

Universidade Federal de Santa Catarina
Programa de Pós-Graduação em
Engenharia de Produção

Ronaldo Lima de Cristo

**Balanceamento de Linhas de Montagem com Uso de
Algoritmo Genético para o Caso de Linhas Simples e
Extensões**

Florianópolis
2010

Universidade Federal de Santa Catarina
Programa de Pós-Graduação em
Engenharia de Produção

Ronaldo Lima de Cristo

**Balancamento de Linhas de Montagem com Uso de
Algoritmo Genético para o Caso de Linhas Simples e
Extensões**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal de Santa Catarina como requisito parcial para obtenção do título de Mestre em Engenharia de Produção.

Orientador: Sérgio Fernando Mayerle, Dr.

Florianópolis
2010

Balanceamento de Linhas de Montagem com Uso de Algoritmo Genético para o Caso de Linhas Simples e Extensões

Ronaldo Lima de Cristo

Esta dissertação foi julgada e aprovada para a obtenção do Título de Mestre em Engenharia de Produção no Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal de Santa Catarina.

Florianópolis, 01 de março de 2010.

Prof. Antônio Cesar Bornia, Dr.
(Coordenador)

Prof. Sérgio Fernando Mayerle, Dr.
(Orientador)

BANCA EXAMINADORA

Prof. Antônio Sérgio Coelho, Dr.
(Presidente)

Prof. Fernando Antônio Forcellini, Dr.
(Membro)

Profa. Mônica Maria Mendes Luna, Dra.
(Membro)

Agradecimentos

A minha família, pelo apreço dado à educação.

À professora Esp. Gheysa Caroline Prado, pelo apoio incondicional.

Ao professor Dr. Sérgio Fernando Mayerle, pela excelente orientação e conhecimento transmitido.

A todos os professores e funcionários do PPGEP/UFSC.

Ao coordenador da Engenharia Logística da Volvo do Brasil, Gustavo Osório Nogueira, por acreditar neste projeto sendo flexível quanto a minha participação.

Resumo

CRISTO, Ronaldo. **Balaceamento de Linhas de Montagem com Uso de Algoritmo Genético para o Caso de Linhas Simples e Extensões.** 2010. 83 p. Dissertação (Mestrado em Engenharia de Produção), Departamento de Engenharia de Produção e Sistemas, Universidade Federal de Santa Catarina, Florianópolis, Brasil.

Neste trabalho foi realizada uma revisão sobre o problema de balanceamento de linha de montagem e suas extensões. Para este, são detalhados os modelos matemáticos, as heurísticas e as meta-heurísticas mais utilizados. Além disso, é apresentada uma taxonomia com as diversas particularidades referentes a este problema. O objetivo foi, através da aplicação de um Algoritmo Genético, trabalhar o balanceamento de linha considerando suas extensões quanto a: linhas de montagem com dois lados, layout em “U” e restrições de alocação de tarefas. Para validar os resultados obtidos, foram realizados testes com 269 (duzentos e sessenta e nove) problemas disponíveis na literatura. Em 90% dos casos estudados, os erros encontrados para as situações de balanceamento simples, de linhas com layout em “U” e de linhas com 2 lados não são superiores a 4, 7 e 8%, respectivamente. Ao comparar a qualidade da solução obtida com a encontrada por outros autores, aquela se mostra satisfatória.

Palavras - chave: balanceamento de linha, restrições de alocação, linhas de dois lados, algoritmo genético.

Abstract

CRISTO, Ronaldo. **Assembly Line Balancing using Genetic Algorithm to the Simple Lines Case and Extensions**. 2010. 83 p. Dissertation (Master in Industrial Engineering), Industrial Engineering and Systems Department, Federal University of Santa Catarina, Florianópolis, Brazil.

In this study it was performed a review on assembly line balancing problem and its extensions. For this one, it's detailed the mathematical models, the heuristics and meta-heuristics most used. Moreover, it's presented a taxonomy about this problem particularities. The objective was, through a Genetic Algorithm application, handle assembly line balancing considering its extensions like: two sided lines, "U" shape layout and assignment restrictions. To validate the outputs achieved, tests with 269 (two hundred and sixty nine) available problems were performed. In 90% of studied situations, the deviations found for simple balance, "U" shape layout and two sided lines were not bigger than 4, 7 and 8%, respectively. Comparing the solution quality with other authors found results, it becomes satisfactory.

Keywords: line balancing, assignment restrictions, two sided assembly lines, genetic algorithm.

Lista de Figuras

Figura 1 – Exemplo de linha em U.....	26
Figura 2 – Esquema de uma linha de montagem com linhas alimentadoras	29
Figura 3 – Tipos de representação do cromossomo	48
Figura 4 – Esquema da heurística utilizada para realização do <i>crossover</i>	53
Figura 7 – Interface do programa desenvolvido.....	62
Figura 8 – Esquema de montagem dos problemas de dois lados	64
Figura 9 – Resultados consolidados dos 269 problemas da literatura para o caso de SALBP-1.	64
Figura 10 – Resultados consolidados dos 269 problemas da literatura para o caso de layout em U.	65
Figura 11 – Resultados consolidados dos 269 problemas da literatura para o caso de linhas com 2 lados.	65

Lista de Quadros e Tabelas

Quadro 1 – Detalhamento do índice “ α ” na classificação dos ALB	19
Quadro 2 – Detalhamento do índice “ β ” na classificação dos ALB	25
Quadro 3 – Detalhamento do índice “ γ ” na classificação dos ALB.....	30
Tabela 1 – Problemas utilizados para o teste da probabilidade de mutação.....	56

Sumário

1	INTRODUÇÃO	10
1.1	Origem.....	10
1.2	Importância.....	12
1.3	Objetivos.....	13
1.4	Limitações	14
1.5	Estrutura do Trabalho	14
2	CARACTERIZAÇÃO DO PROBLEMA	16
2.1	Taxonomia dos problemas de ALB	17
2.1.1	Características do Grafo de Precedência (α)	18
2.1.2	Características da Linha e da Estação (β).....	24
2.1.3	Características quanto aos objetivos (γ)	29
2.2	Modelo Matemático.....	31
2.3	Técnicas de solução	40
2.3.2	Meta-Heurísticas para ALB	42
2.4	Pontos principais	44
3	MODELO E MÉTODO IMPLEMENTADOS	46
3.1	Caracterização do problema	46
3.2	Estrutura geral do método de busca.....	46
3.2.1	Representação do cromossomo	47
3.2.2	Construção da População Inicial	48
3.2.3	Avaliação do Fitness dos Indivíduos.....	50
3.2.4	Cruzamento ou <i>Crossover</i>	51
3.2.5	Mutação.....	54
3.3	Algoritmo de alocação de tarefas	58
3.4	Pontos Principais	60
4	TESTES NUMÉRICOS	61
4.1	Testes com problemas padrão da literatura	63
4.2	Análise dos resultados	66
4.3	Pontos Principais	68
5	CONCLUSÕES E RECOMENDAÇÕES.....	69
5.1	Conclusões.....	69
5.2	Recomendações	70

1 INTRODUÇÃO

1.1 Origem

De acordo com Amen (2001), a primeira linha de montagem foi introduzida por Eli Whitney. Após abandonar sua carreira de professor, Eli foi convidado para ajudar a administrar uma plantação de algodão. Ao acompanhar o trabalho manual necessário no processo de limpeza do produto, enxergou uma oportunidade para mecanizar tal atividade e expandir a produção. Patentada em 14 de março de 1794, a invenção teve rápida expansão pelos Estados Unidos, mas não gerou dividendos ao inventor que teve dificuldades em resgatar os *royalties* e custos de implantação. Já no início do século XIX, Whitney se reergueu idealizando não somente uma máquina, mas desenvolvendo um sistema de produção em que operários, mesmo sendo pouco qualificados, eram capazes de, através da padronização e divisão das tarefas, entregar resultados com qualidade. Este, mais tarde, veio a ser chamado de Sistema Americano de Produção que, por sua vez, configura a base para a produção em massa (ELI WHITNEY MUSEUM AND WORKSHOP, 2009).

Para Smith (1776), um pré-requisito para a produção em grandes volumes é a divisão das tarefas. O aumento na produção advém de três aspectos que são consequência desta divisão:

- a) Maior destreza dos operadores: quanto maior a agilidade em desempenhar certa atividade, melhores os resultados atingidos por um operador. Ao dividir as tarefas em funções elementares e fazer destas as únicas atividades desempenhadas durante toda a vida, tem-se um aumento de destreza.
- b) Redução no tempo que é comumente perdido ao passar de uma função para outra: ao entrar em contato com várias atividades, o funcionário tende a ter um desempenho inferior.
- c) Invenção de máquinas que possibilitem um trabalhador realizar as atividades de muitos: com a divisão do trabalho em atividades elementares, o desenvolvimento de maquinário para absorver tais operações fica facilitado.

Apesar de mais tarde, com a inclusão de conceitos de psicologia nos estudos sobre a relação do homem com o trabalho, mostrar-se que os benefícios na mecanização das funções do homem não são tão diretos

como apontado por Smith (1776), para o conhecimento disponível na época, esta foi uma revolução no modo de produção.

Um século depois de Eli Whitney, a aplicação dos conceitos de Adam Smith e Frederick Wilson Taylor (com a adoção de métodos cartesianos na mecanização do trabalho) culminaram, em 1 de abril de 1913, sob a gestão de Henry Ford, na implantação da primeira linha de montagem de produção em massa (AMEN, 2001). Através deste novo método de manufatura, baseado na padronização do produto e na divisão das atividades, foi possível a montagem do Modelo T a um preço muito competitivo. Isto veio de encontro às necessidades de uma população economicamente deficiente fazendo Ford, em 1926, lançar ao mercado 2 milhões de unidades deste veículo (Corrêa, 2003).

De maneira simplificada, linha de montagem é um fluxo organizado de atividades em que as unidades produtivas, ou seja, cada uma das estações de produção, são organizadas em série (BOYSEN et al., 2006).

Para Fernandes et al. (2008), o conceito de linha de montagem diz respeito a divisão das tarefas em um número finito de elementos. Estes são trabalhados de forma ordenada, atendendo às restrições de precedência. Estas restrições definem, para alguns pares de atividades, qual deve ser realizada *à priori*.

Uma explicação mais detalhada é compartilhada por Becker e Scholl (2003) e Grzechac (2008). Para eles, linhas de montagem consistem em um conjunto de estações distribuídas de forma seqüenciada e conectadas por um dispositivo de movimentação de materiais. Os elementos a serem processados são então alimentados na primeira estação. Eles passam para as estações seguintes em intervalos de tempo determinados, chamados de tempo de ciclo, até atingir o último posto de trabalho. As estações por sua vez reúnem certo número de atividades que são repetidas, dentro do tempo de ciclo, no processamento de cada novo elemento.

Apesar de ter sido idealizada para produção em massa, Boysen et al. (2006) afirma que desde o princípio deste modo de produção houve uma mudança expressiva para atender às demandas diversificadas do mercado. Por exemplo, segundo Meyr (2004), a BMW tem, teoricamente, 10^{32} modelos diferentes para oferecer aos seus clientes. Esta diversidade, entretanto, não impediu o uso de linhas de montagem, mas forçou uma adaptação de seu conceito para garantir esta flexibilidade de produção.

Becker e Scholl (2003) reforçam esta afirmação ao indicar que as linhas de produção têm abrangido também produtos customizados e de

baixo volume. Nestas condições, vê-se que o conceito vem se adaptando às necessidades e variações do mercado, continuando assim, com aplicabilidade garantida.

Ao distribuir as atividades ao longo da linha de produção, são inúmeras as seqüências que podem ser adotadas. Dependendo da ordem com que as tarefas são trabalhadas, obtêm-se resultados distintos na produtividade do sistema. Encontrar aquela seqüência que otimize uma determinada medida de desempenho, como por exemplo, minimizar o número de estações de trabalho ou o tempo de ciclo, ou então maximizar a eficiência da linha ou a sua rentabilidade, é, segundo Farnes e Pereira (2006), o Balanceamento de Linha de Montagem (ALB – sigla em inglês). Estes problemas, devido a sua forte carga combinatorial, são classificados como NP-Completo (KIM et al., 2000) e sua resolução tem sido motivo de estudo nas últimas décadas.

1.2 Importância

Devido ao alto custo de implementação de uma linha de montagem, os trabalhos no sentido de ordenar as atividades gerando o menor custo operacional são, segundo Boysen et al. (2006), de grande relevância. Neste sentido, desde o primeiro modelo matemático elaborado por Salveson, em 1955, para resolver o problema de ALB os trabalhos acadêmicos nesta área tomaram grandes proporções e várias técnicas foram desenvolvidas para atender esta necessidade.

Para Amem (2001), os esforços na resolução dos problemas de ALB sempre terão aplicação garantida. Isto porque, estão relacionados a um objetivo atemporal que é o aumento de produtividade.

Não somente custos diretos justificam os estudos sobre balanceamento. Carnahan et al. (2001) comenta que muitas disfunções musculares e ósseas são, ao menos em parte, resultado das demandas do trabalho. Um correto balanceamento pode evitar a sobrecarga sobre os operários que, ao dispor de plenas condições físicas e psicológicas, indiretamente contribuem positivamente para a produtividade do sistema.

Se por um lado são certos os benefícios que o correto balanceamento traz, por outro lado, Boysen e Flidner (2008) apontam a distância dos trabalhos acadêmicos em relação aos problemas reais. Este desalinhamento é motivado pelas inúmeras particularidades que são encontradas na indústria. Estas, nos estudos teóricos, geralmente são

estudadas isoladamente, não existindo modelos versáteis o suficiente para tratá-las em paralelo.

Esta situação é igualmente retratada por Tasan e Tunali (2008) que, após reunirem diversos trabalhos acadêmicos sobre o problema de ALB, notaram que muito foco é dado aos problemas triviais, sendo que as recentes tendências (por exemplo, diversos produtos compartilhando a mesma linha, layout em U, ambientes mais complexos com objetivos múltiplos) são, muitas vezes, ignoradas. Neste sentido, também no ambiente acadêmico, é evidente a relevância do tema.

Além dos pontos indicados, a própria complexidade dos problemas de ALB deve ser citada. Apesar de não haver formulação exata para conhecer o número total de seqüências possíveis na realização de um conjunto de atividades, uma aproximação é apresentada por Hoffmann (1959 *apud* BOYSEN E FLIEDNER, 2008) como sendo: $n!/2^{|E|}$. Onde n é o número total de atividades e $|E|$ é a quantidade total de restrição de precedência direta entre as tarefas. Como o número de combinações evolui com o fatorial da quantidade de atividades, a resolução exata deste tipo de problema através de modelos matemáticos fica limitada a problemas de pequeno porte. Assim, a busca por novas técnicas de solução tem seu espaço garantido.

1.3 Objetivos

O objetivo geral deste trabalho é desenvolver uma ferramenta de apoio para resolução de problemas de ALB, com uso de um Algoritmo Genético, considerando linhas de montagem com dois lados, layout em U, restrições de alocação de tarefas quanto a proximidade mínima e máxima entre os respectivos postos de trabalho, além de restrições de obrigatoriedade ou incompatibilidade de alocação de determinadas tarefas em determinados postos de trabalho.

Para alcançar este objetivo geral, são apontados os seguintes objetivos específicos:

- a) Classificar os problemas de ALB descritos na literatura;
- b) Revisar os modelos matemáticos empregados na resolução dos problemas de ALB;
- c) Pesquisar os métodos heurísticos e meta-heurísticos aplicados na resolução do ALB e suas variações;

- d) Implementar o Algoritmo Genético para resolução do problema especificado no objetivo geral deste trabalho;
- e) Realizar testes computacionais com problemas disponíveis na literatura.

1.4 Limitações

Este trabalho está limitado ao problema de ALB com objetivo de redução do número de postos de trabalho considerando linhas de montagem com dois lados, layout em U e restrições de alocação de tarefas. Todas as outras particularidades apresentadas por Boysen et al. (2006) referentes a grafos de precedência específicos para os produtos, estrutura do grafo de precedência, tempos de processamento, tempos de realização das tarefas dependente da seqüência, alternativas de processamento, movimento dos produtos, paralelismo, endereçamento de recursos, estação sujeita a incrementos no tempo de processamento e configurações adicionais não são contemplados neste trabalho.

Quanto a aplicação, são utilizados problemas padrão que foram reunidos no trabalho de Scholl (2007) com número total de atividades variando entre 7 e 297.

1.5 Estrutura do Trabalho

O trabalho está dividido em cinco seções, inclusive esta, na qual são apresentados um panorama sobre linhas de montagem e o respectivo problema de balanceamento, bem como a importância do estudo, os objetivos e as limitações do trabalho.

Na seção 2, o problema de balanceamento de linha é detalhado, sendo apresentadas as características dos problemas descritos na literatura segundo a taxonomia proposta por Boysen et al (2006). Esta revisão também contempla os trabalhos publicados que tratam da resolução de algumas das variantes do ALB. Também os modelos de programação linear e métodos de resolução heurísticos e meta-heurísticos fazem parte desta seção.

Na seção 3, são apresentados os fundamentos dos algoritmos genéticos, e como estes podem ser adaptados para resolução do problema de ALB proposto no objetivo geral deste trabalho. Inclui-se

nesta seção a definição da estrutura do cromossomo, os operadores utilizados para realização da seleção, do cruzamento e da mutação, além da função de *fitness* adotada para viabilizar a busca pela melhor solução que atenda as restrições impostas pelo analista.

Na seção 4 são apresentados os testes realizados com a ferramenta computacional, desenvolvida com base no Algoritmo Genético proposto na seção anterior, aplicados sobre 269 problemas básicos disponíveis na literatura. Também uma comparação com os resultados obtidos por outros autores é apresentada.

Finalmente na última seção são apresentadas as conclusões e as recomendações para trabalhos futuros.

2 CARACTERIZAÇÃO DO PROBLEMA

Quatro décadas se passaram desde a implantação da primeira linha de produção em massa até a descrição do problema de balanceamento de linha (ALB) feita por Helgesson em 1954 e formulado matematicamente por Salveson um ano mais tarde (Grzechca, 2008). Durante este período, as decisões sobre a organização do trabalho eram baseadas em processos de tentativa e erro.

Para Becker e Scholl (2003), Farnes e Pereira (2006) e Scholl et al. (2009), o ALB consiste em distribuir uma série de atividades em um conjunto de estações de trabalho de modo a otimizar uma medida de eficiência. O mesmo conceito é compartilhado por Amem (2001), Dimitriadis (2006) e Capacho et al. (2009). Entretanto, estes incluem de forma explícita que as restrições de precedência entre as atividades, ou seja, as relações tecnológicas que obrigam umas tarefas serem realizadas antes de outras, devem ser respeitadas.

O ALB, segundo Baker (1974 apud BAUTISTA e PEREIRA, 2007), faz parte da classe geral de problemas de seqüenciamento. Eles podem ser considerados como uma particularidade dos problemas de Bin Packing, entretanto, com algumas restrições adicionais, como é o exemplo do atendimento às relações de precedência. Dessa forma, apesar de alguns autores omitirem a referência direta a tais restrições, ela se faz necessária para correta caracterização do problema.

Sob estas condições, a definição mais bem elaborada para o ALB é a seguinte:

...ALB foca em agrupar as atividades em um grupo de estações de trabalho de forma eficiente e coerente. Basicamente, o objetivo é alcançar um agrupamento tal que minimize a ineficiência da linha e respeite todas as restrições impostas às atividades e às estações (BAUTISTA e PEREIRA, 2007, p. 2017, tradução nossa).

Assim, fica bem caracterizado que se trata de um problema combinatorial, que almeja encontrar a seqüência ótima para realização das tarefas, respeitando as mais diversas restrições (que serão mais bem detalhadas na próxima seção) inclusive as de precedência.

Esta busca pelo correto escalonamento das atividades, segundo Farnes e Pereira (2006), visa a ótima utilização dos recursos e faz-se necessária sempre que são realizadas modificações no processo de produção, como a inclusão ou exclusão de novas operações, alterações nos tempos de realização das tarefas, na taxa de produção desejada, entre outros.

Amem (2001), comenta que o ALB sempre será justificável, pois atingir níveis mais elevados de produtividade através da divisão do trabalho é um objetivo atemporal.

2.1 Taxonomia dos problemas de ALB

Uma classificação geral dos ALB proposta por Baybars (1986 apud BAUTISTA e PEREIRA, 2009), divide estes modelos em duas classes principais: SALBP (sigla em inglês para Problema Simplificado de Balanceamento de Linha de Montagem - *Simple Assembly Line Balance Problem*) e GALBP (sigla em inglês para Problema Geral de Balanceamento de Linha de Montagem - *General Assembly Line Balance Problem*).

O SALBP é a forma mais primitiva do ALB que, de acordo com Fernandes et al. (2008), leva em consideração as seguintes condições:

- a) todos os parâmetros de entrada são conhecidos com certeza;
- b) uma tarefa não pode ser dividida entre duas ou mais estações;
- c) a alocação de tarefas nas estações deve respeitar as restrições tecnológicas de precedência;
- d) todas as tarefas devem ser alocadas e, portanto, processadas;
- e) qualquer estação possui capacidade tecnológica para processar qualquer tarefa;
- f) o tempo de processamento de qualquer tarefa independe de qual estação a executará, bem como da seqüência de processamento da estação;
- g) a linha toda é considerada seriada e não existem estações em paralelo, sub-linhas alimentadoras ou estações de duplo lado;
- h) a linha só montará um único modelo de produto;
- i) Os principais critérios de otimização são: (i) minimizar o tempo ocioso total para um dado tempo de ciclo fixo (o que é equivalente a minimizar o número de estações de trabalho), também conhecido por SALBP-1; (ii) minimizar o tempo de ciclo para um dado número fixo de estações de trabalho (o que é equivalente a maximizar a taxa de produção), também conhecido por SALBP-2.

Na prática, situações que respeitam todas as condições acima descritas são praticamente inexistentes. Entretanto, neste meio século de evolução, os trabalhos acadêmicos para resolução do SALBP vêm se

multiplicando. Esta busca, porém, mostra-se necessária, pois todos os demais problemas são derivados deste modelo.

Com o objetivo de reunir todas as características que, quando incluídas no problema simplificado, criam os problemas gerais, Boysen et al (2006) faz uma classificação dos ALB baseada na notação $[\alpha, \beta, \gamma]$ que Graham et al. (1979) utilizou para problemas de organização do cronograma de operação de máquinas, onde:

- a) α : refere-se às características dos grafos de precedência. Em outras palavras, enumera as restrições referentes à alocação das atividades;
- b) β : refere-se às características da linha ou estação;
- c) γ : refere-se aos objetivos esperados com o modelo.

Para cada um destes índices, é descrita uma série de possibilidades de restrições que podem ser encontradas em problemas reais, e é feita a sugestão de alguns artigos de referência que tratam daquela particularidade.

2.1.1 Características do Grafo de Precedência (α)

Na notação utilizada, o índice “ α ”, que diz respeito às características do grafo de precedência, abrange seis dimensões para caracterizar o ALB. Cada uma delas indica uma situação especial:

- a) α_1 : Grafos de precedência específicos para os produtos;
- b) α_2 : Estrutura do grafo de precedência;
- c) α_3 : Tempos de processamento;
- d) α_4 : Tempos de realização das tarefas dependente da seqüência;
- e) α_5 : Restrições de alocação;
- f) α_6 : Alternativas de processamento;

Cada uma destas perturbações do SALBP ainda é dividida em pontos específicos conforme indicado no quadro 1.

Dentro da classe “ α_1 ”, acima enumerada, são incluídas as possibilidades de se realizar o balanceamento da linha para o caso da montagem de um único modelo de produto ($\alpha_1=0$) ou para o caso de uma mesma linha servir distintos modelos ($\alpha_1=\text{mix}$ ou $\alpha_1=\text{mult}$).

Quadro 1 – Detalhamento do índice “ α ” na classificação dos ALB

α_1	mix	Modelos mix de produtos
	mult	Modelos multi produtos
	o	Modelo único de produto
α_2	spec	Estruturas especiais do grafo de precedência
	o	Grafo de precedência pode assumir qualquer estrutura acíclica
α_3	t^{sto}	Tempos de processamento estocásticos
	t^{dy}	Tempos de processamento dinâmicos (efeitos de aprendizado)
	o	Tempos de processamento constantes e determinísticos
α_4	Δt^{dir}	Incrementos do tempo da tarefa devido a sucessão direta entre tarefas (troca de ferramentas)
	Δt^{indir}	Incrementos do tempo da tarefa devido a sucessão de tarefas (atividades atrapalham umas as outras)
	o	Incrementos do tempo da tarefa não considerados
α_5	link	Tarefas interligadas devem ser endereçadas para mesma estação
	inc	Tarefas incompatíveis não podem compartilhar a mesma estação
	cum	Restrições cumulativas de tarefas e estações
	fix	Tarefa só pode ser endereçada para uma estação específica
	excl	Tarefa não pode ser endereçada para uma estação específica
	type	Tarefa deve ser endereçada para determinado tipo de estação
	min	Mínima distância entre tarefas deve ser observada
	max	Máxima distância entre tarefas deve ser observada
	o	Restrições de endereçamento não são consideradas
α_6	pa^0	Tempos de processamento e custos são alterados
	pa^{predec}	Restrições de precedência são alteradas
	pa^{subgraph}	Subgrafos de precedência são adicionados
	o	Alternativas de processamento não são consideradas

Fonte: Boysen et. al. (2006).

Em sua concepção, como visto, as linhas de montagem eram indicadas para produção de grandes quantidades de um único produto.

Este apreço pela padronização foi enfatizado com a frase de Henry Ford: “Qualquer cor, desde que seja preto”. Entretanto, como comenta Corêa (2003), com o aumento do poder aquisitivo do consumidor e conseqüente inclusão de valores sociais às decisões de compra, esta massificação dos bens de consumo deu lugar à diversificação. Kara et al. (2007) complementa esta afirmação ao evidenciar o papel crucial que uma linha de múltiplos produtos exerce na resposta às necessidades do consumidor moderno. Segundo Meyr (2004), as configurações de opcionais dos veículos BMW podem gerar, teoricamente, 10^{32} combinações diferentes para oferecer aos seus clientes. Nestas condições, têm-se os casos classificados como “mix” e “mult”.

Para Fokkert e Kok (1997), em ambas as situações, cada modelo tem suas próprias restrições de precedência e o balanceamento não pode violar nenhuma delas. Além disso, as estações são flexíveis o suficiente para absorver as tarefas exigidas por cada uma das variantes em questão (CAO; MA, 2008). O objetivo, nestes casos, se estende também a encontrar a seqüência em que cada modelo deve ser processado para reduzir custos de produção. Fokkert e Kok (1997) diferenciam as classificações da seguinte forma: caso os lotes encontrados sejam unitários, ou seja, os tempos e custos de *setup* sejam desprezíveis, este é um caso “mix”, caso contrário, trata-se de um caso “mult”.

Geralmente, os vários grafos de precedência de cada um dos produtos de uma linha “mix” são combinados em um grafo equivalente para transformá-lo em um problema trivial de balanceamento (BOYSEN et al. 2006). Além desta solução, Fokkert e Kok (1997) apontam saídas através da transformação dos tempos das atividades em tempos relativos que são proporcionais à frequência de montagem de cada modelo.

Para a situação multi modelos, como os custos de *setup* entre um produto e outro são representativos, tem-se o trabalho adicional de determinar quais os lotes ideais para reduzir os custos de produção. Esta particularidade do ALB é pouco explorada na literatura. Wild (1972 apud FOKKERT e KOK, 1997) trabalha o tema sob distintas perspectivas. Caso os lotes de produção sejam pequenos e haja boa similaridade entre os modelos, o balanceamento pode ser orientado como sendo um “mix”. Já no caso de grandes lotes, ou seja, quando os custos de *setup* são bem representativos, cada modelo deveria ser balanceado como sendo um problema trivial, priorizando aquele de maior volume e adaptando as atividades dos demais a esta configuração básica. Desta forma, conseguem-se menores custos nas trocas de produtos.

Com a diversificação, necessária para atender às necessidades de cada cliente, até mesmo o conceito de produto simples passou por algumas adaptações, pois são poucos os exemplos que se encaixariam na descrição original. Dessa forma, caso mais de um produto compartilhe a mesma linha de produção, com tempos de setup desprezíveis e os processos produtivos muito similares, eles podem ser tratados como sendo de um problema de produto simples, como é o caso de CDs de dados (LEBEFROMM, 1999 apud BOYSEN et al., 2006).

Em certas situações, quando trabalhando com um mix de produtos, ao transformar os grafos de precedência em um único grafo equivalente, pode-se criar algumas estruturas cíclicas que não são previstas pelos métodos convencionais de balanceamento ($\alpha_2 = spec$). Ahmadi e Wurgaft (1994 apud FOKKERT e KOK, 1997) apontam duas soluções para este problema. A primeira consiste em dividir a gama de modelos em porções menores que possam ser combinadas sem criar ciclos nas relações de precedência, e a segunda sugere a repetição de algumas tarefas do grafo combinado para remover os ciclos. Com esta alteração, os problemas com esta característica são transformados em problemas simples de balanceamento.

Em relação aos tempos de processamento das atividades, somente em linhas automatizadas estes podem ser determinados e seguidos com exatidão. No caso de linhas dependentes da ação humana, a taxa de produção é afetada pela variabilidade natural da força de trabalho relacionada a aspectos como a habilidade, motivação, situações psicológicas diversas, entre outros. Neste sentido, faz-se necessária a consideração desta variação durante o processo de balanceamento (CHIANG; URBAN, 2006; GAMBERINI, 2006; BECKER; SCHOLL, 2003).

Esta instabilidade, inerente aos processos manuais, é prevista na classe “ α_3 ”, que considera a utilização de tempos estocásticos ($\alpha_3 = t^{sto}$) e efeitos de aprendizagem ($\alpha_3 = t^{dy}$) durante o processo de balanceamento. Para Lui et. al. (2005) esta variação nos tempos de processamento podem ser absorvidas de três formas distintas. A primeira é realizar as alocações considerando os tempos como determinísticos e, arbitrariamente, aplicar um fator de redução no tempo de ciclo. Esta folga, gerada no momento da alocação, absorveria as variações do processo, entretanto, a principal dificuldade está em determinar um fator de redução que evite excessos no tempo ocioso e, paralelamente, garanta que o tempo de ciclo seja respeitado. A segunda forma é realizar as alocações das tarefas nas estações até que a

probabilidade de extrapolar o tempo de ciclo não seja maior que um valor estipulado. A terceira forma leva em consideração os custos de produção em relação ao custo do não cumprimento do tempo de ciclo. Esta, apesar de ser mais interessante sob o aspecto de custo operacional, necessita de informações sobre custos que são dificilmente estimados com boa acuracidade.

Outro efeito no tempo de realização das tarefas está relacionado ao efeito de aprendizagem. Sob esta óptica, a duração das tarefas decresce ao longo do tempo a uma determinada taxa. Isto porque, com a repetição das operações, há uma especialização do operador que fica mais apto a realizar tal atividade. Para Boysen et al. (2006), em balanceamentos de produtos com longo ciclo de vida, este efeito pode ter grande impacto na eficiência do sistema de produção, pois, caso não seja considerado, a capacidade produtiva pode ser superestimada, gerando grande ociosidade ao longo do tempo. Toksari et al. (2008), nesta condição, orientam a realização do balanceamento com tempos já corrigidos pelo efeito do aprendizado.

Shafer et al. (2001) apresenta a situação de produtos com ciclos de vida muito curtos e com alta variação na demanda. Com isso, além dos benefícios de longo prazo do aprendizado não poderem ser plenamente aproveitados, ainda existe o efeito contrário, o do esquecimento. Nessa situação, Cohen et al. (2006) propõe um modelo para otimização dos lotes de produção (quando sujeitos a taxas homogêneas de aprendizado).

Outra classe a ser observada é a “ α_4 ”, que se refere aos incrementos no tempo das tarefas devido a ordem com que são realizadas. Esta pode ser uma relação direta ($\alpha_4 = \Delta t^{dir}$), no caso, por exemplo, de trocas de ferramentas entre atividades consecutivas, ou indiretas ($\alpha_4 = \Delta t^{indir}$) quando, por exemplo, ao colocar várias atividades que não contribuem para correta ergonomia do operador, este investe mais tempo nelas devido aos efeitos da fadiga.

Nas restrições englobadas pela classe “ α_5 ”, estão as limitações de associação entre as tarefas e as estações de trabalho. Em algumas situações específicas, algumas tarefas devem ser alocadas em uma mesma estação ($\alpha_5 = link$). Este é o caso, por exemplo, em que as tarefas compartilham um recurso de alto valor. Para não instalar o mesmo recurso em estações distintas, o que elevaria o custo de implantação da linha, esta sinergia é importante. Há também a situação contrária, ou seja, aquela em que atividades incompatíveis, como solda e abastecimento de combustível, por exemplo, não podem ser alocadas em

uma mesma estação ($\alpha_5 = inc$). Outra limitação possível no endereçamento das tarefas diz respeito aos recursos de uma estação, como por exemplo, o espaço disponível para peças, que pode ser limitado restringindo o número de atividades a serem alocadas para ela ($\alpha_5 = cum$). No caso de instalações antigas, em que, por exemplo, um recurso é difícil de ser realocado, as tarefas que necessitam daquele equipamento devem ser alocadas para aquela posição específica ($\alpha_5 = fix$). Pode ocorrer também, o caso em que uma tarefa não deve ser associada a uma determinada estação ($\alpha_5 = excl$). Este evento pode ocorrer, caso uma estação tenha alto índice de especialização e, para melhor utilizar tais recursos, atividades que não exigem tal tecnologia para sua realização são encaminhadas para outros postos de trabalho. Da mesma forma que a especialização do posto pode orientar a realização da atividade em outro ponto da linha, ela pode sugerir que certas atividades sejam realizadas nas posições em que se tem disponível os recursos necessários ($\alpha_5 = type$). E, por último, distâncias mínimas ($\alpha_5 = min$) ou máximas ($\alpha_5 = max$) entre as tarefas, podendo ser em tempo, número de estações, ou mesmo distância física, podem ser exigidas. Uma situação de distância máxima pode ser encontrada entre a aplicação de certa cola e o posicionamento do objeto a ser colado. Já uma necessidade de distância mínima é observada quando é necessário esperar a secagem desta cola antes de realizar outras operações que envolvam o objeto que foi colado.

A última classe a ser analisada na seção das restrições referentes aos grafos de precedência é a “ α_6 ”. Ela diz respeito às alternativas de processamento e pode ser dividida em três cenários.

No caso em que as diferentes alternativas influenciam somente o tempo e o custo das operações ($\alpha_6 = pa^0$), Boysen et al. (2006) indica a necessidade de uma relação de custo benefício para desenvolver o balanceamento. Isto porque, geralmente, com diferentes máquinas ou habilidades dos operadores, o desempenho e o custo variam. Sendo que, via de regra, maiores investimentos nos recursos representam menor tempo de processamento.

Além do tempo e custo, uma alternativa pode também influenciar localmente o grafo de precedência de outras atividades ($\alpha_6 = pa^{prec}$). Boysen et al. (2007) exemplifica esta ocorrência com a condição de, ao ser necessária a inclusão de um componente no produto, as relações de precedência de outro componente podem ser alteradas.

Em certas situações, cada variante de montagem exige uma ordem distinta na realização das tarefas, nestas condições, cada uma é

representada por um grafo de precedência ($\alpha_6 = pa^{subgraph}$). Para esta configuração, Capacho et al. (2009) trabalha como dois problemas a serem resolvidos simultaneamente: selecionar a alternativa de processamento e realizar o balanceamento. Na heurística proposta, de maneira iterativa, a busca da melhor solução para o problema é realizada através de vários balanceamentos gerados alternando os grafos de precedência e as metodologias de balanceamento.

2.1.2 Características da Linha e da Estação (β)

O índice “ β ”, utilizado na notação apresentada, contempla também seis dimensões. Estas caracterizam as particularidades referentes às linhas ou as estações sendo distribuídas da seguinte forma:

- a) β_1 : movimento dos produtos;
- b) β_2 : layout da linha;
- c) β_3 : paralelismo;
- d) β_4 : endereçamento de recursos;
- e) β_5 : estação sujeita a incrementos no tempo de processamento;
- f) β_6 : configurações adicionais.

Cada uma destas perturbações do SALBP ainda é dividida em pontos específicos conforme indicado no quadro 2.

Na dimensão “ β_1 ” são indicadas condições de contorno para o movimento dos produtos na linha de montagem. A primeira grande subdivisão deste item, apresentada por Boysen et al. (2006), classifica as linhas em compassadas e descompassadas. Aquelas ocorrem quando o material tem um avanço contínuo ou intermitente, desde que todos avancem para próxima estação ao mesmo tempo e que não haja distúrbios no tempo de ciclo. Já para estas, os produtos são passados para as próximas etapas assim que as operações necessárias em determinada estação foram terminadas.

Nas linhas compassadas, a situação elementar é que todas as estações tenham um volume tal de atividades que permita respeitar o tempo de ciclo em todas elas. Entretanto, no caso de linhas “mix”, por exemplo, pode-se considerar que todos os modelos devem respeitar o tempo de ciclo (each) ou que o mesmo deva ser respeitado com certa probabilidade (prob). Outra situação comentada é o caso de estações de teste, por exemplo, que tem tempos de ciclo distintos representando a classificação “div” que trata de tempos de ciclo locais.

Quadro 2 – Detalhamento do índice “ β ” na classificação dos ALB

β_1	$\text{o}\lambda\upsilon$	$\lambda=\text{o}$	Linha compassada e atividades restritas pelo tempo de ciclo (na média)
		$\lambda=\text{each}$	Linha compassada e todos os modelos devem respeitar o tempo de ciclo
		$\lambda=\text{prob}$	Linha compassada e o tempo de ciclo é respeitado com certa probabilidade
		$\upsilon=\text{o}$	Linha compassada e tempo de ciclo único para todas as estações
		$\upsilon=\text{div}$	Linha compassada e tempos de ciclo locais
	unpac^{o}	Linha descompassada e dessincronizada	
	$\text{unpac}^{\text{syn}}$	Linha descompassada e sincronizada	
β_2	o	Linha seriada	
	u^{o}	Linha em U	
	u^{n}	Linha em U formando “n” Us	
β_3	pline^{\wedge}	Linhas em paralelo	
	pstat^{\wedge}	Estações em paralelo	
	ptask^{\wedge}	Atividades em paralelo	
	pwork^{\wedge}	Trabalhos paralelos em uma mesma estação	
	o	Nenhum paralelismo é considerado	
β_4	equip	Problema de seleção de equipamentos	
	res^{o1}	Se duas atividades dividem um recurso, os investimentos são reduzidos na estação	
	res^{max}	A tarefa mais complexa define o nível de qualificação requerido para os recursos	
	res^{o}	Outros tipos de sinergia ou dependência	
β_5	$\Delta\text{t}^{\text{unp}}$	Atividades improdutivas na estação são consideradas	
	o	Não considerados incrementos de tempo na estação	
β_6	buffer	Buffers devem ser alocados e dimensionados	
	feeder	Linhas secundárias devem ser balanceadas simultaneamente	
	mat	Embalagens dos materiais devem ser dimensionadas e posicionadas	
	change	Necessárias máquinas para trocar os produtos de posição	
	o	Aspectos adicionais não são considerados	

Fonte: Boysen et. al. (2006).

Nas linhas descompassadas, a principal diferença está na decisão sobre o momento de transferência do material. Este pode acontecer de forma sincronizada ($unpac^{syn}$), ou seja, assim que todas as estações terminaram suas atividades, ou de forma dessincronizada ($unpac^o$), em outras palavras, quando uma estação termina suas tarefas o produto já é transferido para a próxima etapa.

A dimensão “ β_2 ” trata do layout da linha de montagem propriamente dito que pode ser em linha ou em formato de “U”, conforme figura 1.

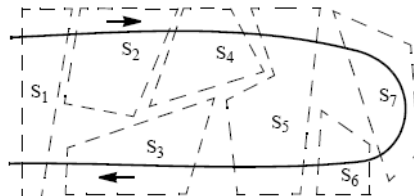


Figura 1 – Exemplo de linha em U

Fonte: Becker e Scholl (2003)

As áreas indicadas por “ S_i ” representam a i -ésima estação de trabalho e as setas caracterizam o movimento dos produtos ao longo da linha de montagem. Para Becker e Scholl (2003), a diferença encontrada no balanceamento de linhas em “U” é que uma estação pode receber não somente aquelas tarefas que tiveram suas predecessoras já alocadas a uma estação anterior, mas também aquelas que tiveram suas sucessoras já endereçadas.

Os autores ainda afirmam que o balanceamento de uma linha com esta configuração pode resultar em soluções melhores se comparadas ao caso das linhas convencionais. Isto porque, as possibilidades de combinações entre tarefas em uma mesma estação são muito maiores. Este fato complementa o comentário feito por Miltenburg (1998) em que linhas com esta característica tendem a necessitar de menor número de operadores.

Chiang e Urban (2006) e Kriengkorakot e Pianthong (2007) adicionam outras qualidades como redução nos desperdícios de movimentação melhorando o fluxo de materiais, redução no inventário em processo por serem soluções de fluxo contínuo, maior flexibilidade da mão-de-obra para tratar variações na demanda e melhor visualização e comunicação entre os operadores.

Outra condição encontrada nos problemas de balanceamento de linha diz respeito ao paralelismo entre atividades e estações. Estas

situações são orientadas pela dimensão “ β_3 ”. Ela é subdividida em quatro categorias.

A primeira apresenta o paralelismo entre linhas ($\beta_3 = pline$). Nesta condição, o objetivo é balancear mais de uma linha ao mesmo tempo (BENZER et al., 2007), sendo que os produtos de cada uma das linhas são diferentes. Scholl e Boysen (2009) consideram esta particularidade como resultado da união de dois problemas: Multi-produtos devem ser alocados para linhas em paralelo e as linhas devem ser balanceadas. Desta forma é lançada a possibilidade de combinar estações adjacentes das linhas vizinhas. Ao contar com estas combinações adicionais, a distribuição das atividades pode resultar em redução no número de operadores.

A situação de estações em paralelo ($\beta_3 = pstat$) é trabalhada pela segunda categoria. Trata-se de duplicar certas estações de trabalho para que estas realizem as operações dos próximos produtos de maneira alternada (BOYSEN et al., 2007). Esta situação é comum quando se tem uma atividade com um tempo de execução superior ao tempo de ciclo esperado para linha de produção. O *download* do *software* de controle das funções do painel dos caminhões de uma empresa montadora de veículos pesados da cidade de Curitiba, por exemplo, toma um tempo três vezes maior que o tempo de ciclo da linha. Neste caso, a estação que trata desta operação foi triplicada para atender às necessidades da linha.

A terceira categoria, que orienta a realização de tarefas em paralelo ($\beta_3 = ptask$), é demonstrada pelos mesmos autores. Neste contexto, é possível a realização de tarefas similares ou idênticas em estações distintas. Como visto anteriormente, esta situação é muitas vezes evitada, pois atividades similares geralmente utilizam os mesmos recursos, sendo sensato que compartilhem a mesma estação e, conseqüentemente, os mesmos recursos. Porém, esta condição não é necessariamente a mais eficiente visto que o endereçamento obrigatório de atividades similares para uma mesma estação pode resultar em perdas de eficiência geral do sistema. Nestas condições Bukchin e Rabinowitch (2006) propõem uma avaliação de custo-benefício em ambos os cenários.

A quarta categoria apresenta o paralelismo entre operadores ($\beta_3 = pwork$). Boysen et.al. (2006) exemplifica esta situação com o caso geralmente encontrado no setor automobilístico em que, devido às dimensões de algumas peças, os trabalhos associados a elas requerem mais de um trabalhador simultaneamente.

A dimensão “ β_4 ”, que apresenta a necessidade de alocação de recursos para as estações, é tratada por Boysen et al. (2007) como duas

situações principais. Na primeira ($\beta_4 = equip$), cada estação deve ter um único recurso associado que é selecionado dentre uma lista de recursos disponíveis. Na segunda, os recursos vão sendo selecionados à medida que as tarefas são endereçadas à estação. Este processo pode se dar de três formas. A primeira é proceder a instalação do recurso se ao menos uma atividade alocada para estação precisar do mesmo ($\beta_4 = res^{01}$). Na segunda, a tarefa com maior exigência define os recursos a serem utilizados ($\beta_4 = res^{max}$). Esta é importante no caso de tratar de quesitos como qualidade, velocidade ou qualificação. Por exemplo, a tarefa que exige maior precisão em uma usinagem vai definir a precisão do equipamento a ser utilizado nesta estação. E por último, os recursos são alocados conforme outras situações de sinergia e dependência que não foram descritas ($\beta_4 = res^0$).

Outra situação presente nas linhas de montagem é a influência no tempo de realização das atividades por condições que não agregam valor. Esta situação é trabalhada por Boysen et al. (2007) com a dimensão “ β_5 ”. Ela pode assumir a notação “ $\beta_5 = \Delta t^{unp}$ ” no caso de considerar tempos de transporte de peças, tempos de deslocamento do operador, tempos de calibragem do equipamento, etc.

Por último, a dimensão “ β_6 ” trata de aspectos gerais que podem ser adicionados ao ALB e também é dividido em algumas subcategorias conforme a situação. No caso de trabalhar com linhas descompassadas ou haver tempos de preparação dos equipamentos, por exemplo, a diferença de tempo na execução de algumas tarefas em relação a outras resultará em acúmulos de materiais em certos pontos da linha de produção (ou *buffers*). Estes *buffers* devem ser dimensionados de modo a gerar o menor custo operacional ($\beta_6 = buffer$). Outra subdivisão utilizada é a $\beta_6 = feeder$ que é caracterizada quando a linha tem um *layout* em aspecto de espinha de peixe, ou seja, quando linhas secundárias se unem a uma principal, conforme esquema da figura 2. Nesta condição, as linhas secundárias devem ser balanceadas juntamente com a linha principal.

Como as montagens geralmente estão associadas à utilização de certos componentes que devem estar disponíveis no momento de execução da tarefa, sua localização na linha de produção pode influenciar os tempos de montagem. Klampfl et al (2006) apresenta o *software* “*e-workcell*”, desenvolvido pela Ford, que, em conjunto com o balanceamento, trabalha o *layout* da borda de linha de modo a minimizar os custos com deslocamento ($\beta_6 = mat$).

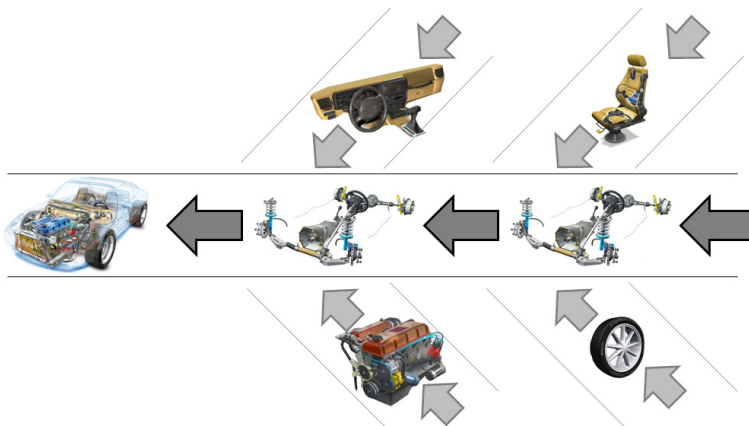


Figura 2 – Esquema de uma linha de montagem com linhas alimentadoras

Nota: Fonte das figuras usadas no esquema: Moving Graphics. Car Parts. 1999. Modelos 3D color. Disponível em: <http://www.movinggraphics.ca/Car_Illustrations.htm>. Acesso em: 16 de novembro de 2009

Fonte: Autor (2010)

E por último é apresentado o caso no qual tarefa exige alguma posição específica dos produtos ($\beta_6 = change$), sugerindo uma avaliação sobre trabalhar em tal posição em toda a estação ou instalar máquinas para movimentar o produto conforme necessidade (BOYSEN et al., 2007).

2.1.3 Características quanto aos objetivos (γ)

O último índice utilizado para caracterizações das diferentes condições apresentadas por um ALB diz respeito aos objetivos almejados (“ γ ”). Com apenas uma dimensão, este índice é subdividido em alguns pontos específicos conforme indicado pelo quadro 3.

Os objetivos básicos de um ALB, como visto anteriormente, são, a partir de um tempo de ciclo definido, encontrar um arranjo tal que represente o menor número de postos de trabalho possível ($\gamma=m$) ou, a partir de um determinado número de estações, minimizar o tempo de ciclo ($\gamma=c$).

Quadro 3 – Detalhamento do índice “ γ ” na classificação dos ALB

γ	m	Minimizar o número de estações
	c	Minimizar o tempo de ciclo
	E	Maximizar a eficiência da linha
	Co	Minimizar custo
	Pr	Maximizar rentabilidade
	SSL ^{stat}	Os tempos das estações devem ser equalizados em uma estação (balanceamento horizontal)
	SSL ^{line}	Os tempos das estações devem ser equalizados entre estações (balanceamento vertical)
	score	Minimizar ou maximizar algum indicador composto
	o	Somente soluções viáveis são testadas

Fonte: Boysen et. al. (2006).

O problema englobado pelo ponto “E” é, segundo Scholl e Becker (2006), o mais geral dos objetivos de um ALB. Nesta situação, deseja-se maximizar a eficiência trabalhando tanto com o número de estações quanto com o tempo de ciclo.

Como o balanceamento de uma linha é geralmente uma decisão de longo prazo, para Kimms (2000), além do investimento inicial, os custos de manutenção associados à solução escolhida são igualmente importantes e devem ser considerados. Assim, é importante considerar variáveis financeiras como a minimização dos custos operacionais ($\gamma=Co$) ou maximização da rentabilidade ($\gamma=Pr$).

Paralelamente, Boysen et al. (2007) apresenta um objetivo complementar que é o nivelamento da carga de trabalho entre as estações que pode ocorrer de duas formas. Na primeira ($\gamma=SSL^{stat}$), também conhecida por balanceamento horizontal, as situações de discrepância na carga de trabalho entre os produtos de uma linha “mix” ou “mult” devem ser amenizadas. Já na segunda ($\gamma=SSL^{line}$), também chamada de balanceamento vertical, todas as estações da linha devem ser similares no que se refere à taxa de ocupação.

Boysen et al. (2006), ao introduzir esta notação, ainda reserva um índice ($\gamma = score$) para o caso de outra função objetivo qualquer utilizada para atender alguma necessidade específica.

2.2 Modelo Matemático

Os problemas de ALB geralmente são formulados com um problema de programação binária em que um conjunto de atividades $V \subset \mathbb{N}^*$ deve ser alocado em um conjunto de estações $M \subset \mathbb{N}^*$. Estas associações são controladas pela variável:

$$x_{ik} = \begin{cases} 1, & \text{se atividade } i \in V \text{ foi endereçada para} \\ & \text{estação } k \in M; \text{ e} \\ 0, & \text{em caso contrário.} \end{cases} \quad (1)$$

Com esta notação, que será compartilhada por todas as equações utilizadas neste trabalho, Boysen e Fliedner (2008) apresentam o SALBP-1 da seguinte forma:

$$\text{Min} \quad N = \sum_{k \in M} x_{nk} \cdot k \quad (2)$$

$$\text{Sujeito a (s.a)} \quad \sum_{k \in M} x_{ik} = 1 \quad \forall i \in V \quad (3)$$

$$\sum_{k \in M} k \cdot x_{ik} \leq \sum_{k \in M} k \cdot x_{jk} \quad \forall (i, j) \in E \quad (4)$$

$$\sum_{i \in V} x_{ik} \cdot t_i \leq c, \quad \forall k \in M \quad (5)$$

$$x_{ik} \in \{0, 1\} \quad \forall i \in V; k \in M \quad (6)$$

A função objetivo representada pela equação 2 orienta a redução do número de postos utilizados. A tarefa de índice n , no caso, corresponde a uma tarefa fictícia, de duração nula, cuja relação de precedência inclui todas as demais tarefas da linha de montagem. Com isto, a função objetivo determina o valor de N , que corresponde ao índice da última estação. As restrições estabelecidas no problema forçam ainda que cada tarefa seja endereçada a uma única estação (equação 3), que as restrições de precedência (indicadas pelo conjunto E , que contém todas as combinações (i, j) no caso de a atividade i preceder a atividade j) sejam respeitadas (equação 4) e que a soma dos tempos t_i das atividades alocadas para cada estação não ultrapasse o tempo de ciclo c em nenhum posto de trabalho (equação 5).

Como dito anteriormente, a utilidade deste modelo fica comprometida, pois envolve tantas condições de contorno que, na forma original, são raramente encontrados em situações reais. Entretanto, ele ainda é muito explorado pois qualquer GALBP é originado a partir dele. Alguns autores, no intuito de flexibilizar a utilização deste modelo, propõem modificações no conjunto de equações originais para adaptar o problema a outras condições de balanceamento.

Para tempos estocásticos ($\alpha_3 = t^{sto}$), por exemplo, como visto na seção 2.2.1, Liu et al. (2005) apontam três maneiras distintas para considerá-los no momento do balanceamento: aplicar um redutor arbitrário ao tempo de ciclo, trabalhar com uma probabilidade do tempo de ciclo ser atendido ou fazer uma relação de custo benefício. As duas primeiras são descritas a seguir.

Na primeira forma, a simples adição de um fator redutor r ao tempo de ciclo expresso na equação 5 se encarrega da adaptação que fica como indicado pela equação 7.

$$\sum_{i \in V} x_{ik} \cdot t_i \leq c \cdot r \quad \forall k \in M \quad (7)$$

Para a segunda situação, ao assumir que a variação no tempo das tarefas segue uma distribuição normal, pode-se iniciar pela relação entre o tempo ocioso de cada estação com o desvio padrão total da mesma. O índice obtido, quando aplicado à função densidade de probabilidade acumulada, $\theta(z)$, retorna a probabilidade que a estação tem de cumprir o tempo de ciclo. O produtório do índice de todas as estações resulta na confiabilidade geral da linha de montagem que não deve ser inferior a um valor *Prob* estabelecido. Para aplicação deste método, a equação 5 deve ser substituída pela equação 8.

$$\prod_{k \in M} \left[\theta \left(\left(c - \sum_{i \in V} x_{ik} \cdot t_i \right) / \sqrt{\sum_{i \in V} x_{ik} \cdot \sigma_i^2} \right) \right] \leq Prob \quad (8)$$

Onde: σ_i = desvio padrão do tempo de execução da atividade i .

Outra particularidade também explorada por modelos matemáticos diz respeito ao incremento no tempo devido a seqüência com que as tarefas são realizadas (α_4). Esta condição é abordada por Scholl et al. (2006). Com adaptações no modelo proposto, esta situação é considerada com a substituição da restrição equação 5 pelas equações 9-15.

$$\sum_{i \in V} \left(x_{ik} \cdot \left(t_i + \sum_{j \in V} sd_{ij} \cdot \delta_{ij} \right) \right) \leq c, \quad \forall k \in M \quad (9)$$

$$y_i = \sum_{k \in M} k \cdot x_{ik} \quad \forall i \in V \quad (10)$$

$$y_i - y_j \leq Z(1 - \delta_{ij}) \quad \forall i, j \in V \quad (11)$$

$$y_j - y_i \leq Z \cdot \delta_{ij} \quad \forall i, j \in V \quad (12)$$

$$\delta_{ij} + \delta_{ji} = 1 \quad \forall i, j \in V \quad (13)$$

$$\delta_{ij} \in \{0; 1\} \quad \forall i, j \in V \quad (14)$$

$$y_i = \{1; 2; 3; \dots\} \quad (15)$$

Onde:

$$\delta_{ij} = \begin{cases} 1, & \text{se tarefa } i \text{ é realizada antes de } j; \text{ e} \\ 0, & \text{caso contrário.} \end{cases}$$

$$sd_{ij} = \text{Tempo adicional na tarefa } i \text{ ao ser realizada antes da tarefa } j.$$

$$y_i = \text{Índice da estação da tarefa } i.$$

Na equação 9 o tempo necessário para realização das tarefas é ajustado de acordo com a seqüência com que são realizadas. Para facilitar a descrição do modelo foi adicionada a variável y_i (equação 10) para coletar o índice da estação da tarefa i . Esta informação é utilizada para, através das equações 11, 12 e 13, armazenar na variável δ_{ij} a ordem em que as tarefas são realizadas. Para isso, utilizou-se um fator Z (que representa um valor muito grande) associado a esta variável. Assim, caso $\delta_{ij} = 1$, ou seja, caso a atividade i seja realizada antes de j , a equação 11 orienta a seqüência de realização das tarefas, caso contrário, a equação 12 assume esta função.

As restrições de alocação (α_5) também podem ser incluídas no sistema de equações apresentado. Para trabalhar as condições *fix*, *excl* e *type*, por exemplo, deve-se substituir a equação 3 pela equação 16:

$$\sum_{k \in M_i} \alpha_{ik} \cdot x_{ik} = 1 \quad \forall i \in V \quad (16)$$

Onde: M_i é o conjunto de estações nas quais a atividade i pode ser endereçada.

Ao informar as distâncias mínimas (d_{ij}^{min}) e máximas (d_{ij}^{max}) entre as atividades, contadas em número de estações, as condições *inc* e *min* são conseguidas com a inclusão da equação 17 e as restrições *link* e *max* com a adição da equação 18.

$$\left| \sum_{k \in M} x_{ik} \cdot k - \sum_{k \in M} x_{jk} \cdot k \right| \geq d_{ij}^{min} \quad \forall i, j \in V \quad (17)$$

$$\left| \sum_{k \in M} x_{ik} \cdot k - \sum_{k \in M} x_{jk} \cdot k \right| \leq d_{ij}^{max} \quad \forall i, j \in V \quad (18)$$

A condição de linhas em U ($\beta_2 = u^1$) com uso de modelos matemáticos é objeto do estudo de Kriengkorakot e Pianthong (2007). Como os próprios autores reforçam que a única diferença desta condição para o SALBP está nas restrições de precedência. Neste caso, para uma dada tarefa, todas as suas predecessoras ou todas as suas sucessoras devem ser distribuídas em estações de índice igual ou inferior ao da tarefa em questão. Nestas condições, com algumas adaptações nas equações propostas, a condição de balanceamento de linhas com layout em U é conseguida pela substituição da equação 4 pelas equações 19-23.

$$\sum_{k \in M} k \cdot x_{jk} \leq M \cdot e_i^a + \sum_{k \in M} k \cdot x_{ik} \quad \forall (j, i) \in E \quad (19)$$

$$\sum_{k \in M} k \cdot x_{jk} \leq M \cdot e_i^b + \sum_{k \in M} k \cdot x_{ik} \quad \forall (i, j) \in E \quad (20)$$

$$e_i^a + e_i^b = 1 \quad \forall i \in V \quad (21)$$

$$e_i^a \in \{0; 1\} \quad \forall i \in V \quad (22)$$

$$e_i^b \in \{0; 1\} \quad \forall i \in V \quad (23)$$

A equação 19 orienta que, para certa atividade, suas antecessoras não sejam alocadas em estações posteriores a sua. Já na equação 20, são as atividades sucessoras que não devem ser alocadas para tais estações. Como em cada atividade somente uma destas restrições deve ser ativada, utilizou-se um fator Z (que representa um valor muito grande) associado a duas variáveis binárias e_i^a e e_i^b (definidas na equação 21) para poder realizar esta seleção. Se para uma dada tarefa i , $e_i^a = 1$,

então $e_i^b = 0$, e vice versa. Em cada atividade, como somente uma destas variáveis binárias assumirá o valor zero, o fator Z associado a ela é desconsiderado em uma das equações. Entretanto, na outra, ele sugere que qualquer resultado é possível, ou seja, ele desativa a segunda restrição.

Passando para as particularidades de paralelismo das estações ($\beta_3 = pstat^\lambda$) e de paralelismo das tarefas ($\beta_3 = ptask^\lambda$), estas são tratadas por Boysen e Flidner (2008) como uma relação de custo benefício. Para entendimento destes modelos, eles apresentam primeiramente o caso que trata da maximização do lucro ($\gamma = Pr$) que fica da seguinte forma:

$$Max \quad N = db \cdot \frac{T}{c} - m \cdot K^s \quad (24)$$

s.a equações de 3 à 6

$$\sum_{k \in M} k \cdot x_{|V|k} \leq m \quad (25)$$

Onde: db = margem de contribuição por produto

T = período total de estudo

K^s = custo fixo por estação

Tem-se, na função objetivo (equação 24), a maximização dos lucros gerados pela linha. Já a equação 25 foi incluída para limitar o número de estações em um dado valor.

A partir deste sistema, ao incluir os custos derivados do paralelismo de estações, Boysen e Flidner (2008) apresentam o seguinte modelo para tratar desta particularidade:

$$Max \quad N = db \cdot \frac{T}{c} - m \cdot K^s - \sum_{i \in V} \sum_{k \in M} x_{ik} \cdot (1 + p_k) \cdot K_i^{AG} \quad (26)$$

s.a equações 3; 4 e 6

$$\left(\frac{\sum_{i \in V} x_{ik} \cdot t_i}{1 + p_k} \right) \leq c \quad \forall k \in M \quad (27)$$

$$\sum_{k \in M} (k \cdot x_{nk} + p_k) \leq m \quad (28)$$

$$p_k = \{0; 1; 2; \dots; P\} \quad \forall k \in M \quad (29)$$

Onde: p_k = número total de estações adicionais em paralelo à estação k (se $p_k=0$, não há paralelismo).

Na função objetivo (26) são incluídos os custos de atribuir atividades à estações em paralelo (K_i^{AG}). Como a estação k terá seus recursos multiplicados em p_k vezes para serem trabalhados de forma alternada, a equação 27 ajusta a capacidade desta estação ao seu nível de paralelismo sendo que este não deve ser superior a um dado valor P (equação 29). A equação 28, ao coletar o índice da última estação e somar todos os paralelismos existentes no problema, registra, em m , o número total de estações utilizadas.

A mesma base é utilizada para retratar a situação de atividades em paralelo que, segundo os mesmos autores, fica da seguinte forma:

$$\text{Max } N = db \cdot \frac{T}{c} - m \cdot K^s - \sum_{i \in V} \sum_{k \in M} x_{ik} \cdot K_i^{AG} \quad (30)$$

s. a equação 6

$$1 \leq \sum_{k \in M} x_{ik} \leq p'_i \quad \forall i \in V \quad (31)$$

$$\max_{k \in M} (k \cdot x_{ik}) \leq \max_{k \in M} (k \cdot x_{jk}) \quad \forall (i, j) \in E \quad (32)$$

$$\max_{k \in M} \left(\sum_{i \in V} \frac{x_{ik} \cdot t_i}{\sum_{k' \in M} x_{ik'}} \right) \leq c \quad (33)$$

$$\max_{k \in M} (k \cdot x_{nk}) \leq m \quad (34)$$

Na função objetivo (equação 30), o paralelismo de cada atividade i é associado ao custo adicional K_i^{AG} . As restrições orientam que todas as tarefas sejam alocadas ao menos uma vez podendo ser repetidas até uma dada quantidade p'_i (equação 31). Na equação 32 são colocadas as restrições de precedência e na equação 33 o tempo de ciclo é ajustado, pois agora o tempo efetivo de execução da atividade depende do seu nível de paralelismo. Já a equação 34 registra em m o número total de estações utilizadas.

Além das situações que objetivam a minimização dos custos ou maximização dos lucros, outra condição que pode ser explorada diz respeito à redução do tempo de ciclo (SALBP-2). Para isto, Boysen e Flidner (2008) indicam a seguinte formulação:

$$\text{Min } c \quad (35)$$

s.a equações de 3 à 6

$$\sum_{k \in M} x_{ik} \cdot k \leq m \quad \forall i \in V \quad (36)$$

Nesta situação o tempo de ciclo (c) deve ser minimizado (equação 35) não extrapolando certo número de estações (equação 36) e respeitando todas as demais restrições do SALBP-1.

Ainda os mesmos autores propõem um modelo para tratar do caso SALBP-E. Neste tipo de problema nem o número de estações e nem o tempo de ciclo são fixos, eles podem variar dentro de limites estabelecidos (conforme equações 38 e 39, respectivamente) para minimizar alguma medida de eficiência. Os autores usam como exemplo a minimização do tempo total disponível para as operações através do seguinte modelo:

$$\text{Min } N = c \cdot m \quad (37)$$

s.a *equações de 3 à 6 e equação 32*

$$m_{\min} \leq m \leq m_{\max} \quad (38)$$

$$c_{\min} \leq c \leq c_{\max} \quad (39)$$

E por último, um dos objetivos deste trabalho é tratar da situação de linhas de produção com dois lados. Esta situação é modelada matematicamente por Kim et al. (2009). Com algumas modificações nas equações propostas, pode-se descrever o problema da seguinte forma:

$$\text{Min } T_{\max} \quad (40)$$

$$\text{s.a } \sum_{k \in M} \sum_{l \in \{1,2\}} x_{ik}^l = 1 \quad \forall i \in V - V_1 \cup V_2, \quad (41)$$

$$\sum_{k \in M} x_{ik}^1 = 1 \quad \forall i \in V_1 \quad (42)$$

$$\sum_{k \in M} x_{ik}^2 = 1 \quad \forall i \in V_2 \quad (43)$$

$$w_i = l \sum_{k \in M} x_{ik}^l \quad \forall i \in V \quad (44)$$

$$y_i = k \sum_{k \in M} \sum_{l \in \{1,2\}} x_{ik}^l \quad \forall i \in V \quad (45)$$

$$T_i = T_i^0 + t_i \quad \forall i \in V \quad (46)$$

$$c(y_i - 1) \leq T_i \leq c \cdot y_i \quad \forall i \in V \quad (47)$$

$$c(y_i - 1) \leq T_i^0 \leq c \cdot y_i \quad \forall i \in V \quad (48)$$

$$T_{max} - T_i \geq 0 \quad \forall i \in V \quad (49)$$

$$T_j^0 - T_i \geq 0 \quad \forall (i, j) \in E \quad (50)$$

$$T_j^0 - T_i + Z(1 - \delta_{ij}) \geq 0 \quad \left. \vphantom{T_j^0} \right\} \forall (i, j) \in V \quad (51)$$

$$T_i^0 - T_j + Z(1 - \delta_{ji}) \geq 0 \quad \left. \vphantom{T_i^0} \right\} \forall (i, j) \in V \quad (52)$$

$$\delta_{ij} + \delta_{ji} = 1 - |w_i - w_j| \quad \left. \vphantom{\delta_{ij}} \right\} \forall (i, j) \in V \quad (53)$$

$$x_{ik}^l \in \{0; 1\} \quad \forall i \in V; k \in M; l \in \{1; 2\} \quad (54)$$

$$\delta_{ij} \in \{0; 1\} \quad \forall i, j \in V \quad (55)$$

$$y_i \in \mathbb{N}^* \quad \forall i \in V \quad (56)$$

$$w_i \in \{1; 2\} \quad \forall i \in V \quad (57)$$

$$T_i^0; T_i \geq 0 \quad \forall i \in V \quad (58)$$

Onde: $x_{ik}^l = 1$ se atividade i é alocada para estação k e no lado l . É zero em caso contrário.

V_1 = conjunto de atividades que só podem ser alocadas no lado esquerdo.

V_2 = conjunto de atividades que só podem ser alocadas no lado direito.

T_{max} = tempo acumulado em relação ao início do processo em que a última atividade termina.

T_i^0 = tempo acumulado em relação ao início do processo em que a atividade i é iniciada

T_i = tempo acumulado em relação ao início do processo em que a atividade i é termina

w_i = armazena o código do lado $\{1;2\}$ para o qual a atividade i foi alocada.

y_i = armazena o código da estação para a qual a atividade i foi alocada.

$\delta_{ij} = 1$, se tarefa i é realizada antes de j e zero em caso contrário.

Trata-se de uma modificação geral no modelo matemático do SALBP-1. Isto porque, ao trabalhar com o problema simples, as tarefas são simplesmente distribuídas em suas respectivas estações sem a

necessidade de observar a ordem com que são executadas. Esta não é uma preocupação nesta situação, pois, para as atividades contidas em uma estação, sempre há uma ordem tal que satisfaça as restrições de precedência existentes. Entretanto, quando trabalhando com linhas de dois lados, deve-se, necessariamente, orientar a alocação das tarefas observando seu tempo de início e fim. No caso de restrições de precedência entre atividades endereçadas para lados distintos, elas não podem se sobrepor no tempo. Isto só foi possível ao introduzir as variáveis T_i^0 e T_i que controlam o tempo acumulado do momento de início e término da atividade i , respectivamente.

Na função objetivo (equação 40) tem-se a minimização do tempo em que a última tarefa foi terminada. Este tempo de término é balizado pela equação 49 que orienta T_{max} a ser um valor superior ao tempo final de todas as atividades. Assim, minimizar T_{max} equivale a reduzir o tempo total da linha e, conseqüentemente, o número de postos de trabalho.

As equações 41, 42 e 43 indicam que todas as atividades devem ser alocadas uma única vez. Aquelas com lado definido, entretanto, só podem ser alocadas em seu respectivo lado (equações 42 e 43).

Algumas variáveis de controle foram incluídas para facilitar a formulação do modelo, como é o caso de w_i , que armazena o código do lado para o qual cada estação foi endereçada (equação 44), e y_i , que armazena o código da estação em que cada atividade foi endereçada (equação 45). Outro artifício utilizado para facilitar a escrita das equações foi a implantação da equação 46 que calcula o tempo de término de cada atividade (T_i) a partir de seu tempo de início.

Trabalhando em conjunto, as equações 47 e 48 orientam que o tempo inicial e final das tarefas deve fazer parte da mesma estação. Em outras palavras, a tarefa deve estar contida dentro do tempo de ciclo.

As restrições de precedência são expressas pela equação 50 que impede uma atividade iniciar antes do término de todas suas antecessoras.

E finalmente, para evitar a sobreposição de atividades ao longo do tempo, tem-se as equações 51, 52 e 53 trabalhando em conjunto. Na equação 51, caso a atividade i seja alocada antes da j , o tempo final de i deve ser igual ou inferior ao tempo inicial de j . Já na equação 52, caso atividade j seja alocada antes da i , o tempo final de j deve ser igual ou inferior ao tempo inicial de i . Para cada par de tarefas alocadas para um mesmo lado somente uma destas restrições deve ser acionada. Para isso, utilizou-se um fator Z , que representa um valor muito grande, que,

associado à variável δ_{ij} , habilita (caso $\delta_{ij} = 1$) ou desabilita (caso $\delta_{ij} = 0$) cada uma das equações. Como estas relações só são válidas para tarefas do mesmo lado, as variáveis w_i e w_j foram incluídas na equação 53. Caso as tarefas i e j sejam de lados distintos, $\delta_{ij} = \delta_{ji} = 0$ e ambas as restrições são desabilitadas. Caso compartilhem o mesmo lado, ou seja $w_i = w_j$, somente δ_{ij} ou δ_{ji} será 1 enquanto o outro será 0. Assim, somente a equação 51 ou 52 será utilizada para aquele par de atividades.

Apesar de ser possível descrever matematicamente as mais diversas particularidades de um problema de ALB, resolvê-los ainda é tarefa difícil. Pela teoria da complexidade computacional, que estuda os recursos necessários durante a resolução de um problema, mesmo o modelo matemático mais simples que foi apresentado (SALBP-1) é classificado como NP-Completo (KIM et. al. 2000).

Assim, para problemas de maior porte, os tempos computacionais podem ser impraticáveis. Por essa razão, os modelos matemáticos raramente são utilizados ou mesmo apresentados na literatura, além disso, não há uma formulação geral para abranger todas as particularidades comentadas. Para contornar este problema, várias técnicas de solução baseadas em heurísticas e meta-heurísticas têm sido desenvolvidas ao longo do tempo. Estas serão apresentadas na seção 2.3.

2.3 Técnicas de solução

Bautista e Pereira (2009) classificam as técnicas de solução de um ALB em três tipos: modelos matemáticos, heurísticas e meta-heurísticas. Mezzana (2000 apud FARNES e PEREIRA, 2006) indica ainda o balanceamento através de simulação.

Nestes procedimentos, exceto no caso de aplicação de modelos matemáticos, não há garantia da solução ótima e o principal critério de comparação é a qualidade da solução (AMEM, 2001).

2.3.1 Heurísticas para ALB

Grande parte das heurísticas de balanceamento faz uso de uma lista ordenada de atividades que são aplicadas a um alocador para encontrar uma solução. As heurísticas, por sua vez, buscam, cada uma a

sua maneira, descobrir a ordenação desta lista de tarefas que, ao ser aplicada no alocador, otimize o objetivo almejado. Assim tem-se três passos principais:

- a) Classificação das tarefas para construção da lista
- b) Aplicação do alocador
- c) Análise dos resultados

Um primeiro grupo de heurísticas são as de primeira alocação. Nelas, os passos acima descritos são realizados uma única vez e o que difere umas das outras é a forma utilizada para classificação das tarefas.

Carravilla (1998) e Aguiar (2007) sugerem que a lista de tarefas seja ordenada em relação decrescente ao tempo de execução das atividades. Ponnambalam et al. (1999) reúnem os trabalhos de vários autores como Helgeson-Birnie (1961), Kilbridge e Wester (1961) e Hackman (1989) que apresentam formulações distintas para ordenação da lista de tarefas. O primeiro usa o somatório dos tempos das tarefas sucessoras, o segundo, toma por base o atendimento às restrições de precedência e o terceiro trabalha com uma composição de multi-critérios (peso posicional, peso posicional reverso, número total de sucessores, número de sucessores diretos, número total de predecessores, número de predecessores diretos) para tal classificação. Grzechca (2008), com o objetivo de comparação, aplica cada um destes critérios individualmente para medir sua eficiência.

Independente da métrica utilizada para ordenação da lista de tarefas, no passo seguinte, ou seja, na alocação destas atividades, Barthold (1993) sugere os seguintes passos.

- a) Encontre a próxima tarefa da lista, aquela que teve todas as antecessoras endereçadas.
- b) Encontre a última estação i para a qual uma antecessora foi endereçada, e para esta estação encontre o último tempo de término t das atividades antecessoras.
- c) Começando na estação i , a partir do tempo t , encontre o primeiro lugar no cronograma no qual a atividade pode ser inserida, observando as restrições impostas.

Outras heurísticas mais elaboradas, mas ainda assim baseadas nos três passos apontados anteriormente (classificar, alocar e comparar resultados), mas agora utilizados de forma recursiva, também são objetivo de estudo de Ponnambalam et al. (1999), como é o caso dos modelos de Moodie-Young (1965) e HOFFMANN (1963). No primeiro caso, uma ordenação decrescente relacionada ao tempo de processamento das tarefas é utilizada para criar uma solução inicial. Posteriormente, são identificados potenciais de transferência de

atividades de uma estação para outra ou de trocas de posição entre atividades de estações distintas. No segundo caso, são iterativamente testadas todas as combinações possíveis para preencher o tempo disponível em uma estação antes de passar para a próxima. Devido a esta grande carga combinatorial, este método tem grande custo computacional.

Como dito, não se pode garantir otimalidade com tais procedimentos e a qualidade de seu resultado está muito relacionada às características do problema em estudo.

2.3.2 Meta-Heurísticas para ALB

De acordo com Hörner (2009), as meta-heurísticas são métodos de busca que, por sua flexibilidade, podem se adaptar a diversos problemas de otimização. Elas são divididas em duas categorias: técnicas de busca local e busca populacional. Busca Tabu e *Simulated Annealing* são exemplos da primeira categoria, pois partem de uma solução inicial e exploram soluções vizinhas na procura de melhor aderência aos objetivos estabelecidos. Já o Algoritmo Genético e *Scatter Search* são exemplos da segunda categoria, pois, a partir de uma população inicial, utilizam diversas operações entre os indivíduos na tentativa de coletar as melhores qualidades neles existentes.

O processo de Busca Tabu foi introduzido por Glover (1989, 1990). Este método consiste basicamente em realizar pequenas perturbações, ou seja, pequenos movimentos na solução até então encontrada, sem que estes resultem em situações restritas pela Lista Tabu. Como o próprio nome sugere, a lista tabu é um conjunto de soluções que não pode ser utilizado (a não ser que um critério de aspiração seja atingido). Ela, para evitar ótimos locais, geralmente é composta por uma memória dos últimos movimentos realizados.

Em ALB, Scholl e Voß (1996), usam a Busca Tabu sempre com objetivo de reduzir o tempo de ciclo (SALBP-2). Quando a intenção é reduzir o número de estações (SALBP-1), aplica-se a mesma metodologia, mas de maneira recursiva, adotando-se um número crescente de estações, até que o tempo de ciclo encontrado seja compatível com o desejado. Esta forma de trabalho foi motivada, pois não há, na aplicação direta ao SALBP-1, uma medida natural para encontrar bons movimentos. Isto porque, seriam necessárias várias

iterações até que o número de estações pudesse reduzir em uma unidade, o que não orientaria claramente o processo de busca.

Tendo em vista este problema na aplicação direta ao SALBP-1, Chiang (1998) faz uso de uma função objetivo mais elaborada que orienta os postos de trabalho a utilizar ao máximo suas capacidades.

Lapierre et. al. (2006), conseguem melhores resultados ao incorporar dois aspectos no modelo de Chiang: a intensificação na busca por oportunidades de melhoramento da solução (procurando elevar ou reduzir o número de postos com meia capacidade preenchida e tentando esvaziar os postos que estão quase vazios) e a redefinição do espaço de soluções para permitir a visita a soluções inviáveis.

Outra meta-heurística utilizada no ALB é o *Simulated Annealing*, que é análoga a reorganização atômica ocorrida durante o processo de aquecimento de materiais. Introduzida por Kirkpatrick et. al. (1983), ela realiza um processo de busca local em que a aceitação de um estado e^* em detrimento ao estado atual e é associada a uma probabilidade P que ocorre conforme indicado na equação 59.

$$P = \begin{cases} 1, & \text{se } e^* \text{ é melhor que } e; \text{ e} \\ e^{(-|e-e^*|/k_b \cdot T)}, & \text{em caso contrário.} \end{cases} \quad (59)$$

A expressão exponencial utilizada para definição da probabilidade é análoga ao fator de Boltzmann onde “ k_b ” é a constante de Boltzmann e “ T ” é a temperatura do sistema que vai diminuindo ao longo das iterações.

McMullen e Frazier (1998), realizando trocas e transferências entre atividades alocadas para estações vizinhas, mostram uma boa adaptação do *Simulated Annealing* em ALB.

Usando uma formulação peculiar para alocação das atividades, em que um vetor de ordem igual ao número de atividades, mas guardando uma referência a regra de classificação utilizada para selecionar as atividades da lista de candidatas, Baykasoglu (2006), obteve grande êxito ao aplicar o *Simulated Annealing* em uma série de problemas disponíveis na literatura.

De maneira mais discreta, mas ainda assim importante, o método de otimização Ant Colony (Colônia de Formigas) também aparece na literatura sobre ALB. Proposta por Marco Dorigo, em 1992 (DORIGO, 2009), trata-se de uma técnica probabilística para encontrar os caminhos mais curtos em grafos. Ela é baseada no comportamento das formigas, que, ao deixar as colônias para buscar alimento, se orientam pelo feromônio deixado pelas outras. O caminho com maior incidência desta substância certamente foi visitado por um número maior de formigas e,

provavelmente, confere o caminho mais curto ao objetivo. Entretanto, a idéia inicial de busca de caminhos mínimos foi estendida para solução de vários problemas de ordem combinatória.

Utilizando tratamento similar ao de Scholl e Voß (1996) para resolução do SALBP-1, Bautista e Pereira (2007) aplicaram a otimização pelo método Colônia de Formigas (com busca local baseada na transferência individual das atividades para outras estações) com resultados próximos aos da Busca Tabu.

Por último, uma meta-heurística que tem destaque nos problemas de ALB é o Algoritmo Genético. Este, por sua vez, foi escolhido como ferramenta de auxílio neste trabalho, pois apresenta uma excelente versatilidade para aplicação em situações diversas. A característica de fácil adaptação faz-se fundamental neste caso, porque, como comentado anteriormente, há uma série de particularidades no problema de balanceamento e cada uma delas exige modificações na estrutura do modelo de resolução. Assim, além de garantir a flexibilidade exigida para tratar dos casos de linhas de dois lados, layouts em “U” e restrições de alocação, que são objeto de estudo deste trabalho, outras particularidades do problema podem ser mais facilmente adicionadas futuramente. O Algoritmo Genético, como objeto de estudo da presente pesquisa, será explorado em detalhes na seção 3.

2.4 Pontos principais

Como apresentado na revisão bibliográfica, não há modelos versáteis o suficiente para tratar conjuntamente as condições especiais que envolvem os problemas de ALB (BOYSEN; FLIEDNER, 2008). Nestas condições a aplicabilidade dos trabalhos acadêmicos fica muito restrita.

A julgar pelo patamar em que se encontram os estudos sobre o tema, a união de todos estes aspectos em um modelo único é uma ambição ainda distante. Para chegar mais próximo deste objetivo, entretanto, algumas características comumente encontradas no ambiente empresarial e enfatizadas na literatura foram reunidas para serem operadas paralelamente.

Por exemplo, com a expansão dos conceitos *Lean* entre os sistemas de produção, algumas configurações especiais, como o caso das linhas em U (β_2), passaram a ter grande representação na indústria (KRIENGGORAKOT; PIANTHONG, 2007).

Outra condição importante em quase todos os problemas da indústria é, para Boysen e Fliedner (2008), a inclusão das restrições de alocação (α_5). Isto porque, além da natural afinidade ou incompatibilidade entre tarefas, muitas vezes o balanceamento é realizado sobre uma linha já implantada. Nestas condições, mudanças de *layout* podem ter um custo proibitivo, sendo mais econômico adaptar a seqüência de atividades para um certo *layout* já em operação.

Outra situação importante e quase inexplorada é o balanceamento em linhas com dois lados. Esta configuração é utilizada na montagem de produtos de grandes dimensões (KIM et al., 2000). Nesta categoria se encaixam os processos finais de quase todas as indústrias automobilísticas com a montagem dos periféricos no veículo em processo. Esta condição, segundo Bartholdi (1993), traz benefícios ao reduzir o comprimento da linha. Isto influencia diretamente nos custos de implantação, na facilidade em criar sinergia na utilização de recursos, na movimentação de materiais, etc.

Atender de uma só vez estas características configura um passo importante na redução da distância entre os problemas da literatura e os encontrados na realidade. Para atender a esta aspiração e, ainda assim, garantir flexibilidade para inclusão de novas particularidades em problemas futuros, é necessária a utilização de uma ferramenta com boa versatilidade como é o caso do Algoritmo Genético.

3 MODELO E MÉTODO IMPLEMENTADOS

3.1 Caracterização do problema

No presente estudo foi implementado um modelo para resolver problemas de ALB utilizando algoritmo genético. Das particularidades que complementam este assunto as seguintes serão tratadas com um método único:

- a) Restrições de alocação (α_5);
- b) Linhas com dois lados;
- c) Linhas com layout em U (β_2).

3.2 Estrutura geral do método de busca

Como visto anteriormente, mesmo em sua forma mais primitiva, os problemas de ALB são classificados como NP-Completo. Para resolução de problemas com esta complexidade geralmente faz-se uso de técnicas de inteligência computacional. Estas ganharam popularidade nas últimas décadas ao tratar de problemas difíceis de serem resolvidos pelos métodos clássicos da matemática (MENG et al, 1999).

Um destes métodos utilizados em problemas e ALB é o Algoritmo Genético. Segundo Hörner (2009), a adaptação da teoria de evolução das espécies para resolução de problemas combinatoriais foi inicialmente estudada por John Holland na década de 1970 originando tal algoritmo.

Com base na descrição de Bittencourt (2009), os passos básicos do algoritmo genético podem ser resumidos da seguinte forma:

Início Algoritmo

- Inicializar a população de cromossomos;
- Avaliar cada cromossomo da população;

Repetir

- Selecionar indivíduos da população atual para realizar as operações genéticas (*crossover* e *mutação*) a fim de gerar novos indivíduos;
- Avaliar os descendentes gerados;

Substituir os ascendentes menos aptos pelos descendentes mais aptos.

Até critério de parada ser alcançado;

Fim Algoritmo.

O uso deste algoritmo em problemas de ALB é destacado por Tasan e Tunalin (2008) que consolidam informações de vários autores comentando os ajustes necessários à técnica original.

3.2.1 Representação do cromossomo

No trabalho de Tasan e Tunalin (2008) foram identificadas seis formas distintas de representar um cromossomo nos problemas de ALB. É importante ressaltar que a clássica representação binária é raramente utilizada devido a dificuldade em conciliar sua estrutura com as informações necessárias sobre os indivíduos. Restam, portanto, cinco tipos, que são:

- a) baseados nas tarefas: o cromossomo tem comprimento igual ao número de tarefas e é formado de tal maneira que sua seqüência não infrinja as relações de precedência. É a forma mais apropriada quando o objetivo é a redução do número de postos de trabalho.
- b) representação de um embrião: similar ao baseado nas tarefas, mas representa somente uma parte das atividades. O tamanho inicial do cromossomo é definido aleatoriamente e, ao longo das iterações, vai incrementando até atingir o número total de tarefas.
- c) baseado nas estações: o cromossomo é representado por um vetor com tantas posições quanto o número de tarefas, porém, armazenando o código das estações. Dessa forma, ao consultar certo parâmetro deste vetor, seu valor corresponde à estação em que a tarefa foi alocada, e seu índice, à tarefa em si. Esta configuração é muito popular em problemas que almejam a redução do tempo de ciclo.
- d) baseado em agrupamentos: para adaptar a representação baseada nas estações para o problema de redução de postos de trabalho, ao final do cromossomo original foram adicionadas mais algumas posições para armazenar a lista das estações utilizadas na solução. Desta forma, reduzir o tamanho do cromossomo equivale a reduzir o número de estações.

- e) indireta baseada em heurísticas: o cromossomo contém uma lista de heurísticas que são usadas, nesta ordem, para selecionar as próximas tarefas a serem alocadas e gerar as soluções desejadas.

Exemplos da representação dos cromossomos para cada um dos cinco tipos apresentados acima são reunidos na figura 3.

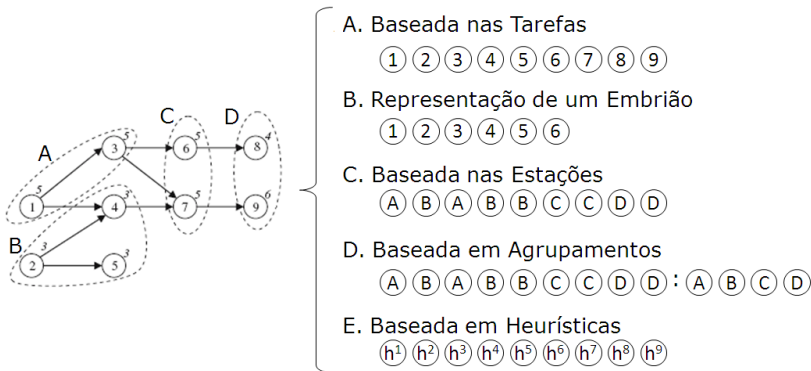


Figura 3 – Tipos de representação do cromossomo

Fonte: Tasan e Tunalin (2008)

Como o objetivo inicial é a redução dos postos de trabalho, a representação utilizada foi a baseada nas tarefas. Nesta forma somente informações sobre a tarefa são armazenadas pelo cromossomo. Para conhecer a solução que cada cromossomo representa deve-se aplicar o alocador de tarefas que será explicado na seção 3.3.

3.2.2 Construção da População Inicial

A entrada de dados principal para resolução do problema é uma *Lista de Tarefas* que contém, para cada uma das atividades, o tempo necessário para execução e a relação das atividades que as antecedem.

A preocupação neste ponto é, a partir desta lista inicial, gerar inúmeras combinações aleatórias garantindo que as restrições de precedência sejam respeitadas para facilitar o trabalho de alocação das tarefas. Para isto, os seguintes passos foram aplicados:

Início Algoritmo**Repetir**

Carregar a *Lista de Tarefas* (conforme entrada de dados) e fazer $i = 0$;

Repetir

Varrer a Lista de Tarefas buscando atividades que já tiveram todas as suas predecessoras alocadas;

Adicionar tais tarefas na *Lista de Candidatas* e excluí-las da *Lista de Tarefas*;

Selecionar aleatoriamente uma tarefa da *Lista de Candidatas*;

Adicioná-la como último elemento do cromossomo; e excluí-la da *Lista de Candidatas*.

Até *Lista de Candidatas* ser nula.

$i = i + 1$.

Até i = tamanho desejado para população.

Fim Algoritmo.

Assim são criados todos os cromossomos que farão parte da população inicial¹. Vale ressaltar que, para o caso de linhas com layout em formato U, a construção da lista de candidatas sofre uma pequena alteração. Para esta situação, além das atividades que tiveram todas as suas predecessoras escolhidas, também devem fazer parte da lista de candidatas, as atividades que tiveram todas as suas sucessoras selecionadas.

Os indivíduos da população gerada passarão por um processo evolutivo, através das operações do algoritmo genético, de modo a garantir a transmissão das características mais adequadas para as próximas gerações, conforme detalhamento nas próximas seções.

Quanto ao tamanho da população, ela não pode ser tão pequena que não gere a diversificação necessária para explorar um bom número de opções, nem tão grande que torne a convergência muito lenta. Em testes empíricos com uma série de problemas da literatura, populações acima de 1000 (mil) indivíduos não geraram melhoras significativas nos resultados, mas comprometeram o tempo computacional. Assim, este número foi adotado como tamanho fixo para a população.

¹ Naturalmente, para o correto funcionamento do modelo apresentado, a entrada de dados deve representar um problema que tenha solução (grafo de precedência acíclico).

3.2.3 Avaliação do Fitness dos Indivíduos

Para que o processo evolutivo caminhe conforme o objetivo desejado, é necessário avaliar a aptidão de cada indivíduo em continuar como integrante das gerações futuras. Esta avaliação é dada por uma função chamada *fitness* que deve orientar cada movimento da meta-heurística adotada. Sem esta orientação, todos os eventos seriam puramente estocásticos não havendo garantia alguma de convergência.

Nos problemas de ALB, cujo objetivo é a redução dos postos de trabalho, esta função é difícil de ser estabelecida (TASAN E TUNALI, 2008). Isto porque, ao indicar uma função trivial, como por exemplo, o número de postos de trabalho utilizados, não haveria clara distinção entre o *fitness* de uma solução e outra até que um posto fosse efetivamente eliminado.

Scholl e Voß (1996) contornam esta situação de forma indireta. O objetivo do problema foi alterado para reduzir o tempo de ciclo ($\gamma=c$). Este foi resolvido diversas vezes, perturbando o número de estações, até encontrar uma quantidade tal de postos que garantam o máximo tempo de ciclo sem ultrapassar um valor dado pelo problema original. Esta condição, apesar de útil, tem um consumo computacional elevado por resolver o mesmo problema diversas vezes.

Outras funções, entretanto, podem ser encontradas na literatura para tratar diretamente deste caso. Algumas delas são reunidas no trabalho de Tasan e Tunalin (2008). De maneira geral, as funções de *fitness*, cada uma a sua maneira, fazem uso do tempo útil ou do tempo ocioso das estações, tentando maximizá-lo ou minimizá-lo respectivamente.

Adaptada a partir da função objetivo de Chiang (1998), que sugere o acoplamento de atividades de modo a preencher o tempo útil das estações de trabalho, a função de *fitness* empregada neste trabalho é expressa pela equação 60.

$$Max \frac{1}{1,1^{|M^{ST}|}} \left[\sum_{k \in M^{ST}} \frac{(\sum_{i \in V} t_i x_{ik})^2}{1,1^k} \right] \quad (60)$$

Ao elevar ao quadrado o somatório dos tempos das atividades de cada estação, aquelas soluções cujas estações apresentam maior carga relativa de trabalho, são nitidamente priorizadas. Quanto ao fator $1,1^k$ adotado como divisor, este tem por objetivo priorizar soluções cujas estações situadas nas primeiras posições estão mais próximas da

capacidade. Quanto ao fator $1,1^{|M^{ST}|}$, este serve de penalização por acréscimos desnecessários no número total de postos de trabalho da solução encontrada.

Para considerar as restrições de alocação (α_5) a função acima descrita é penalizada através da divisão por um fator que mede a inconsistência da solução encontrada. Para cada restrição desrespeitada, seja entre tarefas ou entre tarefa e estação, é medida, em número de postos de trabalho, a distância desta incompatibilidade e depois somada ao fator comentado. Por exemplo, em um problema do tipo $\alpha_5 = fix$ em que certa atividade i deveria ser alocada na estação k mas foi alocada para estação $k+2$, a distância considerada para esta incompatibilidade é igual a dois. Este valor é então acumulado dentro do fator comentado. Este fator soma todas as distâncias encontradas em cada uma das inconsistências e depois é usado como divisor da equação 60. Esta situação cria uma penalidade tal que dificulta a visita de soluções que desrespeitem as restrições de alocação impostas ao problema em estudo.

O *fitness* de cada indivíduo da população está intimamente ligado ao critério de parada adotado para o processo evolutivo do algoritmo genético. No modelo construído, caso a diferença entre o melhor e o pior elemento da população seja inferior a 0,1%, o processo é dado por encerrado.

3.2.4 Cruzamento ou *Crossover*

Para Hörner (2009), o operador de *crossover*, ou cruzamento, é empregado para criar novos indivíduos em uma população através da recombinação das características dos indivíduos existentes.

Tradicionalmente, isto é feito através da permuta entre seções aleatórias dos cromossomos ancestrais. Em problemas de ALB, entretanto, se realizado sem tomar alguns cuidados, este procedimento pode gerar indivíduos cuja ordem das atividades não respeite as restrições de precedência.

Tasan e Tunalin (2008) comentam uma série de métodos encontrados na literatura para realização do *crossover* sem comprometer a viabilidade do elemento criado. São elas:

- a) *Modified Bin Packing Crossover*: desenvolvido especialmente para os cromossomos com representação

baseada em agrupamentos. É aplicado somente a parte do cromossomo referente às tarefas.

- b) *Order Crossover*: Gera duas seções aleatórias em dois cromossomos genitores. As partes inicial e final do primeiro genitor são mantidas no cromossomo gerado. A parte central é preenchida com as tarefas faltantes, mas na ordem em que aparecem no segundo genitor. De forma análoga, entretanto trocando os papéis do primeiro e do segundo genitor, um segundo cromossomo é gerado (SABUNCUOGLU et al., 2000).
- c) *Modified Partially Mapped Crossover*: Com duas seções em dois cromossomos pais, os elementos centrais são trocados para gerar dois filhos. Mantendo a parte central dos novos indivíduos inalterada, as demais atividades são reordenadas até que as restrições de precedência sejam respeitadas.
- d) *Heuristic Structural Crossover*: Usa heurísticas específicas para selecionar grupos de estações para serem copiados para gerações futuras. Incompatibilidades com as restrições de precedência são tratadas com um procedimento de adaptação (KIM et al., 1998).
- e) *Uniform Order-Based Crossover*: a partir de dois pais, dois novos indivíduos são gerados. Primeiramente uma lista binária de mesmo tamanho dos cromossomos é criada. Seguindo os elementos desta lista, caso seu valor seja 1, copia-se para o novo cromossomo o elemento de mesma ordem do primeiro cromossomo pai. Os demais elementos (nas mesmas posições que os índices 0) são preenchidos na ordem em que aparecem no segundo cromossomo pai. O segundo indivíduo gerado segue construção análoga, mas trocando a função do 0 por 1 e vice-versa (CHAN et al., 1998).

Neste trabalho, a ferramenta desenvolvida para realização do *crossover* tem resultado similar ao apresentado por Chan et al. (1998). Entretanto, em vez de trabalhar com a construção de uma lista binária, foi desenvolvida uma heurística de passagem simples que, percorrendo dois indivíduos da população, coleta, aleatoriamente, as tarefas de um ou de outro. Este procedimento é descrito pelo algoritmo abaixo:

Início Algoritmo $i_1 = 0$ e $i_2 = 0$;Selecionar aleatoriamente dois cromossomos pais ($P1$ e $P2$) a partir da população;**Repetir**Escolher aleatoriamente entre $P1$ e $P2$;**Caso $P1$:**Se tarefa i_1 não foi escolhida:

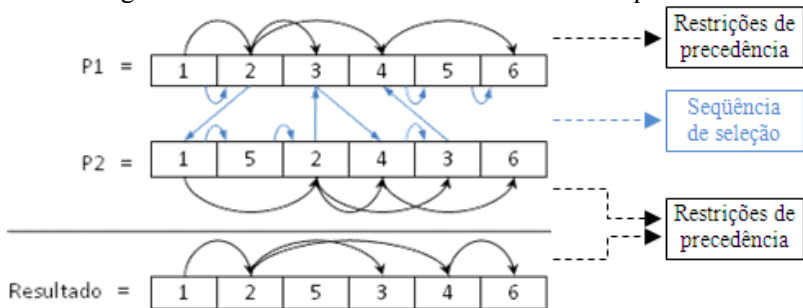
colocá-la no cromossomo filho.

 $i_1 = i_1 + 1$.**Caso $P2$:**Se tarefa i_2 não foi escolhida:

colocá-la no cromossomo filho.

 $i_2 = i_2 + 1$;**Até** i_1 ou $i_2 =$ número de genes dos cromossomos pais,**Fim** Algoritmo.

A figura 4 ilustra o funcionamento da heurística apresentada.

Figura 4 – Esquema da heurística utilizada para realização do *crossover*

Fonte: Autor (2010)

As setas em preto representam as restrições de precedência entre as atividades e as setas em azul mostram um caminho aleatório percorrido pela heurística. Inicialmente P1 é selecionado e seu primeiro elemento, “1”, é copiado para o cromossomo resultado. Na segunda iteração, P1 é novamente escolhido e transfere a atividade “2” para o novo elemento. Na terceira iteração, P2 é escolhido, mas o elemento da ordem, ou seja, “1”, já foi escolhido. Nova iteração ocorre, com a seleção de P2, e a atividade “5” é selecionada. Isto se repete até chegar ao último elemento de P1 ou P2.

Como P1 e P2 estão dispostos de tal forma que todas as restrições de precedência são respeitadas, qualquer novo indivíduo que seja criado com estes procedimentos também as respeitará. Isto porque, um gene de ordem k , nos cromossomos pais, só é visitado se todos os genes anteriores a k (0, 1, ..., $k-1$) já foram selecionados.

3.2.5 Mutação

Após realização do *crossover*, o elemento resultante é submetido ao processo de mutação. Este evento insere, com uma pequena probabilidade, certas perturbações na cadeia de cromossomos. Esta fase da evolução é importante para, segundo Rivera (2001), diversificar a população inicial evitando uma convergência muito rápida para um mínimo local devido ao restrito espaço de busca.

Assim como o *crossover*, a mutação, quando aplicada aos problemas de ALB, pode gerar indivíduos em seqüências inviáveis. Uma série de métodos é enumerada na literatura para realizar este importante passo para o método evolutivo garantindo a viabilidade do elemento criado. Tasan e Tunalin (2008) reúnem alguns destes trabalhos e citam os seguintes métodos:

- a) *Modified Bin Packing Mutation*: também desenvolvido especialmente para os cromossomos com representação baseada em agrupamentos. É aplicado somente a parte do cromossomo referente às tarefas.
- b) *Heuristic Structural Mutation*: com dada probabilidade os elementos do novo cromossomo são selecionados para serem realocados. A troca de posição destas tarefas selecionadas faz uso do mesmo procedimento de adaptação comentado anteriormente (Kim et al., 1998).
- c) *Scramble Sublist Mutation*: monta-se aleatoriamente uma sublista com algumas atividades do cromossomo. Mantendo os demais genes intactos, estes são permutados entre si desde que não interfiram nas relações de precedência (Chan et al., 1998).
- d) *Scramble Mutation*: uma seção é realizada no cromossomo e os elementos anteriores a ela são mantidos. Já os posteriores são aleatoriamente reordenados, sempre respeitando as relações de precedência (Sabuncuoglu et al., 2000).

Neste trabalho, a mutação realizada baseia-se na estrutura do modelo *Heuristic Structural Mutation* de Kim et al. (1998) mas, com pequenas modificações devido ao uso de representações distintas do cromossomo. Os passos para mutação são os seguintes:

Início Algoritmo

Faça $i = 0$;

Repetir

No cromossomo em mutação, com probabilidade de 4%, selecionar a tarefa de ordem i ;

Identificar a posição de maior ordem para a qual uma predecessora da tarefa i foi endereçada. Marcar esta posição como a ;

Identificar a posição de menor ordem para a qual uma sucessora da tarefa i foi endereçada. Marcar esta posição como b ;

Mover a tarefa da posição i para uma posição entre a e b .

$i = i + 1$;

Até $i =$ número de genes do cromossomo em mutação

Fim Algoritmo.

Para o caso de linhas em U os índices a e b são calculados de forma distinta. O índice a é o menor valor entre a maior ordem das predecessora de i e a maior ordem das sucessoras de i . Já o índice b é o menor valor entre a posição de menor ordem para a qual uma sucessora da tarefa i foi endereçada e a posição de menor ordem para a qual uma predecessora da tarefa i foi endereçada.

Durante o processo de mutação, adotou-se 4% como fator de seleção do gene, pois, em testes empíricos com alguns problemas da literatura, notou-se que, com índices inferiores, há uma subotimalidade e, com índices superiores, o tempo computacional aumenta exponencialmente. Sete problemas de tamanhos distintos foram selecionados para realizar os testes conforme tabela 1.

Alterando gradativamente a probabilidade de mutação (de 0 a 10% a passo de 1%), para cada problema e para cada uma das probabilidades testadas foram realizadas 10 (dez) alocações com uso do método descrito. Os valores médios para o número de estações encontrado com estas 10 (dez) operações são apresentados nas figuras 5 e 6. Na figura 5, encontra-se a relação entre a taxa de mutação e o incremento no número de iterações. Já na figura 6, compara-se a mesma taxa com a distância da solução ótima.

Tabela 1 – Problemas utilizados para o teste da probabilidade de mutação.

Problema	Tempo de Ciclo	Numero de Tarefas
Scholl	1394	297
Barthol2	101	148
Arcus2	5755	111
Mukherje	176	94
Warneck	54	58
Kilbridge	56	45
Ghunter	41	35

Nota: Fonte dos problemas usado para teste. Scholl (2007)

Fonte: Autor (2010)

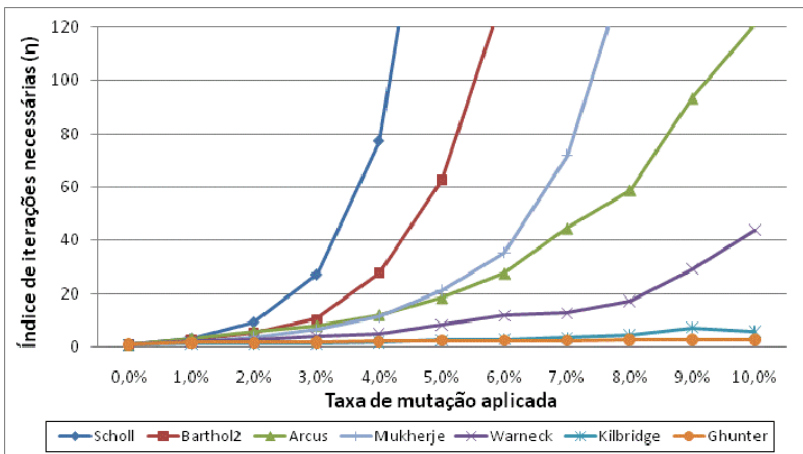


Figura 5 – Relação entre a taxa de mutação e a evolução do número de iterações (média de dez alocações)

Fonte: Autor (2010)

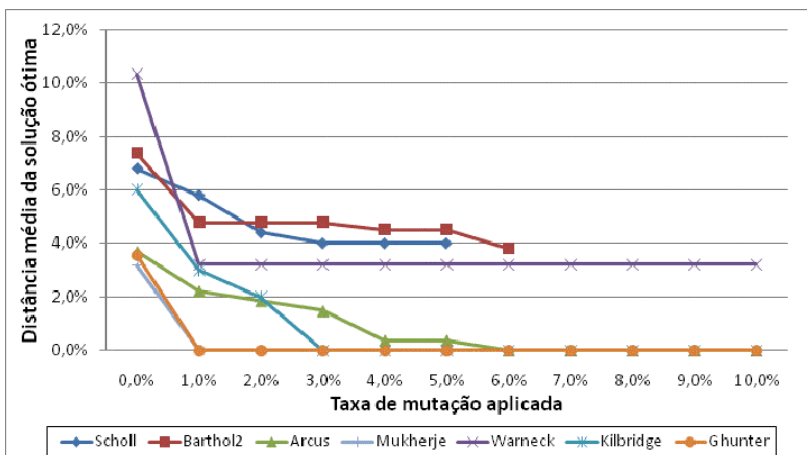


Figura 6 – Relação entre a taxa de mutação e a distância da solução ótima em número de postos (média de dez alocações)

Fonte: Autor (2010)

Na figura 5, para poder comparar a evolução do número de iterações para problemas de dimensões tão distintas, o número de iterações foi normalizado através do índice de iterações necessárias (η) calculado conforme equação 61.

$$\eta = \frac{|\text{itera\c{c}oes}|}{|\text{itera\c{c}oes para 0\% de muta\c{c}ao}|} \quad (61)$$

Ele representa, portanto, quantas vezes mais iterações foram necessárias em relação à condição inicial, em que a taxa de mutação era nula. Ao utilizar taxas superiores a 4%, o número de iterações tem um aumento exponencial o que pode ocasionar tempos computacionais impraticáveis. Este efeito é mais evidente em problemas de maior porte como é o caso de Scholl, Barthol2, Arcus e Mukherje.

Já em relação à qualidade da solução, na figura 6, é possível identificar que taxas de mutação superiores a 4% não representam benefícios expressivos na redução dos postos de trabalho.

Nestas condições, balanceando o desgaste computacional e a qualidade da solução obtida, tal índice demonstra-se apropriado para os exemplos testados, e será adotado como referência para todos os casos estudados.

3.3 Algoritmo de alocação de tarefas

Até agora, todas as operações foram realizadas sobre o cromossomo, que caracteriza o genótipo do indivíduo. Entretanto, a solução propriamente dita, depende da transformação deste genótipo em fenótipo, isto é, depende da distribuição das atividades para suas respectivas estações de trabalho. Para isso, é necessária a aplicação de um alocador que, na ordem definida pelo cromossomo, endereça cada uma das atividades para as estações.

A técnica de alocação de tarefas adotada neste trabalho é similar a apresentada por Barthold (1993). Este autor criou uma heurística que, a partir de uma lista aleatória de atividades, procede com a alocação das tarefas (ver 2.4.1). Segundo o autor, a utilização de uma métrica simples para organizar as tarefas em estações a partir de uma lista qualquer, garante autonomia para o usuário alterar a ordem da lista e obter soluções distintas para o balanceamento.

Esta observação é igualmente importante ao pensar na aplicação de um algoritmo genético aos problemas de ALB. Com a reorganização da seqüência de tarefas através das operações de *crossover* e mutação acima descritas, o algoritmo tenta encontrar aquela combinação que, ao ser submetida ao alocador, garante o resultado ótimo para o problema em questão. Quanto mais simples o procedimento de endereçamento das atividades, menor o trabalho computacional e, conseqüentemente, mais possibilidades podem ser testadas em menor tempo.

Para operação deste procedimento é dada a lista de tarefas contendo, para cada uma, sua duração, o lado em que pode ser endereçada (1 para lado esquerdo, 2 para lado direito e 12 para qualquer lado) e a lista de atividades predecessoras. Com estas informações tem-se o seguinte procedimento:

Início Algoritmo

FimLE \leftarrow 0;

FimLD \leftarrow 0;

i \leftarrow 0;

Enquanto (i < número de tarefas) faça

 Selecionar a i-ésima tarefa do cromossomo;

 TempoMin \leftarrow Tempo de término da predecessora alocada mais próxima;

Caso

```

Lado = 1:
    TempoMin ← Máx (TempoMin, FimLE);
Lado = 2:
    TempoMin ← Máx (TempoMin, FimLD);
Lado = 12:
    Se (FimLE ≤ FimLD)
        Se (TempoMin ≤ FimLE)
            TempoMin ← FimLE;
            Lado ← 1.
        Senão
            TempoMin ← FimLD;
            Lado ← 2.
    Senão
        Se (TempoMin ≤ FimLD)
            TempoMin ← FimLD;
            Lado ← 2.
        Senão
            TempoMin ← FimLE;
            Lado ← 1.
Fim Caso;
Se ((TempoMin + ti) > Próximo tempo de ciclo)
    Criar nova estação;
    TempoMin ← Tempo de início da nova estação.
Alocar tarefa a partir de TempoMin e no Lado definido.
Caso
    Lado = 1:
        FimLE ← TempoMin + ti;
    Lado = 2:
        FimLD ← TempoMin + ti;
Fim Caso;
i = i + 1;
Fim Enquanto;
Fim Algoritmo;

```

Nessas condições, com um procedimento de passagem simples, o cromossomo tem seus genes distribuídos às estações de trabalho. Na seção seguinte, serão apresentados alguns testes com a aplicação deste modelo comparando seus resultados com os obtidos por outros autores.

3.4 Pontos Principais

Mesmo tendo sido comentado algumas vezes, o tempo computacional não foi um objetivo prioritário neste trabalho. Isto porque, as decisões sobre a implantação de novas linhas ou mesmo grandes revisões de suas operações são, principalmente, devido aos custos envolvidos, decisões de longo prazo. Dessa forma, o tempo computacional, desde que não seja exagerado, como no caso das aplicações dos modelos matemáticos, passa a ser uma variável secundária.

Apesar de ser dado foco na resolução de problemas que, além do SALBP-1, tenham restrições de alocação (α_5), linhas com layout em U (β_2) e linhas de dois lados, que é o objetivo deste trabalho, outras situações podem, de maneira indireta, ser igualmente exploradas pelo modelo desenvolvido. Os casos de linhas com produtos mix ($\alpha_1 = mix$) ou com alguma particularidade quanto ao grafo de precedência ($\alpha_2 = spec$), por exemplo, também podem se utilizar do mesmo programa. Como visto na seção 2.1.1, Boysen et al. (2006) sugerem, nestes casos, uma adaptação nos grafos de precedência para convertê-los em problemas triviais. Na mesma seção, os casos de multi produtos ($\alpha_1 = mult$) são também tratados como problema triviais por Wild (1972 apud FOKKERT; KOK, 1997). Nestes casos, ele indica que o balanceamento seja feito para o modelo de maior volume e que as atividades dos demais sejam adaptadas a esta configuração básica.

As variações nos tempos de processamento ($\alpha_3 = t^{sto}$) também podem ser levadas em consideração, ao menos na forma mais simples apresentada por Lui et. al. (2005), que é a aplicação de um redutor ao tempo de ciclo. Esse, apesar de algumas implicações, é um artifício muito utilizado em problemas reais.

Além destas particularidades, um novo objetivo também pode ser explorado: a redução do tempo de ciclo ($\gamma = c$). Para isso, basta rodar o mesmo problema de forma iterativa, alterando o tempo de ciclo até achar o menor deles que possibilite operar com o número desejado de estações de trabalho.

4 TESTES NUMÉRICOS

Na literatura estudada, quando uma nova ferramenta de resolução dos problemas de ALB é desenvolvida ou aprimorada, ela é submetida a alguns problemas padrão para verificação dos resultados atingidos. Neste trabalho, serão utilizados 269 problemas padrão da literatura que foram reunidos no trabalho de Scholl (2007). Para cada um deles são conhecidos ou o resultado ótimo ou o limite inferior em número de estações utilizadas. A diferença percentual entre o número mínimo de estações descrito na literatura e o número de estações encontrado com a aplicação do Algoritmo Genético desenvolvido neste trabalho será a medida para quantificar o desvio em relação a solução ótima (ou ao limite inferior, quando tal solução é desconhecida).

Paralelamente, procedimento similar será utilizado em relação os resultados reportados nos trabalhos de outros autores que desenvolveram outras ferramentas para resolução dos problemas de ALB. Assim, além da distância em relação a solução ótima, é possível identificar se os resultados deste trabalho são comparáveis aos encontrados por outros autores.

Como se trata de um método estocástico de evolução populacional, cada aplicação pode gerar resultados distintos. Para evitar se basear em uma única simulação, que, por exemplo, poderia ser resultado de um mínimo local, cada um dos problemas foi submetido a 10 simulações distintas. Aquela que apresentou o menor número de estações foi utilizada para as comparações descritas acima.

Para possibilitar a resolução de todos estes problemas, os procedimentos descritos na seção 3 foram implementados em Delphi 7.0. A interface do programa desenvolvido é apresentada na figura 7.

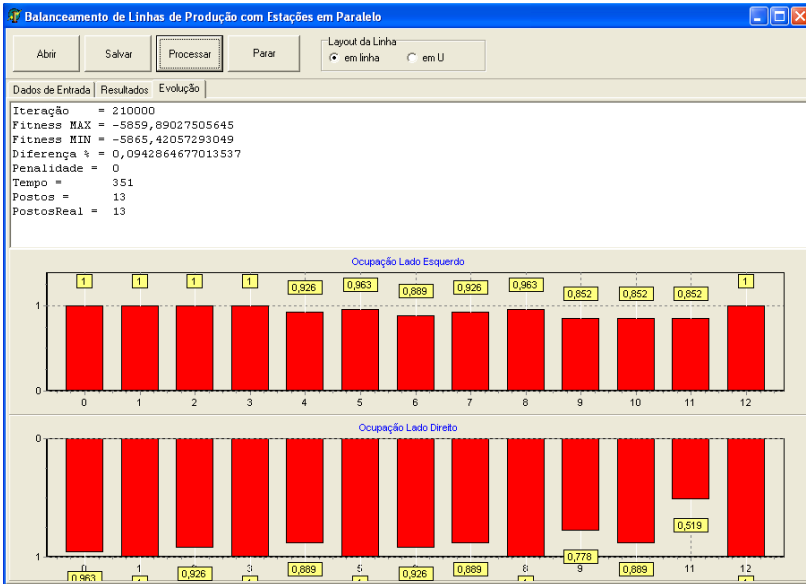


Figura 7 – Interface do programa desenvolvido

Fonte: Autor (2010).

No painel superior existem quatro botões e logo abaixo tem-se acesso a três abas.

O primeiro botão é utilizado para carregar um arquivo texto com os dados do problema. Este arquivo contém o tempo de ciclo e a lista de atividades para o balanceamento. Juntamente com cada atividade, o mesmo arquivo armazena as informações sobre tempo de execução, código do lado para o qual a atividade deve ser endereçada (1 para lado esquerdo, 2 para lado direito e 12 para qualquer lado) e a lista de atividades predecessoras. Quando trabalhando com restrições de alocação do tipo α_5 , elas são declaradas ao final do mesmo arquivo.

Ao abrir um problema qualquer ele pode ser visualizado na aba *dados de entrada*. Nesta aba o usuário pode modificar os dados à vontade bem como criar novos problemas. Caso deseje salvá-los ele utilizará o segundo botão.

O terceiro botão inicia o processo de resolução do problema que pode ser acompanhado pela aba *evolução*. Nela existem informações sobre a evolução da população e um gráfico de acompanhamento do indivíduo com melhor *fitness*. No gráfico são apontadas as taxas de ocupação de cada um dos postos de trabalho. Na parte superior é indicada a taxa de ocupação do lado esquerdo da linha, enquanto que na

parte inferior é apontada a taxa de ocupação do lado direito. Se, por algum motivo, o usuário quiser encerrar o processo de balanceamento e ver a melhor solução encontrada até aquele momento, o quarto botão deve ser acionado.

Ainda no painel superior, pode-se selecionar o layout para o qual o balanceamento será desenvolvido: *em linha* ou em *U*. É importante enfatizar que o programa não faz nenhuma restrição quanto ao balanceamento de linhas de dois lados com layout em *U*. Entretanto, esta combinação não faz sentido. Para tratar este tipo de layout, recomenda-se que, no momento da entrada de dados, todas as atividades sejam declaradas como pertencentes a um único lado. Isto é equivalente a trabalhar com layout em linha.

4.1 Testes com problemas padrão da literatura

Dentre os problemas padrão de SALBP-1 disponíveis na literatura, 25 (vinte e cinco) deles são os mais divulgados e utilizados para qualificar as novas ferramentas desenvolvidas. Estes, aplicados com distintos tempos de ciclo, geram um total de 269 (duzentos e sessenta e nove) problemas que são testados neste trabalho.

Esta mesma base de dados é utilizada para os problemas com layout em *U*. Já para os casos de linhas com dois lados, não existem problemas padrão na literatura. Para conseguir testar esta situação, foram criados problemas específicos a partir dos originais. Para isso, dentro de cada um dos problemas, foram adicionadas novas tarefas, todas endereçadas para o lado oposto, criando uma cópia da estrutura original. Ou seja, as novas atividades têm a mesma duração e contam com restrições de precedência análogas às originais. Assim, a mesma solução encontrada para o SALBP-1 padrão, em número de postos de trabalho, pode ser esperada para o problema transformado. Isto porque, se é possível distribuir as atividades em certo número de estações no problema original, esta mesma estrutura pode ser repetida para o lado oposto da linha ao receber as atividades análogas.

Além disso, como em situações desta natureza existem atividades que podem ser alocadas em qualquer um dos lados da linha, 20% das tarefas foram selecionadas aleatoriamente e cadastradas com esta característica.

Um esquema mostrando os passos para realização da construção dos modelos de dois lados é mostrado na figura 8.



Figura 8 – Esquema de montagem dos problemas de dois lados
Fonte: Autor (2010)

Partindo das atividades originais do problema (a) são criadas atividades análogas (b) com mesmas duração e restrições de precedência. O primeiro grupo é rotulado como pertencente ao lado esquerdo (LE) e o segundo ao lado direito (LD). Então, 20% das atividades, indicadas em amarelo, são selecionadas para poderem assumir qualquer lado no problema montado (c).

Cada um dos problemas foi resolvido 10 vezes e o melhor resultado encontrado para os casos de SALBP-1, layouts em U e linhas de dois lados podem ser observados nas figuras 9, 10 e 11, respectivamente.

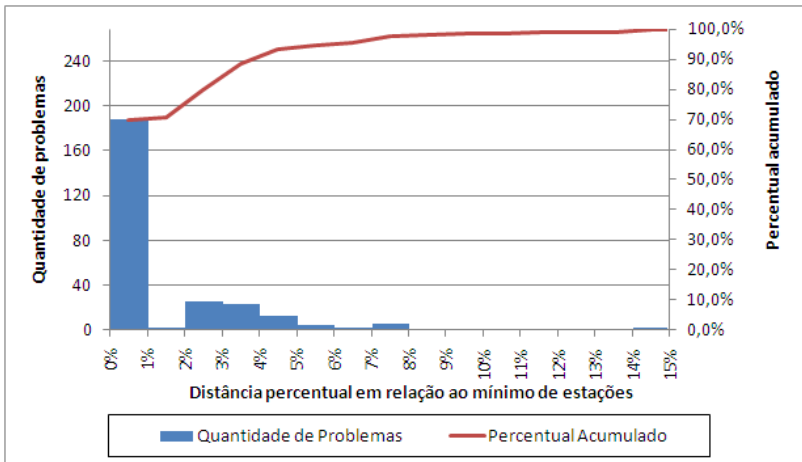


Figura 9 – Resultados consolidados dos 269 problemas da literatura para o caso de SALBP-1.

Nota: Dados extraídos a partir do Apêndice.

Fonte: Autor (2010)

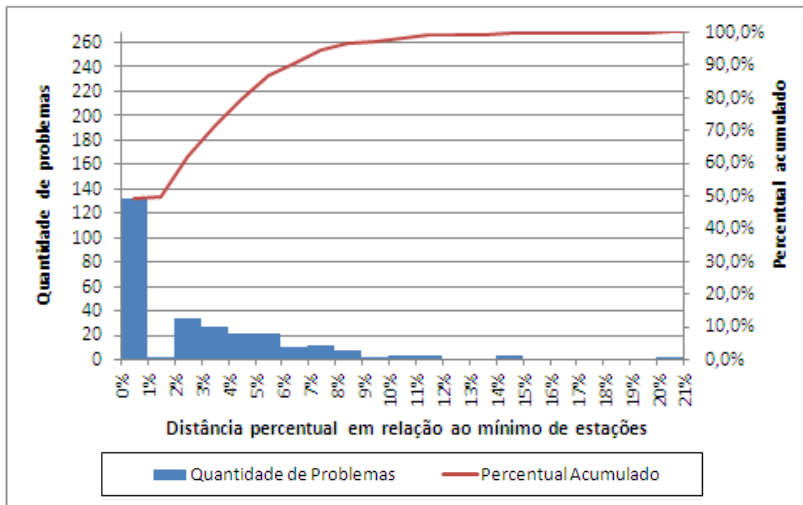


Figura 10 – Resultados consolidados dos 269 problemas da literatura para o caso de layout em U.

Nota: Dados extraídos a partir do Apêndice.

Fonte: Autor (2010).

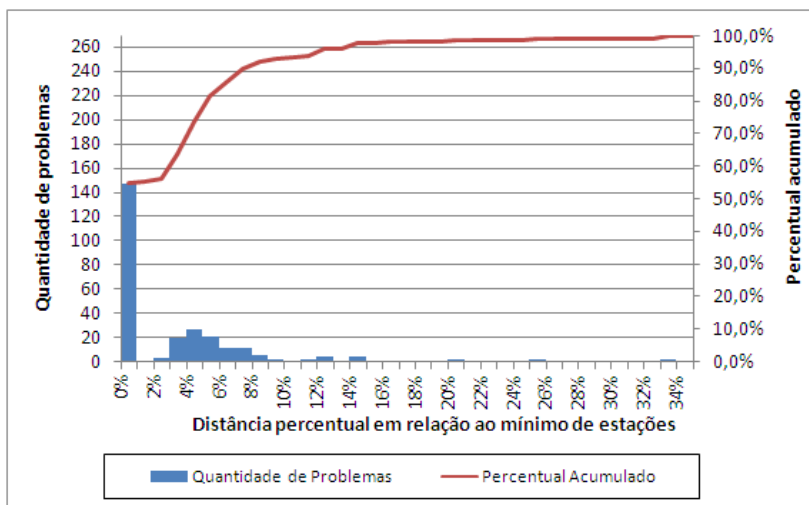


Figura 11 – Resultados consolidados dos 269 problemas da literatura para o caso de linhas com 2 lados.

Nota: Dados extraídos a partir do Apêndice.

Fonte: Autor (2010).

As figuras 9, 10 e 11 mostram, em representação por histogramas, a avaliação dos resultados obtidos em relação a solução ótima (ou ao limite inferior, quando tal solução não é conhecida). No eixo das abscissas está a distância em relação ao número mínimo de estações possíveis para cada problema. No eixo primário das ordenadas (à esquerda) é referenciado o número de problemas que tiveram resultados dentro de cada intervalo indicado. Na curva vermelha, a quantidade de atividades relacionada a cada intervalo é acumulada em termos percentuais e graduada pelo eixo secundário das ordenadas (à direita).

Nos testes realizados, 90% dos casos tiveram desvios abaixo de 4%, 7% e 8% para os casos de SALBP, layout em U e linhas de dois lados, respectivamente.

4.2 Análise dos resultados

Dos 269 problemas utilizados para qualificar a ferramenta desenvolvida neste trabalho, não se conhece a solução ótima de 1 deles para o caso de layout seriado e de 27 deles para o caso de linhas em U. No Apêndice, para estes casos, em vez da solução ótima, é indicado o intervalo entre o limite inferior do número de estações e a melhor solução encontrada até o momento. Os limites inferiores para o número de estações a serem utilizadas no balanceamento derivam daqueles utilizados para o problema de *Bin Packing*. Estes limites são apresentados por Scholl et al. (1997), e os três mais notáveis serão descritos a seguir.

O mais simples consiste na distribuição do tempo de todas as atividades em porções equivalentes ao tempo de ciclo. O número inteiro, igual ou imediatamente superior ao resultado encontrado, é tomado então como limite inferior, conforme indicado pela equação 62.

$$LB_1 = \left\lceil \frac{1}{c} \sum_{i \in V} t_i \right\rceil \quad (62)$$

A segundo forma de cálculo é iniciada ao selecionar todas as atividades cujo tempo de execução seja superior a metade do tempo de ciclo ($i \in V^* \mid t_i > c/2$). Estas atividades certamente ocuparão estações distintas. As demais atividades são então distribuídas ao tempo restante conforme indicado pela equação 63.

$$LB_2 = |V^*| + \max \left[0; \frac{\sum_{i \in (V-V^*)} t_i - (|V^*| \cdot c - \sum_{i \in V^*} t_i)}{c} \right] \quad (63)$$

A terceira forma de encontrar o limite inferior para o número de postos de trabalho faz a associação de pesos para cada uma das atividades conforme sua duração em relação ao tempo de ciclo. Primeiramente as tarefas são separadas em quatro grupos:

- a) V^a ($i \in V^a \mid t_i > (2c/3)$);
- b) V^b ($i \in V^b \mid (c/3) > t_i > (2c/3)$);
- c) V^c ($i \in V^b \mid t_i = (c/3)$);
- d) V^d ($i \in V^d \mid t_i = (2c/3)$).

Cada grupo recebe, respectivamente os pesos 1, 1/2, 1/3 e 2/3 que serão utilizados para realização da soma conforme indicado pela equação 64, que representa o limite inferior para o problema em estudo.

$$LB_3 = \left[|V^a| + \frac{1}{2} \cdot |V^b| + \frac{1}{3} \cdot |V^c| + \frac{2}{3} \cdot |V^d| \right] \quad (64)$$

Os limites inferiores apresentados nos resultados de Scholl (2007) para os problemas em estudo neste trabalho representam o maior limite encontrado pelas técnicas demonstradas por Scholl et al. (1997). Apesar de não haver confirmação de que estes sejam resultados possíveis, os dados apresentados são sempre referentes aos limites mínimos.

Chiang (1998), trabalhando com Busca Tabu, Chong et al. (2008), usando Algoritmos Genéticos e Baykasoglu (2006), com *Simulated Annealing*, resolveram, respectivamente, 61, 44 e 57 dos 269 problemas apresentados neste trabalho para o caso de SALBP-1. Ao comparar os resultados aqui obtidos com os reportados por eles, foi possível melhorar a solução de dois problemas em relação ao primeiro autor e de um problema em relação ao segundo. Todos os outros resultados foram idênticos. Já em relação ao terceiro autor não houve alteração nos resultados. Este, entretanto, aplica também os mesmo problemas para o caso de layout em U. Nesta condição, dos 57 problemas resolvidos, em um deles este trabalho obteve um resultado pior, indicando a necessidade de uso de uma estação adicional.

Os melhores resultados apresentados na literatura para o caso de SALBP-1 são compartilhados por Scholl (2007) em que, 260 problemas têm a solução ótima encontrada e nos demais, não há erros superiores a 4%.

4.3 Pontos Principais

Na seção 2.1.2, foi apresentado o comentário de Miltenburg (1998) em relação ao layout em “U”. Ele afirma que o número de estações utilizados nesta configuração por ser menor em relação ao *layout* seriado. Este resultado é esperado pois, ao trabalhar com esta forma de organização do fluxo da linha de produção, existem mais possibilidades de distribuição das atividades nos postos de trabalho e, conseqüentemente, maior a possibilidade de encontrar uma seqüência tal que evite os tempos ociosos na estações. No APÊNDICE, ao observar os resultados ótimos apresentados para os casos de “Layout em U”, o número mínimo de estações apresentado quando utilizando esta configuração são sempre menores ou iguais aos reportados no caso de “Layout em Linha”. Assim, a afirmação do autor pode ser confirmada.

Sobre os resultados encontrados com o Algoritmo Genético desenvolvido neste trabalho, vale ressaltar que, para aqueles com distância percentual elevada em relação ao ótimo, tratam-se de problemas com um pequeno número de postos de trabalho e, apesar da última estação ter ficado muito próxima de ser eliminada, a diferença percentual é feita em relação ao número inteiro de estações utilizadas. Em condições reais, entretanto, esta situação poderia ser flexibilizada pelo analista para, com um tempo de ciclo ligeiramente superior, reavaliar o balanceamento das atividades podendo eliminar este posto de trabalho quase ocioso, ou mesmo, reduzir o tempo de ciclo para preencher esta última estação e passar a utilizá-la de forma mais efetiva.

Na literatura estudada, não há uma comparação direta entre as soluções encontradas na alicação das diferentes meta-heurísticas para este tipo de problema. Neste trabalho, algumas comparações foram realizadas para verificar se não havia resultados discrepantes em relação a qualidade da solução, mas ainda não se tem informações suficientes para concluir com segurança se algum método é mais bem adaptado aos problemas de ALB.

5 CONCLUSÕES E RECOMENDAÇÕES

5.1 Conclusões

Neste trabalho foi apresentado o problema de ALB e suas extensões quanto a linhas de montagem com dois lados, layout em U e restrições de alocação de tarefas. O estudo foi motivado pois, apesar de mais de quatro décadas de evolução, os modelos disponíveis na literatura ainda se mostram pouco práticos para tratar a complexidade dos problemas reais.

Como indicado na seção 2.4, as três particularidades do problema de ALB abordadas pelo Algoritmo Genético desenvolvido neste trabalho são considerações fundamentais na aplicação em casos reais. Dessa forma, esta pesquisa vem de encontro com as necessidades da indústria que pode contar com mais esta ferramenta para analisar estes fatores de maneira simultânea.

Paralelamente, considerando a lacuna existente nos trabalhos acadêmicos no que se refere aos métodos para aplicação simultânea de múltiplas restrições dos problemas de balanceamento, este estudo também mostra sua contribuição ao avançar nesta discussão.

Após uma revisão sobre o uso de algoritmos genéticos, mostrando como são aplicados por outros autores na resolução de problemas desta natureza, com um modelo desenvolvido em Delphi 7.0, 269 problemas foram testados para qualificar a ferramenta construída.

Nos casos de SALBP-1, linhas com layout em “U” e linhas de 2 lados, respectivamente, em 188, 132 e 147 problemas tiveram a solução ótima encontrada. Nesta mesma seqüência, em 90% dos casos os desvios não superaram 4%, 7% e 8%. Em comparação com outros autores, que trabalharam com outras meta-heurísticas, os resultados foram muito similares exceto por um deles, *benchmarking* em problemas deste tipo, que reporta otimalidade em 260 SALBP-1.

Os resultados para os problemas com layout em U apresentam os piores resultados pois, para 27 deles, não se conhece a solução ótima, sendo então tomados para comparação os limite inferiores de cada um. Caso estes limites não se confirmem em trabalhos futuros, em mais 26 casos ter-se-á o resultado ótimo atingido pelo processo aqui desenvolvido.

Como as soluções dos testes realizados com esta ferramenta de fácil manipulação, baseada na estrutura do Algoritmo Genético, são

compatíveis às encontradas por outros autores, considera-se o objetivo geral deste trabalho atingido. Além disso, a revisão dos tipos de problema de ALB, a demonstração dos modelos matemáticos bem como a reformulação de alguns deles e a apresentação das heurísticas e meta-heurísticas mais utilizadas para resolvê-los, colaboram para o sucesso dos objetivos específicos.

Ainda sobre a qualidade dos resultados, Bartholdi (1993) afirma que, em problemas reais, os resultados ótimos são inapropriados pois, além de outras variações inerentes aos processos de montagem, os próprios dados sobre os tempos de execução das tarefas não são suficientemente precisos para se atingir este nível de detalhe. Assim, para a aplicação na indústria, a construção de uma ferramenta que possibilite fácil operação para testar de distintas condições, como foi o caso, configura um objetivo tão importante quanto a obtenção do resultado em si.

5.2 Recomendações

Neste trabalho foram abordadas apenas algumas restrições do ALB, entretanto, todas as demais apresentadas na seção 1.4 deste trabalho são importantes para poder estender este estudo aos casos reais. Recomenda-se, então, incluí-las em trabalhos futuros, mas operando de forma paralela e não isoladamente como geralmente são tratadas.

O tempo computacional não foi objetivo deste trabalho, porém, dependendo do problema em estudo este pode ser um fator complicador. A revisão das operações de evolução do Algoritmo Genético podem auxiliar na velocidade de convergência. Alguns autores, por exemplo, no momento da criação da população inicial, utilizam-se, além de métodos aleatórios, de algumas heurísticas (ver 2.3.1) na tentativa de conseguir indivíduos mais aptos desde as primeiras iterações. Outro exemplo desta situação pode ser vista em alguns processos de *crossover* que trabalham com heurísticas específicas para orientar melhorias no desenvolvimento da população como é o caso do modelo *Heuristic Structural Crossover* (ver 3.2.4). Entretanto, esta busca orientada através de heurísticas mais elaboradas pode guiar as soluções para mínimos locais. Assim, sua aplicação deve ser avaliada criteriosamente.

Para o caso dos problemas que almejam a redução do número de estações ($\gamma = m$), não há boas informações sobre o comportamento das diversas funções de *fitness* divulgadas na literatura. Como esta

representa a chave para convergência dos problemas, suas diversas formulações deveriam ser melhor estudadas a fim de medir a qualidade de sua aplicação.

Dos problemas testados para o caso de layout em formato de “U”, 28 deles ainda não tem sua solução ótima conhecida. Encontrá-las é importante pois, através delas os novos métodos de resolução dos problemas de ALB são qualificados.

Referências

AGUIAR, G. F.; PEINADO, J.; GRAEML, A. R. Simulação de arranjos físicos por produto e balanceamento de linha de produção: O estudo de um caso real no ensino para estudantes de engenharia. **XXXV Congresso Brasileiro de Educação em Engenharia**. Universidade Positivo. Curitiba. 2007.

AMEN, M. Heuristic methods for cost-oriented assembly line balancing: A comparison on solution quality and computing time. **International Journal of Production Economics**. V. 69 p. 255-264. 2001.

BAYKASOGLU, A. Multi-rule multi-objective simulated annealing algorithm for straight and U type assembly line balancing problems. **Journal of Intelligent Manufacturing**, v. 17, p. 217-232. 2006.

BARTHOLDI, J. J. Balancing two-side assembly line: a case study. **International Journal of Production Research**, v. 31, n. 10, p. 2447-2461. 1993.

BAUTISTA, J.; PEREIRA, J. Ant algorithms for a time and space constrained assembly line balancing problem. **European Journal of Operational Research**, v. 177, p. 2016-2032. 2007.

BAUTISTA, J.; PEREIRA, J. A dynamic programming based heuristic for the assembly line balancing problem. **European Journal of Operational Research**, v. 194, p. 787-794. 2009.

BECKER, C. e SCHOLL, A. **A survey on problems and methods in generalized assembly line balancing**. Arbeits und Diskussionspapiere der Wirtschaftswissenschaftlichen Fakultät der Friedrich-Schiller-Universität Jena. 2003.

BENZER, R., GÖKÇEN, H.; ÇETINYOKUS, T.; ÇERÇIOĞLU, H. A Network Model for Parallel Line Balancing Problem. **Mathematical Problems in Engineering**, v. 2007. 2007.

BITTENCOURT, G. **Inteligência Computacional**. Departamento de Automação e Sistemas. Universidade Federal de Santa Catarina.

Disponível em: <<http://www.das.ufsc.br/gia/softcomp/>>. Acesso em: 1º de dezembro de 2009.

BOYSEN, N; FLIEDNER, M. A versatile algorithm for assembly line balancing. **European Journal of Operational Research**, v. 184, p. 39–56. 2008.

BOYSEN, N; FLIEDNER, M.; SCHOLL, A. **Assembly line problem. Which model use when?** Arbeits und Diskussionspapiere der Wirtschaftswissenschaftlichen Fakultät der Friedrich-Schiller-Universität Jena. 2006.

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

BUKCHIN, Y. RABINOWITCH, I. A branch-and-bound based solution approach for the mixed-model assembly line-balancing problem for minimizing stations and task duplication costs. **European Journal of Operational Research**, v. 174, p. 492-508. 2006.

CAPACHO, L.; PASTOR, P.; DOLGUI, A.; GUSCHINSKAYA, O. An evaluation of constructive heuristic methods for solving the alternative subgraphs assembly line balancing problem. **Journal of Heuristics**, v. 15, p. 109-132. 2009.

CAO, Z.; MA, S. Balancing and Sequencing Optimization of the Mixed Model Assembly Lines. **International Symposium on Computer Science and Computational Technology**. 2008.

CARNAHAN, B. J; NORMAN, B. A.; REDFERN, M. S. Incorporating physical demands criteria into assembly line balancing. **IIE Transactions**, v. 33, p. 875-887. 2001.

CARRAVILLA, M. A. **Layouts: balanceamento de linhas**. Publicação didática da Faculdade de Engenharia da Universidade do Porto. Porto, Potugual. 1998. Disponível em: <<http://hdl.handle.net/10216/574>>. Acesso em: 13 de junho de 2009.

CHAN, K. C. C.; HUI, P. C. L.; YEUNG, K. W.; NG, F. S. F. Handling the assembly line balancing problem in the clothing industry using a

genetic algorithm. **International Journal of Clothing Science and Technology**, v. 10, n. 1, p. 21-37. 1998.

CHIANG, W. C. The application of a tabu search metaheuristic to the assembly line balancing problem. **Annals of Operations Research**, v. 77, p. 209-227. 1998.

CHIANG, W. C.; URBAN, T. L. The stochastic U-line balancing problem: A heuristic procedure. **European Journal of Operational Research**, v. 175, p. 1767-1781. 2006.

CHONG, K. E; OMAR, M. K.; BAKAR, N. A. Solving Assembly Line Balancing Problembusing Genetic Algorithm with Heuristics-Treated Initial Population. **Proceedings of the World Congress on Engineering**, v. 2. 2008.

COHEN, Y; VITNER, G.; SARIN, S. C. Optimal allocation of work in assembly lines for lots with homogenous learning. **European Journal of Operational Research**. v. 168, p. 922-931. 2006.

CORRÊA, H. L. **Teoria Geral da Administração: Abordagem Histórica da Gestão de Produção e Operações**. Ed. Atlas. São Paulo, SP. 2003.

DIMITRIADIS, S.G. Assembly line balancing and group working: A heuristic procedure for workers' groups operating on the same product and workstation. **Computers & Operations Research**, v. 33, p. 2757-2774. 2006.

DORIGO, M. **About Ant Colony Optimization**. Disponível em: <<http://www.aco-metaheuristic.org/about.html>>. Acesso em: 23 de julho de 2009.

ELI WHITNEY MUSEUM AND WORKSHOP. **Eli Whitney: The Inventor**. Disponível em: <<http://www.eliwhitney.org/new/museum/about-eli-whitney/inventor>>. Acesso em: 15 de agosto de 2009.

FARNES, V. C. F; PEREIRA, N. A. Balanceamento de linha de montagem com o uso de heurística e simulação: estudo de caso na linha branca. **Gestão da Produção, Operações e Sistemas**. Ano 2, v. 2, p. 125-136. 2006.

FERNANDES, F. C. F.; GODINHO Filho, M.; CUTIGI, R.A.; GUIGUET, A. M. O uso da programação inteira 0-1 para o balanceamento de linhas de montagem: modelagem, estudos de caso e avaliação. **Produção**. V. 18, n. 2, p. 210-221. 2008.

FOKKERT, J. I. Z; KOK, T. G. The mixed and multi model line balancing problem: a comparison. **European Journal of Operational Research**. v. 100, p. 399-412. 1997.

GAMBERINI, R.; GRASSI, A. RIMINI, B. A new multi-objective heuristic algorithm for solving the stochastic assembly line re-balancing problem. **International Journal of Production Economics**, v. 102, p. 226-243. 2006.

GLOVER, F. Tabu search – Part I. Operations Research Society of America Journal on Computing, v. 1, n. 3. 1989.

GLOVER, F. Tabu search – Part II. Operations Research Society of America Journal on Computing, v. 2, n. 1. 1990.

GRAHAM, R. L; LAWLER, E. L; LENSTRA, J. K; RINNOOY KAN, A. H. G. Optimization and approximation in deterministic sequencing and scheduling: a survey. **Annals of Discrete Mathematics**, v. 5, p. 287-326. 1979.

GRZECHCA, W. **Estimation of Time and Cost Oriented Assembly Line Balancing Problem**. 19th International Conference on Systems Engineering. 2008.

HÖRNER, D. **Resolução do problema das p-medianas não capacitado: Comparação de algumas técnicas heurísticas**. Dissertação de Mestrado. Universidade Federal de Santa Catarina. Florianópolis. 2009.

KARA, Y.; OZCAN, U.; PEKER, A. Balancing and sequencing mixed-model just-in-time U-lines with multiple objectives. **Applied Mathematics and Computation**. V. 184, p. 566-588. 2007.

KIM, Y. K.; KIM, Y. J.; KIM, Y. A heuristic-based genetic algorithm for workload smoothing in assembly lines. **Computers Operational Research**, v. 25, n.2, p. 99-111. 1998.

KIM, Y. K.; KIM, Y.; KIM, Y. J. Two-sided assembly line balancing: a genetic algorithm approach. **Production Planning & Control**, v. 11, n. 1, p. 44-53. 2000.

KIM, Y. K.; SONG, W. S.; KIM J. H. A mathematical model and a genetic algorithm for two-sided assembly line balancing. **Computers & Operations Research**, v. 36, p. 853-865. 2009.

KIMMS, A. Minimal investment budgets for flow line configuration. **IIE Transactions**, v. 32, p. 287-298. 2000.

KIRKPATRICK, S; GELATT, C. D.; VECCHI, M. P. Optimization by Simulated Annealing. **Science**, v. 220, n. 4598. 1983.

KLAMPFL, E; GUSIKHIN, O.; ROSSI, G. Optimization of workcell layouts in a mixed-model assembly line environment. **International Journal of Flexible Manufacturing Systems**, v. 17, p. 277-299. 2006.

KRIENGGORAKOT, N.; PIANTHONG, N. The U-line Assembly Line Balancing Problem. **KKU Engineering Journal**, v. 34, n. 3, p. 267-274. 2007.

McMULLEN, P. R.; FRAZIER, G. V. Using simulated annealing to solve a multiobjective assembly line balancing problem with parallel workstations. **International Journal of Production Research**, v. 36, n. 10, 2717-2741. 1998.

MENG, Q. C.; FENG, T. J.; CHEN, Z.; ZHOU, C. J.; BO, J. H. Genetic algorithms encoding study and a sufficient convergence condition of GAs. **IEEE International Conference**, v. 1, p. 649-652. 1999.

MEYER, H. Supply chain planning in the German automotive industry. **OR Spectrum**, v. 26, p. 447-470. 2004.

MILTENBURG, J. Balancing U-lines in a multiple U-line facility. **European Journal of Operational Research**, v. 109, p. 1-23. 1998.

- LAPIERRE, S. D.; RUIZ, A.; SORIANO, P. Balancing assembly lines with tabu search. **European Journal of Operational Research**, v. 168, p. 826-837. 2006.
- LIU, S.B.; ONG, H.L.; HUANG, H.C. A bidirectional heuristic for stochastic assembly line balancing Type II problem. **The International Journal of Advanced Manufacturing Technology**, v. 25, p. 71-77. 2005.
- PONNAMBALAM, S. G.; ARAVINDAN, P.; NAIDU, G. M. A Comparative Evaluation of Assembly Line Balancing Heuristics. **The International Journal of Advanced Manufacturing Technology**, v. 15, p. 577-586. 1999.
- RIVERA, W. Scalable Parallel Genetic Algorithms. **Artificial Intelligence Review**, v. 16, p. 153-168. 2001.
- SABUNCUOGLU, I.; EREL, E.; TAYNER, M. Assembly line balancing using genetic algorithms. **Journal of Intelligent Manufacturing**, v. 11, p. 295-310. 2000.
- SHAFER, S. M.; NEMBHARD, D. A.; UZUMERI, M. V. The Effects of Worker Learning, Forgetting, and Heterogeneity on Assembly Line Productivity. **Management Science**, v. 47, n. 12, p. 1639-1653. 2001.
- SCHOLL, A. **Data sets for SALBP**. 2007. Disponível em: <<http://www.assembly-line-balancing.de/index.php?content=classview&content2=classview&content3=classviewdlfree&content4=classview&classID=28&type=dl>>. Acesso em: 22 agosto de 2009.
- SCHOLL, A; BECKER, C. State-of-the-art exact and heuristic solution procedures for simple assembly line balancing. **European Journal of Operational Research**, v. 168, p. 666-693. 2006.
- SCHOLL, A; BOYSEN, N. Designing parallel assembly lines with split workplaces: Model and optimization procedure. **International Journal of Production Economics**. 2009.
- SCHOLL, A; FLIEDNER, M.; BOYSEN, N; ABSALOM: Balancing assembly lines with assignment restrictions. **European Journal of Operational Research**. 2009.

SCHOLL, A.; Voß, S. Simple Assembly Line Balancing-Heuristic Approaches. *Journal of Heuristics*, v. 2, p. 217-244. 1996.

SMITH, A., **An Inquiry into the Nature and Causes of the Wealth of Nations**, 1776. Trecho da obra disponível em <<http://www.adamsmith.org>> acessado em 24 de setembro de 2009.

TASAN, O. S.; TUNALI, S. A review of the current applications of genetic algorithms in assembly line balancing. **Journal of Intelligent Manufacturing**, v. 19, p. 49-69. 2008.

The Eli Whitney Museum and Workshop. **Eli Whitney**. Disponível em: <<http://www.eliwhitney.org/new/museum/eli-whitney>>. Acesso em: 22 de setembro de 2009.

TOKSARI, M. D.; ISLEYEN, S. K.; GÜNER, E.; BAYKOÇ, Ö, F. Simple and U-type assembly line balancing problems with a learning effect. **Applied Mathematical Modelling**, v. 32, p. 2954-2961. 2008.

APÊNDICE

Resultados dos testes realizados com 269 problemas de ALB disponíveis na literatura.

Problema	V	c	Layout em Linha		Layout em U		Linhas com 2 lados	
			m*	m ^e	m*	m ^e	m*	m ^e
Arcus1	83	3786	21	21	21	21	21	21
		3985	20	20	20	20	20	21
		4206	19	19	19	19	19	19
		4454	18	18	18	18	18	18
		4732	17	17	17	17	17	17
		5048	16	16	16	16	16	16
		5408	15	15	15	15	15	15
		5824	14	14	14	14	14	14
		5853	14	14	13	14	14	14
		6309	13	13	13	13	13	13
		6842	12	12	12	12	12	12
		6883	12	12	12	12	12	12
		7571	11	11	11	11	11	11
		8412	10	10	10	10	10	10
8898	9	9	9	9	9	9		
10816	8	8	7	8	8	8		
Arcus2	111	5755	27	27	27	27	27	28
		5785	27	27	[26,27]	27	27	28
		6016	26	26	[25,26]	26	26	27
		6267	25	25	[24,25]	25	25	26
		6540	24	24	[23,24]	24	24	25
		6837	23	23	[22,23]	23	23	24
		7162	22	22	[21,22]	22	22	22
		7520	21	21	[20,21]	21	21	21
		7916	20	20	[19,20]	20	20	20
		8356	19	19	[18,19]	19	19	19
		8847	18	18	[17,18]	18	18	18
		9400	17	17	[16,17]	17	17	17
		10027	16	16	[15,16]	16	16	16
		10743	15	15	[14,15]	15	15	15
11378	14	14	14	14	14	14		
11570	13	14	13	14	13	14		
17067	9	9	9	9	9	9		
Buxey	29	27	13	13	13	13	13	13
		30	12	12	11	11	12	12
		33	11	11	10	10	11	11
		36	10	10	9	10	10	10
		41	8	8	8	8	8	9
		47	7	7	7	7	7	8
		54	7	7	6	6	7	7

continua

Problema	V	c	Layout em Linha		Layout em U		Linhas com 2 lados	
			m*	m ^e	m*	m ^e	m*	m ^e
Barthol2	148	84	51	52	51	52	51	54
		85	50	52	50	52	50	53
		87	49	50	49	50	49	51
		89	48	49	48	49	48	50
		91	47	48	47	48	47	49
		93	46	47	46	47	46	48
		95	45	46	45	46	45	47
		97	44	45	44	45	44	46
		99	43	44	43	44	43	45
		101	42	43	42	43	42	44
		104	41	42	41	42	41	43
		106	40	42	40	41	40	42
		109	39	40	39	40	39	41
		112	38	39	38	39	38	40
		115	37	39	37	38	37	39
		118	36	37	36	37	36	38
		121	35	36	35	36	35	37
		125	34	35	34	35	34	36
129	33	34	33	34	33	34		
133	32	33	32	33	32	33		
137	31	32	31	32	31	32		
142	30	31	30	31	30	31		
146	29	30	29	30	29	30		
152	28	29	28	29	28	29		
157	27	28	27	28	27	28		
163	26	27	26	27	26	27		
170	25	26	25	26	25	26		
Barthold	148	403	14	15	14	15	14	15
		434	13	14	13	14	13	14
		470	12	13	12	13	12	13
		513	11	11	11	11	11	12
		564	10	11	10	11	10	11
		626	9	10	9	10	9	10
		705	8	8	8	8	8	9
805	7	8	7	7	7	8		
Bowman	8	20	5	5	4	4	5	5
Gunther	35	41	14	14	12	13	14	14
		44	12	12	12	12	12	12
		49	11	11	10	11	11	11
		54	9	9	9	9	9	9
		61	9	9	8	8	9	9
		69	8	8	7	8	8	8
81	7	7	6	6	7	7		

continua

Problema	V	c	Layout em Linha		Layout em U		Linhas com 2 lados	
			m*	m ^c	m*	m ^c	m*	m ^c
Hahn	53	2004	8	8	8	8	8	8
		2338	7	7	7	7	7	8
		2806	6	6	5	6	6	6
		3507	5	5	5	5	5	5
		4676	4	4	3	3	4	4
Heskiaoff	28	138	8	8	8	8	8	8
		205	5	5	5	5	5	6
		216	5	5	5	5	5	5
		256	4	4	4	4	4	5
		324	4	4	4	4	4	4
342	3	3	3	3	3	4		
Jackson	11	7	8	8	7	7	8	8
		9	6	6	6	6	6	6
		10	5	5	5	5	5	5
		13	4	4	4	4	4	4
		14	4	4	4	4	4	4
21	3	3	3	3	3	3		
Jaeschke	9	6	8	8	8	8	8	8
		7	7	7	7	7	7	7
		8	6	6	6	6	6	6
		10	4	4	4	4	4	4
18	3	3	3	3	3	3		
Kilbridge	45	56	10	10	10	10	10	10
		57	10	10	10	10	10	10
		62	9	9	9	9	9	9
		69	8	8	8	8	8	9
		79	7	7	7	7	7	8
		92	6	6	6	6	6	7
		110	6	6	6	6	6	6
		111	5	5	5	5	5	5
		138	4	4	4	4	4	5
184	3	3	3	3	3	4		
Lutzl	32	1414	11	11	11	11	11	11
		1572	10	10	10	10	10	10
		1768	9	9	9	9	9	9
		2020	8	8	8	8	8	8
		2357	7	7	7	7	7	7
		2828	6	6	6	6	6	6

continua

Problema	V	c	Layout em Linha		Layout em U		Linhas com 2 lados	
			m*	m ^c	m*	m ^c	m*	m ^c
Lutz2	89	11	49	49	45	45	49	50
		12	44	46	41	42	44	46
		13	40	40	38	38	40	41
		14	37	39	35	36	37	39
		15	34	34	33	34	34	34
		16	31	31	31	31	31	33
		17	29	30	29	29	29	30
		18	28	28	27	28	28	29
		19	26	27	26	26	26	27
		20	25	26	25	25	25	26
		21	24	24	24	24	24	
Lutz3	89	75	23	24	22	23	23	24
		79	22	22	21	22	22	22
		83	21	21	20	21	21	22
		87	20	20	19	20	20	21
		92	19	19	18	19	19	19
		97	18	18	17	18	18	18
		103	17	17	16	17	17	17
		110	15	16	15	16	15	16
		118	14	15	14	15	14	15
		127	14	14	13	13	14	14
		137	13	13	12	13	13	
		150	12	12	11	11	12	
Mansoor	11	48	4	4	4	4	4	4
		62	3	3	3	3	3	3
		94	2	2	2	2	2	2
Mertens	7	6	6	6	6	6	6	6
		7	5	5	5	5	5	5
		8	5	5	5	5	5	5
		10	3	3	3	3	3	3
		15	2	2	2	2	2	2
		18	2	2	2	2	2	
Mitchell	21	14	8	8	8	8	8	9
		15	8	8	8	8	8	8
		21	5	5	5	5	5	5
		26	5	5	5	5	5	5
		35	3	3	3	3	3	3
		39	3	3	3	3	3	3
Roszieg	25	14	10	10	9	9	10	10
		16	8	8	8	8	8	9
		18	8	8	7	7	8	8
		21	6	6	6	6	6	6
		25	6	6	5	5	6	6
		32	4	4	4	4	4	4

continua

Problema	V	c	Layout em Linha		Layout em U		Linhas com 2 lados	
			m*	m ^e	m*	m ^e	m*	m ^e
Mukherje	94	176	25	25	[24,25]	25	25	25
		183	24	24	23	24	24	24
		192	23	23	22	23	23	24
		201	22	22	21	22	22	23
		211	21	21	20	21	21	21
		222	20	20	19	20	20	20
		234	19	19	18	19	19	19
		248	18	18	17	18	18	18
		263	17	17	16	17	17	17
		281	16	16	15	16	16	16
		301	15	15	14	15	15	15
		324	14	14	13	14	14	14
		351	13	13	12	13	13	13
Scholl	297	1394	50	52	50	52	50	54
		1422	50	51	[49,50]	51	50	53
		1452	48	50	48	49	48	52
		1483	47	49	47	48	47	51
		1515	46	48	46	47	46	50
		1548	46	47	45	46	46	48
		1584	44	46	44	45	44	47
		1620	44	45	43	44	44	47
		1659	42	44	42	43	42	45
		1699	42	42	41	42	42	44
		1742	40	41	40	41	40	42
		1787	39	40	39	40	39	41
		1834	38	39	38	39	38	40
		1883	37	38	37	38	37	49
		1935	36	37	36	37	36	38
		1991	35	36	35	36	35	37
		2049	34	35	34	35	34	36
		2111	33	34	33	34	33	35
		2177	32	33	32	33	32	34
		2247	31	32	31	32	31	33
2322	30	31	30	31	30	32		
2402	29	30	29	30	29	31		
2488	28	29	28	29	28	29		
2580	27	28	27	28	27	29		
2680	26	27	26	27	26	28		
2787	25	26	25	26	25	26		

continua

Problema	V	c	Layout em Linha		Layout em U		Linhas com 2 lados	
			m*	m ^c	m*	m ^c	m*	m ^c
Tonge	70	160	23	23	[22,23]	23	23	23
		168	22	22	21	22	22	22
		176	21	21	[20,21]	21	21	22
		185	20	20	19	20	20	20
		195	19	19	18	19	19	19
		207	18	18	17	18	18	18
		220	17	17	16	17	17	17
		234	16	16	15	16	16	16
		251	14	15	14	15	14	15
		270	14	14	13	14	14	14
		293	13	13	12	13	13	13
		320	11	12	11	12	11	12
		364	10	10	10	10	10	10
		410	9	9	9	9	9	9
468	8	8	8	8	8	8		
527	7	7	7	7	7	7		
Sawyer	30	25	14	14	14	14	14	14
		27	13	13	13	13	13	14
		30	12	12	11	11	12	12
		33	11	11	10	10	11	11
		36	10	10	9	10	10	10
		41	8	8	8	8	8	8
		47	7	8	7	7	7	8
		54	7	7	6	6	7	7
75	5	5	5	5	5	5		
Warnecke	58	54	31	32	[30,31]	31	31	32
		56	29	30	29	29	29	30
		58	29	29	28	29	29	30
		60	27	28	27	28	27	29
		62	27	28	[26,27]	27	27	28
		65	25	26	[24,25]	26	25	27
		68	24	24	[23,24]	24	24	25
		71	23	23	[22,23]	23	23	24
		74	22	22	[21,22]	22	22	23
		78	21	21	20	21	21	22
		82	20	20	[19,20]	20	20	21
		86	19	19	18	19	19	20
		92	17	18	17	18	17	19
		97	17	17	16	17	17	17
104	15	16	15	16	15	16		
111	14	15	14	15	14	15		

continua

			conclusão					
Problema	V	c	Layout em Linha		Layout em U		Linhas com 2 lados	
			m*	m ^e	m*	m ^e	m*	m ^e
Wee-mag	75	28	63	63	63	63	63	63
		29	63	63	63	63	63	63
		30	62	63	62	62	62	63
		31	62	62	62	62	62	62
		32	61	61	61	61	61	61
		33	61	61	61	61	61	61
		34	61	61	61	61	61	61
		35	60	60	60	60	60	60
		36	60	60	60	60	60	60
		37	60	60	60	60	60	60
		38	60	60	60	60	60	60
		39	60	60	60	60	60	60
		40	60	60	60	60	60	60
		41	59	59	59	59	59	59
		42	55	55	55	55	55	55
		43	50	50	50	50	50	50
		45	38	39	38	38	38	40
		46	34	34	34	34	34	35
		47	[32,33]	33	[32,33]	33	[32,33]	33
		49	32	32	[31,32]	32	32	32
50	32	32	[31,32]	32	32	32		
52	31	31	31	31	31	31		
54	31	31	31	31	31	31		
56	30	30	30	30	30	30		

Nota: m* = Resultado ótimo ou intervalo entre limite inferior e melhor resultado conhecido.

m^e = Resultado encontrado com o método desenvolvido neste trabalho.

Fonte: SCHOLL (2007).