Um modelo matemático para o problema de balanceamento de linhas de montagem com divisão de tarefas

Carlos Alexandre X. Silva
Leslie R. Foulds
Humberto J. Longo
Instituto de Informática – INF
Universidade Federal de Goiás – UFG
Goiânia – GO, Brasil
{carlosalexandre,lesfoulds,longo}@inf.ufg.br

RESUMO

Em uma versão simples do problema de balanceamento de linhas de montagem, *the Simple Assembly Line Balancing Problem* (*SALBP*), tarefas são atribuídas a estações de produção ao longo de uma linha de montagem, com um tempo de ciclo fixo, para minimizar o número de estações. Supõe-se que o trabalho total necessário para fabricação de cada unidade de um produto foi particionado em tarefas economicamente indivisíveis. Na prática pode ser que a redução do número de estações seja possível ao se dividir tarefas particulares de forma limitada, mesmo se a divisão induzir custos adicionais. A permissão da divisão de tarefas leva a um novo problema de balanceamento de linha de montagem, o *TDALBP* (*Task Division Assembly Line Balancing Problem*). Neste trabalho é introduzido um modelo matemático para o TDALBP e apresentados resultados computacionais promissores para a adaptação de algumas instâncias ALBP clássicas da literatura de pesquisa. Na verdade, 18 casos foram resolvidos como problemas TDALBP, gerando planos de balanceamento de linha de montagem com menor número de estações. Os resultados mostram o potencial do TDALBP para a melhora significativa do desempenho de linhas de montagens.

PALAVRAS CHAVE. Sistemas de fabricação, Balanceamento de linhas de montagem, Divisão de tarefas.

OC – Otimização Combinatória, PM – Programação Matemática, IND – PO na Indústria.

ABSTRACT

In one version of the Simple Assembly Line Balancing Problem (SALBP) tasks are assigned to stations along an assembly line with a fixed cycle time in order to minimise the required number of stations. It is assumed that the total work needed for each product unit has been partitioned into economically indivisible tasks. In practice, it may be that the minimal number of stations can be reduced when it is possible to further divide particular tasks in limited ways even with additional time penalty costs. Allowing task division leads to a new assembly line balancing problem, TDALBP (Task Division Assembly Line Balancing Problem) and a solution procedure for it. This work introduces a mathematical model for the TDALBP and presents promising computational results for the adaptation of some classical ALBP instances from the research literature. In fact, 18 instances were solved as TDALBP problems, generating assembly line balancing plans with a smaller number of stations. The results demonstrate that the TDALBP has the potential to significantly improve assembly line performance.

KEYWORDS. Manufacturing systems, Assembly line balancing, Task division.

OC – Combinatorial Optimization, PM – Mathematical Programming, IND – OR in Industry.

1. Introdução

O balanceamento eficaz de uma linha de montagem é importante para a melhoria da produtividade e para a redução de custos de produção. O problema SALBP (*Simple Assembly Line Balancing Problem*) envolve atribuir as tarefas necessárias para produzir cada unidade de um produto entre estações de trabalho ao longo de uma linha de produção, a fim de otimizar alguma medida de desempenho do sistema. No balanceamento de linha de montagem, tradicionalmente, supõe-se que o trabalho total necessário para cada unidade de produto foi particionado em tarefas economicamente indivisíveis, de modo que uma maior divisão cria subtarefas e custos desnecessários. Assim, cada tarefa requerida é indivisível, no sentindo de que ela deve ser realizada em uma única estação. No entanto, mudanças tecnológicas, planejamento de produtos, processos de fabricação e projeto de trabalhos podem criar oportunidades de mudanças significativas, de modo que o conteúdo total de trabalho possa ser dividido em tarefas menores. Por exemplo, quando existe um objetivo orientado ao tempo, tal como a minimização do número de estações para um determinado tempo de ciclo, pode ser que seja possível a redução do número mínimo de estações ao se dividir as tarefas de forma limitada, mesmo se a divisão resultar em custos adicionais.

Quando algumas das tarefas possuem o tempo de processamento relativamente longo e envolvem operações simples, parte dessa tarefa pode ser proveitosamente adiada para outra estação. Por exemplo, suponha que uma tarefa consista no aperto de parafusos e possua um dos tempos de processamento mais longos dentre todas as tarefas. Suponha também que apenas dois parafusos sejam suficientes para fixar com segurança duas peças em seus lugares. Assim, o restante dos parafusos podem ser apertados em uma estação posterior, sem prejuízo para a conclusão de outras tarefas intermediárias. Neste caso, o número mínimo de estações requerido pode ser reduzido, mesmo se a divisão induzir uma penalidade adicional de tempo.

A inclusão da possibilidade de tal divisão de tarefas nos leva a um novo problema de balanceamento de linha de montagem, o qual é denotado por TDALBP (*Task Division Assembly Line Balancing Problem*). O TDALBP foi introduzido por Grzechca e Foulds [2015], quem resolveu com êxito um estudo de caso na linha de montagem em uma fábrica real polonesa onde foi possível dividir determinadas tarefas entre mais de uma estação. Este trabalho baseia-se nesse artigo. A proposta do presente trabalho é preencher essa lacuna na literatura, discutindo uma versão rigorosa do problema, formulando um modelo inteiro binário, propondo métodos, soluções e relatando a experiência computacional.

2. Balanceamento de linhas de montagem

A linha de montagem foi pensada para ser um meio eficiente e altamente produtivo de fabricação de produtos. O problema do balanceamento de linhas de montagem envolve atribuir às estações, ao longo de uma linha de produção, as tarefas necessárias para produzir cada unidade de um produto, a fim de otimizar o desempenho do sistema sobre alguma medida dada. O conceito de produção em linha de montagem para fabricação em massa de automóveis padronizados foi introduzido pela primeira vez por Ransom Eli Olds em 1901, em seu Olds Motor Vehicle Company. Em 1913, Henry Ford tinha melhorado significativamente o conceito de linha de montagem, a quem é atribuída a construção da primeira linha de montagem de carros. Nos dias de hoje, ela é utilizado para produzir grandes volumes de uma ampla variedade de produtos montados padronizados ou personalizados, incluindo não só os veículos, mas também muitos aparelhos domésticos e produtos eletrônicos. A ideia do balanceamento de linha foi introduzida primeiramente por Bryton [1954], como um desafio de planejamento de médio prazo. Um estudo quantitativo publicado por Salveson [1955] formulou o problema como um programa linear e foi a primeiro a introduzir o termo *the Assembly Line Balancing Problem*. Tal como observado por Scholl [1998] e muitos outros, a gestão das linhas de montagem é um desafio a curto-médio prazo significativo no planejamento de produção.

Seguindo a notação e terminologia usada por Scholl et al. [2009], a linha de montagem consiste de um conjunto de m *estações de trabalho* organizadas de forma linear, ligadas por um

dispositivo mecânico de manuseamento de materiais. O movimento básico do material através de uma linha de montagem começa com as peças necessárias para iniciar a criação de cada unidade de um único produto, alimentando consecutivamente a primeira estação. O trabalho necessário para completar cada unidade do produto é dividido em um conjunto de *tarefas* $V = \{1, \ldots, n\}$, onde cada uma delas deve ser atribuída a uma estação. Uma vez que um produto parcialmente montado entra em uma estação ativa, um conjunto de tarefas (não vazio) atribuído é executado sobre ele, o qual, após isto, segue para a próxima estação ativa.

O tempo requerido para completar cada tarefa $j \in V$ é denominado *tempo de processamento* e denotado por t_j, $t_j \in \mathbb{Z}^+$. O conjunto de tarefas atribuído a uma estação k compreende a sua *capacidade*, denotado por S_k. O tempo total requerido para processar todas as tarefas atribuídas a cada estação $k = 1, \ldots, m$; é denominado *tempo de estação* e denotado por $t(S_k)$. O tempo comum disponível para completar todas as tarefas atribuídas em sequência a cada estação é o *tempo de ciclo*, denotado por c. O tempo de estação não deve exceder c. Se $t(S_k)$ é estritamente menor que c, então k tem *tempo ocioso* de $(c - t(S_k))$ unidades de tempo em cada ciclo.

Há sequências de tarefas que devem ser seguidas na montagem de qualquer produto e estas são representadas em um *grafo de precedência* $G = (V, A, t)$, que tem um conjunto de arestas A, representando as relações (i, j) de precedências necessárias entre diferentes tarefas $i, j \in V$. Isto é, a tarefa i deve ser completada antes que a tarefa j seja iniciada. G é acíclico, com V numerado topologicamente e $t(j)$:$V \rightarrow \mathbb{N}$, onde $t(j) = t_j$, $j = 1, \ldots, n$. Para cada tarefa $j \in V$, seu conjunto de predecessores diretos é definido como $P_j = \{i \in V \mid (i, j) \in A\}$ e seu conjunto de sucessores direto é definido como $F_j = \{i \in V \mid (j, i) \in A\}$. Para as relações indiretas de precedência, também são definidos o conjunto de todos os predecessores $P_j^* = \{i \in V \mid$ existe um caminho de arestas de i ate j em $G\}$ e o conjunto de todos os sucessores $F_j^* = \{i \in V \mid$ um caminho de arestas de j para i existe em $G\}$.

Gutjahr e Nemhauser [1964] mostraram que o SALBP é um problema de otimização combinatória pertencente à classe $\mathcal{NP}$-difícil. Além disso, instâncias práticas do SALBP podem ser extremamente grandes, com milhares de tarefas. Os planejadores frequentemente também têm que examinar um grande número de planos alternativos para lidar com informações incertas, restrições tecnológicas e logísticas do modelo. Por estas razões, os métodos heurísticos tornaram-se técnicas populares para resolver o SALBP. No entanto, com o surgimento de métodos de otimização modernos, como o relaxação Lagrangeana e a técnica de geração de colunas, agora existem métodos capazes de resolver instâncias de grande porte do SALBP.

Baybars [1986] especificou os seguintes pressupostos para o SALBP: (A1) todos os parâmetros de entrada são conhecidos com certeza; (A2) nenhuma tarefa pode ser dividida entre duas ou mais estações; (A3) tarefas não podem ser processadas em sequências arbitrárias, devido a requisitos tecnológicos de precedência; (A4) cada tarefa deve ser processada; (A5) todas as estações são identicamente equipadas e capazes de processar qualquer uma das tarefas; (A6) o tempo de processamento de uma tarefa é independente da estação em que é processada e das tarefas que a precedem; (A7) qualquer tarefa pode ser processada em qualquer estação; (A8) a linha é em série, sem paralelismo ou alimentação secundária; e (A9) a linha é projetada para a produção em massa de um modelo de um único produto; e (A10) o tempo de ciclo c é dado e fixo. O objetivo é minimizar m, o numero de estações.

Duas versões do SALBP têm sido extensivamente estudadas, por exemplo por Scholl [1998]. A primeira versão, SALBP-1, é definida como segue. Além das restrições (A-1) – (A-9), presume-se também que (A-10) o tempo de ciclo é dado e fixo. O objetivo é minimizar o número de estações ao longo da linha. A segunda versão do problema, SALBP-2, é o mesmo que SALBP-1, exceto que em vez de (A-10) presume-se que (A-11) o número de estações é dado e fixo. O objetivo é minimizar o tempo de ciclo ou, equivalentemente, maximizar a taxa de produção. A versão SALBP-2 é também conhecida como "ALBP tipo-2" (veja, por exemplo, Baybars [1986]). Neste trabalho, é abordado o SALBP-1, o qual é referenciado apenas por SALBP.

3. Balanceamento de linhas de montagem com divisão de tarefas

Como mencionado anteriormente, este trabalho apresenta um modelo matemático para um novo problema de balanceamento de linhas de montagem, o qual denotamos por TDALBP (*Task Division Assembly Line Balancing Problem*). O TDALBP é uma generalização do SALBP, com a suposição (A2) acima relaxada, no sentido de que certas tarefas podem, potencialmente, ser divididas de maneira particularmente conhecida. A ideia é que em vez de limitar o processamento de cada tarefa a exatamente uma estação, algumas delas podem ser potencialmente divididas, com a atribuição de cada uma das subtarefas para diferentes estações, de forma que a soma do tempo de processamento das subtarefas seja igual ao tempo de processamento da tarefa original. Obviamente, qualquer divisão faz com que uma penalidade de tempo possa ser adicionada à estação em que uma subtarefa for atribuída.

Mudanças na tecnologia, *design* de produto, processos de fabricação e projeto de trabalho podem criar oportunidades de tornar benéficas alterações no modo em que o conteúdo do trabalho total é dividido em tarefas. Isto pode significar que algumas das tarefas consideradas indivisíveis tornem-se potencialmente divisíveis, no sentido de que eles podem ser rentavelmente divididas em subtarefas que podem ser atribuídas a diferentes estações. Este pode ser especialmente o caso quando há um objetivo orientado no tempo, como no SALBP, e há variação significativa entre os tempos de processamento das tarefas, com alguns deles igual, ou perto de, do tempo de ciclo. Quando algumas das tarefas têm tempos de processamento relativamente longos e envolvem operações simples, tais como perfuração ou aperto de porcas/parafusos, parte da tarefa pode ser utilmente adiada para ser finalizada em outra estação. Por exemplo, suponha que uma tarefa consiste no aperto de vários parafusos e tem um dos maiores tempos de processamento dentre todas as tarefas. Suponha que apenas dois parafusos devem ser apertados para prender duas peças com segurança no lugar e o resto dos parafusos podem ser apertados em outra estação, sem prejuízo para a realização de outras tarefas intermediárias. Nesse caso, o número mínimo de estações necessárias pode, possivelmente, ser reduzido, mesmo quando a divisão de tarefas induz custos de penalidade de tempo adicional. Isso será ilustrado posteriormente em um exemplo numérico.

Um grande número de modelos SALBP baseados nos pressupostos (A1) – (A10) têm sido propostos, tais como aqueles de Patterson e Albracht [1975] e Scholl [1998]. Há uma série de outras formulações que relaxam alguns dos pressupostos (A3) – (A9), tais como as formulações do *ASALBP* (*Alternative Subgraphs Assembly Line Balancing Problem*) propostas por Capacho et al. [2009] e Scholl et al. [2009]. A contribuição deste trabalho é apresentar pela primeira vez, um modelo de SALBP com o pressuposto (A2) relaxado. Para permitir a possibilidade de divisão de tarefas, o modelo proposto na próxima seção é significativamente diferente dos modelos SALBP mencionados anteriormente.

Na formulação do TDALBP é também assumido que existem alternativas de processamento, mas apenas para um determinado subconjunto de tarefas potencialmente divisíveis. O nó que representa cada tarefa potencialmente divisível no grafo de precedência é expandido para um conjunto de nós adicionais (cada um com as mesmas relações de precedência que o nó original), onde cada novo nó representa uma possível subtarefa criada a partir da divisão da tarefa original.

Propõe-se agora uma apresentação do TDALBP que permite uma combinação única de subtarefas para cada tarefa potencialmente divisível a ser escolhida. O nó que representa cada tarefa potencialmente divisível em V é expandido para incluir um novo subconjunto de nós em G, os quais representam todas as subtarefas que poderiam ser criadas se a tarefa for dividida. A cada novo nó é atribuído as mesmas relações de precedência do nó original. A seguinte notação é utilizada:

$D =$ o conjunto de tarefas potencialmente divisíveis $(D \subseteq V, D \neq \emptyset)$;

$I =$ o conjunto de tarefas indivisíveis $(I \subseteq V, n \in I, V = D \cup I, D \cap I = \emptyset)$;

$r_j =$ o número de tempos de processamento no qual a tarefa j pode ser processada, $\forall\, j \in V$. (r_j é um conjunto como unidade, $\forall\, j \in I$).

$t_j^q =$ o q^{th} tempo de processamento (inteiro positivo) disponível para a tarefa j, $\forall\, j \in V, q = 1, \ldots, r_j$;

$T_j = \langle t_j^q \,|\, q = 1, \ldots, r_j \rangle =$ sequência (não crescente) dos tempos de processamento dados para os quais a tarefa $j \in V$ pode ser processada, onde $t_j^1 = t_j$, o tempo de processamento original da tarefa j e assim $t_j^p \geq t_j^q$ sempre que $p < q$, para $1 \leq p, q \leq r_j$ (T_j contém apenas a entrada única t_j^1, $\forall\, j \in I$);

$f_j^q =$ a penalidade de tempo (estritamente positiva) caso a tarefa j seja processada por t_j^q unidades de tempo, $\forall\, j \in V$, $q = 1, \ldots, r_j$ (f_j^1 é definido como zero para cada tarefa original j, $\forall\, j \in V$).

Com apenas um leve abuso de terminologia, a partir de agora, os termos tarefas e nós são usados indistintamente. Supõe-se que o nó terminal n, é o único em G sem sucessores. Se isso não ocorrer naturalmente em G, um último nó fictício com tempo nulo de processamento é introduzido. Supõe-se também que a tarefa n é indivisível. Para permitir que certas combinações possivelmente úteis de subtarefas, de cada tarefa potencialmente divisível $j \in D$, repetições de tempos de processamento de subtarefas t_j^q, para determinados valores de q, $2 \leq q \leq r_j$, podem aparecer em T_j, permitindo j ser dividido em subtarefas com tempos iguais de processamento. No entanto, presume-se que tais subtarefas ainda têm de ser processadas em diferentes estações. Por exemplo, se $t_j = 6$, pode ser útil dividir j em duas subtarefas, a serem processadas em diferentes estações, cada uma com tempo de processamento de 3 unidades. Não há nenhuma penalidade de tempo se a tarefa j não for dividida e é por isso que f_j^1 é definido como zero. Se a tarefa j for dividida, e isso resultar em uma subtarefa com tempo de processamento t_j^q, ao ser ativada em qualquer estação, o seu tempo de processamento total é definido como $(t_j^q + f_j^q)$, $\forall\, j \in D, q = 1, \ldots, r_j$. Note-se que todos os valores f_j^q são independentes da estação. Além disso, presume-se que não existem relações de precedência entre subtarefas de uma mesma tarefa potencialmente divisível.

O TDALBP consiste em ativar um subconjunto de subtarefas para cada tarefa $j \in D$ potencialmente divisível, atribuindo-as a estações (onde a tarefa j é considerada uma subtarefa) com as seguintes características:

- a subtarefa ativada satisfaz todas as relações de precedência que envolvem a tarefa j;
- a soma dos tempos de processamento das subtarefas ativadas é igual ao tempo de processamento fornecido t_j, da tarefa j;
- o tempo total de processamento de cada subtarefa ativada é a soma dos seus tempos de processamentos e penalidades de tempo fornecidos; e
- cada subtarefa, ativada, da tarefa j deve ser processada em uma estação distinta, $\forall\, j \in D$.

As subtarefas ativadas, de cada tarefa potencialmente divisível, tem seus arcos incidentes em G ignorados. O objetivo do TDALBP é identificar o melhor conjunto de subtarefas de todas as tarefas potencialmente divisíveis para que todas as tarefas ativadas (indivisíveis, e as subtarefas de tarefas potencialmente divisíveis) possam ser atribuídas ao número mínimo possível de estações, considerando o tempo de processamento final, o tempo de ciclo e as restrições de precedência.

Agora, essas ideias são ilustradas através do exemplo a seguir (vide Figura 1 e Tabela 1). As tarefas que são potencialmente divisível: $D = \{2, 3, 9, 13, 14, 18\}$, são representadas por nós redondos e o conjunto de tarefas indivisíveis: $V \backslash D = \{1, 4, 5, 6, 7, 8, 10, 11, 12, 15, 16, 17, 19, 20, 21, 22, 23\}$, são representadas por nós quadrados na Figura 1. Os tempos originais de processamento das tarefas são mostrados acima dos nós. Todas as subtarefas viáveis, seus tempos de processamento e de penalização são apresentados na Tabela 1.

Figura 1: Grafo de precedência do exemplo numérico.

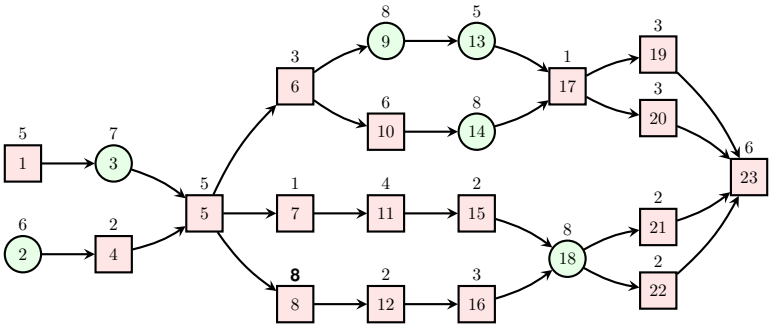


Tabela 1: Dados de entrada para a instância TDALBP.

j	t_j^1	t_j^2	t_j^3	$t_j^1+f_j^1$	$t_j^2+f_j^2$	$t_j^3+f_j^3$	f_j^1	f_j^2	f_j^3	E_j	F_j
2	6	3	3	6	4	4	0	1	1	2	3
3	7	4	3	7	5	4	0	1	1	1	3
9	8	6	2	8	7	3	0	1	1	4	10
13	5	3	2	5	4	3	0	1	1	5	11
14	8	5	3	8	6	4	0	1	1	4	10
18	8	6	2	8	7	3	0	1	1	5	11

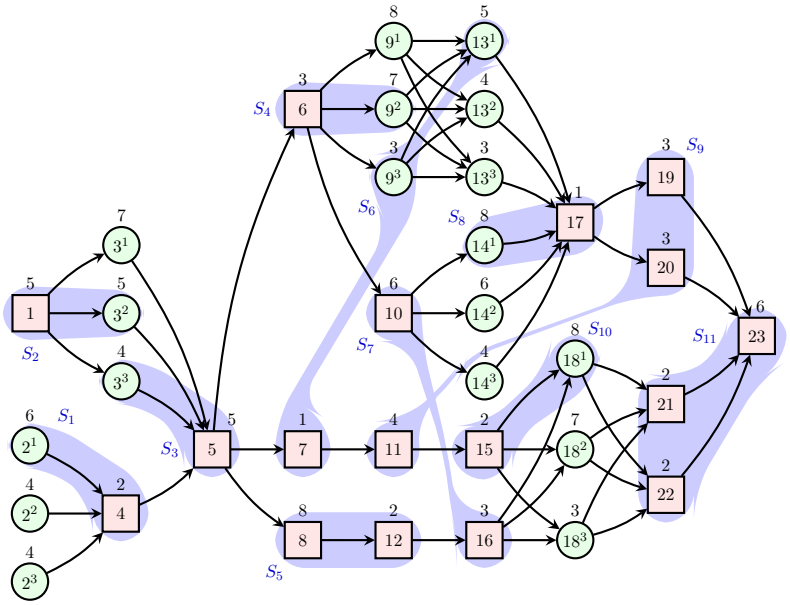
Como pode ser visto na Tabela 1, cada tarefa potencialmente divisível j, tem $r_j = 3$ opções de processamento: $T_j = \langle t_j^1, t_j^2, t_j^3 \rangle$. Note-se que duas subtarefas de uma tarefa potencialmente divisível têm os mesmos tempos de processamento determinados ($t_2^2 = t_2^3 = 3$). Neste exemplo, todas as penalidades de tempo é igual à unidade. A Figura 2 exibe a versão expandida de G, que inclui todas as possíveis sub-tarefas e seus tempos finais de processamento. Para tornar o processo de transformação claro, os índices de tarefas não foram renumerados topologicamente.

Esse exemplo pode ser inicialmente tratado como uma instância do SALBP, sem divisão e, portanto, sem penalidades de tempo. Neste caso, o número minimo de estações é $m^* = 12$. Uma solução ótima para esta instância, com o número mínimo de estações igual a 11 e com uma penalidade adicional de tempo de $f_3^2+f_3^3+f_9^2+f_9^3 = 4$ unidades de tempo, tem estações com cargas de $S_1 = \{2,4\}, S_2 = \{1,3^2\}, S_3 = \{3^3,5\}, S_4 = \{6,9^2\}, S_5 = \{8,12\}, S_6 = \{7,9^3,13\}, S_7 = \{10,16\}, S_8 = \{14,17\}, S_9 = \{11,19,20\}, S_{10} = \{15,18\}, S_{11} = \{21,22,23\}$. Os conjuntos S_i, $i = 1,\ldots,11$, relacionados a essa solução, são demostrados na Figura 2. Comparada à solução por SALBP, o número minimo de estações foi reduzido em uma unidade.

4. Modelo matemático para o TDALBP

Aqui é apresentado um modelo compacto para o TDALBP. Como mencionado anteriormente, uma tarefa n é supostamente indivisível e única no sentido de que não possui tarefas sucessoras. Cada nó j é utilizado para gerar um conjunto de nós $D_j = \{j\} \cup \{j^q \mid q = 2,\ldots,r_j\}$, formado pelas possíveis subtarefas da tarefa j, (incluindo a própria tarefa j) correspondendo as entradas em T_j, i.e.: $t_j^q, \forall\, j \in D, q = 1,\ldots,r_j$. Cada nó $j \in D$ é substituído por D_j em G. A cada novo nó j^q, na versão expandida de G, são atribuídas as mesmas relações de precedência do nó j, $\forall\, j \in D, q = 2,\ldots,r_j$. Isso é realizado da seguinte maneira. Se a aresta $(h^p, j^q) \in A$ para algum p, $1 \le p \le r_h$, e para algum j^q, $j \in D$ e $1 \le q \le r_j$, então as arestas $(h^p, j^q), \forall\, p = 1,\ldots,r_h$; e $q = 1,\ldots,r_j$; são adicionadas a A. Se a aresta $(j^q, i^s) \in A$ para algum j^q, $j \in D$ e $1 \le q \le r_j$, e para algum s, $1 \le s \le r_i$, então as arestas (j^q, i^s), $\forall\, q = 1,\ldots,r_j$; e todos $s = 1,\ldots,r_i$; são adicionados a A. Cada novo nó j^q é atribuído um tempo de processamento de $(t_j^q + f_j^q)$. Os tempos de processamento de todos os nós originais em V mantêm-se inalterados. Assim V agora está particionado em $V = I \cup (\cup_{j \in D} D_j)$ e n é aumentada de modo a representar o novo número

Figura 2: Versão expandida do grafo de precedência do exemplo numérico.



total de nós. Todos os nós no grafo transformado são reindexados topologicamente. Para cada nó potencialmente divisível j, um registro deve ser mantido sobre os nós recém criados das subtarefas em D_j, $\forall\, j \in D$. O processo de transformação é ilustrado na Figura 2.

Cada tarefa indivisível em I deve ser atribuída a exatamente uma estação. Para cada nó potencialmente divisível $j \in D$, considerar as subtarefas in D_j, que incluem a própria tarefa j.Neste caso, (i) a tarefa j é ativada inteira sozinha e atribuída a uma única estação, onde será processada t_j unidades de tempo, sem penalidades; ou (ii) uma ou mais subtarefas de D_j (excluindo a tarefa original j) são ativadas e atribuídas a diferentes estações; onde serão processadas no tempo t_j^q, com a penalidade adicional f_j^q. Os tempos de processamento das subtarefas ativadas devem somar o tempo de processamento t_j, da tarefa j.

Com o objetivo de que o modelo seja o mais compacto possível, a seguinte notação e terminologia é utilizada para o grafo expandido. Para cada tarefa j seja E_j a primeira estação na qual j pode ser atribuída, dada as relações de precedência, os tempos de processamento e o tempo de ciclo, $\forall\, j \in V$. A formula comum para calcular E_j para o SALBP foi primeiramente introduzida por Patterson e Albracht [1975]. Lembrando que para o TDALBP, a sequência T_j, dos tempos de processamento das possíveis subtarefas de cada tarefa potencialmente divisível j é estritamente decrescente e assim $t_j^{r_j}$ é mínimo entre todos os elementos de T_j, $\forall\, j \in D$. Utilizando esse fato, a seguinte modificação da formula de Patterson e Albracht pode ser utilizada para calcular E_j para o ASALBP:

$$E_j = \begin{cases} 1, & \text{se } (t^{r_j} + \sum_{i \in P_j^*} t_i) = 0, \text{ e} \\ \lceil (t^{r_j} + \sum_{i \in P_j^*} t_i)/c \rceil, & \text{caso contrário}, \forall\, j \in V. \end{cases} \tag{1}$$

Seja F_j a última estação viável na qual j pode ser atribuída, dadas as relações de precedência, os tempos de processamento e o tempo de ciclo, $\forall\, j \in V$. Então F_j pode ser calculada analogamente como:

$$F_j = \begin{cases} m', & \text{se } (t^{r_j} + \sum_{i \in F_j^*} t_i) = 0, e \\ m' + 1 - \lceil (t^{r_j} + \sum_{i \in F_j^*} t_i) \rceil, & \text{caso contrário}, \forall\, j \in V, \end{cases} \tag{2}$$

onde m' ($\leq n$) é o número máximo de estações necessário para identificar uma solução viável.

Note-se que m' é frequentemente computado por um algorítimo heurístico, mas se isso não é possível, m' pode ser definida para ser n, implicando que, inicialmente, cada tarefa é atribuída a sua própria estação individual. Assim que os parâmetros definidos em (1) e (2) são estabelecidos, o *intervalo de estação* $SI_j = [E_j, F_j]$, pode ser computado definindo o intervalo de estações no qual a tarefa j pode ser atribuída. Se a tarefa j é indivisível (potencialmente divisível ou uma subtarefa) ela deve ser atribuída a exatamente (uma única estação) em $SI_j, \forall j \in V$. Para tornar mais compacto o modelo proposto, é introduzido o *intervalo de estação* $B_k = \{j \in V \,|\, k \in SI_j\}$, para cada estação $k = 1, \ldots, m'$. O uso dos parâmetros SI_j e B_k habilitam uma redução significante (comparado com os modelos tradicionais de outras variações do SALBP) no número de restrições no modelo proposto. O modelo proposto é introduzido agora:

- Índices: j (para tarefas), k (para estações) e q (para subtarefas e seus respectivos tempos de processamento).

- Parâmetros: $n =$ o número de tarefas ($\forall\, j = 1, \ldots, n$); $m' =$ o número máximo de estações requerido ($\forall k = 1, \ldots, m'$); $V =$ o conjunto de todas as tarefas ($|V| = n$); $I =$ o conjunto de tarefas indivisíveis; $D =$ o conjunto de tarefas potencialmente divisíveis; $D_j =$ o conjunto de subtarefas (incluindo a própria j) no qual a tarefa j pode ser potencialmente dividida ($\forall\, j \in D$); $t_j =$ o tempo de processamento (sem penalidade de tempo) para a tarefa j ($\forall\, j = 1, \ldots, n$); $f_j^q =$ o tempo de penalidade para o processamento da subtarefa j ($\forall\, j = 1, \ldots, n$; $\forall\, q = 1, \ldots, r_j$); P_j $(F_j) =$ o conjunto de predecessores imediatos (sucessores) da tarefa j ($\forall\, j = 1, \ldots, n$); $E_j =$ a primeira estação viável na qual a tarefa j pode ser atribuída ($\forall\, j = 1, \ldots, n$); $L_j =$ a ultima estação viável na qual a tarefa j pode ser atribuída ($\forall j = 1, \ldots, n$); $SI_j = [E_j, L_j] =$ o intervalo de estação no qual a tarefa j pode ser atribuída ($\forall j = 1, \ldots, n$); $B_k =$ o conjunto de tarefas que pode ser potencialmente atribuídas à estação k ($\forall\, k = 1, \ldots, m'$).

- Variáveis de decisão:

$$x_{jk}^q = \begin{cases} 1, & \text{se a subtarefa } q \text{ da tarefa } j \text{ é atribuída a estação } k; \\ 0, & \text{caso contrário; } j = 1, \ldots, n;\ q = 1, \ldots, r_j;\ \forall\, k \in SI_j. \end{cases}$$

O índice da estação na qual a tarefa j é atribuída, se ela for realmente atribuída, deve ser um membro de SI_j e deve ser computado como $\sum_{q=1}^{r_j} \sum_{k \in SI_j} k \cdot x_{jk}^q$, $\forall\, j = 1, \ldots, n$. Se essa expressão resultar no valor zero para um conjunto dado de variáveis de decisão x_{jk}^q, então a tarefa j não é atribuída. Como pode ser visto em (3) a seguir, dado que é assumido que o nó terminal em G não tem sucessores, se j é configurado para n na expressão anterior, o número de estações requerido pode ser expresso como uma simples soma ponderada. Alem disso, apenas tarefas que são membros de B_k podem ser atribuídas à estação k, $\forall\, k = 1, \ldots, m'$. O modelo TDALBP proposto, (3)–(12), o qual envolve apenas variáveis binarias de decisão x_{jk}^q, é dado em seguida. Se a tarefa j é indivisível, isto é, $j \in I$ e $r_j = 1$, então x_{jk}^1 e t_j^1 são simplesmente denotados por x_{jk} e t_j, respectivamente.

$$\text{Minimizar} \sum_{k \in SI_n} k \cdot x_{nk}, \tag{3}$$

sujeito a

$$\sum_{k \in SI_j} x_{ik} = 1, \qquad \forall\, i \in I; \tag{4}$$

$$\sum_{k \in SI_j} \sum_{q \in D_j} t_j^q \cdot x_{jk}^q = t_j^1, \qquad \forall\, j \in D; \tag{5}$$

$$\sum_{q\in D_j} x_{jk}^q \leq 1, \qquad \forall\, j \in D,\ k \in SI_j; \tag{6}$$

$$\sum_{i\in I\cap B_k} t_i{\cdot}x_{ik} + \sum_{j\in D}\sum_{q\in D_j\cap B_k} (t_j^q + f_j^q){\cdot}x_{jk}^q \leq c, \qquad \forall\, k = 1,\ldots,m'; \tag{7}$$

$$\sum_{k\in SI_i} k{\cdot}x_{ik} - \sum_{k\in SI_j} k{\cdot}x_{jk} \leq 0, \qquad \forall\, i \in I,\ j \in I\cap F_i,\ L_i \geq E_j; \tag{8}$$

$$\sum_{k\in SI_i} k{\cdot}x_{ik} - \sum_{k\in SI_j} k{\cdot}x_{jk}^q + m'{\cdot}\sum_{k\in SI_j} x_{jk}^q \leq m', \qquad \forall\, i \in I,\ q \in \bigcup_{j\in D} D_j\cap F_i,\ L_i \geq E_j; \tag{9}$$

$$\sum_{k\in SI_j} k{\cdot}x_{jk}^q - \sum_{k\in SI_i} k{\cdot}x_{ik} + m'{\cdot}\sum_{k\in SI_j} x_{jk}^q \leq m', \qquad \forall\, q \in \bigcup_{j\in D} D_j,\ i \in I\cap F_j,\ L_j \geq E_i; \tag{10}$$

$$\begin{aligned} &\sum_{k\in SI_{j'}} k{\cdot}x_{j'k}^{q'} - \sum_{k\in SI_{j''}} k{\cdot}x_{j''k}^{q''} + \\ &m'{\cdot}\Big(\sum_{k\in SI_{j'}} x_{j'k}^{q''} + \sum_{k\in SI_{j''}} x_{j''k}^{q''}\Big) \leq 2{\cdot}m', \quad \forall\, q' \in \bigcup_{j'\in D} D_{j'}, \\ &\qquad\qquad q'' \in \bigcup_{j''\in D} D_{j''}\cap F_{j'},\ L_{j'} \geq E_{j''}; \end{aligned} \tag{11}$$

$$\begin{aligned} x_{ij}^q \in \{0,1\},\ &\forall\, j = 1,\ldots,n;\ q = 1,\ldots,r_j; \\ &k \in SI_j. \end{aligned} \tag{12}$$

A função objetivo (3) minimiza o índice da estação da tarefa terminal n, o que equivale a minimizar o número de estações requerido. A restrição (4) garante que cada tarefa indivisível é atribuída a uma única estação. A restrição (5) garante que a soma dos tempos de processamento de subtarefas ativadas, de cada tarefa potencialmente divisível, é igual ao tempo de processamento da tarefa. A restrição (6) garante que as subtarefas ativas, de qualquer tarefa potencialmente divisível, são atribuídas a diferentes estações. A restrição de tempo de ciclo é dada em (7), onde, para cada estação, a primeira expressão representa os tempos de processamento de tarefas indivisíveis (sem penalidades de tempo) e a segunda representa aquelas potencialmente divisíveis (com penalidades de tempo). (8)–(11) lida com possíveis restrições de precedência de tarefas que podem surgir da existência da aresta $(i,j) \in A$, dependendo da sua natureza, e das relações entre as tarefas i e j (lembrando que o conjunto de subtarefas de qualquer tarefa potencialmente divisível inclui a própria tarefa). Quando i e j são ambas indivisíveis, a restrição de precedência é dada em (8). Quando i é indivisível, mas j é uma subtarefa de um tarefa potencialmente divisível, a restrição de precedência decorre de que (i,j) deve se manter somente se j é ativada. Isso é, quando j é atribuída a alguma estação, ou seja, $\sum_{k\in SI_j} x_{jk}^q = 1$. As restrições correspondentes são dadas em (9). Quando i é uma subtarefa de uma tarefa potencialmente divisível mas j é indivisível, a restrição de precedência decorre de que (i,j) deve se manter somente se i é ativada. Isto é, quando i é atribuída a alguma estação, ou seja, $\sum_{k\in SI_i} x_{ik} = 1$. A restrição correspondente é dada em (10). Quando ambas i e j são subtarefas de duas tarefas distintas, potencialmente divisíveis, a restrição de precedência decorre de que (i,j) deve se manter apenas se ambas i e j forem ativadas. Isso é, quando i e j são ambas atribuídas a estações, ou seja, $\sum_{k\in SI_i} x_{ik} = \sum_{k\in SI_j} x_{jk}^q = 1$. A restrição correspondente é dada em (11). A restrição (12) é a condição binaria comum em x_{jk}^q's.

O número de variáveis binárias de decisão no modelo (3)–(12) é no máximo $m'{\cdot}n$. Usando-se a notação $\mathcal{O}$, a ordem de magnitude é $\mathcal{O}(|I|{\cdot}|D| + m')$ restrições. Utilizando-se as relações $m' \leq n$, $|I| \leq n$ e $|D| \leq n$, o modelo requer $\mathcal{O}(n^2)$ variáveis e restrições. Essas magnitudes são uma limitação do modelo. A fim de se obter planos práticos viáveis para instâncias industriais de grande escala, pode ser necessário restringir o número de tarefas divisíveis e o número de tempos de processamento (número de subtarefas) para cada uma delas.

5. Resultados computacionais

A fim de examinar o desempenho do pacote de otimização CPLEX [IBM ILOG, 2016] na resolução do modelo descrito na Seção 4, foram realizados experimentos computacionais com base em instâncias de teste de *benchmark* sistematicamente construídas. Todos os testes computacionais foram realizados em um computador pessoal com processador Intel Core i3, de 3GHz e 8GB de RAM, usando apenas um núcleo. Um limite de 1.800 segundos de tempo de processamento foi imposto para cada instância.

As instâncias usadas nos testes foram geradas a partir de um conjunto de 269 instâncias do SALBP, com base em 25 grafos de precedência com 8 a 297 nós, com diferentes tempos de ciclo. A descrição detalhada desse conjunto de dados está disponível em Scholl et al. [2016]. Para gerar tipos diferentes de instâncias TDALBP, foram adaptadas cada uma das instâncias SALBP, mantendo o mesmo tempo de ciclo, tarefas e precedências, mas introduzindo os parâmetros I, D, D_j, r_j e f_j e valores específicos para eles. Os nós do grafo de precedência associado a cada instância gerada foram renumerados em ordem topológica.

Para selecionar o conjunto D de tarefas potencialmente divisíveis, foi utilizada uma heurística simples baseada na mediana dos tempos de execuções das tarefas ($X(t_j)$). As tarefas que possuíam o tempo de execução maior que a mediana foram selecionadas para o conjunto D e as restantes compuseram o conjunto I de tarefas indivisíveis. O conjunto D foi então subdivido em dois subconjuntos, D_1 e D_2, de forma que uma tarefa $j \in D$ foi alocada a D_1 se $t_j > \delta{\cdot}X(t_j)$ ou alocada a D_2 se $t_j \leq \delta{\cdot}X(t_j)$. Os valores $\delta = 1{,}2$ e $\delta = 1{,}5$ foram testados na alocação das tarefas aos subconjuntos D_1 e D_2. As tarefas em D_1 foram divididas em duas subtarefas e aquelas em D_2 foram divididas em três subtarefas, sempre de modo que o somatório dos seus tempos de execução fosse igual ao da tarefa original.

A Tabela 2 resume os resultados dos testes com as instâncias para as quais foram obtidas reduções no número de estações necessárias ao processamento de todas as tarefas. A tabela também contém duas medidas comumente usadas para avaliar uma solução viável: Eficiência de Linha (EL), o percentual de utilização da linha, a qual é expressa como a razão entre o tempo total das estações e o tempo de ciclo, multiplicado pelo número de estações; e o Tempo de Linha (TL), o período de tempo necessário para cada unidade do produto ser completado na linha de montagem.

Os seguintes parâmetros e medidas estão presentes na Tabela 2: c – tempo de ciclo; n – número de tarefas; m^*, EL e TL – respectivamente, número de estações, eficiência de linha e tempo de linha da solução do ALBP; m^*_δ, EL_δ, TL_δ e F_delta – respectivamente, número de estações, eficiência de linha, tempo de linha e penalidade de tempo da solução das instâncias TDALBP construída com os parâmetros $\delta = 1{,}2$ e $\delta = 1{,}5$; t – indicativo se o limite de 1.800 segundos de tempo de processamento foi atingido com as instâncias construídas com $\delta = 1{,}5$ e $\delta = 1{,}2$ (* – indica para as duas classes de instâncias, ** – apenas a instância da classe $\delta = 1{,}5$ e *** – apenas a instância da classe $\delta = 1{,}2$).

A heurística utilizada para selecionar as tarefas potencialmente divisíveis, apesar de sua simplicidade, demonstra o potencial da divisão de tarefas na otimização do desempenho de um sistema de balanceamento de linha de montagem. A variação no parâmetro δ influencia diretamente nos resultados finais, mudando a forma como as tarefas são subdividas. Com o valor $\delta = 1.5$, a redução no número de estações foi alcançada em 16 das 269 instâncias geradas. Já com $\delta = 1.2$, além dessas 16 instâncias, mais duas outras também tiveram redução no número de estações. Contudo, há variações no valor da penalidade, eficiência da linha, tempo de linha e até mesmo no número das estações na melhor solução TDALBP, tanto para melhor quanto para pior. Isso pode ser observado, por exemplo, nas linhas 7, 9 e 11 da tabela. Note-se que nesses três casos o tempo limite de execução foi alcançado e uma solução ainda melhor pode existir para essas instâncias.

6. Conclusões

Neste trabalho, é apresentado um novo problema de balanceamento da linha de montagem de divisão da tarefa, em que é possível dividir ainda mais tarefas particulares de forma limitada,

Tabela 2: Resultados computacionais.

Grafo	c	n	m^*	EL	TL	$m^*_{1,5}$	$F_{1,5}$	$\text{EL}_{1,5}$	$\text{TL}_{1,5}$	$m^*_{1,2}$	$F_{1,2}$	$\text{EL}_{1,2}$	$\text{TL}_{1,2}$	t
GUNTHER	41	35	14	84.15	539	13	9	90.62	532	13	9	90.62	532	
SAWYER30	30	30	12	90.00	353	11	3	98.18	328	11	3	98.18	328	
WEE-MAG	56	75	30	89.23	1676	28	14	96.49	1568	28	26	97.26	1568	*
WEE-MAG	54	75	31	89.55	1669	30	22	92.53	1615	30	25	92.53	1620	*
WEE-MAG	45	75	38	87.66		35	18	95.17	1573	35	21	95.17	1575	*
WEE-MAG	43	75	50	69.72	2134	39	34	89.39	1669	39	28	89.39	1670	
WEE-MAG	42	75	55	64.89	2310	41	36	89.14	1716	40	32	91.13	1679	**
WEE-MAG	41	75	59	61.97	2419	42	34	87.05	1719	42	38	87.05	1717	
WEE-MAG	35	75	60	71.38	2096	45	42	97.84	1575	46	35	95.28	1610	*
WEE-MAG	36	75	60	69.40	2151	44	36	94.63	1584	44	44	94.63	1584	*
WEE-MAG	37	75	60	67.52	2219	43	38	96.61	1589	44	43	94.72	1626	*
WEE-MAG	38	75	60	65.75	2267	43	38	91.74	1631	43	39	91.74	1634	*
WEE-MAG	39	75	60	64.06	2334	42	38	91.51	1634	42	36	91.51	1638	
WEE-MAG	40	75	60	62.46	2381	42	38	89.23	1678	42	38	89.23	1677	
WEE-MAG	33	75	61	74.47	2008	57	40	81.82	1875	57	30	81.29	1876	***
WEE-MAG	34	75	61	72.28	2073	53	34	85.07	1796	53	48	85.85	1800	
WARNECKE	58	58	29	92.03	1679	—	—	—	—	28	27	96.98	1618	***
WARNECKE	54	58	31	92.47	1672	—	—	—	—	30	30	97.41	1618	***

mesmo que a divisão induza custos adicionais de penalidade de tempo. Um exemplo simples é fornecido para ilustrar como a divisão de tarefas no processo de balanceamento pode melhorar a atribuição de tarefas, reduzindo o número de estações necessárias para um determinado ciclo de tempo. É introduzido um modelo matemático do problema com o objetivo de resolver instâncias numéricas por um *software* comercial de otimização. Os resultados preliminares de um extenso estudo computacional demonstram que a abordagem parece promissora. Possíveis direções para futuras pesquisas, a fim de buscar uma melhor compensação entre a qualidade da solução e tempo computacional, incluem a adaptação de métodos de solução do ASALBP de Scholl et al. [2009] e metaheurísticas, tais como algoritmos genéticos e busca tabu. Além disso, um esforço de investigação é necessário para analisar a estrutura matemática do problema mais profundamente.

Agradecimentos

Os autores Carlos Alexandre X. Silva e Leslie R. Foulds agradecem o suporte financeiro recebido da CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) via bolsas DS e PVNS, respectivamente.

Referências

Baybars, I. (1986). A survey of exact algorithms for the simple assembly line balancing problem. *Management Science*, 32(8):909–932.

Bryton, B. (1954). Balancing of a continuous production line. Master's thesis, Northwestern University, Evanston, IL, USA.

Capacho, L., Pastor, R., Dolgui, A., e Guschinskaya, O. (2009). An evaluation of constructive heuristic methods for solving the alternative subgraphs assembly line balancing problem. *Journal of Heuristics*, 15(2):109–132.

Grzechca, W. e Foulds, L. R. (2015). The Assembly Line Balancing Problem with Task Splitting: A Case Study. *IFAC-PapersOnLine*, 48(3):2002–2008. 15th {IFAC} Symposium on Information Control Problems in Manufacturing, INCOM 2015.

Gutjahr, A. L. e Nemhauser, G. L. (1964). An algorithm for the balancing problem. *Management Science*, 11(2):308–315.

IBM ILOG (2016). IBM ILOG CPLEX Optimizer. http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/.

Patterson, J. H. e Albracht, J. J. (1975). Assembly line balancing: Zero-one programming with Fibonacci search. *Operations Research*, 23(1):166–172.

Salveson, M. E. (1955). The assembly line balancing problem. *The Journal of Industrial Engineering*, 6(3):18–25.

Scholl, A., Boysen, N., e Fliedner, M. (2009). Optimally solving the alternative subgraphs assembly line balancing problem. *Annals of Operations Research*, 172(1):243–258.

Scholl, A. (1998). *Balancing and sequencing of assembly line*. Physica-Verlag, Heidelberg.

Scholl, A., Boysen, N., Fliedner, M., e Klein, R. (2016). Assembly line balancing – homepage for assembly line optimization research. www.assembly-line-balancing.de. [Online; accessed 21-April-2016].