao, C. e Cheng, C. C. (2007). A variable neighborhood search for minimizing single machine weighted earliness and tardiness with common due date. *Comput. Ind. Eng.*, 52:404–413.

Lin, S. e Ying, K. (2016). Optimization of makespan for no-wait flowshop scheduling problems using efficient matheuristics. *Omega*, 64:115–125.

Ozkan, O., Ermis, M., e Bekmezci, I. (2015). A hybrid matheuristic approach for designing reliable wireless multimedia sensor networks. In *Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation*, p. 875–882, New York, NY, USA. ACM.

Talbi, E. (2016). Combining metaheuristics with mathematical programming, constraint programming and machine learning. *Annals of Operations Research*, 240:171–215.

Toffolo, T. A., Santos, H. G., Carvalho, M. A., e Soares, J. A. (2016). An integer programming approach to the multimode resource-constrained multiproject scheduling problem. *J. of Scheduling*, 19:295–307.

https://proceedings.science/p/106936?lang=pt-br