# Balanceamento de linhas de produção com trabalhadores deficientes

***Mayron César de Oliveira Moreira***

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito: 16 de maio de 2011

Assinatura: _____

# Balanceamento de linhas de produção com trabalhadores deficientes [1]

***Mayron César de Oliveira Moreira***

**Orientador:** ***Prof. Dr. Alysson Machado Costa***

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação - ICMC-USP, como parte dos requisitos para obtenção do título de Mestre em Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*.

**USP - São Carlos**
**Maio/2011**

[1]Este trabalho foi financiado pela FAPESP e pelo CNPq.

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados fornecidos pelo(a) autor(a)

d835b de Oliveira Moreira, Mayron César
Balanceamento de linhas de produção com trabalhadores deficientes / Mayron César de Oliveira Moreira; orientador Alysson Machado Costa -- São Carlos, 2011.
90 p.

Dissertação (Mestrado - Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2011.

1. Linhas de produção. 2. Trabalhadores deficientes. 3. Otimização inteira mista. 4. Metaheurísticas. 5. Rotação de tarefas. I. Machado Costa, Alysson, orient. II. Título.

"*A alegria está na luta, na tentativa, no sofrimento envolvido e não na vitória propriamente dita.*"

---

Mahatma Gandhi

*Aos meus pais ...*

# Agradecimentos

Napoleão Bonaparte já dizia que uma vitória tem mais de uma centena de pais. Esta conquista não foge dessa regra. A seguir, deixo o meu mais sincero “muito obrigado” a todos os “pais” dessa conquista.

Agradeço a Deus pela vida e aos meus pais Adilson e Vera por todo apoio incondicional.

Aos meus irmãos Mayke e Amanda, pelas longas conversas que tivemos durante esses últimos dois anos.

À toda minha família: tios, primos e avós, em especial à tia Naná, tia Adalúcia e tia Arleth.

Não esqueçamos de quem há algum tempo atrás nos ajudou a estar aqui. Por isso, agradeço ao prof. José Elias Claudio Arroyo, por ter me dado a oportunidade de dar os meus primeiros passos junto à pesquisa, e ao meu grande amigo Januario, parceiro em diversos trabalhos da graduação, uma pessoa de um companheirismo invejável e um bom humor inconfundível.

Ao Alysson, pela oportunidade de trabalhar neste projeto, pelas orientações, pela confiança, respeito e acima de tudo, pela amizade firmada durante todo esse tempo.

Aos professores Marcus Ritt e Antonio Augusto Chaves, muito importantes no desenvolvimento deste projeto, contribuindo através de diversas sugestões técnicas.

À professora Marina Andretta, que apesar de não estar diretamente envolvida neste projeto, foi importante para meus primeiros passos como futuro docente.

A todos os outros professores e funcionários do ICMC.

Aos amigos de Valadares, de Viçosa, da República Esperança e do JBS.

Aos amigos do LOT e de São Carlos, pelos vários momentos de descontração em "*happy hours*" ou nas quarta-feiras na temakeria. Ao pessoal do grupo Amor Maior, pela amizade e os prazerosos momentos nos ensaios.

Finalmente, e não menos importante, ao CNPq e à FAPESP, pelo suporte financeiro mensal e pela oportunidade de aumentar meus conhecimentos científicos e fazer vários amigos nos simpósios e congressos que participei.

# Resumo

Pessoas portadoras de deficiências encontram enormes dificuldades ao tentarem entrar no mercado de trabalho. De fato, sobretudo em países em desenvolvimento, esta parcela significativa da população representa uma fração ínfima dos trabalhadores empregados. Dentre as iniciativas que tentam reverter este quadro, destaca-se a criação de Centros de Trabalhadores Deficientes (CTD's), empresas sem fins lucrativos que empregam pessoas portadoras de deficiências, geralmente em linhas de produção. Um dos fins últimos dos CTD's é expor os trabalhadores a situações encontradas em uma gama diversa de contextos produtivos, de modo que eles possam, eventualmente, vir a compor o quadro de empresas convencionais.

A organização e planejamento da operação de CTD's envolve uma série de dificuldades. Questões ligadas à ergonomia do trabalho ou ao gerenciamento de qualidade, por exemplo, adquirem características particulares neste ambiente. Da mesma forma, problemas clássicos de balanceamento de linhas de produção ganham novas particularidades devido, sobretudo, à enorme heterogeneidade existente entre os trabalhadores. Neste contexto, nos interessamos por problemas referentes ao balanceamento da linha de produção com trabalhadores deficientes, onde se busca obter a maior eficiência produtiva dadas as habilidades específicas de cada trabalhador.

De maneira mais precisa, o problema de balanceamento de linhas de produção em CTD's, conhecido na literatura como *problema de balanceamento e designação de trabalhadores em linhas de produção* (ALWABP, na sigla em inglês) consiste em alocar tarefas e trabalhadores a estações de trabalho, de modo a minimizar o gargalo produtivo e levando em consideração que cada tarefa tem um tempo de duração que depende do trabalhador escolhido para sua execução. Isto dá ao problema um caráter de dupla alocação, aumentando seu caráter combinatório e, consequentemente, sua dificuldade de resolução. Nesta dissertação, estudamos uma variedade de técnicas de resolução do ALWABP. Os objetivos deste estudo são, primeiramente, obter métodos diversos para resolução do problema que sejam eficazes tanto em termos do tempo computacional necessário para sua utilização como em termos da qualidade da solução obtida.

Dentre as abordagens propostas e testadas encontram-se versões de algoritmos com diferentes complexidades, indo desde heurísticas construtivas e estratégias de busca monotônica em vizinhança até meta-heurísticas como GRASP e Busca Tabu. A variedade de técnicas desenvolvidas permitiu a resolução de um problema ainda mais complexo que o ALWABP, que consiste em programar a linha para diversos períodos produtivos, levando em consideração a rotação de tarefas entre os trabalhadores. Deste modo, os trabalhadores podem ser expostos ao maior número de tarefas possível (atendendo, assim, o fim de treinamento almejado no ambiente dos CTD's).

Para resolução do problema de rotação de tarefas, as técnicas desenvolvidas foram utilizadas em um esquema de otimização híbrido que faz uso de um *pool* de soluções (obtidas pelos métodos heurísticos) que são integradas através de modelos de otimização linear inteira mista.

Os resultados obtidos sugerem que as técnicas desenvolvidas são eficientes e flexíveis para o problema ALWABP e que a sua integração permite a obtenção de soluções eficientes para o problema de rotação de tarefas. Deste modo, esta dissertação propõe um esquema completo para o balanceamento de linhas de produção em CTD's.

# Abstract

Disabled workers face enormous difficulties when trying to enter to the labor market. At the present moment, in particular in developing countries, this group constitutes a small portion of the labor force in productive processes. Among the initiatives that attempt to reverse this situation, we highlight the creation of *sheltered work centers for the disabled* (referred to as SWD henceforth), which are non-profit companies that employ people with disabilities, often in assembly lines.

The organization and planning of the operation of a SWD involves a number of challenges. Issues related to ergonomy or production quality management, for instance, acquire particular characteristics in this environment. Likewise, classic assembly lines balancing modeling and solving techniques have to be modified, due to the significant heterogeneity among workers. In this context, we are concerned with problems related to the assembly line balancing with disabled workers, which attempts to achieve the higher production efficiency as possible, given the specific skills of each worker.

More precisely, the assembly line balancing problem in SWD, known in the literature as the *assembly line worker assignment and balancing problem* (ALWABP), consists in assigning tasks and workers to workstations, in order to minimize the bottleneck of the production line while considering that each task duration time depends on the worker chosen for its execution. This double assignment structure leads to a much more complex problem. In this dissertation, we study a variety of techniques for solving the ALWABP. The goals of this study are, first of all, the development of a number of efficient techniques for solving the problem, both in terms of computational time required for their use and in terms of the quality of the obtained solutions.

Among the techniques proposed and tested, we have versions of algorithms with different complexities, ranging from constructive heuristics and monotonic neighborhood search strategies to metaheuristics such as Tabu Search and GRASP. The diversity of the developed techniques

allowed the resolution of a problem even more complex than the ALWABP, which consists of programming the line for a set of periods, taking into account the rotation of tasks among workers. The objective of this new problem is to propose a solution for a given production period that considers the fact that it might be positive to expose the workers to as many tasks as possible (for training, therapeutical and motivational reasons).

In order to solve this job rotation problem, the techniques developed were integrated into a hybrid optimization scheme that uses a pool of solutions (obtained with the heuristic methods) which become inputs of mixed integer linear optimization models.

The results suggest that the techniques developed are efficient and flexible to the ALWABP and their integration allows the obtention of efficient solutions to the job rotation problem. Thus, this dissertation proposes a complete scheme for the resolution of the balancing problem in SWD production lines.

# Lista de Abreviaturas e Siglas

| | |
|---|---|
| AHRT | Algoritmo Híbrido para Rotação de Tarefas |
| ALDP | *Assembly Line Design Problem* |
| ALWABP | *Assembly Line Worker Assignment Balancing Problem* |
| BH | *Best Heuristic* |
| BLRT | Busca Local para Rotação de Tarefas |
| BLTF | Busca Local com Trabalhadores Fixos |
| BLTT | Busca Local com Troca de Trabalhadores |
| BLM | Busca Local para Melhoramento |
| BT | Busca Tabu |
| BTM | Busca Tabu Minimalista |
| BTE | Busca Tabu *Estendida* |
| CS | *Clustering Search* |
| CTD | Centro de Trabalho para Deficientes |
| DT | Diversificação Tarefas |
| DTr | Diversificação Trabalhadores |
| GRASP | *Greedy Randomized Adaptative Search Procedure* |
| HC | Heurísticas Construtivas |
| HGA | *Hybrid Genetic Algorithm* |
| IBS | *Iterated Beam Search* |
| IT | Intensificação Tarefas |
| ITr | Intensificação Trabalhadores |
| LCR | Lista de Candidatos Restrita |
| MDP | Método de Decomposição por Períodos |
| SALBP | *Simple Assembly Line Balancing Problem* |
| STI | *Smoothness index* |

# Sumário

**Agradecimentos** **iii**

**Lista de Abreviaturas e Siglas** **ix**

**1 Introdução** **1**

**2 Estudo do Problema** **7**

2.1 Notações e Definições . . . . . 7

2.2 SALBP . . . . . 9

2.2.1 SALBP-F . . . . . 10

2.2.2 SALBP-1 . . . . . 12

2.2.3 SALBP-2 . . . . . 14

2.2.4 SALBP-E . . . . . 16

2.2.5 Problemas combinatoriais relacionados . . . . . 17

2.3 ALWABP . . . . . 18

2.3.1 Modelagem matemática . . . . . 21

**3 Revisão da Literatura** **23**

**4 Heurísticas Construtivas** **27**

4.1 Extensões para o ALWABP . . . . . 29

4.1.1 Regras de prioridade para as tarefas . . . . . 29

4.1.2 Regras de prioridade para os trabalhadores . . . . . 31

**5 Busca Tabu** **33**

5.1 Representação da solução . . . . . 34

5.2 Estruturas de vizinhança . . . . . 34

5.3 Função objetivo . . . . . 36

5.4 Busca Tabu Minimalista . . . . . 36
5.5 Busca Local . . . . . 39
5.6 Busca Tabu *Estendida* . . . . . 43

**6 GRASP** **45**
6.1 Fase Construtiva . . . . . 46
6.2 Busca Local . . . . . 47

**7 Rotação de tarefas** **49**
7.1 Formulação matemática . . . . . 50
7.2 Algoritmo Híbrido . . . . . 52
7.2.1 Gerenciamento de soluções factíveis . . . . . 53
7.2.2 Modelo linear para o problema de rotação de tarefas . . . . . 54
7.2.3 Busca Local . . . . . 55
7.2.4 Pseudocódigo do AHRT . . . . . 56

**8 Resultados computacionais** **61**
8.1 Instâncias teste . . . . . 61
8.2 Heurísticas Construtivas . . . . . 63
8.3 Busca Tabu . . . . . 65
8.4 Heurísticas Construtivas como solução inicial . . . . . 67
8.5 Outros algoritmos propostos na literatura . . . . . 71
8.6 Algoritmos propostos *vs.* Abordagens da literatura . . . . . 72
8.7 Rotação de Tarefas . . . . . 73

**9 Conclusões e Trabalhos futuros** **79**

# Lista de Figuras


2.1 Grafo (rede) de precedências - Créditos do exemplo: *Balancing and sequencing of assembly lines* (Scholl, 1999) . . . . . 8
2.2 SALBP-F: Solução para $m$ = 5 e $C$ = 10 - Créditos do exemplo: *Balancing and sequencing of assembly lines* (Scholl, 1999) . . . . . 11
2.3 SALBP-1: Solução para $C$ = 12, com $m$ = 3 e $STI$ = 0 . . . . . 13
2.4 SALBP-1: Solução para $C$ = 12, com $m$ = 4 e $STI$ = 7,07 . . . . . 14
2.5 SALBP-2: Solução para $m$ = 4, com $C$ = 12 e $STI$ = 4,47 . . . . . 15
2.6 SALBP-2: Solução para $m$ = 4, com $C$ = 13 e $STI$ = 5,10 . . . . . 16
2.7 SALBP-E: Solução com a combinação ($m$ = 4, $C$ = 12) - Créditos do exemplo: *Balancing and sequencing of assembly lines* (Scholl, 1999) . . . . . 17
2.8 ALWABP: Solução infactível . . . . . 20
2.9 ALWABP: Solução factível . . . . . 20

5.1 ALWABP-2: Solução para $m$ = 4, com $C$ = 39 . . . . . 35
5.2 ALWABP-2: Representação da solução referente ao grafo da Figura 5.1 . . . . 35
5.3 ALWABP-2: Solução infactível, com $InfP = 3$ e $InfT = 2$ . . . . . 37
5.4 ALWABP-2: Representação da solução referente ao grafo da Figura 5.3 . . . . 37

# Lista de Tabelas


2.1 Notações . . . . . 9
2.2 SALBP: exemplo dos tempos de execução das tarefas . . . . . 19
2.3 ALWABP: exemplo dos tempos de execução das tarefas . . . . . 19

8.1 Detalhes sobre as instâncias do ALWABP-2 . . . . . 62
8.2 Comparação de 16 regras de prioridade dos trabalhadores, uma regra de prioridade do trabalhador e dois esquemas de alocação sobre as 320 instâncias. . . . 64
8.3 Comparação dos algoritmos BTM-Rand e BTE-Rand (melhores desvios em negrito). . . . . 66
8.4 Comparação dos algoritmos BTM-Rand, BTM-MaxPW$^-$, BTM-BH e BTM-BH-Rand (melhores desvios em negrito). . . . . 68
8.5 Comparação dos algoritmos BTE-Rand, BTE-MaxPW$^-$, BTE-BH e BTE-BH-Rand (melhores desvios em negrito). . . . . 69
8.6 Comparação dos algoritmos CS-Rand, CS-MaxPW$^-$ e CS-BH (melhores desvios em negrito). . . . . 70
8.7 Comparação de cinco metaheurísticas propostas para o ALWABP-2 (melhores desvios em negrito). . . . . 74
8.8 Resultados para a família Roszieg (melhores desvios em negrito). . . . . 76
8.9 Resultados para a família Heskia (melhores resultados em negrito). . . . . 76
8.10 Percentual de soluções provenientes dos métodos heurísticos escolhidas pelo modelo (7.13)-(7.18) . . . . . 77

CAPÍTULO

1

# Introdução

Segundo a Organização Mundial da Saúde (OMS), cerca de 10% da população mundial apresenta algum tipo de deficiência (ONU, 1993). O Censo 2000 do IBGE, cujos dados foram divulgados no início de 2003, mostrou que pessoas com deficiência estão em número ainda maior em nossa sociedade (Simonelli, 2009). Esses dados mostram que aproximadamente 24,6 milhões de pessoas, o equivalente a 14,5% da população brasileira, são deficientes físicos.

De maneira geral, dos cerca de 610 milhões de pessoas com deficiência presentes no mundo, aproximadamente 386 milhões estão em idade ativa, havendo uma pequena parcela destas que realiza alguma atividade produtiva. As taxas de empregabilidade entre os deficientes variam consideravelmente dependendo do país ou região considerada (Costa & Miralles, 2009). No Reino Unido, por exemplo, o percentual de desemprego para pessoas com deficiência é de 13%, segundo a OMS, enquanto no Brasil, essa taxa é certamente maior. De acordo com uma pesquisa feita pela Secretaria do Trabalho do Município de São Paulo, cerca de 90% dos trabalhadores deficientes deste município estão desempregados (SERPRO, 2004). De fato, esses dados levam a crer que fatores políticos e sociais vem influenciando de maneira muito mais forte a não presença do trabalhador deficiente no mercado de trabalho do que uma suposta incapacidade do mesmo no desenvolvimento de alguma atividade produtiva. Dados mais recentes (e.g. censo 2010) ainda não se encontravam disponíveis no momento da publicação desta dissertação. Em tempo, é importante ressaltar que a obtenção de dados confiáveis sobre a situação do deficiente no Brasil e no mundo é bastante complicada, sendo que fontes diferentes muitas vezes citam estatísticas completamente díspares. De toda forma, acreditamos que a conclusão que a situação do trabalhador deficiente é precária no Brasil pode ser respaldada sem dificuldade.

Esta não participação de pessoas com deficiência no mercado de trabalho é um fator grave. De fato, pesquisas em diversas áreas destacam a importância do trabalho não apenas como forma de sustento, mas também como forma de se obter o desenvolvimento profissional e pessoal do indivíduo. Segundo Fromm (1962), o trabalho pode ser visto como uma maneira do homem expressar tanto suas faculdades mentais como a si mesmo, levando a um auto-desenvolvimento que conduzirá a uma identidade própria, sendo não apenas um meio, mas um fim. Giordano (2000) refere-se ao trabalho não somente como uma atividade de caráter social, mas como uma atividade que remete a fatores psicológicos, com efeito sobre a saúde mental do trabalhador. Mais especificamente relacionado a pessoas deficientes, Costa (2001) afirma que o trabalho pode ser uma forma da pessoa com deficiência desenvolver um sentimento de maior inclusão na sociedade, levando a um aumento em sua auto-estima e segurança no enfrentamento de barreiras sociais.

Levanta-se, então, uma importante questão de inclusão social, que pode ser definida como o processo no qual as pessoas excluídas e a sociedade buscam, em parceria, equacionar problemas, decidir sobre soluções e efetivar a equiparação de oportunidades (Sassaki, 1997). A Declaração dos Direitos das Pessoas Portadoras de Deficiência, aprovada pela Organização das Nações Unidas (ONU) em 1975 já frisava que, entre outras coisas, todo portador de deficiências tem o direito de desfrutar de uma vida decente, tão normal e plena quanto possível (Simonelli, 2009). Visando melhorar as condições desse cenário bastante desfavorável para as pessoas com deficiência, algumas iniciativas foram adotadas.

No Brasil, a Lei 8.213 de 1991, mais conhecida como "Lei das Cotas", obriga que empresas com mais de 100 funcionários reservem de 2% a 5% de suas vagas para funcionários portadores de necessidades especiais. Associadas à lei, existem diversas iniciativas não-governamentais. A AACD (Associação de Assistência à Criança Deficiente), por exemplo, promove desde 2001 o Programa Trabalho Eficiente (PTE), que desenvolve ações como qualificação profissional (informática, cursos de língua estrangeira, etc), cadastramento curricular, parcerias com empresas e instituições, recrutamento e monitoramento pós-contratação. Simonelli (2009) relata empresas nas esferas pública, privada e institucional que empregam e promovem treinamento para pessoas com deficiência.

Uma outra forma de propiciar a inclusão dos trabalhadores deficientes são os Centros de Trabalho para Deficientes (CTD's), presentes em países tais como Espanha, Japão e Inglaterra. Esses centros de trabalho funcionam como uma etapa inicial para a integração dessas pessoas que posteriormente são absorvidas pelo mercado de trabalho convencional. Os CTD's funcionam como empresas comuns, que competem com empresas do mercado, embora não possuam fins lucrativos. De fato, os CTD's devem ser centros empregadores do maior número possível de pessoas deficientes, recebendo funcionários com diferentes níveis de aptidão (Chaves, 2009).

Apesar de todas as iniciativas, a inclusão de pessoas com deficiências no mercado de trabalho encontra ainda várias dificuldades e diversas vezes esbarra em discriminações (Jaime & Carmo,

2005). Dentro deste contexto, fatores como baixo nível de escolaridade (53% dos trabalhadores possuem ensino fundamental e 38% possuem ensino médio, segundo Simonelli (2009)), número insuficiente de programas de capacitação profissional que atendam a demanda dos trabalhadores deficientes e na maioria dos casos, o não cumprimento das diretrizes da "Lei das Cotas" levam ao isolamento dessas pessoas portadoras de necessidades especiais.

Para se ter uma idéia do quadro geral, 75% das empresas brasileiras nos diferentes portes não possuem nenhuma pessoa portadora de necessidades especiais. Mesmo que em 2007 mais de 22 mil pessoas com deficiência tenham sido incorporadas no mercado de trabalho formal, esses números são ainda tímidos comparados a uma economia que gerou, no mesmo ano, 1,61 milhão de empregos formais (Simonelli, 2009). Curiosamente, são as empresas com menos de 100 empregados, às quais a "Lei das Cotas" não se aplica, as maiores empregadoras de pessoas com deficiência.

Neste trabalho, pretendemos contribuir com esta questão analisando problemas e situações relevantes no contexto do emprego de deficientes físicos em linhas de produção. Tal escolha é motivada pelo potencial existente nesse tipo de atividade quanto à empregabilidade de pessoas com necessidades especiais. Simonelli et al. (2006) realizam um estudo na cidade de São Carlos em que listam a seguinte distribuição das vagas para trabalhadores deficientes: produção (88%), tarefas administrativas (6%), serviços gerais (5%) e outras tarefas exigindo nível técnico (1%). Além disso, Costa et al. (2009) realizam um estudo computacional mostrando que com um bom planejamento da linha, a presença de um trabalhador deficiente acarretará em uma perda muito pequena de eficiência, sobretudo em linhas com vários trabalhadores.

O potencial de linhas de produção como geradora de postos de emprego para deficientes não é algo novo na literatura. De fato, em sua auto-biografia, Ford (1922) já relatava que muitas tarefas realizadas na indústria não exigiam plenas capacidades físicas, podendo ser executadas por pessoas com diferentes características. Na linha de produção de um dos seus primeiros veículos, ele estimou haver 7882 tarefas diferentes. Dessas, 949 foram classificadas como pesadas, exigindo força e uma habilidade corporal apurada por parte de quem fosse executá-las; 3338 requeriam homens de comum condicionamento físico e força. O restante das 3595 tarefas foram consideradas atividades que dispensavam esforço físico e poderiam ser realizadas "pelo menor e mais fraco tipo de homem". Na verdade, a maioria delas "poderia ser executadas com êxito por mulheres ou até por crianças". Posteriormente, Ford (1922) fez uma análise considerando apenas as tarefas mais leves. Esse levantamento concluiu que 670 dessas tarefas poderiam ser executadas por homens sem as duas pernas, 2637 tarefas poderiam ser realizadas por homens sem uma das pernas, 2 tarefas por homens sem ambos os braços, 715 tarefas por homens com apenas um dos braços e 10 tarefas por homens com deficiência total da visão. Além disso, das 7882 tarefas presentes na linha, 4030 não requeriam total capacidade psicológica.

O estudo de Ford (ainda que de maneira caricatural) e, sobretudo, os dados apresentados por Simonelli (2009) e Costa et al. (2009) revelam um grande potencial latente em fábricas, mais

especialmente em linhas de produção, para a inclusão do deficiente físico. Apesar disso, no Brasil, confirma-se a tendência de concentração de pessoas com deficiência no setor de serviços (incluindo o comércio). Estima-se que cerca de 48% dos deficientes empregados atuem nesses ramos de atividade. As empresas do setor de serviços e comércio destacam-se por serem as que mais solicitam esses profissionais, correspondendo a 48% da procura. Já a indústria ocupa o segundo lugar como empregadora, dando oportunidade a cerca de 27% das pessoas com deficiência que atuam no mercado formal (Simonelli, 2009).

Tentando contribuir para um melhor aproveitamento do potencial empregatício de deficientes existentes em linhas de produção, neste trabalho será abordado o problema de balanceamento e designação de trabalhadores em linhas de produção (ALWABP, do inglês *assembly line worker assignment balancing problem*), que aparece de maneira prática quando tenta-se balancear linhas de produção com trabalhadores deficientes. Este problema foi introduzido por Miralles et al. (2007) e ainda é pouco estudado na literatura.

O ALWABP pode ser visto como uma extensão do problema simples de balanceamento de linhas de produção (SALBP, do inglês *simple assembly line balancing problem*). No SALBP, deseja-se balancear uma linha de produção dado um conjunto de tarefas que devem ser distribuídas em estações de trabalhos, sendo que cada tarefa possui seu próprio tempo de execução similar para todos os trabalhadores. Além disso, o balanceamento da linha está sujeito a restrições de precedência de tarefas e cada tarefa deve ser designada a apenas uma estação de trabalho. Assim como o SALBP, o ALWABP possui as restrições de precedência de tarefas e cada tarefa deve ser alocada a somente uma única estação. Entretanto, algumas características presentes neste problema aumentam consideravelmente a complexidade de sua resolução.

No ALWABP, os trabalhadores executam as tarefas com tempos bem variados, sendo que para algumas tarefas, existem trabalhadores que nem sequer são capazes de realizá-las. Desta forma, além do problema de distribuir as tarefas nas estações visando deixar a carga de trabalho o mais igualmente distribuída entre elas, devemos nos preocupar também com a designação dos diferentes trabalhadores às estações.

Nosso objetivo quanto ao ALWABP é mostrar que com um bom processo de planejamento, trabalhadores deficientes podem ser tão eficientes como quaisquer outros e desenvolver plenamente atividades em contextos competitivos. Além disso, também nos interessamos pela programação da rotação de tarefas na linha de produção, consistindo em uma extensão do ALWABP cujo objetivo é fazer com que cada trabalhador execute o maior número de tarefas diferentes possível, mantendo níveis de qualidade e eficiência da linha.

Nos capítulos que seguem, abordaremos os seguintes aspectos a respeito deste trabalho. O capítulo 2 apresenta uma formalização do problema de estudo. O capítulo 3 apresenta uma revisão da literatura sobre alguns trabalhos envolvendo linhas de produção em geral, deficientes físicos no mercado de trabalho e o próprio ALWABP e extensões do problema. Os capítulos 4, 5 e 6 apresentam métodos heurísticos utilizados para resolver o ALWABP. Esses métodos são

usados no capítulo 7, que descreve o problema de rotação de tarefas presente no ambiente de um CTD, assim como o método de resolução híbrido adotado para esse problema. Finalmente, os capítulos 8 e 9 apresentamos os resultados computacionais, conclusões e sugestões para a continuidade dos estudos da área.

CAPÍTULO

2

# Estudo do Problema

A definição rigorosa de um problema nos auxilia a delimitar com exatidão a sua extensão e dá pistas importantes sobre a sua resolução. Neste capítulo, focamos na definição do ALWABP. A seção 2.1 apresenta algumas notações e definições utilizadas neste trabalho. A seção 2.2 descreve o problema clássico da literatura denominado SALBP, explicitando suas versões de decisão e otimização. Na seção 2.3, abordamos o ALWABP, uma extensão do SALBP, descrevendo-o e apresentando a modelagem matemática proposta por Miralles et al. (2007) para o problema (subseção 2.3.1).

## 2.1 Notações e Definições

Nesta seção, apresentam-se notações e definições necessárias para a descrição tanto do problema SALBP quanto do ALWABP. Os conceitos utilizados são em grande parte baseados na obra de Scholl (1999). A Tabela 2.1 nos mostra alguns termos adotados neste trabalho.

**Definição 1** (Grafo de Precedências)**.** Um *grafo (rede) de precedência* $G = (N,A,p)$ é um digrafo não-cíclico com um conjunto $N = \{1,...,n\}$ de $n$ nós e um conjunto $A = \{(i,j)$: $i \in N$ e $j \in F_i\}$ de arcos. Os nós representam as tarefas e os arcos representam precedências diretas entre as tarefas. Uma tarefa $i$ que deve ser executada antes que a tarefa $j$ seja executada é denominada *predecessora* de $j$, enquanto $j$ é uma *sucessora* de $i$. A relação de precedência entre duas tarefas $i$ e $j$ é dita *direta* ou *imediata* se nenhuma das tarefas que sucedem $i$ é predecessora de $j$. O conjunto dos sucessores imediatos de $i$ é denotado por $F_i$. O peso no nó $j$, representado por $p_j$, corresponde ao tempo de execução dessa tarefa. O nó correspondente a uma tarefa sem

predecessores (sucessores) é chamado de origem (sumidouro). Essa abordagem é comumente denominada *forward*.

A precedência reversa $G^r$ do grafo é obtida revertendo a orientação de todos os arcos de $G$. Origens tornam-se sumidouros, predecessores tornam-se sucessores, e vice-versa. Neste caso, chamamos de abordagem *backward* do grafo de precedências.

A Figura 2.1 apresenta um exemplo de um grafo de precedência com 11 tarefas[1], em que se percebe, por exemplo, que a tarefa C deve ser executada antes das tarefas F e G, mas depois da tarefa A.

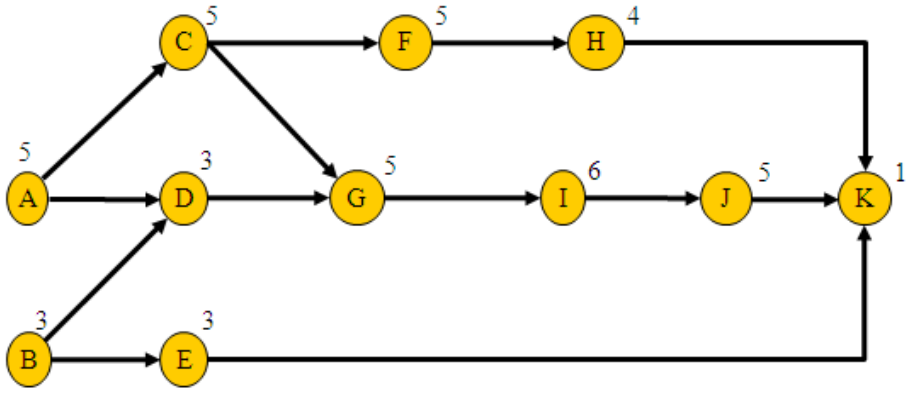


**Figura 2.1:** Grafo (rede) de precedências - Créditos do exemplo: *Balancing and sequencing of assembly lines* (Scholl, 1999)

**Definição 2** (Precedência Transitiva)**.** A *precedência transitiva* $G^*$ = ($N$,$A^*$,$p$) de um grafo $G$ inclui os arcos ($i$,$j$) para todo par de tarefas $i$ e $j$ conectado por um caminho em $G$, ou seja, $A^*$ = {($i$,$j$): $i \in N$ e $j \in F_i^*$}, onde $F_i^*$ contém todos os sucessores da tarefa $i$. A operação $A^*$ - $A$ consiste nos arcos transitivos, representando *relações de precedência indiretas*. Além disso, a cardinalidade de $A^*$ dividida pelo número máximo possível de arcos $n(n-1)/2$ define a complexidade da rede de precedência tanto de $G^*$ quanto de $G$ (Mastor, 1970).

**Definição 3** (Balanceamento de Linha)**.** A partição do conjunto $N$ de todas as tarefas em $m$ subconjuntos disjuntos ordenados $N_s$, com $s$ = 1,...,$m$ é chamado de *designação (ou alocação) de tarefas* se $i \in N_{s_1}$ e $j \in N_{s_2}$, com $s_1 \leq s_2$, $\forall(i,j) \in A$. Em outras palavras, cada tarefa $i$ é designada a exatamente uma estação de trabalho e seus sucessores não devem ser designados a estações anteriores. Uma alocação de tarefas é factível para um dado tempo de ciclo $C$ se o tempo gasto para a execução das tarefas alocadas na estação $s$ ($p(N_s)$) é menor ou igual a $C$, para $s = 1, ..., m$. Assim, uma alocação factível é denominada *balanceamento de linha*. Portanto, uma condição necessária para a existência de um balanceamento de linha é $C \geq p_{max}$, dado que as tarefas são indivisíveis.

**Definição 4** (Smoothness index)**.** O *smoothness index* $STI$ é uma medida da distribuição do trabalho entre as estações e é definido da seguinte forma:

[1]Os números (pesos) presentes acima de cada nó correspondem ao tempo de execução de cada tarefa.

| | |
|---|---|
| $n$ | número de tarefas, |
| $N$ | conjunto das tarefas, |
| $i, j$ | índice para tarefas $(i, j = 1, ..., n)$, |
| $C$ | tempo de ciclo, |
| $m$ | número de estações de trabalho, |
| $s$ | índice para estações de trabalho $(s = 1, ..., m)$, |
| $p_j$ | tempo de execução da tarefa $j$, |
| $p_{max}$ | maior tempo gasto na execução de uma tarefa (= max$\{p_j$: $j = 1, ..., n\}$) |
| $p_{min}$ | menor tempo gasto na execução de uma tarefa (= min$\{p_j$: $j = 1, ..., n\}$) |
| $p_{sum}$ | somatório dos tempos gastos na execução de todas as tarefas (= $\sum_j p_j$) |
| $D_j$ | conjunto de tarefas imediatamente precedentes à tarefa $j$ no grafo de precedência, |
| $F_j$ | conjunto de tarefas imediatamente sucessoras à tarefa $j$ no grafo de precedência, |
| $A$ | conjunto de relações de precedência diretas (=$\{(i, j) : i \in N$ e $j \in F_i\}$), |
| $D_j^*$ | conjunto de predecessores imediatos e transitivos à tarefa $j$, |
| $F_j^*$ | conjunto de sucessores imediatos e transitivos à tarefa $j$, |
| $A^*$ | conjunto de todas as relações de precedência (=$\{(i, j) : i \in N$ e $j \in F_i^*\}$), |
| $S$ | conjunto dos estações de trabalho, |
| $N_s$ | conjunto de tarefas alocadas na estação de trabalho $s$, |
| $p(N_s)$ | tempo gasto para a execução das tarefas alocadas na estação $s$, |
| $E_j$ | estação mais adiantada que pode ser alocada a tarefa $j$, |
| $L_j$ | estação mais tardia que pode ser alocada a tarefa $j$, |
| $SI_j = [E_j, L_j]$ | intervalo de estações ao qual a tarefa $j$ pode ser alocada, |
| $B_s = \{j : s \in SI_j\}$ | conjunto $B_s$ das tarefas que podem eventualmente serem alocadas na estação $s = 1, ..., m$. |

**Tabela 2.1:** Notações

$$STI = \sqrt{\sum_{s=1}^{m} (C - p(N_s))^2} \tag{2.1}$$

Quanto menor o valor de $STI$, mais bem distribuída será a quantidade (tempo) de trabalho entre as estações. Outras definições, tais como *fluxo da linha de produção* e *eficiência da linha* podem ser encontradas em (Scholl, 1999).

## 2.2 SALBP

Em sua forma mais tradicional, uma linha de produção é composta de um conjunto finito de tarefas. Cada tarefa possui um tempo de execução e as tarefas apresentam relações de precedência entre si, isto é, algumas tarefas devem obrigatoriamente ser executadas antes que outras. Tais relações de precedência podem ser representadas através de um grafo de precedências, conceito previamente apresentado na seção 2.1. Nesta seção, descrevemos uma série de proprieda-

des e algumas versões do clássico problema mono-objetivo proposto na literatura, denominado SALBP. Segundo Scholl (1999), as principais características do SALBP são:

- produção em massa de um produto homogêneo, através da execução de $n$ operações para um dado processo produtivo;
- linhas com tempo de ciclo fixo (tipo *paced line*);
- tempos de execução das tarefas determinísticos e inteiros;
- nenhuma restrição de designação além das restrições de precedência;
- layout da linha de tipo serial, com estações de apenas um lado (*one-sided stations*);
- estações igualmente equipadas no quesito estrutural e trabalhadores;
- intervalo fixo entre início e o fim da montagem de produtos = tempo de ciclo

Das versões existentes do SALBP, que abordam distintos objetivos, destacam-se na literatura: SALBP-F, SALBP-1, SALBP-2 e o SALBP-E. Nas próximas subseções, cada um desses problemas é estudado.

### 2.2.1 SALBP-F

O SALBP-F é um *problema de factibilidade* (*feasibility problem*), e consiste em responder a seguinte pergunta: dado um número de estações de trabalho e um determinado tempo de ciclo, existe ou não uma alocação de tarefas na linha de produção que respeite as restrições do problema e os valores fixos de estações e tempo de ciclo. Em outras palavras, o SALBP-F decide se uma linha de $m$ estações pode ser operada com um tempo de ciclo $C$.

A Figura 2.2 ilustra um exemplo do SALBP-F, com o mesmo grafo de precedência da Figura 2.1. Note que para um valor de $m$ = 5 e um valor de $C$ = 10, foi possível encontrar uma alocação das tarefas que levasse ao balanceamento da linha, com valor de $STI$ = 4,12 ($\sqrt{0^2 + 1^2 + 0^2 + 0^2 + 4^2}$). O mesmo exemplo não será factível se adotarmos $m$ = 4 e $C$ = 10.

O SALBP-F pertence a classe dos problemas com *tempo polinomial não-determinístico* (NP, um acrônimo em inglês para *non-deterministic polynomial time*), já que Wee & Magazine (1982) mostraram que o SALBP-F se reduz ao problema de partição, que consiste em decidir se existe um conjunto de números inteiros que pode ser dividido em dois conjuntos com a mesma soma de valores. Essa redução pode ser vista a partir de uma instância particular do SALBP-F, com $m$ = 2, $C = p_{sum}/2$ e na ausência de restrições de precedência. Os mesmos autores e Scholl (1999) mostram que esse problema, por sua vez, pertence à classe dos *NP-completo*.

#### Propriedades do SALBP-F

Uma propriedade muito importante e que pode ser interessante para melhorar os métodos de resolução propostos na literatura consiste em utilizar as relações de precedência entre as tarefas para restringir as estações às quais uma tarefa pode ser inserida. Referimo-nos aos conceitos

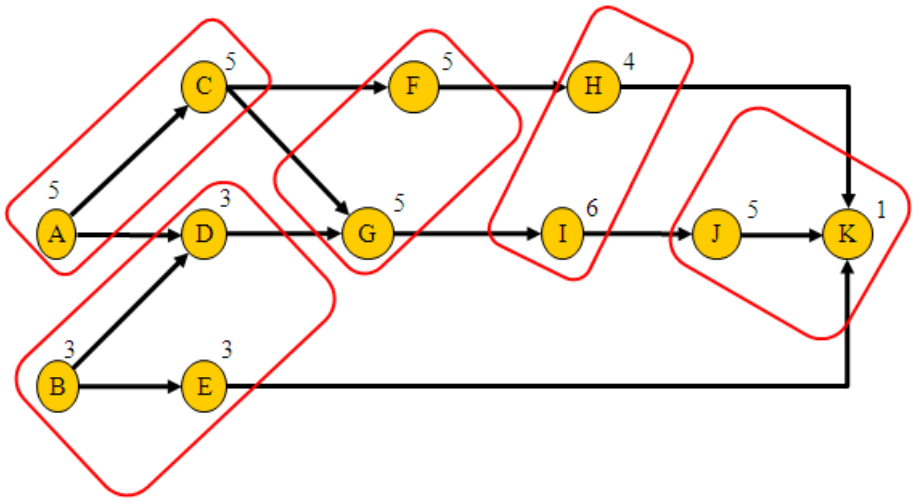


**Figura 2.2:** SALBP-F: Solução para $m$ = 5 e $C$ = 10 - Créditos do exemplo: *Balancing and sequencing of assembly lines* (Scholl, 1999)

de estação mais cedo (*earliest station*) da tarefa $j$, denotada por $E_j$, e estação mais tarde (*latest station*) da tarefa $j$, denotada por $L_j$. Segundo Scholl (1999), $E_j$ e $L_j$ podem ser definidas por:

$$E_j = \lceil (p_j + \sum_{i \in D_j^*} p_i)/C \rceil, \text{ para } j = 1, ..., n \tag{2.2}$$

$$L_j = m + 1 - \lceil (p_j + \sum_{i \in F_j^*} p_i)/C \rceil, \text{ para } j = 1, ..., n \tag{2.3}$$

A fórmula 2.2 leva em conta que a tarefa $j$ não deve ser alocada antes que todas as tarefas precedentes sejam executadas. Assim, um limitante inferior (*lower bound*) para a posição da estação na qual será inserida a tarefa $i$ pode ser dado por (2.2), fórmula que leva em consideração o tempo de todos os predecessores de $i$ e o tempo de ciclo desejado. Analogamente, pode-se usar (2.3) para se obter um limitante superior (*upper bound*) para a posição da estação na qual a tarefa será inserida. A idéia por trás de (2.2) e (2.3) é simplesmente o cálculo do número de estações necessárias para comportar os predecessores ou os sucessores de uma dada tarefa.

Tendo-se conhecimento da estação mais cedo e da estação mais tarde que a tarefa pode ser inserida dado um certo tempo de ciclo, podemos estabelecer um intervalo de estações às quais a tarefa $j$ pode ser alocada, denotado por $SI_j = [E_j, L_j]$. Com isso, obtemos o conjunto $B_s$ das tarefas que podem eventualmente serem alocadas na estação $s = 1, ..., m$: $B_s = \{j : s \in SI_j\}$ (Scholl, 1999).

Tal estratégia de pré-processamento é utilizada para reduzir o número de variáveis em formulações matemáticas ou o espaço de busca de algoritmos heurísticos, levando a uma maior eficiência. Um exemplo do uso de $E_i$ e $L_i$ na redução de um espaço de busca de uma metaheurística é visto na seção 5.2.

### Modelagem matemática

Diversas modelagens considerando o SALBP, e várias de suas extensões, foram propostas na literatura (Scholl, 1999). A seguir, apresentamos a modelagem do SALBP-F proposta por Petterson & Albracht (1975), que se caracteriza por possuir o menor número de restrições e variáveis. Como o SALBP-F consiste em um problema de factibilidade, a função objetivo é omitida.

$$x_{js} = \left\{ \begin{array}{ll} 1 & \text{se a tarefa } j \text{ é alocada na estação } s \\ 0 & \text{caso contrário} \end{array} \right\} \text{ para } j = 1, ..., n \text{ e } s \in SI_j \tag{2.4}$$

sujeito a

$$\sum_{s \in SI_j} x_{js} = 1, \qquad j = 1, ..., n, \tag{2.5}$$

$$\sum_{j \in B_s} p_j x_{js} \leq C, \qquad s = 1, ..., m, \tag{2.6}$$

$$\sum_{s \in SI_i} s x_{is} \leq \sum_{s \in SI_j} s x_{js}, \qquad \forall (i,j) \in A \text{ e } L_i \geq E_j, \tag{2.7}$$

$$x_{js} \in \{0, 1\}, \qquad j = 1, ..., n \text{ e } \forall s \in SI_j. \tag{2.8}$$

O objetivo do SALBP-F é determinar a factibilidade do problema. Assim, o problema com $m$ estações e com o tempo de ciclo $C$ é factível se houver pelo menos uma solução. As restrições (2.5) garantem que a tarefa é executada em uma única estação. As restrições (2.6) restringem o somatório dos tempos de execução das tarefas executadas por cada trabalhador em cada estação ao valor do tempo de ciclo $C$. Finalmente, as restrições (2.7) garantem que as precedências na execução das tarefas vão ser respeitadas.

## 2.2.2 SALBP-1

O SALBP-1 é um problema de otimização que herda as características do SALBP-F, diferindo-se do SALBP-F da seguinte forma: o número de estações de trabalho é uma variável do problema que deve ser minimizada, dado um tempo de ciclo $C$ pré-determinado. Assim, resolver cada instância desse problema leva à resolução de pelo menos uma instância do SALBP-F. Portanto, podemos dizer que o SALBP-1 pertence à classe dos problemas *NP-hard*. As Figuras 2.3 e 2.4 apresentam dois exemplos de soluções válidas para o SALBP-1, sendo a solução representada pela Figura 2.3 de melhor qualidade que a solução da Figura 2.4, pois consegue alocar todas as tarefas da linha com o menor número de estações, respeitando todas as outras restrições do problema.

Na prática, vemos a ocorrência do SALBP-1 em situações produtivas nas quais pode-se ter uma boa estimativa da demanda e então, calcular de antemão a taxa de produtividade desejada. A

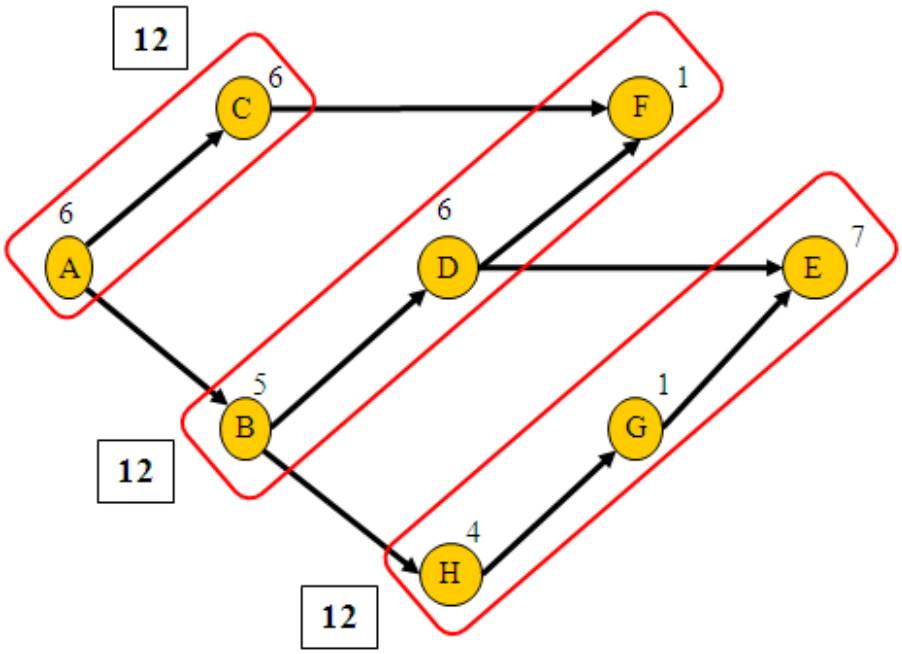


**Figura 2.3:** SALBP-1: Solução para $C$ = 12, com $m$ = 3 e $STI$ = 0

partir disso, uma nova linha de produção deve ser instalada com o menor número de estações possível, a fim de reduzir os custos operacionais.

### Modelagem matemática

Um primeiro passo importante para uma modelagem matemática eficaz do SALBP-1 é a obtenção de um limite superior para o número de estações, já que $m$ não é mais um parâmetro do problema, e sim uma variável de decisão. Com um tal limitante, $\overline{m}$, é possível calcular estimativas para as estações mais tardias a que cada tarefa pode ser alocada e, com isso, reduzir-se o número de variáveis do problema. Diversos métodos estão disponíveis para a obtenção de tal limitante (Scholl, 1999). Um segundo fator a ser considerado refere-se a função objetivo do problema. Diversas maneiras de modelar a função objetivo do SALBP-1 já foram discutidas na literatura. A proposta apresentada a seguir compete ao trabalho de Petterson & Albracht (1975), e baseia-se na introdução um nó destino fictício no grafo de precedências. Assumindo que esse nó destino seja associado a uma tarefa fictícia $n + 1$, o problema consistiria em minimizar o índice da estação ao qual esta tarefa é alocada. A seguir, apresentamos essa formulação proposta por esses dois autores, que mantem as restrições (2.5)-(2.7) e (2.8), substituindo a restrição (2.6) pela equação 2.10.

$$\text{Min} \sum_{s \in SI_n} s x_{n+1s} \tag{2.9}$$

sujeito a

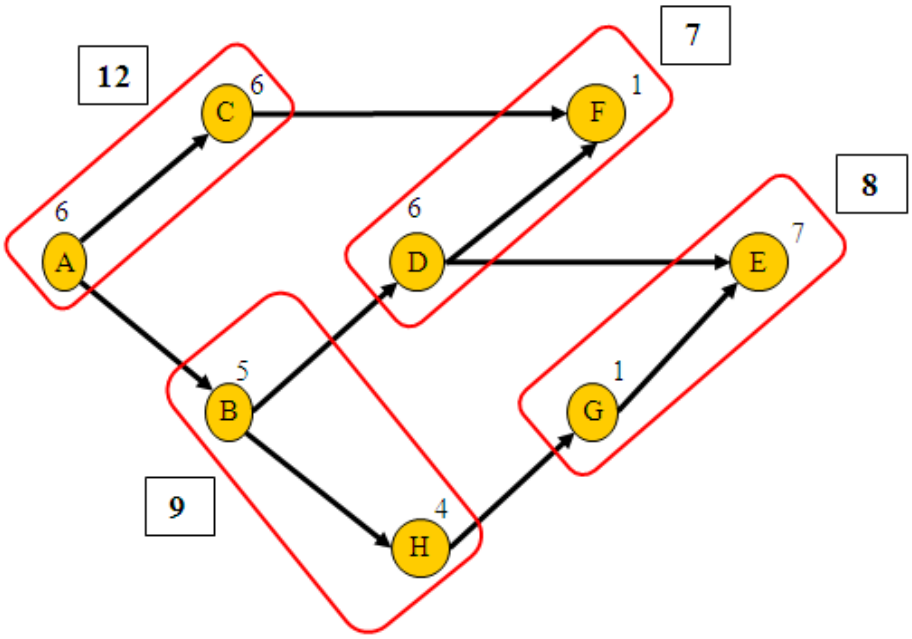


**Figura 2.4:** SALBP-1: Solução para $C$ = 12, com $m$ = 4 e $STI$ = 7,07

$$
\begin{aligned}
&(2.5), (2.7), (2.8), \\
&\sum_{j \in B_s} p_j x_{js} \leq C, \qquad s = 1, ..., \overline{m}. \qquad (2.10)
\end{aligned}
$$

Note que a restrição (2.10) possui o mesmo significado que a restrição (2.6), entretanto, no SALBP-1, considera-se o limite superior do número de estações ($\overline{m}$). Além disso, é importante frisar que o cálculo dos valores de $SI_n$ (função objetivo) e $B_s$ (restrição (2.10)) utiliza o limitante superior $\overline{m}$.

### 2.2.3 SALBP-2

O SALBP-2 é um problema de otimização que também tem sua origem no SALBP-F, possuindo as mesmas características, exceto no que diz respeito à função objetivo, que neste caso, passa a ser a minimização do tempo de ciclo $C$, dado um número fixo de estações. Assim, o SALBP-2 visa a maximização da taxa de produtividade da linha de produção, ocorrendo com muita frequência em situações onde a variação da demanda exige uma reestruturação do processo produtivo. As Figuras 2.5 e 2.6 apresentam dois exemplos de soluções válidas para o SALBP-2 com o mesmo grafo de precedências usado anteriormente e o número de estações fixo em 4. Assim como para o SALBP-1, a resolução do SALBP-2 requer a resolução de pelo menos um problema do tipo SALBP-F, caracterizando-se assim como um problema *NP-hard*.

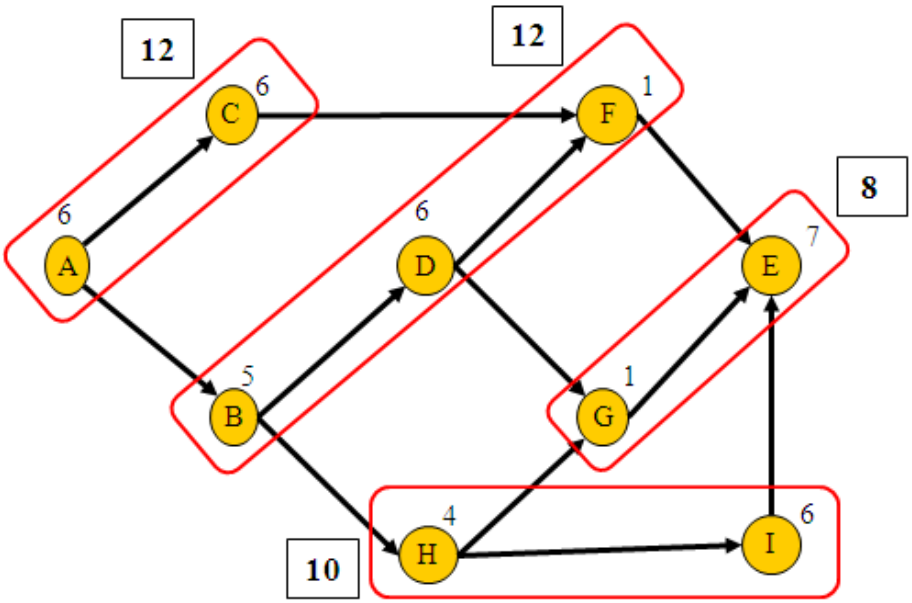


**Figura 2.5:** SALBP-2: Solução para $m$ = 4, com $C$ = 12 e $STI$ = 4,47

Scholl (1999) apresenta vários exemplos de limitantes do tempo de ciclo para o SALBP-2. Entre esses, destacamos dois exemplos, um limitante inferior $\underline{C}$ e um limite superior $\overline{C}$, apresentados abaixo:

$$\underline{C} = \max\{p_{max}, \lceil p_{sum}/m \rceil\} \tag{2.11}$$

$$\overline{C} = \max\{p_{max}, 2 \times \lfloor p_{sum}/m \rfloor\} \tag{2.12}$$

A idéia por trás do limitante (2.11) consiste em escolher o maior valor entre o maior dos tempos de execução das tarefas ou o valor do tempo gasto por todas as tarefas igualmente distribuído na média para as estações. Para a prova de corretude de (2.12), o leitor é referido ao artigo de Hackman et al. (1989).

Os valores das estações mais adiantada e mais tardia são calculados como apresentado pelas equações (2.2) e (2.3), utilizando $\overline{C}$.

### Modelagem matemática

O modelo matemático do SALBP-2 é similar ao modelo proposto para o SALBP-F (vide subseção 2.2.1), considerando porém os valores de $E_j(\overline{C})$, $L_j(\overline{C})$, $SI_j(\overline{C})$ e $B_s(\overline{C})$, dependentes de $\overline{C}$, já que neste caso temos, *a priori*, apenas um limite superior do tempo de ciclo. Uma das formulações proposta na literatura para o SALBP-2 é mostrada em (2.13)-(2.17).

$$\text{Min } C \tag{2.13}$$

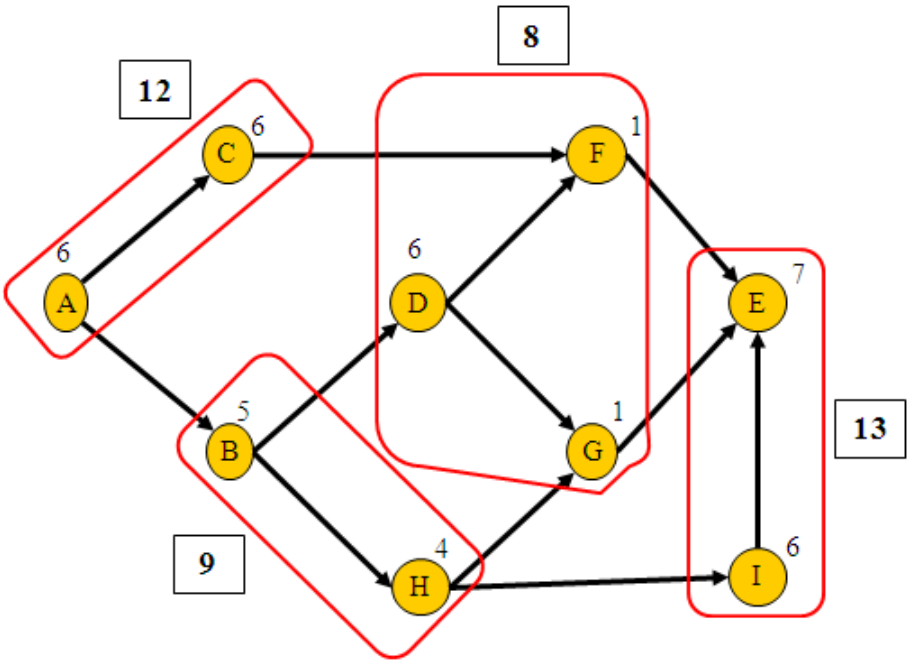


**Figura 2.6:** SALBP-2: Solução para $m$ = 4, com $C$ = 13 e $STI$ = 5,10

sujeito a

$$\sum_{s \in SI_j(\overline{C})} x_{js} = 1, \qquad j = 1, ..., n, \tag{2.14}$$

$$\sum_{j \in B_s(\overline{C})} p_j x_{js} \leq C, \qquad s = 1, ..., m, \tag{2.15}$$

$$\sum_{s \in SI_i(\overline{C})} s x_{is} \leq \sum_{s \in SI_j(\overline{C})} s x_{js}, \qquad \forall (i,j) \in A \text{ e } L_i(\overline{C}) \geq E_j(\overline{C}), \tag{2.16}$$

$$x_{js} \in \{0,1\}, \qquad j = 1, ..., n \text{ e } \forall s \in SI_j(\overline{C}). \tag{2.17}$$

### 2.2.4 SALBP-E

O SALBP-E é um problema de otimização pertencente a classe dos problemas *NP-hard* (Scholl, 1999) e que visa obter a combinação de número de estações $m$ e tempo de ciclo $C$ que leve ao balanceamento da linha, maximizando a eficiência da mesma. Em outras palavras, devemos encontrar a combinação ($m$,$C$) que maximize o valor de $E = \frac{p_{sum}}{mC}$.

Scholl (1999) mostra que existem outras formas de otimizar a eficiência da linhas no SALBP-E sem a necessidade de utilizar a fórmula $\frac{p_{sum}}{mC}$. De fato, os seguintes objetivos são equivalentes à maximização de $E$:

- Minimização da capacidade da linha $T = mC$;
- Minimização do tempo de atraso do balanceamento (*delay time*) $BD = mC - p_{sum}$;

- Minimização da taxa de atraso do balanceamento (*delay ratio*) $BD = 1 - E$;

A Figura 2.7 apresenta uma solução factível para este problema, baseada no grafo de precedências representado pela Figura 2.1. Neste exemplo, obtemos um balanceamento da linha com $m = 4$ e $C = 12$. Como cada parte do produto passa pelas $m$ estações e está disponível em cada uma delas $C$ unidades de tempo, então a capacidade da linha é igual a 48 ($mC$). O tempo de atraso do balanceamento $BD = 48 - 45 = 3$ e é indicado pelas áreas em branco na figura. A eficiência da linha ($E$) é igual a 0,9375, enquanto a taxa de atraso do balanceamento vale 0,00625.

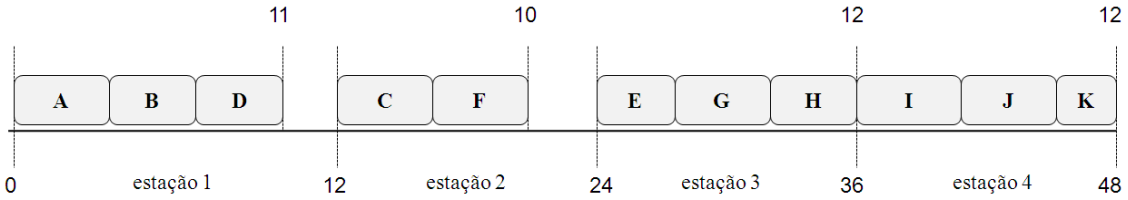


**Figura 2.7:** SALBP-E: Solução com a combinação ($m$ = 4, $C$ = 12) - Créditos do exemplo: *Balancing and sequencing of assembly lines* (Scholl, 1999)

### Modelagem matemática

A formulação para o SALBP-E tem origem na modelagem do SALBP-F, a partir da introdução de $m$ e $C$ como variáveis do problema. A função objetivo torna-se não-linear, já que se pretende minimizar, neste caso, o produto $mC$. Assim, o modelo do SALBP-E é igual ao modelo do SALBP-1 (vide subseção 2.2.2), exceto pela função objetivo, definida a seguir.

$$\text{Min } C \times \sum_{s \in SI_n} s x_{n+1s} \tag{2.18}$$

Novamente, cálculo dos valores de $SI_n$ (função objetivo) utiliza o limitante superior $\overline{m}$ (veja seção 2.2.2).

## 2.2.5 Problemas combinatoriais relacionados

Os problemas de balanceamento de linhas de produção pertencem a classe geral de problemas de otimização combinatória. Assim, podemos encontrar semelhanças das versões do SALBP com outros problemas estudados na literatura, tais como problemas de designação, sequenciamento, agrupamento e seleção. Em muitos casos, a correlação do SALBP com esses problemas se dá através da relaxação de algumas de suas restrições.

As restrições de precedência aumentam significativamente a dificuldade de se obter boas soluções para o problema. Caso essas restrições fossem omitidas, o SALBP-1, por exemplo, se reduziria ao problema de *bin packing*, que visa obter a alocação de todos os objetos utilizando o

menor número possível de *bins* (que podem ser caixas, por exemplo) com capacidade fixa (Wee & Magazine, 1982). O SALBP-2, por sua vez, se reduziria ao problema de escalonamento de $m$ estações paralelas de características similares, em que o objetivo seria a minimização do *makespan* (Coffman et al., 1984).

O problema de roteamento de veículos capacitado (*capacitated vehicle routing problem*) também está relacionado com o SALBP-1, já vez que uma das versões desse problema consiste em minimizar o número de veículos idênticos com uma dada capacidade (no SALBP-1, o tempo de ciclo) que devem fornecer as mercadorias do depósito para os consumidores (Graves & Lamar, 1983). Para o SALBP-2, o mesmo problema pode estar relacionado se fixarmos o número de veículos (no SALBP-2, o número de estações) e considerarmos como função objetivo a maior distância percorrida pelos veículos (Donschke, 1997).

Quando analisamos a alocação das tarefas na linha de produção, podemos associar esse processo a um outro problema conhecido como problema do particionamento de conjuntos (*set partitioning problem*) (Balas & Padberg, 1979), que consiste em encontrar subconjuntos disjuntos de itens tal que a diferença dos valores atribuídos aos itens de cada subconjunto seja minimizada.

Finalmente, a idéia de tempo de ciclo se enquadra perfeitamente no escopo do problema da mochila 0-1 (zero-one *knapsack problem*) (Martello & Toth, 1990), dado que não se pode inserir tarefas em uma estação que ultrapassem a capacidade limitada pelo tempo de ciclo. Um outro problema correlato ao SALBP é o problema do escalonamento de tarefas com uma estação (*one-machine scheduling problem*), se considerada a linha de produção como uma única estação e tendo como objetivo encontrar a melhor sequência de operações (Carlier, 1982).

## 2.3 ALWABP

Ao analisar as características das linhas de produção dos CTD's, Miralles et al. (2007) reuniram elementos suficientes para propor o ALWABP, classificado como um problema *NP-Hard* pois consiste em uma generalização do SALBP.

Tanto no SALBP como no ALWABP, o desafio consiste em atribuir tarefas a cada estação de trabalho respeitando as suas relações de precedência, a fim de otimizar alguma medida de eficiência. De fato, a principal diferença entre os problemas está no fato de que no ALWABP, o tempo de execução de uma tarefa depende do trabalhador que a executa, ao ponto de um trabalhador poder ser incapaz de executar uma determinada tarefa. Surge, então, um problema de dupla alocação, onde além de se alocar tarefas às estações, é necessário atribuir os trabalhadores às estações que possuam tarefas que os mesmos consigam executar, evitando assim a infactibilidade em relação aos pares trabalhador/tarefa. As duas tabelas a seguir mostram um exemplo ilustrativo de duas situações, relacionadas ao tempo de execução de 5 tarefas por 4 trabalhadores, uma em uma linha de produção convencional, caracterizando o ambiente do SALBP, e outra de uma linha de produção presente em um CTD, representando o ALWABP.

| **Tarefas** | **Trabalhadores** | | | |
|---|---|---|---|---|
| | $w_1$ | $w_2$ | $w_3$ | $w_4$ |
| $n_1$ | 10 | 10 | 10 | 11 |
| $n_2$ | 7 | 7 | 7 | 7 |
| $n_3$ | 13 | 13 | 13 | 13 |
| $n_4$ | 19 | 19 | 19 | 20 |
| $n_5$ | 25 | 25 | 25 | 25 |

**Tabela 2.2:** SALBP: exemplo dos tempos de execução das tarefas

| **Tarefas** | **Trabalhadores** | | | |
|---|---|---|---|---|
| | $w_1$ | $w_2$ | $w_3$ | $w_4$ |
| $n_1$ | 10 | 25 | 10 | 30 |
| $n_2$ | 10 | 20 | 15 | 30 |
| $n_3$ | $\infty$ | 30 | 22 | 12 |
| $n_4$ | 80 | 90 | $\infty$ | 31 |
| $n_5$ | 64 | 30 | 23 | $\infty$ |

**Tabela 2.3:** ALWABP: exemplo dos tempos de execução das tarefas

Note que na Tabela 2.2, os tempos dos trabalhadores são bem próximos, podendo ser considerados iguais para fins de modelagem e resolução do problema. Entretanto, a Tabela 2.3 apresenta tempos de execução das tarefas com grande variação entre os trabalhadores. Além disso, existem algumas tarefas, como a $n_3$, $n_4$ e $n_5$ para os trabalhadores $w_1$, $w_3$ e $w_4$, respectivamente, que são infactíveis [2] para os mesmos, ou seja, esses trabalhadores são incapazes de executá-las, devido, possivelmente, a uma incompatibilidade entre a tarefa e as condições particulares de cada trabalhador.

De acordo com Chaves (2009), as principais características do ALWABP são:

- os tempos de procesamento das tarefas e as relações de precedências entre as tarefas são determinísticos;
- existe um dado número de trabalhadores e o tempo de execução de cada tarefa varia de acordo com o trabalhador, uma vez que cada trabalhador possui habilidades e capacidades distintas;
- os trabalhadores são bem distintos entre si, podendo ser eficientes para a execução de determinadas tarefas e ineficientes para outras, podendo mesmo serem incapazes de executarem algumas determinadas tarefas;
- cada estação de trabalho deve ter apenas um trabalhador atribuído a mesma;
- cada tarefa é atribuída somente a uma estação de trabalho, levando em consideração que o trabalhador alocado àquela estação seja capaz de executá-la e a rede de precedência de tarefas seja obedecida.

[2] Por convenção, a infactibilidade da tarefa para algum trabalhador é representada pelo símbolo $\infty$.

Para se ter uma idéia da dificuldade inerente ao novo problema, apresentamos nas Figuras 2.8 e 2.9 dois exemplos de alocações nesse ambiente [3]. Na Figura 2.8, podemos ver que a alocação das tarefas levando em conta a precedência das mesmas foi respeitada, entretanto, a tarefa **G** é infactível para o trabalhador 3, resultando em uma solução inviável na prática. A Figura 2.9 nos mostra uma realidade diferente: fazendo uma pequena modificação no exemplo anterior, alocando o trabalhador 3 às tarefas A-B-C e o trabalhador 1 às tarefas G-F, obtém-se uma solução que além de respeitar a precedência entre as tarefas, consegue factibilidade relacionada à execução das tarefas pelos trabalhadores.

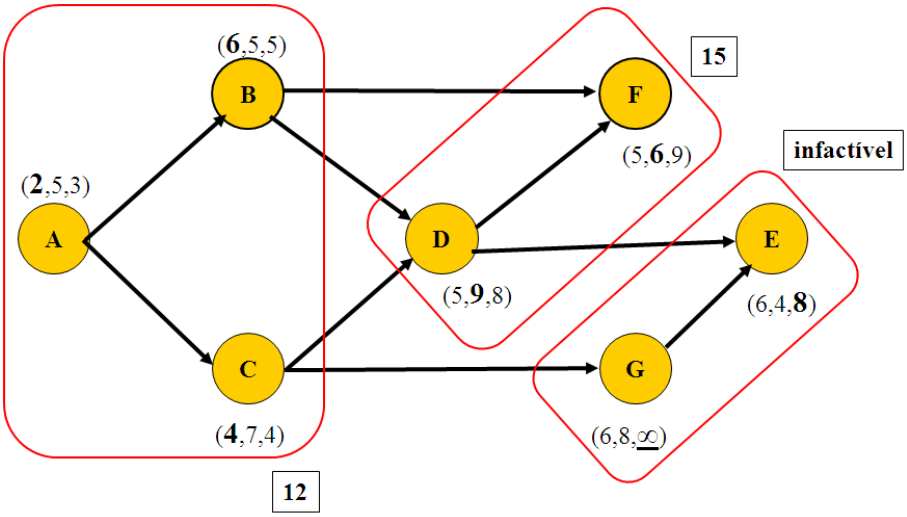


**Figura 2.8:** ALWABP: Solução infactível

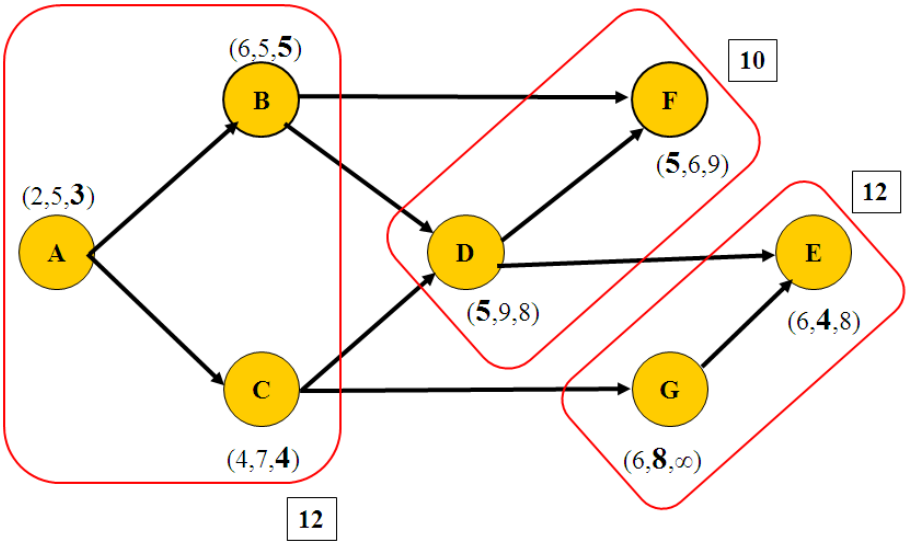


**Figura 2.9:** ALWABP: Solução factível

[3] O peso ($p_1$,$p_2$,$p_3$) indicado em um nó $N_i$ qualquer significa: tempo dos trabahadores 1, 2 e 3 para executarem a tarefa $i$, respectivamente. Se o valor $p_2$, por exemplo, estiver destacado, o trabalhador 2 foi alocado a esta tarefa.

Assim como o SALBP-1 refere-se ao problema de se minimizar o número de estações de trabalho e o SALBP-2 tem como objetivo a minimização do tempo de ciclo (isto é, maximização da taxa de produção), o ALWABP-1 também se refere a otimização do número de estações de trabalho e o ALWABP-2 à minimização do tempo de ciclo.

### 2.3.1 Modelagem matemática

Em um CTD, a situação mais típica se enquadra no ALWABP-2, ou seja, dado um número fixo de trabalhadores disponíveis, busca-se minimizar o tempo de ciclo da linha de produção, maximizando então a eficiência da linha (Chaves, 2009). Isso é coerente no sentido que o objetivo dos CTD's é empregar o maior número possível de trabalhadores mantendo uma boa taxa de produtividade.

Um modelo matemático linear inteiro misto para o problema foi proposto por Miralles et al. (2007). Para a versão adaptada deste modelo reproduzida abaixo (Moreira & Costa, 2009), utilizamos as notações apresentadas na seção 2.1 e introduzimos adicional nomeclatura, como pode ser visto a seguir:

$$\text{Min } C \tag{2.19}$$

sujeito a

$$\sum_{w \in W} \sum_{s \in S} x_{swi} = 1, \qquad \forall i \in N, \tag{2.20}$$

$$\sum_{s \in S} y_{sw} = 1, \qquad \forall w \in W, \tag{2.21}$$

$$\sum_{w \in W} y_{sw} = 1, \qquad \forall s \in S, \tag{2.22}$$

$$\sum_{w \in W} \sum_{s \in S} s \cdot x_{swi} \leq \sum_{w \in W} \sum_{s \in S} s \cdot x_{swj}, \qquad \forall j \in N, \forall i \in D_j, \tag{2.23}$$

$$\sum_{w \in W} \sum_{i \in N} p_{wi} \cdot x_{swi} \leq C, \qquad \forall s \in S, \tag{2.24}$$

$$\sum_{i \in N} x_{swi} \leq |N| y_{sw}, \qquad \forall w \in W, \forall s \in S, \tag{2.25}$$

$$x_{swi} = 0, \qquad \forall w \in W, \forall s \in S, \forall i \in I_w, \tag{2.26}$$

$$y_{sw} \in \{0, 1\}, \qquad \forall s \in S, \forall w \in W, \tag{2.27}$$

$$x_{swi} \in \{0, 1\}, \qquad \forall s \in S, \forall w \in W, \forall i \in N. \tag{2.28}$$

em que,

| | |
|---|---|
| $s$ | índice para estações de trabalho, |
| $w$ | índice para trabalhadores, |
| $W$ | conjunto dos trabalhadores, com $\|W\| = \|S\|$, |
| $p_{wi}$ | tempo de execução da tarefa $i$ quando executada pelo trabalhador $w$, |
| $I_w$ | conjunto de tarefas que o trabalhador w não é capaz de executar, |
| $x_{swi}$ | variável binária. Igual a 1 apenas se a tarefa $i$ é alocada ao trabahador $w$ na estação $s$ |
| $y_{sw}$ | variável binária. Igual a 1 apenas se o trabalhador $w$ é alocado à estação $s$. |

O objetivo da formulação (2.19)-(2.28) é a minimização do tempo de ciclo, isto é, o maior dos tempos gastos em uma estação. As restrições (2.20) garantem que cada tarefa seja executada por um trabalhador, enquanto (2.21) e (2.22) asseguram que cada trabalhador é alocado a uma única estação e que cada estação recebe um único trabalhador, respectivamente. As restrições (2.23) garantem que as precedências de execução entre as tarefas são respeitadas. As restrições (2.24) associam à variável $C$ a medida do tempo de ciclo. As restrições (2.25) garantem que uma tarefa só é executada por um trabalhador $w$ em uma estação $s$ caso este trabalhador realmente esteja alocado a esta estação. Nestas restrições, usou-se um valor de limitante $|N|$, equivalente ao número de tarefas, que por sua vez é mais apertado que o original proposto por Miralles et al. (2007). Finalmente, as restrições (2.26) definem que uma tarefa não é designada a um trabalhador se o mesmo não for capaz de executá-la. Essas restrições poderiam ser definidas diretamente nas restrições (2.28), através da eliminação de variáveis associadas, mas são expressas como um caso particular de restrições por conveniência, uma vez que são usadas para explicar o algoritmo de busca tabu do Capítulo 5.

CAPÍTULO

3

# Revisão da Literatura

Uma revisão clássica de métodos aplicados ao SALBP é apresentada por Baybars (1986). Além disso, o autor realça a diferenciação entre SALBP-1 e SALBP-2. Becker & Scholl (2006) apresentam uma revisão mais geral de problemas envolvendo linhas de produção, tais como o SALBP-F e SALBP-E, além de variações da linha que geram por sua vez outros problemas de maior complexidade. Outras revisões recentes também voltadas à classificação de problemas em linhas de produção podem ser vistas em (Boysen et al., 2007, 2008).

Alguns trabalhos abordam a otimização de linhas de produção considerando diferentes desempenhos dos trabalhadores. Mansoor (1968) propôs uma heurística para linhas de produção com diferentes níveis de desempenho dos trabalhadores. Bartholdi & Eisensteein (1996) estudam a linha *Toyota Swen System* para o caso de trabalhadores com velocidades diferentes. Doerr et al. (2000) consideram uma linha de produção com trabalhadores de diferentes habilidades e com um fator de hora-extra caso a demanda diária da produção não seja alcançada. Hopp et al. (2001) e Gel et al. (2002) estudam o caso onde há dois tipos de trabalhadores, os rápidos e os lentos. Já Corominas et al. (2003) apresentam um modelo de balanceamento de linhas de produção que considera tempos diferentes de execução das tarefas de acordo com a estação à qual as mesmas foram alocadas, sendo inserido em um contexto em que se faz necessária a diferenciação entre trabalhadores experientes e inexperientes.

Em outros contextos, máquinas diferentes podem realizar tarefas diferentes com distintas velocidades. Neste caso, a decisão da seleção de equipamentos é combinada com o balanceamento de rede, recebendo o nome de *problema de projeto de linha de produção* (ALDP, do inglês *assembly line design problem*). Rekiek et al. (2002) apresentam um *survey* de métodos de otimização aplicados ao ALDP. Apesar do ALDP considerar tempos de processamento diferentes

na execução das tarefas, esse problema se mostra distinto do encontrado nos CTD's. De fato, não se tem a intenção de reduzir o número de estações de trabalho em um CTD. Ao contrário, deseja-se empregar o maior número de trabalhadores possível. Além disso, cada trabalhador é único, ao contrário dos equipamentos, os quais podem ser adquiridos em grande quantidade com as mesmas características.

Trabalhos envolvendo deficientes físicos em indústrias não são muito comuns. Simonelli (2005) desenvolve e sistematiza um modelo de análise de tarefas em postos de trabalhos a serem ocupados por pessoas com necessidades especiais. Um estudo de caso em uma indústria aeronáutica localizada no município de Botucatu, São Paulo, foi realizado para testar a aplicabilidade do modelo, obtendo bons resultados. Simonelli (2009) analisa as políticas e práticas de inclusão de pessoas com deficiência no mercado de trabalho brasileiro. Neste trabalho, a visão do modelo biomédico e do modelo social para conceituação da deficiência são contrastadas. No modelo biomédico, o indivíduo precisa ser curado, tratado, reabilitado e habilitado, a fim de ser incluído na sociedade. Já o modelo social transfere a responsabilidade de inclusão à própria sociedade, tendo ela que se adaptar para que o deficiente físico seja de fato incluído. Analisadas as características dos dois modelos, a autora mostra que o modelo biomédico se mostra insuficiente para a efetiva inclusão das pessoas com deficiencia.

Mais especificamente quanto ao ALWABP, apenas recentemente este problema passou a ser estudado na literatura. Miralles et al. (2007) apresentaram o problema pela primeira vez e desenvolveram uma formulação matemática para o ALWABP-2 e um estudo de caso para um CTD espanhol localizado na região de Valência. Em Miralles et al. (2008), os autores propõem um algoritmo *branch-and-bound* para o mesmo problema.

Devido à complexidade do ALWABP, métodos exatos como o *branch-and-bound* só apresentam desempenho satisfatório para pequenas instâncias. Assim sendo, métodos heurísticos passaram a ser investigados para contornarem a dificuldade encontrada pelos métodos exatos. Chaves et al. (2007) desenvolveram uma heurística auxiliada por um procedimento de busca por agrupamento. Esse algoritmo possui quatro componentes que atuam de forma independente: busca de uma solução por meio de uma metaheurística; procedimento de agrupamento iterativo; análise do centro do cluster; aplicação de busca local. Mais detalhes sobre esse trabalho podem ser encontrados em (Chaves, 2009) e em (Chaves et al., 2009). Já Blum & Miralles (2011) propuseram um algoritmo baseado na metaheurística *beam search* para o ALWABP, obtendo alguns dos melhores resultados conhecidos até o momento.

Em relação às extensões, Miralles et al. (2005) propõem uma formulação do ALWABP para o caso de linhas com layout em U. Moreira et al. (2009) e Costa et al. (2009) desenvolvem um modelo que considera a presença de apenas um trabalhador deficiente na linha de produção. Para tanto, considerou-se uma pequena extensão da formulação proposta para o SALBP-2 (vide subseção 2.2.3). Os resultados preliminares desta abordagem sugerem que um bom planejamento da linha de produção pode tornar praticamente invisíveis as deficiências do trabalhador,

afetando pouco ou quase nada a eficiência da linha de produção. Araújo et al. (2010) realizam um estudo de linhas de produção com trabalhadores deficientes com a possibilidade de layouts de linhas com máquinas em paralelo. Desta forma, os autores propõem um modelo linear inteiro misto para o problema e comparam seu desempenho com o modelo serial, obtendo resultados interessantes.

Uma extensão do ALWABP muito encontrada em ambientes reais relaciona-se à idéia de rotação de tarefas. A rotação de tarefas é um recurso de flexibilização do trabalho na linha de produção, que tem por meta assegurar a possibilidade de uso multiplicado da mão-de-obra, proporcionando a polivalência dos trabalhadores. Para o caso mais simples do SALBP, a programação de rotação de tarefas já apresenta uma grande complexidade, tornando-se difícil já na resolução de problemas de tamanho moderado conforme verificado por Carnahan et al. (2000). Esses autores contextualizam o problema de rotação de tarefas para minimizar a realização de tarefas extenuantes por um mesmo trabalhador. Para tanto, aplicam programação linear inteira para problemas com até 128 variáveis de decisão e algoritmos genéticos. Outras técnicas tais como *simulated annealing* (Seçkiner & Kurt, 2007), colônias de formigas (Seçkiner & Kurt, 2008) e algoritmos gulosos com intensificação (Tharmmaphornphilas & Norman, 2007) também foram propostos na literatura.

Em um CTD, a rotação de tarefas pode assumir um caráter de treinamento e mesmo terapêutico, à medida que leva o trabalhador a executar tarefas diferentes que, possivelmente, envolvam novas capacidades e aprendizados. Para rotação de tarefas no ALWABP, ao nosso conhecimento, Costa & Miralles (2009) foram os únicos autores que abordaram esse problema. Esses autores modelaram o problema e definiram como função objetivo a maximização do número de tarefas distintas que cada trabalhador executa durante todos os períodos considerados. Para a resolução das instâncias, utilizaram um método heurístico de decomposição por períodos. Pelo fato de ainda se basearem na resolução exata do ALWABP, os autores conseguem resultados apenas para instâncias de tamanho moderado.

CAPÍTULO

4

# Heurísticas Construtivas

Existem diversos trabalhos com proposições de heurísticas para o SALBP, especialmente para o SALBP-1 (Domschke et al., 1993). Em grande parte, essas abordagens heurísticas consistem em métodos baseados em prioridades e procedimentos de enumeração restrita (Talbot et al., 1986). Em contrapartida ao SALBP-1, existem poucos procedimentos heurísticos para resolver o SALBP-2, sendo que a maioria deles utiliza métodos de busca que resolvem iterativamente as instâncias do SALBP-1 para obter soluções do SALBP-2 (Hackman et al., 1989). Scholl & Voß (1996) apresentam várias heurísticas tanto para o SALBP-1 quanto para o SALBP-2. Além disso, procedimentos de melhoria são desenvolvidos para o SALBP-2 e combinados com um algoritmo de busca tabu.

A ideia deste capítulo é estender as heurísticas construtivas apresentadas por Scholl & Voß (1996). Deste modo, faremos uma descrição dos procedimentos propostos por esses autores, tanto para o SALBP-1 quanto para o SALBP-2 e posteriormente, mostraremos as adaptações que foram necessárias para a implementação desta ideia para o ALWABP.

A estratégia utilizada por Scholl & Voß (1996) na resolução do SALBP-2 consiste em resolver instâncias do SALBP-1 para diferentes tempos de ciclo. Assim, um tempo de ciclo $C$ pode ser considerado um limitante superior para o SALBP-2 se a solução obtida para o SALBP-1 possui um número de estações inferior ao valor desejado $m$. Os métodos heurísticos propostos são baseados em regras de prioridade que ordenam as tarefas de acordo com um dado critério e as alocam nas estações de acordo com essa ordem. Note que, neste contexto, uma tarefa é considerada *disponível* se ela ainda não foi alocada em alguma estação e se todos os seus predecessores já foram alocados. Seja $C$' um valor de tempo de ciclo desejado. A heurística para o SALBP-1 baseada em regras de prioridades é apresentada pelo Algoritmo 1.

**Algoritmo 1** Heurística simples baseada em regras de prioridades para o SALBP-1

1: **entrada:** $C$'
2: $U$: conjunto de tarefas que não foram alocadas (inicialmente, $U \leftarrow N$);
3: $m \leftarrow 0$;
4: $c_m \leftarrow 0$
5: obtenha o conjunto de tarefas disponíveis, $N$' ($N' \subset U$);
6: ordene o conjunto de tarefas $N$' de acordo com a regra de prioridade usada;
7: **enquanto** $U \neq \varnothing$ **faça**
8:  **para** todas as tarefas $i \in N$' **faça**
9:  **se** $c_m + p_i < C$' **então**
10:  aloque a tarefa $i$ na estação $m$:
11:  $c_m \leftarrow c_m + p_i$;
12:  $U \leftarrow U \backslash \{i\}$;
13:  atualize $N$' com as tarefas eventualmente disponíveis;
14:  **fim**
15:  **fim**
16:  **se** $U \neq \varnothing$ **então**
17:  $m \leftarrow m + 1$;
18:  $c_m \leftarrow 0$;
19:  **fim**
20: **fim**
21: **saída:** número de estações necessárias: $m$.

A ideia do método consiste em tentar alocar as tarefas disponíveis à estação de trabalho atual, de acordo com a regra de prioridade adotada (linhas 7-20). Se uma tarefa é alocada a essa estação atual, a lista de tarefas não-alocadas e a lista de tarefas disponíveis são atualizadas (linhas 10-13). Caso não exista alguma tarefa a ser alocada na estação atual sem exceder o tempo de ciclo desejado $C$', então a estação atual é fechada, já que nenhuma tarefa pode ser alocada a ela, e uma nova estação é aberta (inicializando-se tempo gasto para a execução das tarefas) (linhas 17-18).

Um algoritmo para o SALBP-2 pode ser então desenvolvido por meio de sucessivas chamadas do Algoritmo 1 para diferentes valores de $C$'. Scholl & Voß (1996) mencionam diferentes maneiras de variar o tempo de ciclo $C$, tais como a estratégia de busca binária e uma simples busca linear, que inicia com um valor conhecido de um limitante inferior de $\underline{C}$ e o incrementa em uma unidade a cada iteração até que a heurística para o SALBP-1 seja resolvida. Esse método é apresentado pelo Algoritmo 2, que retorna o valor de tempo de ciclo da solução encontrada ou $\infty$, caso a melhor solução encontrada seja pior que o valor de $\overline{C}$ conhecido.

Na sequência, apresentamos as regras de prioridade propostas por Scholl & Voß (1996) que mais obtiveram êxito para o SALBP:

1. **MaxF**: número decrescente de sucessores imediatos e transitivos, $|F_i^*|$;
2. **MaxIF**: número decrescente de sucessores imediatos $|F_i|$;
3. **MaxTime**: tempo de exeução da tarefa decrescente, $p_i$;

**Algoritmo 2** Heurística simples para o SALBP-2

1: **entrada:** $\underline{C}$, $\overline{C}$ e o número de máquinas $m$ desejado;
2: $C_{best} \leftarrow \infty$;
3: **para** $C' \leftarrow \underline{C} \ldots \overline{C}$ **faça**
4: seja $m'$ a saída para o Algoritmo 1 quando chamado com o parâmetro $C'$;
5: **se** $m' \leq m$ **então**
6: $C_{best} \leftarrow C'$;
7: vá ao passo 10;
8: **fim**
9: **fim**
10: **saída:** $C_{best}$.

4. **MaxPW**: pesos de posição decrescentes (*descending positional weights*), $pw_i = p_i + \sum_{j \in F_i^*} p_j$;
5. **MaxTimeL**: valor decrescente do tempo de execução da tarefa pela estação mais tarde, $p_i/L_i$;
6. **MaxTimeSlack**: valor decrescente do tempo de execução da tarefa pelo intervalo da estação mais tarde e estação mais cedo (*slack*), $p_i/sl_i$, onde $sl_i = L_i - E_i$[1].

## 4.1 Extensões para o ALWABP

A extensão dos procedimentos propostos para o SALBP não é trivial no contexto do ALWABP. Primeiramente, para verificar se uma tarefa pode ser alocada a uma estação, o Algoritmo 1 compara o tempo de execução de todas as tarefas já alocadas nesta estação ($c_m$) mais o tempo de execução da tarefa atual com o tempo de ciclo desejado $C'$. Para o ALWABP, essa verificação se torna mais dispendiosa, visto que o tempo de execução de cada tarefa depende de qual trabalhador a executa, levando a um problema de decisão. Além disso, não existem estratégias para selecionar um trabalhador para cada estação de trabalho. Desta forma, apresentamos a seguir modificações propostas ao Algoritmo 1 a fim de lidar com esses dois problemas.

### 4.1.1 Regras de prioridade para as tarefas

Das seis regras de prioridade propostas por Scholl & Voß (1996), as duas primeiras podem ser utilizadas diretamente no ALWABP, uma vez que não dependem do tempo de execução das tarefas pelos trabalhadores. As regras **MaxTime** e **MaxPW** dependem do tempo das tarefas, mas podem ser adaptadas ao ALWABP se considerarmos, por exemplo, os valores mínimos, máximos e médios dos tempos em relação a todos os trabalhadores. Neste trabalho, em particular, consideraram-se valores crescentes e decrescentes para as regras do tipo **MaxTime**. Como as duas últimas regras dependem os valores de estação mais cedo e estação mais tarde, decidiu-se

[1] Se $E_i = L_i$, uma constante de valor pequeno é atribuída a $sl_i$ (Scholl & Voß, 1996)

não aplicar esses critérios, uma vez que os limitantes de estações se encontram ainda fracos para o ALWABP (vide seção 5.2).

Foram propostas cinco novas regras que tentam refletir a estrutura do ALWABP utilizando o tempo de execução das tarefas de acordo com o trabalhador. A ideia é priorizar as tarefas cujos trabalhadores que estão sendo considerados sejam rápidos na execução das mesmas. O conjunto de todas as regras de prioridade são apresentadas e descritas a seguir, em que $\overline{w}_i = \text{argmin}_{w \in W} p_{wi}$ é o trabalhador mais rápido na execução da tarefa $i$ e $p_i^- = \min_{w \in W} p_{wi}$, $p_i^+ = \max_{w \in W} p_{wi}$, $\overline{p}_i = \sum_{w \in W} p_{wi}/|W|$ são os tempos mínimo, máximo e médio de execução da tarefa $i$, respectivamente. No cálculo de $p_i^-$, $p_i^+$ e $\overline{p}_i$, a infactibilidade do tipo trabalhador/tarefa é contornada atribuindo-se o valor do tempo de ciclo considerado $C$' (veja Algoritmo 2) ao $p_{wi}$ correspondente a esta infactibilidade.

1. **MaxF**: número decrescente de sucessores imediatos e transitivos, $|F_i^*|$;
2. **MaxIF**: número decrescente de sucessores imediatos $|F_i|$;
3. **MaxTime$^-$**: valor decrescente do tempo mínimo de execução das tarefas, $p_i^-$;
4. **MaxTime$^+$**: valor decrescente do tempo máximo de execução das tarefas, $p_i^+$;
5. **Max$\overline{\text{Time}}$**: valor decrescente do tempo médio de execução das tarefas, $\overline{p}_i$;
6. **MinTime$^-$**: valor crescente do tempo mínimo de execução das tarefas, $p_i^-$;
7. **MinTime$^+$**: valor crescente do tempo máximo de execução das tarefas, $p_i^+$;
8. **Min$\overline{\text{Time}}$**: valor crescente do tempo médio de execução das tarefas, $\overline{p}_i$;
9. **MaxPW$^-$**: valor decrescente de pesos de posição mínimos, $pw_i^- = p_i^- + \sum_{j \in F_i^*} p_j^-$;
10. **MaxPW$^+$**: valor decrescente de pesos de posição máximos, $pw_i^+ = p_i^+ + \sum_{j \in F_i^*} p_j^+$;
11. **Max$\overline{\text{PW}}$**: valor decrescente de pesos de posição médios, $\overline{pw_i} = \overline{p}_i + \sum_{j \in F_i^*} \overline{p}_j$;
12. **MinD**: valor crescente da diferença dos tempos de execução em relação ao melhor trabalhador, $p_{wi} - p_{\overline{w}i}$;
13. **MinR**: valor crescente da razão dos tempos de execução em relação ao melhor trabalhador, $p_{wi}/p_{\overline{w}i}$;
14. **MaxFTime**: valor decrescente do número de seguidores imediatos e transitivos pelo tempo de execução, $|F_i^*|/p_{wi}$;
15. **MaxIFTime**: valor decrescente do número de seguidores imediatos pelo tempo de execução, $|F_i|/p_{wi}$;
16. **MinRank**: valor crescente da "classificação" (*rank*) do tempo de procesamento das tarefas pelos trabalhadores, $|w' \in W : p_{w'i} < p_{wi}|$.

As regras **MinD** e **MinR** priorizam as tarefas às quais o trabalhador atual seja mais rápido que os outros enquanto **MaxFTime** e **MaxIFTime** priorizam dois fatores importantes: rapidez em sua execução pelo trabalhador atual e desbloqueio da linha de produção quando uma tarefa possui muitos sucessores, alocando-a o mais cedo possível. A regra **MinRank** dá preferência às tarefas com menor tempo de execução comparado com os outros trabalhadores. As outras regras seguem o raciocício original.

### 4.1.2 Regras de prioridade para os trabalhadores

Muitas das regras de prioridade propostas anteriormente analisam a eficiência de um dado trabalhador para uma tarefa em comparação com outros trabalhadores. Isso nos sugere que o Algoritmo 1 deve ser modificado para incluir um laço repetitivo adicional a fim de selecionar o melhor trabalhador disponível a ser designado para a estação atual, de acordo com uma regra de prioridade para os trabalhadores. A modificação pode ser vista no Algoritmo 3, em que $rp_w$ corresponde ao valor da regra de prioridade referente trabalhador $w$.

O algoritmo retorna o número de trabalhadores necessários para a linha. Como o número de trabalhadores é conhecido previamente, se o número de trabalhadores necessários for maior que $|W|$, o algoritmo retorna $\infty$, caracterizando uma solução infactível para o problema. Desta forma, a idéia *gulosa* por trás do algoritmo é, a cada laço das linhas 8-26, simular qual trabalhador possui o melhor valor de regra de prioridade. Desta forma, assim que o trabalhador com o melhor valor de $rp_w$ é alocado à estação de trabalho atual, as listas de tarefas e de trabalhadores disponíveis são atualizadas, como pode ser visto nas linhas 28-29. O procedimento continua até que não existam mais tarefas e trabalhadores disponíveis.

Uma estratégia análoga ao Algoritmo 2 pode ser utilizada para resolver o ALWABP-2, ou seja, basta substituir a chamada da linha 4 pela chamada ao Algoritmo 3. Da mesma forma, o Algoritmo 2 retorna um valor de tempo de ciclo para uma solução factível ou $\infty$ se nenhum valor de tempo de ciclo dentro das faixas dos limitantes levar a uma solução viável.

Foram propostas três regras de prioridades para trabalhadores, mostradas a seguir:

- **MaxTasks**: número decrescente de tarefas alocadas, $|T_w|$;
- **MinBWA**: valor crescente do tempo de ciclo em relação à melhor alocação do trabalhador, $\max_{w'} L'_w$, em que $L'_w = \sum_{i \in J'_w} p_{w'i}$ e $J'_w = \{i : w' = argmin_{w'' \in U_W \setminus \{w\}} p_{w''i}\}$;
- **MinRLB**: valor crescente do limite inferior restritivo (RLB, do inglês *restricted lower bound*), $\sum_{i \in U \setminus T_w} \min_{w' \in U_w \setminus \{w\}} p_{w'i} / (|U_w - 1|)$.

A primeira regra dá prioridade ao trabalhador que é capaz de executar o maior número de tarefas na estação sendo testada. Os últimos dois critérios calculam uma estimativa do tempo de ciclo para o problema considerando as tarefas ainda não alocadas e dão preferência ao trabalhador que reduza essa estimativa. Essas duas regras se diferem na maneira com que obtem as estimativas. A regra **MinBWA** calcula o tempo de ciclo que seria obtido se cada trabalhador ainda disponível fosse alocado às tarefas que executasse com maior rapidez (no caso de empate, a tarefa é atribuída ao trabalhador disponível com menor carga de tarefas). A regra **MinRLB** calcula o limite inferior do tempo de ciclo para as estações ainda disponíveis.

Cada uma dessas três regras de prioridade pode ser combinada com uma das 16 regras de prioridade relativas à tarefas, mostradas na seção 4.1.1. Além disso, se considerarmos que a designação de tarefas pode ser efetuada de forma *forward* (considerando a precedência original do

**Algoritmo 3** Heurística construtiva para o ALWABP-1 baseada em regras de prioridades

1: **entrada:** $C$'
2: $m \leftarrow 0$;
3: $U$: conjunto de tarefas não alocadas ($U \leftarrow N$);
4: $U_W$: conjunto de trabalhadores não alocados ($U_W \leftarrow W$);
5: **enquanto** $U \neq \varnothing$ ou $U_w \neq \varnothing$ **faça**
6: $T_{best} \leftarrow \varnothing$;
7: $rp_{best} \leftarrow \infty$ (regra de prioridade de trabalhadores decrescente) ou $rp_{best} \leftarrow 0$ (regra de prioridade de trabalhadores crescente);
8: **para** todo $w \in U_w$ **faça**
9: $c^t \leftarrow 0$;
10: calcule o conjunto de tarefas disponíveis $T_w$ para o trabalhador $w$;
11: ordene as tarefas em $T_w$ de acordo com a regra de prioridade usada, considerando o trabalhador $w$;
12: **para** cada tarefa $i \in T_w$ **faça**
13: **se** $c^t + p_{wi} < C$' **então**
14: considere essa tarefa para o trabalhador $w$:
15: $T_w \leftarrow T_w \backslash i$;
16: $c^t \leftarrow c^t + p_{wi}$;
17: atualize $T_w$ com as tarefas eventualmente disponíveis;
18: **fim**
19: **fim**
20: calcule $rp_w$
21: **se** $rp_w$ é melhor que $rp_{best}$ **então**
22: $wb \leftarrow w$;
23: $T_{best} \leftarrow T_w$;
24: $rp_{best} \leftarrow rp_w$;
25: **fim**
26: **fim**
27: aloque o trabalhador $wb$ à estação $m$:
28: $U \leftarrow U - T_{best}$;
29: $U_W \leftarrow U_W \backslash \{wb\}$;
30: $m \leftarrow m + 1$;
31: **fim**
32: **se** $U \neq \varnothing$ **então**
33: $m \leftarrow \infty$
34: **fim**
35: **saída:** número de estações necessárias: $m$.

grafo) ou *backward* (considerando a precedência reversa do grafo), um total de 96 combinações são possíveis. A eficiência dessas combinações é comparada na seção 8.2.

CAPÍTULO

5

# Busca Tabu

Busca Tabu (BT) é uma metaheurística baseada em busca local ou busca em vizinhança proposta por Glover (1989, 1990) e utilizada em vários problemas clássicos de otimização combinatória, tais como o *problema roteamento de veículos* e o *problema de localização de facilidades com capacidades* (Glover & Kochenberger, 2003). A BT explora iterativamente o espaço de busca de uma dada solução inicial $x$. A cada iteração, o algoritmo troca a solução atual por um dos seus vizinhos (a noção de vizinhança é explicada na seção 5.2). É interessante notar que não necessariamente a nova solução é melhor que a antiga (em termos de função objetivo, por exemplo).

Para tal efeito, a BT incorpora mecanismos de memória adaptativa (de curta ou longa duração) e estratégias de busca baseadas em memória. A idéia básica é a utilização de memória de curto prazo, isto é, a proibição de certos movimentos que fariam com que o algoritmo retornasse a uma solução visitada recentemente, permitindo superar ótimos locais, prevenir ciclagem e direcionar a busca para regiões não exploradas (Arroyo, 2002). Os movimentos que já foram realizados recentemente são conhecidos na literatura como movimentos *tabu* e a duração da proibição como *período tabu* (*tabu tenure*). Caso no momento da busca o movimento proibido seja considerado particularmente desejável (por exemplo, por levar a uma solução melhor que a melhor solução conhecida), a proibição pode ser violada e o movimento pode ser efetivado. Neste caso, diz-se que o movimento passou em um *critério de aspiração*.

O uso da memória de longo prazo, caracterizada pela frequência com que os atributos ocorrem na busca (quantidade de vezes com que cada trabalhador é alocado em cada estação em soluções factíveis, por exemplo), pode ser justificado pelas aplicações de procedimentos de diversificação e intensificação muito importantes na busca de soluções promissoras. De fato, é

comum que certas regiões do espaço de busca fiquem inexploradas mesmo com a proibição de certos movimentos pela lista *tabu*. Assim, nas estratégias de diversificação, movimentos que produzam soluções com atributos muito frequentes são penalizados enquanto os movimentos menos frequentes são incentivados, a fim de levar a busca por regiões inexploradas ou muito pouco exploradas até o momento.

A intensificação está ligada ao retorno da busca às regiões consideradas promissoras durante a execução da BT. Semelhante à diversificação, a intensificação se baseia em um conjunto de atributos que resultem em um determinado padrão de qualidade das soluções. Entretanto, neste caso, tais atributos são incentivados com o intuito de se realizar uma busca mais intensa em regiões promissoras.

Neste trabalho, duas versões da metaheurística BT foram investigadas. Por um lado, buscou-se o desenvolvimento de um algoritmo minimalista que combinasse características de simplicidade, flexibilidade, precisão e rapidez, de acordo com a proposta sugerida por Cordeau et al. (2002). Em contrapartida, estratégias clássicas na literatura de busca tabu, tais como intensificação e diversificação, foram incorporadas ao método, em uma tentativa de melhorar a qualidade das soluções, tendendo a aumentar, por outro lado, a complexidade do algoritmo e o tempo computacional gasto na resolução dos problemas.

Nas próximas seções, são apresentadas a representação da solução para o problema (seção 5.1), assim como as estruturas da vizinhança adotadas na implementação do algoritmo (seção 5.2). A seção 5.3 mostra a estratégia utilizada na função objetivo para contornar o problema da infactibilidade de soluções. Além disso, apresentamos a descrição do algoritmo de *busca tabu minimalista* (seção 5.4). Finalmente, procedimentos de busca local e uma abordagem estendida da BT são descritos nas seções 5.5 e 5.6, respectivamente.

## 5.1 Representação da solução

Uma solução do ALWABP pode ser codificada por dois vetores. Neste momento, usaremos um abuso de notação ($S$ já foi definido na seção 2.1 para designar o conjunto das estações de trabalho) para nos referirmos também ao primeiro vetor ($S$), que representa a alocação tarefa/estação. Já o vetor ($Tr$) representa a alocação trabalhador/estação. A Figura 5.2 mostra a representação de uma solução com 7 tarefas, 4 trabalhadores e 4 estações, originária da alocação representada pelo grafo da Figura 5.1.

## 5.2 Estruturas de vizinhança

Em um algoritmo de BT (assim como em outras metaheurísticas), é necessário definir a estrutura da vizinhança a ser explorada. Assim, uma solução vizinha foi definida como qualquer solução alcançada a partir da solução atual por meio dos seguintes movimentos:

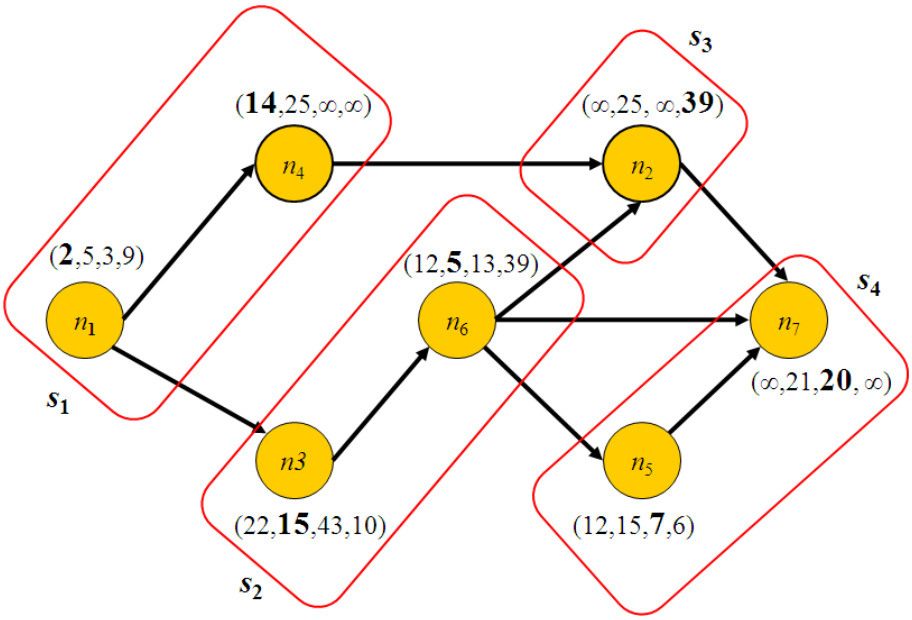


**Figura 5.1:** ALWABP-2: Solução para $m = 4$, com $C = 39$

| | $n_1$ | $n_2$ | $n_3$ | $n_4$ | $n_5$ | $n_6$ | $n_7$ |
|---|---|---|---|---|---|---|---|
| $S$ | $s_1$ | $s_3$ | $s_2$ | $s_1$ | $s_4$ | $s_2$ | $s_4$ |

| | $w_1$ | $w_2$ | $w_3$ | $w_4$ |
|---|---|---|---|---|
| $Tr$ | $s_1$ | $s_2$ | $s_4$ | $s_3$ |

**Figura 5.2:** ALWABP-2: Representação da solução referente ao grafo da Figura 5.1

- **Inserção de Tarefa**: move uma tarefa de uma estação para a outra;
- **Troca de Tarefas**: troca duas tarefas entre duas estações;
- **Troca de Trabalhadores**: troca dois trabalhadores entre duas estações.

Os três movimentos definidos acima podem produzir soluções infactíveis para o problema. Os movimentos de Inserção de Tarefa e de Troca de Tarefas podem causar tanto a infactibilidade trabalhador/tarefa quanto a infactibilidade de precedência de tarefas. Além disso, a Troca de Trabalhadores pode levar que uma tarefa seja alocada a um trabalhador que não seja capaz de executá-la.

Neste trabalho, utilizamos as equações de estação mais cedo e estações mais tarde (vide subseção 2.2.1) para delimitar as estações as quais uma tarefa pode ser inserida, estando isto diretamente ligado aos movimentos de Inserção de Tarefa e Troca de Tarefas. Para tanto, como os tempos das tarefas podem variar para cada trabalhador, definimos $p_i$ ou $p_j$ (veja as equações (2.2) e (2.3)) como o menor tempo de execução das tarefas considerando todos os trabalhado-

res e $C$ o tempo de ciclo da primeira solução factível encontrada. Com isso, reduzimos nosso espaço de busca, levando como consequência a uma diminuição do tempo computacional.

## 5.3 Função objetivo

Neste trabalho, consideramos como objetivo a ser minimizado o tempo de ciclo $C$ da linha de produção, usando adicionalmente termos de penalidade para fornecer a busca por soluções factíveis (Gendreau et al., 1994). Desta forma, uma solução do ALWABP é avaliada da seguinte forma:

$$f = C + \alpha_p \cdot InfP + \alpha_w \cdot InfT, \tag{5.1}$$

em que $\alpha_p$ e $\alpha_w$ são números reais que representam penalidades atribuídas à infactibilidade de precedência de tarefas e de trabalhador/tarefa, respectivamente. $InfP$ corresponde à possível infactibilidade com relação à precedência de tarefas, para cada estação, e $InfT$ é o valor atribuído à quantidade da possível infactibilidade (trabalhador/tarefa), para cada estação de trabalho. A fim de mensurar os valores de $InfP$ e $InfT$, foram propostas as métricas representadas pelas equações abaixo:

$$InfP = \max\left\{0, \sum_{s=1}^{|S|-1} \sum_{s'=s+1}^{|S|} \sum_{i=1}^{|N|} \sum_{j \in D_i} x_{swi} \cdot x_{s'w'j} \cdot (s' - s)\right\} \tag{5.2}$$

$$InfT = \sum_{w=1}^{|W|} \sum_{s=1}^{|S|} \sum_{i \in I_w} x_{swi} \tag{5.3}$$

A Figura 5.4 ilustra a representação de uma solução com a alocação infactível com 7 tarefas, 4 trabalhadores e 4 estações, originária da alocação representada pelo grafo da Figura 5.3.

$InfP$ acontece pois as tarefas imediatamente predecessoras da tarefa $n_2$, $n_4$ e $n_6$, estão alocadas nas estações $s_2$ e $s_3$, respectivamente, enquanto a tarefa $n_2$ está alocada à estação $s_1$. Portanto, $InfP = (2 - 1) + (3 - 1) = 3$.

$InfT$ acontece pois o trabalhador $w_3$, alocado à estação $s_1$, não consegue realizar a tarefa $n_2$ daquela estação e porque o trabalhador $w_4$, alocado à estação $s_4$, não consegue realizar a tarefa $n_7$, também alocada à estação $s_4$.

## 5.4 Busca Tabu Minimalista

Segundo Cordeau et al. (2002), simplicidade e flexibilidade são critérios que, apesar de não serem frequentemente levados em consideração para avaliação dos métodos propostos na literatura, são de suma importância uma vez que a maioria das implementações levadas a cabo em

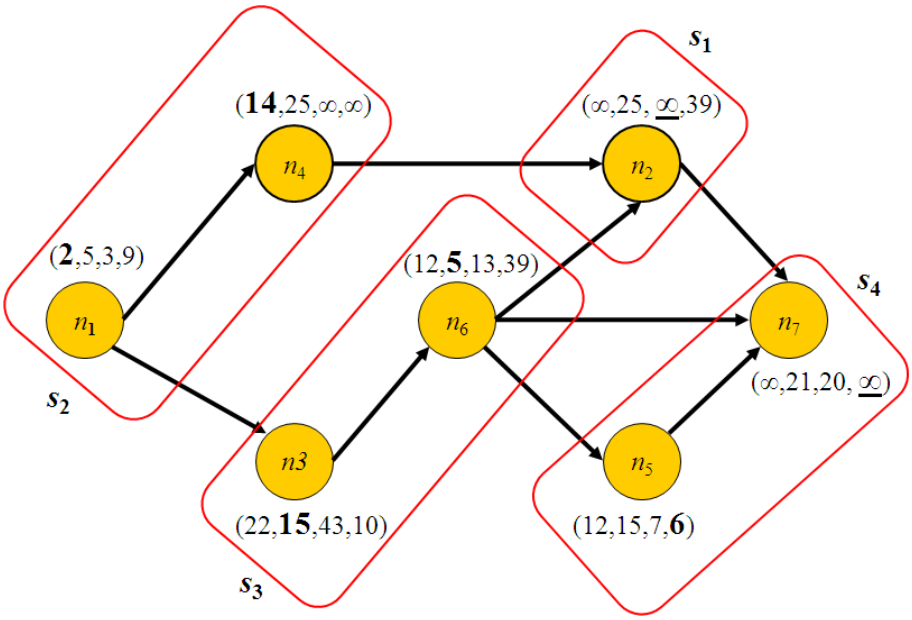


**Figura 5.3:** ALWABP-2: Solução infactível, com $InfP = 3$ e $InfT = 2$

| | $n_1$ | $n_2$ | $n_3$ | $n_4$ | $n_5$ | $n_6$ | $n_7$ |
|---|---|---|---|---|---|---|---|
| $S$ | $s_2$ | $s_1$ | $s_3$ | $s_2$ | $s_4$ | $s_3$ | $s_4$ |

| | $w_1$ | $w_2$ | $w_3$ | $w_4$ |
|---|---|---|---|---|
| $Tr$ | $s_2$ | $s_3$ | $s_1$ | $s_4$ |

**Figura 5.4:** ALWABP-2: Representação da solução referente ao grafo da Figura 5.3

contextos práticos são de algoritmos com essas características. Entretanto, a junção de simplicidade, flexibilidade, precisão e rapidez não é necessariamente fácil, já que estes são critérios conflituosos. De fato, é intuitivo que algoritmos precisos são geralmente lentos e complexos ou que algoritmos flexíveis podem não explorar algumas especificidades do problema e tendem a ser menos precisos quanto aos resultados.

Assim, criou-se uma expectativa em desenvolver um algoritmo que estabelecesse uma relação entre esses quatro fatores desejáveis, melhor especificados abaixo:

- (Simplicidade) Algoritmos devem ser fáceis de explicar e entender; algoritmos não devem confiar ao usuário um número excessivo de parâmetros a serem ajustados. Como regra geral, os parâmetros devem se auto-ajustar e/ou o algoritmo deve ser robusto a eles.
- (Flexibilidade) O algoritmo deve ser de fácil adaptação, de tal forma que extensões do problema original sejam facilmente incorporadas;
- (Precisão) O algoritmo deve ser tão preciso quanto possível, desde que não desobedeça os dois critérios anteriores;

- (Velocidade) O algoritmo deve ser tão rápido quanto possível. Entretanto, deve-se levar em conta não somente o valor absoluto do tempo gasto, mas também o contexto no qual o problema está sendo inserido (problema de planejamento em níveis estratégico, tático ou operacional).

Esses critérios possuem um caráter bem geral, não sendo específicos portanto para nenhum algoritmo em particular. Assim, motivados pela aplicatibidade desses critérios e tendo em vista que uma grande quantidade de problemas mono-objetivos já foram solucionados com êxito através da aplicação desta metaheurística (Glover & Laguna, 1997), foi proposta uma adaptação do algoritmo de BT para o ALWABP, denominado *busca tabu minimalista* (BTM).

O núcleo do algoritmo BTM proposto é apresentado na sequência, assim como as adaptações da metaheurística BT para o ALWABP. Alguns detalhes de implementação incluem a ordem dos movimentos para geração dos vizinhos (neste trabalho, a ordem adotada foi a mesma que a ordem usada para apresentar os movimentos na seção 5.2) e o critério de aspiração que permite que um possível movimento proibido seja executado (neste trabalho, o critério utilizado permite um vizinho obtido através de um movimento tabu seja a nova solução atual se ele for factível e possuir valor de função objetivo melhor do que a melhor solução factível encontrada até o momento).

A principal característica da BTM é sua simplicidade, que remete a estruturas simples e clássicas da BT, além de vizinhanças bem intuitivas. Para lidar com a flexibilidade e precisão das soluções, adotou-se o uso de penalidades na função objetivo. O uso de penalidades tem sido uma prática muito comum em Pesquisa Operacional e é a base de métodos tradicionais, como, por exemplo, a Relaxação Lagrangiana (Fisher, 2004). O específico uso de penalidades como descrito neste trabalho foi proposto por Gendreau et al. (1994) no contexto do *problema de roteamento de veículos*. A idéia mantém a simplicidade do algoritmo e consiste em permitir a infactibilidade de soluções durante a busca, penalizando-as na função objetivo conforme apresentado na equação (5.1). Existem valores de penalidade para cada subconjunto de restrições importantes e seus valores são auto-ajustados da seguinte forma: caso uma solução que respeita um conjunto de restrições é encontrada, o valor da penalidade associada a essas retrições é reduzida (dividida por 2, por exemplo). Analogamente, se a solução corrente viola um conjunto de restrições, o valor da penalidade associada é aumentado (dobrado, por exemplo). A ideia é forçar o algoritmo a percorrer regiões factíveis do espaço de busca uma vez que soluções infactíveis foram encontradas por um número crescente de iterações e permitir que o algoritmo visite soluções infactíveis (como um mecanismo de diversificação), já que o método visitou soluções factíveis por um certo número de iterações. Note que isso é feito automaticamente com mudanças dinâmicas nos valores das penalidades, sem a necessidade de definir procedimentos de factibilização (que podem ser ineficientes, difíceis de implementar e pouco flexíveis).

No caso particular do algoritmo aqui desenvolvido para o ALWABP, foram adotados valores de penalidades associados às restrições de precedência (2.23) e às designações infactíveis (traba-

lhador/tarefas), definidas pelas restrições (2.26). As infactibilidades são mensuradas do mesmo modo como apresentado na seção 5.3.

Um penalidade adicional associada ao número de iterações que o algoritmo gasta sem utilizar o movimento troca de trabalhadores também foi aplicada. A ideia desta penalidade adicional é promover a diversificação de soluções na busca, já que testes preliminares mostraram que sem esta penalidade, o movimento de troca de trabalhadores é raramente utilizado.

O pseudo-código da BTM proposta é apresentada no Algoritmo 4. Desconsideremos por enquanto as linhas destacadas em itálico na cor cinza, uma vez que se referem a funcionalidades adicionais pertencentes aos algoritmos de *busca local* (BL) e *busca tabu estendida* (BTE), explicados nas seções 5.5 e 5.6, respectivamente. Nesse algoritmo, para uma solução $s$, $f(s)$ representa a função objetivo (considerando as penalidades) e $N(s)$ representa o conjunto de vizinhos de $s$.

Algumas observações podem ser feitas acerca do algoritmo. Na linha 3, são definidos os valores iniciais associados às penalidades de precedência, $\alpha_p$, e à infactibilidade (trabalhador/tarefas), $\alpha_w$. Os valores iniciais adotados foram iguais a 60 nos testes realizados, entretanto, o algoritmo se mostrou robusto a esse parâmetro. O laço de repetição principal corresponde às linhas 4-42. Nas linhas 5-14 são feitas as atualizações dos pesos que ponderam os dois tipos de infactibilidades. As linhas 16-33 apresentam a geração da vizinhança a partir dos três movimentos considerados (veja seção 5.2). Assim, caso uma solução seja originada de um movimento considerado tabu, se a mesma for factível e melhor que a melhor solução conhecida, aplica-se o critério de aspiração e a busca se inicia a partir dessa nova solução encontrada (linhas 20-29). Caso contrário, se o movimento não for tabu, verifica-se se o novo vizinho gerado é melhor que o melhor encontrado até o momento nesta iteração (linhas 30-32).

Como se havia planejado, o algoritmo apresentou-se o mais simples possível. É importante notar que o mesmo também é flexível, no sentido que novas características do problema podem ser incorporadas com o uso do mesmo esquema de penalidades desenvolvido para restrições de infactibilidade de precedência de tarefas e de trabalhador/tarefas.

## 5.5 Busca Local

Heurísticas de busca local partem de uma solução do problema e com base em movimentos pré-estabelecidos, constroem uma vizinhança dessa solução. Escolhe-se o melhor vizinho, podendo-se começar a busca novamente desse ponto até que um mínimo local seja alcançado. Segundo Arroyo (2002), a eficiência dessa abordagem depende da solução inicial geradora da vizinhança e das regras de movimentos consideradas para gerarem as novas soluções.

Neste trabalho, aplicamos um procedimento construtivo e dois tipos de busca local na abordagem estendida da BT, que será explicada na seção 5.6. Testes preliminares mostraram que o

**Algoritmo 4** Algoritmo de *busca tabu minimalista* (BTM) para o ALWABP

1: **entrada:** $s$ (solução inicial gerada aleatoriamente ou através das heurísticas);
2: $s_b \leftarrow s$;
3: $\alpha_p \leftarrow 60, \alpha_w \leftarrow 60$;
4: **enquanto** não alcançar um critério de parada **faça**
5:  **se** $s$ é factível em relação a rede de precedência **então**
6:  $\alpha_p \leftarrow \alpha_p/2$;
7:  **senão**
8:  $\alpha_p \leftarrow 2\alpha_p$;
9:  **fim**
10:  **se** $s$ é factível em relação à (trabalhador/tarefa) **então**
11:  $\alpha_w \leftarrow \alpha_w/2$;
12:  **senão**
13:  $\alpha_w \leftarrow 2\alpha_w$;
14:  **fim**
15:  $f(b_n) \leftarrow \infty$; /* *Função objetivo do melhor vizinho até o momento* */
16:  **para** $x \in N(s)$ **faça**
17:  **se** $x$ é factível **então**
18:  *atualiza estruturas de intensificação/diversificação*;
19:  **fim**
20:  **se** $x$ foi obtido por um movimento tabu **então**
21:  **se** $x$ é factível e $f(x) < f(s_b)$ **então**
22:  *aplique BLTF(f(x))*;
23:  *aplique BLTT(x)*;
24:  *aplique BLM(x)*;
25:  aplique o critério de aspiração:
26:  $s_b \leftarrow x$;
27:  $s \leftarrow x$;
28:  **vá para** 4;
29:  **fim**
30:  **senão se** $x$ não foi obtido por um movimento tabu e $f(x) < f(b_n)$ **então**
31:  $b_n \leftarrow x$;
32:  **fim**
33:  **fim**
34:  $s \leftarrow b_n$;
35:  **se** *alcançou critério de intensificação* **então**
36:  *aplique IT*;
37:  *aplique ITr*;
38:  **senão**
39:  *aplique DT*;
40:  *aplique DTr*;
41:  **fim**
42: **fim**
43: **saída:** melhor solução encontrada: $s_b$.

melhor momento para a aplicação desses procedimentos é quando a metaheurística encontra alguma solução factível melhor que a melhor solução obtida até o momento pelo método.

O procedimento construtivo, denominado *procedimento construtivo com trabalhadores fixos* (PCTF), utiliza o algoritmo descrito na seção 4.1, com a diferença que agora os trabalhadores são alocados às estações de acordo com a solução factível utilizada como solução inicial do PCTF. Com os trabalhadores fixados nas estações, utilizamos o procedimento apresentado pelo Algoritmo 3 com a exceção do laço referente à alocação dos trabalhadores, executando as 16 regras de prioridade das tarefas descritas na seção 4.1.1 mais as duas designações do grafo de precedências (*forward* e *backward*), escolhendo assim a melhor solução proveniente das 32 execuções diferentes dessa abordagem. O Algoritmo 5 apresenta o pseudocódigo desse método.

**Algoritmo 5** Procedimento Construtivo com Trabalhadores Fixos (PCTF)

1: **entrada:** $C$'
2: $m \leftarrow 0$;
3: $U$: conjunto de tarefas não alocadas ($U \leftarrow N$);
4: $U_W$: conjunto de trabalhadores ($U_W \leftarrow W$);
5: aloque cada trabalhador em sua respectiva estação de acordo com a solução de partida do PCTF;
6: **enquanto** $U \neq \varnothing$ ou $U_W \neq \varnothing$ **faça**
7: Seja $w$ o primeiro elemento do conjunto $U_W$;
8: $c^t \leftarrow 0$;
9: calcule o conjunto de tarefas disponíveis $T_w$ para o trabalhador $w$;
10: ordene as tarefas em $T_w$ de acordo com a regra de prioridade usada, considerando o trabalhador $w$;
11: **para** cada tarefa $i \in T_w$ **faça**
12: **se** $c^t + p_{wi} < C$' **então**
13: aloque esta tarefa ao trabalhador $w$:
14: $T_w \leftarrow T_w \backslash i$;
15: $c^t \leftarrow c^t + p_{wi}$;
16: atualize $T_w$ com as tarefas eventualmente disponíveis;
17: **fim**
18: **fim**
19: $U \leftarrow U - T_w$;
20: $U_W \leftarrow U_W \backslash \{w\}$;
21: **se** $U \neq \varnothing$ ou $U_W \neq \varnothing$ **então**
22: $m \leftarrow m + 1$;
23: **fim**
24: **fim**
25: **se** $U \neq \varnothing$ **então**
26: $m \leftarrow \infty$
27: **fim**
28: **saída:** número de estações necessárias: $m$.

Para a execução desse algoritmo, basta substituir a chamada da linha 4 no Algoritmo 2 pela chamada ao Algoritmo 5. Note que o formato do Algoritmo 5 difere dos Algoritmos 6 e 7 (explicados na sequência), tanto nos parâmetros de entrada como no retorno do procedimento.

Optamos por essa escolha a fim de deixar mais perceptível ao leitor que o Algoritmo 5 é uma particularização do Algoritmo 3.

O primeiro procedimento de busca local, chamado de *busca local com troca de trabalhadores* (BLTT), gera uma vizinhança a partir dos movimentos de troca entre dois trabalhadores de estações diferentes, tal como mostrado na seção 5.2, com um adicional de que após cada movimento realizado, é feita uma chamada ao algoritmo de PCTF (linha 5 do Algoritmo 6), configurando desta forma várias execuções do PCTF com diferentes arranjos de trabalhadores. O Algoritmo 6 nos mostra como se dá essa segunda abordagem, em que $NT_w(s)$ é a vizinhança obtida pela troca de trabalhadores da solução $s$.

**Algoritmo 6** Busca Local com Troca de Trabalhadores (BLTT)

1: **entrada:** $s$ (solução inicial da busca local);
2: $s_b \leftarrow s$;
3: $f(b_n) \leftarrow f(s)$; /* *Função objetivo do melhor vizinho até o momento* */
4: **para** $x \in NT_w(s)$ **faça**
5: seja $x'$ a melhor solução obtida pelo PCTF tendo como parâmetro de entrada a solução $x$;
6: seja $y$ a melhor solução entre $x$ e $x'$;
7: **se** $y$ é factível e $f(y) < f(s_b)$ **então**
8: $s_b \leftarrow y$;
9: **fim**
10: **fim**
11: **saída:** melhor solução encontrada: $s_b$.

A segunda heurística de busca local investigada foi denominada *busca local para melhoramento* (BLM). Essa metodologia consiste em aplicar os seguintes movimentos de vizinhança:

1. Inserção de tarefas: retira uma tarefa de uma estação crítica (sem *idle time*) e a coloca em outra;
2. Troca de tarefas;
3. Troca de trabalhadores;
4. Troca de duas tarefas em cadeia ("*enchained*"): faz dois movimentos de troca consecutivos antes de avaliar a mudança.

A regra é aplicada (e o processo recomeça) caso a BLM diminua o tempo de ciclo ou mantenha o tempo de ciclo mas diminua o número de estações críticas da solução inicial (linhas 11-14 do Algoritmo 7). O Algoritmo 7 apresenta os passos para a aplicação dessa busca local, em que $N'(s)$ representa o conjunto de vizinhos gerados a partir dos quatro movimentos anteriormente definidos.

**Algoritmo 7** Busca Local para Melhoramento (BLM)

1: **entrada:** $s$ (solução inicial da busca local);
2: $s_b \leftarrow s$;
3: $f(b_n) \leftarrow f(s)$; /* *Função objetivo do melhor vizinho até o momento* */
4: seja $nc$ o número de estações críticas de $s$;
5: **para** $x \in N'(s)$ **faça**
6:  **se** $x$ é factível e $f(x) < f(s_b)$ **então**
7:  $s_b \leftarrow x$;
8:  **fim**
9: **fim**
10: seja $nc'$ o número de estações críticas de $s_b$;
11: **se** $nc' < nc$ ou $f(s_b) < f(s)$ **então**
12:  $s \leftarrow x$;
13:  vá para o passo 2;
14: **fim**
15: **saída:** melhor solução encontrada: $s_b$.

## 5.6 Busca Tabu *Estendida*

No Algoritmo 4, considerando agora as linhas em itálico na cor cinza, temos a abordagem estendida da busca tabu. Caso a BTE encontre uma nova solução melhor que a melhor global, aplicam-se os três procedimentos descritos na seção 5.5, substituindo essa nova solução caso alguma dessas funções a consiga melhorar (linhas 22-24). Ao fim do algoritmo, pode-se observar o uso dos métodos de intensificação e diversificação, utilizados de forma alternada de acordo com um critério de aceitação (linhas 35-41). A seguir, explicamos com mais detalhes o funcionamento desses dois procedimentos.

Estratégias de intensificação e diversificação são aplicadas em várias adaptações da metaheurística BT. Em linhas gerais, trata-se de explorar regiões que se mostram promissoras ou que foram pouca ou nenhuma vez exploradas durante algum tempo na busca. Para tanto, neste trabalho, utilizamos uma estrutura que nos auxilia a determinar quais atributos durante a execução da BT mais se repetiram ou menos ocorreram.

Desta forma, sempre que uma solução factível é encontrada durante algum movimento da vizinhança, atualiza-se uma estrutura que contabiliza quantas vezes um trabalhador executou uma determinada tarefa e quantas vezes um trabalhador foi alocado em uma certa estação de trabalho em soluções factíveis (linhas 17-19). Assim sendo, foi definido que a intensificação e a diversificação originariam quatro procedimentos distintos, apresentados a seguir:

- Intensificação tarefas (IT): consiste em fixar em cada estação as tarefas que ali mais se repetiram. Caso haja empate, prioriza-se a tarefa de menor índice;
- Intensificação trabalhadores (ITr): consiste em fixar em cada estação o trabalhador que mais aparecereu durante o período de busca considerado. Caso haja empate, prioriza-se o trabalhador de menor índice;

- Diversificação tarefas (DT): consiste em fixar em cada estação as tarefas que menos se repetiram. Caso haja empate, prioriza-se a tarefa de menor índice;
- Diversificação trabalhadores (DTr): consiste em fixar em cada estação o trabalhador que menos aparecereu durante o período de busca considerado. Caso haja empate, prioriza-se o trabalhador de menor índice.

Tanto na IT e na DT, fixam-se os trabalhadores da solução atual nas suas respectivas estações e posteriormente, aplica-se a BLM (vide seção 5.5) nesta nova solução. Já nos procedimentos de ITr e DTr, após a fixação dos trabalhadores, a PCTF é executada com os arranjos de trabalhadores nas estações de acordo com a intensificação ou diversificação. Note que tanto na intensificação quanto na diversificação, a solução da próxima iteração corresponde à melhor solução factível obtida por cada procedimento.

CAPÍTULO

6

# GRASP

A metaheurística GRASP (*greedy randomized adaptative search procedure*) é um método de múltiplos inícios, proposta e definida formalmente por Feo & Resende (1989, 1995). Cada iteração do GRASP é composta pelas fases construtiva e de busca local. A primeira fase constrói iterativamente uma solução factível para o problema. Em seguida, essa solução é melhorada por meio de um procedimento de busca local, que constitui a segunda fase dessa metaheurística. A seguir, apresentamos um pseudocódigo genérico do GRASP, que ilustra sua estrutura global. Neste pseudo-código, $s$ corresponde a uma solução do problema e $g(.)$ a uma função critério que escolhe os candidatos da *lista de candidatos restritos* ($LCR$), que será definida na seção 6.1.

**Algoritmo 8** GRASP

1: $s \leftarrow \varnothing$;
2: $f_b \leftarrow \infty$;
3: $s_b \leftarrow \varnothing$;
4: **para** $it \leftarrow 0 \ldots MAX$ **faça**
5: **FaseConstrutiva**$(s, g(.))$;
6: **BuscaLocal**$(s)$;
7: **se** $f(s) < f_b$ **então**
8: $s_b \leftarrow s$;
9: $f_b \leftarrow f(s)$;
10: **fim**
11: **fim**
12: **saída:** melhor solução $s_b$ encontrada pelo algoritmo.

Festa & Ribeiro (2002) e Resende & Ribeiro (2003) apresentam algumas variações dessa metaheurística para a resolução de diferentes problemas de otimização combinatória. Entre as mais conhecidas, podemos citar os desenvolvimentos feitos para os problemas de *scheduling* (Gupta & Smith, 2006; Leone et al., 2010; Goodman et al., 2009; Armentano & Araújo, 2006), projeto de redes (Scaparra & Church, 2005), *packing* (Delorme et al., 2004), dimensionamento de lotes (Nascimento et al., 2010b), designação (Mavridou et al., 1998), agrupamento (*clustering*) (Deng & Bard, 2010; Nascimento et al., 2010a), roteamento de veículos (Duhamel et al., 2011) e linhas de produção (Alpay, 2009).

Nossa motivação diante do GRASP se deu devido aos bons resultados obtidos pelas heurísticas construtivas propostas para o ALWABP (veja seção 8.2). A expectativa é que a parcela de aleatoriedade inserida pelo GRASP possa levar essas heurísticas a obter soluções diversificadas e de boa qualidade. Nas próximas seções, apresentaremos as duas fases que compõem o GRASP proposto para o ALWABP. A seção 6.1 descreve o procedimento de construção da solução inicial de cada iteração, enquanto a seção 6.2 apresenta a heurística de busca local aplicada à solução gerada pela fase anterior.

## 6.1 Fase Construtiva

A fase construtiva do GRASP tem como principal objetivo a geração de uma solução inicial combinando aspectos de qualidade e aleatoriedade, de forma a se situar em regiões diversas e promissoras do espaço de busca. A solução correspondente à iteração atual é construída de forma gradual, a partir de uma lista de candidatos $L$ contendo todos os possíveis elementos que podem fazer parte da solução $s$. Nesta abordagem, a lista $L$ corresponde às tarefas que estão disponíveis a serem alocadas na linha (isto é, tarefas cujas predecessoras já foram alocadas) e cujo tempo de execução somado ao tempo da estação atual não excede o tempo de ciclo desejado.

Através da lista $L$, a $LCR$ é construída. Essa nova lista é composta pelos elementos com os melhores custos incrementais, determinados de acordo com uma função gulosa $g : L \rightarrow \mathbb{R}$. A função $g$ mede o benefício de se escolher cada elemento da lista $L$. Segundo Resende (1998), o método se mostra adaptativo pois esse benefício relacionado a cada elemento se modifica a cada iteração da construção, sendo que a seleção de um candidato interfere na escolha dos próximos que serão inseridos em $s$. A aleatoriedade se dá no sentido de que qualquer elemento de $LCR$ poderá ser selecionado, não sendo necessariamente escolhido aquele de melhor custo incremental. A seguir, no Algoritmo 9, apresentamos o pseudocódigo referente à fase construtiva do GRASP, em que $\alpha \in [0, 1]$ é um parâmetro que controla o nível de aleatoriedade do algoritmo.

No GRASP proposto para o ALWABP, devido ao fato do número de tarefas disponíveis ser frequentemente limitado, adotou-se $\alpha = 1.0$, de modo a introduzir suficiente aleatoriedade no

**Algoritmo 9** Fase Construtiva do GRASP

1: **entrada:** $s$, $g(.)$
2: $s \leftarrow \varnothing$;
3: seja $L$ a lista de candidatos (tarefas) a serem inseridos na solução $s$;
4: **enquanto** $L \neq \varnothing$ **faça**
5: $c_{min} \leftarrow \text{MIN}\{g(i) : i \in L\}$;
6: $c_{max} \leftarrow \text{MAX}\{g(i) : i \in L\}$;
7: $LCR : \{i \in L : g(i) \leq c_{min} + \alpha(c_{max} - c_{min})\}$;
8: selecione aleatoriamente $i \in LCR$;
9: $s \leftarrow s \cup \{i\}$;
10: $LCR \leftarrow LCR \backslash \{i\}$;
11: $L \leftarrow L \backslash \{i\}$;
12: **fim**
13: **saída:** solução $s$ construída pelo algoritmo.

processo de busca. Com relação à função $g$, podemos representá-la por algum dos critérios heurísticos apresentados na seção 4.1. Tendo como base os resultados da seção 8.2, optou-se por executar o GRASP com os três critérios que obtiveram melhores resultados: MaxPW$^-$, MaxPW$^+$ e Max$\overline{\text{PW}}$, com as designações do grafo de precedência *forward* e *backward*. Desta forma, podemos especificar o Algoritmo 8 da maneira mostrada no Algoritmo 10.

**Algoritmo 10** GRASP

1: $s \leftarrow \varnothing$;
2: $f_b \leftarrow \infty$;
3: $s_b \leftarrow \varnothing$;
4: $H$: conjunto de heurísticas que serão utilizadas como função critério na escolha de tarefas na $LCR$;
5: $H \leftarrow \{\text{MaxPW}^-, \text{MaxPW}^+, \text{Max}\overline{\text{PW}}\}$;
6: **para** $h \in H$ **faça**
7: **para** $it \leftarrow 0 \ldots MAX$ **faça**
8: **FaseConstrutiva**$(s, h)$;
9: **BuscaLocal**$(s)$;
10: **se** $f(s) < f_b$ **então**
11: $s_b \leftarrow s$;
12: $f_b \leftarrow f(s)$;
13: **fim**
14: **fim**
15: **fim**
16: **saída:** melhor solução $s_b$ encontrada pelo algoritmo.

## 6.2 Busca Local

A busca local aplicada após fase construtiva do GRASP visa uma melhora qualitativa na solução gerada pelo Algoritmo 9. Para tanto, neste trabalho, adotamos uma vizinhança composta pelos

três movimentos apresentados na seção 5.2: inserção de tarefas, troca de tarefas e troca de trabalhadores.

A busca é feita partindo-se da solução obtida na fase construtiva, através da aplicação sucessiva dos movimentos, na ordem apresentada. Após a execução de todos os movimentos definidos, caso o melhor vizinho factível encontrado seja melhor que a melhor solução atual, atualizamos esta com o novo valor de tempo de ciclo encontrado por esse vizinho.

CAPÍTULO

7

# Rotação de tarefas

Em linhas de produção convencionais, a rotação de tarefas é prática corrente e parte do pressuposto de que a troca das tarefas designadas a cada trabalhador ao longo do dia pode aumentar o desempenho e a satisfação dos empregados. Tal troca pode contribuir para que os trabalhadores tenham uma visão mais ampla do processo e sejam mais generalistas (Ouchi, 1981; Champion et al., 1994), além de ajudar a prevenir doenças comuns do trabalho, tais como estresse muscular e lesões por esforço repetitivo (Michalos et al., 2010).

No caso de trabalhadores deficientes, este último aspecto assume uma importância ainda maior, podendo a rotação de tarefas ser, em última instância, encarada com um viés terapêutico. Assim, uma designação adequada de tarefas a um dado trabalhador poderia ajudá-lo a desenvolver habilidades latentes. Considerando-se o fato de que um dos objetivos de centros de trabalho para deficientes é a preparação dos trabalhadores para o mercado convencional, a rotação de tarefas assumiria um papel chave, onde a exposição de um trabalhador a novas tarefas poderia contribuir para o seu desenvolvimento profissional e a sua preparação para a integração em linhas convencionais.

A dificuldade, entretanto, reside no fato de que as imensas diferenças entre os trabalhadores torna praticamente inviável a estratégia simples adotada no caso SALBP de obter uma configuração inicial e trocar a posição dos trabalhadores mantendo-se as tarefas fixas.

Neste projeto, visamos obter um método mais geral para obtenção da rotação de tarefas para o caso ALWABP. De fato, a ideia central consiste em dividir o período de planejamento em sub-períodos e obter, para cada sub-período, uma solução de balanceamento utilizando um dos diversos métodos analisados ao longo deste trabalho. A ideia é que ao fim do período, cada trabalhador tenha sido exposto ao maior número possível de tarefas diferentes, métrica já utilizada

na literatura (Costa & Miralles, 2009). Para isso, foi proposto um *algoritmo híbrido* (AH) que primeiramente cria um *pool* de soluções de maneira heurística e depois faz uso de um modelo de otimização linear inteiro misto para selecionar as soluções que, em conjunto, respeitem os níveis de qualidade exigidos e maximizem a métrica desejada. Um procedimento de busca local foi também proposto com o objetivo de fazer uma ligação entre as abordagens heurística e exata desenvolvidas.

Nas próximas seções, apresentaremos o modelo proposto por Costa & Miralles (2009) para o problema de rotação de tarefas em linhas de produção em CTD's (seção 7.1), além do algoritmo híbrido proposto para a resolução desse problema (seção 7.2).

## 7.1 Formulação matemática

O problema de rotação de tarefas em linhas de produção com trabalhadores deficientes foi proposto por Costa & Miralles (2009). Como mencionado anteriormente, os autores desenvolveram uma métrica baseada no número de tarefas executadas por cada trabalhador, partindo do pressuposto que quanto maior a exposição dos trabalhadores a tarefas de caráter diferente, mais se atingiria o objetivo do CTD de preparar tais trabalhadores para o mercado convencional.

No contexto desta métrica, o problema de minimização de tempo de ciclo aparece na forma de restrições no modelo, com o objetivo de definir níveis de produção aceitáveis. Desta forma, um gerente poderia escolher um nível de *redução de produtividade* adequado (segundo, por exemplo, a demanda do dia ou da semana) e utilizar o excesso de capacidade para capacitar os trabalhadores através de estratégias de rotação de tarefas. O modelo proposto pelos autores leva em conta esta filosofia e é apresentado na sequência. De modo simplificado, o modelo consiste na repetição do modelo (2.20)-(2.28) para os diversos subperíodos e a inclusão de restrições limitando as perdas produtivas aceitáveis decorrentes da rotação.

$$\text{Max } z = \sum_{w \in W} \sum_{i \in N} z_{wi} \tag{7.1}$$

sujeito a

$$\sum_{w \in W} \sum_{s \in S} x_{swit} = 1, \qquad \forall i \in N, t = 1, ..., T, \tag{7.2}$$

$$\sum_{s \in S} y_{swt} = 1, \qquad \forall w \in W, t = 1, ..., T, \tag{7.3}$$

$$\sum_{w \in W} y_{swt} = 1, \qquad \forall s \in S, t = 1, ..., T, \tag{7.4}$$

$$\sum_{w \in W} \sum_{s \in S} s \cdot x_{swit} \leq \sum_{w \in W} \sum_{s \in S} s \cdot x_{swjt}, \qquad \forall j \in N, \forall i \in D_j, t = 1, ..., T, \tag{7.5}$$

$$\sum_{i \in N} p_{wi} \cdot x_{swit} \leq C_t, \qquad \forall w \in W, \forall s \in S, t = 1, ..., T, \tag{7.6}$$

$$\sum_{i \in N} x_{swit} \leq N y_{swt}, \qquad \forall w \in W, \forall s \in S, t = 1, ..., T, \tag{7.7}$$

$$\sum_{t=1}^{T} C_t \leq TC', \tag{7.8}$$

$$z_{wi} \leq \sum_{t=1}^{T} \sum_{s \in S} x_{swi}, \qquad \forall w \in W, \forall i \in N, \tag{7.9}$$

$$x_{swit} = 0, \qquad \forall w \in W, \forall s \in S, \forall i \in I_w, t = 1, ..., T, \tag{7.10}$$

$$y_{swt} \in \{0, 1\}, \qquad \forall s \in S, \forall w \in W, t = 1, ..., T, \tag{7.11}$$

$$x_{swit} \in \{0, 1\}, \qquad \forall s \in S, \forall w \in W, \forall i \in N, t = 1, ..., T. \tag{7.12}$$

em que,

| | |
|---|---|
| $T$ | número de subperíodos, |
| $t$ | índice para subperíodo de rotação, |
| $C_t$ | tempo de ciclo do subperíodo $t$, |
| $x_{swit}$ | variável binária. Igual a 1 apenas se a tarefa $i$ é alocada ao trabalhador $w$ na estação $s$ no subperíodo $t$, |
| $y_{sw}$ | variável binária. Igual a 1 apenas se o trabalhador $w$ é alocado à estação $s$ no subperíodo $t$, |
| $z_{wi}$ | variável binária. Igual a 1 apenas se o trabalhador $w$ executa a tarefa $i$ em pelo menos um dos subperíodos, |
| $C'$ | tempo de ciclo desejado. |

No modelo, são usadas variáveis $x_{swi}$ e $y_{sw}$ com a inserção de um índice temporal, a fim de representar cada subperíodo. A nova varíavel binária produzida, $z_{wi}$, é igual a 1 se o trabalhador $w$ executa a tarefa $i$ em pelo menos um dos subperíodos do período completo de rotação de tarefas.

A função objetivo (7.1) maximiza o número de tarefas diferentes executadas por cada trabalhador. As restrições (7.2)-(7.7) garantem que as restrições originais do ALWABP são respeitadas

para cada subperíodo. As restrições de acoplamento (7.8) e (7.9) estabelecem que o tempo de ciclo médio da solução não deve ultrapassar o valor desejado $C'$ e que as variáveis $z_{wi}$ representam, de fato, a execução de uma tarefa $i$ pelo trabalhador $w$, respectivamente.

## 7.2 Algoritmo Híbrido

Um *algoritmo híbrido* combina duas ou mais características de outros métodos em um só procedimento. As abordagens encontradas mais comumente na literatura consistem em métodos que englobam mais de uma estrátegia heurística ou combinam métodos exatos com heurísticos. A ideia é, em geral, aproveitar características distintas dos procedimentos e permitir que juntos possam contornar dificuldades observadas quando são abordados de forma isolada. Otimizadores híbridos podem ser vistos com mais detalhes em (El-Abd & Kamel, 2005; Puchinger & Raidl, 2005; Maniezzo et al., 2009; Günther et al., 2010). Doerner & Schmid (2010) apresentam uma revisão de métodos híbridos aplicados ao *problema de roteamento de veículos* (PRV). Segundo esses autores, atualmente, as duas principais linhas de pesquisa dentro desse contexto consistem em algoritmos de integração e algoritmos colaborativos.

Algoritmos de integração são aqueles que possuem uma função específica para resolução do problema, não são aplicados de forma isolada, em que cada componente complementa cada etapa realizada anteriormente. Um exemplo dessa abordagem pode ser visto em (Archetti et al., 2008), aplicado ao PRV. O método proposto por esses autores consiste na integração entre busca através de heurística e resolução exata da formulação do problema. Desta forma, um modelo inteiro é usado para explorar parte do espaço de busca das soluções identificado como promissor por uma busca tabu.

Algoritmos colaborativos são aqueles cujos componentes podem ser executados de forma individual, trocando mutuamente informações e resolvendo o problema de forma conjunta, muitas vezes de maneira paralela ou independente. Schmid (2007) e Schmid et al. (2009) estudam uma variação do PRV aplicada à indústria de concreto. O algoritmo proposto realiza trocas de informações entre o modelo inteiro de fluxo de rede multiproduto (*multicommodity network flow*) e a metaheurística VNS (*variable neighborhood search*), a fim de encontrar soluções de boa qualidade em um tempo computacional razoável, obtendo resultados muito bons para o problema abordado. Considerando o PRV com janelas de tempo, Bouthillier & Crainic (2005) desenvolvem um algoritmo que executa heurísticas em paralelo com a estratégia de armazenamento das melhores soluções (*solution warehouse strategy*) durante a busca. Tal abordagem se mostrou muito eficiente e precisa, apresentando resultados satisfatórios.

Recentemente, a literatura cunhou o termo *matheuristics* para designar algoritmos de otimização projetados através da interação de metaheurísticas e técnicas de programação matemática. Note que as *matheuristics* podem se enquadar tanto na classe dos métodos de integração quanto nos métodos colaborativos. Alguns trabalhos fizeram uso dessa abordagem de maneira similar

àquela desenvolvida neste trabalho. Parragh et al. (2009) propõem uma combinação do método de geração de colunas e o algoritmo VNS para o PRV heterogêneo do tipo *dial-a-ride* com restrições relacionadas ao condutor do veículo. Deste modo, novas colunas são inseridas no problema mestre a partir da geração de colunas e de soluções encontradas pelo VNS. Comparações feitas com a geração de colunas pura mostraram que o algoritmo proposto é bem superior. Outros exemplos de métodos que utilizam abordagens exatas e heurísticas podem ser encontrados em (Fischetti & Lodi, 2003; Hansen et al., 2006; Hashimoto et al., 2008; Prins et al., 2007).

Neste trabalho, vamos focar na resolução do problema de maneira integrativa, por meio de uma combinação de métodos heurísticos e uma formulação matemática. O ALWABP em linhas de produção com o recurso de rotação de tarefas é um problema com dois objetivos conflitantes, isto é, se por um lado queremos que os trabalhadores executem o maior número de tarefas possível dentro de um período completo de rotação, por outro lado desejamos manter níveis de produtividade na linha. Deste modo, propomos um método que tenta conciliar esses objetivos de forma que o índice de produtividade seja respeitado enquanto o número de tarefas diferentes executadas pelos trabalhadores é maximizado.

O *algoritmo híbrido para rotação de tarefas* (AHRT) proposto para essa extensão do ALWABP utiliza as heurísticas construtivas (seção 4.1) e as metaheurísticas BTE (seção 5.6), GRASP (capítulo 6) e HGA (*hybrid genetic algorithm*). Uma breve descrição deste último algoritmo é dada na seção 8.5. As soluções factíveis obtidas durante a busca desses quatro métodos foram armazenadas em uma estrutura global, numa espécie de *pool* de soluções, utilizando-se da idéia de *warehouse solutions* (Rochat & Taillard, 1995). A partir desse conjunto, um modelo linear inteiro auxiliar é aplicado a fim de selecionar as soluções que maximizem o número de tarefas executadas pelos trabalhadores respeitando as restrições do problema de rotação de tarefas original. Posteriormente, uma busca local baseada no modelo apresentado na seção 7.1 é aplicada, tendo como ponto inicial a solução retornada pelo modelo linear auxiliar.

Nas seções que seguem, abordaremos os seguintes pontos em relação ao AHRT. A subseção 7.2.1 mostra como ocorre o gerenciamento da estrutura global de soluções. A subseção 7.2.2 apresenta o modelo linear que resolve o problema com base na estrutura global. Na subseção 7.2.3, apresentamos a busca local desenvolvida para esse problema. Finalmente, a subseção 7.2.4 nos mostra o pseudocódigo referente AHRT proposto.

### 7.2.1 Gerenciamento de soluções factíveis

O conjunto de soluções $P$ pode ser definido como uma estrutura global que armazena, durante a fase de execução dos métodos heurísticos, as soluções factíveis encontradas. Desta forma, sempre que uma solução dessa natureza é encontrada, tenta-se a inserção neste *pool* de soluções. O conjunto $P$ é composto da seguinte forma:

1. $n_{HGA}$ soluções factíveis da última geração do algoritmo HGA são inseridas em $P$ (o número de indivíduos da população do algoritmo HGA é 100);
2. 16 soluções obtidas por cada uma das heurísticas construtivas propostas, considerando alocação *forward*, *backward* e a melhor configuração de escolha dos trabalhadores (seção 4.1);
3. soluções factíveis resultantes da aplicação da BTE, inclusive aquelas obtidas durante procedimentos de busca local (seção 5.5) e intensificação/diversificação (seção 5.6);
4. soluções factíveis resultantes da aplicação do algortimo GRASP, tanto aquelas obtidas durante a fase construtiva quanto as melhores soluções da fase de busca local (capítulo 6).

Note que não é possível inserir todos os indivíduos encontrados durante essa busca, uma vez que $P$ tem tamanho limitado e não pode ser muito grande, já que isso interfere na eficiência do modelo linear auxiliar do AHRT (veja subseção 7.2.2). Assim sendo, caso $P$ já tenha chegado ao seu limite de soluções, uma solução de $P$ que não seja proveniente dos itens 1 e 2 é escolhida ao acaso e substituída pela solução candidata a entrar no conjunto.

### 7.2.2 Modelo linear para o problema de rotação de tarefas

A seguir, apresentamos o modelo linear inteiro auxiliar, componente do AHRP proposto.

$$\text{Max } z = \sum_{w \in W} \sum_{i \in N} z_{wi} \tag{7.13}$$

sujeito a

$$\sum_{k \in P} C_k \cdot \beta_k \leq TC', \tag{7.14}$$

$$z_{wi} \leq \sum_{k \in P} a_{kwi} \cdot \beta_k, \qquad \forall w \in W, \forall i \in N, \tag{7.15}$$

$$\sum_{k \in P} \beta_k = T, \tag{7.16}$$

$$z_{wi} \in \{0, 1\}, \qquad \forall w \in W, \forall i \in N, \tag{7.17}$$

$$\beta_k \in \{0, 1\}, \qquad \forall k \in P. \tag{7.18}$$

em que,

$k$ índice para soluções pertencentes a $P(k = 1, ..., |P|)$

$C_k$ tempo de ciclo da solução $k$ pertencente a $P$,

$a_{kwi}$ parâmetro de entrada binário. Igual a 1 apenas se a tarefa $i$ é alocada ao trabalhador $w$ na solução $k$,

$\beta_k$ variável binária. Igual a 1 apenas se a solução $k$ for escolhida para o período completo de rotação ,

$z_{wi}$ variável binária. Igual a 1 apenas se o trabalhador $w$ executa a tarefa $i$ em pelo menos um.

O objetivo desta nova formulação (7.13) continua sendo a maximização do número de tarefas diferentes executadas pelos trabalhadores durante um período completo de rotação. As restrições (7.14) garantem que as restrições de acoplamento (7.8) sejam respeitadas. Com relação as restrições (7.15), pode-se dizer que são similares às restrições (7.9), uma vez que caso o trabalhador $w$ não execute a tarefa $i$, o valor do somatório é igual a zero e força a variável $z_{wi}$ associada a ser zero, já que nenhuma solução $k \in P$ tal que $a_{kwi} = 1$ pode ser escolhida. As restrições (7.16) estabelecem que $T$ soluções devem ser escolhidas, uma para cada subperíodo da rotação de tarefas. Note que os valores de $C_k$ e $a_{kwi}$ são parâmetros de entrada do modelo, obtidos através da execução das heurísticas construtivas e das metaheurísticas BTE, GRASP e HGA.

### 7.2.3 Busca Local

Durante todo o processo de busca de soluções a serem inseridas no conjunto $P$, não consideramos em nenhum momento o número de tarefas diferentes que os trabalhadores executam durante o período de rotação. De fato, nesta etapa, nos preocupamos apenas com as soluções que respeitem as precedências de tarefas e as factibilidades trabalhador/tarefa em cada estação. Em seguida, o modelo linear apresentado na seção anterior trata de escolher as soluções que maximizem a função objetivo do problema de rotação de tarefas e que respeitem as restrições de acoplamento, mas sem ter a liberdade para alterar as soluções de cada período.

Desta forma, observou-se a necessidade de desenvolver um procedimento que pudesse alterar as soluções em cada período e, ao mesmo tempo, considerar o verdadeira função objetivo do problema, tratando-a de forma direta mas sempre tomando o cuidado de respeitar as restrições originais. O modelo apresentado na seção 7.1 se mostra muito ineficiente para a resolução desse problema (Costa & Miralles, 2009), inclusive para pequenas instâncias, já que um índice referente a cada subperíodo é inserido nas variáveis $x_{swi}$ e $y_{sw}$, o que faz com que o número de variáveis aumente muito. Entretanto, este modelo pode ser útil se alguma redução no espaço de busca for efetuada. Desta forma, a fim de permitir mudanças locais nas soluções obtidas pelo modelo (7.13)-(7.18), foram propstas duas *buscas locais para a rotação de tarefas*, denominadas BLRT1 e BLRT2.

O Algoritmo 11 apresenta a BLRT1 proposta. Esse procedimento possui como entrada o conjunto $P'$ de $T$ soluções obtidas atrávés do modelo (7.13)-(7.18). Duas estratégias foram adotadas para o funcionamento dessa busca local. Primeiramente, a resolução do modelo inicia com a solução resultante do modelo linear auxiliar (veja seção 7.2.2), a fim de fornecer um melhor limitante inferior à formulação. Posteriormente, foram inseridos dois tipos de redução no espaço de busca. O primeiro deles (linhas 3-21) consiste em só permitir que uma tarefa, em um dado período, seja alocada à posição original (resultante da resolução do modelo (7.13)-(7.18)) ou a estações imediatamente vizinhas a ela. Adicionalmente, nas linhas 22-34, estabelece-se que um trabalhador deve permanecer alocado na estação original. Como consequência, é proibida a designação de quaisquer tarefas ao trabalhador $w$, na estação $s$, referente ao subperíodo $t$ (linhas 28-30).

O Algoritmo 12 apresenta os passos da BLRT2. Neste caso, a ideia consiste em fixar $t - 1$ períodos, tanto em relação às tarefas quanto aos trabalhadores nas estações (linhas 10-20), e deixar que o modelo aloque da melhor forma as tarefas em um período determinando, fixando a alocação dos trabalhadores neste período (linhas 6-9). Deste forma, faz-se todas as combinações entre os períodos livres e fixos durante a busca. Note que neste procedimento possui como entrada o conjunto de soluções obtido pela aplicação da BLRT1, a fim de começar a resolução do modelo (7.1)-(7.12) com um bom limitante inferior.

### 7.2.4 Pseudocódigo do AHRT

Os passos do AHRT proposto são mostrados na sequencia, por meio do Algoritmo 13.

**Algoritmo 11** Busca Local para a Rotação de Tarefas 1 (BLRT1)

1: **entrada:** Conjunto $P'$ ($T$ soluções obtidas atrávés do modelo linear da seção 7.2.2);
2: considere a formulação (7.1)-(7.12);
3: **para** $t \leftarrow 1 \ldots T$ **faça**
4:  **para** $s \in S$ **faça**
5:  **para** $w \in W$ **faça**
6:  **para** $i \in N$ **faça**
7:  seja $s'$ a estação a qual a tarefa $i$ está alocada na $t$-ésima solução de $P'$;
8:  seja $w'$ o trabalhador alocado na estação $s$ da $t$-ésima solução de $P'$;
9:  **se** $s' \neq s$ **ou** $w' \neq w$ **então**
10:  **se** ($s' > 1$ **e** $s' < |S|$) **e** ($s' \neq s - 1$) **e** ($s' \neq s + 1$) **então**
11:  adicione ao modelo (7.1)-(7.12) a restrição $x_{swit} = 0$;
12:  **senão se** ($s' = 0$) **e** ($s' \neq s + 1$) **então**
13:  adicione ao modelo (7.1)-(7.12) a restrição $x_{swit} = 0$;
14:  **senão se** ($s' = |S| - 1$) **e** ($s' \neq s - 1$) **então**
15:  adicione ao modelo (7.1)-(7.12) a restrição $x_{swit} = 0$;
16:  **fim**
17:  **fim**
18:  **fim**
19:  **fim**
20:  **fim**
21: **fim**
22: **para** $t \leftarrow 1 \ldots T$ **faça**
23:  **para** $s \in S$ **faça**
24:  **para** $w \in W$ **faça**
25:  seja $w'$ o trabalhador alocado na estação $s$ da $t$-ésima solução de $P'$;
26:  **se** $w' \neq w$ **então**
27:  adicione ao modelo (7.1)-(7.12) a restrição $y_{swt} = 0$;
28:  **para** $i \in N$ **faça**
29:  adicione ao modelo (7.1)-(7.12) a restrição $x_{swit} = 0$;
30:  **fim**
31:  **fim**
32:  **fim**
33:  **fim**
34: **fim**
35: resolva o modelo (7.1)-(7.12) com as restrições adicionais;
36: **nTasks** $\leftarrow \sum_{w \in W} \sum_{i \in N} z_{wi}$;
37: **saída:** número de tarefas diferentes **nTasks** executadas pelos trabalhadores durante a rotação.

**Algoritmo 12** Busca Local para a Rotação de Tarefas 2 (BLRT2)

1: **entrada:** Conjunto $P'$ ($T$ soluções obtidas atrávés da BLRT1);
2: considere a formulação (7.1)-(7.12);
3: considere $N_{st}$ o conjunto de tarefas alocadas à estação $s$ na $t$-ésima solução de $P'$;
4: **nBTasks** $\leftarrow 0$;
5: **para** $t \leftarrow 1 \ldots T$ **faça**
6:  **para** $s \in S$ **faça**
7:  seja $w'$ o trabalhador alocado à estação $s$ na $t$-ésima solução de $P'$;
8:  adicione ao modelo (7.1)-(7.12) a restrição $y_{sw'} = 1$;
9:  **fim**
10:  **para** $t' \leftarrow 1 \ldots T$ **faça**
11:  **se** $t \neq t'$ **então**
12:  **para** $s \in S$ **faça**
13:  seja $w'$ o trabalhador alocado à estação $s$ na $t'$-ésima solução de $P'$;
14:  adicione ao modelo (7.1)-(7.12) a restrição $y_{sw'} = 1$;
15:  **para** $i \in N_{st'}$ **faça**
16:  adicione ao modelo (7.1)-(7.12) a restrição $x_{sw'i} = 1$;
17:  **fim**
18:  **fim**
19:  **fim**
20:  **fim**
21:  resolva o modelo (7.1)-(7.12) com as restrições adicionais;
22:  **nTasks** $\leftarrow \sum_{w \in W} \sum_{i \in N} z_{wi}$;
23:  **se nBTasks < nTasks então**
24:  **nBTasks** $\leftarrow$ **nTasks**;
25:  **fim**
26:  considere o modelo (7.1)-(7.12) sem as restrições adicionais;
27: **fim**
28: **saída:** número de tarefas diferentes **nBTasks** executadas pelos trabalhadores durante a rotação.

**Algoritmo 13** Algoritmo Híbrido para a Rotação de Tarefas (AHRT)

1: $P$: conjunto de soluções factíveis encontradas durante a busca heurística;
2: $P \leftarrow \emptyset$;
3: $P_b \leftarrow \emptyset$ ($|P_b| = T$);
4: $f_b \leftarrow 0$;
5: adiocione a $P$ todas as soluções factíveis obtidas da última geração do HGA;
6: aplique as 16 heurísticas construtivas (seção 4.1) e armazene as 16 melhores soluções em $P$;
7: aplique a BTE;
8: aplique o GRASP;
9: seja $f$ a função objetivo obtida pelo modelo (7.13)-(7.18);
10: seja $P'$ o conjunto das soluções tal que $\sum_{k \in P} \beta_k = 1$; /* *soluções designadas aos subperíodos* */
11: $f_{bl1} \leftarrow$ BLRT1($P'$);
12: **se** $f < f_{bl1}$ **então**
13: $\quad f \leftarrow f_{bl1}$;
14: **fim**
15: seja $P''$ o conjunto das soluções obtido pela aplicação da BLRT1;
16: $f_{bl2} \leftarrow$ BLRT2($P''$);
17: **se** $f < f_{bl2}$ **então**
18: $\quad f \leftarrow f_{bl2}$;
19: **fim**
20: **saída:** $f$ (número de tarefas diferentes executadas pelos trabalhadores durante a rotação de tarefas).

CAPÍTULO

8

# Resultados computacionais

Este capítulo apresenta os resultados computacionais referentes aos métodos propostos neste trabalho. A seção 8.1 mostra as características das instâncias teste utilizadas para a coleta dos dados. A descrição das propriedades dos problemas considerados é baseada em grande parte no trabalho de Chaves (2009) e de Miralles et al. (2007). A seção 8.2 apresenta os resultados das heurísticas construtivas propostas. A seção 8.3 apresenta os resultados das duas abordagens de busca tabu propostas para o ALWABP. Na seção 8.4, é medida a eficiência das heurísticas construtivas como ponto inicial de algoritmos mais sofisticados. A seção 8.5 possui uma breve descrição dos algoritmos CS e HGA, propostos na literatura e utilizados nos experimentos computacionais. Já a seção 8.6 compara as abordagens propostas com os melhores métodos da literatura do ALWABP. Finalmente, a seção 8.7 apresenta os resultados do algoritmo híbrido aplicado à extensão de rotação de tarefas do ALWABP.

## 8.1 Instâncias teste

Por se tratar de um problema novo na literatura, ainda não existe um conjunto de instâncias próprias geradas especificamente para o ALWABP. Devido a isso, são geradas instâncias baseadas em um grupo de problemas clássicos propostos para o SALBP, disponíveis no sítio específico para problemas de balanceamento de linhas de produção (`http://www.assembly-line-balancing.de/`). Para tanto, utilizou-se problemas de natureza bem distinta, com diferentes números de tarefas e trabalhadores e assim como a densidades da rede de precedências (*order strength*).

A rede de precedência dos problemas originais foi mantida, além dos tempos de execução das tarefas para o primeiro trabalhador. Os novos tempos de execução das tarefas foram aleatoriamente gerados de acordo com Miralles et al. (2007). A geração das instâncias considerou:

1. a relação entre o número de tarefas e de trabalhadores;
2. a variabilidade dos tempos da execução das tarefas para cada trabalhador;
3. percentual de infactibilidades (trabalhador/tarefa).

Os tempos de execução de cada tarefa foram aleatoriamente gerados segundo uma distribuição uniforme com a variação selecionada de acordo com o tempo de execução das tarefas pelo primeiro trabalhador ($p_{1i}$) (Chaves, 2009). A partir daí, foram obtidos dois níveis de variabilidade (Var) da distribuição dos tempos, um pequeno (L1) e um grande (H3), considerando U[1,$p_{1i}$] e U[1,3$p_{1i}$], respectivamente. O percentual de incompatibilidade (Inc) trabalhador/tarefa foi definido como 10% (I10) e 20% (I20). A Tabela 8.1 sumariza os problemas utilizados, assim como a variações dos fatores adotados.

| **Família** | $\|N\|$ | $\|W\|$ | **Var** | **Inc** | **Order strength** |
|---|---|---|---|---|---|
| Roszieg | 25 | 4;6 | L1;H3 | I10;I20 | 71,67 |
| Heskia | 28 | 4;7 | L1;H3 | I10;I20 | 22,49 |
| Tonge | 70 | 10;17 | L1;H3 | I10;I20 | 59,42 |
| Wee-mag | 75 | 11;19 | L1;H3 | I10;I20 | 22,67 |

**Tabela 8.1:** Detalhes sobre as instâncias do ALWABP-2

Como critério de comparação da eficiência dos métodos, utilizou-se como medida da qualidade de soluções o desvio médio ($\mathrm{D}_{avg}$) e o desvio máximo ($\mathrm{D}_{max}$) das soluções em relação aos melhores resultados conhecidos na literatura, os quais são ótimos para as famílias de instância Roszieg e Heskia (Blum & Miralles, 2011). Para as famílias Tonge e Wee-Mag, os melhores valores são definidos como o mínimo entre os valores encontrados nos trabalhos de Chaves et al. (2009), Moreira & Costa (2009) e Blum & Miralles (2011). A fórmula para o cálculo de cada uma dessas medidas é apresentada a seguir:

$$\mathrm{D}_{avg}\ (\%) = \frac{\sum_{k=1}^{n_{inst}} \frac{C_k - C_k^{best}}{C_k^{best}}}{n_{inst}} \times 100\% \tag{8.1}$$

$$\mathrm{D}_{max}\ (\%) = \max_{k=1,\ldots,n_{inst}} \left\{ \frac{C_k - C_k^{best}}{C_k^{best}} \right\} \times 100\% \tag{8.2}$$

em que $C_k$ é o tempo de ciclo obtido por algum método referente a instância $k$, $C_k^{best}$ é o melhor valor de tempo de ciclo conchecido para a instância $k$ e $n_{inst}$ o número total de instâncias que estão sendo comparadas. Além dos resultados referentes às soluções, apresentamos também o tempo computacional médio (t) e máximo ($\mathrm{t}_{max}$) (ambos em segundos) gasto pelos algoritmos.

Sob as considerações apresentadas acima, todos os algoritmos propostos foram implementados na linguagem C++ e executados utilizando o compilador g++ 4.4, no sistema operacional Linux Ubuntu. Os testes foram feitos em um computador Core 2 Duo com processador de 2.2 GHz e 3.0 Gb de RAM. Quando modelos lineares inteiros foram aplicados, utilizou-se o pacote CPLEX 12.1, com limite de tempo na resolução de 3600 segundos para cada modelo.

## 8.2 Heurísticas Construtivas

Talbot et al. (1986) mostram que para o SALBP, empates entre tarefas com a mesma prioridade tem um efeito significativo na qualidade das soluções. Desta forma, foram testadas várias regras de desempate, tanto considerando a seleção de tarefas e de trabalhadores. Para a seleção das tarefas, o número descrescente de sucessores imediatos (**MaxIF**) foi utilizado como primeiro nível de desempate e como segundo nível, o tempo de execução da tarefa $p_{wi}$ pelo trabalhador atual $w$. Para a seleção do trabalhador, como primeiro critério de desempate, utilizou-se o valor do **MinRLB** para as regras **MaxTask** e **MinBWA** e **MaxTask** para a regra **MinRLB**. Como segundo nível de desempate, foi considerado o nível de ociosidade presente em uma estação (*worstation idle time*). Tanto para seleção de tarefas e de trabalhadores, o índice original da tarefa e do trabalhador foram considerados como terceiro critério de desempate, a fim de tornar a heurística determinística.

Um detalhe importante de implementação está no fato de algumas regras de prioridade poderem ser aplicadas de maneira estática ou dinâmica em relação à alocação dos trabalhadores nas estações. Como exemplo, podemos dizer que as regras **MinD** e **MinR**, assim como todas as outras regras de prioridade que dependem dos cálculos de mínimo, máximo e média em relação aos trabalhadores disponíveis, estão sujeitas a variações diante da redução do conjunto de trabalhadores não-alocados. Teste preliminares mostraram que estratégias dinâmicas que trabalham apenas com os trabalhadores disponíveis se mostraram mais eficientes e foram usadas em todos os testes subsequentes.

Utilizou-se a seguinte estimativa para o limite inferior do tempo de ciclo: $\max\{\sum_i p_i^*/|W|, \max\{p_i^*\}\}$, em que $p_i^*$ é o menor tempo de execução da tarefa $i$ considerando todos os trabalhadores. Para uma estimativa do limite superior, foi utilizado o valor constante 500, obtido através de uma majoração dos melhores resultados apresentados até hoje na literatura.

Considerando como um critério inicial de avaliação o número de soluções factíveis encontrado pelas heurísticas, podemos afirmar que todas foram eficazes neste aspecto, ou seja, todas encontraram soluções factíveis para todas as 320 instâncias testadas.

Considerando a qualidade dessas soluções, uma conclusão a que se pode chegar é que para todas as combinações possíveis de regras de prioridade de tarefas, o critério de seleção de trabalhadores **MinRLB** foi superior aos critérios **MaxTasks** e **MinBWA**, os quais foram descartados dessa análise. Os resultados referentes às 16 regras de prioridade de tarefas, a regra **MinRLB**

**Tabela 8.2:** Comparação de 16 regras de prioridade dos trabalhadores, uma regra de prioridade do trabalhador e dois esquemas de alocação sobre as 320 instâncias.

| Alocação | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | RP tarefas | MaxF | MaxIF | MaxTime$^-$ | MaxTime$^+$ | Max$\overline{\text{Time}}$ | MinTime$^-$ | MinTime$^+$ | Min$\overline{\text{Time}}$ |
| Forward | $D_{avg}$ (%) | 25,5% | 30,8% | 26,6% | 30,9% | 28,0% | 55,5% | 49,4% | 53,4% |
| | $D_{max}$ (%) | 108,0% | 145,5% | 132,0% | 132,0% | 108,0% | 158,5% | 150,0% | 140,0% |
| | t (s) | 0,02 | 0,02 | 0,02 | 0,02 | 0,02 | 0,02 | 0,02 | 0,02 |
| | $t_{max}$ (s) | 0,08 | 0,08 | 0,08 | 0,10 | 0,10 | 0,10 | 0,13 | 0,10 |
| | RP tarefas | MaxPW$^-$ | MaxPW$^+$ | Max$\overline{\text{PW}}$ | MinD | MinR | MaxFTime | MaxIFTime | MinRank |
| | $D_{avg}$ (%) | 17,0% | 20,2% | 19,0% | 27,2% | 25,6% | 28,5% | 30,8% | 23,4% |
| | $D_{max}$ (%) | 108,0% | 108,0% | 108,0% | 136,0% | 136,0% | 132,0% | 145,5% | 112,0% |
| | t (s) | 0,04 | 0,04 | 0,03 | 0,02 | 0,02 | 0,02 | 0,02 | 0,03 |
| | $t_{max}$ (s) | 0,26 | 0,19 | 0,17 | 0,09 | 0,10 | 0,11 | 0,10 | 0,17 |
| | RP tarefas | MaxF | MaxIF | MaxTime$^-$ | MaxTime$^+$ | Max$\overline{\text{Time}}$ | MinTime$^-$ | MinTime$^+$ | Min$\overline{\text{Time}}$ |
| Backward | $D_{avg}$ (%) | 33,0% | 39,7% | 32,2% | 34,7% | 31,0% | 53,9% | 49,9% | 55,2% |
| | $D_{max}$ (%) | 218,2% | 372,7% | 177,8% | 177,8% | 125,9% | 203,7% | 272,7% | 203,7% |
| | t (s) | 0,02 | 0,02 | 0,02 | 0,02 | 0,02 | 0,03 | 0,03 | 0,03 |
| | $t_{max}$ (s) | 0,09 | 0,09 | 0,11 | 0,10 | 0,10 | 0,11 | 0,11 | 0,15 |
| | RP tarefas | MaxPW$^-$ | MaxPW$^+$ | Max$\overline{\text{PW}}$ | MinD | MinR | MaxFTime | MaxIFTime | MinRank |
| | $D_{avg}$ (%) | 26,2% | 28,7% | 27,8% | 31,8% | 29,2% | 36,9% | 40,0% | 28,0% |
| | $D_{max}$ (%) | 218,2% | 213,6% | 200,0% | 372,7% | 177,8% | 372,7% | 372,7% | 372,7% |
| | t (s) | 0,06 | 0,05 | 0,04 | 0,02 | 0,02 | 0,02 | 0,02 | 0,03 |
| | $t_{max}$ (s) | 0,27 | 0,23 | 0,23 | 0,11 | 0,10 | 0,10 | 0,12 | 0,15 |
| Melhor de todas as regras | $D_{avg}$ (%) | 9,6% | | | | | | | |
| | $D_{max}$ (%) | 55,6% | | | | | | | |
| | t (s) | 0,83 | | | | | | | |
| | $t_{max}$ (s) | 3,66 | | | | | | | |

para seleção dos trabalhadores e as estratégias *forward* e *backward* são apresentados na Tabela 8.2.

A partir da Tabela 8.2, podemos concluir que as heurísticas obtiveram soluções factíveis para o ALWABP em tempos computacionais muito baixos, não excedendo 0,27 segundo no pior caso. A qualidade da solução é bem sensível à regra de prioridade utilizada, sendo regras baseadas em *positional weights* as que apresentaram os melhores resultados. De fato, o critério **MaxPW**$^-$ conseguiu desvios médios de 17,0 % e 26,2 % para as estratégias *forward* e *backward* apresentadas na Tabela 8.2, respectivamente. Considerando a melhor solução para todas as regras, obtivemos um desvio médio de 9,6 %, com o tempo computacional médio menor que 1 segundo. A aplicação da busca local descrita na seção 5.5 apresentou apenas ganhos marginais na qualidade das soluções heurísticas.

Para se ter uma idéia da relevância desses resultados, Chaves (2009) utiliza o *software* comercial CPLEX versão 10.1 para encontrar limitantes para as 320 instâncias. Devido à natureza *NP-hard* do problema, o CPLEX encontrou grandes dificuldades para resolver as instância maiores. Para as instâncias Tonge, o CPLEX termina a execução após diversas horas de cálculo em razão da insuficiência de memória do computador em 79 das 80 instâncias testadas. O mesmo problema ocorre nas instâncias da família Wee-mag, só que em 68 das 80 instâncias.

## 8.3 Busca Tabu

Na Tabela 8.3, são apresentados os resultados dos algoritmos BTM e BTE descritos no capítulo 5. Considerou-se uma solução aleatória como inicial em ambos os métodos, que desta forma, foram denominados BTM-Rand e BTE-Rand.

O critério de parada utilizado pelas versões da BT foi um número pré-determinado de iterações, que no nosso caso foi igual a 20000. Além disso, o tamanho da lista tabu foi determinado da seguinte forma: caso a melhor solução da iteração corrente for gerada por movimentos de Inserção de Tarefas ou por Troca de Tarefas, o movimento fica proibido por $n/4$ iterações; caso contrário, isto é, se essa solução for gerada por um movimento de Troca de Trabalhadores, o movimento fica proibido por $n/2$ iterações. Os critérios de intensificação/diversificação também tiveram como base o número de iterações igual a 3000. Desta forma, a cada 3000 iterações, intensificação e diversificação alternam a execução na BTE proposta. Com o objetivo de demonstrar a robustez do algoritmos, as versões da BT foram executadas 20 vezes e os desvios calculados através das médias das 20 repetiçoes.

Além dos resultados referentes ao desvio médio ($D_{avg}$) e tempo computacional médio (t), considerou-se também o desvio médio correspondente à melhor solução obtida ($D_b$) e o tempo que encontrar as melhores soluções ($t_b$) nas 20 execuções de cada instância.

Para a família Roszieg, podemos ver que ambas as versões da BT obtiveram resultados equivalentes em todos os grupos de instâncias, com os valores médios de $D_b$ e $D_{avg}$ iguais a 0,0%.

**Tabela 8.3:** Comparação dos algoritmos BTM-Rand e BTE-Rand (melhores desvios em negrito).

| **Grupo** | **BTM-Rand** | | | | **BTE-Rand** | | | |
|---|---|---|---|---|---|---|---|---|
| | $D_b$ (%) | $D_{avg}$ (%) | t (s) | $t_b$ (s) | $D_b$ (%) | $D_{avg}$ (%) | t (s) | $t_b$ (s) |
| Roszieg 1 | **0,0%** | **0,0%** | 2,9 | 0,1 | **0,0%** | **0,0%** | 3,3 | 0,1 |
| Roszieg 2 | **0,0%** | **0,0%** | 3,1 | 0,1 | **0,0%** | **0,0%** | 3,8 | 0,3 |
| Roszieg 3 | **0,0%** | **0,0%** | 2,7 | 0,1 | **0,0%** | **0,0%** | 3,1 | 0,2 |
| Roszieg 4 | **0,0%** | **0,0%** | 2,7 | 0,0 | **0,0%** | **0,0%** | 3,0 | 0,1 |
| Roszieg 5 | **0,0%** | **0,0%** | 3,5 | 0,2 | **0,0%** | **0,0%** | 4,2 | 0,4 |
| Roszieg 6 | **0,0%** | **0,0%** | 3,5 | 0,2 | **0,0%** | **0,0%** | 4,6 | 0,7 |
| Roszieg 7 | **0,0%** | **0,0%** | 3,5 | 0,1 | **0,0%** | **0,0%** | 4,5 | 0,5 |
| Roszieg 8 | **0,0%** | **0,2%** | 3,5 | 0,3 | **0,0%** | **0,2%** | 4,5 | 0,5 |
| **Média** | **0,0%** | **0,0%** | **3,2** | **0,1** | **0,0%** | **0,0%** | **3,9** | **0,3** |
| Heskia 1 | **0,0%** | 0,4% | 3,7 | 0,4 | **0,0%** | **0,3%** | 6,7 | 2,0 |
| Heskia 2 | **0,0%** | **0,1%** | 3,7 | 0,3 | **0,0%** | **0,1%** | 7,4 | 2,3 |
| Heskia 3 | **0,0%** | **0,0%** | 3,6 | 0,3 | **0,0%** | **0,0%** | 7,3 | 2,2 |
| Heskia 4 | **0,0%** | 0,2% | 3,6 | 0,6 | **0,0%** | **0,1%** | 7,6 | 2,7 |
| Heskia 5 | **0,0%** | 7,6% | 5,0 | 0,7 | **0,0%** | **5,1%** | 10,4 | 4,8 |
| Heskia 6 | **1,3%** | 4,4% | 5,0 | 0,8 | **1,3%** | **3,5%** | 10,0 | 3,6 |
| Heskia 7 | **0,0%** | **2,0%** | 4,9 | 0,6 | **0,0%** | 2,3% | 11,2 | 5,0 |
| Heskia 8 | 0,2% | 5,2% | 5,0 | 0,8 | **0,0%** | **4,1%** | 12,3 | 6,1 |
| **Média** | **0,2%** | 2,5% | 4,3 | 0,6 | **0,2%** | **1,9%** | 9,1 | 3,6 |
| Tonge 1 | 9,7% | 14,8% | 40,2 | 17,2 | **4,1%** | **12,4%** | 154,0 | 106,6 |
| Tonge 2 | 7,6% | 16,1% | 41,1 | 17,4 | **5,2%** | **14,1%** | 257,6 | 202,7 |
| Tonge 3 | 4,1% | 12,2% | 40,8 | 17,7 | **3,0%** | **10,3%** | 264,1 | 216,2 |
| Tonge 4 | 6,5% | 15,8% | 40,1 | 15,7 | **4,7%** | **14,5%** | 267,1 | 221,0 |
| Tonge 5 | 18,1% | 29,8% | 49,8 | 26,6 | **12,3%** | **24,0%** | 327,0 | 242,7 |
| Tonge 6 | 13,8% | 24,3% | 50,3 | 25,5 | **9,2%** | **20,7%** | 409,1 | 327,6 |
| Tonge 7 | 8,3% | 17,3% | 49,7 | 25,4 | **6,2%** | **14,1%** | 571,9 | 487,8 |
| Tonge 8 | 8,7% | 19,9% | 51,1 | 28,2 | **6,8%** | **16,4%** | 667,9 | 589,2 |
| **Média** | 9,6% | 18,8% | 45,4 | 21,7 | **6,4%** | **15,8%** | 364,8 | 299,2 |
| Wee-mag 1 | 23,7% | 35,2% | 53,0 | 29,9 | **19,3%** | **32,1%** | 124,0 | 63,7 |
| Wee-mag 2 | 22,4% | 35,1% | 51,2 | 28,3 | **20,6%** | **30,7%** | 144,7 | 83,4 |
| Wee-mag 3 | 19,6% | 30,2% | 50,7 | 28,0 | **19,0%** | **28,4%** | 189,8 | 131,7 |
| Wee-mag 4 | 18,2% | 29,3% | 50,6 | 26,5 | **16,8%** | **26,6%** | 190,0 | 127,5 |
| Wee-mag 5 | 83,3% | 118,4% | 61,8 | 25,7 | **41,5%** | **69,0%** | 176,4 | 63,2 |
| Wee-mag 6 | 74,7% | 120,1% | 62,3 | 26,4 | **38,0%** | **61,2%** | 174,4 | 65,5 |
| Wee-mag 7 | 63,9% | 94,8% | 61,6 | 30,6 | **31,0%** | **53,9%** | 251,6 | 130,8 |
| Wee-mag 8 | 55,1% | 90,6% | 61,9 | 29,9 | **30,1%** | **51,0%** | 250,2 | 123,6 |
| **Média** | 45,1% | 69,2% | 56,7 | 28,2 | **27,0%** | **44,1%** | 187,6 | 98,7 |
| **Média final** | 13,7% | 22,6% | | | **8,4%** | **15,5%** | | |

Os dois algoritmos se diferiram pouco em relação à família Heskia (0,6% em relação ao $D_{avg}$ final), à exceção do grupo Heskia 5, em que a BTE obteve $D_{avg}$ inferior em 2,5% .

Nas famílias Tonge e Wee-mag, a metaheurística BTE mostrou-se bem superior à BTM, em todos os grupos de instâncias das duas famílias, com destaque para os grupos Tonge 5, Wee-mag 7 e Wee-mag 8. Na média final, a BTE obteve valores de $D_b$ e $D_{avg}$ iguais a 8,4% e 15,5%, enquanto a BTM produziu soluções com $D_b$ e $D_{avg}$ iguais a 13,7% e 22,6%. Com base nessas constatações a respeito da qualidade dos resultados, podemos ver que as heurísticas de busca local e as estratégias de intensificação/diversificação tiveram grande influencia no desempenho da melhor versão da BT.

Os tempos computacionais para a família Roszieg foram praticamente iguais, se diferindo mais no grupo de instâncias Heskia. Nas famílias Tonge e Wee-mag, pode-se observar a influencia dos algoritmos de busca local no tempo computacional dos métodos, uma vez com uma solução inicial aleatória, existe uma maior chance de execução desses procedimentos, já que sempre são chamadas quando a solução atual é factível e melhor que a melhor solução conhecida (veja Algoritmo 4).

## 8.4 Heurísticas Construtivas como solução inicial

As heurísticas construtivas para o ALWABP, apresentadas no capítulo 4, são utilizadas como solução inicial das metaheurísticas BTM, BTE e CS (Chaves, 2009), a fim de mostrar as vantagens de se inicializar tais metaheurísticas com soluções de boa qualidade obtidas rapidamente. Para tanto, com base nos resultados da seção 8.2, optou-se pela escolha da heurística baseada na regra de prioridade $MaxPW^-$ e a melhor solução considerando todas as heurísticas, a qual chamamos de BH (*best heuristic*), gerando portanto os algoritmos BTM-$MaxPW^-$, BTM-BH, BTE-$MaxPW^-$, BTE-BH, CS-$MaxPW^-$ e CS-BH.

Outras duas versões do algortimo de BT foram propostas. Nestas abordagens, a BT começa com a melhor das soluções heurísticas (BH) e após 10000 iterações, executa-se um procedimento de reinício e a busca é retomada a partir de uma solução aleatória. Assim, considerou-se em nossa comparação os algoritmos BTM-BH-Rand e BTE-BH-Rand. Como esses dois métodos sofrem o efeito da aleatoriedade e também são comparados com os algoritmos BTM-Rand e BTE-Rand, convencionou-se a execução de 20 vezes para cada instância do ALWABP por uniformidade com a literatura (Chaves, 2009). A Tabela 8.4 nos mostra os resultados obtidos por esses algoritmos. É importante observar que como os algoritmos BTM-$MaxPW^-$, BTM-BH, BTE-$MaxPW^-$ e BTE-BH são determinísticos, levamos em conta somente a coluna de desvio médio percentual $D_b$.

Primeiramente, consideramos os algoritmos derivados da BTM. A partir dos resultados envolvendo a qualidade das soluções, podemos afirmar que a utilização das heurísticas construtivas como solução inicial influencia muito na obtenção de bons resultados, já que, por exemplo, o

**Tabela 8.4:** Comparação dos algoritmos BTM-Rand, BTM-MaxPW$^{-}$, BTM-BH e BTM-BH-Rand (melhores desvios em negrito).

| **Grupo** | **BTM-Rand** | | | | **BTM-MaxPW$^{-}$** | | | **BTM-BH** | | | **BTM-BH-Rand** | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $D_b$ (%) | $D_{avg}$ (%) | t (s) | $t_b$ (s) | $D_b$ (%) | t (s) | $t_b$ (s) | $D_b$ (%) | t (s) | $t_b$ (s) | $D_b$ (%) | $D_{avg}$ (%) | t (s) | $t_b$ (s) |
| Roszieg 1 | **0,0%** | **0,0%** | 2,9 | 0,1 | **0,0%** | 2,2 | 0,0 | **0,0%** | 2,1 | 0,0 | **0,0%** | **0,0%** | 2,9 | 0,0 |
| Roszieg 2 | **0,0%** | **0,0%** | 3,1 | 0,1 | **0,0%** | 2,8 | 0,0 | **0,0%** | 2,5 | 0,0 | **0,0%** | **0,0%** | 3,4 | 0,0 |
| Roszieg 3 | **0,0%** | **0,0%** | 2,7 | 0,1 | **0,0%** | 2,2 | 0,0 | **0,0%** | 1,9 | 0,1 | **0,0%** | **0,0%** | 2,5 | 0,1 |
| Roszieg 4 | **0,0%** | **0,0%** | 2,7 | 0,0 | **0,0%** | 2,2 | 0,0 | **0,0%** | 1,9 | 0,0 | **0,0%** | **0,0%** | 2,4 | 0,0 |
| Roszieg 5 | **0,0%** | **0,0%** | 3,5 | 0,2 | **0,0%** | 2,8 | 0,1 | **0,0%** | 2,4 | 0,0 | **0,0%** | **0,0%** | 3,0 | 0,1 |
| Roszieg 6 | **0,0%** | **0,0%** | 3,5 | 0,2 | **0,0%** | 2,9 | 0,3 | **0,0%** | 2,6 | 0,1 | **0,0%** | **0,0%** | 3,3 | 0,2 |
| Roszieg 7 | **0,0%** | **0,0%** | 3,5 | 0,1 | **0,0%** | 2,9 | 0,0 | **0,0%** | 2,5 | 0,0 | **0,0%** | **0,0%** | 3,3 | 0,1 |
| Roszieg 8 | 0,0% | 0,2% | 3,5 | 0,3 | 0,0% | 2,8 | 0,1 | 0,0% | 2,6 | 0,0 | 0,0% | 0,0% | 3,3 | 0,1 |
| **Média** | **0,0%** | 3,2% | 0,1 | 0,7 | **0,0%** | 2,6% | 0,1 | **0,0%** | 2,3 | 0,0 | **0,0%** | **0,0%** | 3,0 | 0,1 |
| Heskia 1 | **0,0%** | 0,4% | 3,7 | 0,4 | 1,0% | 3,5 | 0,2 | **0,0%** | 3,1 | 0,4 | 0,1% | **0,1%** | 4,4 | 0,2 |
| Heskia 2 | **0,0%** | **0,1%** | 3,7 | 0,3 | 0,1% | 3,5 | 0,1 | 0,1% | 3,0 | 0,0 | 0,1% | **0,1%** | 4,6 | 0,1 |
| Heskia 3 | **0,0%** | **0,0%** | 3,6 | 0,3 | **0,0%** | 3,0 | 0,2 | **0,0%** | 3,0 | 0,0 | **0,0%** | **0,0%** | 4,2 | 0,2 |
| Heskia 4 | **0,0%** | 0,2% | 3,6 | 0,6 | 0,1% | 3,1 | 0,3 | 0,1% | 3,0 | 0,1 | 0,1% | **0,1%** | 4,4 | 0,3 |
| Heskia 5 | **0,0%** | 7,6% | 5,0 | 0,7 | 4,0% | 4,3 | 0,4 | 1,2% | 4,1 | 0,3 | 0,8% | **1,1%** | 5,8 | 0,2 |
| Heskia 6 | 1,3% | 4,4% | 5,0 | 0,8 | 2,5% | 4,6 | 0,8 | 1,5% | 4,5 | 0,4 | **1,1%** | **1,4%** | 6,2 | 0,6 |
| Heskia 7 | **0,0%** | 2,0% | 4,9 | 0,6 | 2,2% | 4,7 | 0,1 | 0,6% | 4,4 | 0,1 | 0,6% | **0,6%** | 6,0 | 0,2 |
| Heskia 8 | **0,2%** | 5,2% | 5,0 | 0,8 | 3,2% | 4,8 | 0,5 | 2,3% | 4,2 | 0,0 | 2,1% | **2,3%** | 6,0 | 0,1 |
| **Média** | **0,2%** | 2,5% | 4,3 | 0,6 | 1,6% | 3,9 | 0,3 | 0,7% | 3,7 | 0,2 | 0,6% | **0,7%** | 5,2 | 0,2 |
| Tonge 1 | 9,7% | 14,8% | 40,2 | 17,2 | 11,8% | 50,3 | 18,9 | **8,8%** | 32,2 | 4,8 | 9,6% | **9,6%** | 51,1 | 4,9 |
| Tonge 2 | **7,6%** | 16,1% | 41,1 | 17,4 | 15,1% | 50,5 | 14,8 | 11,7% | 33,5 | 3,5 | 12,1% | **12,1%** | 53,0 | 7,3 |
| Tonge 3 | **4,1%** | 12,2% | 40,8 | 17,7 | 11,4% | 50,7 | 8,0 | 9,5% | 34,5 | 6,7 | 9,7% | **9,7%** | 54,5 | 7,5 |
| Tonge 4 | **6,5%** | 15,8% | 40,1 | 15,7 | 12,3% | 49,7 | 26,5 | 10,5% | 32,3 | 7,0 | 11,1% | **11,1%** | 54,9 | 5,6 |
| Tonge 5 | 18,1% | 29,8% | 49,8 | 26,6 | 18,7% | 39,6 | 2,1 | **9,3%** | 41,1 | 0,0 | **9,3%** | **9,3%** | 62,0 | 3,4 |
| Tonge 6 | 13,8% | 24,3% | 50,3 | 25,5 | 12,1% | 39,6% | 7,8 | **7,5%** | 40,8 | 0,0 | **7,5%** | **7,5%** | 65,1 | 3,7 |
| Tonge 7 | **8,3%** | 17,3% | 49,7 | 25,4 | 11,1% | 41,2 | 7,3 | 11,0% | 40,5 | 0,0 | 11,0% | **11,0%** | 68,3 | 5,3 |
| Tonge 8 | **8,7%** | 19,9% | 51,1 | 28,2 | 18,6% | 42,4 | 16,3 | 9,0% | 39,6 | 7,1 | 9,7% | **9,7%** | 68,4 | 8,5 |
| **Média** | **9,6%** | 18,8% | 45,4 | 21,7 | 13,9% | 12,7 | 21,7 | 9,7% | 36,8 | 3,6 | 10,0% | **10,0%** | 59,7 | 5,8 |
| Wee-mag 1 | 23,7% | 35,2% | 53,0 | 29,9 | 19,7% | 46,1 | 6,7 | **15,1%** | 46,7 | 0,0 | **15,1%** | **15,1%** | 69,1 | 1,2 |
| Wee-mag 2 | 22,4% | 35,1% | 51,2 | 28,3 | 21,4% | 47,0 | 1,5 | **18,4%** | 47,3 | 0,0 | **18,4%** | **18,4%** | 72,4 | 1,5 |
| Wee-mag 3 | **19,6%** | 30,2% | 50,7 | 28,0 | 22,7% | 46,7 | 10,0 | 20,4% | 48,1 | 3,9 | 21,3% | **21,3%** | 69,6 | 2,3 |
| Wee-mag 4 | **18,2%** | 29,3% | 50,6 | 26,5 | 20,3% | 46,8 | 2,4 | 20,3% | 49,5 | 0,0 | 20,3% | **20,3%** | 73,2 | 2,4 |
| Wee-mag 5 | 83,3% | 118,4% | 61,8 | 25,7 | 21,1% | 64,1 | 0,0 | **16,3%** | 56,4 | 0,0 | **16,3%** | **16,3%** | 85,1 | 1,8 |
| Wee-mag 6 | 74,7% | 120,1% | 62,3 | 26,4 | 20,5% | 65,3 | 0,0 | **14,6%** | 57,6 | 0,0 | **14,6%** | **14,6%** | 84,6 | 1,9 |
| Wee-mag 7 | 63,9% | 94,8% | 61,6 | 30,6 | 23,1% | 64,3 | 0,0 | **15,3%** | 57,3 | 0,0 | **15,3%** | **15,3%** | 84,6 | 2,9 |
| Wee-mag 8 | 55,1% | 90,6% | 61,9 | 29,9 | 18,7% | 64,4 | 0,0 | **12,0%** | 56,1 | 0,0 | **12,0%** | **12,0%** | 85,4 | 2,6 |
| **Média** | 45,1% | 69,2% | 56,7 | 28,2 | 21,0% | 55,6 | 2,6 | **16,5%** | 52,4 | 0,5 | 16,6% | **16,6%** | 78,0 | 2,1 |
| **Média final** | 13,7% | 22,6% | | | 9,1% | | | **6,7%** | | | 6,8% | **6,8%** | | |

**Tabela 8.5:** Comparação dos algoritmos BTE-Rand, BTE-MaxPW$^{-}$, BTE-BH e BTE-BH-Rand (melhores desvios em negrito).

| **Grupo** | **BTE-Rand** | | | | **BTE-MaxPW$^{-}$** | | | **BTE-BH** | | | **BTE-BH-Rand** | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $D_b$ (%) | $D_{avg}$ (%) | t (s) | $t_b$ (s) | $D_b$ (%) | t (s) | $t_b$ (s) | $D_b$ (%) | t (s) | $t_b$ (s) | $D_b$ (%) | $D_{avg}$ (%) | t (s) | $t_b$ (s) |
| Roszieg 1 | **0,0%** | **0,0%** | 3,3 | 0,1 | **0,0%** | 2,2 | 0,0 | **0,0%** | 2,1 | 0,0 | **0,0%** | **0,0%** | 2,9 | 0,1 |
| Roszieg 2 | **0,0%** | **0,0%** | 3,8 | 0,3 | **0,0%** | 2,8 | 0,0 | **0,0%** | 2,5 | 0,0 | **0,0%** | **0,0%** | 3,5 | 0,1 |
| Roszieg 3 | **0,0%** | **0,0%** | 3,1 | 0,2 | **0,0%** | 2,2 | 0,0 | **0,0%** | 1,9 | 0,1 | **0,0%** | **0,0%** | 2,6 | 0,0 |
| Roszieg 4 | **0,0%** | **0,0%** | 3,0 | 0,1 | **0,0%** | 2,2 | 0,0 | **0,0%** | 1,9 | 0,0 | **0,0%** | **0,0%** | 2,4 | 0,0 |
| Roszieg 5 | **0,0%** | **0,0%** | 4,2 | 0,4 | **0,0%** | 2,8 | 0,1 | **0,0%** | 2,4 | 0,0 | **0,0%** | **0,0%** | 3,2 | 0,1 |
| Roszieg 6 | **0,0%** | **0,0%** | 4,6 | 0,7 | **0,0%** | 2,9 | 0,3 | **0,0%** | 2,6 | 0,1 | **0,0%** | 0,1% | 3,5 | 0,2 |
| Roszieg 7 | **0,0%** | **0,0%** | 4,5 | 0,5 | **0,0%** | 2,9 | 0,0 | **0,0%** | 2,5 | 0,0 | **0,0%** | **0,0%** | 3,5 | 0,1 |
| Roszieg 8 | **0,0%** | 0,2% | 4,5 | 0,5 | **0,0%** | 2,8 | 0,1 | **0,0%** | 2,6 | 0,0 | **0,0%** | **0,0%** | 3,6 | 0,1 |
| **Média** | **0,0%** | **0,0%** | 3,9 | 0,3 | **0,0%** | 2,6% | 0,1 | **0,0%** | 2,3 | 0,0 | **0,0%** | **0,0%** | 3,2 | 0,1 |
| Heskia 1 | **0,0%** | 0,3% | 6,7 | 2,0 | 1,0% | 3,5 | 0,2 | **0,0%** | 3,1 | 0,4 | **0,0%** | **0,0%** | 5,3 | 0,5 |
| Heskia 2 | **0,0%** | 0,1% | 7,4 | 2,3 | 0,1% | 3,5 | 0,1 | 0,1% | 3,0 | 0,0 | 0,1% | **0,1%** | 5,6 | 0,5 |
| Heskia 3 | **0,0%** | **0,0%** | 7,3 | 2,2 | **0,0%** | 3,0 | 0,2 | **0,0%** | 3,0 | 0,0 | **0,0%** | **0,0%** | 5,4 | 0,5 |
| Heskia 4 | **0,0%** | 0,1% | 7,6 | 2,7 | 0,1% | 3,1 | 0,3 | 0,1% | 3,0 | 0,1 | 0,1% | **0,1%** | 5,5 | 0,5 |
| Heskia 5 | **0,0%** | 5,1% | 10,4 | 4,8 | 4,0% | 4,3 | 0,4 | 1,2% | 4,1 | 0,3 | 0,6% | **1,2%** | 6,8 | 0,5 |
| Heskia 6 | 1,3% | 3,5% | 10,0 | 3,6 | 2,5% | 4,6 | 0,8 | 1,5% | 4,5 | 0,4 | **0,5%** | **1,7%** | 7,4 | 0,8 |
| Heskia 7 | **0,0%** | 2,3% | 11,2 | 5,0 | 2,2% | 4,7 | 0,1 | 0,6% | 4,4 | 0,1 | **0,0%** | **0,5%** | 7,1 | 0,6 |
| Heskia 8 | **0,0%** | 4,1% | 12,3 | 6,1 | 3,2% | 4,8 | 0,5 | 2,3% | 4,2 | 0,0 | 1,3% | **1,6%** | 7,7 | 1,0 |
| **Média** | **0,2%** | 1,9% | 9,1 | 3,6 | 1,6% | 3,9 | 0,3 | 0,7% | 3,7 | 0,2 | 0,3% | **0,7%** | 6,4 | 0,6 |
| Tonge 1 | **4,1%** | 12,4% | 154,0 | 106,6 | 11,8% | 50,3 | 18,9 | 8,8% | 32,2 | 4,8 | 8,2% | **9,5%** | 69,5 | 14,9 |
| Tonge 2 | **5,2%** | 14,1% | 257,6 | 202,7 | 15,1% | 50,5 | 14,8 | 11,7% | 33,5 | 3,5 | 8,1% | **12,3%** | 76,2 | 14,1 |
| Tonge 3 | **3,0%** | 10,3% | 264,1 | 216,2 | 11,4% | 50,7 | 8,0 | 9,5% | 34,5 | 6,7 | 4,9% | **8,1%** | 99,7 | 38,8 |
| Tonge 4 | **4,7%** | 14,5% | 267,1 | 221,0 | 12,3% | 49,7 | 26,5 | 10,5% | 32,3 | 7,0 | 6,5% | **9,8%** | 109,4 | 46,4 |
| Tonge 5 | 12,3% | 24,0% | 327,0 | 242,7 | 18,7% | 39,6 | 2,1 | **9,3%** | 41,1 | 0,0 | **9,3%** | **9,3%** | 98,7 | 18,9 |
| Tonge 6 | 9,2% | 20,7% | 409,1 | 327,6 | 12,1% | 39,6% | 7,8 | 7,5% | 40,8 | 0,0 | **6,4%** | **7,5%** | 98,6 | 21,5 |
| Tonge 7 | **6,2%** | 14,1% | 571,9 | 487,8 | 11,1% | 41,2 | 7,3 | 11,0% | 40,5 | 0,0 | 8,0% | **9,8%** | 108,9 | 27,5 |
| Tonge 8 | **6,8%** | 16,4% | 667,9 | 589,2 | 18,6% | 42,4 | 16,3 | 9,0% | 39,6 | 7,1 | 7,0% | **8,3%** | 150,2 | 58,9 |
| **Média** | **6,4%** | 15,8% | 364,8 | 299,2 | 13,9% | 12,7 | 21,7 | 9,7% | 36,8 | 3,6 | 7,3% | **9,3%** | 101,4 | 30,1 |
| Wee-mag 1 | 19,3% | 32,1% | 124,0 | 63,7 | 19,7% | 46,1 | 6,7 | **15,1%** | 46,7 | 0,0 | **15,1%** | **15,1%** | 76,9 | 1,3 |
| Wee-mag 2 | 20,6% | 30,7% | 144,7 | 83,4 | 21,4% | 47,0 | 1,5 | **18,4%** | 47,3 | 0,0 | **18,4%** | **18,4%** | 78,2 | 2,7 |
| Wee-mag 3 | **19,0%** | 28,4% | 189,8 | 131,7 | 22,7% | 46,7 | 10,0 | 20,4% | 48,1 | 3,9 | 19,2% | **20,4%** | 85,3 | 6,4 |
| Wee-mag 4 | **16,8%** | 26,6% | 190,0 | 127,5 | 20,3% | 46,8 | 2,4 | 20,3% | 49,5 | 0,0 | 17,4% | **19,5%** | 88,1 | 8,3 |
| Wee-mag 5 | 41,5% | 69,0% | 176,4 | 63,2 | 21,1% | 64,1 | 0,0 | **16,3%** | 56,4 | 0,0 | **16,3%** | **16,3%** | 115,9 | 1,8 |
| Wee-mag 6 | 38,0% | 61,2% | 174,4 | 65,5 | 20,5% | 65,3 | 0,0 | **14,6%** | 57,6 | 0,0 | **14,6%** | **14,6%** | 109,4 | 1,9 |
| Wee-mag 7 | 31,0% | 53,9% | 251,6 | 130,8 | 23,1% | 64,3 | 0,0 | **15,3%** | 57,3 | 0,0 | **15,3%** | **15,3%** | 116,9 | 2,8 |
| Wee-mag 8 | 30,1% | 51,0% | 250,2 | 123,6 | 18,7% | 64,4 | 0,0 | **12,0%** | 56,1 | 0,0 | **12,0%** | **12,0%** | 116,3 | 2,6 |
| **Média** | 27,0% | 44,1% | 187,6 | 98,7 | 21,0% | 55,6 | 2,6 | 16,5% | 52,4 | 0,5 | **16,0%** | **16,4%** | **98,4** | **3,5** |
| **Média final** | 8,4% | 15,5% | | | 9,1% | | | 6,7% | | | **5,9%** | **6,6%** | | |

**Tabela 8.6:** Comparação dos algoritmos CS-Rand, CS-MaxPW$^{-}$ e CS-BH (melhores desvios em negrito).

| **Grupo** | **CS-Rand** | | | | **CS-MaxPW$^{-}$** | | | | **CS-BH** | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $D_b$ (%) | $D_{avg}$ (%) | t (s) | $t_b$ (s) | $D_b$ (%) | $D_{avg}$ (%) | t (s) | $t_b$ (s) | $D_b$ (%) | $D_{avg}$ (%) | t (s) | $t_b$ (s) |
| Roszieg 1 | **0,0%** | 0,6% | 1,8 | 0,5 | **0,0%** | 0,4% | 1,5 | 0,2 | **0,0%** | **0,3%** | 1,8 | 0,2 |
| Roszieg 2 | **0,0%** | 2,5% | 1,7 | 0,4 | 0,8% | 2,5% | 1,5 | 0,2 | **0,0%** | **1,3%** | 1,8 | 0,1 |
| Roszieg 3 | **0,0%** | 1,0% | 1,8 | 0,5 | **0,0%** | 0,2% | 1,5 | 0,3 | **0,0%** | **0,1%** | 1,8 | 0,1 |
| Roszieg 4 | **0,0%** | 0,3% | 1,8 | 0,3 | **0,0%** | 0,3% | 1,5 | 0,2 | **0,0%** | **0,0%** | 1,8 | 0,0 |
| Roszieg 5 | 4,3% | 11,4% | 2,4 | 0,9 | **0,0%** | 5,8% | 2,0 | 0,2 | 2,2% | **2,2%** | 2,4 | 0,0 |
| Roszieg 6 | 4,9% | 13,3% | 2,4 | 1,1 | 3,9% | 11,4% | 2,0 | 0,3 | **0,0%** | **5,9%** | 2,4 | 0,3 |
| Roszieg 7 | 3,2% | 7,5% | 2,4 | 1,0 | **0,0%** | 1,7% | 2,0 | 0,1 | **0,0%** | **0,5%** | 2,4 | 0,1 |
| Roszieg 8 | **0,0%** | 5,7% | 2,4 | 1,1 | 1,4% | 3,1% | 2,0 | 0,1 | 0,6% | **1,9%** | 2,4 | 0,1 |
| **Média** | 1,6% | 5,3% | 2,1 | 0,7 | 0,8% | 3,2% | 1,8 | 0,2 | **0,4%** | **1,5%** | 2,1 | 0,1 |
| Heskia 1 | **0,0%** | 0,4% | 2,3 | 0,8 | **0,0%** | 0,5% | 1,9 | 0,6 | **0,0%** | **0,2%** | 2,3 | 0,4 |
| Heskia 2 | **0,0%** | 0,7% | 2,3 | 0,8 | **0,0%** | 0,3% | 1,9 | 0,4 | **0,0%** | **0,2%** | 2,3 | 0,2 |
| Heskia 3 | **0,0%** | 1,3% | 2,3 | 0,8 | **0,0%** | 0,4% | 1,9 | 0,5 | **0,0%** | **0,0%** | 2,3 | 0,2 |
| Heskia 4 | **0,0%** | 0,3% | 2,3 | 0,8 | **0,0%** | 0,3% | 1,9 | 0,5 | **0,0%** | **0,1%** | 2,3 | 0,5 |
| Heskia 5 | 0,9% | 9,1% | 3,3 | 1,6 | **0,0%** | 5,3% | 2,8 | 0,7 | 0,3% | **1,2%** | 3,3 | 0,3 |
| Heskia 6 | 1,4% | 5,0% | 3,3 | 1,5 | 1,7% | 4,2% | 2,8 | 0,8 | **0,5%** | **2,4%** | 3,3 | 0,5 |
| Heskia 7 | 0,6% | 3,8% | 3,3 | 1,2 | **0,0%** | 1,3% | 2,7 | 0,5 | **0,0%** | **1,1%** | 3,3 | 0,1 |
| Heskia 8 | 0,8% | 6,9% | 3,3 | 1,6 | 0,3% | 4,6% | 2,8 | 0,8 | **0,2%** | **1,4%** | 3,3 | 0,4 |
| **Média** | 0,5% | 3,4% | 2,8 | 1,1 | 0,3% | 2,1% | 2,3 | 0,6 | **0,1%** | **0,8%** | 2,8 | 0,3 |
| Tonge 1 | 14,5% | 39,3% | 45,8 | 35,5 | 9,0% | 18,7% | 38,2 | 6,7 | **8,1%** | **10,5%** | 45,9 | 2,5 |
| Tonge 2 | 17,9% | 50,1% | 46,2 | 34,6 | 14,7% | 23,3% | 38,6 | 5,3 | **9,9%** | **13,8%** | 46,5 | 2,9 |
| Tonge 3 | 16,1% | 33,8% | 45,9 | 36,1 | 10,5% | 18,9% | 38,5 | 11,6 | **7,6%** | **12,2%** | 46,3 | 5,6 |
| Tonge 4 | 17,1% | 37,9% | 46,0 | 35,2 | 12,1% | 20,7% | 38,5 | 11,3 | **7,9%** | **14,3%** | 46,3 | 4,2 |
| Tonge 5 | 41,4% | 70,5% | 58,2 | 41,4 | 19,9% | 20,4% | 48,4 | 1,3 | **9,6%** | **9,6%** | 48,4 | 1,2 |
| Tonge 6 | 41,0% | 73,5% | 59,2 | 44,4 | 18,4% | 18,4% | 51,0 | 1,3 | **9,0%** | **9,0%** | 48,8 | 1,2 |
| Tonge 7 | 28,3% | 55,1% | 58,9 | 42,7 | 16,1% | 17,1% | 49,1 | 1,9 | **10,5%** | **11,0%** | 49,0 | 1,3 |
| Tonge 8 | 30,9% | 59,6% | 58,9 | 42,7 | 19,9% | 26,2% | 49,0 | 3,8 | **11,2%** | **11,6%** | 49,0 | 1,3 |
| **Média** | 25,9% | 52,5% | 52,4 | 39,1 | 15,1% | 20,5% | 43,9 | 5,4 | **9,2%** | **11,5%** | 47,5 | 2,5 |
| Wee-mag 1 | 45,6% | 68,8% | 53,0 | 36,6 | **6,0%** | 15,5% | 45,0 | 12,9 | 7,8% | **13,9%** | 44,9 | 6,3 |
| Wee-mag 2 | 43,1% | 69,4% | 53,6 | 35,8 | **8,8%** | 16,1% | 44,7 | 13,6 | 9,3% | **15,2%** | 44,8 | 9,9 |
| Wee-mag 3 | 42,1% | 65,2% | 53,8 | 36,3 | 8,4% | 17,0% | 44,4 | 16,9 | **6,9%** | **15,6%** | 44,5 | 17,4 |
| Wee-mag 4 | 44,2% | 69,1% | 53,7 | 38,0 | **6,3%** | 15,8% | 44,3 | 14,9 | 6,5% | **15,5%** | 44,6 | 16,3 |
| Wee-mag 5 | 112,9% | 165,0% | 67,5 | 39,3 | 18,3% | 21,0% | 60,1 | 2,0 | **16,3%** | **16,3%** | 60,0 | 1,2 |
| Wee-mag 6 | 113,8% | 164,0% | 68,0 | 43,9 | 15,6% | 19,8% | 60,0 | 5,1 | **12,9%** | **14,5%** | 60,4 | 1,6 |
| Wee-mag 7 | 100,0% | 150,9% | 68,3 | 43,7 | 12,5% | 20,6% | 59,9 | 11,8 | **8,5%** | **14,4%** | 59,2 | 5,4 |
| Wee-mag 8 | 98,5% | 147,8% | 68,4 | 42,1 | 8,9% | 17,3% | 61,9 | 29,9 | **7,3%** | **11,5%** | 59,2 | 3,9 |
| **Média** | 75,0% | 112,5% | 60,8 | 39,5 | 10,6% | 17,9% | 52,2 | 10,9 | **9,4%** | **14,6%** | 52,2 | 7,7 |
| **Média final** | 25,7% | 43,4% | | | 6,7% | 10,9% | | | **4,8%** | **7,1%** | | |

desvio médio final resultante da BTM-MaxPW$^-$, BTM-BH e BTM-BH-Rand foram de 9,1%, 6,7% e 6,8%, respectivamente. Note que a BTM-BH-Rand foi muito robusta nas 20 repetições das instâncias em cada grupo. Isso pode explicado devido ao fato de que as iterações após o procedimento de reinício do algoritmo (através de uma solução aleatória) não conseguiram gerar uma solução melhor que a encontrada até a primeira metade do método.

Os algoritmos provenientes da BTE também se comportaram muito bem com as soluções iniciais vindas das heurísticas construtivas, como pode ser observado na Tabela 8.5. A BTE-MaxPW$^-$ obteve seu desvio médio final (9,1%) melhor qoe o obtido pela BTE-Rand (8,4%). Já a BTE-BH foi superior às duas anteriores, com valor do $D_b$ igual a 6,7%. Finalmente, a abordagem BTE-BH-Rand foi a melhor entre as três anteriores, com os resultados de $D_b$ e $D_{avg}$ iguais a 5,9% e 6,6%. Observe que neste caso, o procedimento de reinício surtiu efeito no método, em especial nos grupos Heskia 5-8 e Tonge 1-8.

As versões CS-MaxPW$^-$ e CS-BH, baseadas no CS proposto por Chaves (2009), produziram soluções de qualidade superior à versão original, denominada CS-Rand. De fato, os desvios médios $D_b$ do CS-MaxPW$^-$ e CS-BH foram de 6,7% e 4,8%, respectivamente, contra 25,7% do CS-Rand.

Em todos os métodos que utilizam as heurísticas construtivas como ponto inicial, pode-se dizer que os tempos computacionais de execução e de se encontrar a melhor solução foram reduzidos, fornecendo mais uma justificativa para o uso das heurísticas construtivas propostas.

## 8.5 Outros algoritmos propostos na literatura

A *busca por agrupamentos* (CS, do inglês *clustering search*) tem origem na *busca evolutiva por agrupamentos* (ECS, do inlgês *evolutionary clustering search*) proposta por Oliveira (2004). Segundo Chaves (2009), o CS procura dividir o espaço de busca e localizar regiões promissoras por meio do enquadramento dessas em *clusters*.

O CS funciona da seguinte forma: partindo de uma solução inicial $x$ obtida por uma metaheurística, tenta-se agrupar $x$ no *cluster* $C_j$ mais similar a essa solução. Caso o volume (número de soluções) $v_j$ de $C_j$ atinja um nível mínimo pré-determinado, o cluster pode estar em uma região considerada promissora. Caso contrário, através de um índice máximo de eficácia estabelecido, um procedimento de busca local pode ser aplicado em $C_j$ ou perturbações nesse *cluster* são feitas. Caso o critério de parada não seja alcançado, o algoritmo recomeça gerando outra solução a partir da metaheurística, repetindo os passos anteriormente citados.

Além do CS, descreveremos brevemente o algoritmo HGA proposto por Moreira et al. (2011) para o ALWABP, uma adaptação da metaheurística BRKGA (*biased random-key genetic algorithm*), que difere do RKGA na escolha dos pais para fase de recombinação.

Em linhas gerais, os componentes básicos de um RKGA são indivíduos representados vetores em $\mathbb{R}^n$ (utilizaremos um abuso de notação e neste momento, consideraremos $n$ como um parâ-

metro que representa a dimensão do problema) e um decodificador que mapeia essa estrutura em uma solução factível. Operações como seleção, recombinação e mutação (às vezes substitutída pela inserção de soluções aleatórias na próxima geração) são análogas às observadas nos algoritmos genéticos usuais.

No HGA estudado, em cada geração, um conjunto de soluções de elite é mantido na população, e uma parcela de soluções aleatórias é inserida a cada iteração, operação que faz vezes ao mecanismo de mutação. O restante da população é constituída pelos filhos gerados através do operador de cruzamento uniforme (*uniform crossover*).

Para a construção do HGA para o ALWABP, ao invés de definir as prioridades de execução das tarefas pelos trabalhadores por regras, cada indivíduo foi representado por uma matriz $M^{|W|\times|N|}$ a qual cada elemento $m_{wi} \in M^{|W|\times|N|} : m_{wi} \in [0, 1]$ corresponde à prioridade do trabalhador $w$ em executar a tarefa $i$. Como decodificador, Moreira et al. (2011) utilizaram o Algoritmo 3, apresentado na seção 4.1. Além disso, a cada novo filho gerado, o procedimento de busca local BLM (veja seção 7) é executado, a fim de melhorar o qualidade da solução. Os parâmetros utilizados na execução do algoritmo HGA foram: tamanho da população $P = 100$ e probabilidade de *crossover* $p_c = 0,5$. O algoritmo, a exemplo das versões randômicas da BT, foi executado 20 vezes para cada instância. Como critério de parada, adotou-se um número máximo de 200 iterações ou 100 iterações sem melhoramento da melhor solução atual. Os experimentos relativos aos HGA foram conduzidos em um computador Intel Core i7 930 com processador 2.8 GHz e 3.0 GB de RAM.

## 8.6 Algoritmos propostos *vs.* Abordagens da literatura

Nesta seção, comparamos os cinco algoritmos propostos para o ALWABP. Entre eles, consideramos melhor versão da BT (BTE-BH-Rand), a versão existente da metaheurística GRASP, além dos métodos CS-BH, HGA (Moreira et al., 2011) e IBS (Blum & Miralles, 2011).

O critério de parada do algoritmo GRASP, descrito no capítulo 6, foi o número pré-determinado de iterações igual a 1000 para cada regra de prioridade utilizada, perfazendo, portanto, um total de 3000 iterações (3 regras de prioridade $\times$ 1000 iterações). Além disso, testes computacionais preliminares mostraram que o tamanho limitado da lista de candidatos restrito $LCR$ igual a $n/5$ produziram os melhores resultados, sendo desta forma adotado nos testes finais. Devido ao fator estocástico do GRASP proposto, cada instância foi executada 3 vezes, número de execuções considerado suficiente para observar a robustez dos resultados (a partir de testes preliminares com mais execuções).

A Tabela 8.7 apresenta os resultados das comparações entre os cinco algoritmos. Para a família Roszieg, a BTE-BH-Rand e o IBS são os algoritmos que melhor se comportaram, obtendo

desvios médios de 0,0%, em tempos computacionais bem reduzidos. Considerando a família Heskia, o IBS manteve sua superioridade, com desvio médio de 0,0%. Apesar disso, vale ressaltar o bom desempenho do HGA, com valores de $D_b$ e $D_{avg}$ iguais a 0,1% e 0,3%, respectivamente. Dos outros três algoritmos, o CS-BH foi o melhor comparado ao BTE-BH-Rand e ao GRASP.

Relativo à família Tonge, o algoritmo HGA foi o que conseguiu melhores valores de desvios relativos às melhores soluções, com 0,4% em média. Além disso, para os grupos Tonge 1-3, o HGA obteve valores de desvios melhores que os conhecidos na literatura. Analisando as colunas $D_{avg}$, vemos que o IBS foi superior aos demais, com 2,4% de desvio médio. Note que entre os outros três algoritmos considerados para a análise, o GRASP obteve resultados melhores que o BTE-BH-Rand e que o CS-BH, particularmente nos grupos Tonge 5-8.

Finalmente, analisando a família Wee-mag, o IBS obteve o melhor desvio médio, com valores de $D_b$ e $D_{avg}$ iguais a 0,2% e 1,7%. Entretanto, para os grupos Wee-mag 1 e Wee-mag 4, o HGA conseguiu resultados superiores aos melhores conhecidos na literatura (veja a coluna $D_b$ para esse algoritmo). Entre os outros métodos, o CS-BH se sobressaiu em relação aos demais, a exceção do grupo Wee-mag 8, no qual o GRASP foi melhor que a BTE-BH-Rand e o CS-BH. No geral, apesar de algumas discrepâncias de desempenhos computacionais, o tempo de execução de todos os algortimos pode ser considerado aceitável em contextos práticos.

Vale a pena ressaltar o seguinte fato: apesar do GRASP proposto ter sido na maioria das situações pior que a BTE-BH-Rand, por outro lado, teve uma grande importância na geração de soluções para o *pool* utilizado pelo algortimo AHRT (veja seção 7.2), já que tinha como característica produzir soluções com alocações de tarefas e de trabalhadores bem variadas, fundamental para o problema de rotação de tarefas.

## 8.7 Rotação de Tarefas

O AHRT proposto foi comparado com o *método de decomposição por períodos* (MDP), proposto por Costa & Miralles (2009) que, ao nosso conhecimento, é o método que apresenta os melhores resultados para o problema de rotação de tarefas no contexto do ALWABP. Para cada instância, consideramos os tempos de ciclo aumentados de um fator $R$, em que $R = \{1{,}05; 1{,}10; 1{,}25; 1{,}50\}$. Isso indica que podemos permitir uma perda da produtividade em função de priorizar o maior número de execução das tarefas diferentes pelos trabalhadores, de tal forma que o tempo de ciclo desejado para a instância $k$ é dado por $C' = R \cdot C_k^{best}$. Além disso, levamos em conta três valores diferentes de $T$ para os períodos de rotação, em que $T = \{2; 4; 7\}$, resultando num total de 12 cenários. Como se trata de um problema de maximização, o valor de $D_{avg}$ é calculados da seguinte forma:

**Tabela 8.7:** Comparação de cinco metaheurísticas propostas para o ALWABP-2 (melhores desvios em negrito).

| Grupo | CS-BH | | | | BTE-BH-Rand | | | | GRASP | | | | HGA | | | | IBS | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $D_b$ (%) | $D_{avg}$ (%) | t (s) | $t_b$ (s) | $D_b$ (%) | $D_{avg}$ (%) | t (s) | $t_b$ (s) | $D_b$ (%) | $D_{avg}$ (%) | t (s) | $t_b$ (s) | $D_b$ (%) | $D_{avg}$ (%) | t (s) | $t_b$ (s) | $D_b$ (%) | $D_{avg}$ (%) | $t_b$ (s) |
| Roszieg 1 | **0,0%** | 0,3% | 1,8 | 0,2 | **0,0%** | **0,0%** | 2,9 | 0,1 | **0,0%** | **0,0%** | 5,9 | 0,1 | **0,0%** | **0,0%** | 1,5 | 0,0 | **0,0%** | **0,0%** | 0,0 |
| Roszieg 2 | **0,0%** | 1,3% | 1,8 | 0,1 | **0,0%** | **0,0%** | 3,5 | 0,1 | 6,0% | 6,0% | 12,3 | 0,0 | 5,6% | 5,7% | 2,4 | 0,0 | **0,0%** | **0,0%** | 0,1 |
| Roszieg 3 | **0,0%** | 0,1% | 1,8 | 0,1 | **0,0%** | **0,0%** | 2,6 | 0,0 | 1,6% | 1,6% | 7,7 | 0,0 | 0,9% | 0,9% | 2,0 | 0,0 | **0,0%** | **0,0%** | 0,1 |
| Roszieg 4 | **0,0%** | **0,0%** | 1,8 | 0,0 | **0,0%** | **0,0%** | 2,4 | 0,0 | **0,0%** | **0,0%** | 7,0 | 0,0 | **0,0%** | **0,0%** | 1,6 | 0,0 | **0,0%** | **0,0%** | 0,0 |
| Roszieg 5 | 2,2% | 2,2% | 2,4 | 0,0 | **0,0%** | **0,0%** | 3,2 | 0,1 | 1,1% | 1,5% | 7,4 | 0,0 | **0,0%** | **0,0%** | 1,9 | 0,0 | **0,0%** | **0,0%** | 0,0 |
| Roszieg 6 | **0,0%** | 5,9% | 2,4 | 0,3 | **0,0%** | 0,1% | 3,5 | 0,2 | 4,6% | 4,6% | 9,0 | 0,0 | 2,0% | 2,0% | 2,2 | 0,1 | **0,0%** | **0,0%** | 0,0 |
| Roszieg 7 | **0,0%** | 0,5% | 2,4 | 0,1 | **0,0%** | **0,0%** | 3,5 | 0,1 | **0,0%** | **0,0%** | 10,3 | 0,0 | **0,0%** | **0,0%** | 2,4 | 0,0 | **0,0%** | **0,0%** | 0,0 |
| Roszieg 8 | 0,6% | 1,9% | 2,4 | 0,1 | **0,0%** | **0,0%** | 3,6 | 0,1 | **0,0%** | **0,0%** | 10,4 | 0,0 | **0,0%** | **0,0%** | 2,5 | 0,0 | **0,0%** | **0,0%** | 0,0 |
| **Média** | 0,4% | 1,5% | 2,1 | 0,1 | **0,0%** | **0,0%** | 3,2 | 0,1 | 1,7% | 1,7% | 8,8 | 0,0 | 1,1% | 1,1% | 2,1 | 0,0 | **0,0%** | **0,0%** | 0,0 |
| Heskia 1 | **0,0%** | 0,2% | 2,3 | 0,4 | **0,0%** | **0,0%** | 5,3 | 0,5 | 0,2% | 0,6% | 22,5 | 0,8 | **0,0%** | **0,0%** | 4,1 | 0,1 | **0,0%** | **0,0%** | 8,2 |
| Heskia 2 | **0,0%** | 0,2% | 2,3 | 0,2 | 0,1% | 0,1% | 5,6 | 0,5 | 0,2% | 0,4% | 27,1 | 1,7 | 0,1% | 0,1% | 5,2 | 0,2 | **0,0%** | **0,0%** | 3,0 |
| Heskia 3 | **0,0%** | **0,0%** | 2,3 | 0,2 | **0,0%** | **0,0%** | 5,4 | 0,5 | **0,0%** | **0,0%** | 31,5 | 0,7 | **0,0%** | **0,0%** | 5,2 | 0,2 | **0,0%** | **0,0%** | 5,6 |
| Heskia 4 | **0,0%** | 0,1% | 2,3 | 0,5 | 0,1% | 0,1% | 5,5 | 0,5 | 0,4% | 0,6% | 33,0 | 3,4 | **0,0%** | 0,3% | 5,1 | 0,2 | **0,0%** | **0,0%** | 5,2 |
| Heskia 5 | 0,3% | 1,2% | 3,3 | 0,3 | 0,6% | 1,2% | 6,8 | 0,5 | 0,9% | 1,2% | 26,7 | 0,8 | **0,0%** | 0,6% | 5,2 | 0,2 | **0,0%** | **0,0%** | 1,1 |
| Heskia 6 | 0,5% | 2,4% | 3,3 | 0,5 | 0,5% | 1,7% | 7,4 | 0,8 | 0,8% | 1,0% | 24,7 | 1,2 | 0,5% | 0,6% | 5,3 | 0,2 | **0,0%** | **0,0%** | 2,5 |
| Heskia 7 | **0,0%** | 1,1% | 3,3 | 0,1 | **0,0%** | 0,5% | 7,1 | 0,6 | 0,5% | 1,0% | 25,4 | 0,6 | **0,0%** | 0,3% | 5,3 | 0,2 | **0,0%** | **0,0%** | 1,7 |
| Heskia 8 | 0,2% | 1,4% | 3,3 | 0,4 | 1,3% | 1,6% | 7,7 | 1,0 | 1,7% | 2,5% | 36,2 | 1,3 | **0,0%** | 0,7% | 6,7 | 0,9 | **0,0%** | **0,0%** | 2,5 |
| **Média** | 0,1% | 0,8% | 2,8 | 0,3 | 0,3% | 0,7% | 6,4 | 0,6 | 0,6% | 0,9% | 28,4 | 1,3 | 0,1% | 0,3% | 5,3 | 0,3 | **0,0%** | **0,0%** | 3,7 |
| Tonge 1 | 8,1% | 10,5% | 45,9 | 2,5 | 8,2% | 9,5% | 69,5 | 14,9 | 9,5% | 10,2% | 1015,4 | 129,6 | **-0,9%** | **2,0%** | 169,6 | 35,4 | 1,0% | 2,8% | 86,4 |
| Tonge 2 | 9,9% | 13,8% | 46,5 | 2,9 | 8,1% | 12,3% | 76,2 | 14,1 | 8,2% | 9,3% | 1299,7 | 384,0 | **-0,6%** | **1,2%** | 225,3 | 35,8 | 0,0% | 1,2% | 92,2 |
| Tonge 3 | 7,6% | 12,2% | 46,3 | 5,6 | 4,9% | 8,1% | 99,7 | 38,8 | 7,6% | 8,9% | 1921,5 | 386,1 | **-1,4%** | **1,5%** | 327,2 | 75,9 | 0,6% | 2,5% | 160,1 |
| Tonge 4 | 7,9% | 14,3% | 46,3 | 4,2 | 6,5% | 9,8% | 109,4 | 46,4 | 9,4% | 10,9% | 1915,5 | 437,8 | **0,0%** | **1,4%** | 312,6 | 46,3 | 0,6% | 1,5% | 171,4 |
| Tonge 5 | 9,6% | 9,6% | 48,4 | 1,2 | 9,3% | 9,3% | 98,7 | 18,9 | 4,3% | 5,9% | 1271,8 | 88,6 | 3,0% | 6,8% | 345,4 | 86,6 | **0,0%** | **3,3%** | 88,0 |
| Tonge 6 | 9,0% | 9,0% | 48,8 | 1,2 | 6,4% | 7,5% | 98,6 | 21,5 | 4,8% | 5,9% | 1350,6 | 105,9 | 0,3% | 4,5% | 342,6 | 82,3 | **0,0%** | **2,5%** | 70,5 |
| Tonge 7 | 10,5% | 11,0% | 49,0 | 1,3 | 15,3% | 15,3% | 116,9 | 2,8 | 5,0% | 7,0% | 1954,8 | 134,3 | 0,7% | 5,2% | 511,4 | 170,2 | **0,0%** | **2,3%** | 124,3 |
| Tonge 8 | 11,2% | 11,6% | 49,0 | 1,3 | 7,0% | 8,3% | 150,2 | 58,9 | 3,4% | 5,8% | 2368,3 | 548,7 | 2,0% | 4,4% | 571,8 | 125,8 | **0,0%** | **2,9%** | 156,4 |
| **Média** | 9,2% | 11,5% | 47,5 | 2,5 | 7,3% | 9,3% | 101,4 | 30,1 | 6,5% | 8,0% | 1637,2 | 276,9 | 0,4% | 3,4% | 350,7 | 82,3 | **0,3%** | **2,4%** | 118,7 |
| Wee-mag 1 | 7,8% | 13,9% | 44,9 | 6,3 | 15,1% | 15,1% | 76,9 | 1,3 | 33,8% | 36,9% | 547,3 | 79,5 | **-1,3%** | 5,3% | 127,6 | 40,0 | 0,6% | **4,3%** | 104,9 |
| Wee-mag 2 | 9,3% | 15,2% | 44,8 | 9,9 | 18,4% | 18,4% | 78,2 | 2,7 | 27,7% | 30,5% | 605,6 | 77,5 | 0,9% | 5,9% | 146,4 | 45,6 | **0,6%** | **4,5%** | 89,8 |
| Wee-mag 3 | 6,9% | 15,6% | 44,5 | 17,4 | 19,2% | 20,4% | 85,3 | 6,4 | 37,0% | 39,8% | 984,2 | 283,6 | **0,1%** | 8,4% | 217,3 | 70,7 | 1,8% | **4,8%** | 161,5 |
| Wee-mag 4 | 6,5% | 15,5% | 44,6 | 16,3 | 17,4% | 19,5% | 88,1 | 8,3 | 35,7% | 40,4% | 995,9 | 172,8 | **-1,8%** | 6,8% | 217,0 | 69,3 | 0,4% | **4,2%** | 135,5 |
| Wee-mag 5 | 16,3% | 16,3% | 60,0 | 1,2 | 16,3% | 16,3% | 115,9 | 1,8 | 15,7% | 18,1% | 690,4 | 38,4 | 5,0% | 13,0% | 276,0 | 71,0 | **0,0%** | **3,7%** | 54,1 |
| Wee-mag 6 | 12,9% | 14,5% | 60,4 | 1,6 | 14,6% | 14,6% | 109,4 | 1,9 | 12,8% | 15,0% | 743,7 | 69,1 | 6,9% | 11,9% | 275,1 | 84,6 | **0,0%** | **3,9%** | 59,3 |
| Wee-mag 7 | 8,5% | 14,4% | 59,2 | 5,4 | 15,3% | 15,3% | 116,9 | 2,8 | 14,0% | 15,4% | 1071,2 | 158,5 | 6,7% | 12,6% | 360,4 | 128,8 | **0,0%** | **4,3%** | 77,6 |
| Wee-mag 8 | 7,3% | 11,5% | 59,2 | 3,9 | 12,0% | 12,0% | 116,3 | 2,6 | 8,4% | 11,6% | 951,4 | 119,9 | 4,7% | 11,4% | 359,8 | 142,9 | **0,0%** | **4,2%** | 90,0 |
| **Média** | 9,4% | 14,6% | 52,2 | 7,7 | 16,0% | 16,4% | 98,4 | 3,5 | 23,1% | 26,0% | 823,7 | 124,9 | 2,7% | 9,3% | 247,5 | 81,6 | **0,4%** | **4,2%** | 96,6 |
| **Média** | 4,8% | 7,1% | | | 5,9% | 6,6% | | | 8,0% | 9,1% | | | 1,0% | 3,5% | | | **0,2%** | **1,7%** | |

$$\mathrm{D}_{avg}\,(\%) = \frac{\sum_{k=1}^{n_{inst}} \frac{z_k^{best} - z_k}{z_k}}{n_{inst}} \times 100\% \tag{8.3}$$

em que $z_k$ é o número máximo de tarefas diferentes executadas pelos trabalhadores obtido por algum método referente a instância $k$ e $z_k^{best}$ é o melhor valor dessa função objetivo conhecido para a instância $k$. Como informação adicional, apresenta-se também o desvio padrão ($\sigma$) dos valores obtidos por cada método.

Por questões relativas à tempo computacional, consideramos a BTE-BH-Rand com 10000 iterações, sendo que o procedimento de reinício com a solução aleatória ocorre quando esse algoritmo atinge 5000 iterações. Cada instância foi executada 5 vezes e um mesmo *pool* foi utilizado para a resolução dos 12 cenários em cada vez que o problema era resolvido. Para o critério de comparação, utilizou-se como conjunto de referência para o cálculo dos desvios as melhores soluções para cada instância, referentes tanto ao AHRT quanto do algoritmo de Costa & Miralles (2009).

A Tabela 8.8 nos mostra os resultados referentes à família de instâncias Roszieg. Nos 12 cenários analisados, o AHRT foi superior ao MDP, obtendo novos melhores valores para diversas instâncias. Os tempos computacionais foram bem satisfatórios e à exceção de dois cenários com 7 períodos, o AHRT foi mais eficiente que o MDP.

Na Tabela 8.9[1], apresentamos os resultados de todos os cenários para a família Heskia. Considerando $T$ = 2, pode-se observar que o AHRT obteve melhor desvio médio $D_{avg}$ com 5%, 10% e 25% de tolerância no tempo de ciclo médio. Com $R = 1,50$, o MDP foi ligeiramente superior ao AHRT, com diferença de 0,1% no desvio médio das soluções. Em todos esses cenários, os tempos computacionais do AHRT foram menores que os tempos do MDP.

Para $T$ = 4, novamente o AHRT foi superior nos três primeiros cenários de $R$, tanto na qualidade das soluções quanto no tempo computacional gasto. Para $R = 1,50$, o MDP obteve $D_{avg}$ igual a 1,0%, enquanto o AHRT também consegue 1,0%, só que com um tempo computacional bem menor. Considerando $T = 7$, o AHRT obteve bons resultados para $R$ = 1,05 e 1,10, perdendo nos dois últimos cenários de tolerância no tempo de ciclo médio. Note que para todas as situações, os valores dos desvios padrões $\sigma$ de ambos os métodos foram bem próximos de zero, mostrando a robustez das duas abordagens.

Com relação às BLRTs, seu efeito na qualidade das soluções foi muito significativo. Considerando a média de todos os cenários da família Roszieg, pode-se afirmar que com a aplicação das BLRTs, o valor de $\mathrm{D}_{avg}$ que era de 1,26% passou a ser 0,16%. Na família Heskia, os valores de $\mathrm{D}_{avg}$ considerando a média de todos os cenários, que eram de 4,49% foram para 0,95%.

A Tabela 8.10 nos mostra uma informação importante no que diz respeito à origem das soluções selecionadas pelo modelo (7.13)-(7.18) e utilizadas como base para a redução do espaço

[1]Para a instância 55 da família Heskia (T28-W7-L1-I20), consideramos o valor de $R$ = 1,06 ao invés de $R$ = 1,05, já que o AHRT não obteve um plano de rotação de tarefas para essa tolerância, devido a ausência de soluções de qualidade suficiente no *pool* de soluções.

de busca do Algoritmo 11 em todas as repetições das instâncias, para todos os cenários. Considerando a família Roszieg, podemos ver que a maior parte das soluções foi gerada através da BTE-BH-Rand, com 71,9%, seguido pelo GRASP, com 22% do total. Já para a família Heskia, o GRASP forneceu mais indivíduos como componentes dos subperíodos da rotação, com 61,9% do total, em comparação com os 28,9% de soluções geradas pela BTE-BH-Rand. Note que em ambas as famílias, mesmo em menor quantidade, a última geração do HGA e as heurísticas construtivas (coluna HC) forneceram soluções, mostrando de fato a contribuição de todos os métodos para a resolução deste problema.

**Tabela 8.8:** Resultados para a família Roszieg (melhores desvios em negrito).

| $T$ | $R$ | **AHRT** | | | | **MDP** | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | $D_{avg}$ (%) | $\sigma$ | t (s) | $t_{max}$ (s) | $D_{avg}$ (%) | $\sigma$ | t (s) | $t_{max}$ (s) |
| 2 | 1,05 | **0,1%** | **0,0** | 10,8 | 32,9 | 2,3% | 0,0 | 73,7 | 305,1 |
| | 1,10 | **0,1%** | **0,0** | 11,0 | 32,9 | 2,1% | 0,0 | 81,1 | 327,9 |
| | 1,25 | **0,1%** | **0,0** | 11,1 | 32,9 | 1,0% | 0,0 | 66,3 | 325,6 |
| | 1,50 | **0,0%** | **0,0** | 11,1 | 32,9 | 0,5% | 0,0 | 51,9 | 252,4 |
| 4 | 1,05 | **0,2%** | **0,1** | 12,1 | 32,9 | 6,5% | **0,1** | 108,5 | 488,7 |
| | 1,10 | **0,1%** | **0,0** | 12,0 | 33,1 | 7,1% | 0,1 | 125,4 | 508,6 |
| | 1,25 | **0,1%** | **0,0** | 13,3 | 42,5 | 4,9% | 0,0 | 134,6 | 594,1 |
| | 1,50 | **0,2%** | **0,0** | 16,1 | 100,4 | 4,0% | 0,0 | 84,8 | 295,0 |
| 7 | 1,05 | **0,3%** | **0,1** | 41,0 | 503,6 | 9,6% | **0,1** | 151,2 | 720,1 |
| | 1,10 | **0,1%** | **0,0** | 54,8 | 1045,8 | 9,3% | 0,1 | 183,2 | 796,9 |
| | 1,25 | **0,2%** | **0,0** | 41,3 | 396,5 | 4,0% | 0,1 | 198,6 | 745,4 |
| | 1,50 | **0,4%** | **0,0** | 54,8 | 777,8 | 1,5% | 0,1 | 134,5 | 528,7 |

**Tabela 8.9:** Resultados para a família Heskia (melhores resultados em negrito).

| $T$ | $R$ | **AHRT** | | | | **MDP** | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | $D_{avg}$ (%) | $\sigma$ | t (s) | $t_{max}$ (s) | $D_{avg}$ (%) | $\sigma$ | t (s) | $t_{max}$ (s) |
| 2 | 1,05 | **0,5%** | **0,0** | 33,5 | 75,4 | 1,5% | **0,0** | 87,1 | 893,3 |
| | 1,10 | **0,3%** | **0,0** | 34,4 | 76,8 | 1,4% | **0,0** | 102,7 | 795,9 |
| | 1,25 | **0,3%** | **0,0** | 35,1 | 81,0 | 0,6% | **0,0** | 106,4 | 1124,9 |
| | 1,50 | 0,1% | **0,0** | 35,7 | 83,0 | **0,0%** | **0,0** | 64,1 | 771,2 |
| 4 | 1,05 | **1,4%** | 0,1 | 36,4 | 136,4 | 4,1% | **0,0** | 130,4 | 960,7 |
| | 1,10 | **0,7%** | **0,0** | 37,7 | 104,4 | 2,5% | **0,0** | 187,3 | 1200,1 |
| | 1,25 | **1,0%** | **0,0** | 53,7 | 323,5 | 1,1% | **0,0** | 242,5 | 1506,3 |
| | 1,50 | 1,2% | **0,0** | 65,9 | 559,8 | **1,0%** | **0,0** | 267,0 | 1866,6 |
| 7 | 1,05 | **2,0%** | **0,1** | 46,5 | 174,1 | 4,4% | **0,1** | 188,0 | 1096,8 |
| | 1,10 | **1,1%** | **0,0** | 55,8 | 349,0 | 3,2% | **0,0** | 271,2 | 1698,1 |
| | 1,25 | 1,1% | **0,0** | 131,0 | 846,8 | **0,9%** | **0,0** | 396,1 | 2461,1 |
| | 1,50 | 1,8% | **0,0** | 243,3 | 2076,0 | **0,2%** | **0,0** | 475,3 | 4260,2 |

**Tabela 8.10:** Percentual de soluções provenientes dos métodos heurísticos escolhidas pelo modelo (7.13)-(7.18)

| **Família** | **Métodos Heurísticos** | | | |
|---|---|---|---|---|
| | **HGA** | **HC** | **BTE-BH-Rand** | **GRASP** |
| Roszieg | 5,6% | 0,5% | 71,9% | 22,0% |
| Heskia | 5,4% | 3,7% | 28,9% | 61,9% |

CAPÍTULO

9

# Conclusões e Trabalhos futuros

O *problema do balanceamento e designação de trabalhadores em linhas de produção*, conhecido na literatura como ALWABP, aparece de maneira prática quando tenta-se balancear linhas de produção com trabalhadores deficientes. Este problema apresenta uma grande dificuldade em sua resolução, pois além de possuir as restrições de precedência de tarefas, os trabalhadores executam as tarefas com tempos bem variados, sendo que para algumas tarefas, existem trabalhadores que nem sequer são capazes de realizá-las.

Neste trabalho, foram propostas heurísticas construtivas para o ALWABP basedas em regras de prioridades de tarefas. Por meio de uma série de testes computacionais, mostrou-se que essa abordagem é eficiente e eficaz tanto se considerarmos de forma isolada (somente a solução heurística) quanto como ponto de partida de metaheurísticas mais complexas (BTM, BTE e CS). Além disso, as heurísticas construtivas foram utilizadas como decodificador de um *hybrid genetic algorithm* (HGA) proposto por Moreira et al. (2011), que obteve resultados comparáveis aos melhores conhecidos até o momento na literatura.

Duas adaptações da metaheurística *busca tabu* (BT) para o ALWABP foram desenvolvidas neste trabalho. A primeira delas, denominada *busca tabu minimalista* (BTM), consiste em estabelecer uma relação entre os critérios de simplicidade e flexibilidade, que segundo Cordeau et al. (2002), são de suma importância uma vez que a maioria das implementações levadas a cabo em contextos práticos são de algoritmos com essas características. Além disso, propusemos outra abordagem de BT, denominada *busca tabu estendida* (BTE), que engloba três heurísticas de busca local e procedimentos de intensificação/diversificação. Tanto para a BTM quanto para a BTE, foram geradas variações que consideravam a solução inicial da seguinte forma: aleatória (Rand), o resultado da heurística com a regra de prioridade $\text{MaxPW}^{-}$, a melhor solução consi-

derando todas as heurísticas (BH) e finalmente, uma alternativa que após o início do tipo BH, reiniciava-se a busca na metade das iterações por meio de uma solução aleatória.

Os resultados obtidos pelas versões da BT foram competitivos comparados com os melhores métodos desenvolvidos na literatura. De fato, tem-se uma grande expectativa que uma investigação maior, em especial relacionada à BTE, possa levar a resultados mais precisos, sem a perda da eficiência computacional do algoritmo.

Os bons resultados das heurísticas sugeriram uma adaptação da metaheurística GRASP (*greedy randomized adaptative search procedure*), proposta inicialmente por Feo & Resende (1989). Apesar de não obter os melhores resultados para o ALWABP, o GRASP foi de fundamental importância para o problema de rotação de tarefas, uma vez que conseguiu encontrar soluções cujas alocações de tarefas e trabalhadores variavam consideravelmente.

A melhor BT proposta (BTE-BH-Rand), juntamente com o GRASP e mais três outros algoritmos (CS-BH, HGA e IBS Blum & Miralles (2011)) foram comparados entre si a fim de medir suas qualidade e eficiência. O IBS foi superior aos demais, pois conseguiu os melhores desvios médios em todas as quatro famílias de instâncias. Apesar disso, o HGA obteve soluções superiores às melhores conhecidas em 5 grupos de instâncias das famílias de maior porte. Entre os outros três algoritmos, concluiu-se que o CS-BH, o BTE-BH-Rand e GRASP tiveram os melhores desempenhos nessa ordem, respectivamente.

Finalmente, estudamos neste trabalho uma extensão do ALWABP, referente ao contexto de rotação de tarefas nas linhas de produção de um CTD. De fato, esse problema tem uma grande relevância, uma vez que contribui para que os trabalhadores tenham uma visão mais ampla do processo e sejam mais generalistas (Ouchi, 1981; Champion et al., 1994), além de ajudar a prevenir doenças comuns do trabalho, tais como estresse muscular e lesões por esforço repetitivo (Michalos et al., 2010). A grande dificuldade desta abordagem reside no fato de que o recurso de rotação de tarefas é um problema com dois objetivos conflitantes, isto é, se por um lado queremos que os trabalhadores executem o maior número de tarefas possível dentro de um período completo de rotação, por outro lado deseja-se manter níveis de produtividade na linha. Diante disso, adotou-se como função objetivo a maximização o número de tarefas diferentes executadas pelos trabalhadores, inserindo o objetivo de eficicência da linha como restrição do problema.

Para tanto, propusemos um *algoritmo híbrido para a rotação de tarefas* (AHRT), uma *matheuristic* que reúne o HGA e os outros métodos desenvolvidos neste trabalho (heurísticas construtivas, BTE-BH-Rand e GRASP). Além disso, utilizou-se a idéia de *pool* de soluções para armazenar as soluções factíveis obtidas durantes as buscas nas heurísticas e dois modelos lineares: o primeiro deles que escolhe as soluções do *pool* que maximizam a função objetivo e as utilizam para a geração de cortes na busca local baseada na formulação desse problema, aplicada imediatamente após esse procedimento.

Analisaram-se 12 cenários para esse problema, variando o número de períodos ($T = \{$ 2; 4; 7 $\}$) e a tolerância aceita na perda de eficiência da linha ($R = \{$ 1,05; 1,10; 1,25; 1,50 $\}$). Os resultados apresentados foram bem animadores. De fato, para a família Roszieg (com a maior densidade na rede de precedência de tarefas), um novo *benchmark* foi produzido em todos os cenários, com tempos computacionais mais eficientes na maioria dos casos. Na família Heskia, apesar do AHRT ser inferior 4 casos da tolerância ser de 25% ou 50%, outros bons resultados foram produzidos, em especial para cenários com $R$ = 1,05 e 1,10. Adicionalmente, podemos concluir que as BLRTs propostas foram muito satisfatórias, melhorando em todos os cenários a programação da rotação de tarefas inicialmente determinada pelo modelo linear auxiliar. Uma característica fundamental do método proposto é a sua escalabilidade, que permitiu que as maiores instâncias existentes (das famílias Tonge e Wee-mag) fossem pela primeira vez resolvidas ainda que aproximadamente.

Uma informação interessante a respeito da fase da construção do *pool* de soluções pelo AHRT diz respeito a origem das soluções utilizadas pelos procedimentos de programação matemática aplicados *a posteriori*. Por meio desses resultados, podemos ver que cada método heurístico utilizado teve sua contribuição ao final, sendo a BTE-BH-Rand e o GRASP os que mais geraram soluções para compor o período completo de rotação.

Continuações possíveis deste projeto são a investigação de novas heurísticas de busca local a serem aplicadas ao problema de rotação de tarefas no ambiente de um CTD, provavelmente baseadas em programação matemática. Além disso, a fim de mostrar uma maior flexibilidade do AHRT para instâncias de grande porte, serão testados os problemas pertencentes às famílias Tonge e Wee-mag. Relacionado ao ALWABP, uma paralelização do algoritmo GRASP se faz necessária, uma vez que as heurísticas podem ser executadas de forma independente, melhorando assim o desempenho dessa metaheurística de forma global.

De uma forma geral, os desenvolvimentos e testes realizados até o momento mostraram a dificuldade tanto do problema geral (ALWABP) quanto do problema de rotação de tarefas, mas também a possibilidade de tratá-lo de forma eficiente com o auxílio de estratégias clássicas como heurísticas construtivas e métodos consolidados como a BT e GRASP. Esses resultados são ainda mais encorajadores, na medida em que lembramos do caráter aplicado e social do ALWABP.

# Referências Bibliográficas


ALPAY, S. Grasp with path relinking for a multiple objective sequencing problem for a mixed-model assembly line. *International Journal of Production Research*, v. 47, n. 21, p. 6001–6017, 2009.

ARAÚJO, F. F. B.; COSTA, A. M.; MIRALLES, C. Balanceamento de linhas de produção com trabalhadores deficientes e máquinas paralelas. In: *Annals of the 5th Americas International Conference on Production Research*, 2010.

ARCHETTI, C.; SPERANZA, M.; SAVELSBERGH, M. An optimization-based heuristic for the split delivery vehicle routing problem. *Transportation Science*, v. 42, n. 1, p. 22–31, 2008.

ARMENTANO, V.; ARAÚJO, O. Grasp with memory-based mechanisms for minimizing total tardiness in single machine scheduling with setup times. *Journal of Heuristics*, v. 12, p. 427–446, 2006.

ARROYO, J. E. C. *Heurísticas e metaheurísticas para otimização combinatória multiobjetivo*. Tese de Doutorado, Universidade Estadual de Campinas, 2002.

BALAS, E.; PADBERG, E. W. *Set partitioning: a survey*. N. Christofides and A. Mingozzi and P. Toth and C. Sandi, 151–210 p., 1979.

BARTHOLDI, J.; EISENSTEEIN, D. A production line that balances itself. *Operations Research*, v. 44, p. 21–34, 1996.

BAYBARS, I. A survey of exact algorithms for the simple assembly line balancing problem. *Management Science*, v. 32, p. 909–932, 1986.

BECKER, C.; SCHOLL, A. A survey on problems and methods in generalized assembly line balancing. *European Journal of Operational Research*, v. 168, p. 694–715, 2006.

BLUM, C.; MIRALLES, C. On solving the assembly line worker assignment and balancing problem via beam search. *Computers and Operations Research*, v. 38, p. 328–339, 2011.

BOUTHILLIER, A. L.; CRAINIC, T. A cooperative parallel meta-heuristic for the vehicle routing problem with time windows. *Computers & Operations Research*, v. 32, n. 7, p. 1685–1708, 2005.

BOYSEN, N.; FLIEDNER, M.; SCHOLL, A. A classification of assembly line balancing problems. *European Journal of Operational Research*, v. 183, p. 674–693, 2007.

BOYSEN, N.; FLIEDNER, M.; SCHOLL, A. Assembly line balancing: Which model to use when? *International Journal of Production Economics*, v. 111, p. 509–528, 2008.

CARLIER, J. The one-machine sequencing problem. *European Journal of Operational research*, v. 11, p. 42–47, 1982.

CARNAHAN, B. J.; REDFERN, M. S.; NORMAN, B. Designing safe job rotation schedules using optimization and heuristic search. *Ergonomics*, v. 43, p. 543–560, 2000.

CHAMPION, M.; CHERASKIN, L.; STEVENS, M. Career-related antecedents and outcomes of job rotation. *Academy of Management Journal*, v. 37, n. 6, p. 1518–1542, 1994.

CHAVES, A. A. *Uma metaheurística híbrida de busca por agrupamentos aplicada a problemas de otimização combinatória*. Tese de Doutorado, Instituto Nacional de Pesquisas Espaciais, 2009.

CHAVES, A. A.; LORENA, L. A. N.; MIRALLES, C. Hybrid metaheuristic for the assembly line worker assignment and balancing problem. *Lecture Notes on Computer Science*, v. 5818, p. 1–14, 2009.

CHAVES, A. A.; MIRALLES, C.; LORENA, L. A. N. Clustering search approach for the assembly line worker assignment and balancing problem. In: *Proceedings of the 37th International Conference on Computers and Industrial Engineering, Alexandria, Egypt*, 2007, p. 1469–1478.

COFFMAN, E. G.; GAREY, M. R.; JOHNSON, D. S. *Approximation algorithms for bin-packing - un updated survey*. An algorithm design for a computer system design: Springer, Berlin, 49–106 p., 1984.

CORDEAU, J. F.; GENDERAU, M.; LAPORTE, G.; POTVIN, J. Y.; SEMET, F. A guide to vehicle routing heuristics. *Journal of the Operational Research Society*, v. 53, p. 512–522, 2002.

COROMINAS, A.; PASTOR, R.; PLANS, J. Línea de montaje con tiempos dependientes del tipo de operario. In: *Congreso Nacional de Estadística e Investigación Operativa, Universidad de Lleida*, 2003.

COSTA, A. M.; MIRALLES, C. Job rotation in assembly lines employing disabled workers. *International Journal of Production Economics*, v. 120, p. 625–632, 2009.

COSTA, A. M.; SANTOS, L. M.; MOREIRA, M. C. O.; MIRALLES, C. Análise da eficiência de linhas de produção com trabalhadores deficientes. In: *Anais do XLI Simpósio Brasileiro de Pesquisa Operacional, Porto Seguro, Brasil*, 2009.

COSTA, V. A. *Formação na perspectiva da teoria crítica da sociedade: As experiências dos trabalhadores deficientes visuais do serviço federal de processamento de dados*. Tese de Doutorado, Pontifícia Universidade Católica de São Paulo, 2001.

DELORME, X.; GANDIBLEUX, X.; RODRIGUEZ, J. Grasp for set packing problems. *European Journal of Operational Research*, v. 153, n. 3, p. 564–580, 2004.

DENG, Y.; BARD, J. A reactive grasp with path relinking for capacitated clustering. *Journal of Heuristics*, online First, 10 March 2010, 2010.

DOERNER, K.; SCHMID, V. Survey: Matheuristics for rich vehicle routing problems. *Lecture Notes on Computer Science*, v. 6373, p. 206–221, 2010.

DOERR, K.; KLASTORIN, T. D.; MAGAZINE, M. J. Synchronous unpaced flow lines with worker differences and overtime cost. *Management Science*, v. 46, n. 3, p. 421–435, 2000.

DOMSCHKE, W.; SCHOLL, A.; VOSS, S. *Produktionsplanung-ablauforganisatorische aspekte*. 1993.

DONSCHKE, W. *Logistik: Rundreisen und touren*. 4th. 1997.

DUHAMEL, C.; LACOMME, P.; QUILLIOT, A.; TOUSSAINT, H. A multi-start evolutionary local search for the two-dimensional loading capacitated vehicle routing problem. *Computers and Operations Research*, v. 38, p. 617–640, 2011.

EL-ABD, M.; KAMEL, M. A taxonomy of cooperative search algorithms. *Lecture Notes on Computer Science*, v. 3636, p. 32–41, 2005.

FEO, T.; RESENDE, M. A probabilistics heuristic for a computationally difficult set covering problem. *Operations Research Letters*, v. 8, p. 67–71, 1989.

FEO, T.; RESENDE, M. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, v. 6, p. 109–133, 1995.

FESTA, P.; RIBEIRO, M. *Essays and surveys on metaheuristics*, cap. An annotated bibliography Kluwer Academic Publishers, p. 325–367, 2002.

FISCHETTI, M.; LODI, A. Local branching. *Mathematical Programming*, v. 98, n. 1–3, p. 23–47, 2003.

FISHER, M. L. The lagrangean relaxation method for solving integer programming problems. *Management Science*, v. 50, p. 1861–1871, 2004.

FORD, H. *My life and work - an autobiography of henry ford*. 78–87 p., 1922.

FROMM, E. *Conceito marxista de homem*. 1962.

GEL, E. S.; HOPP, W. J.; OYEN, M. P. V. Factors affecting opportunity of worksharing as a dynamic line balancing mechanism. *IEEE Transactions*, v. 10, p. 847–863, 2002.

GENDREAU, M.; HERTZ, A.; LAPORTE, G. A tabu search heuristic for the vehicle routing problem. *Management Science*, v. 40, p. 1276–1290, 1994.

GIORDANO, B. W. *(d)eficiência e trabalho: analisando suas representações*. Annablume: FAPESP, São Paulo, 2000.

GLOVER, F. Tabu search, part i. *ORSA Journal on Computing*, v. 1, p. 190–206, 1989.

GLOVER, F. W. Tabu search, part ii. *ORSA Journal on Computing*, v. 2, p. 4–32, 1990.

GLOVER, F. W.; KOCHENBERGER, G. A. *Handbook of metaheuristics*. F.S. Hillier, 2003.

GLOVER, F. W.; LAGUNA, M. *Tabu search*. 1997.

GOODMAN, M.; DOWSLAND, K.; THOMPSON, J. A grasp-knapsack hybrid for a nurse-scheduling problem. *Journal of Heuristics*, v. 15, n. 4, p. 351–379, 2009.

GRAVES, S. C.; LAMAR, B. W. In integer programming procedure for assembly system design problems. *Operations Research*, v. 31, p. 522–545, 1983.

GÜNTHER, R.; PUCHINGER, J.; BLUM, C. *Handbook of metaheuristics*, cap. Metaheuristic Hybrids Springer, p. 469–496, 2010.

GUPTA, S.; SMITH, J. Algorithms for single machine total tardiness scheduling with sequence dependent setups. *European Journal of Operational Research*, v. 175, n. 2, p. 722–739, 2006.

HACKMAN, S.; MAGAZINE, M.; WEE, T. Fast, effective algorithms for simple assembly line balancing problems. *Operations Research*, v. 37, p. 916–924, 1989.

HANSEN, P.; MLADENOVIĆ, N.; UROŠEVIĆ, D. Variable neighborhood search and local branching. *Computers and Operations Research*, v. 33, n. 10, p. 3034–3045, 2006.

HASHIMOTO, H.; YAGIURA, M.; IBARAKI, T. An iterated local search algorithm for the time-dependent vehicle routing problem with time windows. *Discrete Optimization*, v. 5, n. 2, p. 434–456, 2008.

HOPP, W. J.; TEKIN, E.; OYEN, M. P. V. *Benefits of skill chaining in production lines with cross trained workers*. Relatorio Tecnico, WorkSmart Laboratory, 2001.

JAIME, L. R.; CARMO, J. C. *A inserção da pessoa com deficiência no mercado de trabalho*. 2005.

LEONE, R. D.; FESTA, P.; MARCHITTO, E. A bus driver scheduling problem: a new mathematical model and a grasp approximate solution. *Journal of Heuristics*, online First, 23 July 2010, 2010.

MANIEZZO, V.; STÜTZLE, T.; VOSS, S. Matheuristics, hybridizing metaheuristics and mathematical programming. *Annals of Information Systems*, v. 10, 2009.

MANSOOR, E. M. Assembly line balancing: a heuristic algorithm for variable operator performance levels. *Journal of Industrial Engineering*, v. 19, p. 618–628, 1968.

MARTELLO, S.; TOTH, P. *Knapsack problems: algorithms and computer implementations*. 1990.

MASTOR, A. A. An experimental investigation and comparative evaluation of production line balancing techiniques. *Management Science*, v. 16, p. 728–746, 1970.

MAVRIDOU, T.; PARDALOS, P.; PITSOULIS, L.; RESENDE, M. A grasp for the biquadratic assinment problem. *European Journal of Operational Research*, v. 105, n. 3, p. 613–621, 1998.

MICHALOS, G.; MAKRIS, S.; RENTZOS, L.; CHRYSSOLOURIS, G. Dinamic job rotation for workload balancing in human based assembly systems. *Journal of Manufacturing Science and Technology*, v. 2, p. 153–160, 2010.

MIRALLES, C.; GARCÍA-SABATER, J.; ANDRÉS, C.; CARDÓS, M. Application of u-lines principles to the assembly line worker assignment and balancing problem (ualwabp). a model and solving procedure. In: *Proceedings of the Operational Research Peripatetic Postgraduate Programme International Conference (ORP3), Valencia, Spain*, 2005.

MIRALLES, C.; GARCÍA-SABATER, J.; ANDRÉS, C.; CARDÓS, M. Advantages of assembly lines in sheltered work centers for disabled: a case study. *International Journal of Production Economics*, v. 110, n. 1–2, p. 187–197, 2007.

MIRALLES, C.; GARCÍA-SABATER, J.; ANDRÉS, C.; CARDÓS, M. Branch and bound procedures for solving the assembly line worker assignment and balancing problem: application to sheltered work centers for disabled. *Discrete Applied Mathematics*, v. 156, n. 3, p. 352–367, 2008.

MOREIRA, M. C. O.; COSTA, A. M. A minimalist yet efficient tabu search for the balancing assembly line with disabled workers. In: *Anais do XLI Simpósio Brasileiro de Pesquisa Operacional, Porto Seguro, Brasil*, 2009.

MOREIRA, M. C. O.; COSTA, A. M.; RITT, M.; CHAVES, A. A. Simple heuristics for the assembly line worker assignment and balancing problem, submitted to Journal of Heuristics - March, 2011.

MOREIRA, M. C. O.; SANTOS, L. M.; COSTA, A. M. Trabalhadores com deficiências em linhas de produção: modelos, resultados e discussões. In: *Anais da XIV Escuela Latinoamericana de Verano de Investigacíon de Operaciones, El Fuerte, México*, 2009.

NASCIMENTO, M.; CARVALHO, F.; CARVALHO, A. Investigation of a new grasp-based clustering algorithm applied to biological data. *Computers and Operations Research*, v. 37, p. 1381–1388, 2010a.

NASCIMENTO, M.; RESENDE, M.; TOLEDO, F. Grasp heuristic with path-relinking for the multi-plant capacitated lot sizing problem. *European Journal of Operational Research*, v. 200, n. 3, p. 747–754, 2010b.

OLIVEIRA, A. *Algoritmos evolutivos híbridos com detecção de regiões promissoras em espaços de busca contínuo e discreto*. Tese de Doutorado, Instituto Nacional de Pesquisas Espaciais (INPE), 2004.

ONU Standart on equalization of opportunities for persons with disabilities. In: *85th Plenary Meeting of the United Nations, New York*, 1993.

OUCHI, W. *How american business can meet the japanese challenge*. Addison-Wesley, 1981.

PARRAGH, S.; CORDEAU, J.; DOERNER, K.; HARTL, R. *Models and algorithms for the heterogeneous dial-a-ride problem with driver related constraints*. Relatorio Tecnico, University of Vienna, 2009.

PETTERSON, J. H.; ALBRACHT, J. J. Assembly-line balancing: Zero-one programming with fibonacci search. *Operations Research*, v. 23, p. 166–172, 1975.

PRINS, C.; PRODHON, C.; RUIZ, A.; SORIANO, P.; CALVO, R. Solving the capacitated location-routing problem by a cooperative lagrangean relaxation-granular tabu search heuristic. *Transportation Science*, v. 41, n. 4, p. 470–483, 2007.

PUCHINGER, J.; RAIDL, G. Combining metaheuristics and exact algorithms in combinatorial optimization: A survey and classification. *Lecture Notes on Computer Science*, v. 3562, p. 41–53, 2005.

REKIEK, B.; DOLGUI, A.; DELCHAMBRE, A.; BRATCU, A. State of art of optimization methods for assembly line design. *Annuals Reviews on Control*, v. 26, p. 163–174, 2002.

RESENDE, M. *Greedy randomized adaptative search procedures (grasp)*. Relatorio Tecnico, AT&T Labs Research, 1998.

RESENDE, M.; RIBEIRO, C. *Handbook of metaheuristics*, cap. Greedy randomized adaptative search procedures: Advances and applications Kluwer Academic Publishers, p. 219–249, 2003.

ROCHAT, Y.; TAILLARD, E. Probabilistic diversification and intensification in local search for vehicle routing. *Journal of Heuristics*, v. 1, p. 147–167, 1995.

SASSAKI, R. K. *Inclusão. construindo uma sociedade para todos*. 1997.

SCAPARRA, M.; CHURCH, R. A grasp and path relinking heuristic for rural road network development. *Journal of Heuristics*, v. 11, p. 89–108, 2005.

SCHMID, V. *Trucks in movement: Hybridization of exact approaches and variable neighborhood search for the delivery of ready-mixed concrete*. Tese de Doutorado, University of Vienna, 2007.

SCHMID, V.; DOERNER, K.; HARTL, R.; SAVELSBERGH, M.; STOECHER, W. A hybrid solution approach for ready-mixed concrete delivery. *Transportation Science*, v. 43, n. 1, p. 70–85, 2009.

SCHOLL, A. *Balancing and sequencing of assembly lines*. W. A. Müller and M. Bihn, 1999.

SCHOLL, A.; VOSS, S. Simple assembly line balancing - heuristic approaches. *Journal of Heuristics*, v. 2, p. 217–244, 1996.

SEÇKINER, S. U.; KURT, M. A simulated annealing approach to the solution of job rotation scheduling problems. *Applied Mathematics and Computation*, v. 188, p. 31–45, 2007.

SEÇKINER, S. U.; KURT, M. Ant colony optimization for the job rotation scheduling problem. *Applied Mathematics and Computation*, 2008.

SERPRO http://www.serpro.gov.br/noticias-antigas/noticias-2004/20041221, ultima visita em 7 de abril de 2009., 2004.

SIMONELLI, A. P. *Desenvolvimento e sistematização de modelo de análise de tarefas industriais para a inclusão de portadores de necessidades especiais no trabalho*. Dissertacao de Mestrado, Departamento de Engenharia de Produção - Universidade Federal de São Carlos, 2005.

SIMONELLI, A. P. *Contribuições da análise da atividade e do modelo social para a inclusão no trabalho de pessoas com deficiencia*. Tese de Doutorado, Departamento de Engenharia de Produção, Universidade Federal de São Carlos, 2009.

SIMONELLI, A. P.; RODRIGUES, D. S.; SOARES, L. Caracterização do perfil de trabalhadores afastados e de pessoas com deficiência no mercado de trabalho do município de são carlos. In: *Anais das XIV Jornadas de Jovens Pesquisadores da Associação de Universidades do Grupo Montevidéu*, 2006.

TALBOT, F. B.; PATTERSON, J. H.; GEHRLEIN, W. V. A comparative evaluation of heuristic line balancing techniques. *Management Science*, v. 32, p. 430–454, 1986.

THARMMAPHORNPHILAS, W.; NORMAN, B. A. A methodology to create robust job rotation schedules. *Annals of Operations Research*, v. 155, p. 339–360, 2007.

WEE, T. S.; MAGAZINE, M. J. Assembly line balancing as generalized bin packing. *Operations Research Letters*, v. 1/2, p. 56–58, 1982.