



Rotação de tarefas em linhas de produção com trabalhadores deficientes e estações em paralelo

Felipe Francisco Bezerra Araújo

Universidade de São Paulo
Instituto de Ciências Matemáticas e de Computação, USP, Brasil
felipefb@icmc.usp.br

Alysson Machado Costa

University of Melbourne
Department of Mathematics and Statistics - University of Melbourne, Australia
alysson.costa@unimelb.edu.au

Cristóbal Miralles

Universtat Politècnica de València
ROGLE - Dpto. Organización de Empresas , UPV, Espanha
cmiralles@omp.upv.es

RESUMO

Neste trabalho estudamos o problema de rotação de tarefas em linhas de produção com trabalhadores deficientes. Nosso objetivo é mostrar que a flexibilidade obtida pelo uso de estações em paralelo pode ser útil para a obtenção de soluções mais eficientes. Apresentamos um modelo matemático e desenvolvemos uma heurística de solução que faz uso de programação inteira mista. Uma bateria de testes computacionais com instâncias conhecidas na literatura é realizada e os resultados confirmam as hipóteses apresentadas.

PALAVRAS CHAVE. Linhas de produção. Trabalhadores heterogêneos. Rotação de tarefas.

Área Principal: Otimização Combinatória, Programação Matemática

ABSTRACT

We study the job rotation problem in assembly lines employing workers with disabilities. We propose that the use of parallel workstations may result in more efficient solutions. We present a mathematical model and a hybrid heuristic that makes use of mixed integer programming. Computational tests with widely used literature instances confirm the presented hypotheses.

Keywords: Assembly lines. Heterogeneous workers. Job rotation.

Main Area: Combinatorial Optimization, Mathematical Programming

1. Introdução

O *problema de balanceamento de linhas de produção* (ALBP) é um problema clássico de pesquisa operacional. Dado um conjunto de estações de trabalho, um conjunto de tarefas a serem executadas e uma ordem de precedência parcial entre estas tarefas, deseja-se encontrar uma alocação de tarefas a estações de modo a otimizar alguma métrica. No caso mais simples, os tempos das tarefas são conhecidos e independentes da estação nas quais são executados. Este problema é conhecido na literatura como *problema simples de balanceamento de linhas de produção* (SALBP) (Baybars; 1986; Becker e Scholl; 2006).

Uma variante de interesse do SALBP é o *problema de balanceamento de linhas e designação de trabalhadores* (ALWABP). Nesta extensão, os tempos de execução das tarefas são conhecidos, porém dependem do trabalhador que as executa, representando bem a situação encontrada, por exemplo, em linhas de produção de Centros de Trabalho para Deficientes (CTDs). O ALWABP foi proposto por Miralles et al. (2007) que modelaram o problema e realizaram um estudo de caso em um CTD na Espanha. Devido à complexidade do ALWABP, outras pesquisas recentes focaram em métodos heurísticos e metaheurísticos, como *clustering search* (Chaves et al.; 2007, 2009), *busca tabu* (Moreira e Costa; 2009), *iterated beam search* (Blum; 2010), heurísticas construtivas (Moreira et al.; 2012) e algoritmos genéticos (Moreira e Costa; 2012; Mutlu et al.; 2013).

Neste trabalho, nos interessamos por uma variante do ALWABP que considera além da heterogeneidade dos trabalhadores, a aplicação de rotação de tarefas. A motivação prática é obter balanceamentos que possam se utilizar dos benefícios associados à diversificação das alocações de tarefas a trabalhadores (Azizi et al.; 2010; Otto e Scholl; 2012, e.g).

O problema de rotação de tarefas no contexto do ALWABP já foi brevemente estudado em (Costa e Miralles; 2009; Moreira e Costa; 2012). Neste trabalho, estendemos estas pesquisas para lidar com casos em que as linhas contém layouts em paralelo (Araújo et al.; 2010a,b, 2014). De um ponto de vista prático, queremos demonstrar que a introdução destas novas configurações pode ser benéfica por aumentar o espaço possível de soluções. De fato, conforme será visto nas seções 4 e 5, em alguns casos é possível provar matematicamente que estas soluções são melhores do que qualquer solução serial para a mesma instância. De um ponto de vista teórico, trata-se de um problema combinatorial de difícil resolução para o qual propomos métodos eficientes de solução.

O restante deste artigo está dividido da seguinte forma: nas seções 2 e 3 apresentamos um modelo linear e um método híbrido de resolução para o problema, respectivamente. Na seção 4 descrevemos uma medida de qualidade das soluções do problema. A seção 5 apresenta os resultados obtidos com o modelo e o método desenvolvidos. Por fim, na seção 6 apresentamos algumas conclusões, bem como propostas de trabalhos futuros.

2. Modelo matemático

Nesta seção apresentamos um modelo linear para o problema de rotação de tarefas com estações em paralelo. Este modelo é baseado naquele apresentado por Araújo et al. (2012) para o ALWABP com estações em paralelo. Usando a seguinte notação:

w	Índice dos trabalhadores
i, j	Índices das tarefas
s	Índice das estações
p	Índice dos períodos
k	Índice do número de estações em paralelo
W	Conjunto de trabalhadores
N	Conjunto de tarefas
S	Conjunto de estações
P	Conjunto de períodos
D_j	Conjunto das tarefas que precedem a tarefa j no grafo de precedência

K_{Max}	Número máximo de estações em paralelo
e_{wi}	Tempo de execução da tarefa i quando realizada pelo trabalhador w
\underline{F}	Número mínimo de ítems a serem processados

E as variáveis:

z_{wi}	Variável binária. Igual a um se o trabalhador w executa a tarefa i em pelo menos um período
t_{isp}	Variável binária. Igual a um se a tarefa i é designada à estação s no período p
x_{swip}	Variável binária. Igual a um se a tarefa i é designada ao trabalhador w na estação s no período p
a_{skp}	Variável binária. Igual a um se o estágio s do período p possui k estações em paralelo
F_p	Taxa de produção da linha no período p
f_{swp}	Taxa de produção do trabalhador w no estágio s do período p
v_{swip}	Variável auxiliar. Usada na linearização do cálculo da taxa de produção

Podemos escrever o seguinte modelo para o problema de rotação de tarefas com trabalhadores em paralelo:

$$Max \quad \sum_{w \in W} \sum_{i \in N} z_{wi} \quad (1)$$

$$s.a. \quad \sum_{s \in S} st_{isp} \leq \sum_{s \in S} st_{jsp} \quad \forall i, j \in N | i \in D_j, \forall p \in P \quad (2)$$

$$\sum_{w \in W} x_{swip} \geq k(t_{isp} + a_{skp} - 1) \quad \forall s \in S, \forall k \in [0, K_{Max}], \forall i \in N, \forall p \in P \quad (3)$$

$$x_{swip} \leq t_{isp} \quad \forall s \in S, \forall w \in W, \forall i \in N, \forall p \in P \quad (4)$$

$$\sum_{k=0}^{K_{Max}} a_{skp} = 1 \quad \forall s \in S, \forall p \in P \quad (5)$$

$$\sum_{w \in W} y_{swp} = \sum_{k=0}^{K_{Max}} ka_{skp} \quad \forall s \in S, \forall p \in P \quad (6)$$

$$\sum_{s \in S} y_{swp} = 1 \quad \forall w \in W, \forall p \in P \quad (7)$$

$$\sum_{s \in S} t_{isp} = 1 \quad \forall i \in N, \forall p \in P \quad (8)$$

$$x_{swip} \leq y_{swp} \quad \forall s \in S, \forall w \in W, \forall i \in N, \forall p \in P \quad (9)$$

$$x_{swip} = 0 \quad \forall w \in W, \forall s \in S, \forall i \in I_w, \forall p \in P \quad (10)$$

$$\sum_{w \in W} f_{swp} \geq F_p - Mz_{s0p} \quad \forall s \in S, \forall p \in P \quad (11)$$

$$\sum_{i \in N} e_{wi} v_{swip} = y_{swp} \quad \forall w \in W, \forall s \in S, \forall p \in P \quad (12)$$

$$f_{swp} \leq M y_{swp} \quad \forall w \in W, \forall s \in S, \forall p \in P \quad (13)$$

$$v_{swip} \geq f_{swp} - M(1 - x_{swip}) \quad \forall s \in S, \forall w \in W, \forall i \in N, \forall p \in P \quad (14)$$

$$v_{swip} \leq M x_{swip} \quad \forall s \in S, \forall w \in W, \forall i \in N, \forall p \in P \quad (15)$$

$$z_{wi} \leq \sum_{p \in P} x_{swip} \quad \forall s \in S, \forall w \in W, \forall i \in N \quad (16)$$

$$\sum_{p \in P} F_p \geq \underline{F} \quad (17)$$

A função objetivo (1) maximiza o número de alocações tarefa-trabalhador. As restrições (2) - (15) garantem que cada período possui uma configuração válida para a linha de produção. As restrições (2) garantem as relações de precedência entre as tarefas. As restrições (3) definem que em um estágio com k estações em paralelo cada tarefa deve ser executada por k trabalhadores. As restrições (4) garantem que cada trabalhador só executa tarefas no estágio ao qual ele foi alocado. As restrições (5) dizem que cada estágio possui apenas um nível de paralelismo. As restrições (6) e (7) definem que cada trabalhador é alocado a um único estágio e que cada estágio possui k trabalhadores. As restrições (8) dizem que cada tarefa deve ser alocada a um único estágio em cada período. As restrições (9) dizem que um trabalhador só pode executar tarefas no estágio ao qual ele foi alocado. As restrições (10) evitam a alocação de tarefas ineficazes. As restrições (11) - (15) correspondem àquelas usadas por Araújo et al. (2012) para linearizar o cálculo da taxa de produção. As restrições (16) calculam o número de tarefas distintas executadas por cada trabalhador. Por fim, a restrição (17) garante a produtividade mínima desejada em todos os períodos.

Este modelo define formalmente o problema. Ele também foi usado para resolver instâncias de pequeno e médio porte com pacotes comerciais, porém ele se mostrou difícil de resolver. Mais detalhes destes testes podem ser vistos na seção 5. Com isto, surgiu a necessidade de desenvolvimento de métodos mais rápidos, como o descrito na seção seguinte.

3. Método de solução

Nesta seção propomos um método híbrido para resolver o problema de rotação de tarefas com estações em paralelo. Este método usa heurísticas construtivas simples e um método baseado na metaheurística *greedy randomized adaptive search procedure* (GRASP) para gerar um *pool* de configurações para a linha de produção e, em seguida, utiliza um modelo linear para encontrar a melhor solução usando estas configurações. Este método é uma adaptação daquele desenvolvido por Moreira e Costa (2012) para o ALWABP serial com rotação de tarefas.

3.1. GRASP

O GRASP é uma metaheurística com múltiplos reinícios. Em cada iteração, o método gera uma solução usando uma heurística construtiva e então aplica um método de busca local. Este processo é repetido por um determinado número de iterações e o método retorna então a melhor solução encontrada.

Neste trabalho, adaptamos as heurísticas construtivas propostas por Araújo et al. (2012) para o problema ALWABP com estações em paralelo. Estas heurísticas iniciam com um limitante inferior para o tempo de ciclo e utilizam diversos critérios de seleção de tarefas e trabalhadores para tentar construir uma solução. Caso encontremos uma solução viável, ela é retornada como a solução do método. Caso contrário, incrementamos o tempo de ciclo em uma unidade e repetimos o processo até encontramos uma solução viável ou atingirmos um limitante superior para o tempo de ciclo. Na modificação proposta, entretanto, em vez de decidir pela tarefa a ser alocada de acordo com a heurística selecionada, utilizamos uma lista com as β melhores tarefas, sendo que o valor destas pode estar, no máximo, a uma razão α do valor da melhor tarefa de acordo com a heurística. Isto é, se $l[i]$ é o valor da tarefa i e l_{min} é o valor da melhor tarefa, o método seleciona uma tarefa aleatória j tal que $l[j] < l_{min} * (1 + \alpha)$.

De modo a aumentar a variabilidade das configurações encontradas pelo método, nós também modificamos as heurísticas construtivas de modo a aumentar a probabilidade de novas

alocações tarefa-trabalhador. Se a tarefa j foi alocada a um trabalhador w em k configurações, nós multiplicamos $l[j]$ por $(1 + k)$ antes de decidir quais tarefas são candidatas a serem escolhidas pelo método. Isto evita que a mesma alocação seja usada em muitas configurações, aumentando a variabilidade do *pool*.

3.2. Busca local

Para cada configuração gerada pelo GRASP nós aplicamos dois procedimentos de busca local. A vizinhança em ambos os casos corresponde a mover uma tarefa a um estágio adjacente. O primeiro procedimento seleciona o movimento que resulta no menor tempo de ciclo e termina caso nenhum movimento válido reduza o tempo de ciclo. Já o segundo movimento visa aumentar ainda mais a variabilidade das configurações e, para isso, em cada iteração, movemos uma tarefa para um trabalhador adjacente que não executa esta tarefa em nenhuma outra configuração. Caso mais de uma tarefa e/ou trabalhador sejam candidatos, selecionamos o movimento que resulte no menor tempo de ciclo. Todas as configurações geradas durante a busca local são adicionadas ao *pool*.

3.3. Modelo linear inteiro misto

O último passo do método consiste na resolução de um modelo linear inteiro misto para encontrar a melhor solução que utilize as configurações presentes no *pool*, aos moldes do apresentado em (Moreira e Costa; 2012). Usando a notação:

w	Índice dos trabalhadores
i, j	Índices das tarefa
r	Índice das configurações
W	Conjunto de trabalhadores
N	Conjunto de tarefas
P	Conjunto de períodos
R	Conjunto de configurações
F_r	Taxa de produção da configuração r
\underline{F}	Número mínimo de itens a serem processados
q_{rwi}	Igual a um se a tarefa i é alocada ao trabalhador w na configuração r e igual a zero caso contrário

E as variáveis:

z_{wi}	Variável binária. Igual a um se o trabalhador w executa a tarefa i em pelo menos um período
u_r	Variável inteira. Corresponde ao número de vezes que a configuração u pertence à solução

Podemos escrever o seguinte modelo para o problema de rotação de tarefas com estações em paralelo::

$$Max \sum_{w \in W} \sum_{i \in N} z_{wi} \quad (18)$$

$$s.a. \sum_{r \in R} F_r * u_r \geq \underline{F} \quad (19)$$

$$z_{wi} \leq \sum_{r \in R} q_{rwi} * u_r \quad \forall w \in W, \forall i \in N \quad (20)$$

$$\sum_{r \in R} u_r = |P| \quad (21)$$

A função objetivo (18) maximiza o número de alocações tarefa-trabalhador distintas. A restrição (19) garante a produtividade mínima. As restrições (20) calculam o número de alocações

tarefa-trabalhador de acordo com as configurações selecionadas. Finalmente, as restrições (21) determinam que o número total de configurações selecionadas corresponde ao número de períodos.

Embora o número de configurações no *pool* possa ser bastante elevado, o que dificultaria a resolução deste modelo, podemos realizar um pré-processamento para eliminar configurações indesejáveis (cujo tempo de ciclo seja muito alto). Para isto, consideremos a restrição (19). Para que uma configuração i pertença à solução, temos que $F_i + \sum_{r \in R} F_r * u_r \geq \underline{F}$ e $\sum_{r \in R|i} u_r + 1 = |P|$. Seja r^* a configuração de menor tempo de ciclo no *pool*. A solução que permite a maior folga para F_i corresponde a selecionar r^* para todos os outros períodos. Assim, temos que $F_i + F_{r^*}(|P| - 1) \geq \underline{F}$. Logo:

$$F_i \geq \underline{F} - F_{r^*}(|P| - 1) \quad (22)$$

Esta fórmula (22) permite eliminar do *pool* configurações cujo tempo de ciclo seja muito alto, com base na taxa de produção desejada.

4. Medida de qualidade

Nesta seção definimos uma medida de qualidade de uma solução para o problema de rotação de tarefas. Esta medida leva em consideração os limitantes do número de alocações tarefa-trabalhador para o problema serial, que dependem do número de tarefas, trabalhadores e períodos.

Seja z o número de alocações tarefa-trabalhador distintas em uma dada solução com T períodos. Sejam z_{min} e z_{max} os limitantes inferior e superior de z , respectivamente. Definimos a qualidade Q de uma solução através da seguinte expressão:

$$Q = \frac{z - z_{min}}{z_{max} - z_{min}} \quad (23)$$

Para cada configuração serial da linha de produção, temos que $z = |N|$. Desta forma, a pior solução possível corresponde a duplicar a mesma configuração em todos os períodos. Portanto, $z_{min} = |N|$. Por outro lado, se relaxarmos a restrição referente à produtividade mínima desejada, se desconsiderarmos as tarefas ineficazes, na melhor solução possível para $|P| \leq |W|$ cada tarefa é executada por um trabalhador diferente em cada período. Com isto, $z_{max} = |P| * |N|$ para $|P| \leq |W|$. Uma vez que o número total de alocações possíveis corresponde a $|W| * |N|$ (desconsiderando as tarefas ineficazes), temos que $z_{max} = \min(|P|, |W|) * |N|$. Portanto, a medida de qualidade (23) corresponde a:

$$Q = \frac{z - |N|}{\min(|P|, |W|) * |N| - |N|} = \frac{z - |N|}{(\min(|P|, |W|) - 1) * |N|} \quad (24)$$

Para $|P| \geq 2$, temos que $Q = 0\%$ se $z = z_{min}$ e $Q = 100\%$ se $z = z_{max}$. Observe que, uma vez que no caso com estações em paralelo as tarefas são duplicadas em cada estação, é possível que uma solução utilizando paralelismo possua $z > z_{max}$ e, portanto, $Q > 100\%$. Isto significa que esta solução possui mais alocações trabalhador-tarefa distintas do que qualquer solução serial possível, independente da restrição de produtividade mínima. Conforme será visto na próxima seção, esta é uma situação que pode ocorrer mesmo quando altos níveis de produtividade são exigidos.

Observe que esta medida não leva em consideração as tarefas ineficazes para cada trabalhador. De fato, o cálculo do valor exato de z_{max} para $|P| < |W|$ requer a análise do grafo de precedência e das tarefas ineficazes. No entanto, isto tornaria a medida muito complexa, o que vai de encontro à nossa proposta de apresentar uma medida simples, de fácil aplicação e que permita a comparação dos resultados entre diferentes métodos e extensões.

5. Resultados computacionais

Nesta seção apresentamos os resultados computacionais para o método desenvolvido. Nós utilizamos quatro famílias de instâncias (Roszieg, Heskia, Tonge e Wee-Mag) desenvolvidas por Chaves et al. (2007), totalizando 320 instâncias. A tabela 1 indica as características destas instâncias, como o número de tarefas, número de trabalhadores e a Order Strength (uma medida das propriedades estruturais do grafo de precedência). O método foi implementado em C++ e testado em um computador com processador Intel Core 2 Duo T5450, 1,66 GHz e 3 GB de RAM. Para a resolução dos modelos, utilizamos o pacote comercial IBM ILOG CPLEX 12.5.

Tabela 1: Características das instâncias

	Tarefas	Trabalhadores	Order Strength
Heskia	28	4 e 7	22,49
Roszieg	25	4 e 6	71,67
Wee-Mag	75	11 e 19	22,67
Tonge	70	10 e 17	59,42

Para o modelo apresentado na seção 2, selecionamos inicialmente 80 instâncias das famílias Roszieg e Heskia e limitamos o tempo de execução em uma hora. O modelo se mostrou incapaz de encontrar uma solução viável para estas instâncias, o que nos levou a abandoná-lo rapidamente e iniciar o desenvolvimento de um método híbrido.

Para o método descrito na seção 3 geramos um *pool* contendo as configurações encontradas pelo GRASP após 1000 iterações, com $\alpha = 150\%$ e $\beta = 6$. Estes valores foram fixados após testes preliminares com $\alpha = [25\%, 50\%, 100\%, 150\%, 200\%]$ e $\beta = [3..10]$. Além disso, adicionamos ao *pool* as soluções encontradas pelas heurísticas construtivas descritas por Araújo et al. (2010a) e as soluções da última iteração do algoritmo genético desenvolvido por Moreira et al. (2012).

A tabela 2 apresenta os resultados encontrados para a heurística híbrida com dois períodos, enquanto que a tabela 3 apresenta os mesmos resultados com o número de períodos igual ao número de trabalhadores. Os resultados da tabela 3 para as famílias Tonge e Wee-Mag correspondem aos testes realizados com as 40 primeiras instâncias de cada família. Para a produtividade mínima F utilizamos a taxa de produção de uma das configurações multiplicada pelo número de períodos e por alfa, cujos possíveis valores variam entre 1,05, 1,1, 1,25 e 1,5. Para cada instância realizamos dois testes: o primeiro usa como base a taxa de produção da melhor configuração conhecida enquanto que o segundo utiliza a melhor configuração encontrada pelo GRASP. Na coluna GRASP indicamos, para cada família de instâncias, a diferença entre o tempo de ciclo da melhor solução encontrada pelo GRASP e da melhor solução conhecida, em média. As colunas Q, T. Par. e T. Exec correspondem à média da medida de qualidade, ao número médio de tarefas em paralelo e ao tempo de execução médio do método em segundos, respectivamente.

Para as famílias Roszieg e Heskia o método conseguiu resolver a maior parte das instâncias, encontrando poucas dificuldades mesmo com $\alpha=1,05$. De fato, usando como base a melhor configuração conhecida, encontramos 12 soluções com $Q>100\%$. Isto quer dizer que estas soluções são melhores que qualquer solução viável para o ALWABP com rotação de tarefas, independente do valor de alfa. O tempo de execução foi bastante curto, raramente ultrapassando um minuto.

O método encontrou dificuldades para resolver as instâncias das famílias Tonge e Wee-Mag. Isto se deve principalmente devido à baixa qualidade das configurações encontradas para estas instâncias, como pode ser visto na coluna GRASP. Com $\alpha=1,05$, usando a melhor configuração conhecida como base e com dois períodos, o método foi capaz de resolver apenas cinco instâncias da família Tonge e uma da família Wee-Mag. Embora quatro dessas soluções sejam triviais (isto é, correspondem à mesma configuração em ambos os períodos e, portanto, possuem $Q=0\%$),

Tabela 2: Resultados para o método de solução para $IPI=2$

Família	GRASP	Alfa	Melhor solução conhecida			Melhor solução GRASP		
			Q	T. Par.	T. Exec.	Q	T. Par.	T. Exec.
roszieg	1,32%	1,05	59,90%	4,4	0	83,05%	6,4	0
		1,10	96,70%	9,0	0	111,55%	11,0	0
		1,25	140,50%	16,7	0	145,50%	17,5	0
		1,50	156,10%	19,5	0	157,65%	19,6	0
heskia	1,90%	1,05	51,56%	4,4	5	76,65%	5,3	5
		1,10	86,65%	7,8	5	103,62%	9,9	5
		1,25	140,40%	20,0	5	151,29%	22,7	5
		1,50	173,44%	28,2	5	177,32%	29,7	5
tonge	14,87%	1,05	2,89%	0,5	120	87,25%	9,5	128
		1,10	7,64%	0,8	121	114,88%	23,0	127
		1,25	85,82%	16,5	124	164,82%	55,7	132
		1,50	169,86%	59,1	132	207,34%	84,1	133
weemag	19,91%	1,05	0,00%	0,0	100	77,37%	7,6	108
		1,10	0,12%	0,1	101	96,62%	15,2	107
		1,25	54,73%	6,3	110	144,93%	49,5	111
		1,50	144,15%	48,3	128	191,45%	78,9	112

Tabela 3: Resultados para o método de solução para $IPI=|W|$

Família	GRASP	Alfa	Melhor solução conhecida			Melhor solução GRASP		
			Q	T. Par.	T. Exec.	Q	T. Par.	T. Exec.
roszieg	1,32%	1,05	31,29%	6,7	0	41,42%	9,0	0
		1,10	46,45%	11,5	0	52,81%	13,7	0
		1,25	61,49%	16,0	0	62,77%	16,5	0
		1,50	66,93%	17,8	0	67,14%	17,6	0
heskia	1,90%	1,05	25,14%	7,7	5	39,38%	9,1	5
		1,10	43,37%	13,2	5	52,62%	17,7	5
		1,25	65,23%	25,5	5	68,11%	27,5	5
		1,50	73,50%	31,3	7	74,17%	32,2	7
tonge	14,87%	1,05	1,23%	2,0	119	31,06%	37,8	126
		1,10	3,69%	4,7	119	42,14%	73,4	129
		1,25	32,48%	46,6	165	57,06%	122,1	487
		1,50	57,69%	122,8	495	63,23%	152,9	1349
weemag	19,91%	1,05	0,00%	0,0	97	24,68%	36,6	104
		1,10	0,58%	0,9	97	34,42%	83,8	107
		1,25	18,69%	38,3	115	50,25%	148,8	466
		1,50	49,34%	138,9	524	60,78%	184,7	1327

duas soluções encontradas são de alta qualidade. Ambas as soluções são da família Tonge, com $Q=95,71\%$ e $Q=135,71\%$, respectivamente. Isto somado ao fato de que as soluções que utilizam a melhor solução do GRASP como base foram de alta qualidade para as quatro famílias de instâncias nos leva a crer que melhorias no *pool* podem levar a soluções ainda melhores para as famílias Tonge e Wee-Mag. Para $\alpha > 1,25$ o método conseguiu encontrar soluções de melhor qualidade em ambas as famílias. O tempo de execução foi relativamente curto, nunca chegando a 8 minutos.

Outro fator importante a ser observado é que para os testes com $|P|=2$ o aumento no valor de α teve pouco impacto no tempo de execução do método, enquanto que para $|P|=|W|$ ele cresce rapidamente, especialmente para as famílias Tonge e Wee-Mag. Isto se deve às configurações rejeitadas pela equação 22. Uma vez que estas famílias possuem o maior número de trabalhadores (10 e 11, respectivamente) poucas configurações são rejeitadas, o que aumenta o tempo de execução do modelo linear. De fato, para $|P|=2$, a execução do modelo corresponde a menos de 10% do tempo total de execução do método, enquanto que para $|P|=|W|$ ele pode chegar a 60% do mesmo.

6. Conclusões

Neste trabalho estudamos o ALWABP com rotação de tarefas e estações em paralelo. Apresentamos um modelo linear e um método híbrido para o problema. Testes com instâncias disponíveis na literatura mostraram que o método é eficiente para instâncias de pequeno e médio porte. Para instâncias maiores, o método encontrou dificuldades em casos com limitantes apertados, mas ainda assim conseguiu obter resultados de alta qualidade em algumas situações. De maneira geral, estes resultados indicam que configurações em paralelo, se consideradas durante o balanceamento de linhas de produção com rotação de tarefas, podem levar a ganhos significativos na diversidade das tarefas executadas pelos trabalhadores.

Os próximos passos dessa linha de pesquisa consistem no estudo do problema de rotação de tarefas com linhas em paralelo. Nesta variação do problema os trabalhadores são divididos em grupos e alocados a linhas de produção independentes. Araújo et al. (2014) mostram que esta abordagem pode levar a configurações com alocações tarefa-trabalhador bastante distintas daquelas encontradas no caso serial. Acreditamos que isto irá enriquecer ainda mais o método, gerando soluções de alta qualidade.

Agradecimentos

Agradecemos à CAPES-Brasil, FAPESP-Brasil e MEC-Espanha financiadoras deste projeto através dos processos CAPES DGU 258-12 / PHB2011-0012-PC e 2011/00803-0. Agradecemos também ao Dr. Marcus Ritt, por gentilmente ter fornecido as soluções do seu algoritmo genético para as instâncias das famílias Roszieg, Heskia, Tonge e Wee-Mag.

Referências

- Araújo, F. F. B., Costa, A. M. e Miralles, C. (2012). Two extensions for the assembly line worker assignment and balancing problem: parallel stations and collaborative approach, *International Journal of Production Economics* **140**: 483–495.
- Araújo, F. F. B., Costa, A. M. e Miralles, C. (2010a). Balanceando linhas de produção com trabalhadores deficientes e estações paralelas, *V International conference on production research - Americas*.
- Araújo, F. F. B., Costa, A. M. e Miralles, C. (2010b). Extensão do problema alwabp: melhorando a flexibilidade através de estações paralelas, *XLII Simpósio Brasileiro de Pesquisa Operacional*.
- Araújo, F. F. B., Costa, A. M. e Miralles, C. (2014). Balancing parallel assembly lines with disabled workers.
URL: arxiv.org/abs/1304.1423



- Azizi, N., Zolfaghari, S. e Liang, M. (2010). Modeling job rotation in manufacturing systems: The study of employee's boredom and skill variations, *International Journal of Production Economics* **123**: 69–85.
- Baybars, I. (1986). A survey of exact algorithms for the simple assembly line balancing problem, *Management Science* **32**: 909–932.
- Becker, C. e Scholl, A. (2006). A survey on problems and methods in generalized assembly line balancing, *European Journal of Operational Research* **168**: 694–715.
- Blum, C. (2010). Iterative beam search for simple assembly line balancing with a fixed number of work stations, *SORT* **35**: 145–164.
- Chaves, A., Lorena, L. e Miralles, C. (2009). Hybrid Metaheuristic for the Assembly Line Worker Assignment and Balancing Problem, in M. Blesa, C. Blum, L. Gaspero, A. Roli, M. Sampels e A. Schaerf (eds), *Hybrid Metaheuristics*, Vol. 5818 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 1–14.
- Chaves, A., Miralles, C. e Lorena, L. (2007). Clustering search approach for the assembly line worker assignment and balancing problem, *Proceedings of ICC&IE*, pp. 1469–1478.
- Costa, A. M. e Miralles, C. (2009). Job rotation in assembly lines employing disabled workers, *International Journal of Production Economics* **120**: 625–632.
- Miralles, C., Garcia-Sabater, J. P., Andres, C. e Cardos, M. (2007). Advantages of assembly lines in Sheltered Work Centres for Disabled. A case study, *International Journal of Production Economics* **110**: 187–197.
- Moreira, M. C. O. e Costa, A. M. (2009). A minimalist yet efficient tabu search for balancing assembly lines with disabled workers, *Anais do XLI Simposio Brasileiro de Pesquisa Operacional*, Porto Seguro, Brazil, pp. 660–671.
- Moreira, M. C. O. e Costa, A. M. (2012). Hybrid Heuristics for planning job rotation in Assembly Lines with disabled workers, *International Journal of Production Economics* **141**: 552–560.
- Moreira, M. C. O., Ritt, M., Costa, A. M. e Chaves, A. A. (2012). Simple heuristics for the assembly line worker assignment and balancing problem, *Journal of Heuristics* **18**: 505–524.
- Mutlu, O., Polat, O. e Supciller, A. A. (2013). An iterative genetic algorithm for the assembly line worker assignment and balancing problem of type-II, *Computers & Operations Research* **40**: 418–426.
- Otto, A. e Scholl, A. (2012). Reducing ergonomic risks at mixed-model assembly lines by job rotation scheduling, *OR Spectrum*.