UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA DE PRODUÇÃO
CURSO DE GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

VITOR LUIS NESELLO

FORMULAÇÕES MATEMÁTICAS PARA O PROBLEMA DE SEQUENCIAMENTO DE TAREFAS COM MANUTENÇÕES PERIÓDICAS E TEMPOS DE SETUP

JOÃO PESSOA
2017

VITOR LUIS NESELLO

FORMULAÇÕES MATEMÁTICAS PARA O PROBLEMA DE SEQUENCIAMENTO DE TAREFAS COM MANUTENÇÕES PERIÓDICAS E TEMPOS DE SETUP

Orientador: Prof. Dr. Anand Subramanian

JOÃO PESSOA
2017

VITOR LUIS NESELLO

FORMULAÇÕES MATEMÁTICAS PARA O PROBLEMA DE SEQUENCIAMENTO DE TAREFAS COM MANUTENÇÕES PERIÓDICAS E TEMPOS DE *SETUP*

Monografia apresentada e aprovada, em 28 de julho de 2017, como requisito parcial à obtenção do grau de Engenheiro de Produção da Universidade Federal da Paraíba, pela comissão formada pelos seguintes membros.

BANCA EXAMINADORA

Prof. D.Sc. Anand Subramanian — Orientador
Departamento de Sistemas de Computação — UFPB

Prof. D.Sc. Hugo Harry F. R. Kramer — Examinador
Departamento de Engenharia de Produção — UFPB

Prof. D.Sc. Lucidio dos Anjos F. Cabral — Examinador
Departamento de Computação Científica — UFPB

Agradecimentos

Agradeço ao meu professor, orientador, mentor e amigo Anand Subramanian pelas incontáveis lições técnicas e pessoais.

Agradeço a todos que trabalharam comigo no querido LabMeQA e prontamente me ajudaram quando senti necessidade.

Agradeço aos meus pais pelo incentivo, sabedoria e porto seguro.

Agradeço aos meus companheiros de turma pela verdadeira amizade construída durante esses bons anos de graduação.

Resumo

O problema de sequenciamento em uma máquina estudado neste trabalho tem como objetivo ordenar tarefas em apenas uma máquina com períodos de indisponibilidade fixos, levando em consideração tempos de *setup* dependentes da sequência. O objetivo é minimizar o *makespan*. Neste trabalho é proposto um algoritmo exato que resolve, iterativamente, uma de três formulação matemáticas de arcos indexados no tempo apresentadas. Experimentos computacionais extensivos são conduzidos em 420 instâncias da literatura de até 50 tarefas, e em 360 instâncias, envolvendo até 125 tarefas, propostas neste trabalho. Os resultados são comparados com aqueles obtidos pela melhor formulação matemática disponível na literatura. Pela primeira vez, todas as instâncias do conjunto existente foram resolvidas na otimalidade.

Palavras-chave: Manutenções Periódicas, Formulação de Arcos Indexados no Tempo. *Makespan*. Algoritmo Exato.

Abstract

The single-machine scheduling problem with periodic maintenances and sequence-dependent setup times aims at scheduling jobs on a single machine in which periodic maintenances and setups are required. The objective is the minimization of the makespan. We propose an exact algorithm based on the iterative solution of three alternative arc-time-indexed models. Extensive computational experiments are carried out on 420 benchmark instances with up to 50 jobs, and on 360 newly proposed instances involving up to 125 jobs. We compare the results found by all formulations with those obtained by the best available mathematical formulation. All instances from the existing dataset are solved to optimality for the first time.

Keywords: Periodic Maintenances. Arc-Time-Indexed Formulation. Makespan. Exact algorithm.

Sumário

Glossário

1MPS	:	*One machine, periodic maintenances and setup times*
B&B	:	*Branch and bound*
MILP	:	Programação linear inteira mista
PO	:	Pesquisa operacional
AVRP	:	*Asymmetrical Vehicle Routing Problem*
GRASP	:	*Greedy randomized adaptative search procedure*
CF	:	Formulação compacta
VRPTW	:	*Vehicle routing problem with time windows*
ATF	:	*Arc-time indexed formulation*
ATF-M	:	*Arc-time indexed formulation with maintenance index*
ATF-P	:	*Arc-time indexed formulation with parallel paths*

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

1.1 Definição do tema

Os problemas de sequenciamento (do inglês *scheduling*) consistem em determinar a ordem em que um conjunto de tarefas será processado em um conjunto de máquinas (ou processadores), tendo como objetivo a otimização de algum critério específico. Em geral, um problema de sequenciamento específico pode considerar diversos atributos, tais atributos definem as características fundamentais do problema, como o critério a ser otimizado (função objetivo) e as restrições que serão levadas em conta. Alguns exemplos de função objetivo são: minimização do *makespan*, que é o tempo de término da última tarefa a ser processada; minimização do atraso total, que pode ser ponderado e é considerado em situações em que cada tarefa possui uma data de entrega; minimização do maior atraso; entre outras. Outras características comumente encontradas são: preempção, que é a possibilidade de interromper o processamento de uma tarefa para que essa seja finalizada posteriormente; tempos de *setup*, que é o tempo de reconfiguração necessário antes e/ou depois do processamento de uma tarefa, podendo depender da sequência de tarefas; número de processadores; relação de precedência entre processadores; relação de precedência entre tarefas, entre outros Pinedo (2016).

O problema de sequenciamento em uma máquina com manutenções periódicas e tempos e *setup* (do inglês *one machine, periodic maintenances and setup times problem* - 1MPS), problema abordado neste trabalho, consiste em sequenciar um conjunto J de tarefas em apenas uma máquina, levando em consideração que é necessário realizar atividades de *setup* e manutenções preventivas periódicas. As manutenções ocorrem a cada P unidades de tempo até que todas as tarefas $j \in J$ tenham sido processadas. O tempo de processamento de uma manutenção é exatamente p_0 unidades de tempo (o índice 0 será

usado para indexar as atividades de manutenção). Uma vez que uma tarefa ou manutenção $i \in J_+ = J \cup \{0\}$ tenha sido finalizada, a máquina deve ser reconfigurada para que a próxima tarefa ou manutenção $j \in J_+$ seja processada. Cada atividade de *setup* requer $s_{ij} \geq 0$ unidades de tempo para ser completada. Durante as paradas para manutenção e atividades de *setup*, as atividades de produção são interrompidas e considera-se que preempção não é permitido. Cada tarefa $i \in J$ possui um tempo de processamento $p_i > 0$. A função objetivo minimiza o *makespan*, ou seja, o instante de tempo C_{max} em que a máquina finaliza o processamento da última tarefa. Segundo a notação de três campos de Graham et al. (1979), o 1MPS pode ser identificado como $1|p_m, s_{ij}|C_{max}$. A Figura 1.1 representa uma solução para uma instância do 1MPS com 5 tarefas. Os blocos com o número 0 indicam uma manutenção, os blocos hachurados indicam atividades de *setup* e espaços em branco indicam tempos ociosos, que podem ocorrer devido à periodicidade das atividades de manutenção e ao fato de que preempção não é permitido.

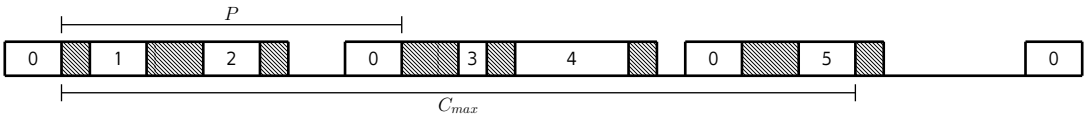


Figura 1.1: Exemplo da solução de uma instância do 1MPS com 5 tarefas.

Geralmente, problemas dessa natureza são resolvidos por algoritmos heurísticos (SUBRAMANIAN; FARIAS, 2017); ou métodos exatos, como algoritmos *Branch and Bound* (B&B), e modelos de programação linear inteira (MILPs). Nesse trabalho, são desenvolvidos MILPs baseados em variáveis de arcos indexados no tempo (PESSOA et al., 2010), tais modelos são implementados e comparados com os melhores modelos disponíveis na literatura do 1MPS.

1.2 Justificativa

Na literatura da Pesquisa Operacional (PO), os problemas de sequenciamento são considerados clássicos e estão entre os mais estudados. Isso se deve à grande relevância e variedade de suas aplicações práticas, bem como sua importância teórica no que tange o desenvolvimento de métodos exatos e heurísticos que podem ser aplicados à outros problemas da PO.

A importância prática dos problemas de sequenciamento está relacionada com a necessidade de empresas possuírem alta eficiência produtiva. O meio para alcançar tal eficiência é a eliminação de desperdícios, que podem ser minimizados ao adotar um sequenciamento

de boa qualidade. Os desperdícios decorrentes de uma programação da produção mau feita se traduzem em prazos não atendidos, grande volume produtos em processo, perda de matéria-prima, gastos excessivos com mão-de-obra, dentre outros.

Em algumas situações a adoção de uma estratégia de manutenção pode ser necessária. Nesses casos, o sequenciamento é afetado de maneira direta, pois os períodos de indisponibilidade dos processadores devem ser levados em consideração. As políticas de manutenção, que antes eram vistas como centros de custos, passaram a serem vistas como oportunidades de geração de lucro e vantagem competitiva. Como evidenciam Alsyouf (2007) e Swanson (2001), existem ganhos significativos ao aplicar uma estratégia de manutenção proativa. Alsyouf (2007) demonstra que o ganho financeiro ao evitar paradas não planejadas pode ser próximo da casa de um milhão de Dólares ao ano, em certas situações.

Muitos dos problemas de sequenciamento encontrados na prática fazem parte da classe $\mathcal{NP}$-Difícil. Um problema dito $\mathcal{NP}$-Difícil não é resolvido em tempo polinomial por nenhum algoritmo de que se tenha conhecimento. O número de soluções possíveis para tais problemas aumenta exponencialmente com a dimensão do problema, ou seja, no caso de sequenciamento de tarefas, conforme o número de tarefas a serem sequenciadas aumenta, o número de soluções aumenta em uma proporção muito maior. Portanto, problemas de sequenciamento de tarefas são, em geral, problemas de alta complexidade computacional. Para serem resolvidos em um horizonte de tempo aceitável requerem algoritmos sofisticados que fazem uso da teoria da PO.

Vale mencionar também a relevância do 1MPS não apenas em contextos de manufatura, mas também no setor de serviço. Por exemplo, ao assumir que as atividades de manutenção modelam o tempo em que funcionários de uma dada empresa de serviços estão ociosos (por exemplo, das 17:00 às 9:00 do dia seguinte), o 1MPS pode quantificar o tempo necessário para determinado funcionário finalizar um conjunto de tarefas, levando em conta o tempo de preparação necessário entre duas tarefas consecutivas.

Neste trabalho são utilizados métodos exatos para resolver o 1MPS. O trabalho com os melhores resultados exatos até então é o de Pacheco, Ángel-Bello e Álvarez (2013), que implementa e testa um modelo matemático baseado em variáveis de arcos, bem como uma meta-heurística baseada em *tabu search*. A decisão de utilizar modelos com variáveis de arcos indexados no tempo se baseia em dois argumentos. O primeiro é o fato de que tais modelos são conhecidos por possuir uma boa relaxação linear (PESSOA et al., 2010). O segundo é o fato de que modelos com variáveis em arcos (sem o índice do tempo) já

foram explorados em trabalhos anteriores envolvendo o 1MPS (PACHECO; ÁNGEL-BELLO; ÁLVAREZ, 2013).

1.3 Objetivos

1.3.1 Objetivo geral

Propor novas formulações matemáticas com variáveis de arcos indexados no tempo para o 1MPS.

1.3.2 Objetivos específicos

- Estudar e implementar a melhor formulação matemática conhecida para o 1MPS
- Desenvolver e implementar três formulações matemáticas baseadas em arcos indexados no tempo para o 1MPS
- Propor novas instâncias para o problema
- Comparar os resultados experimentais obtidos pelas formulações implementadas

1.4 Estrutura do trabalho

O restante deste trabalho é organizado como se segue. O Capítulo 2 trata de uma extensa revisão bibliográfica, apresentando trabalhos que envolvem sequenciamento de tarefas, com ênfase naqueles que consideram manutenções periódicas. No Capítulo 3 é apresentado o melhor MILP presente na literatura de conhecimento do autor. O Capítulo 4 descreve as formulações matemáticas propostas, bem como o algoritmo iterativo usado em conjunto com estas. O Capítulo 5 compara a performance dos três modelos de arcos indexados no tempo integrados (e, quando possível, não) ao algoritmo iterativo e apresenta resultados computacionais extensivos, ressaltando a formulação mais adequada para resolver o 1MPS. Além disso, os resultados obtidos são comparados com aqueles alcançados por Pacheco, Ángel-Bello e Álvarez (2013). As conclusões do trabalho se seguem no Capítulo 6.

Capítulo 2

Trabalhos relacionados

Neste capítulo, algumas contribuições relevantes relacionadas ao 1MPS são compiladas e brevemente descritas. Observa-se que existe um grande número de variantes do 1MPS, pois, ao considerar atividades de manutenção, abre-se um grande leque de possibilidades no que diz respeito às políticas de manutenção adotadas e como essas são modeladas. Por exemplo, Sbihi e Varnier (2008) consideram não só a situação em que P é fixo, mas também aquela em que P é determinado pelo tempo de produção contínuo da máquina, sendo esse fixo.

Em uma situação produtiva, os processadores podem estar indisponíveis por diversas razões, sejam algumas delas: ocorrência de falhas aleatórias que impossibilitem a operação; atividades de manutenção, podendo essas serem periódicas ou não; períodos de descanso de funcionários; fatores naturais imprevisíveis. Levando esses como possíveis cenários, classifica-se os trabalhos que envolvem períodos de indisponibilidade de processadores em dois grandes grupos (MA; CHU; ZUO, 2010). O primeiro considera que a indisponibilidade é uma das variáveis do problema, o que ocorre, por exemplo, em Xu, Yin e Li (2009). O segundo cenário trata a indisponibilidade como um processo determinístico pré-definido (objeto de estudo do presente trabalho); ou um processo estocástico, onde a indisponibilidade da máquina está relacionada a fatores aleatórios, como, por exemplo, a ocorrência de falhas que impossibilitem a operação (AHMADI et al., 2016; PEI et al., 2016). Ma, Chu e Zuo (2010) citam e descrevem trabalhos envolvendo períodos de indisponibilidade determinísticos.

Restringindo os estudos apenas ao caso em que os períodos de indisponibilidade são pré-definidos de maneira determinística, a principal diferença entre os trabalhos observados diz respeito ao número de períodos de indisponibilidade considerados. O caso mais simples é quando apenas uma atividade de manutenção é considerada. Nesse caso (con-

siderado por Lee e Lin (2001), Chen (2006b), Benmansour et al. (2014), Luo, Cheng e Ji (2015) e Yin et al. (2016)), o objetivo é sequenciar as atividades levando em conta apenas um período de indisponibilidade.

Considerando apenas trabalhos com múltiplos períodos de indisponibilidade, que é o caso do 1MPS, a variabilidade se encontra nas características tradicionais de problemas de sequenciamento (função objetivo, *setup*, preempção, etc). Liao e Chen (2003) propõem um método heurístico e um algoritmo B&B para a minimização do atraso máximo. Chen (2006) e Chen (2007) propõem métodos heurísticos e algoritmos B&B para a minimização do tempo de fluxo total (soma dos tempos de término de cada tarefa), porém o segundo considera preempção. Chen (2009), Lee e Kim (2012) e Liu et al. (2016) minimizam o número de tarefas atrasadas. Os dois primeiros propõem um algoritmo heurístico e o segundo, além disso, também desenvolve um MILP. Chen (2009) e Liu et al. (2016) propõem algoritmos B&B, com o último sendo o mais eficiente. Em relação à minimização de *makespan*, a maioria dos trabalhos não consideram tempos de *setup*, como observado na Tabela 2.1. Quando não se considera tempos de *setup*, os problemas podem ser modelados como um dos problemas de otimização combinatória mais estudados (*bin packing*) e os métodos de solução envolvem algoritmos B&B, bem como algoritmos aproximativos. Alguns trabalhos consideram máquinas paralelas, como Li et al. (2017) e Xu e Yang (2013). Esses propõem um algoritmo B&B e aqueles propõem um MILP e um algoritmo aproximativo.

Quando é considerado o sequenciamento de tarefas em apenas um processador sujeito a períodos de indisponibilidades fixos periódicos, tempos de *setup* e a função objetivo é a minimização do *makespan*, o problema é exatamente o 1MPS. De acordo com o conhecimento do autor, existem apenas três trabalhos que consideram o 1MPS como objeto de estudo, são eles Ángel-Bello et al. (2011b), Ángel-Bello et al. (2011a) e Pacheco, Ángel-Bello e Álvarez (2013). Tais trabalhos são explicados mais detalhadamente.

Ángel-Bello et al. (2011b) propuseram um MILP com número polinomial de variáveis e restrições. Tal formulação faz uso de variáveis de arcos, ou seja, variáveis que modelam a sequência entre tarefas. O desempenho do MILP foi melhorado com o auxílio de duas técnicas. A primeira foi a adição de desigualdades válidas à formulação a fim de melhorar sua relaxação linear. A segunda se deu por meio de um método heurístico que fornece soluções viáveis como dados de entrada para o resolvedor. Para executar seus experimentos, os outores utilizaram um conjunto de instâncias gerado no seu trabalho. Tal conjunto de instâncias é descrito no Capítulo 5. Os resultados computacionais mostram que o MILP leva, em média, 3,94 segundos para resolver instâncias de 10 tarefas.

Ángel-Bello et al. (2011a) propuseram outro MILP, modelado por novas variáveis de arcos. Tal formulação tira proveito das desigualdades desenvolvidas em Ángel-Bello et al. (2011b). Além disso, um algoritmo heurístico de duas fases baseado no *greedy randomized adaptative search procedure* (GRASP) foi desenvolvido. A primeira corresponde a um procedimento iterativo de construção e melhoramento da solução corrente à medida que cada nova tarefa é adicionada à solução. A segunda fase corresponde a um procedimento de busca local aplicado à solução completa gerada pela primeira fase.

Tabela 2.1: Trabalhos relacionados ao 1MPS

Autores	**Objetivo**	**Características**	**Algoritmos**
Benmansour et al. (2014)	Min soma do atraso e adiantamento máximo ponderados	Datas de entrega e liberação	MILP heurística
Liao e Chen (2003)	Min max atraso	Datas de entrega	B&B, heurística
Sbihi e Varnier (2008)	Min max atraso	Períodos flexíveis	B&B, heurística
Chen (2006)	Min tempo de fluxo total	-	B&B, heurística
Chen (2006a)	Min tempo de fluxo médio	Períodos flexíveis	MILP, heurística
Chen (2007)	Min tempo de fluxo total e max atraso	preempção	B&B, heurística
Chen (2009)	Min # tarefas atrasadas	-	B&B, heurística
Lee e Kim (2012)	Min # tarefas atrasadas	-	MILP, heurística
Liu et al. (2016)	Min # tarefas atrasadas	-	B&B
Ji, He e Cheng (2007)	Min makespan	-	Alg. aproximativo
Chen (2008)	Min makespan	Períodos flexíveis	MILP, heurística
Xu, Yin e Li (2009)	Min makespan	Períodos flexíveis	Alg. aproximativo
Low, Hsu e Su (2010)	Min makespan	-	Colônia de formigas
Hsu, Low e Su (2010)	Min makespan	Max # tarefas	heurísticas
Ángel-Bello et al. (2011b)	Min makespan	Tempos de *setup*	B&B, heurística
Ángel-Bello et al. (2011a)	Min makespan	Tempos de *setup*	GRASP
Pacheco, Ángel-Bello e Álvarez (2013)	Min makespan	Tempos de *setup*	MILP, busca tabu
Xu e Yang (2013)	Min makespan	Máquinas paralelas	MILP, Alg. aproximativo
Zade e Fakhrzad (2013)	Min makespan	-	MILP, alg. genético
Yu, Zhang e Steiner (2014a)	Min makespan	-	Alg. aproximativo
Cui e Lu (2014)	Min makespan	Datas de liberação	B&B
Yu, Zhang e Steiner (2014b)	Min makespan	-	Alg. aproximativo
Li et al. (2017)	Min makespan	Máquinas paralelas	heurística

O estudo mais recente cujo objeto de estudo é o 1MPS foi desenvolvido por Pacheco, Ángel-Bello e Álvarez (2013). Nele é desenvolvido um método heurístico *multi-start* que faz uso de um procedimento de busca tabu para melhoramento da solução. A construção da solução inicial se dá da seguinte maneira: as tarefas são escalonadas usando a regra de menor aumento no custo de inserção. O custo de inserção é calculado levando em consideração a função objetivo do problema (*makespan*) somada de um parâmetro sujeito à um processo de aprendizagem que ocorre durante o procedimento de melhora da solução, que, por sua vez, se dá por meio de chamadas iterativas a estruturas de vizinhança até que um critério de término seja atingido. Uma lista tabu é usada a fim de diversificar o espaço de busca durante os procedimentos de descida. Um movimento é classificado como tabu se esse pode levar ao reestabelecimento de um estado anterior (mas não muito antigo) da solução. Após ser efetuado, um movimento é considerado tabu durante um número específico de iterações do algoritmo. Além disso, Pacheco, Ángel-Bello e Álvarez (2013)

apresentam uma formulação compacta (CF), com desempenho superior àquela apresentada em Ángel-Bello et al. (2011a). CF é descrita em detalhes no Capítulo 3. De acordo com a literatura, dentre os MILPs que descrevem o 1MPS, CF é o de melhor performance, com capacidade de resolver instâncias de 10 tarefas em um tempo computacional médio de 0,59 segundo.

Capítulo 3

Formulação matemática de Pacheco, Ángel-Bello e Álvarez (2013)

A formulação CF considera como *bloco* o conjunto de instantes de tempo compreendido entre o início (ou final) duas atividades de manutenção consecutivas. As variáveis binárias v_{ij}^1 e v_{ij}^2 definem a sequência entre tarefas. Mais precisamente, v_{ij}^1 recebe valor 1 se, e apenas se, a tarefa j é sequenciada logo após a tarefa i no último bloco. De maneira análoga, v_{ij}^2 recebe valor 1 se, e apenas se, a tarefa j é alocada logo após a tarefa i em qualquer bloco anterior ao último. As variáveis y_j recebem valor 1 caso a tarefa j seja processada no último bloco e o número total de blocos é modelado pela variável m. Por fim, as variáveis u_j modelam o tempo de término da tarefa j com relação ao começo do bloco o qual j é processado. CF é explicitamente definida a seguir.

$$\text{minimizar} \quad P(m-1) + \sum_{i=0}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} (s_{ij} + p_j) v_{ij}^1 \tag{3.1}$$

sujeito a

$$\sum_{j \in J} v_{0j}^1 = \sum_{i \in J} v_{i0}^1 = 1 \tag{3.2}$$

$$\sum_{j \in J} v_{0j}^2 = \sum_{i \in J} v_{i0}^2 = m - 1 \tag{3.3}$$

$$\sum_{\substack{j \in J_+ \\ i \neq j}} v_{ij}^1 = y_i \qquad i \in J \tag{3.4}$$

$$\sum_{\substack{i \in J_+ \\ i \neq j}} v_{ij}^1 = y_j \qquad j \in J \tag{3.5}$$

$$\sum_{\substack{j\in J_+\\ i\neq j}} v_{ij}^2 = 1 - y_i \qquad i \in J \tag{3.6}$$

$$\sum_{\substack{i\in J_+\\ i\neq j}} v_{ij}^2 = 1 - y_j \qquad j \in J \tag{3.7}$$

$$u_i - u_j + (P + s_{ij} + p_j)(v_{ij}^1 + v_{ij}^2) + (P - s_{ji} - p_i)(v_{ji}^1 + v_{ji}^2) \leq P \qquad i, j \in J, j \neq i \tag{3.8}$$

$$(s_{0i} + p_i)(v_{0i}^1 + v_{0i}^2) \leq u_i \leq P - (s_{i0} + p_0)(v_{i0}^1 + v_{i0}^2) \qquad i \in J \tag{3.9}$$

$$\sum_{i\in J_+} \sum_{\substack{j\in J_+\\ i\neq j}} (p_i + s_{ij}) v_{ij}^2 \leq P \times m \tag{3.10}$$

$$\sum_{i\in J_+} \sum_{\substack{j\in J_+\\ i\neq j}} (p_i + s_{ij}) v_{ij}^1 \leq P \tag{3.11}$$

$$v_{ij}^k \in \{0, 1\} \qquad k \in \{1, 2\}, \; i, j \in J_+, j \neq i \tag{3.12}$$

$$u_i, y_i \geq 0 \qquad i \in J \tag{3.13}$$

$$m \geq 0. \tag{3.14}$$

A função objetivo (3.1) calcula o *mkaespan* como sendo a soma das durações dos blocos (com exceção do último) e do instante de tempo em que a máquina finaliza o processamento da última tarefa. As restrições (3.2) e (3.3) fazem com que apenas uma tarefa seja processada depois e antes de uma atividade de manutenção. As restrições (3.4)–(3.7) impõe as condições de sequenciamento. As restrições (3.8) limitam a duração máxima de um bloco, e as restrições (3.9) definem o processamento das tarefas e os tempos de *setup*. As restrições (3.12) e (3.13) especificam o domínio das variáveis. Os conjuntos de restrições (3.10) e (3.11) correspondem as desigualdades válidas desenvolvidas por Ángel-Bello et al. (2011b) e Pacheco, Ángel-Bello e Álvarez (2013).

A relaxação linear de CF pode ser melhorada fazendo uso dos planos de cortes conhecidos como *k-path cuts*, introduzidos por Kohl et al. (1999) para o problema de roteamento de veículos com janelas de tempo (VRPTW). Seja $K(S)$ o número mínimo de blocos necessários para escalonar um subconjunto de tarefas $S \subset J$. Os *k-path cuts* consistem em impor que $\sum_{i\in J\setminus\{S\}} \sum_{j\in S} (v_{ij}^1 + v_{ij}^2) \geq K(S)$. O cálculo de $K(S)$ é $\mathcal{NP}$-difícil, pois para tanto é necessário resolver o problema $1|s_{ij}|C_{max}$ (ATSP) em S. Quando $K(S) = 2$, essas desigualdades são conhecidas como *2-path cuts*, que são escaláveis em termos de separação exata, pois $|S|$ é comumente pequeno. Foi implementado um algoritmo que utiliza *2-path cuts* com um auxílio de um procedimento de separação baseado naqueles propostos em

Kohl et al. (1999) para encontrar cortes violados. Os experimentos revelaram que, na prática, só existem ganhos (marginais) em instâncias com pequenos valores de P, portanto são reportados apenas os resultados para a versão que não faz uso dos *2-path cuts*.

Capítulo 4

Formulações matemáticas propostas

Como mostrado em Pacheco, Ángel-Bello e Álvarez (2013), o 1MPS pode ser formulado como um MILP de número polinomial de variáveis e restrições; modelos dessa natureza podem ser resolvidos diretamente usando um resolvedor comercial de programação inteira. Porém, o modelo de Pacheco, Ángel-Bello e Álvarez (2013) não é capaz de resolver algumas instâncias envolvendo apenas 20 tarefas. A seguir são apresentados três modelos alternativos para o 1MPS.

Embora os três modelos apresentados neste capítulo descrevam formalmente o 1MPS, os dois primeiros não podem ser resolvidos diretamente por resolvedores comerciais, pois o número de blocos necessário para escalonar todas as atividades é um dos índices de algumas restrições. Na seção 4.3, é explicado como os modelos podem ser resolvidos por meio de um procedimento iterativo.

Todos os modelos propostos neste capítulo fazem uso de variáveis de arcos indexados no tempo, que relacionam a precedência de tarefas a instantes de tempo. O número de variáveis é tipicamente proporcional à dimensão do horizonte de tempo (pseudo-polinomial), e é, portanto, arbitrariamente grande. O sucesso das formulações baseadas em variáveis de arcos indexados no tempo é devido à sua forte relaxação linear, quando comparada com formulações de número polinomial de variáveis, porém sua fraqueza é sua dimensão. O primeiro modelo de variáveis indexadas no tempo para um problema de sequenciamento com apenas um processador foi proposto por Dyer e Wolsey (1990). Desde então, a pesquisa se voltou à adoção de formulações indexadas no tempo resolvidas por algoritmos que mitigam a explosão da dimensão do espaço de busca.

Exemplos de métodos baseados em formulações de arcos indexados no tempo bem sucedidos são: Sourd (2009), Tanaka, Fujikuma e Araki (2009) e Pessoa et al. (2010).

Nogueira, Carvalho e Ravetti (2014) comparam a performance de formulações para problemas de sequenciamento com apenas um processador, datas de liberação e tempos de *setup* dependentes da sequência; tais resultados confirmam que formulações de arcos indexados no tempo produzem limitantes inferiores de melhor qualidade e superam outras formulações para instâncias de até 15 tarefas.

No que se segue, T_{ij} denota o conjunto de instantes de tempo os quais uma tarefa ou manutenção j pode ser escalonada imediatamente após outra tarefa ou manutenção i sem que nenhuma restrição do problema seja violada, por exemplo, uma tarefa j não pode ser escalonada imediatamente após outra tarefa i se o tempo de início de j somado com seu tempo de processamento p_j ultrapasse o instante de tempo inicial da atividade de manutenção posterior mais próxima. Note que o conjunto T_{ij} pode ser diferente dependendo da formulação em questão.

4.1 ATF

O primeiro modelo desenvolvido neste trabalho é chamado de ATF, acrônimo para *arc-time-formulation*. Seja o conjunto $J_{ATF} = J \cup \{0, M_1, \ldots, M_{m-1}\}$ o qual 0 é o estado inicial e final da rede e $M_1, \ldots, M_{m-1}$ são as atividades de manutenção a serem realizadas. As variáveis de decisão binárias x_{ij}^t recebem valor 1 caso $j \in J_{ATF}$ comece a ser processado exatamente após $i \in J_{ATF}$ no instante de tempo $t \in T_{ij}$. Como o tempo de início de cada atividade de manutenção é fixo e preempção não é permitido, instantes de tempo ociosos podem ocorrer. Para considerar os tempos ociosos, são introduzidas variáveis to tipo x_{jj}^t. Tais variáveis significam que a máquina está ociosa entre os instantes de tempo $t-1$ e t. Sem prejuízo de generalidade, defina p'_{ij}, que recebe valor 1 caso $j = i$ e p_j caso contrário. Por fim, o sequenciamento começa e termina na manutenção 0 após processar todas as outras tarefas. ATF é apresentado no que se segue:

$$minimizar \quad \sum_{i \in J} \sum_{t \in T_{i0}} (t - s_{i0}) x_{i0}^t \tag{4.1}$$

sujeito a

$$\sum_{j \in J} x_{0j}^{s_{0j}} = 1 \tag{4.2}$$

$$\sum_{\substack{i \in J_{ATF} \\ i \neq j}} \sum_{t \in T_{ij}} x_{ij}^t = 1 \qquad j \in J_{ATF} \tag{4.3}$$

$$\sum_{\substack{j\in J_{ATF}\\ t\in T_{ji}}} x_{ji}^{t} - \sum_{\substack{j\in J_{ATF}\\ t'\in T_{ij}\\ t'=t+p'_{ij}+s_{ij}}} x_{ij}^{t'} = 0 \qquad i \in J_{ATF}, t = 0, \ldots, mP \tag{4.4}$$

$$x_{ij}^{t} \in \{0,1\} \qquad i,j \in J_{ATF}, t \in T_{ij} \tag{4.5}$$

$$m \geq 0. \tag{4.6}$$

A função objetivo (4.1) calcula o *makespan*. A restrição (4.2) força que apenas uma tarefa seja processada imediatamente após a manutenção 0. As restrições (4.3) garantem que cada tarefa e manutenção sejam executadas apenas uma vez. As restrições (4.4) asseguram a conservação do fluxo. Note que essas restrições também asseguram que cada atividade de manutenção será executada no instante de tempo correto. Além disso, o número de restrições depende explicitamente de m, o que impossibilita resolver o modelo diretamente com um resolvedor comercial. As restrições (4.5) e (4.6) definem o domínio das variáveis.

É possível representar graficamente a solução gerada pelo modelo ATF se os vértices da rede forem plotados em função do tempo. Tal representação descreve a sequência de tarefas processadas e suas durações como um caminho da rede, proporcionando uma interpretação visual eficaz das soluções e da dimensão do modelo. Um exemplo dessa representação é fornecido pela Figura 4.1. A Tabela 4.1 contém os dados da instância usada como exemplo, na qual a duração P do bloco é de 8 unidades de tempo, e as atividades de manutenção duram apenas uma unidade. Quatro tarefas devem ser sequenciadas e as tabelas fornecem os tempos de processamento e de *setup*. Note que o *setup* também é necessário mesmo que uma tarefa seja processada imediatamente antes ou depois de uma manutenção ou do estado 0. A figura 4.1 mostra a rede que representa a solução correspondente à sequencia $(3, 2, 1, 4)$ das tarefas. Os arcos verticais correspondem aos tempos de *setup*, enquanto que os arcos horizontais correspondem a tempos de processamento ou tempos ociosos.

Tabela 4.1: Instância do 1MPS com $P = 8$

Tempos de *setup*					
	$0/M_i$	1	2	3	4
$0/M_i$	0	2	1	2	1
1	1	0	2	3	1
2	1	1	0	3	3
3	2	2	1	0	2
4	1	2	4	2	0

Tempos de processamento	
j	p_j
p_0/p_{M_i}	1
1	1
2	1
3	2
4	1

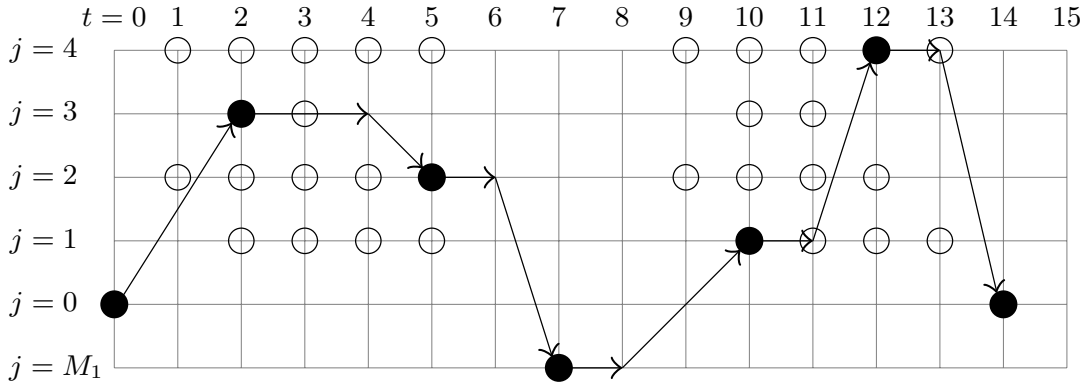


Figura 4.1: Representação da solução da formulação ATF.

4.1.1 ATF-M

O segundo modelo, ATF-M ("M" faz referência ao índice dado à atividade de manutenção), possui como principal característica o índice extra de suas variáveis, usado para identificar os blocos. Seja $K = \{1, \ldots, m\}$ o conjunto de blocos. As variáveis binárias z_{ij}^{tk} recebem valor 1 caso a tarefa $j \in J$ comece a ser processada imediatamente após outra tarefa $i \in J_+$ no bloco $k \in K$ e no instante de tempo $t \in T_{ij}$. Note que não há necessidade de modelar tempos ociosos na formulação ATF-M, pois os caminhos da rede podem retornar à atividade de manutenção antes do seu tempo de início. Portanto, as variáveis z_{jj}^{tk} não são necessárias. ATF-M é apresentado no que se segue:

$$\text{minimizar} \quad P(m-1) + \sum_{i \in J} \sum_{t \in T_{i0}} (t - s_{i0}) z_{i0}^{tm} \tag{4.7}$$

sujeito a

$$\sum_{k \in K} \sum_{i \in J_+} \sum_{t \in T_{ij}} z_{ij}^{tk} = 1 \qquad j \in J \tag{4.8}$$

$$\sum_{t \in T_{ij}} \sum_{j \in J} z_{j0}^{tk} = \sum_{j \in J} z_{0j}^{s_{0j},k} = 1 \qquad k \in K \tag{4.9}$$

$$\sum_{\substack{j \in J_+ \\ t \in T_{jik}}} z_{ji}^{tk} - \sum_{\substack{j \in J_+ \\ t' \in T_{ij} \\ t' = t + p_i + s_{ij}}} z_{ij}^{t'k} = 0 \qquad k \in K, i \in J$$

$$t = 0, \ldots, P - p_0 \tag{4.10}$$

$$z_{ij}^{tk} \in \{0, 1\} \qquad i, j \in J_+, i \neq j$$

$$k \in K, t \in T_{ij} \tag{4.11}$$

$$m \geq 0. \tag{4.12}$$

As restrições (4.8) fazem com que cada tarefa seja executada exatamente uma vez. As restrições (4.9) definem que, para cada bloco $k \in K$, apenas uma tarefa deve ser processada imediatamente após e imediatamente antes do estado 0. As restrições (4.10) asseguram a conservação de fluxo. Note que o número de restrições (4.9) e (4.10) depende de m, impossibilitando que ATF-M seja resolvida diretamente por resolvedores comerciais. Não é apresentada graficamente a solução retornada por ATF-M, pois a visualização das variáveis de quatro dimensões seria de difícil interpretação, portanto, não-informativa.

4.1.2 ATF-P

O último modelo proposto, ATF-P, considera apenas um bloco de $P - p_0$ unidades de tempo (daí o "P"). As variáveis de decisão são as mesmas usadas por ATF. A principal diferença entre ATF e ATF-P é a dimensão do horizonte de tempo, que em ATF-P é restringido à $P - p_0$ unidades de tempo, em contraste com mP unidades em ATF. Para que seja possível modelar o 1MPS utilizando o horizonte de tempo restrito e ainda assim computar o *makespan*, é necessário definir m caminhos paralelos ao longo da rede da seguinte maneira. O estado 0 é conectado com exatamente $m - 1$ caminhos e apenas um é conectado com o estado auxiliar $0'$, que representa o último bloco, portanto, $J_{ATF\text{-}P} = J \cup \{0, 0'\}$. Como em ATF-M, as variáveis x^t_{jj} não são necessárias em ATF-P, pois os tempos ociosos não precisam ser levados em conta. ATF-P é apresentado no que se segue:

$$\text{minimizar} \quad P(m-1) + \sum_{\substack{i \in J \\ t \in T_{i0'}}} (t - s_{i0'}) x^t_{i0'} \tag{4.13}$$

sujeito a

$$\sum_{j \in J} x_{0j}^{s_{0j}} = m - 1 \tag{4.14}$$

$$\sum_{j \in J} x_{0'j}^{s_{0'j}} = 1 \tag{4.15}$$

$$\sum_{\substack{i \in J_{ATF\text{-}P} \\ i \neq j}} \sum_{t \in T_{ij}} x^t_{ij} = \begin{cases} m-1 & \text{if } j = 0 \\ 1 & \text{caso contrário} \end{cases} \qquad j \in J_{ATF\text{-}P} \tag{4.16}$$

$$\sum_{\substack{j \in J_{ATF\text{-}P} \\ t \in T_{ji}}} x^t_{ji} - \sum_{\substack{j \in J_{ATF\text{-}P} \\ t' \in T_{ij} \\ t' = t + p_i + s_{ij}}} x^{t'}_{ij} = 0 \qquad i \in J, t = 0, \ldots, P - p_0 \tag{4.17}$$

$$x^t_{ij} \in \{0, 1\} \qquad i, j \in J_{ATF\text{-}P}, t \in T_{ij} \tag{4.18}$$

$$m \geq 0. \tag{4.19}$$

A função objetivo (4.13) minimiza o *makespan*. A restrição (4.14) garante que $m-1$ tarefas sejam processadas imediatamente após o estado 0. As restrições (4.15) forçam que apenas uma tarefa seja processada imediatamente após o estado auxiliar $0'$ no instante de tempo $t = s_{0'j}$. As restrições (4.16) garantem que todas as tarefas sejam processadas exatamente uma vez, e que o vértice 0 seja o último vértice dos $m-1$ caminhos da rede conectados ao estado 0. As restrições (4.17) asseguram a conservação de fluxo. Por fim, as restrições (4.18) e (4.19) definem o domínio das variáveis. Note que, ao contrário de ATF e ATF-M, ATF-P pode ser resolvida diretamente por resolvedores de programação inteira caso m seja definido como uma variável da formulação.

A solução referente à instância utilizada como exemplo na Seção 4.1 é representada, adotando a rede de ATF-P, pela Figura 4.2. É possível perceber que ATF-P reduz drasticamente o espaço de busca; $m-1$ caminhos são iniciados no vértice 0 e retornam à ele em até $P - p_0$ unidades de tempo, enquanto que o último caminho sai de $0'$ e retorna ao mesmo vértice no mesmo horizonte de tempo. Essa representação enfatiza as semelhanças entre ATF-P e problemas de sequenciamento em máquinas paralelas, com cada um dos blocos representando um processador paralelo.

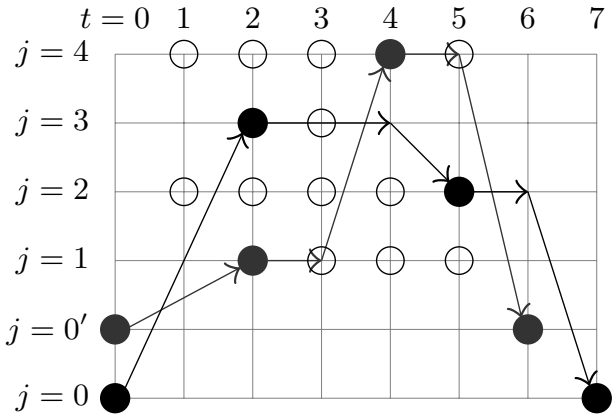


Figura 4.2: Representação da rede retornada por ATF-P

4.2 Dimensão dos modelos

A tabela 4.2 proporciona uma comparação compreensiva dos números de variáveis e restrições para os modelos apresentados. Note que os três modelos propostos neste trabalho possuem um número pseudo-polinomial de variáveis e restrições, enquanto que os mesmos números, para CF, são polinomiais. Todavia, ATF-P possui um número de variáveis e restrições proporcional a $P - p_0$, que, mesmo sendo arbitrariamente grande, é significativamente menor que mP e $m(P - p_0)$ (coeficientes das fórmulas de complexi-

dade de ATF e ATF-M). Pela Tabela 4.2, fica evidente que o tamanho dos modelos com variáveis de arcos indexados no tempo é muito sensível ao horizonte de tempo. ATF-P é sempre menor do que ATF e ATF-M, independentemente dos parâmetros da instância, porém a relação entre ATF-P e CF depende diretamente de tais parâmetros. Note que o desempenho prático dos modelos depende de outros fatores que vão além do número de variáveis ou restrições.

Tabela 4.2: Dimensão dos modelos

Formulação	**#Restrições**	**#Variáveis**
CF	$\|J\|^2 + 5\|J\| + 6$	$2(\|J\| + 1)^2 + 2\|J\| + 1$
ATF	$1 + (\|J\| + m) + Pm(\|J\| + m)$	$(\|J\| + m)^2(\max_{ij}\{\|T_{ij}\|\})$
ATF-M	$\|J\|m(P - p_0) + \|J\| + m$	$(\|J\| + 1)^2(P - p_0)m$
ATF-P	$\|J\|(P - p_0 + 1) + 4$	$(\|J\| + 2)^2(P - p_0) + 1$

4.3 Algoritmo iterativo para o 1MPS

O algoritmo proposto neste trabalho resolve iterativamente o 1MPS fixando o número de blocos, começando com um limitante inferior trivial, até que o menor número de manutenções que viabilize o escalonamento de todas as atividades seja atingido. Em cada iteração existem dois possíveis resultados, o primeiro é a solução do problema em sua otimalidade pelo MILP associado à formulação usada e o número de blocos fornecido, terminando o algoritmo; o segundo é a identificação de inviabilidade, significando que ao menos uma manutenção extra é necessária para que o MILP seja viável.

Cada formulação integrada ao algoritmo iterativo assume que o número de manutenções a serem executadas é conhecido *a priori*. Isso permite resolver ATF e ATF-M utilizando um resolvedor, assim como ATF-P e CF.

O algoritmo começa computando o limitante inferior trivial (4.20) para o número de blocos necessário para processar todas as tarefas.

$$\mathsf{m} = \left\lfloor \frac{|J| \times \min_{i,j \in J_+} s_{ij} + \sum_{j \in J} p_j}{P - p_0} \right\rfloor . \tag{4.20}$$

Tanto os modelos apresentados neste capítulo quanto CF são inicializados assumindo que exatamente **m** blocos são necessários para completar o escalonamento e realizar o *setup* final para a última manutenção. Se a inviabilidade é detectada (podendo ser depois de ter encontrado um limitante inferior), **m** é incrementado em uma unidade e o modelo é resolvido novamente; caso contrário, **m** é o número ótimo de manutenções e obtém-se

uma solução viável do modelo.

O número máximo de iterações desse algoritmo é limitado por $|J|$, mas os experimentos computacionais indicam que, normalmente, poucas iterações são necessárias, e que o tempo computacional necessário para identificar a inviabilidade para um dado $\mathbf{m}$ é geralmente pequeno. Mais detalhes em relação à performance do algoritmo iterativo são apresentados no Capítulo 5.

Capítulo 5

Resultados Computacionais

Os experimentos foram conduzidos utilizando apenas um núcleo do processador Intel Core i7 com 3.4 Ghz e 16 GB de memória RAM, com o sistema operacional Ubuntu 12.04. Todas as formulações foram resolvidas pelo CPLEX 12.6 usando as seguintes configurações: limitante de tempo de 7200 segundos; e limitante de memória de 10 GB. O mesmo ambiente computacional foi usado para executar os testes referentes a CF.

5.1 Instâncias

Nesta seção são descritas as instâncias usadas nos experimentos. As instâncias propostas por Pacheco, Ángel-Bello e Álvarez (2013) são descritas e, em seguida, o conjunto de instâncias proposto neste trabalho é introduzido.

5.1.1 Instâncias de Pacheco, Ángel-Bello e Álvarez (2013)

O número de tarefas nas instâncias de Pacheco, Ángel-Bello e Álvarez (2013) é $|J| = 10, 12, 15, 20, 30, 40, 50$. Estes autores usaram uma distribuição discreta uniforme para gerar os tempos de processamento e *setup* dentro dos seguintes intervalos: $[2, 8]$; $[4, 12]$; e $[5, 20]$. Para simplificar a notação, é associado um valor S a cada intervalo, com $S = 1, 2$ e 3, respectivamente. Cada intervalo, respectivamente, admite um comprimento diferente para os blocos, que pode ser calculado da seguinte maneira: $\alpha \times d_m$, onde $d_m = \max_{i \in J}\{(s_{0i} + p_i + s_{i0} + p_0)/2\}$ e α pode assumir valores 2,25; 2,5; 3 e 4. Foram geradas 5 instâncias para cada combinação de n, S e comprimento do bloco (P), totalizando $5 \times 7 \times 3 \times 4 = 420$ instâncias.

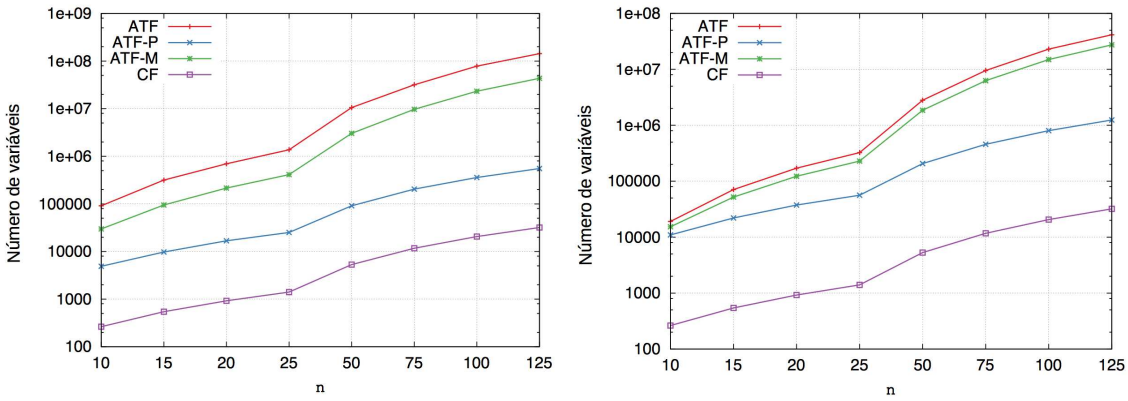
5.1.2 Instâncias propostas neste trabalho

Como mostrado nos experimentos, é possível resolver todas as instâncias propostas por Pacheco, Ángel-Bello e Álvarez (2013) na otimalidade, então um novo conjunto de instâncias foi gerado da seguinte maneira. O número de tarefas foi definido como $|J| = 10, 15, 20, 25, 50, 75, 100, 125$. Os tempos de processamento e *setup* foram gerados aleatoriamente usando uma distribuição discreta uniforme com intervalos $[2, 4]$ $(S = 1)$, $[5, 10]$ $(S = 2)$ e $[10, 20]$ $(S = 3)$. O comprimento do bloco é calculado como $\alpha \times \max_{i,j \in J_+}\{p_i + s_{ij}\}$, onde α pode assumir os valores 2, 3 e 4. Cinco instâncias foram geradas para cada combinação de n, S e α, totalizando $5 \times 8 \times 3 \times 3 = 360$ instâncias. Esse conjunto de instâncias difere daquele proposto por Pacheco, Ángel-Bello e Álvarez (2013) pelas seguintes razões: esse não possui desigualdades triangulares satisfeitas, pois em situações práticas isso não necessariamente é respeitado; os intervalos dos valores para tempos de processamento e *setup* não se sobrepõe; e o comprimento do bloco é computado de maneira diferente, usando parâmetros diferentes.

As Figuras 5.1 e 5.2 mostram, em escala logarítmica, o número de variáveis e restrições de cada formulação para as instâncias propostas. A fim de manter o trabalho conciso, os gráficos são apresentados apenas para $\alpha = 2$ e $\alpha = 4$. Note que a dimensão dos modelos é diretamente proporcional ao valor de α. Como apresentado no Apêndice A, a performance das formulações é fortemente influenciada pelo valor de tal parâmetro. De acordo com a Tabela 4.2, observa-se que o número de variáveis e restrições de CF dependem apenas do número de tarefas. Porém, não é o caso das formulações de arcos indexados no tempo, cujos números de variáveis e restrições são evidentemente sensíveis à α, que regula a dimensão dos blocos. Mais precisamente, o número médio de variáveis e restrições de ATF e ATF-M tendem a aumentar rapidamente para pequenos valores de α, o que não acontece com ATF-P. É interessante observar que, para grandes valores de n, o número médio de restrições de ATF-P tende a ser menor que CF.

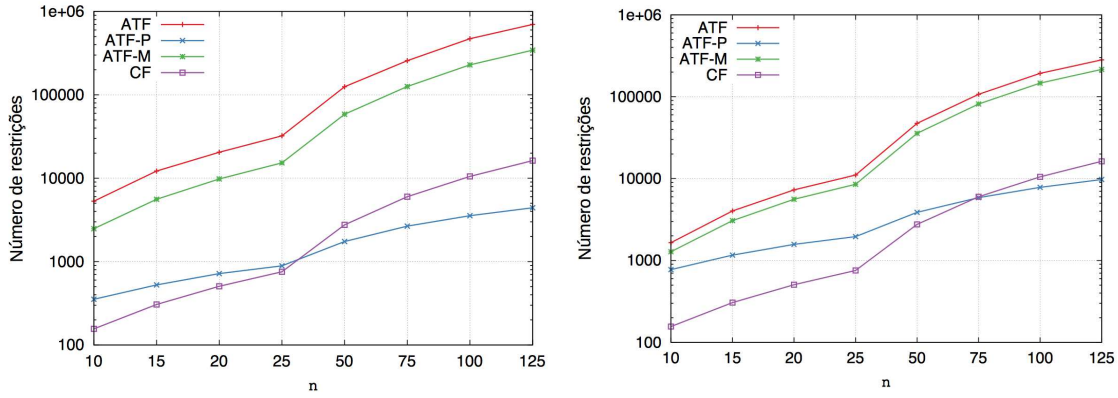
5.2 Algoritmo iterativo *versus* formulação completa

Nesta seção o desempenho do algoritmo iterativo é avaliado. Para tanto, as formulações ATF-P e CF (modelos que podem ser resolvidos diretamente pelo CPLEX) foram testadas com e sem o algoritmo. Note que não é possível estender essa comparação aos modelos ATF e ATF-M, pois seu número de restrições depende de m. A Tabela 5.1 apresenta os resultados referentes às instâncias propostas por Pacheco, Ángel-Bello e Álvarez



(a) $\alpha = 2$ (b) $\alpha = 4$

Figura 5.1: Número médio de variáveis



(a) $\alpha = 2$ (b) $\alpha = 4$

Figura 5.2: Número médio de restrições

(2013), e a Tabela 5.2 reporta os resultados referentes às instâncias propostas neste trabalho. As tabelas são organizadas da seguinte maneira: cada linha apresenta a média dos resultados obtidos para o conjunto de instâncias com mesmo valor de n e α. Os resultados de ATF-P (com e sem o algoritmo iterativo) são apresentados primeiro, seguidos dos resultados obtidos com CF. Para cada método, são reportados: o número de soluções ótimas **#opt** alcançadas, a média dos tempos de execução $\boldsymbol{t^*}$**(s)** para as instâncias em que ambos os métodos alcançaram a solução ótima, e a média do *gap* percentual **Gap*****(%)** em relação à relaxação linear para as instâncias em que ambos os métodos não acham a solução ótima. Note que para as instâncias de Pacheco, Ángel-Bello e Álvarez (2013), a formulação ATF-P, com e sem o algoritmo iterativo, sempre encontra a solução ótima, portanto apenas a coluna de tempo é reportada.

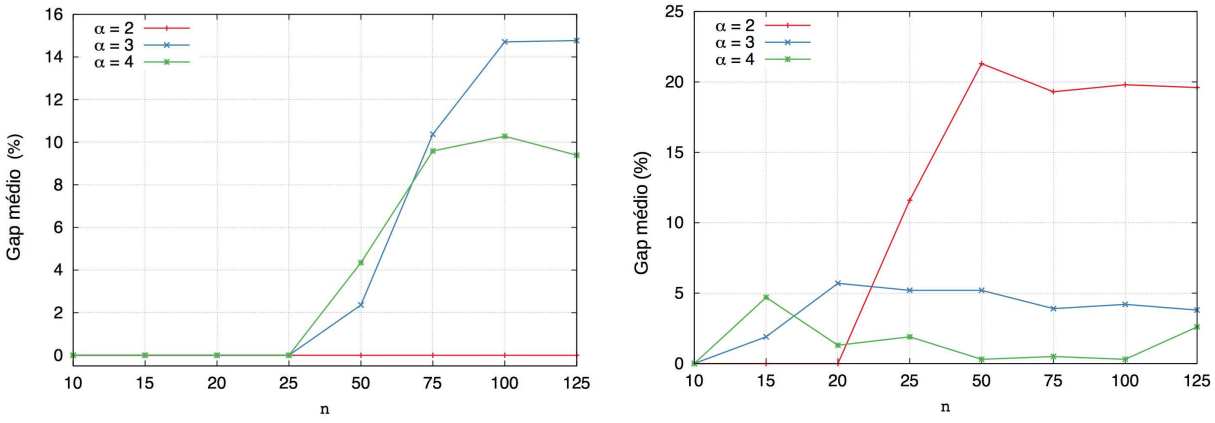
O algoritmo iterativo é sistematicamente capaz de encontrar ao menos o mesmo número de soluções ótimas encontradas pelas formulações completas e em um tempo de execução inferior. Quando ambos os métodos não encontram a solução ótima, o algoritmo iterativo retorna soluções com menor *gap* percentual. Tendo em vista que, empiricamente, esse método se mostrou mais eficiente, as comparações apresentadas na Seção 5.3 fazem uso do algoritmo iterativo para todas as formulações.

5.3 Resultados condensados

Nesta seção são apresentados os resultados condensados para todas as formulações, que são resolvidas com o algoritmo iterativo. Primeiramente, é analisada a influência do parâmetro α no comportamento do *gap*. Observa-se que o desempenho de ATF-M e CF é influenciado por α, como mostra a Figura 5.3. ATF e ATF-P não estão inclusos nesta análise pois a primeira formulação retorna valores de *gap* muito altos para as instâncias grandes, e a mesma métrica para ATF-P não se distinguiria da linha de *gap* 0. ATF-M retorna melhores limitantes inferiores para valores pequenos de α, enquanto que CF se comporta de maneira oposta, especialmente para $\alpha = 4$, onde CF é capaz de encontrar limitantes inferiores de qualidade relativamente boa. Como os limitantes superiores de CF não são de boa qualidade, CF não é capaz de provar a otimalidade para instâncias com $\alpha = 4$ e $n > 25$. Isso pode ser explicado pelo fato de que o *Branch and Bound* abre um número muito elevado de nós em sua árvore, algumas vezes excedendo o limite de memória de 10 GB antes de atingir o limite de tempo de 7200 segundos para $n > 20$. Estes resultados mostram que a qualidade dos limitantes inferiores de CF também é sensível à α, mesmo que os números de variáveis e restrições deste modelo não sejam funções de α.

Tabela 5.1: Comparação entre o algoritmo iterativo e as formulações completas para as instâncias de Pacheco, Ángel-Bello e Álvarez (2013)

		CF						ATF-P	
		Form. completa			Iterativo			Form. completa	Iterativo
n	α	# opt	t^* (s)	Gap* (%)	# opt	t^* (s)	Gap* (%)	t^* (s)	t^* (s)
10	2,25	15	5,1	-	15	1,6	-	0,0	0,0
	2,50	15	1,0	-	15	0,5	-	0,0	0,0
	3,00	15	5,2	-	15	0,1	-	0,1	0,1
	4,00	15	1,1	-	15	0,1	-	0,7	0,4
12	2,25	15	16,6	-	15	5,7	-	0,0	0,0
	2,50	15	9,4	-	15	3,1	-	0,0	0,1
	3,00	15	17,8	-	15	0,4	-	1,0	0,1
	4,00	15	9,5	-	15	0,1	-	0,8	0,7
15	2,25	15	147,7	-	15	71,2	-	0,0	0,1
	2,50	15	21,6	-	15	11,4	-	0,1	0,1
	3,00	15	150,9	-	15	2,6	-	1,1	0,3
	4,00	15	22,3	-	15	0,2	-	4,6	0,6
20	2,25	7	2580,0	11,3	8	1471,4	9,6	0,1	0,1
	2,50	13	1373,7	8,2	14	1191,7	0,7	0,6	0,3
	3,00	15	64,2	-	15	86,7	-	2,0	0,7
	4,00	15	4,1	-	15	4,9	-	151,6	2,3
30	2,25	0	-	10,7	0	-	9,8	0,7	0,7
	2,50	1	648,4	7,9	1	2241,3	7,5	1,8	1,0
	3,00	4	215,2	3,2	4	495,5	3,3	4,9	3,0
	4,00	14	900,7	-	15	770,3	-	15,8	10,8
40	2,25	0	-	11,4	0	-	11,3	1,2	1,0
	2,50	0	-	10,0	0	-	9,7	3,2	3,3
	3,00	0	-	5,6	0	-	5,6	6,0	10,7
	4,00	4	270,8	1,5	4	493,8	1,4	65,6	37,0
50	2,25	0	-	13,5	0	-	12,9	1,4	2,5
	2,50	0	-	11,2	0	-	10,8	25,0	3,8
	3,00	0	-	5,7	0	-	5,5	136,8	15,0
	4,00	2	113,1	2,4	3	822,1	2,4	659,8	98,5
Méd.			229,2	8,1		184,1	7,7	38,7	6,9
Tot.		255			259				



(a) ATF-M (b) CF

Figura 5.3: Comparação dos *gaps* percentuais médios de ATF-M e CF para diferentes valores de α

Mesmo levando clara vantagem em termos de limitantes inferiores e superiores, o desempenho de ATF-P também depende do valor de α, como evidenciado pelo fato de que ATF-P resolve todas as instâncias com $\alpha = 2$, porém não é capaz de resolver algumas instâncias com $\alpha = 3$ e $\alpha = 4$ (para $n = 75$ e $n = 100$). O *gap* médio para cada valor de n

Tabela 5.2: Comparação entre o algoritmo iterativo e as formulações completas para as instâncias propostas neste trabalho

		CF						ATF-P					
		Form. completa			Iterativo			Form. completa			Iterativo		
n	α	# opt	t^* (s)	Gap* (%)	# opt	t^* (s)	Gap* (%)	# opt	t^* (s)	Gap* (%)	# opt	t^* (s)	Gap* (%)
	2	15	0,7	-	15	0,6	-	15	0,0	-	15	0,1	-
10	3	15	3,0	-	15	0,7	-	15	0,1	-	15	0,1	-
	4	15	1,8	-	15	1,8	-	15	4,8	-	15	0,4	-
	2	15	78,1	-	15	32,2	-	15	0,0	-	15	0,2	-
15	3	9	207,1	1,8	10	113,2	1,9	15	0,3	-	15	0,3	-
	4	14	34,2	4,5	14	98,4	4,7	15	8,5	-	15	1,3	-
	2	12	861,8	-	15	318,1	-	15	0,1	-	15	0,3	-
20	3	4	3366,5	5,3	3	3432,7	5,7	15	0,8	-	15	0,9	-
	4	9	1353,3	1,1	10	1165,1	1,3	15	17,9	-	15	4,8	-
	2	3	669,8	15,3	7	114,2	11,6	15	0,1	-	15	0,5	-
25	3	1	1745,8	5,5	1	1928,1	5,2	15	2,6	-	15	2,0	-
	4	6	940,2	1,7	5	692,6	1,9	15	25,8	-	15	14,5	-
	2	0	-	21,3	0	-	21,3	15	0,3	-	15	3,9	-
50	3	0	-	5,2	0	-	5,2	15	20,1	-	15	20,0	-
	4	0	-	0,4	0	-	0,3	15	651,8	-	15	545,0	-
	2	0	-	19,3	0	-	19,3	15	0,6	-	15	12,2	-
75	3	0	-	3,9	0	-	3,9	15	91,5	-	14	142,9	-
	4	0	-	0,5	0	-	0,5	10	1173,0	1,0	11	460,2	0,5
	2	0	-	19,8	0	-	19,8	15	1,0	-	15	27,6	-
100	3	0	-	4,2	0	-	4,2	15	271,7	-	14	155,5	-
	4	0	-	0,3	0	-	0,3	10	1892,4	1,4	10	649,7	0,5
	2	0	-	19,7	0	-	19,6	15	3,5	-	15	56,7	-
125	3	0	-	3,8	0	-	3,8	14	948,0	-	14	746,1	-
	4	0	-	2,6	0	-	2,6	6	1866,6	25,8	9	1024,0	25,5
Méd.			385,1	7,7		287,7	7,6		203,8	10,0		120,4	9,4
Tot.		118			125			340			342		

é sempre menor que 1%, demonstrando a boa qualidade dos limitantes de ATF-P. Além disso, para melhor ilustrar a influencia de α no seu desempenho, a Figura 5.4 apresenta (em escala logarítmica) o tempo médio de execução, em segundos, gasto pelo algoritmo iterativo executando tal formulação para diferentes valores de α.

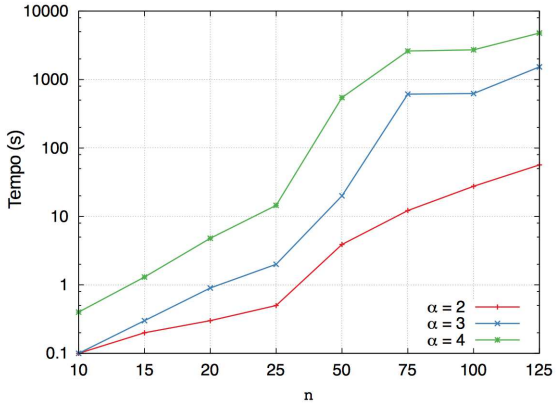


Figura 5.4: Tempo médio de execução em segundos gastos pelo algoritmo iterativo com ATF-P

As Tabelas 5.3 e 5.4 resumem os resultados obtidos em relação às instâncias de Pacheco, Ángel-Bello e Álvarez (2013) e àquelas propostas neste trabalho, respectivamente. Nesse caso, **Gap(%)** é o *gap* percentual médio entre o limitante inferior do modelo em

questão e o melhor limitante superior conhecido; $\boldsymbol{t}$**(s)** representa o tempo médio de execução em segundos; e ($\mathbf{Gap_r}$) é o *gap* percentual médio da raiz do *Branch and Bound* para a última iteração do algoritmo (também em relação à melhor solução conhecida), permitindo uma noção acurada da qualidade da relaxação linear de cada formulação. O símbolo "-" significa que não foi possível encontrar nenhum limitante inferior para o grupo de instâncias em questão dentro do tempo limite de duas horas, fazendo com que o cálculo do Gap(%) seja impossível.

Vale mencionar que os valores de Gap(%) estão associados com a última iteração do algoritmo, ou seja, aquela cujo número de manutenções é o mínimo viável. Os tempos de execução reportados para tais formulações correspondem ao tempo total de execução do algoritmo. Um tempo limite de 7200 segundos e um limite de memória de 10 GB foram impostos. Note que os valores de Gap(%) são calculados com respeito ao melhor limitante superior de que se tem conhecimento, portanto, Gap(%) $= 0,00$ não significa que a solução ótima foi encontrada, pois para que o Gap(%) seja 0%, basta que o limitante inferior seja igual ao melhor limitante superior conhecido, que pode ter sido encontrado por outro modelo. Além disso, um tempo de execução menor que 7200s não implica otimalidade, pois a execução pode ter sido interrompida devido ao critério de parada relacionado ao uso de memória.

Tabela 5.3: Resultados condensados para as instâncias de Pacheco, Ángel-Bello e Álvarez (2013)

	CF				ATF				ATF-M				ATF-P			
n	Gap (%)	Gap_r (%)	t (s)	# opt	Gap (%)	Gap_r (%)	t (s)	# opt	Gap (%)	Gap_r (%)	t (s)	# opt	Gap (%)	Gap_r (%)	t (s)	# opt
10	0,00	9,19	0,59	60	0,00	0,14	1,6	60	0,00	0,00	0,5	60	0,00	0,00	0,1	60
12	0,00	8,87	2,34	60	0,00	0,35	3,5	60	0,00	0,59	1,1	60	0,00	0,00	0,2	60
15	0,00	8,20	21,37	60	0,00	0,64	9,6	60	0,00	0,66	2,3	60	0,00	0,02	0,3	60
20	1,14	9,24	1551,79	52	0,00	0,90	76,3	60	0,00	0,97	13,8	60	0,00	0,24	0,9	60
30	4,79	7,82	5153,43	20	0,00	1,38	707,0	60	0,00	1,21	205,1	60	0,00	0,17	3,9	60
40	6,89	7,49	4463,76	4	0,67	0,92	4345,4	37	0,07	0,89	1190,9	56	0,00	0,45	13,0	60
50	7,76	9,11	3762,19	3	0,02	0,64	4530,9	38	0,12	0,76	2208,1	49	0,00	0,29	29,9	60
Total	-	-	-	259	-	-	-	377	-	-	-	405	-	-	-	420

Tabela 5.4: Resultados condensados para as instâncias propostas neste trabalho

	CF				ATF				ATF-M				ATF-P			
n	Gap (%)	Gap_r (%)	t (s)	# opt	Gap (%)	Gap_r (%)	t (s)	# opt	Gap (%)	Gap_r (%)	t (s)	# opt	Gap (%)	Gap_r (%)	t (s)	# opt
10	0,00	7,93	1,04	45	0,00	0,19	11,3	45	0,00	0,00	1,9	45	0,00	0,03	0,2	45
15	0,32	9,30	1172,64	39	0,00	0,42	181,3	45	0,00	0,11	10,9	45	0,00	0,00	0,6	45
20	1,55	10,74	3479,09	28	0,00	0,33	1187,4	45	0,00	0,28	65,5	45	0,00	0,02	2,0	45
25	4,04	10,62	5379,65	13	0,09	0,35	3425,7	30	0,00	0,26	238,2	45	0,00	0,04	5,7	45
50	8,96	9,13	4532,21	0	-	-	-	-	2,24	0,88	2889,0	32	0,00	0,07	189,6	45
75	7,89	7,92	5233,19	0	-	-	-	-	6,66	1,84	5091,1	19	0,05	0,08	1078,8	40
100	8,12	8,12	6256,39	0	-	-	-	-	8,93	1,63	6147,5	12	0,06	0,07	1161,5	39
125	8,65	8,67	6925,65	0	-	-	-	-	10,51	10,51	6815,0	3	0,84	0,85	1713,2	38
Total	-	-	-	125	-	-	-	165	-	-	-	246	-	-	-	342

Considerando as 420 instâncias de Pacheco, Ángel-Bello e Álvarez (2013), ATF, ATF-

M, ATF-P e CF resolvem na otimalidade, respectivamente, 89,8%, 96,4%, 100,0%, e 61,7% do número total de instâncias. Para as 360 instâncias propostas, ATF, ATF-M, ATF-P, e CF são capazes de resolver na otimalidade, respectivamente, 45,8%, 68,3%, 95,0%, e 34,7% do número total de instâncias. Além disso, a qualidade da relaxação linear das formulações de arcos indexados no tempo, na última iteração do algoritmo, é superior àquelas de CF. No caso de ATF-P, o *gap* percentual médio é sempre inferior a 1,0%.

Capítulo 6

Conclusões

Este trabalho considerou o problema de sequenciamento em uma máquina com manutenções periódicas e tempos de *setup* dependentes da sequência de tarefas (1MPS). Esse problema de otimização combinatória foi resolvido integrando formulações de arcos indexados no tempo com um algoritmo iterativo. Tal algoritmo varia o número de manutenções necessárias para escalonar todas as tarefas, começando com um limitante inferior trivial. A cada iteração é resolvido uma formulação de arcos indexados no tempo, que determina se o número de manutenções é viável (portanto ótimo), ou se é necessário pelo menos uma manutenção extra. Caso a inviabilidade seja detectada, o número de manutenções é incrementado em uma unidade e o procedimento é repetido até que uma solução viável seja encontrada.

Os três modelos de arcos indexados no tempo introduzidos neste trabalho são simples devido ao horizonte de tempo claramente definido pelo algoritmo (função dos dados de entrada do problema e do número de blocos). Isso possibilita resolver dois modelos (ATF e ATF-M) que não podem ser resolvidos diretamente por resolvedores comerciais. O terceiro modelo, ATF-P, contêm um número de variáveis e restrições que é uma função pseudo-polinomial da duração de apenas um bloco, e não do horizonte de tempo completo. ATF-P pode ser resolvido diretamente pelo CPLEX ou integrado ao algoritmo iterativo.

Extensivos experimentos computacionais confirmam que resolver o 1MPS utilizando qualquer uma das formulações de arcos indexados no tempo deste trabalho é claramente melhor do que o modelo proposto por (PACHECO; ÁNGEL-BELLO; ÁLVAREZ, 2013). ATF-P é o melhor dos quatro modelos usados na comparação, sendo resolvido na sua forma completa ou integrado ao algoritmo iterativo. Ao combinar ATF-P com o algoritmo iterativo, encontra-se as soluções ótimas para todas as instâncias de até 50 tarefas e para todas, menos 18, instâncias dentre as 360 instâncias propostas nesse trabalho.

Referências

AHMADI, E. et al. A multi objective optimization approach for flexible job shop scheduling problem under random machine breakdown by evolutionary algorithms. *Computers & Operations Research*, v. 73, p. 56 – 66, 2016.

ALSYOUF, I. The role of maintenance in improving companies' productivity and profitability. *International Journal of Production Economics*, v. 105, n. 1, p. 70 – 78, 2007. ISSN 0925-5273.

ÁNGEL-BELLO, F. et al. A heuristic approach for a scheduling problem with periodic maintenance and sequence-dependent setup times. *Computers & Mathematics with Applications*, v. 61, p. 797–808, 2011.

ÁNGEL-BELLO, F. et al. A single machine scheduling problem with availability constraints and sequence-dependent setup costs. *Applied Mathematical Modelling*, v. 35, p. 2041–2050, 2011.

BENMANSOUR, R. et al. Minimizing the weighted sum of maximum earliness and maximum tardiness costs on a single machine with periodic preventive maintenance. *Computers & Operations Research*, v. 47, p. 106 – 113, 2014.

CHEN, J. Single-machine scheduling with flexible and periodic maintenance. *Journal of the Operational Research Society*, v. 57, p. 703–710, 2006.

CHEN, J.-S. Optimization models for the machine scheduling problem with a single flexible maintenance activity. *Engineering Optimization*, v. 38, n. 1, p. 53–71, 2006.

CHEN, J.-S. Scheduling of nonresumable jobs and flexible maintenance activities on a single machine to minimize makespan. *European Journal of Operational Research*, v. 190, p. 90–102, 2008.

CHEN, W.-C. Minimizing number of tardy jobs on a single machine subject to periodic maintenance. *Omega*, v. 37, p. 591–599, 2009.

CHEN, W.-J. Minimizing total flow time in the single-machine scheduling problem with periodic maintenance. *Journal of the Operational Research Society*, v. 57, p. 410–415, 2006.

CHEN, W.-J. An efficient algorithm for scheduling jobs on a machine with periodic maintenance. *International Journal of Advanced Manufacturing Technology*, v. 34, n. 11–12, p. 1173–1182, 2007.

CUI, W.-W.; LU, Z. Integrated production scheduling and periodic maintenances on a single machine with release dates. In: *2014 IEEE International Conference on Automation Science and Engineering (CASE)*. Taipei, Taiwan: [s.n.], 2014.

DYER, M. E.; WOLSEY, L. A. Formulating the single machine sequencing problem with release dates as a mixed integer program. *Discrete Applied Mathematics*, v. 26, p. 255–270, 1990.

GRAHAM, R. et al. Optimization and approximation in deterministic sequencing and scheduling: a survey. In: HAMMER, E. J. P.; KORTE, B. (Ed.). *Discrete Optimization II Proceedings of the Advanced Research Institute on Discrete Optimization and Systems Applications of the Systems Science Panel of NATO and of the Discrete Optimization Symposium co-sponsored by IBM Canada and SIAM Banff, Aha. and Vancouver*. [S.l.]: Elsevier, 1979, (Annals of Discrete Mathematics, v. 5). p. 287 – 326.

HSU, C.-J.; LOW, C.; SU, C.-T. A single-machine scheduling problem with maintenance activities to minimize makespan. *Applied Mathematics and Computation*, v. 215, n. 11, p. 3929–3935, 2010.

JI, M.; HE, Y.; CHENG, T. Single-machine scheduling with periodic maintenance to minimize makespan. *Computers & Operations Research*, v. 34, p. 1764–1770, 2007.

KOHL, N. et al. 2-path cuts for the vehicle routing problem with time windows. *Transportation Science*, v. 33, n. 1, p. 101–116, 1999.

LEE, C.-Y.; LIN, C.-S. Single-machine scheduling with maintenance and repair rate-modifying activities. *European Journal of Operational Research*, v. 135, n. 3, p. 493 – 513, 2001.

LEE, J.-Y.; KIM, Y.-D. Minimizing the number of tardy jobs in a single-machine scheduling problem with periodic maintenance. *Computers & Operations Research*, v. 39, p. 2196–2205, 2012.

LI, G. et al. Parallel-machine scheduling with machine-dependent maintenance periodic recycles. *International Journal of Production Economics*, v. 186, p. 1 – 7, 2017.

LIAO, C.; CHEN, W. Single-machine scheduling with periodic maintenance and nonresumable jobs. *Computers & Operations Research*, v. 30, p. 1335–1347, 2003.

LIU, M. et al. An improved exact algorithm for single-machine scheduling to minimise the number of tardy jobs with periodic maintenance. *International Journal of Production Research*, v. 54, n. 12, p. 3591–3602, 2016.

LOW, C.; HSU, C.-J.; SU, C.-T. A modified particle swarm optimization algorithm for a single-machine scheduling problem with periodic maintenance. *Expert Systems with Applications*, v. 37, p. 6429–6434, 2010.

LUO, W.; CHENG, T.; JI, M. Single-machine scheduling with a variable maintenance activity. *Computers & Industrial Engineering*, v. 79, p. 168 – 174, 2015.

MA, Y.; CHU, C.; ZUO, C. A survey of scheduling with deterministic machine availability constraints. *Computers & Industrial Engineering*, v. 58, n. 2, p. 199 – 211, 2010.

NOGUEIRA, T.; CARVALHO, C. de; RAVETTI, M. *Analysis of mixed integer programming formulations for single machine scheduling problems with sequence dependent setup times and release dates*. [S.l.], 2014.

PACHECO, J.; ÁNGEL-BELLO, F.; ÁLVAREZ, A. A multi-start tabu search method for a single-machine scheduling problem with periodic maintenance and sequence-dependent set-up times. *Journal of Scheduling*, v. 16, p. 661–673, 2013.

PEI, J. et al. Minimizing the makespan for a serial-batching scheduling problem with arbitrary machine breakdown and dynamic job arrival. *The International Journal of Advanced Manufacturing Technology*, v. 86, n. 9, p. 3315–3331, 2016.

PESSOA, A. et al. Exact algorithm over an arc-time-indexed formulation for parallel machine scheduling problems. *Mathematical Programming Computation*, Springer-Verlag, v. 2, p. 259–290, 2010.

PINEDO, M. *Scheduling Theory, Algorithms, and Systems*. [S.l.]: Springer, 2016.

SBIHI, M.; VARNIER, C. Single-machine scheduling with periodic and flexible periodic maintenance to minimize maximum tardiness. *Computers & Industrial Engineering*, v. 55, p. 830–840, 2008.

SOURD, F. New exact algorithms for one-machine earliness-tardiness scheduling. *INFORMS Journal on Computing*, v. 21, p. 167–175, 2009.

SUBRAMANIAN, A.; FARIAS, K. Efficient local search limitation strategy for single machine total weighted tardiness scheduling with sequence-dependent setup times. *Computers & Operations Research*, v. 79, p. 190 – 206, 2017.

SWANSON, L. Linking maintenance strategies to performance. *International Journal of Production Economics*, v. 70, n. 3, p. 237 – 244, 2001.

TANAKA, S.; FUJIKUMA, S.; ARAKI, M. An exact algorithm for single-machine scheduling without machine idle time. *Journal of Scheduling*, v. 12, p. 575–593, 2009.

XU, D.; YANG, D.-L. Makespan minimization for two parallel machines scheduling with a periodic availability constraint: Mathematical programming model, average-case analysis, and anomalies. *Applied Mathematical Modelling*, v. 37, n. 14, p. 7561 – 7567, 2013.

XU, D.; YIN, Y.; LI, H. A note on scheduling of nonresumable jobs and flexible maintenance activities on a single machine to minimize makespan. *European Journal of Operational Research*, v. 197, p. 825–827, 2009.

YIN, Y. et al. Approximation schemes for single-machine scheduling with a fixed maintenance activity to minimize the total amount of late work. *Naval Research Logistics*, v. 63, n. 2, p. 172–183, 2016.

YU, X.; ZHANG, Y.; STEINER, G. Single-machine scheduling with periodic maintenance to minimize makespan revisited. *Journal of Scheduling*, v. 17, n. 3, p. 263–270, 2014.

YU, X.; ZHANG, Y.; STEINER, G. Single-machine scheduling with periodic maintenance to minimize makespan revisited. *Journal of Scheduling*, v. 17, n. 3, p. 263–270, 2014.

ZADE, A.; FAKHRZAD, M. A dynamic genetic algorithm for solving a single machine scheduling problem with periodic maintenance. *ISRN Industrial Engineering*, v. 2013, p. 11 pages, 2013. Article ID 936814.

APÊNDICE A

A.1 Resultados detalhados para as instâncias de Pacheco, Ángel-Bello e Álvarez (2013)

A tabela A.1 mostra os resultados agregados para cada formulação implementada com o método iterativo para as instâncias propostas por Pacheco, Ángel-Bello e Álvarez (2013), onde **Tam. árvore** representa a média de nós da árvore do *branch-and-bound*.

Tabela A.1: Resultados agregados para as instâncias propostas por Pacheco, Ángel-Bello e Álvarez (2013)

			CF				ATF				ATF-M				ATF-P			
n	S	α	Gap (%)	Tam. árvore	t (s)	# opt	Gap (%)	Tam. árvore	t (s)	# opt	Gap (%)	Tam. árvore	t (s)	# opt	Gap (%)	Tam. árvore	t (s)	# opt
10	1	2,25	0,00	1646,8	1,6	5	0,00	1,0	0,6	5	0,00	1,2	0,6	5	0,00	1,0	0,0	5
		2,50	0,00	715,6	0,2	5	0,00	1,0	0,8	5	0,00	1,2	0,8	5	0,00	1,0	0,0	5
		3,00	0,00	35,6	0,1	5	0,00	1,0	0,4	5	0,00	1,3	0,4	5	0,00	1,0	0,0	5
		4,00	0,00	5,0	0,0	5	0,00	1,0	0,9	5	0,00	1,4	0,9	5	0,00	1,0	0,2	5
	2	2,25	0,00	3749,4	1,9	5	0,00	1,0	2,3	5	0,00	1,3	2,3	5	0,00	1,0	0,0	5
		2,50	0,00	3439,4	1,1	5	0,00	1,0	1,2	5	0,00	1,2	1,2	5	0,00	1,0	0,0	5
		3,00	0,00	685,0	0,2	5	0,00	4,2	1,4	5	0,00	1,4	1,4	5	0,00	1,0	0,1	5
		4,00	0,00	48,4	0,1	5	0,00	1,0	2,0	5	0,00	1,9	2,0	5	0,00	1,0	0,6	5
	3	2,25	0,00	2434,8	1,4	5	0,00	1,0	3,0	5	0,00	1,5	3,0	5	0,00	1,0	0,0	5
		2,50	0,00	532,4	0,2	5	0,00	1,0	1,8	5	0,00	1,4	1,8	5	0,00	1,0	0,0	5
		3,00	0,00	184,0	0,1	5	0,00	1,0	1,5	5	0,00	1,8	1,5	5	0,00	1,0	0,1	5
		4,00	0,00	12,4	0,1	5	0,00	5,6	3,7	5	0,00	2,7	3,7	5	0,00	1,0	0,5	5
12	1	2,25	0,00	22091,6	9,1	5	0,00	1,0	1,5	5	0,00	1,3	1,5	5	0,00	1,0	0,0	5
		2,50	0,00	9987,2	4,4	5	0,00	1,0	1,0	5	0,00	1,3	1,0	5	0,00	1,0	0,0	5
		3,00	0,00	274,8	0,1	5	0,00	1,0	0,8	5	0,00	1,5	0,8	5	0,00	1,0	0,1	5
		4,00	0,00	27,6	0,1	5	0,00	1,0	1,1	5	0,00	2,0	1,1	5	0,00	1,0	0,3	5
	2	2,25	0,00	6886,4	4,1	5	0,00	1,0	3,5	5	0,00	1,4	3,5	5	0,00	1,0	0,0	5
		2,50	0,00	6128,8	2,7	5	0,00	1,0	3,0	5	0,00	1,5	3,0	5	0,00	1,0	0,0	5
		3,00	0,00	1208,0	0,6	5	0,00	1,0	1,8	5	0,00	1,7	1,8	5	0,00	1,0	0,1	5
		4,00	0,00	110,8	0,1	5	0,00	4,8	2,6	5	0,00	2,6	2,6	5	0,00	1,0	0,5	5
	3	2,25	0,00	9010,4	4,0	5	0,00	1,0	5,9	5	0,00	2,0	5,9	5	0,00	1,0	0,1	5
		2,50	0,00	3977,2	2,4	5	0,00	1,0	3,5	5	0,00	2,0	3,5	5	0,00	1,0	0,1	5
		3,00	0,00	813,0	0,5	5	0,00	38,6	10,9	5	0,00	4,2	10,9	5	0,00	1,4	0,2	5
		4,00	0,00	92,6	0,1	5	0,00	8,2	5,9	5	0,00	4,2	5,9	5	0,00	1,0	1,2	5

Continua na página seguinte

			CF				ATF				ATF-M				ATF-P			
n	S	α	Gap (%)	Tam. árvore	t (s)	# opt	Gap (%)	Tam. árvore	t (s)	# opt	Gap (%)	Tam. árvore	t (s)	# opt	Gap (%)	Tam. árvore	t (s)	# opt
15	1	2,25	0,00	$1{,}4E+5$	91,9	5	0,00	9,4	4,3	5	0,00	1,7	4,3	5	0,00	1,0	0,1	5
		2,50	0,00	4244,8	4,7	5	0,00	1,0	2,5	5	0,00	1,7	2,5	5	0,00	1,0	0,1	5
		3,00	0,00	3373,6	1,8	5	0,00	5,8	3,8	5	0,00	2,1	3,8	5	0,00	1,0	0,2	5
		4,00	0,00	73,2	0,1	5	0,00	3,8	1,9	5	0,00	1,9	1,9	5	0,00	2,2	0,4	5
	2	2,25	0,00	$1{,}8E+5$	88,8	5	0,00	1,0	8,8	5	0,00	2,2	8,8	5	0,00	1,0	0,1	5
		2,50	0,00	52303,2	26,3	5	0,00	1,0	9,3	5	0,00	2,8	9,3	5	0,00	1,0	0,1	5
		3,00	0,00	4748,6	3,1	5	0,00	23,2	11,8	5	0,00	4,2	11,8	5	0,00	1,0	0,2	5
		4,00	0,00	548,6	0,3	5	0,00	28,8	11,7	5	0,00	4,1	11,7	5	0,00	1,0	0,7	5
	3	2,25	0,00	60084,4	3,0	5	0,00	1,0	14,5	5	0,00	3,0	14,5	5	0,00	1,0	0,1	5
		2,50	0,00	5877,2	3,3	5	0,00	1,0	11,4	5	0,00	3,6	11,4	5	0,00	1,0	0,1	5
		3,00	0,00	5130,4	2,9	5	0,00	25,6	19,1	5	0,00	5,3	19,1	5	0,00	1,0	0,4	5
		4,00	0,00	232,0	0,2	5	0,00	45,0	16,7	5	0,00	6,5	16,7	5	0,00	1,0	0,8	5
20	1	2,25	0,00	$1{,}0E+6$	1451,5	5	0,00	1,0	13,8	5	0,00	3,1	13,8	5	0,00	1,0	0,1	5
		2,50	0,00	$8{,}6E+5$	661,8	5	0,00	3,6	10,5	5	0,00	3,2	10,5	5	0,00	1,0	0,2	5
		3,00	0,00	35765,4	27,7	5	0,00	27,2	11,7	5	0,00	4,0	11,7	5	0,00	1,0	0,3	5
		4,00	0,00	690,6	0,9	5	0,00	6,8	8,1	5	0,00	4,6	8,1	5	0,00	4,4	0,6	5
	2	2,25	10,61	$6{,}5E+6$	6736,7	1	0,00	1,0	66,9	5	0,00	7,4	66,9	5	0,00	1,0	0,1	5
		2,50	0,14	$2{,}1E+6$	2915,6	4	0,00	14,2	56,8	5	0,00	9,7	56,8	5	0,00	3,2	0,4	5
		3,00	0,00	$1{,}0E+5$	78,1	5	0,00	57,8	60,3	5	0,00	13,7	60,3	5	0,00	4,0	0,7	5
		4,00	0,00	4656,8	4,7	5	0,00	66,0	48,4	5	0,00	34,5	48,4	5	0,00	15,8	2,1	5
	3	2,25	2,89	$4{,}2E+6$	4926,1	2	0,00	2,8	87,4	5	0,00	9,2	87,4	5	0,00	1,0	0,2	5
		2,50	0,00	$2{,}3E+6$	1654,8	5	0,00	6,4	65,8	5	0,00	11,7	65,8	5	0,00	1,0	0,5	5
		3,00	0,00	$2{,}1E+5$	154,4	5	0,00	140,8	216,1	5	0,00	27,6	216,1	5	0,00	1,0	1,0	5
		4,00	0,00	8258,6	9,1	5	0,00	250,4	270,4	5	0,00	48,5	270,4	5	0,00	48,0	4,3	5
30	1	2,25	7,31	$3{,}3E+6$	7192,8	0	0,00	34,2	105,3	5	0,00	23,8	14,8	5	0,00	3,0	0,8	5
		2,50	4,45	$3{,}0E+6$	6202,4	1	0,00	71,0	121,0	5	0,00	59,4	25,5	5	0,00	15,8	1,0	5
		3,00	0,82	$9{,}3E+5$	1834,7	4	0,00	47,6	101,7	5	0,00	47,6	42,6	5	0,00	12,2	1,1	5
		4,00	0,00	10392,2	30,4	5	0,00	115,0	85,3	5	0,00	48,6	56,1	5	0,00	5,2	1,4	5
	2	2,25	12,43	$3{,}1E+6$	7190,8	0	0,00	28,4	678,3	5	0,00	18,8	34,9	5	0,00	1,0	0,5	5
		2,50	9,95	$3{,}3E+6$	7191,2	0	0,00	55,6	506,5	5	0,00	55,4	54,4	5	0,00	2,8	0,7	5
		3,00	4,09	$3{,}5E+6$	7191,6	0	0,00	220,8	789,5	5	0,00	148,2	199,6	5	0,00	58,2	3,3	5
		4,00	0,00	$1{,}6E+6$	2721,0	5	0,00	150,4	604,1	5	0,00	98,2	272,8	5	0,00	9,4	3,7	5
	3	2,25	9,59	$3{,}4E+6$	7190,9	0	0,00	17,6	1049,3	5	0,00	4,4	48,6	5	0,00	2,4	0,6	5
		2,50	6,50	$3{,}2E+6$	7191,3	0	0,00	1194,2	1494,2	5	0,00	111,0	133,7	5	0,00	12,4	1,4	5
		3,00	2,36	$3{,}4E+6$	7083,1	0	0,00	623,0	1985,7	5	0,00	248,0	379,4	5	0,00	87,4	4,5	5
		4,00	0,00	$4{,}2E+5$	821,0	5	0,00	175,6	963,0	5	0,00	455,6	1199,1	5	0,00	470,4	27,3	5
40	1	2,25	10,10	$2{,}1E+6$	7191,7	0	0,92	341,2	1924,7	4	0,00	16,6	72,4	5	0,00	1,0	0,7	5
		2,50	8,62	$1{,}2E+6$	4612,3	0	0,00	106,4	675,5	5	0,00	503,8	118,3	5	0,00	1,0	0,6	5
		3,00	4,51	$1{,}3E+6$	4134,4	0	1,09	115,4	1975,3	4	0,00	109,4	292,9	5	0,00	2,0	1,6	5
		4,00	0,43	$9{,}5E+5$	2505,4	3	0,00	22,8	299,1	5	0,00	62,8	275,1	5	0,00	1,0	1,9	5
	2	2,25	11,43	$1{,}6E+6$	6303,6	0	0,00	23,2	5374,4	5	0,00	42,2	121,0	5	0,00	1,0	0,7	5
		2,50	10,60	$1{,}1E+6$	4297,7	0	0,00	43,2	5539,8	5	0,00	13,8	140,9	5	0,00	5,0	1,0	5
		3,00	7,66	$8{,}6E+5$	3219,7	0	0,19	28,4	4062,6	4	0,00	67,8	455,5	5	0,00	14,8	3,3	5
		4,00	1,27	$4{,}4E+5$	1780,3	1	0,34	133,4	4323,9	3	0,00	137,8	1266,2	5	0,00	183,6	24,7	5
	3	2,25	12,28	$1{,}4E+6$	5422,9	0	1,45	1,0	7200,0	0	0,00	35,6	267,7	5	0,00	4,4	1,5	5
		2,50	9,96	$1{,}3E+6$	5323,4	0	1,72	25,6	6922,9	1	0,00	272,0	1156,1	5	0,00	251,6	8,3	5
		3,00	4,49	$1{,}2E+6$	5290,9	0	1,39	66,5	7200,0	0	0,00	535,8	3676,5	5	0,00	792,8	27,3	5
		4,00	1,26	$8{,}7E+5$	3483,0	0	1,27	338,3	6652,2	1	0,81	336,0	6448,6	1	0,00	746,6	84,5	5

Continua na página seguinte

			CF				ATF				ATF-M				ATF-P			
n	S	α	Gap (%)	Tam. árvore	t (s)	# opt	Gap (%)	Tam. árvore	t (s)	# opt	Gap (%)	Tam. árvore	t (s)	# opt	Gap (%)	Tam. árvore	t (s)	# opt
50	1	2,25	5,57	$6{,}7E+5$	3664,4	0	0,15	98,4	4162,8	4	0,00	41,0	146,2	5	0,00	1,0	0,7	5
		2,50	9,60	$4{,}8E+5$	3341,2	0	0,00	22,6	3759,6	5	0,00	45,4	320,8	5	0,00	1,0	1,1	5
		3,00	4,27	$6{,}5E+5$	3237,3	0	0,00	128,0	3555,9	5	0,00	164,2	1080,1	5	0,00	1,0	2,1	5
		4,00	0,53	$9{,}2E+5$	3655,9	2	0,00	20,0	1313,5	5	0,00	150,8	1371,1	5	0,00	1,0	5,0	5
	2	2,25	20,39	$5{,}8E+5$	4965,8	0	0,00	52,5	5154,7	4	0,00	24,2	442,9	5	0,00	1,0	5,0	5
		2,50	13,68	$4{,}7E+5$	3302,3	0	0,00	31,6	4073,7	5	0,00	62,6	711,6	5	0,00	1,0	6,9	5
		3,00	7,12	$7{,}1E+5$	4320,3	0	0,00	64,2	1767,1	5	0,00	73,6	912,9	5	0,00	14,6	13,5	5
		4,00	2,04	$7{,}8E+5$	3943,1	1	0,00	60,4	1783,8	5	0,00	160,2	2181,0	5	0,00	9,8	26,8	5
	3	2,25	12,69	$6{,}9E+5$	5459,2	0	-	1,0	7200,0	0	0,00	103,4	1748,1	5	0,00	12,0	1,9	5
		2,50	9,00	$4{,}8E+5$	3566,8	0	-	1,0	7200,0	0	0,00	320,4	3199,9	4	0,00	49,0	3,3	5
		3,00	5,00	$3{,}9E+5$	2788,5	0	-	1,0	7200,0	0	1,29	188,2	7200,0	0	0,00	812,8	29,3	5
		4,00	3,22	$4{,}0E+5$	2901,7	0	-	1,0	7200,0	0	0,18	50,8	7200,0	0	0,00	718,6	263,7	5

A tabela A.2 mostra os resultados agregados para o algoritmo iterativo aplicado à formulação ATF-P para as instâncias de Pacheco, Ángel-Bello e Álvarez (2013). **#it** corresponde à média do número de iterações, $\boldsymbol{t'(s)}$ indica o tempo médio (em segundos) despendido nas iterações inviáveis do algoritmo iterativo, e $\boldsymbol{t_{Last}(s)}$ denota o tempo médio (em segundos) gasto pela última iteração.

A.2 Resultados detalhados para as instâncias propostas neste trabalho

Tabela A.3: Resultados agregados para as instâncias propostas neste trabalho

			CF				ATF				ATF-M				ATF-P			
n	S	α	Gap (%)	Tam. árvore	t (s)	# opt	Gap (%)	Tam. árvore	t (s)	# opt	Gap (%)	Tam. árvore	t (s)	# opt	Gap (%)	Tam. árvore	t (s)	# opt
10	1	2	0,00	1434,8	0,5	5	0,00	1,0	1,7	5	0,00	1,0	0,1	5	0,00	1,0	0,0	5
		3	0,00	591,4	0,8	5	0,00	3,0	1,0	5	0,00	1,0	0,4	5	0,00	1,0	0,1	5
		4	0,00	66,6	0,0	5	0,00	1,0	0,8	5	0,00	1,0	0,3	5	0,00	1,4	0,1	5
	2	2	0,00	1,0	0,1	5	0,00	1,0	8,0	5	0,00	1,0	0,3	5	0,00	1,0	0,1	5
		3	0,00	2317,0	0,8	5	0,00	1,0	3,1	5	0,00	1,0	1,8	5	0,00	1,0	0,1	5
		4	0,00	3700,4	0,8	5	0,00	22,0	9,1	5	0,00	1,0	1,3	5	0,00	1,0	0,3	5
	3	2	0,00	465,0	1,1	5	0,00	1,0	27,9	5	0,00	1,0	0,7	5	0,00	1,0	0,1	5
		3	0,00	2309,2	0,6	5	0,00	1,0	18,6	5	0,00	1,0	5,5	5	0,00	1,0	0,2	5
		4	0,00	18671,8	4,7	5	0,00	19,2	31,6	5	0,00	1,8	6,3	5	0,00	1,0	0,7	5

Continua na página seguinte

			CF				ATF				ATF-M				ATF-P			
n	S	α	Gap (%)	Tam. árvore	t (s)	# opt	Gap (%)	Tam. árvore	t (s)	# opt	Gap (%)	Tam. árvore	t (s)	# opt	Gap (%)	Tam. árvore	t (s)	# opt
		2	0,00	$1{,}5E+5$	68,4	5	0,00	1,0	14,5	5	0,00	1,0	0,6	5	0,00	1,0	0,1	5
	1	3	0,18	$3{,}5E+6$	1462,4	4	0,00	9,2	6,4	5	0,00	1,0	1,6	5	0,00	1,0	0,1	5
		4	0,00	626,8	0,5	5	0,00	10,4	11,6	5	0,00	9,6	2,9	5	0,00	1,0	0,3	5
		2	0,00	285,6	4,9	5	0,00	1,0	91,0	5	0,00	1,0	1,6	5	0,00	1,0	0,1	5
15	2	3	0,00	$4{,}9E+5$	180,1	5	0,00	7,4	22,7	5	0,00	1,0	5,3	5	0,00	1,0	0,2	5
		4	0,00	$2{,}1E+5$	68,3	5	0,00	26,6	107,1	5	0,00	15,8	13,6	5	0,00	1,0	1,1	5
		2	0,00	8544,6	23,3	5	0,00	1,0	604,9	5	0,00	1,0	3,2	5	0,00	1,0	0,3	5
	3	3	1,71	$9{,}9E+6$	7101,9	1	0,00	1,6	183,7	5	0,00	1,0	13,6	5	0,00	1,0	0,6	5
		4	0,95	$1{,}4E+6$	1643,9	4	0,00	34,6	590,0	5	0,00	22,4	55,6	5	0,00	1,0	2,5	5
		2	0,00	$1{,}6E+5$	1396,3	5	0,00	1,0	87,7	5	0,00	1,0	1,8	5	0,00	1,0	0,1	5
	1	3	0,44	$7{,}6E+6$	4931,0	3	0,00	12,4	51,1	5	0,00	6,0	6,8	5	0,00	1,0	0,4	5
		4	0,00	$1{,}5E+6$	901,3	5	0,00	17,0	57,5	5	0,00	22,6	15,7	5	0,00	1,0	1,0	5
		2	0,00	2340,8	51,5	5	0,00	1,0	1216,9	5	0,00	1,0	4,6	5	0,00	1,0	0,3	5
20	2	3	4,70	$1{,}0E+7$	7184,1	0	0,00	1,0	365,9	5	0,00	1,0	21,2	5	0,00	1,0	0,6	5
		4	0,81	$8{,}4E+6$	4991,6	2	0,00	35,0	701,1	5	0,00	140,6	121,7	5	0,00	2,8	3,5	5
		2	0,00	12695,4	616,1	5	0,00	1,0	3791,7	5	0,00	1,0	9,9	5	0,00	1,0	0,6	5
	3	3	7,48	$8{,}9E+6$	7187,0	0	0,00	6,0	2237,5	5	0,00	1,0	127,7	5	0,00	1,0	1,7	5
		4	0,50	$6{,}7E+6$	4052,9	3	0,00	35,6	2177,6	5	0,00	57,0	279,6	5	0,00	2,4	9,8	5
		2	10,30	$3{,}8E+6$	6424,1	1	0,00	1,0	437,3	5	0,00	1,0	4,5	5	0,00	1,0	0,2	5
	1	3	1,78	$4{,}8E+6$	6137,8	1	0,00	29,8	203,0	5	0,00	1,0	12,7	5	0,00	1,0	0,3	5
		4	0,00	$7{,}6E+5$	692,6	5	0,00	13,0	168,5	5	0,00	26,0	29,1	5	0,00	1,0	1,3	5
		2	1,26	$1{,}6E+5$	1634,6	4	0,00	1,0	4806,6	3	0,00	1,0	10,9	5	0,00	1,0	0,5	5
25	2	3	7,77	$5{,}3E+6$	7188,0	0	0,00	2,6	2175,3	5	0,00	1,0	54,7	5	0,00	1,0	1,1	5
		4	1,23	$6{,}5E+6$	7187,5	0	0,00	17,6	2292,8	5	0,00	65,4	244,0	5	0,00	65,2	7,4	5
		2	7,06	$2{,}8E+6$	5460,5	2	-	1,0	7198,7	0	0,00	1,0	19,6	5	0,00	1,0	0,9	5
	3	3	4,90	$5{,}5E+6$	7188,1	0	0,73	1,0	6943,9	1	0,00	316,4	403,8	5	0,00	1,0	4,7	5
		4	2,10	$4{,}5E+6$	6503,6	0	0,53	3,0	6604,8	1	0,00	143,2	1364,4	5	0,00	100,0	34,9	5
		2	9,36	$1{,}1E+6$	6510,7	0	-	-	-	-	0,00	1,0	68,8	5	0,00	1,0	1,1	5
	1	3	1,57	$1{,}1E+6$	4814,4	0	-	-	-	-	0,00	56,2	471,4	5	0,00	1,0	3,2	5
		4	0,00	$1{,}1E+6$	4092,1	0	-	-	-	-	0,00	60,6	1914,2	5	0,00	5,4	11,4	5
		2	28,54	$7{,}6E+5$	6892,6	0	-	-	-	-	0,00	1,0	188,1	5	0,00	1,0	3,6	5
50	2	3	5,90	$8{,}7E+5$	4574,2	0	-	-	-	-	0,00	12,8	2028,0	5	0,00	5,4	7,9	5
		4	0,27	$5{,}4E+5$	2715,9	0	-	-	-	-	0,10	29,0	6524,6	2	0,00	133,8	37,6	5
		2	25,98	$5{,}9E+5$	4306,4	0	-	-	-	-	0,00	1,0	411,4	5	0,00	1,0	6,8	5
	3	3	8,23	$6{,}5E+5$	4366,4	0	-	-	-	-	7,08	1,0	7200,0	0	0,00	594,4	49,0	5
		4	0,76	$3{,}9E+5$	2517,1	0	-	-	-	-	12,96	1,0	7200,0	0	0,00	3125,2	1586,0	5
		2	10,47	$2{,}5E+5$	5078,8	0	-	-	-	-	0,00	1,0	990,3	5	0,00	1,0	3,7	5
	1	3	0,89	$2{,}6E+5$	5953,5	0	-	-	-	-	0,00	206,6	5844,6	4	0,06	$4{,}0E+5$	1451,2	4
		4	0,00	$2{,}8E+5$	3927,2	0	-	-	-	-	4,16	1,0	7200,0	0	0,00	60,2	56,8	5
		2	22,99	$2{,}2E+5$	6283,7	0	-	-	-	-	0,00	1,0	863,5	5	0,00	1,0	10,4	5
75	2	3	4,36	$3{,}1E+5$	6487,8	0	-	-	-	-	11,86	1,0	7200,0	0	0,00	1,0	30,1	5
		4	0,71	$2{,}0E+5$	3283,9	0	-	-	-	-	11,07	1,0	7200,0	0	0,00	3977,6	863,7	5
		2	24,52	$2{,}0E+5$	6793,9	0	-	-	-	-	0,00	1,0	2113,8	5	0,00	1,0	22,4	5
	3	3	6,42	$2{,}1E+5$	4850,1	0	-	-	-	-	19,28	1,0	7200,0	0	0,00	778,4	357,3	5
		4	0,67	$2{,}2E+5$	4439,8	0	-	-	-	-	13,55	1,0	7200,0	0	0,36	3236,6	6913,3	1

Continua na página seguinte

			CF				ATF				ATF-M				ATF-P			
n	S	α	Gap (%)	Tam. árvore	t (s)	# opt	Gap (%)	Tam. árvore	t (s)	# opt	Gap (%)	Tam. árvore	t (s)	# opt	Gap (%)	Tam. árvore	t (s)	# opt
		2	11,73	$1{,}4E+5$	7190,2	0	-	-	-	-	1,18	3,4	2474,4	4	0,00	1,0	8,3	5
	1	3	0,56	$1{,}1E+5$	6627,1	0	-	-	-	-	7,35	1,0	7200,0	0	0,00	63,0	38,0	5
		4	0,00	$1{,}3E+5$	5734,1	0	-	-	-	-	9,33	1,0	7200,0	0	0,00	90,6	178,8	5
		2	24,89	$1{,}1E+5$	6698,5	0	-	-	-	-	0,00	1,0	3329,2	3	0,00	1,0	23,5	5
100	2	3	5,11	$1{,}1E+5$	5840,0	0	-	-	-	-	18,16	1,0	7200,0	0	0,00	5,8	79,6	4
		4	0,27	$1{,}1E+5$	6426,2	0	-	-	-	-	10,01	1,0	7200,0	0	0,00	970,0	1120,5	5
		2	22,92	$1{,}1E+5$	7058,3	0	-	-	-	-	0,00	1,0	6175,2	3	0,00	1,0	50,9	5
	3	3	6,93	$1{,}0E+5$	5802,9	0	-	-	-	-	18,61	1,0	7200,0	0	0,02	2350,0	1756,4	5
		4	0,67	$1{,}1E+5$	4930,1	0	-	-	-	-	12,27	1,0	7200,0	0	0,54	249,8	7200,0	0
		2	8,84	66593,0	7189,6	0	-	-	-	-	0,00	1,0	6315,6	2	0,00	1,0	17,9	5
	1	3	0,53	51849,6	7190,5	0	-	-	-	-	11,96	1,0	7200,0	0	0,00	56,0	108,7	5
		4	0,03	69688,6	6758,5	0	-	-	-	-	8,55	1,0	7200,0	0	0,03	5883,0	1909,6	4
		2	25,53	63251,2	7189,4	0	-	-	-	-	0,00	1,0	5774,4	1	0,00	1,0	50,3	5
125	2	3	5,57	62828,6	6587,0	0	-	-	-	-	17,59	1,0	7200,0	0	0,04	4099,0	1555,1	4
		4	0,19	53401,2	6318,6	0	-	-	-	-	10,23	1,0	7200,0	0	0,00	169,6	1857,4	5
		2	24,46	44825,8	7191,0	0	-	-	-	-	-	-	-	0	0,00	657,4	2621,5	5
	3	3	5,19	63752,0	6935,2	0	-	-	-	-	-	-	-	0	0,00	1,0	99,2	5
		4	7,56	46238,8	6971,1	0	-	-	-	-	-	-	-	0	7,49	67,6	7200,0	0

Tabela A.2: Resultados agregados do algoritmo iterativo aplicado à ATF-P para as instâncias propostas por Pacheco, Ángel-Bello e Álvarez (2013)

S	P	$n = 10$			$n = 12$			$n = 15$			$n = 20$		
		#it	t' (s)	t_{Last} (s)	#it	t' (s)	t_{Last} (s)	#it	t' (s)	t_{Last} (s)	#it	t' (s)	t_{Last} (s)
1	$2,25d_m$	2,6	0,0	0,0	2,6	0,0	0,0	3,0	0,0	0,0	3,0	0,1	0,0
	$2,50d_m$	2,2	0,0	0,0	2,2	0,0	0,0	2,0	0,0	0,0	2,4	0,1	0,1
	$3,00d_m$	1,2	0,0	0,0	1,6	0,0	0,1	1,8	0,1	0,1	1,8	0,1	0,2
	$4,00d_m$	1,8	0,1	0,1	1,2	0,0	0,3	1,0	0,0	0,4	1,2	0,1	0,6
2	$2,25d_m$	3,2	0,0	0,0	2,6	0,0	0,0	2,6	0,0	0,0	4,0	0,1	0,0
	$2,50d_m$	2,0	0,0	0,0	2,2	0,0	0,0	2,4	0,1	0,1	3,0	0,1	0,2
	$3,00d_m$	1,6	0,0	0,0	2,0	0,1	0,1	1,6	0,1	0,2	2,2	0,2	0,5
	$4,00d_m$	1,6	0,1	0,5	1,0	0,0	0,5	1,4	0,1	0,6	1,4	0,2	1,9
3	$2,25d_m$	2,8	0,0	0,0	3,0	0,0	0,0	3,0	0,1	0,0	3,4	0,1	0,0
	$2,50d_m$	2,0	0,0	0,0	2,0	0,0	0,0	2,2	0,1	0,1	3,0	0,2	0,2
	$3,00d_m$	1,6	0,0	0,1	2,0	0,1	0,1	2,2	0,2	0,3	2,0	0,3	0,7
	$4,00d_m$	2,0	0,3	0,2	1,8	0,4	0,8	1,2	0,1	0,7	2,0	0,9	3,4

S	P	$n = 30$			$n = 40$			$n = 50$		
		#it	t' (s)	t_{Last} (s)	#it	t' (s)	t_{Last} (s)	#it	t' (s)	t_{Last} (s)
1	$2,25d_m$	3,0	0,3	0,5	4,2	0,5	0,2	4,2	0,5	0,3
	$2,50d_m$	2,2	0,2	0,8	3,2	0,3	0,3	3,8	0,7	0,4
	$3,00d_m$	1,4	0,2	0,9	2,0	0,8	0,7	2,6	0,9	1,1
	$4,00d_m$	1,0	0,0	1,4	1,2	0,2	1,7	1,2	0,3	4,6
2	$2,25d_m$	4,4	0,3	0,2	4,8	0,5	0,2	10,0	4,4	0,6
	$2,50d_m$	3,2	0,3	0,4	3,8	0,6	0,3	7,8	5,9	1,0
	$3,00d_m$	2,0	0,5	2,8	3,0	1,3	1,9	5,4	8,5	5,0
	$4,00d_m$	1,0	0,1	3,6	1,4	1,1	23,6	3,8	10,9	15,8
3	$2,25d_m$	4,4	0,4	0,2	5,2	1,0	0,5	6,4	1,3	0,6
	$2,50d_m$	3,2	0,6	0,8	3,8	1,3	7,1	4,2	1,6	1,7
	$3,00d_m$	2,2	0,9	3,6	2,4	1,9	25,4	3,0	12,6	16,7
	$4,00d_m$	1,8	1,7	25,6	1,8	3,1	81,4	1,6	3,4	260,3

Tabela A.4: Resultados agregados do algoritmo iterativo aplicado à formulação ATF-P para as instâncias propostas neste trabalho

S	α	$n = 10$			$n = 15$			$n = 20$			$n = 25$		
		#it	t' (s)	t_{Last} (s)	#it	t' (s)	t_{Last} (s)	#it	t' (s)	t_{Last} (s)	#it	t' (s)	t_{Last} (s)
1	2	2,8	0,0	0,0	4,2	0,1	0,0	5,0	0,1	0,0	7,0	0,2	0,0
	3	1,2	0,0	0,1	1,2	0,0	0,1	1,8	0,1	0,3	1,6	0,1	0,2
	4	1,0	0,0	0,1	1,2	0,0	0,3	1,2	0,1	0,9	1,0	0,0	1,2
2	2	5,2	0,1	0,0	7,8	0,1	0,0	8,4	0,3	0,0	11,0	0,5	0,0
	3	2,0	0,1	0,0	1,0	0,0	0,1	2,0	0,2	0,4	2,8	0,5	0,5
	4	1,6	0,1	0,2	1,8	0,3	0,8	2,0	0,9	2,6	1,8	1,1	6,2
3	2	5,2	0,1	0,0	6,6	0,2	0,0	8,2	0,5	0,0	7,8	0,8	0,1
	3	2,0	0,1	0,1	2,0	0,3	0,3	2,0	0,6	1,2	2,2	1,4	3,3
	4	1,6	0,2	0,5	1,8	0,9	1,6	1,6	1,5	8,3	1,4	2,2	32,7

S	α	$n = 50$			$n = 75$			$n = 100$			$n = 125$		
		#it	t' (s)	t_{Last} (s)	#it	t' (s)	t_{Last} (s)	#it	t' (s)	t_{Last} (s)	#it	t' (s)	t_{Last} (s)
1	2	8,8	1,1	0,1	13,0	3,5	0,2	18,2	7,9	0,4	16,4	16,9	0,9
	3	2,2	1,3	1,9	2,8	5,0	1446,3	3,4	12,8	25,2	9,8	49,8	0,5
	4	2,0	2,9	8,5	2,0	9,0	47,8	2,0	23,8	155,0	21,4	98,1	1,1
2	2	20,4	3,6	0,1	22,4	10,2	0,2	32,6	23,2	0,3	41,0	75,7	33,1
	3	3,4	4,1	3,8	4,2	17,7	12,4	6,0	50,3	29,2	11,2	1507,4	47,6
	4	1,8	8,4	29,2	2,2	52,0	811,7	2,4	149,6	970,9	49,0	2390,9	230,7
3	2	17,2	6,6	0,2	24,8	22,0	0,4	30,6	50,0	0,9	37,4	1734,2	175,4
	3	3,8	14,0	35,0	4,8	56,0	301,3	6,2	206,4	1550,0	11,2	1241,1	616,3
	4	1,6	25,4	1560,6	2,4	239,5	6673,8	3,0	1077,9	6122,1	21,4	4612,8	2586,2