

## **UMA METAHEURÍSTICA HÍBRICA APLICADA AO PROBLEMA DE BALANCEAMENTO E DESIGNAÇÃO DE TRABALHADORES EM LINHA DE PRODUÇÃO**

**Antonio Augusto Chaves**

Instituto Nacional de Pesquisas Espaciais (INPE)  
Av. dos Astronautas, 1758 São José dos Campos – SP – Brasil  
chaves@lac.inpe.br

**Cristóbal Miralles**

ROGLE-Dpto. Organización de Empresas. Universidad Politécnica de Valencia  
Camí Vera s/n, 46022, Valencia – Espanha  
cmiralles@omp.upv.es

**Luiz Antonio Nogueira Lorena**

Instituto Nacional de Pesquisas Espaciais (INPE)  
Av. dos Astronautas, 1758 São José dos Campos – SP – Brasil  
lorena@lac.inpe.br

### **RESUMO**

O Problema de Balanceamento e Designação de Trabalhadores em Linha de Produção (ALWABP) consiste em otimizar a alocação de trabalhadores e tarefas às estações de trabalho em uma linha de produção com pessoas deficientes, objetivando aumentar a produtividade de tais linhas e tornar as deficiências individuais tão transparente quanto possível. Este trabalho propõe uma nova abordagem para resolver o ALWABP, utilizando uma metaheurística híbrida denominada *Clustering Search* (CS). A validação das soluções obtidas será através da comparação com os resultados encontrados pelo CPLEX.

**PALAVRAS CHAVE.** Problema de Balanceamento de linha de Produção, *Clustering Search*, Metaheurística.

**Área principal (Metaheurística)**

### **ABSTRACT**

The Assembly Line Worker Assignment and Balancing Problem (ALWABP) consists of optimizing the allocation of workers and tasks to stations in an assembly line with disabled people. The goal is to increase the productivity of the assembly line but always respecting the worker limitations; trying to make transparent the individual disabilities. This paper proposes a new approach called *Clustering Search* (CS) to solve the ALWABP using hybrid metaheuristics. The validation of the obtained solutions will be through the comparison with the results found by the CPLEX.

**KEYWORDS.** Assembly Line Balancing Problem. Metaheuristic. *Clustering Search*.

**Main area (Metaheuristic)**

## 1. Introdução

A Organização Mundial de Saúde (OMS) estima que 10% da população mundial, cerca de 610 milhões de pessoas, são portadores de algum tipo de deficiência. Destas, mais da metade estão na idade ativa de trabalho e sofrem com altas taxas de desemprego (variando de 13% no Reino Unido a 80% em países em desenvolvimento).

Práticas atuais para o tratamento de indivíduos com deficiências físicas e mentais prescrevem atividades de trabalho, objetivando que estes indivíduos tenham uma vida mais longa e também uma maior integração destes cidadãos à sociedade (Belamy et al., 1979). Como forma de facilitar a inclusão destes trabalhadores no mercado de trabalho, alguns países adotaram a estratégia de criar Centros de Trabalho para Deficientes (CTD's). Estes centros funcionam como uma primeira etapa na integração destas pessoas que, eventualmente, serão absorvidas pelo mercado *normal* de trabalho.

Este modelo de integração sócio-trabalho procura afastar-se do estereótipo tradicional que considera as pessoas deficientes como incapazes de desenvolver um trabalho profissional contínuo. Da mesma maneira que em qualquer outra empresa, um CTD compete em mercados reais e precisa ser suficientemente flexível e eficiente para adaptar-se às variações do mercado, sendo que, a única diferença é que um CTD é uma organização sem fins lucrativos.

Sendo assim, o benefício que geralmente pode ser obtido com a melhora de eficiência é o crescimento do CTD, ou seja, mais trabalhos para as pessoas deficientes, gradualmente integrando pessoas com níveis mais elevados de inaptidão, o que é de fato o objetivo principal de todo CTD.

Em Miralles et al. (2007) é mostrado como a utilização de linhas de produção nestes centros provê muitas vantagens, sendo que, a divisão tradicional do trabalho em tarefas únicas pode se tornar uma ferramenta perfeita para fazer invisíveis determinadas deficiências do trabalhador. Além disso, uma atribuição apropriada da tarefa também pode transformar-se em um bom método terapêutico para reabilitação das deficiências. Mas algumas restrições específicas relacionadas à variabilidade dos tempos surgem neste ambiente, e então o procedimento de balanceamento aplicado neste ambiente deveria conciliar os seguintes objetivos (que não devem ser vistos como contraditórios, mas complementares):

- (1) Maximizar a eficiência da linha de produção balanceando a carga de trabalho atribuída a cada trabalhador disponível em cada estação;
- (2) Satisfazer e respeitar as restrições existentes neste ambiente devido aos fatores humanos ao atribuir tarefas aos trabalhadores.

Após analisar alguns CTD's, Miralles et al. (2007) estabeleceram algumas características específicas que podem ser encontradas nesses ambientes, sendo elas:

- Existe geralmente uma grande diferença entre os tempos médios de operação para cada tarefa dependendo de qual trabalhador a executa;
- Em muitos casos o tempo da tarefa não é apenas alto, mas algumas tarefas específicas podem ser impossíveis de ser realizadas por alguns trabalhadores (isso é bastante comum ao falar sobre deficiências físicas ou sensoriais);
- A própria variabilidade de tempos de operação para um trabalhador deficiente é mais elevado do que o usual. Normalmente não há trabalhadores mais lentos ou mais rápidos. Ao invés disso, trabalhadores podem ser muito lentos, ou até mesmo incapazes, ao executar certas tarefas, mas muito eficientes ao desenvolver outras.
- O objetivo principal dos CTD's é promover um ambiente de trabalho que ajude os trabalhadores com deficiência ter uma evolução positiva e constante de suas próprias capacidades, a fim de integrá-los o mais rápido possível aos centros reais de trabalho. Assim, é comum que muitos trabalhadores deixem o CTD quando alcançam seus melhores rendimentos. O CTD deve então substituí-los por novos trabalhadores, o que, dependendo de suas deficiências, pode acarretar em mudanças nas atribuições de tarefas existentes.

- O absentismo (falta ao trabalho) também é muito comum nestes ambientes, uma vez que trabalhadores deficientes enfrentam mais problemas de saúde.
- Há também um suporte psicológico periódico que é obrigatório no CTD.

Estas características expostas serviram de motivação para definir um novo problema de linha de produção, chamado Problema de Balanceamento e Designação de Trabalhadores em Linhas de Produção (ALWABP, do inglês: *Assembly Line Worker Assignment and Balancing Problem*), o qual foi apresentado e modelado matematicamente em Miralles et al. (2008) incluindo a maioria das características descritas anteriormente.

O ALWABP pode ser classificado como um problema NP-difícil, pois este é uma generalização do Problema Simples de Balanceamento da Linha de Produção (SALB, do inglês: *Simple Assembly Line Balancing Problem*), onde cada tarefa possui um tempo de execução fixo independente do trabalhador que a executa. Sendo o SALBP classificado como NP-difícil (Scholl e Becker, 2006). Esta característica justifica a aplicação de metaheurísticas ao ALWABP visando conseguir bons resultados em um tempo computacional competitivo.

Assim sendo, propõe-se utilizar neste trabalho a metaheurística híbrida *Clustering Search* (CS) para resolver de forma aproximada o ALWABP. O CS consiste em detectar dinamicamente regiões promissoras de busca baseando-se na frequência em que são amostradas nestas regiões as soluções geradas por uma metaheurística. Estas regiões promissoras devem ser exploradas tão logo sejam descobertas, através de métodos de busca local. O *software* CPLEX é utilizado para resolver as formulações matemáticas do ALWABP, objetivando validar os resultados computacionais obtidos pelo CS.

O restante do trabalho está organizado da seguinte forma. A seção 2 faz uma descrição do ALWABP e apresenta uma formulação matemática para este. Na seção 3 tem-se uma revisão bibliográfica deste problema. A seção 4 descreve o CS, e a seção 5 apresenta, em detalhes, o CS aplicado ao ALWABP. A seção 6 apresenta os resultados computacionais e na seção 7 são descritas algumas conclusões a respeito desse trabalho.

## 2. Definição e Formulação Matemática do ALWABP

De uma forma geral, uma linha de produção consiste de um conjunto finito de tarefas, cada uma tendo um tempo de execução, e um conjunto de relações de precedência, as quais especificam a ordem de execução das tarefas. O problema do ALWABP é atribuir as tarefas para uma seqüência de estações, tal que as relações de precedências sejam satisfeitas e alguma medida de eficiência seja otimizada. Entretanto, como nos CTD alguns trabalhadores podem ser muito lentos, ou até incapazes, de executar certas tarefas, mas muito rápidos em outras, o problema não consiste apenas em atribuir tarefas para estações, mas também trabalhadores disponíveis para estações, sempre respeitando as incompatibilidades quando atribuir tarefas para trabalhadores.

As principais características do ALWABP são descritas a seguir:

- Os tempos de processamento das tarefas e as relações de precedências são determinísticos.
- A linha produzirá um único modelo de um único produto.
- É definida uma linha serial onde áreas de armazenagem (*buffers*) não são consideradas.
- Existe um certo número de trabalhadores disponíveis, sendo que o tempo de processamento das tarefas pode ser diferente dependendo de qual dos trabalhadores executa a tarefa (uma vez que os trabalhadores têm diferentes habilidades e capacidades).
- Não existem trabalhadores lentos ou rápidos. Ao invés disso, trabalhadores podem ser muito lentos, ou até incapazes, de executar algumas tarefas, mas muito eficientes quando executam outras tarefas.
- Todo trabalhador é atribuído para somente uma estação de trabalho.
- Toda tarefa é atribuída para somente uma estação de trabalho, contanto que o trabalhador selecionado para aquela estação seja capaz de realizar a tarefa, e que as relações de precedência sejam satisfeitas.

Quanto aos objetivos, os problemas de balanceamento de linha de produção são usualmente divididos em 2 tipos:

- (a) ALWABP-1: se o objetivo é minimizar o número de estações de trabalho, para uma dada taxa de produção (ou um dado tempo de ciclo).
- (b) ALWABP-2: se o objetivo é maximizar a taxa de produção (ou minimizar o tempo de ciclo) para um dado número de estações.

Devido ao fato dos CTD's não terem fins lucrativos, o objetivo mais apropriado para estes é o ALWABP-2, uma vez que este visa o aumento da eficiência da produção sem retirar nenhum posto de trabalho existente.

A formulação matemática para o ALWABP, proposta por Miralles et al. (2008), leva em consideração a seguinte notação:  $x_{shi}$  é uma variável binária igual a 1 somente se a tarefa  $i$  é atribuída para o trabalhador  $h$  na estação  $s$ , e  $y_{sh}$  é uma variável binária igual a 1 somente quando o trabalhador  $h$  é atribuído para a estação  $s$ . Sendo,  $t_{hi}$  o tempo de operação para o trabalhador  $h$  executar a tarefa  $i$  e  $D_j$  o conjunto de tarefas que precedem imediatamente a tarefa  $j$  na rede de precedência. O tempo de ciclo ( $C_{tempo}$ ) é a variável de decisão que representa a quantidade de tempo que um produto gasta para ser processado por uma estação da linha de produção, tendo relação direta com a taxa de produção da linha.

$$\text{Min } z = C_{tempo} \quad (1)$$

sujeito a

$$\sum_{h \in H} \sum_{s \in S} x_{shi} = 1; \quad \forall i \in N \quad (2)$$

$$\sum_{s \in S} y_{sh} \leq 1; \quad \forall h \in H \quad (3)$$

$$\sum_{h \in H} y_{sh} \leq 1; \quad \forall s \in S \quad (4)$$

$$\sum_{h \in H} \sum_{s \in S} s \cdot x_{shi} \leq \sum_{h \in H} \sum_{s \in S} s \cdot x_{shj} \quad \forall i, j / i \in D_j \quad (5)$$

$$\sum_{i \in N} t_{hi} \cdot x_{shi} \leq C_{tempo} \quad \forall h \in H; \forall s \in S \quad (6)$$

$$\sum_{i \in N} x_{shi} \leq P \cdot y_{sh} \quad \forall h \in H; \forall s \in S \quad (7)$$

$$y_{sh} \in [0,1] \forall s \in S, h \in H \quad (8)$$

$$x_{shi} \in [0,1] \forall s \in S, h \in H, i \in N \quad (9)$$

A função objetivo do ALWABP-2 (1) é minimizar o tempo de ciclo  $C_{tempo}$ . As restrições (2) garantem que toda tarefa  $i$  é atribuída para apenas uma estação  $s$  e um trabalhador  $h$ . As restrições (3) e (4) asseguram que todo trabalhador pode ser atribuído para somente uma única estação, e que em toda estação existe somente um trabalhador. O conjunto de restrições (5) reflete as relações de precedência entre as tarefas  $i$  e  $j$ , onde  $i$  é predecessor de  $j$ . As restrições (6) e (7) implicam que todo trabalhador  $h$  atribuído para estação  $s$  pode ter mais que uma tarefa, desde que, o tempo de ciclo  $C_{tempo}$  não seja extrapolado. Como  $C_{tempo}$  e  $y_{sh}$  são ambas variáveis, as restrições (6) e (7) são definidas separadamente para que o modelo seja mantido linear. A constante  $P$  precisa ter um valor maior que o somatório de todos os tempos de processamento. As restrições (8) e (9) definem as variáveis  $y_{sh}$  e  $x_{shi}$  como binárias.

### 3. Revisão Bibliográfica

Ao nosso conhecimento, apenas muito recentemente o problema encontrado nos CTD's começou a ser tratado na literatura. A maioria dos problemas sobre linha de produção encontrados na literatura trabalha com tempos de operação fixos, independentemente de qual trabalhador executa a tarefa. Esta simplificação é justificada devido ao fato de que, na maioria dos casos de linhas de produção reais, a variação de tempo é muito pequena.

Miralles et al. (2007) realizaram estudos sobre os CTD's, observando que a utilização

de linhas de produção nestes centros provê várias vantagens, sendo assim, introduziram o ALWABP e apresentaram uma formulação matemática para este. Miralles et al. (2008) desenvolveram um algoritmo *Branch e Bound* com diferentes estratégias de busca para o ALWABP. Os autores realizaram um estudo de caso para o ALWABP-2 em um CTD espanhol utilizando este algoritmo.

Chaves, Miralles e Lorena (2007) realizaram estudos preliminares sobre a utilização da metaheurística híbrida *Clustering Search* (CS) aplicado ao ALWABP. Os resultados apresentados foram bons ao combinar a metaheurística *Simulated Annealing* com um processo de agrupamento de soluções, porém o CS apresentou algumas dificuldades nas instâncias maiores demonstrando que seria necessário algumas modificações neste para melhorar seu desempenho.

São poucos os trabalhos encontrados na literatura que consideram tempos diferentes de operação das tarefas dependendo de qual trabalhador executa a tarefa. Mansoor (1968) propôs uma heurística para resolução de uma linha de produção considerando níveis variáveis de performance dos trabalhadores. Doerr et al. (2000) estudaram o desenvolvimento de uma linha de produção considerando trabalhadores com diferentes habilidades e onde a hora-extra poderia ser usada quando a quota de produção diária não fosse alcançada. Hoop et al. (2001) e Gel et al. (2002) apresentaram casos no quais existem trabalhadores lentos e rápidos, independentemente de qual tarefas eles executam, porém as soluções partem de uma linha de produção balanceada e o objetivo é apenas ter uma rotação eficiente dos trabalhadores e não balancear a linha. Corominas et al. (2003) apresentaram um modelo para um problema de balanceamento de linha de produção que considera tempos variáveis de duração das tarefas dependendo de qual estação ela está alocada. Esta situação é encontrada em várias situações industriais, onde é necessário diferenciar tipos de trabalhadores, mas somente entre dois tipos: trabalhadores experientes e inexperientes.

#### 4. Clustering Search

A metaheurística híbrida *Clustering Search* (CS) é baseada no *Evolutionary Clustering Search* (ECS), proposto por Oliveira e Lorena (2007). O CS consiste em dividir o espaço de busca em regiões, ou seja, sub-espacos de busca definidos por uma relação de vizinhança, e por meio de um processo de agrupamento de soluções detectar regiões supostamente promissoras.

O processo de agrupamento é executado simultaneamente com uma metaheurística, identificando grupos de soluções que merecem especial interesse. As regiões destes grupos devem ser exploradas tão logo sejam detectadas, por meio de heurísticas de busca local específicas. Uma região se torna promissora quando o número de soluções geradas pela metaheurística e agrupadas nesta região ultrapassa um certo limitante, indicando que um certo padrão de soluções está sendo gerado pela metaheurística.

O objetivo do CS é aplicar os métodos de busca local somente nas regiões consideradas promissoras, diminuindo assim o esforço computacional, principalmente se comparado com metaheurísticas híbridas que aplicam busca local em todas as soluções. O CS visa também acrescentar um pouco de *inteligência* na escolha de quais soluções aplicar a busca local, não escolhendo aleatoriamente. Espera-se uma melhoria no processo de convergência associado a uma diminuição no esforço computacional em virtude do emprego mais racional dos métodos de busca local.

O CS procura localizar regiões promissoras através do enquadramento destas por *clusters*. Um *cluster* é formado pelo **centro**  $c_i$  e pelo **volume**  $v_i$ . O centro  $c_i$  é uma solução que representa o *cluster*  $i$ , identificando a sua localização dentro do espaço de busca. Ao invés de armazenar todas as soluções agrupadas no *cluster*, apenas parte das informações destas soluções são atribuídas ao centro do *cluster*. O volume  $v_i$  é a quantidade de soluções geradas pela metaheurística e agrupadas no *cluster*  $i$ . Este volume é o que determinará quando um *cluster* se torna promissor.

Inicialmente os *clusters* são gerados de forma a representarem o máximo possível o espaço de busca. Para tal, gera-se um conjunto  $N$  com  $n$  soluções aleatórias e objetiva-se selecionar um subconjunto  $M \subset N$  com  $m$  soluções que possuam a maior diversidade entre si. A

medida de diversidade entre duas soluções pode ser representada pela distancia entre elas. A distancia neste caso é calculada com base no conjunto de atributos das soluções de  $N$  através de uma métrica de distancia definida a priori. O valor de  $m$  é uma constante do CS, representando o número de *clusters* existentes.

O CS consiste em duas etapas que vão sendo executadas iterativamente. Na primeira etapa gera-se uma solução por meio de uma metaheurística. Na segunda etapa realiza-se o processo de agrupamento, onde a solução gerada é agrupada no *cluster* mais similar e caso este *cluster* seja considerado promissor, aplica-se uma heurística de busca local neste. O CS possui três componentes conceitualmente independentes e com diferentes atribuições:

- uma metaheurística (MH);
- um agrupador iterativo (AI);
- uma busca local (BL).

A Figura 1 apresenta o diagrama para a metaheurística híbrida *Clustering Search* (CS).

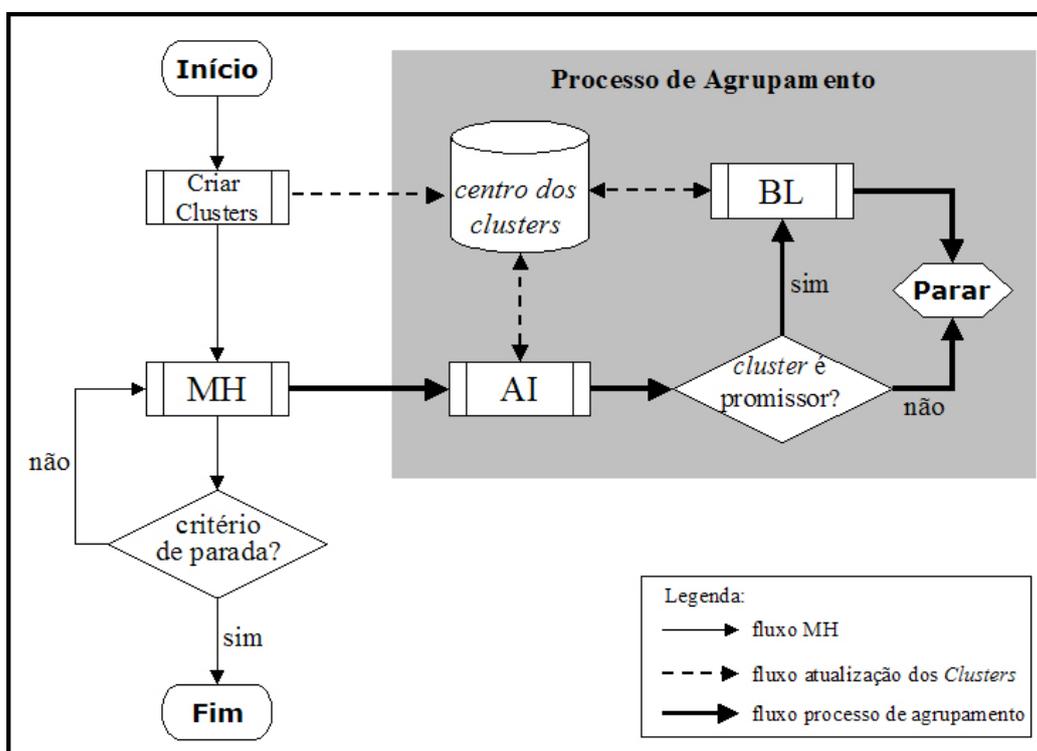


Figura 1 - Diagrama conceitual do CS

A metaheurística (MH) trabalha como um gerador de soluções de tempo integral. O algoritmo é executado independentemente dos componentes restantes e este precisa ser capaz de gerar soluções de forma contínua para o processo de agrupamento. Simultaneamente, *clusters* são mantidos para representar estas soluções.

O agrupador iterativo (AI) objetiva reunir soluções similares dentro dos *clusters*, mantendo uma solução no centro do *cluster* que seja representativa para as demais soluções. Os *clusters* são progressivamente alimentados por soluções geradas em cada iteração da MH. Uma métrica de distância também precisa ser definida inicialmente, permitindo uma medida de similaridade para o processo de agrupamento. Cada solução recebida pelo AI é agrupada no *cluster* que for mais similar a ela, causando uma perturbação no centro deste. Tal perturbação é chamada de assimilação e consiste basicamente em atualizar o centro com a nova solução.

Após o AI definir qual é o *cluster* mais similar e perturbar seu centro, é realizada uma análise do volume deste *cluster*. Caso o volume do *cluster* tenha atingido um certo limitante  $\lambda$ , este é considerado uma região de busca promissora. Sendo assim, tal *cluster* precisa ser melhor investigado, para acelerar o processo de convergência. Toda vez que um *cluster* for considerado promissor é necessário diminuir o seu volume, evitando que este *cluster* seja novamente

considerado promissor na próxima iteração em que uma solução seja agrupada nele.

Por fim, a busca local (BL) é um módulo de busca que provê a exploração de uma suposta região promissora, delimitada pelo *cluster*. Este processo acontece após ser descoberto um *cluster* promissor e uma busca local é aplicada no centro do *cluster*. BL pode ser considerado como uma heurística específica para o problema que será empregada dentro do *cluster*, analisando sua vizinhança.

### 5. CS aplicado ao ALWABP

Nesta seção será apresentada uma abordagem do problema ALWABP-2 por meio da metaheurística híbrida *Clustering Search* (CS). O componente MH, responsável pela geração de soluções para o processo de agrupamento, será a metaheurística *Simulated Annealing* (SA) (Kirkpatrick, 1983), a qual é capaz de gerar um grande número de soluções diferentes para este processo. Este e os demais componentes do CS serão explicados a seguir.

Uma solução do ALWABP será formada por dois vetores. O primeiro vetor representa a alocação tarefa/estação. O segundo vetor representa a alocação trabalhador/estação. A Figura 2 ilustra a representação de uma solução com 11 tarefas, 5 trabalhadores e 5 estações.

Tarefa	1	2	3	4	5	6	7	8	9	10	11
Estação	1	1	2	2	3	3	3	3	4	4	5
Trabalhador	1	2	3	4	5						
Estação	1	2	3	4	5						

Figura 2 - Representação de uma solução do ALWABP

#### 5.1 Geração dos Clusters Iniciais

Os clusters iniciais precisam representar adequadamente diversas regiões do espaço de busca. Sendo assim, é interessante que exista uma grande diversidade entre os clusters.

A geração dos clusters iniciais consiste em, a partir de um conjunto  $N$  com  $n$  soluções geradas aleatoriamente, selecionar um subconjunto  $M \subset N$  com  $m$  soluções que possuam a maior diversidade possível entre elas. A medida de diversidade entre duas soluções é representada pela distância ente elas. A distância  $d(i, j)$ , neste caso, é calculada com base no número de tarefas atribuídas para estações diferentes entre as soluções  $i$  e  $j$ .

Inicialmente é necessário montar uma matriz simétrica  $D[n \times n]$  que armazena os índices de diversidade entre cada uma das soluções de  $N$ , preenchida da seguinte maneira:  $d(i, j) = d(j, i)$  e  $d(i, i) = 0 \forall (i, j) \in N$ . O valor da diversidade de  $M$  é determinado pelo somatório das diversidades entre as soluções selecionadas para este conjunto.

A primeira solução do conjunto  $M$ ,  $m_1$ , é escolhida aleatoriamente entre todos os elementos pertencentes a  $N$ . A segunda solução a ser selecionada deve ser a solução  $j \in N - m_1$  que ofereça o maior índice de diversidade entre  $m_1$  e  $j$ .

A partir da terceira solução, devemos calcular  $D_{soma}(j), \forall j \in N - M_{c-1}$ , que é a soma das diversidades entre o candidato  $j$  e todas as soluções pertencentes ao conjunto parcial  $M_{c-1}$ . A seguir, a solução que tiver o maior valor de  $D_{soma}$  é selecionada para fazer parte do conjunto parcial. Este processo é repetido até que  $m$  soluções sejam selecionadas.

Os clusters iniciais serão as  $m$  soluções selecionadas para fazer parte do subconjunto  $M$ . Neste trabalho foram utilizados  $n = 300$  e  $m = 20$ .

#### 5.2 Simulated Annealing

A metaheurística *Simulated Annealing* (SA) (Kirkpatrick, 1983) será utilizada no componente MH, sendo esta responsável por gerar soluções para o processo de agrupamento.

O SA parte de uma solução inicial aleatória, a qual é obtida escolhendo aleatoriamente as alocações das tarefas para as estações e de trabalhadores disponíveis para estações. Os passos

seguintes seguem o esquema do algoritmo tradicional do SA. Dada uma temperatura  $T$ , o algoritmo aleatoriamente seleciona um dos movimentos para uma vizinhança e computa a variação da função objetivo. Se este melhorar a solução corrente o movimento é aceito, caso contrário, existe uma probabilidade de aceitação do movimento que é menor em temperaturas baixas.

Três movimentos diferentes podem ser definidos para compor tipos distintos de vizinhança, chamados  $N^1$ ,  $N^2$  e  $N^3$ , de uma solução.  $N^1$  é obtido pela troca de alocação de dois trabalhadores de diferentes estações.  $N^2$  pela troca de alocação de duas tarefas de diferentes estações.  $N^3$  pela retirada de uma tarefa de uma estação e adição em outra estação.

Os parâmetros de controle do SA são a razão de resfriamento  $\alpha$ , o número de iterações para cada temperatura ( $SA_{max}$ ) e a temperatura inicial  $T_0$ . Neste trabalho foi utilizado  $\alpha = 0,95$ ;  $SA_{max} = 1000$ ; e  $T_0 = 1000000$ .

### 5.3 Processo de Agrupamento

O componente AI é o núcleo do CS, trabalhando como um classificador, mantendo no sistema somente informações relevantes e direcionando a busca para regiões supostamente promissoras. Definidos os clusters iniciais, cada cluster possui seu próprio centro  $c_i$  e volume  $v_i$  igual a 1.

As soluções geradas pelo SA são passadas para o AI que procura agrupa-las como uma solução conhecida, de acordo com uma métrica (distância). O *cluster* que possuir a menor distância em relação à solução gerada pelo SA é considerado o *cluster* mais similar. A informação da solução gerada pelo SA deve causar uma perturbação (assimilação) no centro do *cluster* mais similar. A medida de distância utilizada neste trabalho foi o número de tarefas atribuídas para estações diferentes entre a solução SA e o centro do *cluster*, sendo que, quanto maior o número de tarefas atribuídas a diferentes estações entre as duas soluções, maior será a distância entre elas e menor será a similaridade.

No processo de assimilação será utilizado o método *path-relinking* (Glover, 1996), o qual realiza movimentos exploratórios na trajetória que interconecta o centro do *cluster* mais similar ( $c_i$ ) e a solução gerada pelo SA ( $s_k$ ). Assim sendo, o próprio processo de assimilação já é uma forma de busca local dentro do *cluster*, pois o centro vai ser deslocado para a melhor solução avaliada nessa trajetória.

*Path-relinking* começa de duas soluções, a primeira é o centro do *cluster* mais similar (inicial) e a segunda é a solução gerada pelo SA (guia). Partindo da solução inicial para a solução guia, caminhos são gerados e explorados na busca de soluções melhores. Para gerar estes caminhos, movimentos são selecionados através da troca da alocação da tarefa  $i$  na solução inicial pela alocação da tarefa  $i$  na solução guia, mudando a alocação tarefa/estação na solução. A busca continua a partir da melhor solução em um movimento.

Após agrupar a solução ao *cluster* mais similar e aplicar o processo de assimilação no centro deste, é necessário verificar se o *cluster* já pode ser considerado promissor. Um *cluster* se torna promissor quando o volume atinge um certo limitante  $\lambda$ . Caso o volume do *cluster* seja maior que esse limitante significa que um certo padrão de informação se tornou predominante no processo de busca, sendo o centro deste *cluster* explorado através do componente BL. Neste trabalho foi utilizado  $\lambda = 20$ .

O componente BL é ativado quando um suposto *cluster* promissor é identificado. No BL foram implementadas duas heurísticas, objetivando melhorar a solução presente no centro de um *cluster* promissor. A primeira heurística, chamada SWAP, consiste em examinar todas as possíveis trocas de duas tarefas que estão atribuídas a diferentes estações. A melhor troca é realizada e o processo é repetido novamente até que não ocorra mais melhora. A segunda heurística, chamada SHIFT, consiste em examinar todas as relações de tarefas e estações. A heurística remove uma tarefa de uma estação e insere esta em uma outra estação. O melhor movimento é realizado e todo o processo é repetido novamente até não ocorrer mais melhora. Caso BL encontre uma solução que seja melhor que o centro do *cluster*, este é atualizado com a solução encontrada.

A Figura 3 apresenta o pseudo-código da abordagem CS aplicada ao ALWABP.

```

1  procedimento CS ( $\alpha, T_0, SA_{max}, m, \lambda$ )
2      Criar os clusters iniciais
3      //componente MH
4      Criar uma solução inicial (sol)
5      IterT = 0
6      T =  $T_0$ 
7      enquanto ( $T > 0,0001$ ) faça
8          enquanto ( $IterT < SA_{max}$ ) faça
9              IterT = IterT + 1
10             Gerar um vizinho (sol') aleatoriamente na vizinhança  $N^k(sol)$ 
11              $\Delta = f(sol') - f(sol)$ 
12             se ( $\Delta < 0$ ) então
13                 sol = sol'
14             senão
15                 seja  $x \in [0,1]$ 
16                 se ( $x < e^{-\Delta T}$ ) então
17                     sol = sol'
18             fim-senão
19             //componente AI
20             Calcular a distância da solução para cada um dos clusters
21             Inserir a solução no cluster mais similar ( $j$ )  $\rightarrow v_j = v_j + 1$ 
22             Perturbar o centro do cluster  $j \rightarrow$  Path-Relinking ( $c_j, sol'$ )
23             //componente BL
24             se ( $v_j > \lambda$ ) então
25                  $c' =$  Busca Local ( $c_j$ )
26                 Diminuir o volume do cluster  $j$ 
27                 se ( $f(c') < f(c_j)$ ) então
28                      $c_j = c'$ 
29             fim-se
30         fim-enquanto
31     fim-procedimento

```

Figura 3 – Pseudo-código do CS aplicado ao ALWABP

## 6. Resultados Computacionais

O CS para ALWABP foi codificado em C++ e os testes computacionais executados em um Pentium IV 3 GHz com 512 MB de memória. Os experimentos foram realizados com objetivo de validar a abordagem proposta, mostrando que algoritmos de busca por agrupamentos podem ser competitivos para resolução do ALWABP.

Devido ao fato do ALWABP ser um problema inovador, não existe um conjunto de problemas-teste que possam ser usados para avaliar o CS. Portanto, Chaves, Miralles e Lorena (2007) geraram instâncias baseadas em uma coleção de problemas clássicos do SALBP (Hoffmann, 1990), disponíveis no site <http://www.assembly-line-balancing.de>.

Os problemas selecionados para fazer parte do conjunto de problemas-teste do ALWABP foram considerados de acordo com cinco fatores, sendo eles:

- Tam: número de tarefas;
- OS: característica da rede de precedência;
- NrT: número de trabalhadores;
- Var: variabilidade dos tempos de processamento (L1, H3);
- Inc: porcentagem de incompatibilidades tarefa/trabalhador (I10, I20);

Através destes fatores, os problemas selecionados foram: Heskia: (28 tarefas; OS = 22,49), Roszieg: (25 tarefas; OS = 71,67), Wee-mag: (75 tarefas; OS = 22,67) e Tonge: (70 tarefas; OS = 59,42). Para cada combinação possível foram geradas 10 instâncias, sendo assim, tem-se um conjunto de 80 instâncias para cada um dos problemas selecionados. Este conjunto de problemas-teste permite extrair conclusões sobre o comportamento global da metaheurística CS

nos diversos tipos de problema encontrados.

Para avaliar a qualidade das soluções encontradas pelo CS, executou-se o CPLEX versão 10.0.1 para cada uma das 320 instâncias. Devido ao fato do ALWABP ser um problema NP-difícil, há uma grande dificuldade para encontrar a solução ótima, especialmente para o conjunto de problemas maiores. Na maioria das instâncias de Wee-Mag e Tonge, o CPLEX terminou a execução devido ao estouro de memória do computador, sendo que, as soluções inteiras encontradas estavam com *gaps* acima de 90% entre o limitante inferior e superior. Em algumas instâncias, o CPLEX não consegue nem ao menos encontrar uma solução inteira.

As tabelas a seguir apresentam os resultados dos testes computacionais realizados neste trabalho. Devido ao fato do número elevado de instâncias, optou-se por apresentar os valores médios para cada uma das combinações dos problemas. As Tabelas 1 e 2 apresentam os seguintes dados: solução inteira (*SI*), tempo de execução (*Tempo*) e *gap* entre o limitante inferior e superior do CPLEX, e a melhor solução encontrada (*Sol\**), a solução média (*Sol*) e o tempo de execução (*Tempo*) do CS e do SA sem o processo de agrupamento. Para as instâncias Wee-Mag e Tonge é apresentada também a melhor solução não inteira do CPLEX (*SNI*).

A Tabela 1 apresenta os resultados para os problemas Roszieg e Heskia. Para as 160 instâncias presentes neste dois conjuntos de problemas, o CS encontrou a solução ótima em todos os casos, mostrando que este é um método interessante para resolução do ALWABP. O CPLEX consegue resolver todas as instâncias de Roszieg e Heskia, porém, o CPLEX é sensível em relação à variação do número de trabalhadores, uma vez que, o tamanho da matriz tarefa/trabalhador é aumentado e conseqüentemente a complexidade também. No CS esse efeito praticamente não é observado, sendo que, a variação de tempo é pequena em relação ao aumento do número de trabalhadores. O SA sem processo de agrupamento também obteve bons resultados, porém não encontrou todas as soluções ótimas e as soluções médias são piores se comparadas com as do CS.

Tabela 1 – Resultados Computacionais (problemas Roszieg e Heskia)

	Tam	NrT	Var	Inc	CPLEX			CS			SA			
					Sol*	Tempo	gap	Sol*	Sol	Tempo	Sol*	Sol	Tempo	
Roszieg	25	4	L1	110	20,10	4,72	0	20,10	20,19	5,19	20,10	20,25	4,43	
				120	31,50	3,77	0	31,50	34,33	5,15	31,50	38,74	4,43	
		H3	110	28,10	5,46	0	28,10	28,14	5,17	28,10	28,16	4,42		
			120	28,00	4,27	0	28,00	28,13	5,17	28,00	28,23	4,43		
		6	L1	110	9,70	332,21	0	9,70	10,16	6,00	9,70	10,21	4,83	
				120	11,00	281,03	0	11,00	11,93	6,03	11,40	12,14	4,86	
	H3	110	16,00	390,07	0	16,00	16,22	6,00	16,00	16,31	4,83			
		120	15,10	686,75	0	15,10	15,43	6,00	15,10	15,54	4,84			
	Heskia	28	4	L1	110	102,30	6,56	0	102,30	103,51	5,80	102,30	104,69	4,76
					120	122,60	5,78	0	122,60	123,71	5,74	122,80	126,17	4,76
			H3	110	172,50	6,70	0	172,50	173,07	5,78	172,50	173,91	4,74	
				120	171,20	7,30	0	171,20	171,75	5,79	171,20	171,91	4,75	
7			L1	110	34,90	177,36	0	34,90	36,39	7,45	35,00	36,64	5,57	
				120	42,60	216,38	0	42,60	44,32	7,39	43,30	44,58	5,55	
H3		110	75,20	260,93	0	75,20	76,42	7,36	75,20	76,60	5,51			
		120	67,20	341,83	0	67,20	70,19	7,38	67,20	70,38	5,53			

A Tabela 2 apresenta os resultados para os problemas Wee-Mag e Tonge. Os resultados obtidos pelo CPLEX para estes problemas não foram satisfatórios, sendo que, para as 160 instâncias testadas foi possível encontrar apenas soluções viáveis com *gaps* muito grandes, e em alguns casos, as soluções obtidas possuem atribuições incompatíveis entre tarefa e trabalhador. Além disso, em algumas instâncias o CPLEX não consegue encontrar nenhuma solução viável antes que ocorra estouro de memória do computador. Os resultados obtidos pelo CS foram melhores que as soluções viáveis providas pelo CPLEX em todos os casos. O tempo computacional do CS foi de aproximadamente 100 segundos nos piores casos, enquanto que no CPLEX o tempo computacional é muito alto e, além disso, o CPLEX é interrompido devido ao estouro de memória do computador na maioria das instâncias. Novamente o SA sem o processo

de agrupamento obteve resultados piores que o CS, porém com tempo computacional menor.

O CS encontrou dificuldades na resolução dos problemas Tonge e Wee-Mag, sendo que, em algumas execuções não foi possível obter uma solução viável devido ao fato da restrição de precedência não ser satisfeita. Sendo assim, as soluções médias do CS ficaram com valores muito altos. Este fato mostra que é necessário realizar novos estudos visando resolver a rede de precedência com maior eficiência.

Tabela 2 – Resultados Computacionais (problemas Tonge e Wee-Mag)

	Tam	NrT	Var	Inc	CPLEX				CS			SA			
					SI	SNI	Tempo	gap	Sol*	Sol	Tempo	sol*	sol	tempo	
Tonge	70	10	L1	I10	144,6	10,56	20222,9	92,6	107,5	732,7	58,0	113,4	838,8	19,6	
				I20	162,4	12,03	23863,7	92,4	141,8	826,0	59,3	225,3	887,5	20,5	
		H3	I10	276,7	16,45	18286,1	93,9	179,5	778,6	57,9	271,7	872,1	20,0		
			I20	292,8	18,59	21241,8	93,6	206,4	755,2	57,8	293,4	864,5	20,0		
		17	L1	I10	107,2	1,61	57480,3	98,5	71,9	877,6	83,9	623,8	950,8	22,8	
				I20	772,2	1,74	44656,3	98,7	83,9	913,1	86,0	819,1	994,9	24,3	
	H3	I10	231,4	2,63	54316,3	98,9	132,6	911,9	84,8	743,2	982,3	23,7			
		I20	376,8	2,64	57772,4	98,9	113,8	875,9	84,9	831,1	1000,5	23,6			
	Wee-Mag	75	11	L1	I10	42,4	2,80	17884,8	93,3	33,6	59,0	69,4	35,2	76,3	19,9
					I20	50,8	3,40	21507,3	93,2	38,5	45,8	71,3	41,3	45,9	21,6
			H3	I10	77,4	4,92	17227,8	93,6	56,2	67,1	69,8	59,7	85,2	20,7	
				I20	72,9	4,69	17039,1	93,5	55,2	73,4	69,8	58,1	74,1	20,6	
19			L1	I10	373,8	0,84	33323,6	99,1	19,4	504,1	99,3	117,5	561,6	24,2	
				I20	1258,5	0,85	43084,5	99,8	21,8	544,8	102,2	29,2	611,7	26,5	
H3		I10	659,5	0,90	51772,6	99,4	30,6	369,9	100,9	36,3	463,4	25,4			
		I20	470,0	0,87	56884,9	99,4	27,9	443,5	101,1	32,4	499,8	25,5			

A figura 4 apresenta um gráfico com um exemplo da convergência dos métodos CS e SA, mostrando que o CS converge para boas soluções mais rapidamente que o SA sem o processo de agrupamento. Além disso, o CS neste caso encontrou uma solução melhor que o SA.

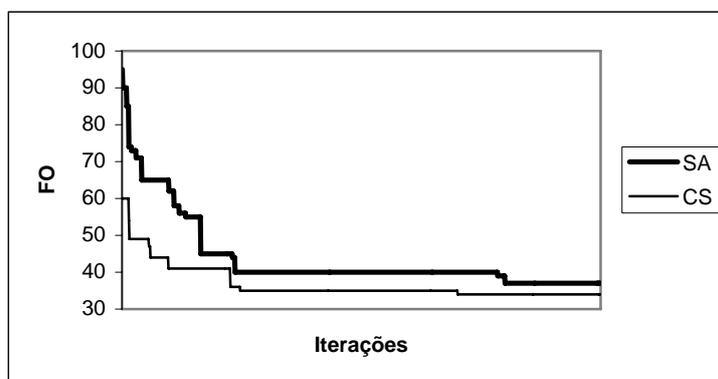


Figura 4 – Convergência dos métodos CS e SA

Por fim, podemos observar que os outros fatores (Inc: % de incompatibilidade tarefa/trabalhador, Var: variabilidade dos tempos das tarefas, e OS: característica da rede de precedência) não influenciam na robustez dos métodos.

### 7. Considerações Finais

Este artigo apresentou uma solução para o Problema de Balanceamento e Designação de Trabalhadores em Linha de Produção (ALWABP, do inglês *Assembly Line Worker Assignment and Balancing Problem*) utilizando a metaheurística híbrida *Clustering Search* (CS). Os resultados mostram que o CS é muito competitivo para resolução deste problema em tempos computacionais razoáveis. Uma vantagem adicional desta abordagem é o fato de não utilizar nenhum *solver* comercial, o que facilitará uma implementação real para ser usada pelos seus beneficiários finais: os CTD's.

No senso prático, os tempos computacionais baixos encontrados pelo CS possibilitam um rápido balanceamento das linhas de produção. Isto é muito importante se levarmos em conta que, devido ao alto absentismo e as consultas físicas e psicológicas periódicas, o responsável pelo CTD conhece somente no início de cada dia de trabalho quais trabalhadores estão disponíveis. Portanto abordagens como o CS, que provêem boas soluções em tempos computacionais pequenos, são muito desejáveis para montar uma linha de produção diariamente. Para as rotinas de rotação de trabalhadores, que são necessárias no ambiente dos CTD's, também é importante ter um ágil balanceamento da linha para que se tenha uma vasta combinação de soluções facilmente.

Os trabalhos futuros que podem ser feitos sobre o ALWABP englobam desde uma implementação real deste problema, até a análise de outras metaheurísticas para gerar soluções para o processo de agrupamento do CS, e a implementação de novos métodos de busca local.

### Agradecimentos

Este trabalho é desenvolvido sob o projeto de pesquisa TRENCADIS (GVA 2007-241) apoiado pelo *Regional Valencian Government* na Espanha. Os autores agradecem também ao Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq pelo suporte parcial a esta pesquisa, ao CTD envolvido nesta pesquisa pela colaboração e aos revisores pela contribuição ao trabalho.

### Referências

- Bellamy, G. T., Horner, R. H. e INMAN, D. P.**, Vocational Habilitation of Severely Retarded Adults: A direct Service Technology. *University Press*, Baltimore-USA, 1979.
- Chaves, A. A., Miralles, C. e Lorena, L. A. N.** (2007), Clustering search approach for the assembly line worker assignment and balancing problem. 37<sup>a</sup> International Conference on Computers and Industrial Engineering, 1469-1478.
- Corominas, A., Pastor, R. e Plans, J.** (2003), Línea de montaje con tiempos dependientes del tipo de operario. 27<sup>o</sup> Congreso Nacional de Estadística e Investigación Operativa, 2789-2795.
- Doerr, K., Klastorin, T. D. e Magazine, M. J.**, Synchronous unpaced flow lines with worker differences and overtime cost. *Management Science*, v. 46, 421-435, 2000.
- Gel, E. S., Hopp, W. J. e Oyen, M. P. V.**, Factors affecting opportunity of worksharing as a dynamic line balancing mechanism. *IIE Transactions*, v. 34, 847-863, 2002.
- Glover, F.**, Tabu search and adaptive memory programming: Advances, applications and challenges, em Barr, R. S., Helgason, R. V. e Kennington, J. L. (Eds.), *Interfaces in Computer Science and Operations Research*. Kluwer, 1-75, 1996.
- Hoffmann, T. R.**, Assembly line balancing: A set of challenging problems. *International Journal of Production Research*, 28, 1807-1815, 1990.
- Hopp, W. J., Tekin, E. e Oyen, M. P. V.**, Benefits of skill chaining in Production Lines with Cross Trained workers. *Technical Report - WorkSmart Laboratory*, 2001.
- Kirkpatrick, S., Gellat, D. C. e Vecchi, M. P.**, Optimization by simulated annealing. *Science*, v. 220, 671-680, 1983.
- Mansoor, E. M.**, Assembly line balancing: A heuristic algorithm for variable operator performance levels. *Journal of Industrial Engineering*, v. 19, 618-628, 1968.
- Miralles, C., García-Sabater, J., Andrés, C. e Cardós, M.**, Advantages of assembly lines in sheltered work centres for disabled. a case study. *International Journal of Production Economics*, 187-197, 2007.
- Miralles, C., García-Sabater, J., Andrés, C. e Cardós, M.**, Branch and bound procedures for solving the assembly line worker assignment and balancing problem. application to sheltered work centres for disabled. *Discrete Applied Mathematics*, 156, 352-367, 2008.
- Oliveira, A. C. M. e Lorena, L. A. N.**, Hybrid evolutionary algorithms and clustering search, em Grosan, C., Abraham, A. e Ishibuchi, H. (Eds.), *Hybrid Evolutionary Systems - Studies in Computational Intelligence*. Springer SCI Series, 81-102, 2007.
- Scholl, A. e Becker, C.**, State-of-the-art exact and heuristic solution procedures for simple assembly line balancing. *European Journal of Operational Research*, 168, 666-693, 2006.