RAFAEL DE FREITAS LEMOS

HEURÍSTICAS PARA O PROBLEMA ESTOCÁSTICO DE
PROGRAMAÇÃO DE MÁQUINA ÚNICA COM MINIMIZAÇÃO DE
EARLINESS E TARDINESS

São Paulo
2014

RAFAEL DE FREITAS LEMOS

HEURÍSTICAS PARA O PROBLEMA ESTOCÁSTICO DE
PROGRAMAÇÃO DE MÁQUINA ÚNICA COM MINIMIZAÇÃO DE
EARLINESS E TARDINESS

Dissertação apresentada à Escola Politécnica
da Universidade de São Paulo para obtenção
do Título de Mestre em Engenharia de
Produção.

São Paulo
2014

RAFAEL DE FREITAS LEMOS

HEURÍSTICAS PARA O PROBLEMA ESTOCÁSTICO DE
PROGRAMAÇÃO DE MÁQUINA ÚNICA COM MINIMIZAÇÃO DE
EARLINESS E TARDINESS

Dissertação apresentada à Escola Politécnica
da Universidade de São Paulo para obtenção
do Título de Mestre em Engenharia de
Produção.

Área de Concentração:
Engenharia de Produção

Orientadora:
Profa. Dra. Débora P. Ronconi

São Paulo
2014

FICHA CATALOGRÁFICA

Lemos, Rafael de Freitas

Heurística para o problema estocástico de programação de máquina única com minimização de earliness e tardiness / R.F. Lemos. -- São Paulo, 2014.

91 p.

Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Produção.

1.Programação estocástica 2.Heurística 3.Determinação de datas 4.Máquina única 5.Earliness/tardiness I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Produção II.t.

DEDICATÓRIA

Aos amigos, por todo o suporte dispendido.

E aos inimigos, por todos os obstáculos que me fortaleceram.

AGRADECIMENTOS

Aos meus pais, Márcia e Rinaldo, por seu privilegiado *pool* genético. E por todo amor incondicional.

À Emileane, o amor da minha vida, a quem jamais serei capaz de retribuir à altura os sinceros sorrisos, os calorosos ósculos e as eventuais doces lágrimas.

À Professora Débora, por sua orientação, e pela oportunidade de resgatar dos confins da memória integrais duplas e o teorema do sanduíche.

A todos aqueles cujos nomes não citei. Não por serem menos importantes, mas unicamente pela pequenez das margens, as quais não permitem descrever a contento o maravilhoso apoio de tantos.

*"A humanidade se divide em dois grupos,
os que concordam comigo e os equivocados."*

Ariano Suassuna

RESUMO

O presente trabalho aborda o problema de determinação de datas de entrega e o sequenciamento de tarefas com tempos de processamento estocásticos. O ambiente considerado constitui em uma máquina simples e tarefas com custos individuais e distintos de adiantamento e atraso de entrega (*earliness* e *tardiness* ou simplesmente E/T). O objetivo é determinar a sequência e as datas de entrega ótimas que simultaneamente minimizam o custo total esperado de E/T. Para a determinação de sequências candidatas, são apresentadas diversas heurísticas construtivas com tempo de execução polinomial baseadas em um método de inserção de tarefas. Considerando tarefas com distribuição normal, experimentos computacionais comprovam a eficácia dos algoritmos para problemas de menor porte, os quais fornecem soluções ótimas em 99,85% dos casos avaliados. Quando aplicadas a um conjunto com uma maior quantidade de tarefas, as heurísticas apresentaram resultados melhores do que o algoritmo disponível na literatura em mais de 80% dos casos. Consideradas tarefas com distribuição lognormal, obteve-se um percentual de otimalidade entre 93,87% e 96,45%, a depender da heurística aplicada. Demonstra-se ainda para o caso com distribuição normal que os métodos propostos são assintoticamente ótimos e, portanto, são indicados para a resolução de problemas de grande porte.

Palavras-chave: *earliness/tardiness*, programação estocástica de tarefas, determinação de datas, máquina única, heurística.

ABSTRACT

This work addresses the problem of concurrent due-date assignment and sequencing of a set of jobs on a stochastic single-machine environment with distinct job earliness and tardiness penalty costs. It is assumed that the jobs processing times are statistically independent and follow a normal distribution whose mean and variance are provided and they are not necessarily integer values. The objective is to determine the job sequence and the integer due dates which minimize the expected total earliness and tardiness costs. Several efficient insertion-based construction heuristics are proposed in order to find candidates for the optimal sequence with polynomial time complexity. For the normal distribution problem, numerical experiments show that the proposed heuristic methods are able to find the optimal solution in 99,85% when applied to problems with a smaller size. When applied to problems with a bigger size, the solutions found by the proposed heuristics had better costs than the solutions described in the literature in more than 80% of cases. For the lognormal distribution problem, the proposed heuristic methods provided solutions with a percentage of optimality between 93,87% and 96,45%. Furthermore, for the normal distribution case, it was proven that the heuristics are asymptotically optimal, i.e., it can be used for problems of any size.

Keywords: stochastic scheduling; single machine; earliness/tardiness; due-date assignment; heuristic.

SUMÁRIO

1	INTRODUÇÃO.....	14
2	CARACTERIZAÇÃO DO PROBLEMA.....	16
2.1	O PROBLEMA ESTOCÁSTICO DE E/T PARA MÁQUINA SIMPLES	16
2.2	O PROBLEMA ESTOCÁSTICO DE E/T COM DETERMINAÇÃO DE DATAS DE ENTREGA	17
3	REVISÃO BIBLIOGRÁFICA E PROPOSTA DE PESQUISA	18
3.1	PROBLEMA DETERMINÍSTICO.....	18
3.2	PROBLEMA ESTOCÁSTICO.....	20
3.3	PROPOSTA DA PESQUISA	22
4	FORMULAÇÃO DO PROBLEMA COM DISTRIBUIÇÃO NORMAL E DESENVOLVIMENTO MATEMÁTICO.....	23
4.1	FORMULAÇÃO MATEMÁTICA	23
4.2	DATAS DE ENTREGA ÓTIMAS.....	29
4.3	SEQUÊNCIA ÓTIMA	33
4.4	REGRA SOR	34
5	HEURÍSTICAS PARA O PROBLEMA COM DISTRIBUIÇÃO NORMAL	42
5.1	HEURÍSTICA NEH	42
5.2	HEURÍSTICAS NEHL	43
6	EXPERIMENTOS COMPUTACIONAIS PARA O PROBLEMA COM DISTRIBUIÇÃO NORMAL.....	49
6.1	DESCRIÇÃO DO EXPERIMENTO	49
6.2	RESULTADOS DAS HEURÍSTICAS NEHL.....	51
7	OTIMALIDADE ASSINTÓTICA PARA O PROBLEMA COM DISTRIBUIÇÃO NORMAL	55
7.1	OTIMALIDADE ASSINTÓTICA DA SEQUÊNCIA $\tilde{s}$	55
7.2	OTIMALIDADE ASSINTÓTICA DAS HEURÍSTICAS NEHL	62
8	FORMULAÇÃO DO PROBLEMA COM DISTRIBUIÇÃO LOGNORMAL E DESENVOLVIMENTO MATEMÁTICO.....	64
8.1	FORMULAÇÃO MATEMÁTICA	64
8.2	DATAS DE ENTREGA ÓTIMAS.....	70
8.3	SEQUÊNCIA ÓTIMA	73
9	HEURÍSTICAS PARA O PROBLEMA COM DISTRIBUIÇÃO LOGNORMAL	75
9.1	HEURÍSTICAS NEHL-LOG.....	75

10	EXPERIMENTOS COMPUTACIONAIS PARA O PROBLEMA COM DISTRIBUIÇÃO LOGNORMAL	80
10.1	DESCRIÇÃO DOS EXPERIMENTOS	80
10.2	RESULTADOS DO MÉTODO DE FENTON-WILKINSON	82
10.3	RESULTADOS DAS HEURÍSTICAS NEHL-LOG	83
11	CONCLUSÕES.....	87
	REFERÊNCIAS.....	88

LISTA DE TABELAS

Tabela 1 – Parâmetros para o exemplo ilustrativo	38
Tabela 2 – Dados auxiliares para solução do exemplo ilustrativo	38
Tabela 3 – Resultados numéricos para a sequência $\hat{s}$	39
Tabela 4 – Resultados numéricos para a sequência $\tilde{s}$	39
Tabela 5 – Aplicação da heurística NEHL-1 para o exemplo ilustrativo da Tabela 1	47
Tabela 6 – Comparação entre os algoritmos NEHL e SOR para o Conjunto 1	52
Tabela 7 – Comparação entre os algoritmos NEHL e SOR para o Conjunto 2	52
Tabela 8 – Comparação entre os algoritmos NEHL para o Conjunto 2	53
Tabela 9 – Comparação entre os algoritmos NEHL-log para o Conjunto 1	84
Tabela 10 – Comparação entre os algoritmos NEHL-log para o Conjunto 2	84
Tabela 11 – Comparação entre os algoritmos NEHL-log e NEHL-log2 para o Conjunto 2	86

LISTA DE FIGURAS

Figura 1 – Representação gráfica do atraso e adiantamento das tarefas.....	24
Figura 2 – Gráfico de Gantt preditivo para o exemplo ilustrativo	40
Figura 3 – Gráfico de Gantt preditivo para a tarefa 2	40
Figura 4 – Gráfico de Gantt preditivo para a tarefa 7	41
Figura 5 – Gráfico de Gantt preditivo para a solução obtida pela heurística NEHL-1	47
Figura 6 – Representação gráfica do termo entre colchetes da equação (8.23).....	75
Figura 7 – Densidades de probabilidade para a soma de duas variáveis lognormais	82
Figura 8 – Densidades acumuladas para a soma de duas variáveis lognormais.....	83

1 INTRODUÇÃO

O sucesso alcançado pela filosofia de produção *Just in Time* (JIT) colocou em destaque a necessidade do cumprimento de prazos de entrega. Determinados setores produtivos (e.g. petróleo e gás), por estarem inseridos em projetos com enormes montantes financeiros envolvidos, percebem a importância estratégica do planejamento de datas de entrega. Tais datas geralmente são resultado de um processo de negociação com os clientes e não são conhecidas com antecedência. Dado este cenário, a determinação de datas de entrega consiste uma função essencial no processo de planejamento da produção e programação de tarefas neste tipo de ambiente.

Uma das medidas mais utilizadas para avaliação do desempenho do chão de fábrica são os custos decorrentes de adiantamento e atraso da entrega (respectivamente *earliness* e *tardiness* em inglês). A experiência mostra que os tempos de processamento das tarefas não são completamente conhecidos, mas podem ser definidos como variáveis aleatórias, cuja distribuição de probabilidade é conhecida. Todavia quando da definição das datas de entrega é usual considerar valores determinísticos.

Este trabalho tem por objetivo o desenvolvimento de uma heurística construtiva para a resolução do problema de determinação de datas de entrega em uma máquina única com tarefas sujeitas a custos de *earliness* e *tardiness* (E/T) e com tempos de processamento estocásticos. Este é um problema com alto grau de dificuldade, não sendo conhecidos métodos exatos de resolução com tempos de execução razoáveis (i.e., polinomiais). Desta forma justifica-se a investigação de métodos alternativos de resolução capazes de fornecer soluções de boa qualidade em tempos razoáveis de execução.

São destacados como objetivos: *i)* apresentar um levantamento bibliográfico do problema de *Earliness* e *Tardiness*, desde suas origens no caso determinístico até a posterior extensão para o domínio estocástico; *ii)* apresentar o problema de determinação de datas de entrega para o problema estocástico sujeito a custos de atraso e adiantamento; *iii)* considerada a modelagem com distribuição normal, propor heurísticas construtivas capazes de fornecer soluções melhores do que os métodos disponíveis na literatura; *iv)* ampliar os resultados para um problema ainda inédito na literatura, no qual a duração das tarefas segue uma distribuição lognormal.

Para tanto, o trabalho é estruturado como segue. No Capítulo 2 é apresentada a caracterização do problema. O Capítulo 3 contém a revisão bibliográfica pertinente e resume a justificativa e objetivo do presente trabalho. O problema de tempos de processamento com distribuição normal é contemplado nos seguintes capítulos: Capítulo 4, no qual se apresenta a

formulação matemática, assim como desenvolvimentos teóricos; Capítulo 5, no qual são propostas duas heurísticas construtivas para a resolução do problema; Capítulo 6, em que se apresenta a comparação entre as heurísticas propostas e o algoritmo disponível na literatura; Capítulo 7, dedicado à demonstração da otimalidade assintótica dos algoritmos apresentados. Já o problema de tempos de processamento com distribuição lognormal é tratado nos capítulos: Capítulo 8, no qual são apresentados os desenvolvimentos matemáticos do problema; Capítulo 9, em que são apresentadas as heurísticas para obtenção de soluções candidatas; Capítulo 10, no qual se expõem os resultados computacionais obtidos da aplicação das heurística propostas. Finalmente o Capítulo 11 dedica-se às conclusões obtidas com o desenvolvimento deste trabalho.

2 CARACTERIZAÇÃO DO PROBLEMA

Este capítulo se dedica à apresentação do problema em estudo. Inicialmente é descrito o problema de sequenciamento de tarefas com tempos de processamento estocásticos e datas de entrega conhecidas em uma máquina simples, tarefas estas sujeitas a custos de atraso e adiantamento individuais. A seguir o problema é expandido para um cenário de determinação de datas de entrega.

2.1 O Problema Estocástico de E/T para Máquina Simples

Seja o problema determinístico de atraso e adiantamento definido pela trinca $1||\sum(\alpha_j E_j + \beta_j T_j)$, conforme notação descrita por Pinedo (2002). O objeto de estudo consiste na contraparte estocástica deste problema e pode ser definido como segue. Considere um conjunto de n tarefas a serem processadas uma única vez. As tarefas são disponibilizadas simultaneamente em um instante inicial conhecido para processamento em uma máquina, a qual é capaz de executar apenas uma tarefa por vez. Os tempos de configuração (*setup*) da máquina para cada tarefa são independentes de sua sequência e estão inclusos em seus tempos de processamento. Tais tempos não são completamente definidos, mas caracterizados como variáveis aleatórias independentes e identicamente distribuídas com distribuição de probabilidade conhecida. Esta premissa é válida nos casos em que o programador da produção não tem acesso aos tempos de processamento com acurácia; há erros envolvidos nos processos de medição dos tempos de processamento; o operador ou a máquina estão sujeitos a oscilações aleatórias ou ainda quando os tempos de *setup* da máquina variam de forma aleatória. Esta aleatoriedade pode ser ainda uma representação do risco de falha inerente a qualquer empreendimento que envolva o processamento de tarefas por um conjunto de recursos definido. Considere ainda que a máquina está continuamente disponível e assim que uma tarefa comece a ser processada não são permitidas interrupções (i.e. sem *preemption*).

O problema de otimização estará completo ao considerar que para cada uma das tarefas encontra-se definida uma data de entrega e que esse conjunto de operações deverá ser executado de modo a minimizar os custos médios totais de atraso e adiantamento, os quais são individuais de cada tarefa. O custo de atraso é definido como a diferença absoluta entre o instante de término da tarefa e sua data de entrega previamente acordada, ponderada por determinado fator individual da tarefa (e.g. multas diárias por atraso, custos adicionais para entrega etc.). Já o custo de adiantamento é a contraparte correspondente quando o término da tarefa ocorre em momento anterior à data de entrega acordada. Ainda que esta constitua uma medida majoritariamente interna, i.e., não afete diretamente a relação com clientes, pode

representar custos relevantes no processo fabril como os de armazenagem ou ainda os relativos à liberação prematura de recursos do caixa, como taxas de financiamento e empréstimo.

2.2 O Problema Estocástico de E/T com Determinação de Datas de Entrega

Na seção 2.1 admitiu-se que as datas de entregas são dadas previamente, i.e., compõem o conjunto de parâmetros do problema. Todavia é perfeitamente factível a possibilidade do programador de produção possuir influência sobre o processo de negociação das datas de entrega das tarefas, em específico quando as etapas de produção estão inseridas num ambiente orientado a projetos. Neste cenário, tais datas passariam então a compor o conjunto de variáveis de decisão do problema de otimização, juntamente com o sequenciamento das tarefas. Um resultado imediato da liberdade de escolha das datas de entrega é que não há a necessidade de inserção de tempo de espera entre as tarefas, característica bastante comum nos problemas de E/T.

Considere um problema com características idênticas às apresentadas na seção 2.1, à exceção de que as datas de entrega são variáveis e devem ser determinadas simultaneamente ao sequenciamento de tarefas. A este problema define-se como problema estocástico de E/T para máquina simples com determinação das datas de entrega, o qual constitui o objeto de estudo deste trabalho.

3 Revisão Bibliográfica e Proposta de Pesquisa

Neste capítulo é apresentada uma revisão bibliográfica do tema, desde os primeiros estudos a abordarem o problema de E/T em sua variedade determinística, perpassando pelo problema de determinação de datas no ambiente de produção, assim como estudos mais recentes dedicados ao problema estocástico. Ao final é apresentado o objetivo da pesquisa.

3.1 Problema Determinístico

O problema de programação de tarefas sujeito a *earliness* e *tardiness* tem suas origens no trabalho seminal de Sidney (1977), o qual considerou um problema de uma máquina, com tempos de processamento determinístico e tarefas que precisavam ser executadas dentro de uma janela de tempo pré-definida e incorreriam custos apenas no caso em que o processamento da tarefa ocorresse fora desse intervalo. Sua estratégia de resolução dividiu-se em duas etapas: primeiramente seria preciso definir uma sequência para as tarefas; de posse dessa sequência o passo seguinte seria definir as datas de início de processamento para cada tarefa.

Seguindo uma linha de pesquisa diferente, Kanet (1981) considera o problema de *earliness* e *tardiness* (sem custos) em torno de uma data de entrega comum. Assumindo que esta data de entrega é no mínimo igual a soma dos tempos de processamento das tarefas (chamada de versão irrestrita do problema), Kanet apresentou um algoritmo com execução em tempo polinomial capaz de fornecer a solução ótima do problema.

Anos depois o mesmo problema tratado por Kanet foi objeto de estudo de Hall e Posner (1991), contudo considerando custos de atraso e adiantamento iguais para cada tarefa. Entre as diversas propriedades demonstradas destacam-se o sequenciamento em “V” da solução (i.e., tarefas finalizadas antes da data de entrega são organizadas em ordem não-crescente de tempos de processamento, enquanto as iniciadas após são organizadas em ordem não-decrescente) e que o problema de decisão associado é NP-Completo. Hall, Kubiak e Sethi (1991) também mostram que a versão restrita do problema (i.e., a data de entrega é menor que o somatório dos tempos de processamento), desconsiderados os custos unitários, é também NP-hard.

Contudo é natural considerar o problema em que os custos unitários de *earliness* e *tardiness* são distintos (mas cada um deles assuma o mesmo valor para cada uma das tarefas), pois é esperado que os eventos de atraso ou adiantamento sejam penalizados de forma diferente. Bagchi, Sullivan e Chang (1986) estudaram ambas as versões do problema e verificaram que para a versão irrestrita as propriedades do problema se mantêm, fornecendo

um algoritmo alternativo com execução em tempo polinomial. Para a versão restrita do problema, contudo, os autores não foram capazes de determinar condições de otimalidade, exceto impondo que a data de início do processamento seja o instante inicial zero.

Um problema mais sofisticado considera que os custos unitários de atraso e adiantamento, além de diferentes são individuais para cada tarefa. Conforme apontam Baker e Scudder (1990), este problema foi objeto de estudo de diversos autores, os quais verificaram que propriedades como o sequenciamento em “V” continuam válidas. Como esperado, o caráter *NP-hard* do problema se mantém, conforme mostrado pelo trabalho de Hoogeveen e van de Velde (1991). Como condições de otimalidade ainda não foram definidas, o uso de heurísticas é justificado e foi apresentado por vários autores (para estudos mais recentes ver, por exemplo, Hino, Ronconi e Mendes (2005) e Liao e Cheng (2007)).

Por fim, pode-se considerar que o problema admite datas de entrega diferentes para cada uma das tarefas, o que claramente torna-o ainda mais complexo. Propriedades anteriormente utilizadas para busca de soluções não são mais válidas em relação aos problemas apresentados (o sequenciamento não necessariamente é em forma de “V” por exemplo). Garey, Tarjan e Wilfong (1988) mostraram que este problema é *NP-hard* e apresentaram um algoritmo para o caso em que todas as tarefas possuem o mesmo tempo de processamento com execução em tempo polinomial e que fornece a solução ótima.

Dada sua complexidade, diversos autores se dedicaram a fornecer heurísticas para o problema de E/T com datas e custos distintos. Entre os diversos estudos destacam-se os de Lee e Choi (1995), os quais aplicaram uma abordagem baseada em algoritmos genéticos com bons resultados, e Mazzini e Armentano (2001), cuja heurística construtiva além de apresentar bons resultados é capaz de inserir simultaneamente diferentes tempos ociosos entre as tarefas de forma ótima.

Contudo como mencionado anteriormente, há especial interesse no caso em que a data de entrega é uma variável de decisão do problema, pois, como afirmam Gordon, Proth e Chu (2002), na prática é comum ocorrer a negociação de datas de entrega com clientes. Este problema tem suas origens no trabalho de Seidmann, Panwalkar e Smith (1981). O problema estudado consistia, além da determinação da sequência que minimizava os custos (constantes para todas as tarefas) de *earliness* e *tardiness* (acrescido de um termo para evitar tempos de término de processamento muito grandes), na definição das datas de entrega para cada tarefa. Contudo, diferentemente dos problemas anteriores, este se mostrou de simples resolução: para determinar as datas de entrega ótimas, basta igualá-las aos instantes de término de

processamento de cada tarefa, sendo provavelmente este o motivo pelo qual obteve pouca atenção na literatura.

Panwalkar, Smith e Seidmann (1982) estudaram um problema similar, contudo considerando que a data de entrega a ser definida era comum para todas as tarefas. Os autores apresentaram um algoritmo com execução em tempo polinomial capaz de fornecer a solução ótima para o problema e apontam que a quantidade de tarefas que serão processadas com atraso independe do tempo de processamento destas, sendo uma razão dos custos de atraso e adiantamento.

Uma generalização natural deste problema consiste em considerar que cada uma das tarefas seja caracterizada por custos de atraso e adiantamento individuais e não necessariamente iguais. Hall e Posner (1991) mostraram que este problema é NP-*hard*, mesmo que se considere que os custos de *earliness* e *tardiness* sejam iguais para cada tarefa (o chamado problema simétrico) e apresentam um algoritmo com programação dinâmica capaz de fornecer soluções ótimas mesmo para grandes instâncias.

Conforme atestam Gordon, Proth e Chu (2002), diversos autores se dedicaram a esse problema e identificaram que propriedades do problema com datas de entrega dadas continuam válidas, como o formato em “V” do sequenciamento ótimo. Diversas heurísticas foram propostas para o problema, como, por exemplo, o método de geração de colunas de van den Akker, Hoogeveen e van de Velde (2002) e as heurísticas propostas por Biskup e Feldmann (2001), cujos resultados são considerados um bom *benchmark* para o problema.

3.2 Problema Estocástico

Dos trabalhos referenciados, percebe-se grande interesse na área acadêmica pelo problema com tempos de processamento determinísticos, sendo objeto de estudo de diversos autores ao longo dos últimos anos, conforme indicam os resumos compilados por Cheng e Gupta (1989), Baker e Scudder (1990) e Gordon, Proth e Chu (2002). Até o momento, de acordo com a pesquisa bibliográfica efetuada, o problema com tempos de processamento estocástico não recebeu o mesmo nível de atenção da comunidade acadêmica, não disponibilizando de um quadro completo de seu atual estágio de evolução.

Um dos primeiros autores a considerar aspectos estocásticos no problema de sequenciamento em máquina única com E/T foi Cheng (1986 e 1991). O autor estudou o problema de definição de datas de entrega e sequenciamento de tarefas com tempos de processamento aleatórios em uma máquina única, cuja função custo depende dos valores das datas de entrega e dos desvios quadráticos dos tempos de término em relação a tais datas.

Cheng apresentou condições de otimalidade para alguns casos especiais e desenvolveu algoritmos (não-ótimos) com tempo de execução polinomial, contudo sua pesquisa não apresentou resultados computacionais, limitando-se a desenvolvimentos teóricos.

Sarin, Erel e Esteiner (1991) consideraram o problema estocástico de E/T com data de entrega comum para todas as tarefas, cujos tempos de processamento obedecem a uma distribuição normal. Seus resultados apontam que impondo algumas condições para as médias dos tempos de processamento, a sequência ótima possui formato em “V”, o que pode guiar métodos heurísticos. Os autores apresentaram um método ótimo de busca hierárquico aplicável a problemas de pequeno porte e uma heurística com bons resultados para problemas maiores.

Soroush e Fredendall (1994) efetuam uma aprofundada investigação do problema de *earliness* e *tardiness* com custos individuais de atraso e adiantamento e datas de entrega conhecidas previamente. Considerando tempos de processamento modelados como variáveis aleatórias com distribuições normais, os autores apresentam um desenvolvimento teórico completo do problema e fornecem três heurísticas para sua resolução. O mesmo problema foi estudado por Hussain e Sastry (1999) que utilizaram uma abordagem baseada em algoritmos genéticos, a qual apresentou resultados melhores que os de Soroush e Fredendall e com menor tempo de execução.

Al-Turki, Mittenthal e Raghavachari (1996) também estudaram um problema estocástico de E/T com custos individuais de atraso e adiantamento, contudo considerando um termo inicial na função objetivo para lidar com datas de entrega muito grandes, tempos de processamento modelados como variáveis aleatórias com distribuição geométrica. Expressões para a determinação das datas de entrega ótimas (tanto para o caso em que todas as tarefas partilham a mesma data comum de entrega, quanto o caso de datas distintas) são fornecidas. Os autores investigam ainda o caso especial em que a aleatoriedade dos tempos de processamento decorre de quebras de maquinário.

Cai e Zhou (1997) investigaram o problema de determinação de uma data de entrega comum e tempos de processamento independentes caracterizados por uma distribuição normal, acrescido de um termo para limitar o tempo total de fluxo da sequência. Interessante notar que além de fornecer uma heurística para o problema, baseados na lei dos grandes números, os autores investigam sua eficácia ao se considerarem diferentes distribuições (a saber, uniforme, Laplace e exponencial).

Uma investigação bastante completa do problema considerando a determinação de datas de entrega distintas e custos individuais de atraso e adiantamento foi o objeto de estudo de Soroush (1999). O autor apresentou uma heurística com execução em tempo polinomial

composta por duas regras de despacho e definiu expressões para determinar as datas de entrega ótimas, dada uma sequência de tarefas, as quais são modeladas como variáveis aleatórias com distribuição normal. A abordagem apresentada por Soroush foi profundamente investigada por Portougal e Trietsch (2006a), os quais apresentaram diversas condições de otimalidade do problema e propriedades que justificam a boa performance do método. Soroush (1996) dedica-se ainda a extensões da função objetivo, considerando termos exponenciais e quadráticos em sua composição.

Um problema ligeiramente diferente é estudado por Baker e Trietsch (2009b), no qual é considerada uma função cujo objetivo é minimizar o valor esperado do atraso e uma data comum de entrega para tarefas com tempos de processamento caracterizados por uma distribuição normal. Uma posterior extensão do problema mostra que ele é equivalente a minimizar os custos de atraso e adiantamento e o tempo de término de cada tarefa. São apresentadas e comparadas diversas heurísticas para a resolução do problema, uma delas apresentando resultados ótimos na grande maioria dos casos avaliados.

3.3 Proposta da Pesquisa

Conforme apresentado, o problema estocástico de atraso e adiantamento de máquina única ainda demanda estudos mais aprofundados, visto que é uma área de estudo ainda relativamente recente no meio acadêmico. É de especial interesse o problema estudado por Soroush (1999). Baseado em suas investigações e no trabalho de Soroush e Fredendall (1994), o presente trabalho apresentará uma heurística construtiva capaz de fornecer soluções melhores do que aquelas apresentadas por Soroush, com tempo de execução polinomial. Tal se justifica dada a atestada complexidade do problema e o inviável custo computacional para execução de algoritmos exatos, como será apresentado. De posse do conhecimento acumulado no caso de tempos de processamento com distribuição normal, o mesmo problema será expandido considerando-se que os tempos de processamento são modelados como variáveis aleatórias com distribuição lognormal.

4 FORMULAÇÃO DO PROBLEMA COM DISTRIBUIÇÃO NORMAL E DESENVOLVIMENTO MATEMÁTICO

Este capítulo é dedicado à formulação matemática do problema, suas notações, assim como desenvolvimentos e demonstrações necessários para o problema com distribuição normal, a qual é candidata frequente quando da modelagem de fenômenos aleatórios. A escolha deste tipo de distribuição é natural, dado seu largo uso nos mais diversos campos do conhecimento. Feitas as devidas adaptações para o problema (essencialmente considerar apenas sua porção positiva para os tempos de processamento), suas propriedades matemáticas permitem a dedução de expressões, senão fechadas, ao menos com custo computacional relativamente baixo. Além disso, como será visto mais adiante, o Teorema Central do Limite garante ainda que para uma quantidade razoável de tarefas, os resultados obtidos para o caso normal são extensíveis para outros tipos de distribuição.

Todos os desenvolvimentos apresentados neste capítulo são baseados no trabalho apresentado por Soroush e Fredendall (1994).

4.1 Formulação Matemática

Os seguintes índices e variáveis serão utilizados no decorrer deste trabalho:

Parâmetros

n	Quantidade de tarefas disponíveis para processamento;
$s = ([1], [2] \dots [i] \dots [n])$	Sequência de tarefas, onde $[i]$ ($i = 1, 2, 3, \dots, n$) indica uma tarefa ocupando a i -ésima posição da sequência;
$p_{[i]}$	Tempo de processamento da tarefa $[i]$;
$f_{[i]}(X)$	Função densidade de probabilidade da variável normal X associada à tarefa $[i]$;
$m_{[i]}$	Média do tempo de processamento $p_{[i]}$;
$v_{[i]}$	Variância do tempo de processamento $p_{[i]}$;
$\alpha_{[i]}$	Custo unitário de adiantamento da tarefa $[i]$;
$\beta_{[i]}$	Custo unitário de atraso da tarefa $[i]$;

Variáveis

$t_{[i]}$	Tempo de término da tarefa i , onde $t_{[i]} = \sum_{k=1}^i p_{[k]}$;
-----------	--

Média do tempo de término ;
 Variância do tempo de término ;
 Data de entrega da tarefa ;
 Adiantamento da tarefa ;
 Atraso da tarefa ;
 Função de custo de atraso e adiantamento da tarefa dependente da data de entrega .

O custo de cada tarefa é dado pela função definida como:

(4.1)

Graficamente o atraso e o adiantamento podem ser representados conforme Figura 1 a seguir.

Figura 1 – Representação gráfica do atraso e adiantamento das tarefas

Visto que os tempos de processamento , são variáveis aleatórias normais, depreendesse que o tempo de término da tarefa, , será também uma variável aleatória. É possível definir a expressão do tempo de término efetuando a convolução do tempo de processamento da tarefa com os tempos de processamento de todas as tarefas que a precedem. Por exemplo, para , tem-se que:

Assim:

$$t_{[2]}(x) = f_{[1]} * f_{[2]} = \int_{-\infty}^{\infty} f_{[1]}(x-t)f_{[2]}(t) dt$$

Dado que $f_{[i]}(t) = \frac{1}{\sqrt{2\pi v_{[i]}}} e^{-\frac{(t-m_{[i]})^2}{2v_{[i]}}}$, é possível demonstrar que (ver, por exemplo,

Ross (2010)):

$$t_{[2]}(x) = \frac{1}{\sqrt{2\pi(v_{[1]} + v_{[2]})}} e^{-\frac{[x-(m_{[1]}+m_{[2]})]^2}{2(v_{[1]}+v_{[2]})}}$$

Ou seja, o tempo de término da tarefa 2 é caracterizado por uma distribuição normal com média igual à soma das médias dos tempos de processamento das tarefas 1 e 2 e com variância igual à soma das variâncias dos tempos de processamento destas tarefas. Pode-se então generalizar que a média e a variância do tempo de término da tarefa $[i]$ são dadas por:

$$\mu_{[i]} = \sum_{k=1}^i m_{[k]} \quad (4.2)$$

$$\sigma_{[i]} = \sqrt{\sum_{k=1}^i v_{[k]}} \quad (4.3)$$

Da mesma forma, o custo $c(d_{[i]})$ também será uma variável aleatória. Por conseguinte, uma medida razoável associada ao custo é o seu valor esperado, dado por:

$$E[c(d_{[i]})] = C(d_{[i]}) = E[\alpha_{[i]} \max(d_{[i]} - t_{[i]}, 0) + \beta_{[i]} \max(t_{[i]} - d_{[i]}, 0)] \quad (4.4)$$

Onde $E[X]$ é a função que define o valor esperado de uma variável aleatória X . A equação (4.4) acima pode ser reescrita como:

$$\mathcal{C}(d_{[i]}) = \mathcal{E}(d_{[i]}) + \mathcal{T}(d_{[i]}) \quad (4.5)$$

Onde:

$$\mathcal{E}(d_{[i]}) = \alpha_{[i]} \int_{-\infty}^{d_{[i]}} (d_{[i]} - t_{[i]}) f_{[i]}(t) dt \quad (4.6)$$

$$\mathcal{T}(d_{[i]}) = \beta_{[i]} \int_{d_{[i]}}^{+\infty} (t_{[i]} - d_{[i]}) f_{[i]}(t) dt \quad (4.7)$$

(Note que as características próprias das funções de atraso e adiantamento foram aplicadas para simplificação dos limites das integrais).

Desta forma, a função objetivo do problema de otimização é dada por:

$$\mathcal{C}(d) = \sum_{i=1}^n \mathcal{E}(d_{[i]}) + \mathcal{T}(d_{[i]}) \quad (4.8)$$

Onde d é o vetor composto pelas datas de entrega $d_{[i]}$, ou seja, $d = (d_{[1]}, d_{[2]}, d_{[3]}, \dots, d_{[n]})$.

Todavia uma das premissas do problema em estudo é que os tempos de processamento das tarefas seguem uma distribuição normal. Tal propriedade é de extrema importância, pois permite a aplicação dos teoremas apresentados a seguir.

Teorema 1. O valor esperado do adiantamento $\mathcal{E}(d_{[i]})$ da tarefa $[i]$ da sequência é dado por:

$$\mathcal{E}(d_{[i]}) = \alpha_{[i]} \sigma_{[i]} [\phi(z_{[i]}) + z_{[i]} \Phi(z_{[i]})] \quad (4.9)$$

Onde $z_{[i]} = (d_{[i]} - \mu_{[i]}) / \sigma_{[i]}$ é uma função ainda desconhecida e dependente da data de entrega $d_{[i]}$ e $\phi(X)$ e $\Phi(X)$ são respectivamente a função densidade de probabilidade e a função probabilidade cumulativa para uma variável aleatória normal X padronizada.

Demonstração. Considerando uma distribuição normal, a equação (4.6) pode ser escrita como:

$$\mathcal{E}(d_{[i]}) = \alpha_{[i]} \int_{-\infty}^{d_{[i]}} (d_{[i]} - t_{[i]}) \frac{1}{\sqrt{2\pi}\sigma_{[i]}} e^{-\frac{(t-\mu_{[i]})^2}{2\sigma_{[i]}^2}} dt$$

Ou ainda:

$$\mathcal{E}(d_{[i]}) = \alpha_{[i]} \int_{-\infty}^{d_{[i]}} \left(\frac{d_{[i]} - \mu_{[i]}}{\sigma_{[i]}} - \frac{t_{[i]} - \mu_{[i]}}{\sigma_{[i]}} \right) \frac{1}{\sqrt{2\pi}} e^{-\frac{(t-\mu_{[i]})^2}{2\sigma_{[i]}^2}} dt$$

Efetuada as substituições $z_{[i]} = (d_{[i]} - \mu_{[i]})/\sigma_{[i]}$ e $x = (t_{[i]} - \mu_{[i]})/\sigma_{[i]}$, é possível escrever que:

$$\mathcal{E}(d_{[i]}) = \alpha_{[i]}\sigma_{[i]} \int_{-\infty}^{z_{[i]}} (z_{[i]} - x) \phi(x) dx \quad (4.10)$$

Dado que x é, por construção, uma variável aleatória normal padrão e $\phi(x)$ é a função densidade de probabilidade normal padrão.

Tem-se que:

$$\mathcal{E}(d_{[i]}) = \alpha_{[i]}\sigma_{[i]} \left[\int_{-\infty}^{z_{[i]}} (z_{[i]} - x) \phi(x) dx + \int_{z_{[i]}}^{\infty} (z_{[i]} - x) \phi(x) dx + \int_{z_{[i]}}^{\infty} (x - z_{[i]}) \phi(x) dx \right]$$

$$\mathcal{E}(d_{[i]}) = \alpha_{[i]}\sigma_{[i]} \int_{-\infty}^{\infty} (z_{[i]} - x) \phi(x) dx + \alpha_{[i]}\sigma_{[i]} \int_{z_{[i]}}^{\infty} (x - z_{[i]}) \phi(x) dx$$

Ou seja:

$$\mathcal{E}(d_{[i]}) = \alpha_{[i]}\sigma_{[i]} [E(z_{[i]} - X) + Q_{[i]}(z_{[i]})] \quad (4.11)$$

Com $Q_{[i]}(z_{[i]}) = \int_{z_{[i]}}^{\infty} (x - z_{[i]}) \phi(x) dx$. Substituindo $\phi(x)$ por sua expressão completa, tem-se:

$$\begin{aligned}
Q_{[i]}(z_{[i]}) &= \int_{z_{[i]}}^{\infty} (x - z_{[i]}) \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx \\
&= \int_{z_{[i]}}^{\infty} \frac{x}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx - z_{[i]} \int_{z_{[i]}}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx \\
&= \phi(z_{[i]}) - z_{[i]}[1 - \Phi(z_{[i]})]
\end{aligned} \tag{4.12}$$

Dado que $\Phi(z_{[i]}) = \int_{-\infty}^{z_{[i]}} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx$ e ainda:

$$\int_{z_{[i]}}^{\infty} \frac{x}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx = \frac{1}{\sqrt{2\pi}} e^{-\frac{z_{[i]}^2}{2}} = \phi(z_{[i]})$$

Substituindo (4.12) em (4.11) e considerando que $E(z_{[i]} - x) = z_{[i]} - E(x) = z_{[i]}$, segue que:

$$\mathcal{E}(d_{[i]}) = \alpha_{[i]} \sigma_{[i]} [z_{[i]} + \phi(z_{[i]}) - z_{[i]}[1 - \Phi(z_{[i]})]]$$

Por fim:

$$\mathcal{E}(d_{[i]}) = \alpha_{[i]} \sigma_{[i]} [\phi(z_{[i]}) + z_{[i]} \Phi(z_{[i]})] \tag{c.q.d.}$$

Teorema 2. O valor esperado do atraso $\mathcal{T}(d_{[i]})$ da tarefa $[i]$ da sequência é dado por:

$$\mathcal{T}(d_{[i]}) = \beta_{[i]} \sigma_{[i]} \{\phi(z_{[i]}) - z_{[i]}[1 - \Phi(z_{[i]})]\} \tag{4.13}$$

Demonstração. Idêntica à demonstração da equação (4.9), motivo pelo qual será omitida.

É possível levantar questionamentos quanto à premissa de atribuir-se uma distribuição normal aos tempos de processamento. Conforme apontam Cai e Zhou (1997), pelo Teorema

Central do Limite, sabe-se que será garantida a normalidade de uma amostra desde que esta possua uma quantidade razoável de tarefas. Desta forma, ainda que na prática os tempos de processamento das tarefas não sejam necessariamente variáveis aleatórias normais, à medida que se aumenta o tamanho do problema, a somatória das tarefas (ou mais formalmente a convolução destas) se aproximará de uma distribuição normal, reduzindo os erros do modelo.

Substituindo as equações (4.9) e (4.13) na equação (4.8), segue que o custo médio total será dado por:

$$\mathcal{C}(d) = \sum_{i=1}^n \alpha_{[i]} \sigma_{[i]} [\phi(z_{[i]}) + z_{[i]} \Phi(z_{[i]})] + \beta_{[i]} \sigma_{[i]} \{\phi(z_{[i]}) - z_{[i]} [1 - \Phi(z_{[i]})]\} \quad (4.14)$$

Onde $d = (d_{[1]}, d_{[2]}, d_{[3]}, \dots, d_{[n]})$ e $z_{[i]} = (d_{[i]} - \mu_{[i]}) / \sigma_{[i]}$. Como o custo médio total é função do desvio do tempo de término de cada tarefa ($\sigma_{[i]}$), este é dependente da sequência.

4.2 Datas de entrega ótimas

Considere que o conjunto de tarefas do problema foi previamente sequenciado, ou seja, já estão definidos a média do tempo de término $\mu_{[i]}$ e o desvio do tempo de término $\sigma_{[i]}$ para cada uma das tarefas (o método utilizado para a geração dessa sequência não é importante por ora). De posse de tal sequência, tem-se como objetivo determinar as datas de entrega que otimizam o custo médio total $\mathcal{C}(d)$.

A primeira derivada da equação (4.14) em relação ao vetor de datas de entrega d é dada por:

$$\frac{\partial \mathcal{C}(d)}{\partial d_{[i]}} = (\alpha_{[i]} + \beta_{[i]}) \left[\sigma_{[i]} \frac{\partial \phi(z_{[i]})}{\partial d_{[i]}} + \Phi(z_{[i]}) + (d_{[i]} - \mu_{[i]}) \frac{\partial \Phi(z_{[i]})}{\partial d_{[i]}} \right] - \beta_{[i]} \quad (4.15)$$

Pela regra da cadeia, tem-se que a derivada parcial dependente da função $\phi(z_{[i]})$ pode ser escrita como:

$$\frac{\partial \phi(z_{[i]})}{\partial d_{[i]}} = \frac{\partial \phi(z_{[i]})}{\partial z_{[i]}} \frac{\partial z_{[i]}}{\partial d_{[i]}}$$

Dessa forma:

$$\begin{aligned}
\frac{\partial \phi(z_{[i]})}{\partial d_{[i]}} &= \frac{\partial}{\partial z_{[i]}} \left(\frac{1}{\sqrt{2\pi}} e^{-\frac{z_{[i]}^2}{2}} \right) \frac{\partial z_{[i]}}{\partial d_{[i]}} \\
&= \left(-\frac{z_{[i]}}{\sqrt{2\pi}} e^{-\frac{z_{[i]}^2}{2}} \right) \frac{1}{\sigma_{[i]}} \\
\frac{\partial \phi(z_{[i]})}{\partial d_{[i]}} &= -\frac{(d_{[i]} - \mu_{[i]})}{\sigma_{[i]}^2} \phi(z_{[i]}) \tag{4.16}
\end{aligned}$$

Já para derivada parcial dependente da função de probabilidade acumulada $\Phi(z_{[i]})$, tem-se:

$$\begin{aligned}
\frac{\partial \Phi(z_{[i]})}{\partial d_{[i]}} &= \frac{\partial}{\partial d_{[i]}} \int_{-\infty}^{z_{[i]}} \phi(x) dx \\
&= \left(\frac{\partial}{\partial z_{[i]}} \int_{-\infty}^{z_{[i]}} \phi(x) dx \right) \frac{\partial z_{[i]}}{\partial d_{[i]}} \\
&= \phi(x) \Big|_{-\infty}^{z_{[i]}} \frac{1}{\sigma_{[i]}} \\
\frac{\partial \Phi(z_{[i]})}{\partial d_{[i]}} &= \frac{1}{\sigma_{[i]}} \phi(z_{[i]}) \tag{4.17}
\end{aligned}$$

Por fim, substituindo as equações (4.16) e (4.17) em (4.15), resulta:

$$\begin{aligned}
\frac{\partial \mathcal{C}(d)}{\partial d_{[i]}} &= (\alpha_{[i]} + \beta_{[i]}) \left[-\sigma_{[i]} \frac{(d_{[i]} - \mu_{[i]})}{\sigma_{[i]}^2} \phi(z_{[i]}) + \Phi(z_{[i]}) \right. \\
&\quad \left. + (d_{[i]} - \mu_{[i]}) \frac{1}{\sigma_{[i]}} \phi(z_{[i]}) \right] - \beta_{[i]} \\
\frac{\partial \mathcal{C}(d)}{\partial d_{[i]}} &= (\alpha_{[i]} + \beta_{[i]}) \Phi(z_{[i]}) - \beta_{[i]} \tag{4.18}
\end{aligned}$$

Uma solução para esta expressão pode ser obtida igualando-a a zero, ou seja:

$$\frac{\partial \mathcal{C}(d)}{\partial d_{[i]}} = (\alpha_{[i]} + \beta_{[i]})\Phi(z_{[i]}) - \beta_{[i]} = 0 \quad (4.19)$$

A quantidade $z_{[i]}^*$ que anula a equação (4.19) acima será dada por:

$$z_{[i]}^* = \Phi^{-1}\left(\frac{\beta_{[i]}}{\alpha_{[i]} + \beta_{[i]}}\right) \quad (4.20)$$

Onde $\Phi^{-1}(X)$ é a função inversa de $\Phi(X)$ para o ponto $\beta_{[i]}/(\alpha_{[i]} + \beta_{[i]})$, a qual pode ser obtida da tabela da função probabilidade cumulativa normal. Observe que como $\Phi(z_{[i]}^*)$ é uma expressão do nível de serviço esperado para a conclusão da tarefa $[i]$, este é uma função direta dos custos de adiantamento e atraso próprios da tarefa. Como $z_{[i]}^*$ é função de $d_{[i]}^*$, i.e., a data de entrega da tarefa $[i]$ da sequência que otimiza o custo médio total, pode-se concluir que:

$$d_{[i]}^* = \mu_{[i]} + \sigma_{[i]}\Phi^{-1}\left(\frac{\beta_{[i]}}{\alpha_{[i]} + \beta_{[i]}}\right) \quad (4.21)$$

Percebe-se da equação (4.21) que, dada uma sequência das tarefas, a data de entrega ótima (local, até o momento) é uma função da média do tempo de término da tarefa acrescida de um fator de correção dependente dos custos individuais de atraso e adiantamento (o nível de serviço requerido pela tarefa) e do desvio do tempo de término. Este resultado é uma expansão consistente com o caso determinístico, visto que para tal cenário $\sigma_{[i]} = 0$ e, dessa forma, $d_{[i]}^* = \mu_{[i]}$, ou seja, a data de entrega ótima de determinada tarefa será obtida igualando seu valor ao seu instante de término.

Mais que isso, esse fator de correção constitui um termo adicional de segurança que cresce quanto maior for a quantidade de tarefas anteriores à tarefa $[i]$ na sequência (visto que os tempos de processamento são independentes e, por conseguinte, $\sigma_{[i]} = \sqrt{\sum_{k=1}^i v_{[k]}}$ para o problema em questão), o qual diminui a necessidade de executar um novo sequenciamento em caso de imprevistos durante o processamento. Caso a tarefa possua custos de atraso maiores que os custos de adiantamento, este fator será somado à média do tempo de término, caso

contrário, será subtraído. Essa proteção contra eventos indesejados mostra então que a solução obtida é robusta.

Para que tal resultado seja um ótimo global, basta demonstrar que a função $\mathcal{C}(d)$ é convexa em d , ou seja, sua matriz Hessiana é pelo menos positiva semidefinida para todo $d_{[i]}$. Da equação (4.18) pode-se escrever que:

$$\begin{aligned}\frac{\partial^2 \mathcal{C}(d)}{\partial d_{[i]}^2} &= \frac{\partial}{\partial d_{[i]}} [(\alpha_{[i]} + \beta_{[i]})\Phi(z_{[i]}) - \beta_{[i]}] \\ &= (\alpha_{[i]} + \beta_{[i]}) \frac{\partial \Phi(z_{[i]})}{\partial d_{[i]}}\end{aligned}\quad (4.22)$$

Para os termos da diagonal da matriz, substituindo a equação (4.17) em (4.22), vem:

$$\frac{\partial^2 \mathcal{C}(d)}{\partial d_{[i]}^2} = \frac{(\alpha_{[i]} + \beta_{[i]})}{\sigma_{[i]}} \phi(z_{[i]}) \quad (4.23)$$

Dado que todos os termos componentes do lado direito da equação (4.23) são positivos para todo i , conclui-se que $\partial^2 \mathcal{C}(d)/\partial d_{[i]}^2 \geq 0$. Para os termos fora da diagonal:

$$\begin{aligned}\frac{\partial^2 \mathcal{C}(d)}{\partial d_{[i]} \partial d_{[k]}} &= \frac{\partial}{\partial d_{[k]}} [(\alpha_{[i]} + \beta_{[i]})\Phi(z_{[i]}) - \beta_{[i]}] \\ \frac{\partial^2 \mathcal{C}(d)}{\partial d_{[i]} \partial d_{[k]}} &= \frac{(\alpha_{[i]} + \beta_{[i]})}{\sigma_{[i]}} \phi(z_{[i]}) \frac{\partial \Phi(z_{[i]})}{\partial d_{[k]}}\end{aligned}\quad (4.24)$$

Com $k \neq i, k = 1, 2, \dots, n$. Como $z_{[i]}$ não é função de $d_{[k]}$, conclui-se que $\partial^2 \mathcal{C}(d)/\partial d_{[i]} \partial d_{[k]} = 0$. Dessa forma a matriz Hessiana de $\mathcal{C}(d)$ é positiva semidefinida, indicando que $\mathcal{C}(d)$ é uma função convexa em d . Conclui-se então que as datas de entrega obtidas por meio da equação (4.21) correspondem aos ótimos globais da função.

Todavia este resultado não garante que os tempos de entrega assumam valores inteiros apenas. Note que o resultado, além de global, é único, dado que a equação (4.19)

admite apenas uma solução. Dessa forma, para determinar as soluções inteiras, basta efetuar uma busca local na vizinhança da solução obtida, conforme procedimento a seguir:

Algoritmo 1 (Datas de entrega ótimas inteiras – distribuição normal)

1. Calcular $d_{[i]}^*$ conforme equação (4.21);
2. Calcular o piso e o teto do valor obtido no passo anterior (respectivamente $d_{[i]}^f$ e $d_{[i]}^c$);
3. A partir dos valores de piso e teto obtidos, calcular as quantidades $z_{[i]}^f$ e $z_{[i]}^c$;
4. Aplicar as equações (4.5), (4.9) e (4.13) para obter os custos associados $C(d_{[i]}^f)$ e $C(d_{[i]}^c)$;

A data de entrega inteira ótima será o valor $d_{[i]}^\#$ tal que:

$$C(d_{[i]}^\#) = \min [C(d_{[i]}^f), C(d_{[i]}^c)] \quad (4.25)$$

Deste modo, dada uma sequência, a aplicação das equações (4.21) e (4.25) define completamente o vetor de datas de entrega ótimas e inteiras.

4.3 Sequência Ótima

O objetivo desta seção é determinar a sequência que minimiza o custo médio total dado pela equação (4.14). Definida tal sequência, as datas de entrega ótimas serão obtidas aplicando-se a equações (4.21) e (4.25), conforme mostrado na seção 4.2.

Contudo é importante observar que o custo dado pela equação (4.14) é uma função não-linear das médias e desvios dos tempos de término das tarefas em uma sequência e não é nem uma função aditiva nem multiplicativa separável dos valores dos custos. Com base nesse fato, Soroush (1999) apontou que uma recursão aditiva ou multiplicativa com uso de programação dinâmica ou mesmo o uso de técnicas de *branch and bound* não são possíveis para encontrar uma solução ótima. Assim sendo, a única abordagem possível para buscar a solução exata é a enumeração completa de todas as $n!$ possíveis sequências e a seguir aplicar a equação (4.14) para cada uma delas de forma a determinar aquela que possui o menor custo.

Devido ao elevado esforço computacional requerido por essa tarefa, é razoável buscar métodos que forneçam soluções aproximadas para o problema, o que será discutido a seguir.

4.4 Regra SOR

É notório que o método de enumeração pode consumir bastante tempo de acordo com o tamanho do problema. Desta forma, Soroush (1999) apresentou um algoritmo com tempo de execução polinomial composto por duas diferentes regras estáticas de despacho para fornecer soluções candidatas ao problema. Tal abordagem será aqui definida como Regra SOR.

Para uma melhor compreensão dessa regra, alguns desenvolvimentos fazem-se necessários. O primeiro consiste em substituir a equação (4.20) em (4.14), o que resulta em:

$$\begin{aligned} \mathcal{C}(d^*) &= \sum_{i=1}^n \alpha_{[i]} \sigma_{[i]} \left[\phi(z_{[i]}^*) + z_{[i]}^* \frac{\beta_{[i]}}{\alpha_{[i]} + \beta_{[i]}} \right] + \beta_{[i]} \sigma_{[i]} \left\{ \phi(z_{[i]}^*) - z_{[i]}^* \left[1 - \frac{\beta_{[i]}}{\alpha_{[i]} + \beta_{[i]}} \right] \right\} \\ \mathcal{C}(d^*) &= \sum_{i=1}^n (\alpha_{[i]} + \beta_{[i]}) \sigma_{[i]} \phi(z_{[i]}^*) \end{aligned} \quad (4.26)$$

Onde $d^* = (d_{[1]}^*, d_{[2]}^*, \dots, d_{[n]}^*)$ é o vetor de datas de entrega obtidas pela aplicação da equação (4.20) e $z_{[i]}^* = (d_{[i]}^* - \mu_{[i]}) / \sigma_{[i]}$ (importante notar que Soroush em sua abordagem ignora a premissa de datas de entrega com valores inteiros).

Pode-se perceber mais uma vez que o resultado obtido é coerente com o caso determinístico associado, dado que para este cenário $\sigma_{[i]} = 0$, o que implica que $\mathcal{C}(d^*) = 0$. Este é um resultado esperado, visto que o vetor d^* é escolhido de forma a minimizar o custo médio total de atraso e adiantamento (i.e. $d_{[i]}^* = \mu_{[i]}$).

Expandindo a função densidade de probabilidade $\phi(X)$, é possível reescrever a equação (4.26) como:

$$\begin{aligned} \mathcal{C}(d^*) &= \sum_{i=1}^n (\alpha_{[i]} + \beta_{[i]}) \sigma_{[i]} \frac{1}{\sqrt{2\pi}} e^{-\frac{z_{[i]}^{*2}}{2}} \\ \mathcal{C}(d^*) &= \sum_{i=1}^n \frac{\gamma_{[i]} \sigma_{[i]}}{\sqrt{2\pi}} \end{aligned} \quad (4.27)$$

Onde $\gamma_{[i]} = (\alpha_{[i]} + \beta_{[i]})e^{-\frac{z_{[i]}^*{}^2}{2}}$. A equação (4.27) evidencia a característica de inseparabilidade da função custo, visto que $\sigma_{[i]} = \sqrt{\sum_{k=1}^i v_{[k]}}$.

A aplicação da Regra SOR prevê a resolução de dois problemas os quais consistem em limitantes superiores do problema caracterizado pela equação (4.27). Para a construção do primeiro limitante, note que:

$$\sigma_{[i]} = \sqrt{\sum_{k=1}^i v_{[k]}} \leq \sum_{k=1}^i \sqrt{v_{[k]}}$$

Dessa forma:

$$\mathcal{C}(d^*) = \sum_{i=1}^n \frac{\gamma_{[i]} \sigma_{[i]}}{\sqrt{2\pi}} \leq \sum_{i=1}^n \frac{\gamma_{[i]}}{\sqrt{2\pi}} \sum_{k=1}^i \sqrt{v_{[k]}} = \sum_{i=1}^n \sqrt{v_{[i]}} \sum_{k=i}^n \frac{\gamma_{[k]}}{\sqrt{2\pi}} = \mathcal{C}^u(d^*) \quad (4.28)$$

O segundo limitante superior pode ser construído assumindo-se que $\sum_{k=1}^i v_{[k]} \geq 1$, caso em que a seguinte inequação é válida:

$$\sigma_{[i]} = \sqrt{\sum_{k=1}^i v_{[k]}} \leq \sum_{k=1}^i v_{[k]}$$

O que resulta em:

$$\mathcal{C}(d^*) = \sum_{i=1}^n \frac{\gamma_{[i]} \sigma_{[i]}}{\sqrt{2\pi}} \leq \sum_{i=1}^n \frac{\gamma_{[i]}}{\sqrt{2\pi}} \sum_{k=1}^i v_{[k]} = \sum_{i=1}^n v_{[i]} \sum_{k=i}^n \frac{\gamma_{[k]}}{\sqrt{2\pi}} = \mathcal{C}^v(d^*) \quad (4.29)$$

A partir desses resultados é possível apresentar os teoremas que seguem.

Teorema 3. A sequência $\hat{s}$ que minimiza o limitante superior dado pela inequação (4.28) é tal que:

$$\left(\frac{\sqrt{v_{[1]}}}{\gamma_{[1]}} \leq \frac{\sqrt{v_{[2]}}}{\gamma_{[2]}} \leq \dots \leq \frac{\sqrt{v_{[n]}}}{\gamma_{[n]}} \right)$$

Demonstração. Consiste na aplicação do método de troca de pares adjacentes. Considere uma sequência s qualquer tal que para determinadas posições l e $l+1$, ocupadas respectivamente pelas tarefas p e q , $\sqrt{v_{p[l]}}/\gamma_{p[l]} > \sqrt{v_{q[l+1]}}/\gamma_{q[l+1]}$ (note que, para fins de demonstração, a notação original foi adaptada para identificar além da posição, cada uma das tarefas).

Tem-se que:

$$C_s^u(d) = \sum_{i=1}^n \sqrt{v_{[i]}} \sum_{k=i}^n \frac{\gamma_{[k]}}{\sqrt{2\pi}} \quad (4.30)$$

Onde $C_s^u(d)$ é o custo médio total do problema caracterizado pela inequação (4.28) para a sequência s . Este pode ser reescrito como:

$$\begin{aligned} C_s^u(d) = & \sum_{i=1}^{l-1} \sqrt{v_{[i]}} \sum_{k=i}^n \frac{\gamma_{[k]}}{\sqrt{2\pi}} + \sqrt{v_{p[l]}} \left(\frac{\gamma_{p[l]}}{\sqrt{2\pi}} + \frac{\gamma_{q[l+1]}}{\sqrt{2\pi}} + \sum_{k=l+2}^n \frac{\gamma_{[k]}}{\sqrt{2\pi}} \right) \\ & + \sqrt{v_{q[l+1]}} \left(\frac{\gamma_{q[l+1]}}{\sqrt{2\pi}} + \sum_{k=l+2}^n \frac{\gamma_{[k]}}{\sqrt{2\pi}} \right) \\ & + \sum_{i=l+2}^n \sqrt{v_{[i]}} \sum_{k=i}^n \frac{\gamma_{[k]}}{\sqrt{2\pi}} \end{aligned} \quad (4.31)$$

Considere agora a sequência s' , resultado da troca de posições entre as tarefas p e q .

Tem-se que:

$$\begin{aligned} C_{s'}^u(d) = & \sum_{i=1}^{l-1} \sqrt{v_{[i]}} \sum_{k=i}^n \frac{\gamma_{[k]}}{\sqrt{2\pi}} + \sqrt{v_{q[l]}} \left(\frac{\gamma_{q[l]}}{\sqrt{2\pi}} + \frac{\gamma_{p[l+1]}}{\sqrt{2\pi}} + \sum_{k=l+2}^n \frac{\gamma_{[k]}}{\sqrt{2\pi}} \right) \\ & + \sqrt{v_{p[l+1]}} \left(\frac{\gamma_{p[l+1]}}{\sqrt{2\pi}} + \sum_{k=l+2}^n \frac{\gamma_{[k]}}{\sqrt{2\pi}} \right) + \sum_{i=l+2}^n \sqrt{v_{[i]}} \sum_{k=i}^n \frac{\gamma_{[k]}}{\sqrt{2\pi}} \end{aligned} \quad (4.32)$$

Desta forma, subtraindo as equações (4.31) e (4.32), segue:

$$\mathcal{C}_s^u(d) - \mathcal{C}_{s'}^u(d) = \frac{1}{\sqrt{2\pi}} (\sqrt{v_{p[l]}}\gamma_{q[l+1]} - \sqrt{v_{q[l+1]}}\gamma_{p[l]}) \quad (4.33)$$

Dada a premissa inicial que $\sqrt{v_{p[l]}}/\gamma_{p[l]} > \sqrt{v_{q[l+1]}}/\gamma_{q[l+1]}$, resulta:

$$\mathcal{C}_s^u(d) - \mathcal{C}_{s'}^u(d) > 0 \quad (4.34)$$

Ou seja, a troca de pares entre as tarefas p e q reduziu o custo médio total. Dessa forma, a sequência que minimiza o problema definido pela inequação (4.28) deve ser tal que $\sqrt{v_{p[l]}}/\gamma_{p[l]} \leq \sqrt{v_{q[l+1]}}/\gamma_{q[l+1]}$, ou de forma mais geral:

$$\left(\frac{\sqrt{v_{[1]}}}{\gamma_{[1]}} \leq \frac{\sqrt{v_{[2]}}}{\gamma_{[2]}} \leq \dots \leq \frac{\sqrt{v_{[n]}}}{\gamma_{[n]}} \right) \quad (\text{c.q.d.})$$

Teorema 4. A sequência $\tilde{s}$ que minimiza o limitante superior dado pela inequação (4.29) é tal que:

$$\left(\frac{v_{[1]}}{\gamma_{[1]}} \leq \frac{v_{[2]}}{\gamma_{[2]}} \leq \dots \leq \frac{v_{[n]}}{\gamma_{[n]}} \right)$$

Demonstração. Idêntica ao Teorema 3, motivo pelo qual será omitida.

É possível agora descrever a Regra SOR como enunciada por Soroush:

Algoritmo 2 (Regra SOR)

1. Determinar as sequências candidatas $\hat{s}$ e $\tilde{s}$ conforme os Teoremas 3 e 4;
2. Aplicar a equação (4.27) para obter os custos associados a cada uma das sequências candidatas, respectivamente $\mathcal{C}_{\hat{s}}(d)$ e $\mathcal{C}_{\tilde{s}}(d)$;
3. Definir como solução a sequência s^* tal que $\mathcal{C}_{s^*}(d) = \min[\mathcal{C}_{\hat{s}}(d), \mathcal{C}_{\tilde{s}}(d)]$;

De posse da sequência s^* , aplica-se em seguida a equação (4.21) para obter as datas de entrega ótimas e, caso seja de interesse, aplicar o Algoritmo 1 para obtenção de datas de entrega com valores inteiros.

Levando em consideração apenas as etapas referentes à determinação de sequências candidatas (dado que o método para determinação de datas de entrega, como será visto, será mantido para todos os algoritmos), conclui-se que a Regra SOR tem complexidade $O(n \log n)$ (complexidade típica do passo 1). Soroush (1999) mostrou que a aplicação desse algoritmo fornece boas soluções para o problema em estudo, por vezes até resultados ótimos.

É apresentado na Tabela 1 um exemplo que ilustra a aplicação deste algoritmo:

Tabela 1 – Parâmetros para o exemplo ilustrativo

Tarefa	m_i	v_i	α_i	β_i
1	3,00	0,20	2,50	0,05
2	7,00	2,80	0,80	37,80
3	8,00	3,50	1,50	3,50
4	3,60	0,90	5,00	0,13
5	10,00	8,00	0,30	5,70
6	15,00	15,00	1,00	2,00
7	14,50	12,00	0,50	2,00

Aplicando $z_i = \Phi^{-1}\left(\frac{\beta_i}{\alpha_i + \beta_i}\right)$ e $\gamma_i = (\alpha_i + \beta_i)e^{-z_i^2/2}$ aos valores da Tabela 1, é possível construir a Tabela 2.

Tabela 2 – Dados auxiliares para solução do exemplo ilustrativo

Tarefa	z_i	γ_i	v_i/γ_i	$\sqrt{v_i}/\gamma_i$
1	-2,06	0,30	0,66	1,47
2	2,04	4,83	0,58	0,35
3	0,52	4,36	0,80	0,43
4	-1,95	0,76	1,18	1,25
5	1,64	1,55	5,16	1,82
6	0,43	2,73	5,49	1,42
7	0,84	1,75	6,84	1,97

Desta forma ordenando as sequências de acordo com os teoremas 3 e 4, são obtidas as sequências candidatas $\hat{s}$: 2-3-4-1-6-5-7 e $\tilde{s}$: 2-1-3-4-5-6-7. De posse das sequências candidatas, o passo seguinte consiste em aplicar a equação (4.27) para obtenção do custo médio total de cada sequência. Os resultados são apresentados na Tabela 3 e na Tabela 4.

Tabela 3 – Resultados numéricos para a sequência $\hat{s}$

Posição	Tarefa	$\mu_{[i]}$	$\sigma_{[i]}$	Custo Médio
1	2	7,00	1,67	3,22
2	3	15,00	2,51	4,36
3	4	18,60	2,68	0,81
4	1	33,60	4,71	5,14
5	6	36,60	4,73	0,57
6	5	46,60	5,51	3,41
7	7	61,10	6,51	4,56
Total				22,08

Tabela 4 – Resultados numéricos para a sequência $\tilde{s}$

Posição	Tarefa	$\mu_{[i]}$	$\sigma_{[i]}$	Custo Médio
1	2	7,00	1,67	3,22
2	1	10,00	1,73	0,21
3	3	18,00	2,55	4,43
4	4	21,60	2,72	0,82
5	5	31,60	3,92	2,43
6	6	46,60	5,51	6,01
7	7	61,10	6,51	4,56
Total				21,69

Como $\mathcal{C}_{\hat{s}}(d) = 22,08$ e $\mathcal{C}_{\tilde{s}}(d) = 21,69$, conclui-se que a sequência selecionada como solução será $\tilde{s}$. Contudo, uma busca exaustiva por todas as $7! = 5040$ possíveis sequências indica que a sequência ótima é s^* : 1-2-3-4-5-6-7, com $\mathcal{C}_{s^*}(d) = 21,65$, uma diferença de 0,20%. Percebe-se que, para este caso, a Regra SOR não obteve a sequência ótima, contudo foi capaz de fornecer um valor muito próximo do ótimo.

Para uma análise gráfica do resultado obtido, será utilizado o conceito de gráfico de Gantt preditivo, conforme introduzido por Baker e Trietsch (2009a). Este gráfico diferencia-se do gráfico de Gantt tradicional, pois, vista a aleatoriedade dos tempos de processamento e, por conseguinte, das datas de término, o eixo das ordenadas passa a indicar a probabilidade de término de uma tarefa para um dado instante de tempo, conforme ilustrado na Figura 2.

Figura 2 – Gráfico de Gantt preditivo para o exemplo ilustrativo

Cada uma das áreas realçadas está associada à probabilidade de início e término de uma tarefa (identificada pelo círculo numerado). Tome-se, por exemplo, a Figura 3 a seguir a qual destaca a tarefa 2.

Figura 3 – Gráfico de Gantt preditivo para a tarefa 2

Como esta é a primeira tarefa da sequência, seu início já está definido, ocorrendo no instante 0. Dadas suas características (média 7,00 e variância 2,80), pode-se afirmar que há 50% de probabilidade da tarefa 2 terminar no instante 7 (bastando recordar a simetria da distribuição normal). De posse de uma tabela de probabilidade acumulada para a distribuição normal padrão é possível determinar a probabilidade de término para outros instantes. Por exemplo, uma consulta a tal tabela indica que no instante igual a 10 há probabilidade de 96,35% da tarefa 2 estar concluída.

Tome-se agora a tarefa 7. Da caracterização do problema, como as datas de entrega são variáveis a definir, não há necessidade de inserção de tempo de espera. Desta forma a

data de início da Tarefa 7 será igual à data de término daquela que a precede (tarefa 6). Já a data de término será o resultado direto da convolução de seu tempo de processamento e da data de término da tarefa anterior. A aplicação da equação (4.27) garante que seu custo esperado será igual a 4,57, enquanto o Algoritmo 1 fornece uma data de entrega ótima inteira de valor igual a 67, a qual foi destacada na Figura 4.

Figura 4 – Gráfico de Gantt preditivo para a tarefa 7

Verifica-se que para este instante a probabilidade do processamento da tarefa ter-se iniciado é de praticamente 100% enquanto sua probabilidade de término é de aproximadamente 82%. Para a sequência dada, este é o valor de nível de serviço mais seguro (em termos de custos esperados totais) a ser acordado com o cliente. Note que caso haja a necessidade, é possível acordar com o cliente um maior nível de serviço, todavia isto impactará a data de entrega planejada e, por conseguinte, os custos de adiantamento da tarefa em questão.

Por exemplo, caso se deseje ajustar o nível de serviço de entrega da Tarefa 7 para um valor de 95%, isto corresponderia a uma data de entrega igual a 72 dias. Contudo nesta situação o custo da tarefa seria de 5,77 (um aumento de mais de 26%).

5 HEURÍSTICAS PARA O PROBLEMA COM DISTRIBUIÇÃO NORMAL

O objetivo deste capítulo é apresentar heurísticas construtivas capazes de fornecer sequências com custos totais esperados menores que os obtidos por meio da aplicação da Regra SOR. Para tanto será apresentada a heurística conhecida como NEH, a qual serviu de inspiração ao método proposto. A seguir serão introduzidas as heurísticas desenvolvidas.

5.1 Heurística NEH

O algoritmo NEH foi introduzido por Nawaz, Ensore e Ham (1983) para a obtenção de soluções para o problema de programação de tarefas em ambientes de *flowshop* permutacional. Conforme afirmam Kalczynski e Kamburowski (2007), esta é a melhor heurística em termos de performance e qualidade de solução para o problema de minimização de *makespan* (tempo de término da última tarefa processada) em um *flowshop*, o qual pode ser definido como segue.

Considere um conjunto de n tarefas, disponibilizadas simultaneamente em um instante inicial conhecido, as quais devem ser processadas por m máquinas ordenadas, as quais são capazes de processar apenas uma tarefa por vez. Cada tarefa deve ser processada na primeira máquina, a seguir na segunda, até a m -ésima máquina e a sequência de processamento das tarefas deve ser a mesma em cada uma das máquinas. O objetivo é determinar a sequência de tarefas que minimize uma função de custo definida e conhecida.

O algoritmo NEH pode ser descrito como:

Algoritmo 3 (Heurística NEH)

1. Para cada tarefa calcular $P_i = \sum_{j=1}^m p_{i,j}$, onde $p_{i,j}$ é o tempo de processamento da tarefa i na máquina j ;
2. Dispor as tarefas em ordem decrescente de P_i . Denotar esta lista de tarefas como lista candidata;
3. Tomar as duas primeiras tarefas da lista candidata e encontrar a sequência parcial que possui o melhor valor de *makespan* entre as duas sequências possíveis. Não alterar a ordem relativa dessas duas tarefas nos próximos passos. Fazer $i = 3$;
4. Tomar a tarefa na i -ésima posição da lista candidata e encontrar a melhor sequência parcial (em termos de *makespan*) colocando-a em todas as

possíveis i posições da sequência parcial obtida no passo anterior (não alterar a posição relativa das tarefas já fixadas);

5. Parar se $i = n$, do contrário fazer $i = i + 1$ e retornar ao passo 4.

Assumindo que os cálculos de *makespan* do passo 4 necessitam de $m \times n$ operações, o algoritmo NEH terá complexidade $O(mn^3)$. Note que a determinação das sequências candidatas do passo 4 requer $\frac{n(n+1)}{2} - 1$ operações, com complexidade $O(n^2)$ e que a complexidade para o passo 2 (tipicamente $O(n \log n)$) é desprezível frente às demais operações.

Percebe-se que esta heurística possui duas características cruciais para a obtenção de bons resultados: o arranjo da lista candidata inicial (passo 2) e a fase de inserção de tarefas (passo 4). Framinan, Leisten e Rajendran (2003) mostraram que entre 177 diferentes ordenações para a lista candidata, a ordenação apresentada nos passos 1 e 2 é a que apresenta melhores resultados. Já a fase de inserção garante a construção de conjuntos de candidatas com tempo de execução polinomial razoável.

5.2 Heurísticas NEHL

Com base nos pontos fortes da heurística NEH apresentada na seção 5.1, tem-se como objetivo determinar algoritmos que apresentem melhores resultados que os obtidos por meio da aplicação da Regra SOR para o problema de máquina única apresentado no Capítulo 2.

Para tanto será utilizada como lista candidata a sequência $\tilde{s}$ já descrita na Regra SOR, tal que:

$$\left(\frac{v_{[1]}}{\gamma_{[1]}} \leq \frac{v_{[2]}}{\gamma_{[2]}} \leq \dots \leq \frac{v_{[n]}}{\gamma_{[n]}} \right)$$

Esta escolha não é arbitrária. Conforme será apresentado posteriormente, Portugal e Trietsch (2006a) demonstraram que tal sequência é a única assintoticamente ótima para o problema de E/T com tempos estocásticos, i.e., para problemas cada vez maiores ($n \rightarrow \infty$), o custo obtido se aproxima do custo médio total ótimo.

De posse da lista candidata, os passos seguintes consistem na construção de sequências parciais, as quais serão avaliadas continuamente até que se determine a melhor sequência, da mesma forma como a heurística NEH.

Para o primeiro algoritmo (a partir de agora definido como heurística NEHL-1), são selecionadas na lista candidata as duas primeiras tarefas (i.e. as duas tarefas com os menores valores $k_{[i]} = v_{[i]}/\gamma_{[i]}$) para a construção de duas sequências parciais. O custo total esperado de ambas é avaliado e é selecionada aquela que apresentar o menor custo. Após definida a sequência parcial, a posição relativa das tarefas é mantida fixa para os passos seguintes.

A seguir é selecionada a terceira tarefa da lista candidata. A mesma será inserida nas 3 possíveis posições da sequência parcial determinada anteriormente (no início, entre as duas tarefas ou no final da sequência). O custo total esperado para as três sequências é calculado e mais uma vez será selecionada aquela que apresentar o menor custo (da mesma forma a posição relativa das três tarefas será mantida). Este procedimento é repetido até que a lista candidata esteja vazia e a sequência completa seja determinada. O passo a passo do algoritmo é apresentado a seguir.

Algoritmo 4 (heurística NEHL-1)

1. Considere a sequência $[1, 2, \dots, n]$ e para cada tarefa calcular $k_{[i]} = v_{[i]}/\gamma_{[i]}$, onde $v_{[i]}$ é a variância do tempo de processamento da tarefa $[i]$ e $\gamma_{[i]} = (\alpha_{[i]} + \beta_{[i]})\exp(-z_{[i]}^*/2)$;
2. Dispor as tarefas em ordem não-decrescente de $k_{[i]}$. Denotar esta lista de tarefas como lista candidata;
3. Tomar as duas primeiras tarefas da lista candidata e encontrar a sequência parcial que possui o melhor valor de custo total esperado aplicando a equação (4.27) para as duas sequências possíveis. Não alterar a ordem relativa dessas duas tarefas nos próximos passos. Fazer $i = 3$;
4. Tomar a tarefa na i -ésima posição da lista candidata e colocá-la em todas as possíveis i posições da sequência parcial obtida no passo anterior. Aplicar a equação (4.27) e selecionar a sequência parcial de menor custo total esperado (não alterar a posição relativa das tarefas já fixadas);
5. Parar se $i = n$ e denotar a sequência obtida como s^B , do contrário fazer $i = i + 1$ e retornar ao passo 4.

De posse da sequência s^B , basta aplicar a equação (4.21) e o Algoritmo 1 para obtenção das datas de entrega ótimas inteiras.

Baseado no que foi exposto para a heurística NEH, o cálculo de custos do passo 4 possui complexidade $O(n)$, enquanto a fase de inserção demanda $O(n^2)$. Como o passo 2 é tipicamente de complexidade igual a $O(n \log n)$, conclui-se que a heurística NEHL-1 possui complexidade $O(n^3)$. Observa-se que a complexidade da heurística NEHL-1 é maior que a da Regra SOR. Note, contudo, que o principal objetivo é obter algoritmos que apresentem soluções melhores em termos de valores de custo, não tempo de execução, dado que a Regra SOR consiste basicamente de uma regra de despacho, sendo bastante difícil superar sua performance em termos de tempo de execução.

De toda forma, investigou-se um segundo algoritmo o qual fosse capaz de apresentar melhores soluções que a Regra SOR, mas com possibilidade de tempos de execução menores que a heurística NEHL-1. Visto que a fase de inserção é a que demanda maior tempo, decidiu-se inserir um critério de parada para reduzi-la.

O segundo algoritmo (definido a partir de agora como NEHL-2) possui características e passos semelhantes aos da heurística NEHL-1, com a seguinte diferença: na fase de inserção inicia-se a construção das sequências parciais do fim para o começo da sequência e o cálculo de custos é efetuado tão logo seja construída a sequência. Caso a sequência parcial seguinte apresente um custo maior que o da anterior, esta é selecionada e as demais sequências são descartadas. Faz-se necessário um exemplo ilustrativo.

Admita um problema de três tarefas, cuja sequência parcial 1-2 foi definida como de menor custo. Desta forma seriam possíveis três sequências parciais, a saber: 1-2-3, 1-3-2 e 3-2-1. De acordo com o exposto, pela heurística NEHL-2 seria calculado e armazenado o custo da sequência 1-2-3. A seguir seria calculado o custo da sequência 1-3-2 e, caso este seja maior, imediatamente a sequência 1-2-3 é eleita como melhor solução e a sequência 3-2-1 sequer é avaliada. O passo a passo desse algoritmo é apresentado a seguir.

Algoritmo 5 (heurística NEHL-2)

1. Considere a sequência $[1, 2, \dots, n]$ e para cada tarefa calcular $k_{[i]} = v_{[i]} / \gamma_{[i]}$, onde $v_{[i]}$ é a variância do tempo de processamento da tarefa $[i]$ e $\gamma_{[i]} = (\alpha_{[i]} + \beta_{[i]}) \exp(-z_{[i]}^2 / 2)$;
2. Dispor as tarefas em ordem não-decrescente de $k_{[i]}$. Denotar esta lista de tarefas como lista candidata;
3. Tomar as duas primeiras tarefas da lista candidata e encontrar a sequência parcial que possui o melhor valor de custo total esperado aplicando a equação

- (4.27) para as duas sequências possíveis. Não alterar a ordem relativa dessas duas tarefas nos próximos passos. Fazer $i = 3$ e $j = 0$;
4. Tomar a tarefa na i -ésima posição da lista candidata e colocá-la na posição $i - j$ da sequência parcial obtida no passo anterior. Aplicar a equação (4.27) para calcular seu custo. Selecionar como melhor sequência e fazer $j = j + 1$;
 5. Colocar a i -ésima tarefa na posição $i - j$ e calcular o custo dessa sequência parcial por meio da equação (4.27). Caso o custo seja maior que aquele obtido no passo anterior, fazer $j = 0$ e ir para o passo 6 (não alterar a posição relativa das tarefas já fixadas). Caso contrário, fazer $j = j + 1$ e retornar para o passo 4;
 6. Parar se $i = n$ e denotar a sequência obtida como s^B , do contrário fazer $i = i + 1$ e retornar ao passo 4.

Como anteriormente, de posse da sequência s^B é possível aplicar a equação (4.21) e o Algoritmo 1 para obtenção das datas de entrega ótimas inteiras.

Embora a heurística NEHL-2 mantenha uma complexidade de ordem $O(n^3)$, percebe-se que com a inclusão do critério de parada na fase de inserção, o tempo de execução pode variar drasticamente de acordo com os parâmetros do problema. No melhor caso o algoritmo será executado com tempo proporcional a n (quando a primeira sequência parcial construída é a melhor) e no pior caso proporcional a n^3 (quando uma das duas últimas sequências parciais é a melhor). Em outras palavras, no pior caso a heurística NEHL-2 terá tempo de execução idêntico à heurística NEHL-1. Naturalmente a qualidade da solução fornecida é impactada por este novo critério, o que será evidenciado por meio dos experimentos computacionais.

Será agora apresentada em detalhes a aplicação de cada uma das heurísticas para o problema ilustrativo descrito pela Tabela 1. A lista candidata será a sequência $\tilde{s}$ que, conforme mostrado na Seção 4.4, é a sequência 2-1-3-4-5-6-7. Desta forma as duas primeiras sequências parciais avaliadas são: 1-2 e 2-1. Aplicando-se a equação (4.27) em ambas obtêm-se respectivamente custos $C_{12}(d) = 3,39$ e $C_{21}(d) = 3,43$. Desta forma a sequência parcial 1-2 é escolhida.

A seguir, observados os passos da heurística NEHL-1, a inserção da tarefa 3 é avaliada por meio das sequências 3-1-2, 1-3-2 e 1-2-3, respectivamente com custos $C_{312}(d) = 8,40$, $C_{132}(d) = 8,31$ e $C_{123}(d) = 7,82$, concluindo-se que a sequência parcial escolhida será 1-2-3. Este procedimento é repetido até que se obtenha finalmente a sequência completa, a saber, 1-2-3-4-5-6-7. Esta, conforme comentado anteriormente, é também a solução ótima do

problema em questão. A Tabela 5 a seguir apresenta os resultados para cada uma das sequências parciais determinadas ao executar os passos 3 a 5 da heurística NEHL-1.

Tabela 5 – Aplicação da heurística NEHL-1 para o exemplo ilustrativo da Tabela 1

Iteração	Custos das sequências parciais			Sequência selecionada
1	;			1-2
2	;			1-2-3
3	;			1-2-3-4
4	;			1-2-3-4-5
5	;			1-2-3-4-5-6
6	;			1-2-3-4-5-6-7

A Figura 5 a seguir ilustra o gráfico de Gantt preditivo para esta sequência.

Figura 5 – Gráfico de Gantt preditivo para a solução obtida pela heurística NEHL-1

Em relação à Regra SOR, a sequência obtida difere apenas quanto à posição relativa das tarefas 1 e 2, dessa forma os resultados obtidos anteriormente para as demais tarefas se mantêm inalterados. Para comprovar essa assertiva, observe que o tempo de término da segunda tarefa da sequência em ambas as soluções é resultado da convolução dos tempos de processamentos das tarefas 1 e 2. Dessa forma, como as variáveis aleatórias que regem os

tempos de processamento são distribuições normais e independentes, em termos de média e variância, este “bloco” de duas tarefas mantém as mesmas características, não afetando as tarefas posteriores. Essa propriedade do problema será utilizada posteriormente quando da demonstração da otimalidade assintótica das heurísticas NEHL.

Para a heurística NEHL-2, a primeira sequência parcial será idêntica, i.e., 1-2. Para a inserção da tarefa 3, é calculado o custo da sequência 1-2-3. Como visto, o custo da sequência 1-3-2 é maior, assim já se define como melhor sequência 1-2-3. No passo seguinte é avaliada a sequência 1-2-3-4, de custo $C_{1234}(d) = 8,65$. Como a sequência parcial seguinte, 1-2-4-3 apresenta um custo $C_{1243}(d) = 8,72$, já está definida a sequência desta iteração, que é 1-2-3-4. Continuando com a aplicação da heurística, obtém-se como solução a mesma sequência 1-2-3-4-5-6-7, a qual é ótima. Para fins de comparação, note que enquanto a heurística NEHL-2 construiu e comparou 12 sequências parciais (duas sequências em cada uma das iterações do método), a heurística NEHL-1 executou o mesmo procedimento para 27 sequências. A performance de cada uma delas será evidenciada a seguir nos experimentos computacionais.

6 EXPERIMENTOS COMPUTACIONAIS PARA O PROBLEMA COM DISTRIBUIÇÃO NORMAL

Neste capítulo são descritos os conjuntos de problemas utilizados nos experimentos realizados para avaliação das diversas heurísticas apresentadas, assim como os resultados obtidos e as medidas utilizadas para comparação dos métodos.

6.1 Descrição do Experimento

Para uma investigação aprofundada da eficiência de cada um dos métodos apresentados, foram gerados dois diferentes conjuntos de problemas de acordo com a quantidade de tarefas e com a viabilidade da aplicação de um método de enumeração completa para determinar as soluções ótimas.

O primeiro conjunto (a partir de agora denotado como Conjunto 1) é composto por amostras que variam de 5 a 12 tarefas, num total de 680 problemas. O segundo conjunto (denotado como Conjunto 2) é composto por amostras que variam de 13 a 1000 tarefas, perfazendo um total de 480 problemas. Em ambos os casos, médias e variâncias dos tempos de processamento, os quais são variáveis aleatórias normalmente distribuídas, foram aleatoriamente gerados a partir de uma distribuição uniforme discreta entre 5 e 20. Adicionalmente, assim como em Soroush (1999), admitiu-se que o coeficiente de variação dos tempos de processamento não poderia ser maior do que 0,29. Esse limite superior reduz a probabilidade dos tempos de processamento assumirem valores negativos, além de garantir que suas variâncias não sejam exageradamente maiores do que suas médias. Os custos unitários de atraso e adiantamento foram gerados a partir de uma distribuição uniforme discreta variando entre 0 e 20.

Todas as heurísticas foram codificadas em C++ e os experimentos foram executados num computador tipo PC dotado de um processador Intel Core i7 de 2.80 GHz e 8.0 GB de RAM, rodando versão 64-bit do Windows 7 Professional com Service Pack 1.

Para determinar a qualidade das soluções obtidas, adotou-se como um dos parâmetros de comparação o percentual de sobrecusto, definido como a diferença percentual entre os custos totais esperados de uma sequência obtida por meio da aplicação de uma das heurísticas apresentadas e da sequência ótima, ou seja:

$$\%custo = \frac{C_{cand} - C_{opt}}{C_{opt}} \times 100\% \quad (6.1)$$

Onde C_{cand} é o custo médio total da sequência candidata e C_{opt} é o custo médio total associado à sequência ótima.

Uma segunda medida de comparação utilizada foi o percentual de otimalidade, definido como:

$$\%otimal = \frac{\text{no. de soluções ótimas}}{\text{no. total de soluções}} \times 100 \quad (6.2)$$

Ou seja, o percentual de otimalidade é uma medida da frequência com que determinada heurística fornece soluções ótimas.

Como é computacionalmente inviável a determinação de soluções ótimas para problemas maiores que 12 tarefas, as medidas descritas não podem ser aplicadas ao segundo conjunto de problemas. Para comparar as soluções obtidas, foram utilizadas duas medidas: o percentual de melhores soluções e o índice de diferença relativa (IDR).

O percentual de melhores soluções é definido como:

$$\%best = \frac{\text{no. de melhores soluções}}{\text{no. total de soluções}} \times 100 \quad (6.3)$$

Onde o numerador representa a quantidade de soluções que apresentaram uma solução melhor do que aquela obtida pela Regra SOR (excluídos os empates).

Já o IDR pode ser expresso como:

$$IDR = \frac{C_{NEHL} - C_{SOR}}{C_{SOR}} \quad (6.4)$$

Onde C_{SOR} é o custo obtido ao aplicar-se a Regra SOR e C_{NEHL} corresponde ao custo obtido ao aplicar uma das heurísticas propostas (NEHL-1 ou NEHL-2). Note que todos os valores computados são resultado da média do número de instâncias para cada um dos tamanhos de problemas.

É importante ressaltar que foram omitidos os cálculos para determinação das datas de entregas. Tal foi adotado visto que, conforme equação (4.21) (e consequentemente o Algoritmo 1), a qualidade das datas de entrega é função direta da sequência de tarefas, bastando a comparação das sequências obtidas para avaliar a qualidade de cada algoritmo.

6.2 Resultados das Heurísticas NEHL

A Tabela 6 apresenta os resultados obtidos ao aplicar as heurísticas NEHL para o Conjunto 1, comparando-as aos resultados obtidos pela Regra SOR e à busca por enumeração completa. A análise foi efetuada para diferentes tamanhos de problema (Coluna 1), cada um deles composto por um conjunto de diferentes instâncias (Coluna 2). Além das medidas de percentual de otimalidade (Colunas 3, 6 e 9) e percentual de sobrecusto (Colunas 4, 7 e 9) já definidas, é apresentado o tempo de execução médio para cada algoritmo (Colunas 5, 8 e 11). Para fins de comparação, incluiu-se ainda o tempo médio de execução para o método de enumeração completa (Coluna 12).

Percebe-se que a heurística NEHL-1 apresentou um alto percentual de otimalidade, obtendo a solução ótima na quase totalidade dos casos (99,85% das instâncias avaliadas), enquanto a Regra SOR resultou em soluções ótimas em 73,53% dos casos (em 180 das 680 instâncias a sequência obtida não forneceu soluções ótimas). Note que ambos os algoritmos apresentaram baixo percentual de sobrecusto, com vantagem para a heurística NEHL-1. De fato, descartadas as 500 instâncias em que ambos os algoritmos empataram obtendo a solução ótima, a heurística NEHL-1 apresentou melhores resultados nos 180 problemas restantes, e em apenas um deles não obteve a sequência ótima, com um percentual de sobrecusto de 0,02%. Para os tamanhos de problema analisados, a diferença entre os tempos de execução é desprezível. Para a heurística NEHL-2 os resultados obtidos são idênticos aos verificados para a heurística NEHL-1. Até mesmo a quantidade de soluções ótimas é a mesma (679 problemas), contudo para o único problema com solução não-ótima, a heurística NEHL-2 obteve uma sequência com percentual de sobrecusto de 0,10%.

A Tabela 7 apresenta os resultados comparativos considerando-se o Conjunto 2 de instâncias. Mais uma vez a análise foi efetuada para diferentes tamanhos de problema (Coluna 1), cada um deles composto por um conjunto de diferentes instâncias (Coluna 2). São apresentados os tempos de execução de cada algoritmo (Colunas 5, 8 e 12), o índice de melhores soluções (Colunas 3 e 6) e o índice de diferença relativa entre os métodos (Colunas 4 e 7).

Tabela 6 – Comparação entre os algoritmos NEHL e SOR para o Conjunto 1

No. tarefas	No. instâncias	SOR			NEHL-1			NEHL-2			Enumer.
		%otimal	%custo	Tempo (s)	%otimal	%custo	Tempo (s)	%otimal	%custo	Tempo (s)	Tempo (s)
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)
5	100	80,00	0,03	9,30E-04	100,00	0,00	9,30E-04	100,00	0,00	2,00E-04	3,10E-04
6	100	77,00	0,04	9,30E-04	100,00	0,00	1,23E-03	100,00	0,00	2,30E-04	3,27E-03
7	100	77,00	0,03	1,09E-03	100,00	0,00	1,03E-03	100,00	0,00	2,40E-04	0,01
8	100	74,00	0,03	9,30E-04	100,00	0,00	1,09E-03	100,00	0,00	2,70E-04	0,07
9	100	72,00	0,03	1,09E-03	100,00	0,00	1,13E-03	100,00	0,00	3,00E-04	0,73
10	60	65,00	0,03	1,03E-03	100,00	0,00	4,17E-04	100,00	0,00	3,50E-04	7,85
11	60	63,33	0,03	5,17E-04	98,33	<u>0,00</u> ¹	4,67E-04	98,33	<u>0,00</u> ¹	3,83E-04	95,13
12	60	71,67	0,02	3,83E-04	100,00	0,00	4,83E-04	100,00	0,00	3,83E-04	1.249,24
Média		73,53	0,03		99,85	0,00		99,85	0,00		

¹ Para o problema de 11 tarefas as heurísticas não encontraram a solução ótima para um dos problemas (sobrecustos de 0,02% para NEHL-1 e 0,10% para NEHL-2)

Tabela 7 – Comparação entre os algoritmos NEHL e SOR para o Conjunto 2

No. tarefas	No. instâncias	NEHL-1			NEHL-2			SOR
		%best	IDR	Tempo (s)	%best	IDR	Tempo (s)	Tempo (s)
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
13	60	43,33	-1,23E-04	5,67E-04	43,33	-1,13E-04	4,17E-04	1,30E-03
15	60	63,33	-2,81E-04	7,17E-04	63,33	-2,81E-04	5,00E-04	1,30E-03
20	60	66,67	-1,16E-04	1,18E-03	66,67	-1,16E-04	6,33E-04	1,30E-03
50	60	93,33	-3,50E-05	0,01	93,33	-3,50E-05	1,85E-03	9,10E-03
100	60	100,00	-9,42E-06	0,08	100,00	-9,42E-06	4,78E-03	9,87E-03
200	60	100,00	-4,06E-06	0,60	100,00	-4,06E-06	0,01	0,01
500	60	100,00	-9,35E-07	9,31	100,00	-9,19E-07	0,07	0,02
1000	60	100,00	-4,03E-04	74,53	100,00	-4,03E-04	0,27	0,03
Média		83,33	-1,22E-04		83,33	-1,21E-04		

Para o Conjunto 2 percebe-se que ambas as heurísticas apresentaram melhores resultados que a Regra SOR em 83,33% dos casos em média. Em termos absolutos isso significa que as heurísticas NEHL apresentaram melhores soluções em 400 dos 480 problemas avaliados (houve empate para os 80 problemas restantes). Este resultado evidencia a superioridade das heurísticas apresentadas em termos de qualidade de solução, contudo é importante ressaltar que para a heurística NEHL-1 à medida que aumenta o tamanho dos problemas, a diferença entre os tempos de execução cresce sensivelmente. Já para a heurística NEHL-2 fica evidenciado o impacto do critério de parada, o qual torna seu tempo de execução em relação à Regra SOR praticamente desprezível.

Por fim é apresentada a Tabela 8, a qual apresenta uma comparação entre as heurísticas NEHL-1 e NEHL-2 para o segundo conjunto de problemas.

Tabela 8 – Comparação entre os algoritmos NEHL para o Conjunto 2

No. tarefas	No. instâncias	IDR
(1)	(2)	(3)
13	60	-1,08E-05
15	60	0,00E+00
20	60	0,00E+00
50	60	0,00E+00
100	60	0,00E+00
200	60	0,00E+00
500	60	-1,58E-08
1000	60	-2,79E-09
Média		-1,36E-06

As descrições de colunas são idênticas às já apresentadas, exceção feita ao índice de diferença relativa, o qual foi adaptado para:

$$IDR = \frac{C_{NEHL-1} - C_{NEHL-2}}{C_{NEHL-1}} \quad (6.5)$$

A heurística NEHL-2 não forneceu as mesmas soluções que a heurística NEHL-1 em apenas 4 problemas (com uma diferença relativa máxima da ordem de $1,00E-05$). Considerado o tempo de execução, contudo, foi verificada uma grande diferença entre os métodos. Para o problema de 1000 tarefas, por exemplo, o tempo de execução foi reduzido em aproximadamente 280 vezes, evidenciando o impacto do critério de parada da heurística NEHL-2. Considerados tempo de execução e qualidade das soluções fornecidas, pode-se concluir que a heurística NEHL-2 é preferível quando comparada à heurística NEHL-1 e à Regra SOR.

Os resultados apresentados fazem crer que à medida que se aumenta o número de tarefas, o índice de diferença relativa entre os métodos SOR e NEHL decresce. Como comentado, Portugal e Trietsch (2006a) demonstraram que a sequência $\tilde{s}$ é assintoticamente ótima, o que indica que as heurísticas NEHL também o sejam. Esta assertiva será detalhada na próxima seção quando será demonstrada a otimalidade assintótica das heurísticas abordadas.

7 OTIMALIDADE ASSINTÓTICA PARA O PROBLEMA COM DISTRIBUIÇÃO NORMAL

Este capítulo tem por objetivo demonstrar a otimalidade assintótica dos algoritmos apresentados, a qual garante que estes podem ser aplicados para problemas de grande porte sem prejuízo da qualidade das soluções fornecidas. Todos os desenvolvimentos são baseados no trabalho apresentado por Portugal e Triestch (2006a).

7.1 Otimalidade Assintótica da Sequência $\tilde{s}$

A chamada otimalidade assintótica é um recurso utilizado em diversas pesquisas de programação de tarefas (como, por exemplo, visto em Portugal e Triestch (2006a), Portugal e Triestch (2006b), Chou *et al.* (2006) e Baker e Triestch (2009b)). Baker e Triestch (2009a) definem uma heurística como assintoticamente ótima se, caso a quantidade de tarefas n aumente indefinidamente, a diferença relativa entre a solução fornecida pela heurística e a solução ótima se torna desprezível.

Mais formalmente, seja $f(S^*)$ o valor da função objetivo para a sequência ótima S^* , e seja $f(S^H)$ o valor da função objetivo associado a determinada heurística. A heurística é assintoticamente ótima se:

$$\lim_{n \rightarrow \infty} \frac{f(S^*) - f(S^H)}{f(S^*)} = 0 \quad (7.1)$$

Portugal e Triestsch (2006a) demonstraram que a sequência $\tilde{s}$ dada por:

$$\left(\frac{v_{[1]}}{\gamma_{[1]}} \leq \frac{v_{[2]}}{\gamma_{[2]}} \leq \dots \leq \frac{v_{[n]}}{\gamma_{[n]}} \right)$$

é assintoticamente ótima. Para demonstrar tal propriedade, serão apresentados a seguir dois teoremas auxiliares.

Teorema 5. Considere o problema de programação de tarefas definido no Capítulo 4 cuja função objetivo é dada pela equação (4.27) reproduzida a seguir:

$$\mathcal{C}(d^*) = \sum_{i=1}^n \frac{\gamma_{[i]} \sigma_{[i]}}{\sqrt{2\pi}}$$

Onde $\gamma_{[i]} = (\alpha_{[i]} + \beta_{[i]}) \exp(-z_{[i]}^{*2}/2)$. Admita que todas as tarefas possuam variância do tempo de processamento constante, i.e., $v_{[i]} = L^2$, $\forall i = 1, 2, \dots, n$. A sequência ótima é a sequência $\tilde{s}$ tal que:

$$\left(\frac{v_{[1]}}{\gamma_{[1]}} \leq \frac{v_{[2]}}{\gamma_{[2]}} \leq \dots \leq \frac{v_{[n]}}{\gamma_{[n]}} \right)$$

Demonstração. Consiste na aplicação do método de troca de pares adjacentes. Considere uma sequência s qualquer tal que para determinadas posições l e $l+1$, ocupadas respectivamente pelas tarefas p e q , $1/\gamma_{p[l]} > 1/\gamma_{q[l+1]}$ (note que, para fins de demonstração, uma vez mais a notação foi adaptada para identificar as tarefas). Tem-se que:

$$C_s(d) = \sum_{i=1}^n \frac{\gamma_{[i]} L \sqrt{i}}{\sqrt{2\pi}} \quad (7.2)$$

Onde $C_s(d)$ é o custo médio total do problema definido pela equação (4.27), o qual pode ser reescrito como:

$$C_s(d) = \sum_{i=1}^{l-1} \frac{\gamma_{[i]} L \sqrt{i}}{\sqrt{2\pi}} + \frac{\gamma_{p[l]} L \sqrt{l}}{\sqrt{2\pi}} + \frac{\gamma_{q[l+1]} L \sqrt{l+1}}{\sqrt{2\pi}} + \sum_{i=l+2}^n \frac{\gamma_{[i]} L \sqrt{i}}{\sqrt{2\pi}} \quad (7.3)$$

Considere agora a sequência s' , resultado da troca de posições entre as tarefas p e q . Tem-se que:

$$C_{s'}(d) = \sum_{i=1}^{l-1} \frac{\gamma_{[i]} L \sqrt{i}}{\sqrt{2\pi}} + \frac{\gamma_{q[l]} L \sqrt{l}}{\sqrt{2\pi}} + \frac{\gamma_{p[l+1]} L \sqrt{l+1}}{\sqrt{2\pi}} + \sum_{i=l+2}^n \frac{\gamma_{[i]} L \sqrt{i}}{\sqrt{2\pi}} \quad (7.4)$$

Desta forma, subtraindo as equações (7.3) e (7.4), segue:

$$C_s(d) - C_{s'}(d) = \frac{\gamma_{p[l]}L\sqrt{l} - \gamma_{p[l+1]}L\sqrt{l+1}}{\sqrt{2\pi}} + \frac{\gamma_{q[l+1]}L\sqrt{l+1} - \gamma_{q[l]}L\sqrt{l}}{\sqrt{2\pi}} \quad (7.5)$$

Dado que $\gamma_{[i]}$ é independente da posição da tarefa na sequência, pode-se escrever que $\gamma_{p[l]} = \gamma_{p[l+1]} = \gamma_p$ e $\gamma_{q[l]} = \gamma_{q[l+1]} = \gamma_q$, seque que:

$$C_s(d) - C_{s'}(d) = \frac{(\gamma_q - \gamma_p)L(\sqrt{l+1} - \sqrt{l})}{\sqrt{2\pi}} \quad (7.6)$$

Dada a premissa inicial que $1/\gamma_{p[l]} > 1/\gamma_{q[l+1]}$, resulta:

$$C_s(d) - C_{s'}(d) > 0 \quad (7.7)$$

Ou seja, a troca de pares entre as tarefas p e q reduziu o custo médio total. Dessa forma, a sequência que minimiza a função objetivo definida pela equação (4.27) deve ser tal que $1/\gamma_{p[l]} < 1/\gamma_{q[l+1]}$, ou de forma mais geral:

$$\left(\frac{v_{[1]}}{\gamma_{[1]}} \leq \frac{v_{[2]}}{\gamma_{[2]}} \leq \dots \leq \frac{v_{[n]}}{\gamma_{[n]}} \right) \quad (\text{c.q.d.})$$

Para prosseguir, admita agora o mesmo problema caracterizado pela equação (4.27), contudo com um relaxamento o qual permite o *preemption* de tarefas. Em tais condições, é possível criar m partições das tarefas ($m > 1$) de forma tal que a variância do tempo de processamento e o peso γ sejam frações dos valores das tarefas originais e iguais a $f_k v_{[i]}$ e $f_k \gamma_{[i]}$ ($f_k < 1$). Como $\sum_{k=1}^m f_k = 1$, a variância total das partições de qualquer tarefa interrompida será igual a $v_{[i]}$ e o peso total a $\gamma_{[i]}$. Por simplicidade, este problema será denominado de problema relaxado, o qual possui uma importante propriedade dada pelo Teorema 6.

Teorema 6. Seja o problema relaxado definido. A solução ótima é a sequência $\tilde{s}$ tal que:

$$\left(\frac{v_{[1]}}{\gamma_{[1]}} \leq \frac{v_{[2]}}{\gamma_{[2]}} \leq \dots \leq \frac{v_{[n]}}{\gamma_{[n]}} \right)$$

Demonstração. Sem perda de generalidade, admita que todas as tarefas sejam particionadas de forma tal que todas as frações possuam a mesma variância. Pelo Teorema 5, as frações serão alocadas na sequência em ordem crescente da razão $v_{[i]}/\gamma_{[i]}$. Contudo, por construção, cada uma das frações de uma tarefa $[i]$ possui a mesma razão $v_{[i]}/\gamma_{[i]}$ da tarefa original. Assim sendo, a sequência que otimiza o problema relaxado é tal que:

$$\left(\frac{v_{[1]}}{\gamma_{[1]}} \leq \frac{v_{[2]}}{\gamma_{[2]}} \leq \dots \leq \frac{v_{[n]}}{\gamma_{[n]}} \right) \quad (\text{c.q.d.})$$

Interessante notar que a solução do problema relaxado não requer *preemption* das tarefas.

Claramente o problema relaxado apresentará custo médio total menor ou igual ao problema original. Basta atentar que para uma tarefa particionada, o custo da k -ésima fração será dado por:

$$C_k = \frac{1}{\sqrt{2\pi}} f_k \gamma_{[i]} \sqrt{\sum_{j=1}^k f_j \sigma_{[i]}^2} \quad (7.8)$$

Dessa forma, considerando que a tarefa foi particionada em m partes, seu custo no problema relaxado será:

$$C_{[i]} = \sum_{k=1}^m \frac{1}{\sqrt{2\pi}} f_k \gamma_{[i]} \sqrt{\sum_{j=1}^k f_j \sigma_{[i]}^2} \quad (7.9)$$

Esse valor será sempre menor que $\gamma_{[i]} \sigma_{[i]} / \sqrt{2\pi}$, dado que $f_k < 1$. Assim, percebe-se que a solução do problema relaxado consiste num limitante inferior do problema original. Este fato auxiliará na demonstração do Teorema 7, o qual atesta a otimalidade assintótica da sequência $\tilde{s}$.

Teorema 7. Considere o problema de programação de tarefas cuja função objetivo é definida pela equação (4.27). Seja f_n^{SOR} o valor dessa função objetivo do problema com n tarefas ao

aplicar a sequência $\tilde{s}$ e seja F_n a solução ótima. Desse modo, para $n \rightarrow \infty$, tem-se que $f_n^{SOR}/F_n \rightarrow 1$.

Demonstração. Dados um limitante superior e um limitante inferior para o problema definido pela equação (4.27) baseados na sequência $\tilde{s}$, é suficiente mostrar que a diferença entre esses dois limitantes é desprezível à medida que se aumenta o número de tarefas (i.e., a contribuição de uma nova tarefa para a função objetivo é desprezível quando n é grande).

Seja σ_0^2 a variância do tempo de término do conjunto de tarefas anterior à tarefa $[i]$ (para a primeira tarefa faça $\sigma_0^2 = 0$). Seja $\sigma_{[i]}^2 = \sigma_0^2 + v_{[i]}$ a variância do conjunto citado somado à tarefa $[i]$. Um limitante superior do problema é a própria sequência $\tilde{s}$ (ainda que esta não seja necessariamente uma solução ótima), cuja contribuição marginal de uma tarefa $[i]$ é dada por $C_{[i]} = \gamma_{[i]} \sigma_{[i]} / \sqrt{2\pi}$. Conforme comentado, um limitante inferior é o problema relaxado tratado pelo Teorema 6. Lembrando que $\sigma_{[i]}^2 = \sigma_0^2 + v_{[i]}$, a contribuição marginal $C_{[i]}^{rel}$ da tarefa $[i]$ no problema relaxado será dada por:

$$C_{[i]}^{rel} = \sum_{k=1}^m \frac{1}{\sqrt{2\pi}} f_k \gamma_{[i]} \sqrt{\sigma_0^2 + \sum_{j=1}^k f_j v_{[i]}} \quad (7.10)$$

Sem perda de generalidade, admita que esta tarefa $[i]$ é particionada em frações infinitesimais. Nessas condições, $m \rightarrow \infty$, $f_k \rightarrow 0$ e a equação (7.10) se reduz a:

$$C_{[i]}^{rel} = \frac{1}{\sqrt{2\pi}} \int_{w=0}^{w=\gamma_{[i]}} \int_{x=0}^{x=1} \sqrt{\sigma_0^2 + xv_{[i]}} dx dw$$

Como w e x são independentes, é possível escrever que:

$$C_{[i]}^{rel} = \frac{1}{\sqrt{2\pi}} \int_{w=0}^{\gamma_{[i]}} dw \int_{x=0}^1 \sqrt{\sigma_0^2 + xv_{[i]}} dx$$

Resolvendo a integral em w e tomando $\sigma_0^2 + xv_{[i]} = y^2$, segue:

$$C_{[i]}^{rel} = \frac{2}{\sqrt{2\pi}} \frac{\gamma_{[i]}}{v_{[i]}} \int_{y=\sigma_{[0]}}^{\sigma_{[i]}} y^2 dy$$

Finalmente:

$$C_{[i]}^{rel} = \frac{2}{3\sqrt{2\pi}} \frac{\gamma_{[i]}}{v_{[i]}} (\sigma_{[i]}^3 - \sigma_{[0]}^3) \quad (7.11)$$

Denote por Δ a diferença relativa entre os custos dos limitantes definidos, segue então que:

$$\begin{aligned} \Delta &= \frac{C_{[i]} - C_{[i]}^{rel}}{C_{[i]}} = \frac{1}{\gamma_{[i]}\sigma_{[i]}/\sqrt{2\pi}} \left[\gamma_{[i]}\sigma_{[i]}/\sqrt{2\pi} - \frac{2}{3\sqrt{2\pi}} \frac{\gamma_{[i]}}{v_{[i]}} (\sigma_{[i]}^3 - \sigma_0^3) \right] \\ &= \frac{1}{\sigma_{[i]}} \left[\sigma_{[i]} - \frac{2}{3} \frac{(\sigma_{[i]} - \sigma_0)(\sigma_{[i]}^2 + \sigma_0^2 + \sigma_{[i]}\sigma_0)}{(\sigma_{[i]} - \sigma_0)(\sigma_{[i]} + \sigma_0)} \right] \\ &= \frac{1}{3\sigma_{[i]}} \left[\frac{3\sigma_{[i]}(\sigma_{[i]} + \sigma_0) - 2(\sigma_{[i]}^2 + \sigma_0^2 + \sigma_{[i]}\sigma_0)}{(\sigma_{[i]} + \sigma_0)} \right] \\ &= \frac{1}{3\sigma_{[i]}} \left[\frac{3(\sigma_{[i]}^2 + \sigma_{[i]}\sigma_0) - 2(\sigma_{[i]}^2 + \sigma_0^2 + \sigma_{[i]}\sigma_0)}{(\sigma_{[i]} + \sigma_0)} \right] \\ &= \frac{1}{3\sigma_{[i]}} \left[\frac{\sigma_{[i]}^2 - \sigma_0^2 + \sigma_{[i]}\sigma_0 - \sigma_0^2}{(\sigma_{[i]} + \sigma_0)} \right] \\ &= \frac{1}{3\sigma_{[i]}} \left[\frac{v_{[i]} + \sigma_{[i]}\sigma_0 + v_{[i]} - \sigma_{[i]}^2}{(\sigma_{[i]} + \sigma_0)} \right] \\ \Delta &= \frac{1}{3\sigma_{[i]}} \left[\frac{2v_{[i]} - \sigma_{[i]}(\sigma_{[i]} - \sigma_0)}{(\sigma_{[i]} + \sigma_0)} \right] \quad (7.12) \end{aligned}$$

Da equação (7.12), é possível perceber que para $i > 1$, $\Delta < 2v_{[i]}/3\sigma_0^2$ (note que para $i = 1$ é imediato que $\Delta = 1/3$). Como $v_{[i]}$ independe da quantidade de tarefas e claramente $\sigma_0^2 \rightarrow \infty$ para $n \rightarrow \infty$, conclui-se que:

$$\lim_{n \rightarrow \infty} \Delta = 0 \quad (7.13)$$

Ou ainda, consequentemente:

$$\lim_{n \rightarrow \infty} \frac{C_{[i]}^{rel}}{C_{[i]}} = 1 \quad (7.14)$$

Denote agora $f_n^{SOR_{rel}}$ como o valor da função objetivo do problema relaxado de n tarefas ao aplicar-se a sequência $\tilde{s}$. Da razão definida em (7.14), pode-se afirmar que:

$$\lim_{n \rightarrow \infty} \frac{f_n^{SOR_{rel}}}{f_n^{SOR}} = 1 \quad (7.15)$$

Sabe-se que:

$$f_n^{SOR_{rel}} \leq F_n \leq f_n^{SOR} \quad (7.16)$$

O que pode ser reescrito como:

$$1 \leq \frac{f_n^{SOR}}{F_n} \leq \frac{f_n^{SOR}}{f_n^{SOR_{rel}}} \quad (7.17)$$

Pelo teorema do confronto e visto o resultado definido em (7.15), conclui-se que:

$$\lim_{n \rightarrow \infty} \frac{f_n^{SOR}}{F_n} = 1 \quad (7.18)$$

(c.q.d.)

7.2 Otimalidade Assintótica das Heurísticas NEHL

Considere a sequência $S^S = \{[1], [2] \dots [i] \dots [n]\}$ tal que $v_{[1]}/\gamma_{[1]} \leq v_{[2]}/\gamma_{[2]} \leq \dots \leq v_{[n]}/\gamma_{[n]}$, i.e, a sequência $\tilde{s}$ da Regra SOR. Embora as heurísticas NEHL tenham sido inicialmente apresentadas como um método construtivo, estas podem ser consideradas como um conjunto de trocas de pares adjacentes em uma sequência completa de forma tal que apenas movimentos para a esquerda sejam permitidos. Por exemplo, dada uma sequência 1-2-3-4 e assumindo que a posição relativa das três primeiras tarefas já foi fixada num passo anterior, as possíveis candidatas seriam 1-2-3-4, 1-2-4-3, 1-4-2-3 e 4-1-2-3.

Seja NEHL-1* o algoritmo NEHL-1 adaptado a essa nova visão. Este se inicia tomando a sequência S^S como candidata e efetuando uma troca à esquerda da segunda tarefa, gerando uma segunda sequência. Como comentado no Capítulo 5, esta troca não afetará os custos da terceira tarefa em diante. Para avaliar ambas as sequências é suficiente a comparação dos custos esperados do “bloco” de duas tarefas em cada uma delas. Aquela com “bloco” de menor custo será denotada como sequência candidata.

No passo seguinte será tomada a sequência candidata do passo anterior e apenas trocas da terceira tarefa à esquerda serão permitidas. Mais uma vez a comparação das três possíveis sequências pode ser realizada avaliando-se somente os “blocos” de três tarefas de cada uma. A sequência com o “bloco” de menor custo será denotada como candidata e o procedimento será repetido até a n -ésima tarefa. Ao final do procedimento, terão sido geradas e avaliadas $n(n+1)/2$ diferentes sequências e a solução será aquela de menor custo dentre elas.

Como a sequência S^S é utilizada como ponto de partida, por construção o algoritmo NEHL-1* irá sempre fornecer soluções com custo menor ou igual ao dessa sequência. Como é clara a equivalência entre o procedimento de construção de sequências parciais e o de trocas de pares à esquerda de uma sequência completa, pode-se então concluir que a heurística NEHL-1 irá fornecer soluções de custo menor ou igual ao da sequência S^S . Argumento semelhante pode ser utilizado para provar que a heurística NEHL-2 sempre fornecerá soluções de custo menor ou igual ao da sequência S^S .

Para mostrar a otimalidade assintótica das heurísticas NEHL, considere f_n^{NEHL} o valor da função objetivo ao aplicar o Algoritmo 4 ou o Algoritmo 5 para o problema de n tarefas. Do exposto, é possível afirmar que $f_n^{NEHL} \leq f_n^{SOR}$ para qualquer n . Denote F_n como a solução ótima do problema, então é possível escrever que:

$$1 \leq \frac{f_n^{NEHL}}{F_n} \leq \frac{f_n^{SOR}}{F_n} \quad (7.19)$$

Pelo Teorema 7, $\lim_{n \rightarrow \infty} \frac{f_n^{SOR}}{F_n} = 1$. Assim sendo pelo teorema do confronto, resulta:

$$\lim_{n \rightarrow \infty} \frac{f_n^{NEHL}}{F_n} = 1 \quad (7.20)$$

Ou seja, ambas as heurísticas NEHL são assintoticamente ótimas.

8 FORMULAÇÃO DO PROBLEMA COM DISTRIBUIÇÃO LOGNORMAL E DESENVOLVIMENTO MATEMÁTICO

Este capítulo é dedicado à formulação matemática do problema considerando-se que os tempos de processamento são caracterizados por uma distribuição lognormal. Esta é uma distribuição cujas características permitem sua aplicação em diferentes áreas do conhecimento, desde redes de comunicação sem fio (Stüber (2002) comenta sobre diversos desses estudos), como também em aplicações no mercado financeiro (sendo o mais conhecido o estudo sobre precificação de opções financeiras apresentado por Black e Scholes (1973)).

A prática mostra que a duração de tarefas não assume valores negativos e estas apresentam a tendência de finalizar bem acima da sua média (i.e., sua distribuição de probabilidade apresenta uma obliquidade para a direita). Dadas tais condições, a distribuição lognormal é candidata natural para esta modelagem, opinião esta defendida por diversos autores (Trietsch *et al.* (2012) apresentam uma lista bastante completa dos defensores do uso da lognormal). Um ponto de atenção quanto ao uso desta distribuição refere-se à soma de variáveis lognormais. Diferentemente da distribuição normal, a qual encerra uma expressão fechada para a convolução de duas variáveis normais, o mesmo não ocorre para a distribuição lognormal. Será visto que, ainda que aproximadamente, a chamada versão lognormal do Teorema Central do Limite garante a sua aplicação. No decorrer do capítulo serão apresentadas suas propriedades, assim como desenvolvimentos específicos para o problema em estudo.

8.1 Formulação Matemática

A distribuição lognormal é aquela cujo logaritmo é normalmente distribuído, ou seja, seja Y uma variável aleatória lognormal, segue que $X = \ln Y$ é uma variável normal. Esta é caracterizada pela seguinte função densidade de probabilidade:

$$f_{[i]}(t) = \begin{cases} \frac{1}{t\sqrt{2\pi v_{[i]}}} e^{-\frac{(\ln t - m_{[i]})^2}{2v_{[i]}}}, & t > 0 \\ 0, & t \leq 0 \end{cases} \quad (8.1)$$

Onde $m_{[i]}$ e $v_{[i]}$ são os parâmetros da distribuição normal associada. Da definição de média e variância é possível demonstrar que:

$$M_{[i]} = e^{m_{[i]} + \frac{v_{[i]}}{2}} \quad (8.2)$$

$$V_{[i]} = (e^{v_{[i]}} - 1)e^{2m_{[i]} + v_{[i]}} = (e^{v_{[i]}} - 1)M_{[i]}^2 \quad (8.3)$$

Onde $M_{[i]}$ e $V_{[i]}$ são respectivamente a média e a variância da distribuição lognormal.

Sejam os parâmetros e variáveis definidos no Capítulo 4. Todas as definições se mantêm, todavia $f_{[i]}(Y)$ agora corresponde à função densidade de probabilidade da variável lognormal Y associada ao tempo de processamento da tarefa $[i]$. Importante notar que $m_{[i]}$ e $v_{[i]}$ neste caso não possuem uma interpretação física, correspondendo unicamente à média e à variância da distribuição normal correspondente (a média e a variância dos tempos de processamento propriamente ditos são dadas pelas equações (8.2) e (8.3)).

Assim como no caso normal, o tempo de término da tarefa $[i]$ corresponderá à convolução de seu tempo de processamento com os tempos de processamento das tarefas que a precedem. Por exemplo, para $i = 2$, tem-se que:

$$t_{[2]} = p_{[1]} + p_{[2]}$$

Ou seja:

$$t_{[2]}(y) = f_{[1]} * f_{[2]} = \int_{-\infty}^{\infty} f_{[1]}(y - t) f_{[2]}(t) dt$$

Da expressão da função densidade de probabilidade, resulta:

$$t_{[2]}(y) = \int_{-\infty}^{\infty} \frac{1}{(y - t)\sqrt{2\pi v_{[1]}}} e^{-\frac{[\ln(y-t) - m_{[1]}]^2}{2v_{[1]}}} \frac{1}{t\sqrt{2\pi v_{[2]}}} e^{-\frac{(\ln t - m_{[2]})^2}{2v_{[2]}}} dt \quad (8.4)$$

A integral dada pela equação (8.4) não é redutível a uma expressão simplificada. Desta forma duas abordagens são possíveis para sua resolução: numericamente por meio de simulações (e.g. Monte Carlo) ou aproximações algébricas.

Entre as diversas aproximações, destaca-se o método de Fenton-Wilkinson (Fenton, 1960), o qual é largamente utilizado para o cálculo de probabilidade de interrupções em redes celulares de comunicação (Stüber, 2002). Seu princípio básico consiste em definir parâmetros

m_z, v_z (respectivamente a média e a variância da distribuição normal associada) de forma a igualar suas média e variância àquelas da soma das duas variáveis lognormais.

Sejam W e Y duas variáveis lognormais com parâmetros m_w, v_w e m_y, v_y , respectivamente. A aplicação do método de Fenton-Wilkinson indica que a variável $Z = W + Y$, é uma variável lognormal caracterizada por parâmetros m_z, v_z tais que:

$$v_z = \ln \left[\frac{e^{2m_w+v_w}(e^{v_w} - 1) + e^{2m_y+v_y}(e^{v_y} - 1)}{\left(e^{m_w+\frac{v_w}{2}} + e^{m_y+\frac{v_y}{2}}\right)^2} + 1 \right] \quad (8.5)$$

$$m_z = \ln \left(e^{m_w+\frac{v_w}{2}} + e^{m_y+\frac{v_y}{2}} \right) - \frac{v_z}{2} \quad (8.6)$$

Schwartz e Yeh (1982) afirmam que este método apresenta boa acurácia para $v_w, v_y \leq 1$. Conforme será visto nos experimentos computacionais, tais valores de variância são suficientes para o problema em questão, contudo caso v_z assuma um valor maior do que a unidade, o método se tornaria inviável para problemas com mais de duas tarefas. A Propriedade 1 a seguir garante o uso do método de Fenton-Wilkinson para qualquer número de tarefas.

Propriedade 1. Considere duas variáveis lognormais W e Y com parâmetros m_w, v_w e m_y, v_y , respectivamente, e $Z = W + Y$. Deste modo, se $v_w, v_y \leq 1$, então a variância v_z resultante da aplicação do método de Fenton-Wilkinson será sempre menor ou igual a 1.

Demonstração. Dados dois números reais positivos w e y quaisquer, é válido o seguinte:

$$(w + y)^2 \geq w^2 + y^2 \rightarrow \frac{1}{(w + y)^2} \leq \frac{1}{w^2 + y^2}$$

Com base nesta relação e na equação (8.5) é possível escrever que:

$$v_z \leq \ln \left[\frac{e^{2m_w+v_w}(e^{v_w} - 1) + e^{2m_y+v_y}(e^{v_y} - 1)}{e^{2m_w+v_w} + e^{2m_y+v_y}} + 1 \right]$$

Como $v_w, v_y \leq 1$, segue que:

$$v_z \leq \ln \left[\frac{(e^{2m_w+v_w} + e^{2m_y+v_y})(e-1)}{e^{2m_w+v_w} + e^{2m_y+v_y}} + 1 \right]$$

Ou seja:

$$v_z \leq \ln[e] \Rightarrow v_z \leq 1$$

(c.q.d.)

A aplicação do método de Fenton-Wilkinson (ou mesmo do método de Monte Carlo) pressupõe que uma soma de variáveis lognormais será aproximadamente uma variável lognormal, independentemente de sua quantidade. Tal assertiva é garantida pela versão lognormal do Teorema Central do Limite, o qual pode ser enunciado como segue.

Teorema Central do Limite (Lognormal). Seja Z uma variável aleatória resultante da convolução de n variáveis aleatórias independentes, não-negativas e contínuas, com médias positivas e coeficiente de variação finito. Dessa forma, para n suficientemente grande, Z tende a ser uma variável lognormal.

Demonstração. Ver Baker e Triestch (2010).

Similarmente ao caso com distribuição normal (desde que atendidas as condições do teorema), é possível utilizar os resultados obtidos para o presente caso para outros tipos de distribuição de probabilidade, desde que o problema possua uma quantidade razoável de tarefas.

Definido o método de tratamento da soma das tarefas, é possível seguir os mesmos passos desenvolvidos anteriormente no Capítulo 4. Assim como no caso com distribuição normal, os custos de atraso e adiantamento serão dados por:

$$\mathcal{E}(d_{[i]}) = \alpha_{[i]} \int_{-\infty}^{d_{[i]}} (d_{[i]} - t_{[i]}) f_{[i]}(t) dt \quad (8.7)$$

$$\mathcal{T}(d_{[i]}) = \beta_{[i]} \int_{d_{[i]}}^{+\infty} (t_{[i]} - d_{[i]}) f_{[i]}(t) dt \quad (8.8)$$

E, similarmente, o custo total esperado é dado por:

$$\mathcal{C}(d) = \sum_{i=1}^n \mathcal{E}(d_{[i]}) + \mathcal{T}(d_{[i]}) \quad (8.9)$$

Onde d é o vetor composto pelas datas de entrega $d_{[i]}$, ou seja, $d = (d_{[1]}, d_{[2]}, d_{[3]}, \dots, d_{[n]})$.

Considerando que os tempos de processamento são definidos por uma distribuição lognormal, é possível enunciar os Teoremas 8 e 9 a seguir.

Teorema 8. Para o problema com distribuição lognormal, o valor esperado do adiantamento $\mathcal{E}(d_{[i]})$ da tarefa $[i]$ da sequência é dado por:

$$\mathcal{E}(d_{[i]}) = \alpha_{[i]} \left[d_{[i]} \Phi(z_{[i]}) - e^{\mu_{[i]} + \frac{\sigma_{[i]}^2}{2}} \Phi(z_{[i]} - \sigma_{[i]}) \right] \quad (8.10)$$

Onde $z_{[i]} = (\ln d_{[i]} - \mu_{[i]}) / \sigma_{[i]}$ é uma função ainda desconhecida e dependente do tempo de término $d_{[i]}$ e $\phi(X)$ e $\Phi(X)$ são respectivamente a função densidade de probabilidade e a função probabilidade cumulativa para uma variável aleatória normal X padronizada.

Demonstração. Considerando uma distribuição lognormal, a equação (8.7) pode ser escrita como:

$$\mathcal{E}(d_{[i]}) = \alpha_{[i]} \int_{-\infty}^{d_{[i]}} (d_{[i]} - t_{[i]}) \frac{1}{t_{[i]} \sqrt{2\pi} \sigma_{[i]}} e^{-\frac{(\ln t_{[i]} - \mu_{[i]})^2}{2\sigma_{[i]}^2}} dt$$

Ou ainda:

$$\mathcal{E}(d_{[i]}) = \alpha_{[i]} \left[\int_{-\infty}^{d_{[i]}} d_{[i]} \frac{1}{t_{[i]} \sqrt{2\pi} \sigma_{[i]}} e^{-\frac{(\ln t_{[i]} - \mu_{[i]})^2}{2\sigma_{[i]}^2}} dt - \int_{-\infty}^{d_{[i]}} \frac{1}{\sqrt{2\pi} \sigma_{[i]}} e^{-\frac{(\ln t_{[i]} - \mu_{[i]})^2}{2\sigma_{[i]}^2}} dt \right]$$

Efetuando as substituições $z_{[i]} = (\ln d_{[i]} - \mu_{[i]}) / \sigma_{[i]}$ e $x = (\ln t_{[i]} - \mu_{[i]}) / \sigma_{[i]}$ (o que é equivalente a $t_{[i]} = e^{\mu_{[i]} + x\sigma_{[i]}}$), é possível escrever que:

$$\begin{aligned}\mathcal{E}(d_{[i]}) &= \alpha_{[i]} \left[\int_{-\infty}^{z_{[i]}} d_{[i]} \frac{1}{\sqrt{2\pi}\sigma_{[i]}} e^{-\frac{x^2}{2}} dx - \int_{-\infty}^{z_{[i]}} \frac{1}{\sqrt{2\pi}} e^{\mu_{[i]} + x\sigma_{[i]}} e^{-\frac{x^2}{2}} dx \right] \\ &= \alpha_{[i]} \left[\int_{-\infty}^{z_{[i]}} d_{[i]} \frac{1}{\sqrt{2\pi}\sigma_{[i]}} e^{-\frac{x^2}{2}} dx - \int_{-\infty}^{z_{[i]}} \frac{1}{\sqrt{2\pi}} e^{\mu_{[i]} + \frac{\sigma_{[i]}^2}{2}} e^{-\frac{(x-\sigma_{[i]})^2}{2}} dx \right]\end{aligned}$$

Ou seja:

$$\mathcal{E}(d_{[i]}) = \alpha_{[i]} \left[d_{[i]} \Phi(z_{[i]}) - e^{\mu_{[i]} + \frac{\sigma_{[i]}^2}{2}} \Phi(z_{[i]} - \sigma_{[i]}) \right]$$

(c.q.d.)

Teorema 9. Para o problema com distribuição lognormal, o valor esperado do atraso $\mathcal{T}(d_{[i]})$ da tarefa $[i]$ da sequência é dado por:

$$\mathcal{T}(d_{[i]}) = \beta_{[i]} \left\{ e^{\mu_{[i]} + \frac{\sigma_{[i]}^2}{2}} [1 - \Phi(z_{[i]} - \sigma_{[i]})] - d_{[i]} [1 - \Phi(z_{[i]})] \right\} \quad (8.11)$$

Demonstração. Idêntica à demonstração da equação (8.10), motivo pelo qual será omitida.

Substituindo as equações (8.10) e (8.11) na equação (8.9), segue que o custo médio total será dado por:

$$\begin{aligned}\mathcal{C}(d) &= \sum_{i=1}^n \alpha_{[i]} \left[d_{[i]} \Phi(z_{[i]}) - e^{\mu_{[i]} + \frac{\sigma_{[i]}^2}{2}} \Phi(z_{[i]} - \sigma_{[i]}) \right] \\ &\quad + \beta_{[i]} \left\{ e^{\mu_{[i]} + \frac{\sigma_{[i]}^2}{2}} [1 - \Phi(z_{[i]} - \sigma_{[i]})] - d_{[i]} [1 - \Phi(z_{[i]})] \right\}\end{aligned} \quad (8.12)$$

Onde $d = (d_{[1]}, d_{[2]}, d_{[3]}, \dots, d_{[n]})$ e $z_{[i]} = (\ln d_{[i]} - \mu_{[i]}) / \sigma_{[i]}$. Como o custo médio total é função da média e do desvio do tempo de término de cada tarefa (respectivamente $\mu_{[i]}$ e $\sigma_{[i]}$), este é dependente da sequência.

8.2 Datas de entrega ótimas

O procedimento é idêntico ao efetuado para o problema com distribuição normal, i.e., será considerado que o conjunto de tarefas do problema foi previamente sequenciado e serão definidas as datas de entrega que otimizam o custo total $\mathcal{C}(d)$.

A primeira derivada da equação (8.12) em relação ao vetor de datas de entrega d é dada por:

$$\begin{aligned} \frac{\partial \mathcal{C}(d)}{\partial d_{[i]}} = & \alpha_{[i]} \left[\Phi(z_{[i]}) + d_{[i]} \frac{\partial \Phi(z_{[i]})}{\partial d_{[i]}} - e^{\mu_{[i]} + \frac{\sigma_{[i]}^2}{2}} \frac{\partial \Phi(z_{[i]} - \sigma_{[i]})}{\partial d_{[i]}} \right] \\ & + \beta_{[i]} \left\{ -e^{\mu_{[i]} + \frac{\sigma_{[i]}^2}{2}} \left[\frac{\partial \Phi(z_{[i]} - \sigma_{[i]})}{\partial d_{[i]}} \right] - 1 + \Phi(z_{[i]}) + d_{[i]} \frac{\partial \Phi(z_{[i]})}{\partial d_{[i]}} \right\} \end{aligned} \quad (8.13)$$

Pela regra da cadeia, tem-se que a derivada parcial dependente da função $\Phi(z_{[i]})$ pode ser escrita como:

$$\frac{\partial \Phi(z_{[i]})}{\partial d_{[i]}} = \frac{\partial \Phi(z_{[i]})}{\partial z_{[i]}} \frac{\partial z_{[i]}}{\partial d_{[i]}}$$

Dessa forma:

$$\begin{aligned} \frac{\partial \Phi(z_{[i]})}{\partial d_{[i]}} = & \frac{\partial}{\partial z_{[i]}} \left(\int_{-\infty}^{z_{[i]}} d_{[i]} \frac{1}{\sqrt{2\pi}\sigma_{[i]}} e^{-\frac{x^2}{2}} dx \right) \frac{\partial z_{[i]}}{\partial d_{[i]}} \\ \frac{\partial \Phi(z_{[i]})}{\partial d_{[i]}} = & \frac{\phi(z_{[i]})}{\sigma_{[i]} d_{[i]}} \end{aligned} \quad (8.14)$$

Substituindo a equação (8.14) em (8.13), resulta:

$$\begin{aligned} \frac{\partial \mathcal{C}(d)}{\partial d_{[i]}} = \alpha_{[i]} & \left[\Phi(z_{[i]}) + d_{[i]} \frac{\phi(z_{[i]})}{\sigma_{[i]} d_{[i]}} - e^{\mu_{[i]} + \frac{\sigma_{[i]}^2}{2}} \frac{\phi(z_{[i]} - \sigma_{[i]})}{\sigma_{[i]} d_{[i]}} \right] \\ & + \beta_{[i]} \left\{ -e^{\mu_{[i]} + \frac{\sigma_{[i]}^2}{2}} \frac{\phi(z_{[i]} - \sigma_{[i]})}{\sigma_{[i]} d_{[i]}} - 1 + \Phi(z_{[i]}) + d_{[i]} \frac{\phi(z_{[i]})}{\sigma_{[i]} d_{[i]}} \right\} \end{aligned}$$

$$\frac{\partial \mathcal{C}(d)}{\partial d_{[i]}} = (\alpha_{[i]} + \beta_{[i]}) \left[\Phi(z_{[i]}) + \frac{\phi(z_{[i]})}{\sigma_{[i]}} - e^{\mu_{[i]} + \frac{\sigma_{[i]}^2}{2}} \frac{\phi(z_{[i]} - \sigma_{[i]})}{\sigma_{[i]} d_{[i]}} \right] - \beta_{[i]}$$

Como $z_{[i]} = (\ln d_{[i]} - \mu_{[i]}) / \sigma_{[i]}$, i.e., $d_{[i]} = e^{\mu_{[i]} + z_{[i]} \sigma_{[i]}}$, segue que:

$$\begin{aligned} \frac{\partial \mathcal{C}(d)}{\partial d_{[i]}} &= (\alpha_{[i]} + \beta_{[i]}) \left[\Phi(z_{[i]}) + \frac{\phi(z_{[i]})}{\sigma_{[i]}} - \frac{e^{\mu_{[i]} + \frac{\sigma_{[i]}^2}{2}}}{\sigma_{[i]}} e^{-\mu_{[i]} - z_{[i]} \sigma_{[i]}} \frac{1}{\sqrt{2\pi}} e^{\frac{-(z_{[i]} - \sigma_{[i]})^2}{2}} \right] - \beta_{[i]} \\ &= (\alpha_{[i]} + \beta_{[i]}) \left[\Phi(z_{[i]}) + \frac{\phi(z_{[i]})}{\sigma_{[i]}} - \frac{e^{\frac{\sigma_{[i]}^2}{2}}}{\sigma_{[i]}} e^{-z_{[i]} \sigma_{[i]}} \frac{1}{\sqrt{2\pi}} e^{\frac{-z_{[i]}^2 - \sigma_{[i]}^2}{2} + z_{[i]} \sigma_{[i]}} \right] - \beta_{[i]} \\ &= (\alpha_{[i]} + \beta_{[i]}) \left[\Phi(z_{[i]}) + \frac{\phi(z_{[i]})}{\sigma_{[i]}} - \frac{1}{\sigma_{[i]}} \frac{1}{\sqrt{2\pi}} e^{\frac{-z_{[i]}^2}{2}} \right] - \beta_{[i]} \end{aligned}$$

Ou seja:

$$\frac{\partial \mathcal{C}(d)}{\partial d_{[i]}} = (\alpha_{[i]} + \beta_{[i]}) \Phi(z_{[i]}) - \beta_{[i]} \quad (8.15)$$

Que é exatamente a mesma expressão obtida anteriormente para o problema com distribuição normal. Uma solução para esta expressão pode ser obtida igualando-a a zero, ou seja:

$$\frac{\partial \mathcal{C}(d)}{\partial d_{[i]}} = (\alpha_{[i]} + \beta_{[i]}) \Phi(z_{[i]}) - \beta_{[i]} = 0 \quad (8.16)$$

A quantidade $z_{[i]}^*$ que anula a equação (8.16) acima será dada por:

$$z_{[i]}^* = \Phi^{-1}\left(\frac{\beta_{[i]}}{\alpha_{[i]} + \beta_{[i]}}\right) \quad (8.17)$$

Ou seja:

$$d_{[i]}^* = e^{\mu_{[i]} + \sigma_{[i]} \Phi^{-1}\left(\frac{\beta_{[i]}}{\alpha_{[i]} + \beta_{[i]}}\right)} \quad (8.18)$$

As expressões dadas pelas equações (8.17) e (8.18) são similares ao caso normal, sendo válidos os mesmos resultados discutidos anteriormente quanto ao nível de serviço e à robustez da solução. Da mesma forma, este resultado é uma expansão consistente com o caso determinístico, visto que para tal cenário $\sigma_{[i]} = 0$ e, dessa forma, $d_{[i]}^* = e^{\mu_{[i]}}$, i.e., a data de entrega ótima será obtida igualando seu valor à média do tempo de término.

Para demonstrar que este resultado é um ótimo global, basta mostrar que a função $\mathcal{C}(d)$ é convexa em d . Da equação (8.15) pode-se escrever que:

$$\begin{aligned} \frac{\partial^2 \mathcal{C}(d)}{\partial d_{[i]}^2} &= \frac{\partial}{\partial d_{[i]}} [(\alpha_{[i]} + \beta_{[i]})\Phi(z_{[i]}) - \beta_{[i]}] \\ &= (\alpha_{[i]} + \beta_{[i]}) \frac{\partial \Phi(z_{[i]})}{\partial d_{[i]}} \end{aligned} \quad (8.19)$$

Para os termos da diagonal da matriz, substituindo a equação (8.14) em (8.19), vem:

$$\frac{\partial^2 \mathcal{C}(d)}{\partial d_{[i]}^2} = \frac{(\alpha_{[i]} + \beta_{[i]})}{\sigma_{[i]} d_{[i]}} \phi(z_{[i]}) \quad (8.20)$$

Dado que todos os termos componentes do lado direito da equação (8.20) são positivos para todo i , conclui-se que $\partial^2 \mathcal{C}(d)/\partial d_{[i]}^2 \geq 0$. Para os termos fora da diagonal:

$$\frac{\partial^2 \mathcal{C}(d)}{\partial d_{[i]} \partial d_{[k]}} = \frac{\partial}{\partial d_{[k]}} [(\alpha_{[i]} + \beta_{[i]})\Phi(z_{[i]}) - \beta_{[i]}]$$

$$\frac{\partial^2 \mathcal{C}(d)}{\partial d_{[i]} \partial d_{[k]}} = \frac{(\alpha_{[i]} + \beta_{[i]})}{\sigma_{[i]}} \phi(z_{[i]}) \frac{\partial \Phi(z_{[i]})}{\partial d_{[k]}} \quad (8.21)$$

Com $k \neq i, k = 1, 2, \dots, n$. Como $z_{[i]}$ não é função de $d_{[k]}$, conclui-se que $\partial^2 \mathcal{C}(d) / \partial d_{[i]} \partial d_{[k]} = 0$. Dessa forma a matriz Hessiana de $\mathcal{C}(d)$ é positiva semidefinida e $\mathcal{C}(d)$ é uma função convexa em d . Conclui-se então que as datas de entrega obtidas por meio da equação (8.18) correspondem aos ótimos globais da função.

Naturalmente é possível adaptar o Algoritmo 1 para a obtenção das datas de entregas ótimas inteiras, bastando aplicar as equações correspondentes ao caso com distribuição lognormal, conforme exposto a seguir.

Algoritmo 6 (Datas de entrega ótimas inteiras – distribuição lognormal)

1. Calcular $d_{[i]}^*$ conforme equação (8.18);
2. Calcular o piso e o teto do valor obtido no passo anterior (respectivamente $d_{[i]}^f$ e $d_{[i]}^c$);
3. A partir dos valores de piso e teto obtidos, calcular as quantidades $z_{[i]}^f$ e $z_{[i]}^c$;
4. Aplicar a equação (8.12) para obter os custos associados $\mathcal{C}(d_{[i]}^f)$ e $\mathcal{C}(d_{[i]}^c)$;

A data de entrega inteira ótima será o valor $d_{[i]}^\#$ tal que:

$$\mathcal{C}(d_{[i]}^\#) = \min [\mathcal{C}(d_{[i]}^f), \mathcal{C}(d_{[i]}^c)] \quad (8.22)$$

8.3 Sequência Ótima

O objetivo desta seção é determinar a sequência que minimiza o custo médio total dado pela equação (8.12). Definida tal sequência, as datas de entrega ótima serão obtidas por meio da aplicação do Algoritmo 6.

Para tanto se faz necessário substituir a equação (8.18) em (8.12), resultando:

$$\mathcal{C}(d^*) = \sum_{i=1}^n (\alpha_{[i]} + \beta_{[i]}) e^{\mu_{[i]} + \frac{\sigma_{[i]}^2}{2}} [\Phi(z_{[i]}^*) - \Phi(z_{[i]}^* - \sigma_{[i]})] \quad (8.23)$$

Onde d^* é o vetor de datas de entrega obtidas pela aplicação da equação (8.18) e $z_{[i]}^* = (\ln d_{[i]}^* - \mu_{[i]}) / \sigma_{[i]}$.

Pode-se perceber mais uma vez que o resultado obtido é coerente com o caso determinístico associado, dado que para este cenário $\sigma_{[i]} = 0$, o que implica que $\mathcal{C}(d^*) = 0$. Isto é esperado, visto que o vetor d^* é escolhido de forma a minimizar o custo total de atraso e adiantamento (i.e., $d_{[i]}^* = e^{\mu_{[i]}}$).

Interessante notar ainda que, diferentemente do caso com distribuição normal, a função do custo médio total agora inclui a média do tempo de término das tarefas em sua expressão. Ainda que qualquer conjectura acerca desse resultado seja precipitada, este parece ser decorrência do formato da distribuição lognormal, a qual não é simétrica.

Da mesma forma como foi apresentado na seção 4.3 para o caso com distribuição normal, o único método conhecido para determinação da sequência ótima é o de enumeração completa, cabendo o uso de métodos heurísticos para busca de soluções, o que será discutido a seguir.

9 HEURÍSTICAS PARA O PROBLEMA COM DISTRIBUIÇÃO LOGNORMAL

Neste capítulo serão apresentadas quatro diferentes heurísticas para obtenção de soluções candidatas para o problema com distribuição lognormal, baseadas no aprendizado adquirido a partir do problema normal e das expressões matemáticas desenvolvidas no Capítulo 8.

9.1 Heurísticas NEHL-log

Diferentemente da equação (4.26), a qual define o custo médio local para o caso de distribuição normal, não foi possível definir limitantes superiores com termos separáveis para a equação (8.23), sua contraparte para o problema com distribuição lognormal. Dessa forma viu-se necessário definir uma aproximação da expressão, de forma a gerar soluções candidatas iniciais para aplicação de métodos heurísticos.

Uma aproximação razoável pode ser obtida notando-se que o termo entre colchetes da equação (8.23) representa uma faixa de probabilidade entre os pontos $z_{[i]}^*$ e $z_{[i]}^* - \sigma_{[i]}$ no gráfico da distribuição normal, conforme é possível observar na Figura 6 a seguir.

Figura 6 – Representação gráfica do termo entre colchetes da equação (8.23)

Assim sendo, para valores pequenos de $\sigma_{[i]}$, é possível afirmar que:

$$c(d^*) \cong \sum_{i=1}^n (\alpha_{[i]} + \beta_{[i]}) e^{\mu_{[i]} + \frac{\sigma_{[i]}^2}{2}} \sigma_{[i]} \phi(z_{[i]}^*) \quad (9.1)$$

Ou seja, a faixa de probabilidade pode ser aproximada pela área de um retângulo de base $\sigma_{[i]}$ e altura $\phi(z_{[i]}^*)$.

A equação (9.1) é bastante semelhante à equação (4.26), exceto pelo termo $e^{\mu_{[i]} + \frac{\sigma_{[i]}^2}{2}}$, i.e., a média do tempo de término da tarefa. Com base nisto, são propostas duas diferentes sequências candidatas, as quais seriam as equivalentes mais próximas das sequências $\bar{s}$ e $\hat{s}$ da Seção 4.4, quando da apresentação da Regra SOR.

A primeira sequência candidata, denotada como $\bar{s}$, é tal que:

$$\left(\frac{\sqrt{v_{[1]}} e^{m_{[1]} + \frac{v_{[1]}}{2}}}{\gamma_{[1]}} \leq \frac{\sqrt{v_{[2]}} e^{m_{[2]} + \frac{v_{[2]}}{2}}}{\gamma_{[2]}} \leq \dots \leq \frac{\sqrt{v_{[n]}} e^{m_{[n]} + \frac{v_{[n]}}{2}}}{\gamma_{[n]}} \right)$$

Já a segunda sequência candidata, denotada como s^+ é tal que:

$$\left(\frac{v_{[1]} e^{m_{[1]} + \frac{v_{[1]}}{2}}}{\gamma_{[1]}} \leq \frac{v_{[2]} e^{m_{[2]} + \frac{v_{[2]}}{2}}}{\gamma_{[2]}} \leq \dots \leq \frac{v_{[n]} e^{m_{[n]} + \frac{v_{[n]}}{2}}}{\gamma_{[n]}} \right)$$

Onde $\gamma_{[i]} = (\alpha_{[i]} + \beta_{[i]}) \exp(-z_{[i]}^{*2}/2)$.

Conforme comentado, não foi possível definir limitantes superiores para a o problema definido pelo custo dado pela equação (8.23). Desta forma, a escolha das sequências $\bar{s}$ e s^+ consiste em uma tentativa de replicar os resultados obtidos no caso com distribuição normal, ainda que não seja possível demonstrar sua validade. A eficácia de tal escolha será verificada por meio dos experimentos computacionais.

De posse de uma lista candidata, os passos seguintes se reduzem àqueles apresentados na heurística NEHL-1, como exposto a seguir (definiu-se como NEHL-log1 a heurística que utiliza a sequência $\bar{s}$ como candidata inicial e NEHL-log2 aquela que utiliza a sequência s^+).

Algoritmo 7 (heurística NEHL-log1)

1. Considere a sequência $[1, 2, \dots, n]$ e para cada tarefa calcular $k_{[i]} = \sqrt{v_{[i]}} e^{m_{[i]} + \frac{v_{[i]}}{2}} / \gamma_{[i]}$, onde $m_{[i]}$ e $v_{[i]}$ são respectivamente a média e a variância da variável aleatória normal associada ao tempo de processamento da tarefa $[i]$ e $\gamma_{[i]} = (\alpha_{[i]} + \beta_{[i]}) \exp(-z_{[i]}^{*2}/2)$;
2. Dispor as tarefas em ordem não-decrescente de $k_{[i]}$. Denotar esta lista de tarefas como lista candidata;
3. Tomar as duas primeiras tarefas da lista candidata e encontrar a sequência parcial que possui o melhor valor de custo total esperado aplicando a equação (8.23) para as duas sequências possíveis. Não alterar a ordem relativa dessas duas tarefas nos próximos passos. Fazer $i = 3$;
4. Tomar a tarefa na i -ésima posição da lista candidata e colocá-la em todas as possíveis i posições da sequência parcial obtida no passo anterior. Aplicar a equação (8.23) e selecionar a sequência parcial de menor custo total esperado (não alterar a posição relativa das tarefas já fixadas);
5. Parar se $i = n$ e denotar a sequência obtida como s^B , do contrário fazer $i = i + 1$ e retornar ao passo 4.

Algoritmo 8 (heurística NEHL-log2)

1. Considere a sequência $[1, 2, \dots, n]$ e para cada tarefa calcular $k_{[i]} = v_{[i]} e^{m_{[i]} + \frac{v_{[i]}}{2}} / \gamma_{[i]}$, onde $m_{[i]}$ e $v_{[i]}$ são respectivamente a média e a variância da variável aleatória normal associada ao tempo de processamento da tarefa $[i]$ e $\gamma_{[i]} = (\alpha_{[i]} + \beta_{[i]}) \exp(-z_{[i]}^{*2}/2)$;
2. Dispor as tarefas em ordem não-decrescente de $k_{[i]}$. Denotar esta lista de tarefas como lista candidata;
3. Tomar as duas primeiras tarefas da lista candidata e encontrar a sequência parcial que possui o melhor valor de custo total esperado aplicando a equação (8.23) para as duas sequências possíveis. Não alterar a ordem relativa dessas duas tarefas nos próximos passos. Fazer $i = 3$;
4. Tomar a tarefa na i -ésima posição da lista candidata e colocá-la em todas as possíveis i posições da sequência parcial obtida no passo anterior. Aplicar a

equação (8.23) e selecionar a sequência parcial de menor custo total esperado (não alterar a posição relativa das tarefas já fixadas);

5. Parar se $i = n$ e denotar a sequência obtida como s^B , do contrário fazer $i = i + 1$ e retornar ao passo 4.

De posse da sequência s^B , bastará aplicar a equação (8.18) e o Algoritmo 6 para obtenção das datas de entrega ótimas inteiras.

Por serem idênticos a heurística NEHL-1 (a menos da sequência inicial e da função de avaliação de custo), é esperado que os algoritmos acima apresentem complexidade $O(n^3)$.

É possível ainda construir algoritmos equivalentes à heurística NEHL-2, bastando adotar o mesmo critério de parada anteriormente apresentado nos Algoritmos 7 e 8. O resultado desta implementação pode ser descrito como abaixo.

Algoritmo 9 (heurística NEHL-log3)

1. Considere a sequência $[1, 2, \dots, n]$ e para cada tarefa calcular $k_{[i]} = \sqrt{v_{[i]}} e^{m_{[i]} + \frac{v_{[i]}}{2}} / \gamma_{[i]}$, onde $m_{[i]}$ e $v_{[i]}$ são respectivamente a média e a variância da variável aleatória normal associada ao tempo de processamento da tarefa $[i]$ e $\gamma_{[i]} = (\alpha_{[i]} + \beta_{[i]}) \exp(-z_{[i]}^{*2} / 2)$;
2. Dispor as tarefas em ordem não-decrescente de $k_{[i]}$. Denotar esta lista de tarefas como lista candidata;
3. Tomar as duas primeiras tarefas da lista candidata e encontrar a sequência parcial que possui o melhor valor de custo total esperado aplicando a equação (8.23) para as duas sequências possíveis. Não alterar a ordem relativa dessas duas tarefas nos próximos passos. Fazer $i = 3$ e $j = 0$;
4. Tomar a tarefa na i -ésima posição da lista candidata e colocá-la na posição $i - j$ da sequência parcial obtida no passo anterior. Aplicar a equação (8.23) para calcular seu custo. Selecionar como melhor sequência e fazer $j = j + 1$;
5. Colocar a i -ésima tarefa na posição $i - j$ e calcular o custo dessa sequência parcial por meio da equação (8.23). Caso o custo seja maior que aquele obtido no passo anterior, fazer $j = 0$ e ir para o passo 6 (não alterar a posição relativa das tarefas já fixadas). Caso contrário, fazer $j = j + 1$ e retornar para o passo 4;

6. Parar se $i = n$ e denotar a sequência obtida como s^B , do contrário fazer $i = i + 1$ e retornar ao passo 4.

Algoritmo 10 (heurística NEHL-log4)

1. Considere a sequência $[1, 2, \dots, n]$ e para cada tarefa calcular $k_{[i]} = v_{[i]} e^{m_{[i]} + \frac{v_{[i]}}{2}} / \gamma_{[i]}$, onde $m_{[i]}$ e $v_{[i]}$ são respectivamente a média e a variância da variável aleatória normal associada ao tempo de processamento da tarefa $[i]$ e $\gamma_{[i]} = (\alpha_{[i]} + \beta_{[i]}) \exp(-z_{[i]}^2 / 2)$;
2. Dispor as tarefas em ordem não-decrescente de $k_{[i]}$. Denotar esta lista de tarefas como lista candidata;
3. Tomar as duas primeiras tarefas da lista candidata e encontrar a sequência parcial que possui o melhor valor de custo total esperado aplicando a equação (8.23) para as duas sequências possíveis. Não alterar a ordem relativa dessas duas tarefas nos próximos passos. Fazer $i = 3$ e $j = 0$;
4. Tomar a tarefa na i -ésima posição da lista candidata e colocá-la na posição $i - j$ da sequência parcial obtida no passo anterior. Aplicar a equação (8.23) para calcular seu custo. Selecionar como melhor sequência e fazer $j = j + 1$;
5. Colocar a i -ésima tarefa na posição $i - j$ e calcular o custo dessa sequência parcial por meio da equação (8.23). Caso o custo seja maior que aquele obtido no passo anterior, fazer $j = 0$ e ir para o passo 6 (não alterar a posição relativa das tarefas já fixadas). Caso contrário, fazer $j = j + 1$ e retornar para o passo 4;
6. Parar se $i = n$ e denotar a sequência obtida como s^B , do contrário fazer $i = i + 1$ e retornar ao passo 4.

Note que as heurísticas NEHL-log3 e NEHL-log4 consistem nas contrapartes das heurísticas NEHL-log1 e NEHL-log2, respectivamente, considerada a inserção do critério de parada. Como anteriormente, definida a sequência s^B , bastará a aplicação da equação (8.18) e do Algoritmo 6 para obtenção das datas de entrega ótimas inteiras. Importante notar ainda que para estes algoritmos continua válida toda a discussão para a heurística NEHL-2 quanto a complexidade e tempos de execução.

O desempenho de cada uma das heurísticas será evidenciado a seguir nos experimentos computacionais.

10 EXPERIMENTOS COMPUTACIONAIS PARA O PROBLEMA COM DISTRIBUIÇÃO LOGNORMAL

Neste capítulo são apresentadas as descrições de dois procedimentos adotados para análise dos algoritmos apresentados. Inicialmente é realizada uma investigação de métodos computacionais para somas de variáveis lognormais. A seguir são apresentados os resultados obtidos pela aplicação das heurísticas apresentadas no Capítulo 9, assim como as medidas utilizadas para comparação dos algoritmos.

10.1 Descrição dos Experimentos

Foram realizados dois diferentes experimentos. No primeiro foi avaliado o impacto gerado pela utilização do método de Fenton-Wilkinson nas somas dos tempos de processamento, comparando-o ao resultado simulado por meio da aplicação do método de Monte Carlo (um detalhamento sobre este método pode ser obtido em Rubinstein e Kroese (2007)). O segundo experimento consistiu da aplicação das heurísticas propostas para diversas instâncias do problema.

Cada experimento foi realizado com o auxílio de diferentes ferramentas computacionais: enquanto as heurísticas foram codificadas em C++, a análise do método de Fenton-Wilkinson foi desenvolvida em uma planilha eletrônica (o que se justifica, dado que o objetivo principal desta análise é verificar a eficácia do método em termos qualitativos). Todos os experimentos foram executados num computador tipo PC dotado de um processador Intel Core i7 de 2.80 GHz e 8.0 GB de RAM, rodando versão 64-bit do Windows 7 Professional com Service Pack 1.

Para todos os experimentos, médias e variâncias (da distribuição normal associada) dos tempos de processamento foram aleatoriamente geradas a partir de uma distribuição uniforme discreta. As médias foram selecionadas no intervalo entre 1,5 e 2,5, enquanto as variâncias foram sorteadas no intervalo de extremos 0 e 1 (a aplicação das equações (8.2) e (8.3) mostra que as médias dos tempos de processamento irão variar aproximadamente entre 4 e 20 e as variâncias entre 0 e 26, valores bem próximos aos utilizados no problema normal). Os custos unitários de atraso e adiantamento foram gerados a partir de uma distribuição uniforme discreta variando entre 0 e 20.

A análise do método de Fenton-Wilkinson foi dividida em duas dimensões: tempo de execução e qualidade das soluções fornecidas.

Para avaliação do tempo de execução, considerou-se como caso de teste uma instância de 4 tarefas do problema em estudo. Por meio de um procedimento de enumeração, foram geradas todas as possíveis sequências do problema; a seguir a média e a variância do tempo de término de cada tarefa foram calculadas para cada uma dessas sequências através da aplicação dos referidos métodos. Para o método de Fenton-Wilkinson, o passo seguinte consistiu do cálculo do custo esperado para cada sequência por meio da equação (8.23) e posterior seleção da sequência de menor custo. Também foi calculada a data ótima de entrega conforme equação (8.18). Esta informação, combinada à equação 1, permitiu o cálculo do custo médio esperado pelo método de Monte Carlo, o qual foi feito em uma amostra de 50 mil observações para cada tempo de processamento. Já para avaliação da qualidade das soluções fornecidas, utilizou-se a planilha desenvolvida para 1000 diferentes instâncias, comparando-se diretamente os resultados fornecidos pelos métodos.

O segundo experimento pretendeu avaliar a eficácia das heurísticas, a qual foi investigada por meio da construção de dois diferentes conjuntos de tarefas. O primeiro conjunto (Conjunto 1) é composto por amostras que variam de 5 a 11 tarefas, num total de 620 problemas. O segundo conjunto (denotado como Conjunto 2) é composto por amostras que variam de 12 a 1000 tarefas, perfazendo um total de 480 problemas.

Para determinar a qualidade das soluções obtidas, continuou-se adotando como parâmetros de comparação o percentual de sobrecusto e o percentual de otimalidade, conforme definidos pelas equações (6.1) e (6.2). Para a comparação dos resultados fornecidos pelos métodos no Conjunto 2, foi adotado o índice de desvio relativo ($IDvR$), conforme definido por Kim (1995):

$$IDvR = \frac{C_{cand} - C_b}{C_w - C_b} \quad (10.1)$$

Onde C_{cand} é o custo médio total da sequência candidata e C_b e C_w são respectivamente o melhor e o pior custos médios totais encontrados entre os quatro métodos comparados.

Calculou-se ainda o percentual de melhores soluções para cada um dos métodos de acordo com a equação (6.3). Todavia no presente caso o numerador desta equação representa a quantidade de vezes em que o método apurado forneceu a melhor solução dentre todos os métodos (incluídos os empates).

Importante ressaltar que, assim como no caso normal, todos os valores computados são resultado da média do número de instâncias para cada um dos tamanhos de problemas e que foram omitidos os cálculos para determinação das datas de entregas, visto que estes não influenciam na qualidade dos algoritmos.

10.2 Resultados do Método de Fenton-Wilkinson

Em termos de tempo de execução, como esperado, o método de Fenton-Wilkinson apresentou tempo sensivelmente menor: 0,08s contra 5,40s do método de Monte Carlo (ou seja, mais de 60 vezes mais rápido).

Da análise dos dados, foi possível concluir que o método de Fenton-Wilkinson tende a apresentar média e variância da distribuição normal associada maiores do que os verificados pelo método de Monte Carlo. É possível verificar na Figura 7 como tal erro impacta a densidade de probabilidade da soma de duas variáveis lognormais.

Figura 7 – Densidades de probabilidade para a soma de duas variáveis lognormais

Como consequência, espera-se que o custo esperado total tenda a ser maior pelo método de Fenton-Wilkinson. De fato, em média este método apresentou um custo médio total 5,78% maior do que o método de Monte Carlo (observou-se um custo maior em 982 instâncias).

A Figura 8 a seguir evidencia ainda o efeito de maiores médias e variâncias sobre a densidade de probabilidade acumulada.

Figura 8 – Densidades acumuladas para a soma de duas variáveis lognormais

Todavia é relevante ainda analisar se as soluções ótimas de cada método convergem para a mesma sequência, visto que todas as heurísticas são baseadas na comparação de seus custos, não havendo importância no valor absoluto, mas no valor relativo dos custos médios totais das diferentes sequências candidatas. Os resultados das simulações indicam que ambos os métodos convergiram para a mesma solução em 86,50% das instâncias. Interessante notar que para o restante do conjunto as sequências diferiam apenas por uma troca de pares adjacentes, i.e., ainda que estas soluções não tenham apresentado convergência, o método de Fenton-Wilkinson elegeu como solução uma sequência vizinha à sequência ótima.

Desta forma, para o problema em questão preferiu-se o uso do método de Fenton-Wilkinson para a soma de variáveis lognormais, dada sua facilidade de implementação, seu baixo tempo de execução e sua razoável confiabilidade quando comparado ao método de Monte Carlo.

10.3 Resultados das Heurísticas NEHL-log

A Tabela 9 apresenta os resultados obtidos ao aplicar as heurísticas NEHL-log para o Conjunto 1. A análise foi efetuada para diferentes tamanhos de problema (Coluna 1), cada um deles composto por um conjunto de diferentes instâncias (Coluna 2). Além das medidas de percentual de otimalidade (Colunas 3, 6, 9 e 12) e percentual de sobrecusto (Colunas 4, 7, 10 e 13) já definidas, é apresentado o tempo de execução médio para cada algoritmo (Colunas 5, 8, 11 e 14). Para fins de comparação, incluiu-se ainda o tempo médio de execução para o método de enumeração completa (Coluna 15).

Tabela 9 – Comparação entre os algoritmos NEHL-log para o Conjunto 1

No. tarefas	No. instâncias	NEHL-log1			NEHL-log2			NEHL-log3			NEHL-log4			Enumer.
		%otimal	%custo	Tempo (s)	%otimal	%custo	Tempo (s)	%otimal	%custo	Tempo (s)	%otimal	%custo	Tempo (s)	Tempo (s)
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)
5	100	99,00	0,02	3,00E-04	98,00	0,02	2,50E-04	98,00	0,02	2,30E-04	97,00	0,01	2,40E-04	8,40E-04
6	100	98,00	0,00	3,30E-04	97,00	0,01	3,30E-03	94,00	0,01	2,70E-04	95,00	0,04	3,10E-03	4,89E-03
7	100	95,00	0,04	3,90E-04	96,00	0,02	3,90E-03	94,00	0,05	3,50E-04	93,00	0,03	3,30E-03	0,04
8	100	97,00	0,00	5,00E-04	95,00	0,01	5,10E-03	91,00	0,03	3,80E-04	94,00	0,02	3,80E-03	0,36
9	100	92,00	0,04	6,50E-03	97,00	0,02	6,30E-03	91,00	0,07	5,00E-04	95,00	0,03	4,50E-03	3,62
10	60	100,00	0,00	8,50E-03	96,67	0,02	8,00E-04	96,67	0,02	5,17E-04	96,67	0,02	5,83E-04	40,78
11	60	95,00	0,00	9,67E-04	93,33	0,00	9,33E-04	93,33	0,02	6,17E-04	88,33	0,02	5,83E-04	497,39
Média		96,45	0,02		96,29	0,01		93,87	0,03		94,35	0,02		

Tabela 10 – Comparação entre os algoritmos NEHL-log para o Conjunto 2

No. tarefas	No. instâncias	NEHL-log1			NEHL-log2			NEHL-log3			NEHL-log4		
		%best	IDvR	Tempo (s)	%best	IDvR	Tempo (s)	%best	IDvR	Tempo (s)	%best	IDvR	Tempo (s)
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)
12	60	95,00	0,04	1,13E-03	96,67	0,01	2,32E-03	85,00	0,14	7,00E-04	88,33	0,11	7,67E-04
15	60	93,33	0,04	2,22E-03	98,33	0,00	2,67E-03	90,00	0,10	9,33E-04	93,33	0,07	1,02E-03
20	60	91,67	0,08	4,32E-03	98,33	0,00	4,98E-03	66,67	0,32	1,47E-03	78,33	0,20	1,25E-03
50	60	76,67	0,08	0,06	83,33	0,01	0,05	26,67	0,63	0,01	50,00	0,37	0,01
100	60	60,00	0,05	0,45	75,00	0,02	0,44	5,00	0,76	0,05	8,33	0,70	0,04
200	60	55,00	0,01	3,67	50,00	0,01	3,27	0,00	0,68	0,35	0,00	0,86	0,23
500	60	40,00	0,00	53,23	60,00	0,00	52,31	0,00	0,54	5,18	0,00	0,97	3,06
1000	60	8,33	0,01	436,17	91,67	0,00	436,83	0,00	0,28	42,70	0,00	1,00	21,84
Média		65,00	0,04		81,67	0,01		34,17	0,43		39,79	0,53	

Percebe-se que para o Conjunto 1, nenhuma das heurísticas apresentou desempenho notadamente superior: enquanto as heurísticas NEHL-log1 e NEHL-log2 apresentaram percentual de otimalidade de 96,45% e 96,29%, respectivamente, as heurísticas NEHL-log3 e NEHL-log4 encontraram soluções ótimas em 93,87% e 94,35% dos problemas, respectivamente. Relativamente ao percentual de sobrecusto, todos os métodos apresentaram ótima performance, com sobrecusto máximo de 0,04%. Para os tamanhos de problema analisados, a diferença entre os tempos de execução é desprezível.

A Tabela 10 apresenta os resultados comparativos considerando-se o Conjunto 2 de instâncias. Mais uma vez a análise foi efetuada para diferentes tamanhos de problema (Coluna 1), cada um deles composto por um conjunto de diferentes instâncias (Coluna 2). São apresentados os tempos de execução de cada algoritmo (Colunas 5, 8, 11 e 14), o índice de melhores soluções (Colunas 3, 6, 9 e 12) e o índice de desvio relativo entre os métodos (Colunas 4, 7, 10 e 13).

Para o Conjunto 2 percebe-se que as heurísticas NEHL-log1 e NEHL-log2 apresentaram desempenho sensivelmente superior, com vantagem para o segundo método. De fato, a heurística NEHL-log2 encontrou a solução de menor custo entre todos os métodos em 81,67% dos problemas, enquanto a heurística NEHL-log1 encontrou a solução de menor custo em 65,00%. O índice de desvio relativo foi notadamente maior para as heurísticas NEHL-log3 e NEHL-log4 (0,43 e 0,53 respectivamente). A primeira encontrou a melhor solução em 34,17% dos problemas, enquanto a segunda forneceu a melhor solução em 39,79% dos casos. Em termos de tempo de execução, o critério de parada dessas heurísticas permitiu, por exemplo, uma redução de até 20 vezes para as instâncias de 1000 tarefas.

Visto que o índice de desvio relativo aponta um desempenho superior da heurística NEHL-log2 em relação às demais, procedeu-se a um comparativo em pares desta contra todos os demais métodos, a fim de determinar em termos absolutos a qualidade das soluções fornecidas. Para tanto foi construída a Tabela 11, apresentada a seguir:

Tabela 11 – Comparação entre os algoritmos NEHL-log e NEHL-log2 para o Conjunto 2

No. tarefas (1)	No. instâncias (2)	IDR1 (3)	IDR3 (4)	IDR4 (5)
12	60	8,70E-07	1,79E-04	1,81E-04
15	60	1,47E-04	2,91E-04	5,25E-05
20	60	2,41E-05	1,43E-04	7,86E-05
50	60	-4,71E-07	8,71E-05	8,54E-05
100	60	2,90E-07	7,15E-05	9,60E-05
200	60	5,58E-08	4,08E-05	7,64E-05
500	60	6,13E-08	3,07E-05	9,80E-05
1000	60	3,64E-07	2,05E-05	1,26E-04
Média		3,64E-05	1,08E-04	9,92E-05

As descrições de colunas são idênticas às já apresentadas, exceção feita ao índice de diferença relativa, o qual foi adaptado para:

$$IDR = \frac{C_{Cand} - C_{NEHL-Log2}}{C_{NEHL-Log2}} \quad (10.2)$$

Onde $IDR1$, $IDR3$ e $IDR4$ representam, respectivamente, os índices de diferença relativa para as heurísticas NEHL-log1, NEHL-log3 e NEHL-log4.

Com base nos resultados apresentados, pode-se concluir que de fato a heurística NEHL-log2 apresentou desempenho superior, ainda que a heurística NEHL-log1 tenha apresentado resultados bastante próximos.

Analisando-se os resultados obtidos para ambos os tipos de distribuição de probabilidade, é possível inferir que em geral ordenações baseadas em variância apresentam resultados melhores do que aquelas baseadas no desvio padrão do tempo de processamento. Diferentemente do caso com distribuição normal, contudo, o critério de parada teve razoável impacto na qualidade das soluções para o problema com distribuição lognormal. Posto isso, para problemas de maior porte a seleção de um dos algoritmos dependerá da relação de compromisso desejada entre a qualidade das soluções e o tempo de execução de cada um dos métodos.

11 CONCLUSÕES

No presente trabalho abordou-se o problema estocástico de minimização de *earliness* e *tardiness* para máquina única com determinação das datas de entrega. Foram demonstradas diversas propriedades do problema, entre elas a determinação de datas de entrega ótimas com valores inteiros. Quanto ao sequenciamento, para o problema com distribuição normal foram apresentadas duas heurísticas construtivas; para o problema com distribuição lognormal foram apresentadas quatro heurísticas. Verificou-se que todas possuem execução em tempo polinomial e são capazes de fornecer boas soluções.

Os resultados obtidos mostraram que para o conjunto de problemas com soluções ótimas conhecidas, todas as heurísticas forneceram os resultados ótimos na maioria dos casos. Para o caso normal, verificou-se que a qualidade das soluções fornecidas por ambas é superior ao algoritmo disponível na literatura, inclusive para problemas com maior número de tarefas. Tal fato foi atestado por meio da demonstração da otimalidade assintótica das heurísticas. Para o caso lognormal, os resultados obtidos para problemas com maior número de tarefas ficam à disposição da comunidade acadêmica para fins de *benchmark*.

Por fim, mostrou-se ainda que os resultados obtidos são robustos, visto que as datas de entrega são protegidas por um *buffer* de segurança. Este *buffer* é ajustado de acordo com a quantidade de tarefas e com a relevância dos custos unitários de atraso e de adiantamento e é capaz de minimizar a necessidade de revisão do sequenciamento contra eventos inesperados, tais como atrasos e quebras de maquinário.

Para o caso com distribuição lognormal, os resultados obtidos não foram conclusivos quanto à otimalidade assintótica das heurísticas propostas. E, dada a complexidade do problema lognormal, não foi demonstrado se o problema admite algum algoritmo assintoticamente ótimo (visto que sequer foi possível definir limitantes superiores para o problema). Do exposto, este se torna um ponto de atenção para futuras pesquisas.

Também é de grande relevância um estudo que aborde o problema em que as tarefas possuam algum tipo de correlação (e.g. precedência) e uma extensão além do problema de máquina única, mais notadamente para ambientes de projetos.

REFERÊNCIAS

- AL-TURKI, U.M., MITTENTHAL, M., RAGHAVACHARI, M. The single machine absolute deviation early-tardy problem with random completion times. *Naval Research Logistics*, 43, 573–587, 1996.
- BAGCHI, U.; SULLIVAN, R., CHANG, Y. Minimizing mean absolute deviation of completion times about a common due date. *Naval Research Logistics Quarterly*, 33, 227–240, 1986.
- BAKER, K.R., SCUDDER, G.D. Sequencing with earliness and tardiness penalties: A review. *Operations Research*, 38, 22–36, 1990.
- BAKER, K.R., BERTRAND, J.W.M. A comparison of due date selection rules. *AIIE Transactions*, 13, 123–131, 1991.
- BAKER, K.R., TRIETSCH, D. In: Klastorin, T. (Ed.), *Tutorials in Operations Research*, INFORMS (Capítulo 5), 2007.
- BAKER, K.R., TRIETSCH, D. *Principles of Sequencing and Scheduling*. John Wiley, 2009a.
- BAKER, K.R., TRIETSCH, D. Safe scheduling: Setting due dates in single-machine problems. *European Journal of Operational Research*, 196, 69–77, 2009b.
- BAKER, K.R., TRIETSCH, D. Research Notes for Appendix A (In: *Principles of Sequencing and Scheduling*). Wiley, 2010.
- BISKUP, D., FELDMANN, M. Benchmarks for scheduling on a single-machine against restrictive and unrestrictive common due dates. *Computers and Operations Research*, 28, 787–801, 2001.
- BLACK, F., SCHOLES, M. *The Pricing of Options and Corporate Liabilities*. Journal of Political Economy, 81, 637–654, 1973.
- CAI, X., ZHOU, S. Scheduling stochastic jobs with asymmetric earliness and tardiness penalties. *Naval Research Logistics*, 44, 531–557, 1997.
- CHENG, T.C.E. Optimal due-date assignment for a single machine sequencing problem with random processing times. *International Journal of Systems Science*, 17, 1139–1144, 1986.
- CHENG, T.C.E. Optimal assignment of slack due-dates and sequencing of jobs with random processing times on a single machine. *European Journal of Operational Research*, 51, 348–353, 1991.
- CHENG, T.C.E., GUPTA, M.C. Survey of scheduling research involving due dates determination decisions. *European Journal of Operational Research*, 38, 156–66, 1989.
- CHOU, M.C., LIU, H., QUEYRANNE, M., SIMCHI-LEVI D. On the asymptotic optimality of a simple on-line algorithm for the stochastic single-machine weighted completion time problem and its extensions. *Operations Research*, 54, 464–474, 2006.

- FENTON, L.F. The sum of log-normal probability distributions in scattered transmission systems. *IRE Transactions on Communication Systems*, 8, 57–67, 1960.
- FRAMINAN, J.M., LEISTEN, R., RAJENDRAN, C. Different initial sequences for the heuristic of Nawaz, Ensore and Ham to minimize makespan, idle time or flowtime in the static permutation flowshop problem. *International Journal of Production Research*, 41, 121–148, 2007.
- GAREY, M., TARJAN, R., WILFONG, G. One-processor scheduling with symmetric earliness and tardiness penalties. *Mathematics of Operations Research*, Vol. 13, 330–348, 1988.
- GORDON, V., PROTH, J.-M., CHU, C. A survey of the state-of-art of common due date assignment and scheduling research. *European Journal of Operational Research*, 139, 1–25, 2002.
- HALL, N.; KUBIAK, W., SETHI, S. Earliness-tardiness scheduling problems, II: Deviation of completion times about a restrictive common due date. *Operations Research*, 39, 847–856, 1991.
- HALL, N.; POSNER, M.E. Earliness-tardiness scheduling problems, I: Weighted deviation of completion times about a restrictive common due date. *Operations Research*, 39, 836–846, 1991.
- HINO, C.M., RONCONI, D.P., MENDES, A.B. Minimizing earliness and tardiness penalties in a single-machine problem with a common due date. *European Journal of Operational Research*, 160, 190–201, 2005.
- HOOGEVEEN, J.A. , VAN DE VELDE, S.L. Scheduling around a small common due date. *European Journal of Operational Research*, 55, 237–242, 1991.
- HUSSAIN, S.A., SASTRY, V.U.K. Application of genetic algorithm to stochastic single machine scheduling problem with earliness and tardiness costs. *International Journal of Computer Mathematics*, 70, 383–391, 1999.
- KALCZYNSKI, P.J., KAMBUROWSKI, J. On the NEH heuristic for minimizing the makespan in permutation flow shops. *Omega*, 35, 53–60, 2007.
- KANET, J. Minimizing the average deviation of job completion times about a common due date. *Naval Research Logistics Quarterly*, 28, 643–651, 1981.
- KIM, Y. D., A backward approach in list scheduling algorithms for multi-machine tardiness problems. *Computers and Operations Research*, 22, 307–319, 1995.
- LEE, C.Y., CHOI, J.Y. A genetic algorithm for job sequencing problems with distinct due dates and general early-tardy penalty weights. *Computers and Operations Research*, 22, 857–869, 1995.

- LIAO, C.-J.; CHENG, C.-C. A variable neighborhood search for minimizing single machine weighted earliness and tardiness with common due date. *Computers and Industrial Engineering*, 52, 404–413, 2007.
- MAZZINI, A., ARMENTANO, V.A. A heuristic for single machine scheduling with early and tardy costs. *European Journal of Operational Research*, 128, 129–146, 2001.
- NAWAZ, M., ENSCORE JR., E., HAM, I. A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. *Omega*, 11, 91–95, 1987.
- PANWALKAR, S.S., SMITH, M.L., SEIDMANN, A. Common due date assignment to minimize total penalty for the one machine scheduling problem. *Operations Research*, 30, 391–399, 1982.
- PINEDO, M. *Scheduling: Theory, Algorithms, and Systems*. Prentice Hall, 2002.
- PORTOUGAL, V., TRIETSCH, D. Setting due dates in a stochastic single machine environment. *Computers and Operational Research*, 33, 1681–1694, 2006a.
- PORTOUGAL, V., TRIETSCH, D. Johnson's problem with stochastic processing times and optimal service level. *European Journal of Operational Research*, 169, 751–760, 2006b.
- ROSS, S. *A First Course in Probability*. Pearson Education, 2010.
- RUBINSTEIN, R.Y., KROESE, D.P. RUBINSTEIN, R.Y. *Simulation and the Monte Carlo Method*. Wiley, 2007.
- SARIN, S.C., EREL, E., STEINER, G. Sequencing jobs on a single machine with a common due date and stochastic processing times. *European Journal of Operational Research*, 51, 188–198, 1991.
- SCHWARTZ, S.C., YEH, Y.S. On the distribution function and moments of power sums with log-normal components. *Bell System Technical Journal*, 61, 1441–1462, 1982.
- SEIDMANN, A., PANWALKAR, S.S., SMITH, M.L. Optimal assignment of due-dates for a single processor scheduling problem. *International Journal of Production Research*, 19, 393–399, 1981.
- SIDNEY, J.B. Optimal single-machine scheduling with earliness and tardiness penalties. *Operations Research*, 25, 62–69, 1977.
- SOROUSH, H.M., FREDENDALL, L.D. The stochastic single machine scheduling problem with earliness and tardiness costs. *European Journal of Operational Research*, 77, 287–302, 1994.
- SOROUSH, H.M. Optimal sequences in stochastic single machine shops. *Computers and Operations Research*, 23, 705–721, 1996.
- SOROUSH, H.M. Sequencing and due-date determination in the stochastic single machine problem with earliness and tardiness costs. *European Journal of Operations Research*, 113, 450–468, 1999.

STÜBER, G.L. *Principles of Mobile Communication*. Kluwer Academic Publishers, 2002.

TAILLARD, E. Some efficient heuristic methods for the flowshop sequencing problem. *European Journal of Operational Research*, 47, 65–74, 1990.

TAILLARD, E. Benchmarks for basic scheduling problems. *European Journal of Operational Research*, 64, 278–85, 1993.

TRIETSCH, D., MAZMANYAN, L., GEVORGYAN, L., BAKER, K.R. Modeling activity times by the Parkinson distribution with a lognormal core: Theory and validation. *European Journal of Operational Research*, 216, 386–396, 2012.

VAN DEN AKKER, M., HOOGEVEEN, M., VAN DE VELDE, S. Combining column generation and Lagrangean relaxation to solve a single-machine common due date problem. *INFORMS Journal on Computing*. 14, 35–51, 2002.

XIA, Y., CHEN, B., YUE, J. Job sequencing and due date assignment in a single machine shop with uncertain processing times. *European Journal of Operational Research*, 184, 63–75, 2008.