Aplicação de meta-heurísticas na abordagem do problema de otimização do ciclo de montagem de uma insersora automática de componentes

Arthur Tórgo Gómez¹ Diogo Alberto Borges¹

Resumo: Este artigo apresenta a aplicação de meta-heurísticas na abordagem do problema de otimização do ciclo de operação de uma insersora automática de componentes (pick-and-place) em máquinas do tipo multicabeças. Um algoritmo híbrido foi desenvolvido utilizando as metaheurísticas Busca Tabu e Algoritmo Genético. Na abordagem do problema são considerados os problemas de escalonamento das ferramentas, de escalonamento da sequência de inserção de componentes e, por fim, o problema da alocação dos componentes no alimentador. Como resultados foram obtidas soluções de boa qualidade aplicáveis a sistemas reais.

Palavras-chave: Escalonamento. Meta-heurísticas. Otimização combinatória.

Abstract: This paper presents the use of metaheuristics to approaching the Optimization Problem of an Operation Cycle in an Automatic Placement Multi-head Machine (pick-and-place) of components. A hybrid algorithm was developed using the meta-heuristics Tabu Search and Genetic Algorithm. In developing the algorithm are considered problems Scheduling Tools, Scheduling Sequence Insertion of Components and finally the Allocation Problem of the components in the feeder. The results obtained are good quality solutions applicable to real systems.

Keywords: Combinatorial optimization. Meta-heuristics. Scheduling.

1 Introdução

Em indústrias de produtos eletrônicos, que utilizam componentes de Tecnologia de Montagem de Superfície (TMS), as linhas de montagem têm máquinas comumente chamadas de pick-and-place. Essas máquinas inserem componentes do tipo TMS em placas de circuito impresso. Neste artigo, é apresentado um algoritmo híbrido desenvolvido utilizando os conceitos das meta-heurísticas Busca Tabu e Algoritmo Genético. O algoritmo busca minimizar o tempo gasto com a inserção dos componentes. Para isso são abordados os problemas de escalonamento das ferramentas, de escalonamento da sequência de inserção de componentes e de alocação dos componentes no alimentador de componentes. Como resultados interessantes tem-se o desenvolvimento de um algoritmo híbrido, capaz de encontrar soluções de boa qualidade para o problema de otimização associado ao processo de inserção de componentes em máquinas pick-and-place e de uma metodologia que contempla, em uma abordagem integrada, dois problemas de escalonamento juntamente com o problema de alocação dos componentes no alimentador.

O artigo foi estruturado da seguinte maneira. Na seção 2, são apresentados o problema e a abordagem proposta. Na seção 3, são descritos o modelo, sua formulação matemática e a arquitetura computacional. Na seção 4, são apresentados os experimentos computacionais e, na seção 5, as conclusões finais.

http://dx.doi.org/10.5335/rbca.2014.3577

¹ Programa Interdisciplinar de Pós-Graduação em Computação Aplicada— Universidade do Vale do Rio dos Sinos (Unisinos) – São Leopoldo -RS – Brasil. {breno,diogo@unisinos.br}

2 O problema de otimização do ciclo de montagem de uma máquina insersora automática de componentes

Uma máquina insersora automática de componentes consiste em um caso particular de máquinas multicabeças, em que apenas uma pinça é montada na cabeça insersora. Crama [1] dividiu esse problema de otimização, aplicável a uma máquina simples, em quatro subproblemas, a saber:

- a) alocação dos componentes no alimentador de componentes;
- b) sequência de inserção de componentes em uma placa de circuito impresso;
- c) plano de carga dos componentes;
- d) controle de movimento.

Posteriormente, Ayob [2] acrescentou o subproblema (e): a otimização do conjunto de ferramentas (pinças a vácuo) utilizadas no processo de inserção dos componentes. Esse subproblema é importante, uma vez que as trocas de ferramentas constituem a operação que mais demanda tempo da máquina [3].

Este artigo aborda quatro, dos cinco subproblemas citados: a alocação de componentes no alimentador, a sequência de inserção dos componentes em uma placa de circuito impresso, o plano de carga dos componentes e a otimização do conjunto de ferramentas. Para o controle de movimento, assume-se que o sistema eletrônico da máquina fica responsável por essa tarefa, pois o problema está intrinsecamente ligado ao projeto eletromecânico da máquina. Não sendo considerado o controle de movimento serão abordadas somente as operações de alto nível como comandos de troca de ferramentas, posicionamento da cabeça de inserção, carga e inserção de componentes. Essa abordagem é utilizada na grande maioria dos trabalhos estudados sobre o tema [1]. Em função do modelo de máquina abordada neste trabalho ser do tipo multicabeça, o plano de carga e a sequência de inserção foram considerados um único problema.

2.1 Abordagem proposta

O problema da otimização do ciclo de montagem de uma máquina insersora automática de componentes será aqui abordado em três problemas distintos:

- escalonamento das ferramentas: definição do conjunto de ferramentas no magazine que fornecera as pinças à cabeça insersora, de modo a minimizar o número de trocas de ferramentas e maximizar a quantidade de componentes pinçados. A associação entre componentes e ferramentas será abordada como um problema de família de partes. Já a sequência de carregamento e troca de ferramentas será abordada como um problema de escalonamento;
- alocação de componentes no alimentador: definição da posição ótima no alimentador de componentes, de modo a minimizar a trajetória da cabeça de inserção;
- sequência das inserções de componentes: definição da ordem de execução das inserções, baseada na locação dos componentes no alimentador de componentes e nos grupos de ferramentas, de modo a minimizar o deslocamento linear total e também a quantidade de trocas de ferramentas. Esse problema pode ser modelado como um problema de escalonamento.

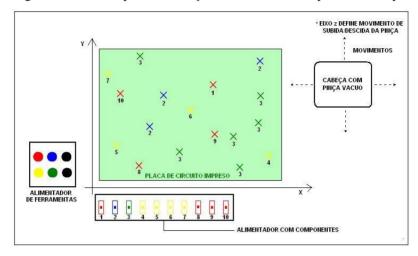
A solução dos três problemas passa por um ponto em comum: a redução do tempo total de montagem de uma placa de circuito impresso. A visão particionada do problema é importante, pois permite compreender de forma mais clara sua dimensão. Abordá-lo de forma direta é inviável, já que o problema de escalonamento, conforme Viana [4] é classificado como um problema NP-Difícil, e o problema de família de partes [5] é caracterizado como um problema NP-Completo.

Considerando os problemas descritos é possível formalizar o enunciado do problema geral como: otimizar o conjunto de ferramentas e a sequência de trocas dessas na cabeça insersora, a alocação dos componentes no alimentador e a sequência de inserção dos componentes em uma placa de circuito impresso de modo a reduzir o tempo total de inserção dos componentes na placa. O enunciado anterior valida à afirmação: "Enunciar problemas de otimização combinatória costuma ser simples, porém devido à quantidade de soluções, resolvê-los é uma tarefa complicada." [6].

3 Modelagem

A Figura 1 ilustra a vista superior de uma máquina de inserção automática de componentes do tipo multicabeças. A máquina tem a mesa de fixação da placa de circuito impresso e o alimentador de componentes fixos.

Figura 1: Diagrama de uma máquina de inserção automática de componentes do tipo multicabeças



Na Figura 1 observa-se, à esquerda, o alimentador de ferramentas. Na parte inferior da figura está localizado o alimentador de componentes com apenas dez componentes ilustrados. Com o objetivo de modelar uma máquina real, o alimentador do modelo terá a capacidade para até cem componentes. No lado direito da figura está localizada a cabeça insersora com uma ou mais pinças a vácuo. No modelo proposto o número de pinças é um parâmetro que pode variar de um a vinte. A cabeça de inserção tem liberdade para movimentar-se por toda superfície da máquina (eixos X e Y), atingindo o alimentador de ferramentas, o alimentador de componentes e a placa de circuito impresso. As pinças são independentes e têm liberdade de movimento no eixo Z, com o objetivo de carregar e descarregar componentes. O quadro com marcações em X ilustra uma possível placa de circuito impresso, na qual os componentes serão inseridos. As marcações dentro da placa indicam a posição em que os componentes devem ser inseridos.

3.1 Formulação matemática

Para definição do modelo matemático foi utilizada formulação própria, usando notação semelhante à utilizada por Ayob e Kendall [7]. O modelo por eles proposto não considera os tempos de troca de ferramentas, além de ter sido desenvolvido para uma insersora com características diferentes da adotada neste trabalho, o que altera completamente as características do modelo. Baseado nisso, apenas a notação foi mantida. A função-objetivo é apresentada na equação (1):

minimizar

$$TC = \sum_{i=1}^{N} \left[\frac{D_n}{V_r} \right] + N.T_i + N.T_c + P.T_f , quando i = N, f(n+1) = f(n)$$
 (1)

onde,

TC Representa o tempo total de montagem (tempo de ciclo).

Número de componentes inseridos.

K Número de componentes diferentes.

P Número de troca de ferramentas efetuadas.

 D_n Representa a função que descreve a distância percorrida pela cabeça

de inserção na inserção n.

 T_i Tempo para inserir um componente.

 T_c Tempo para carregar um componente.

 T_f Tempo para troca de uma ferramenta.

 V_r Velocidade média da cabeça de inserção.

A função D_n é apresentada na equação (2):

$$D_n = D_{C_1} \cdot A + D_{C_2} \cdot B + D_{C_3} \cdot C \tag{2}$$

onde,

 D_{C1} , D_{C2} e D_{C3} representam as distâncias percorridas pela cabeça de inserção em três casos distintos, apresentados na Figura 2. A, B e C representam variáveis condicionais que anulam termos da equação geral da distância (2) aplicáveis somente a casos específicos:

$$A = \begin{cases} 1, caso \ o \ pr\'oximo \ passo \ seja \ uma \ inserç\~ao \ simples (Figura 2(a)) \\ 0, caso \ contr\'ario \end{cases} \tag{3}$$

$$B = \begin{cases} 1, caso \ ocorra \ apenas \ carregamento \ de \ componente \ s \\ antes \ da \ inserção \ (Figura 2(b) \\ 0, caso \ contrário \end{cases}$$
 (4)

$$C = \begin{cases} 1, caso \ ocorra \ troca \ de \ ferramentas \ e \ carregamento \ de \ componentes \\ antes \ da \ inserção \ (figura 2(c)) \\ 0, caso \ contrário \end{cases} \tag{5}$$

$$A,B,C \in \{0,1\}$$

D_{C1}, para o caso ilustrado na Figura 2 (a), é expressa por:

$$D_{C1} = D_1 = \sqrt{(X_n - X_{n+1})^2 + (Y_n - Y_{n+1})^2}.$$
 (7)

 D_{C2} , para o caso ilustrado na Figura 2 (b), é expressa por:

$$D_{C2} = D_1 + D_2 + D_3 \tag{8}$$

onde,

$$D_{1} = \sqrt{(X_{n} - Cx_{n+1})^{2} + (Y_{n} - Cy_{n+1})^{2}}$$
(9)

$$D_2 = \sum_{j=n+1}^{n+CR} \sqrt{(Cx_j - Cx_{j+1})^2 + (Cy_j - Cy_{j+1})^2}$$
 (10)

$$D_3 = \sqrt{\left(Cx_{n+CR} - X_{n+1}\right)^2 + \left(Cy_{n+CR} - Y_{n+1}\right)^2}$$
 (11)

D_{C3}, para o caso ilustrado na Figura 2(c), é expressa por:

$$D_{C3} = D_1 + D_2 + D_3 + D_4 (12)$$

onde,

$$D_{1} = \sqrt{(X_{n} - Fx)^{2} + (Y_{n} - Fy)^{2}}$$
(13)

$$D_2 = \sqrt{(Fx - Cx_{n+1})^2 + (Fy - Cy_{n+1})^2}$$
(14)

$$D_{3} = \sum_{j=n+1}^{n+CR} \sqrt{\left(Cx_{j} - Cx_{j+1}\right)^{2} + \left(Cy_{j} - Cy_{j+1}\right)^{2}}$$
 (15)

$$D_4 = \sqrt{\left(Cx_{n+CR} - X_{n+1}\right)^2 + \left(Cy_{n+CR} - CY_{n+1}\right)^2}$$
 (16)

Para todos os casos as restrições são:

$$\begin{split} &D_{C1}, D_{C2}, D_{C3} \geq 0 \\ &D_{1}, D_{2}, D_{3}, D_{4} \geq 0 \\ &1 \leq CR \leq F_{MAX} \\ &CR \notin \{1, 2, ..., F_{MAX}\} \end{split} \tag{17}$$

Para as distâncias expressas pelas equações D_{C1}, D_{C2} e D_{C3}, a nomenclatura utilizada é:

- D_{CI} : distância percorrida pela cabeça de inserção para o caso ilustrado na Figura 5 (a).
- D_{C2} : distância percorrida pela cabeça de inserção para o caso ilustrado na Figura 5 (b).
- D_{C3:} distância percorrida pela cabeça de inserção para o caso ilustrado na Figura 5 (c).
- D₁,D₂,D₃,D₄: distâncias percorridas pela cabeça de inserção para os casos expressos na Figura 5.
- X_n, Y_n : coordenadas cartesianas da inserção indexada por n.
- Cx_n, Cy_n : coordenadas cartesianas do componente no alimentador, inserido na inserção n.
- F_x,F_y: coordenadas cartesianas do centro médio do alimentador de ferramentas.
- CR: número de componentes carregados do alimentador em uma operação de carregamento.
- F_{MAX} : número máximo de ferramentas simultâneas na cabeça de inserção.
- J: indexador de carregamentos.
- N: indexador de inserções.

Na Figura 2 são apresentados os três casos de inserção considerados no modelo proposto. Em (a) é apresentado o caso de inserção simples, em (b) é apresentado o caso de inserção com recarga de componente(s) e, por fim, em (c) é apresentado o caso da inserção com troca de ferramentas e recarga de componentes.

ALIMENTADOR DE FERRAMENTAS

(Xn,Yn)

ALIMENTADOR COM COMPONENTES

(Cx1,Cy1)...(CxK,CyK)

(Xn,Yn)

ALIMENTADOR COM COMPONENTES

(Cx1,Cy1)...(CxK,CyK)

(Xn,Yn)

ALIMENTADOR COM COMPONENTES

(Cx1,Cy1)...(CxK,CyK)

ALIMENTADOR COM COMPONENTES

(Cx1,Cy1)...(CxK,CyK)

ALIMENTADOR COM COMPONENTES

(Cx1,Cy1)...(CxK,CyK)

ALIMENTADOR COM COMPONENTES

(Cx1,Cy1)...(CxK,CyK)

Figura 2: Possibilidades de inserções consideradas

(c)

Na inserção simples, exemplificada na Figura 2 (a), a distância percorrida pela cabeça insersora é formada apenas do movimento da cabeça insersora da posição do componente indexada pela variável n para a posição do componente indexado por n+1. Essa distância é representada por D_I . Esse tipo de movimento somente ocorre em máquinas com mais de uma pinça a vácuo na cabeça insersora (multicabeça). Máquinas com apenas uma pinça, sempre realizam uma nova recarga após uma inserção. No caso da inserção com recarga de componentes, exemplificado na Figura 2 (b), a distância percorrida é composta por três movimentos distintos. A cabeça insersora sai da posição N e vai até o primeiro componente a ser carregado (distância representada por D_I), a seguir, a cabeça executa um número de deslocamentos igual à quantidade de componentes carregados. A soma desses deslocamentos é representada por D_2 . Após o último componente ser carregado, a cabeça insersora desloca-se até a posição do componente indexado por n+1. Essa distância é representada por D_3 . Cabe salientar um caso particular, em que máquinas com apenas uma pinça na cabeça insersora (CR = 1) à distância D_2 é sempre nula, uma vez que apenas um componente é carregado. Finalmente, para o caso inserção com troca de ferramentas e recarga de componentes, exemplificado na Figura 2 (c), a distância percorrida é composta por quatro movimentos distintos. Inicialmente, a cabeça de inserção sai da posição N e desloca-se até o alimentador de ferramentas para executar à sua troca. Essa distância é representada por D_I . Especificamente para esse caso, o modelo sofreu uma simplificação em relação ao caso real. Em função da proximidade das ferramentas, foi considerado que todas as ferramentas estão localizadas em um mesmo ponto (Fx, Fy), sendo esse ponto o centro geométrico do alimentador de ferramentas. No passo seguinte, a cabeça de inserção passa do alimentador de ferramentas para o primeiro componente a ser carregado, e essa distância é representada por D_2 . A seguir, a cabeça de inserção executa um número de deslocamentos igual à quantidade de componentes carregados. A soma desses deslocamentos é representada por D₃. Após o último componente ser carregado, a cabeça insersora desloca-se até a posição do componente indexado por n+1, cuja distância é representada por D_4 . Conforme o caso apresentado em 5 (b), uma simplificação devido a um caso particular também ocorre. Em máquinas com apenas uma pinça na cabeça insersora (sequential pick-and-place), à distância D₃ também é sempre nula.

A função objetivo (FO) proposta na equação (1) é apresentada de forma simplificada na equação (18).

$$TC = \sum_{i=1}^{N} \left[\frac{D_n}{V_r} \right] + NT_i + NT_c + PT_f$$
(18)

Onde, o número de componentes inseridos N é um parâmetro fornecido pelo CAD. Ele não pode ser alterado sendo uma constante na FO. Isso vale para T_i e T_c , onde esses parâmetros descrevem tempos que a máquina leva para executar determinadas operações. Com base nisso os termos $N.T_i$ e $N.T_c$ são constantes que em nada afetam o processo de otimização. A vantagem de esses termos estarem presentes na FO está no fato de que o valor fornecido, pela essa, deixa de ser comparativo e passa a ser quantitativo, ou seja, permite comparar diretamente o resultado obtido com um caso real.

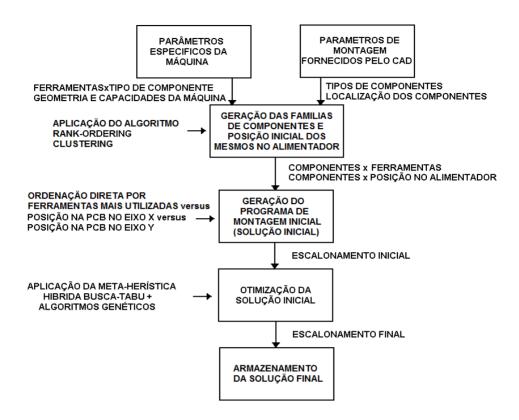
3.2 Arquitetura do modelo

A arquitetura do modelo computacional é ilustrada na Figura 3. A arquitetura foi dividida em cinco blocos. No primeiro, são coletados os dados fornecidos pelo CAD, que gerou o projeto do circuito eletrônico e dos parâmetros da máquina, relativos às suas capacidades e à sua geometria.

No segundo módulo, utilizando conceitos da Teoria de Grupo (TG) [8] e aplicando o algoritmo *rank* order clustering [9], são geradas as famílias de partes [10]. Este algoritmo, baseado em ordenação, associa em grupos os componentes e as pinças a vácuo disponíveis, levando em consideração os componentes utilizados na montagem e as capacidades e quantidades de pinças à disposição. Também é realizada a definição da posição inicial dos componentes no alimentador. Essa definição apenas precisa ser válida, assim os componentes são ordenados em função da utilização, a partir do centro do alimentador.

No terceiro módulo é gerada a solução inicial para a implementação da meta-heurística Busca Tabu. A solução inicial é gerada, por meio da ordenação direta da posição geométrica dos componentes na placa de circuito impresso, considerando os grupos de ferramentas mais utilizadas. Esse procedimento visa gerar o menor número de troca de ferramentas e, também, que a inserção seguinte à atual seja vizinha não muito distante dessa, evitando grandes deslocamentos por parte da cabeça de inserção. A política utilizada para determinação do escalonamento do *setup* visa maximizar o número de componentes transportados, de forma simultânea, pela cabeça de inserção no alimentador de componentes.

Figura 3: Arquitetura do modelo computacional



No quarto módulo é executado o algoritmo híbrido. Na implementação da meta-heurística híbrida a Busca Tabu [11] [12] [13] é responsável pela busca de novas soluções, privilegiando a política de intensificação do espaço de busca enquanto que o algoritmo genético [14] [15] explora uma política de diversificação. Na Figura 4 é apresentado o diagrama em blocos do algoritmo híbrido.

A Busca Tabu é uma meta-heurística proposta por Fred Glover, em 1986 [11] [12] [13]. Essa consiste de um conjunto de conceitos e práticas que são usadas para resolver problemas de otimização combinatória. Tal técnica é utilizada para encontrar soluções aproximadas de problemas complexos, quando o tempo computacional para encontrar a solução ótima é exponencial ou fatorial. A Busca Tabu é uma meta-heurística baseada em um processo de busca na vizinhança, também chamada de Busca Local, devido às suas similaridades com o método de Busca Local. Entretanto, a principal diferença da Busca Tabu para a Busca Local é a existência de uma lista que restringe a execução de movimentos reversos, chamada de Lista Tabu. Por intermédio dessa técnica, o algoritmo apresenta a habilidade de escapar de ótimos locais, aumentando a busca no espaço de pesquisa, o que permite obter soluções de melhor qualidade.

O algoritmo genético foi proposto por John Holland, em 1975 [14]. Esse algoritmo tem sido utilizado para encontrar soluções para problemas de otimização combinatória em diversas áreas, tais como, matemática, física, biologia e automação industrial, dentre outras. Os algoritmos genéticos são algoritmos de pesquisa baseados nos mecanismos da genética e da seleção natural. O critério de parada do Algoritmo Genético é o número de gerações. O operador de crossover é responsável por cruzar a informação genética dos pais para gerar crianças individualmente, e o operador de mutação, por sua vez, tem como objetivo manter a diversidade da população. Ambos os operadores terão uma probabilidade de ocorrência associada a ser definida de acordo com a aplicação.

No diagrama da Figura 4, a partir de uma solução inicial, a BT é executada até que o critério de parada seja atingido. No caso, um número de iterações (nbmax) sem melhora no valor da função objetiv. Durante a execução, uma lista com as melhores vinte soluções encontradas é armazenada. Caso o número de vinte soluções não for atingido, as faltantes serão geradas randomicamente. Essa lista serve como população inicial para o AG,

que é executado até que o critério de parada tenha sido atingido. No caso, um número de gerações predeterminado. Posteriormente, a solução gerada pelo AG é comparada com a solução encontrada anteriormente pela BT, em caso de melhora, a BT é novamente executada, caso o valor da função objetivo não melhore, o Algoritmo Híbrido é finalizado.

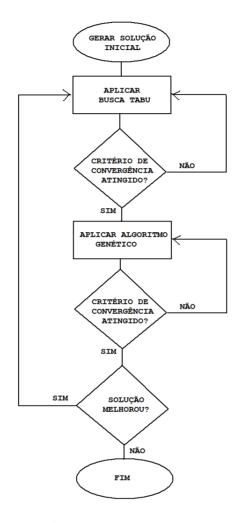


Figura 4: algoritmo híbrido

No quinto módulo, a partir da solução final obtida, um programa de montagem é gerado. Esse programa está adequado a detalhes específicos da máquina em que esse será aplicado, incluindo, também, a definição da posição de cada componente nos alimentadores que são armazenados para uso posterior pela máquina insersora.

3.3 Validação do modelo

Para validação do modelo foram utilizadas instâncias de teste disponíveis e já analisadas por outros autores e por outras técnicas. Em função de o problema estudado ser composto por três subproblemas, a validação do algoritmo foi realizada para cada um desses separadamente. Esse procedimento facilitou a comparação com os resultados de outros autores.

A análise individual de cada subproblema foi realizada da seguinte forma: o escalonamento das ferramentas foram tratadas como um problema de família de partes associado a um algoritmo determinístico para a geração da solução inicial. Posteriormente, durante a otimização, o problema de família de partes é considerado resolvido e o problema do *setup* de ferramentas passa a ser tratado como um problema de escalonamento [16]

[17] [18]. A alocação componentes no alimentador e a definição da sequência das inserções dos componentes foram tratados como um problema do caixeiro-viajante [19].

3.3.1 Sequência das inserções dos componentes (SIC)

Para análise deste problema, a função objetivo apresentada na equação (1) foi simplificada. Todos os parâmetros de tempo relativos a *setup* assumiram o valor zero. A codificação do avaliador da função objetivo foi alterada e todas as operações de carga e de movimentação relacionadas com a posição dos componentes nos alimentadores foram anuladas. Com isso, o modelo passou a representar uma máquina que dispõe de todas as ferramentas necessárias sempre disponíveis e prontas para o uso nas cabeças de inserção (ausência de *setups*), as quais têm todos os componentes necessários carregados de forma simultânea, o que evita as operações e movimentações de carregamento. Devido às alterações, o problema fica simplificado à otimização da trajetória da cabeça de inserção, de modo a reduzir a distância total percorrida. Essa simplificação permite analisar o problema como um problema do caixeiro-viajante, que tem um grande número de instâncias disponíveis para teste.

Nesta abordagem, foram realizados três experimentos, classificados de forma a encaixarem-se na realidade do problema estudado: problema de pequena escala (até sessenta componentes), problema de média escala (até quinhentos componentes) e problema de grande escala (mais de 2 mil componentes). Cabe salientar que o percentual de placas de circuito impresso com mais de 2 mil componentes inseridos automaticamente é extremamente baixo. Em média, as placas têm entre cem e trezentos componentes inseridos de forma automatizada.

Para realizar esses testes foram escolhidas três instâncias disponíveis na literatura e amplamente testadas. Essas instâncias são parte da biblioteca TSPLIB95, mantidas pela Universidade de Heidelberg, localizada na Alemanha. As instâncias selecionadas foram: eil51 (contendo 51 pontos de visitação), d493 (essa contendo 493 pontos de visitação) e, finalmente, a d2103 (com 2.103 pontos de visitação). Na Tabela 1 são apresentados os resultados obtidos comparados com os resultados das instâncias.

Tabela 1: Análise comparativa com instâncias conhecidas e aplicadas ao problema da sequência de inserção

Instância	Resultado da literatura (s)	Resultado do algoritmo híbrido (s)	Diferença percentual
eil51	426	426,25	+0,059%
d493	35002	35002,36	+0,001%
d2103	80450	80448,19	-0,0022%

Pode-se observar na Tabela 1 que os valores são idênticos, caso a parte fracionária seja desconsiderada. Por definição, da entidade que administra as instâncias, as soluções sempre são fornecidas com resultados inteiros. Segundo os dados de teste, as soluções das instâncias eil51 (426) e d493 (35002) são consideradas ótimos globais. Caso a definição adotada pelos mantenedores das instâncias seja usada, é possível dizer que o algoritmo é capaz de atingir os mesmos resultados de outros trabalhos, para instâncias de até médio porte e, até mesmo, apresentar desempenho superior para as instâncias aqui consideradas de grande porte. Cabe salientar que, na Tabela 1, foi apresentado o melhor valor obtido em duzentas execuções para cada uma das instâncias testadas. Para a instância eil51, em toda série de experimentos, 121 das duzentas execuções apresentaram o valor de R\$ 426,25. Em termos percentuais, 60,5% dos experimentos atingiram o valor apresentado na tabela. Para a instância d493, observou-se que 83 dos duzentos experimentos atingiram o valor apresentado na Tabela 1. Esse valor corresponde a 41,5% do total. Para a instância d2103 dos duzentos experimentos feitos, 12 chegaram ao valor apresentado na Tabela 1, o que corresponde a 6% do total de experimentos.

3.3.2 Alocação de componentes nos alimentadores (ACA)

Para validar o subproblema de alocação de componentes nos alimentadores, foi adotada a mesma metodologia do teste anterior. Entretanto, o enfoque da aplicação da instância foi diferente. No subproblema anterior, cada item da instância representava uma posição de inserção de um componente na placa de circuito impresso. Agora, cada item da instância representa a posição física de um alimentador. Para esse teste, todos os tempos gastos com *setup* e movimentações de inserção de componentes foram considerados nulos. A função

objetivo foi alterada, de modo a considerar apenas os tempos gastos com movimentação, visando à carga de componentes. Após a carga do item ter sido efetuada, todos os movimentos gastos com inserções de componentes não foram contabilizados. Com isso, novamente é possível modelar o subproblema com um PCV, onde o resultado final para a ordem dos itens no alimentador de componentes é o caminho percorrido e não a distância total percorrida. Com esse teste é possível verificar se os movimentos realizados pelo algoritmo híbrido são efetivos. Apesar dos movimentos codificados serem os mesmos do teste anterior, esses são aplicados a partes diferentes (matrizes) da solução em análise, o que justifica o teste. Na Tabela 2, do mesmo modo como no teste anterior, são apresentados os resultados do teste.

Tabela 2: Análise comparativa aplicad	Tabela 2: Análise comparativa aplicada ao problema da alocação de componentes						
Resultado da literatura	Resultado do algoritmo	Diferenca					

Instância	Resultado da literatura	Resultado do algoritmo	Diferença
	(s)	híbrido (s)	percentual
Eil51	426	426,25	+0,059%
d493	35002	35002,36	+0,001%
d2103	80450	80448,19	-0,0022%

Analisando a Tabela 2 e comparando-a com a tabela anterior, pode-se observar que os mesmos resultados foram encontrados. No teste da Tabela 2, os movimentos foram realizados de modo a alterar a alocação dos itens no alimentador de componentes. Essa análise comparativa mostra que os movimentos efetuados pela metaheurística são efetivos para a solução desse problema, como também foram no problema anterior. Novamente, neste teste, foram efetuados duzentos experimentos para cada instância testada, sendo apresentado o melhor resultado.

3.3.3 Escalonamento de ferramentas (EF)

A determinação do escalonamento de ferramentas é o único problema que não pode ser analisado separadamente. Isso ocorre, pois todos os *setups* de ferramentas são dependentes de forma direta dos componentes inseridos. Em função disso, esse problema é sempre dependente da ordem de execução das inserções. Essa característica torna complicada a comparação direta com instâncias públicas para o problema sem pesadas alterações no algoritmo.

Na metodologia adotada para a validação definiu-se que a ordem de execução das inserções fosse determinada de forma eficiente pela técnica proposta. Baseado nisso, a função objetivo foi alterada de forma que somente os tempos relativos ao *setup* de ferramentas fossem computados. Para padronizar os testes, as mesmas instâncias dos problemas anteriores foram utilizadas. Em todos os experimentos, o total de componentes inseridos foi dividido em três grupos, sendo que cada grupo necessita de um tipo diferente de ferramenta para sua inserção.

Na Tabela 3 são apresentados os resultados dos testes efetuados. Nessa é realizada uma análise comparativa entre a solução inicial do *setup*, gerada pelo algoritmo determinístico, e o tempo final gasto com *setup*, após a execução da técnica proposta que otimiza o *setup* de forma conjunta com a sequência de inserção. Nesta tabela é possível verificar que, independente da instância utilizada, a solução final encontrada para cada um dos três casos propostos (1,4 e dez pinças na cabeça insersora) é a mesma, quando a comparação é realizada entre instâncias. Os resultados encontrados são esperados por dois fatores:

- a) o primeiro fator deve-se ao fato de que para todas as instâncias são usadas três ferramentas para inserção de todos os componentes;
- b) no segundo, a nova função objetivo teve todas as variáveis não relacionadas ao setup anuladas.

Com isso, independente do número de componentes inseridos, as inserções são agrupadas pela metaheurística em três grandes grupos, onde, no pior dos casos, três *setups* são necessários. Nos problemas da Tabela 3, estipulou-se o tempo para inserção ou remoção de ferramenta, como sendo 1 segundo e o tempo para troca de ferramenta, igual a 2 segundos.

Observa-se nos casos em que a máquina tem apenas uma cabeça de inserção, que as três operações de *setup* realizadas foram, respectivamente: inserir uma ferramenta, seguido de duas trocas posteriores de ferramentas. Nos demais problemas, a meta-heurística optou por deixar ferramentas na cabeça de inserção

ociosas, de modo a minimizar o setup. Em todos os casos, as três operações de setup foram à inserção de todas as ferramentas necessárias no início do programa de montagem.

Cabecas Solução inicial Solução final % de melhora da insersoras solução final em Instância % de uso % de uso disponíveis relação à solução das Tempo(s) Tempo (s) das inicial na máquina cabeças cabeças 5 51 1 5 100 100 (1/1) 0 Ei151 3

100

100

100

100

100

100

100

100

Tabela 3: Análise comparativa entre as soluções inicial e final

3

5

3

3

5

3

3

75 (3/4)

33 (3/10)

100 (1/1)

75 (3/4)

33 (3/10)

100 (1/1)

75 (3/4)

33 (3/10)

25

15

0

20

7,5

0

20

6.7

Outra observação, que demonstra a eficiência da técnica proposta, é baseada na solução inicial. O setup inicial é gerado por um algoritmo determinístico, no qual a regra básica é manter a cabeca de inserção o menos ociosa possível, maximizando o número de itens transportados. Para os casos apresentados, devido à simplificação na FO, em máquinas com grande número de ferramentas na cabeça insersora a solução inicial gerada, apesar de válida, é de baixa qualidade. Percebe-se, com isso, que mesmo partindo de uma solução desfavorável, a meta-heurística é capaz de alterar todo o escalonamento de inserções, de modo a unificar os componentes em três grupos contínuos para reduzir a quantidade de setups. A técnica converge, mesmo com soluções iniciais altamente desfavoráveis, como é o caso da instância d2103, aplicada a uma máquina com dez ferramentas nas cabeças de inserção, em que o tempo de setup baixou de 45 segundos (indício de componentes desagrupados no programa de montagem) para três segundos.

4 **Experimentos computacionais**

4

10

1

4

10

1

4

10

Ei151

d493

d493

d493

d2103

d2103

d2103

12

20

5

15

40

5

15

45

Com o objetivo de testar o comportamento dinâmico do modelo e a contribuição de cada subproblema na composição da solução final, alguns experimentos, utilizando testes reais foram propostos. Posteriormente, visando analisar os ganhos relativos a tempo e o custo em um caso real, o resultado de um programa de montagem, gerado pela meta-heurística, foi confrontado com a solução gerada pelo software da máquina em

4.1 Análise de significância dos subproblemas

Para realização desse experimento foram utilizadas três instâncias que correspondem a itens montados em uma fábrica onde uma insersora foi disponibilizada para essa análise. Em cada caso, para avaliar o peso individual dos subproblemas, a seguinte metodologia foi adotada:

- o algoritmo híbrido foi executado sem restrição alguma nos três casos. O melhor resultado de duzentos experimentos (para cada subproblema) foi considerado como base de comparação;
- para cada instância, a solução inicial foi a mesma.

Na Tabela 4 são apresentadas algumas características das três instâncias. Optou-se modelar uma máquina com quatro pinças de inserção, pois essa condição permite analisar, de forma mais efetiva, a capacidade do algoritmo em gerar soluções de boa qualidade.

Tabela 4: Características das instâncias

Instância	Número total de componentes inseridos	Número total de ferramentas utilizadas	Número de componentes diferentes
DECEDTOD 22		umzauas	7
RECEPTOR_23	23	3	/
ALARME_57	57	3	12
CENTRAL_315	315	3	22

Na Tabela 4, as três instâncias foram escolhidas baseadas nos critérios de quantidade de itens, dentre esses, diversidade de itens inseridos e diversidade no uso das ferramentas. A primeira coluna da tabela apresenta o nome da instância, a segunda, a quantidade de componentes inseridos, a terceira, a quantidade de ferramentas utilizadas para inserção de todos os componentes e, finalmente, a quarta coluna, informa a quantidade de componentes diferentes, o que se traduz na quantidade de alimentadores utilizados.

Na Tabela 5 são apresentados os resultados encontrados para cada instância, desativando, no algoritmo híbrido, a otimização de cada um dos subproblemas separadamente.

Tabela 5: Soluções geradas forçando limitações no algoritmo híbrido

Instância	Solução inicial (s)	solução	Solução final com otimizações desativa		es desativadas (s)
	final (s)	final (s)	EF	SIC	ACA
Receptor_23	29,59	12,67	28,03	13,96	12,98
Alarme_57	46,10	27,93	42,05	28,405	28,86
Central_315	148,58	133,73	136,49	136,29	135,01

Na Tabela 5, a primeira coluna representa o nome da instância e, a segunda, apresenta o valor da solução inicial gerada pelo algoritmo determinístico. A terceira coluna apresenta o valor da melhor solução encontrada pelo algoritmo híbrido sem aplicar restrições. As colunas quatro, cinco e seis apresentam a solução encontrada pelo algoritmo híbrido, sem a otimização do subproblema, indicado no título da coluna.

Pelos resultados, pode-se observar, para os casos em que há uma quantidade relativamente pequena de componentes, nas instâncias Receptor_23 e Alarme_57, que a regra básica utilizada para gerar a solução inicial (minimizar a ociosidade das ferramentas na cabeça de inserção, maximizando o número de componentes transportados) é ineficiente, fazendo com que a otimização do *setup* seja o subproblema predominante na busca por uma solução de boa qualidade. Paralelamente, os subproblemas da sequência de inserção e alocação de alimentadores tem significância semelhante na composição da solução, visto que as características da instância definem qual dos dois é predominante. Na instância Receptor_23, a sequência de inserção predomina sobre a alocação dos componentes no alimentador, o que é demonstrado pela solução de pior qualidade quando esse subproblema é ignorado. Na instância Alarme_57, alocação dos componentes no alimentador predomina sobre a sequência de inserção. Especificamente na instância Alarme_315, observa-se que não há dominância de nenhum subproblema na composição da reposta final. Essa instância permite visualizar, de forma clara, como os subproblemas estão correlacionados. Caso apenas um desses seja ignorado, o valor da nova solução ótima local é, no mínimo, 1% maior que no caso completo.

A Tabela 5, apesar de permitir a visualização da importância de cada subproblema na composição de uma boa solução final, não permite analisar de forma quantitativa, o peso individual de cada subproblema. Para que essa análise fosse possível, o avaliador de soluções do algoritmo híbrido foi alterado, de modo a quantificar de forma separada os tempos gastos com *setup* e movimentações relacionadas à carga e à inserção.

4.2 Aplicação

Nesta seção é apresentada a aplicação do algoritmo híbrido em dois casos reais. A máquina utilizada foi uma insersora multicabeça, denominada Quadra Basic. Essa máquina é produzida pela *TWS Automation*. O equipamento apresenta como características principais:

- cabeça insersora com até duas pinças;
- capacidade para até 120 alimentadores;
- inserção média nominal de 3.600 componentes por hora.

Essa máquina dispõe de um software muito simples, que não realiza qualquer tipo de otimização no programa de montagem. A ordem dos componentes nos alimentadores é especificada pelo usuário. A sequência de inserção é ordenada, primeiramente em função da ferramenta mais utilizada e, posteriormente pode ser efetuada uma ordenação em função das coordenadas de cada componente em X e Y na placa de circuito impresso. Existe, ainda, a possibilidade da sequência de inserção não ser ordenada, nesse caso a lista oriunda do CAD, com a posição de cada componente na placa de circuito impresso, será considerada como uma sequência de montagem. Na Tabela 6 são apresentados os parâmetros da máquina.

Tabela 6: Caracterização da máquina insersora

Parâmetro	Descrição	Valor	
F_{MAX}	Número máximo de ferramentas	2 ??	
Γ_{MAX}	simultâneas na cabeça de inserção	2 ::	
V_r	Velocidade média da cabeça de	1200 mm/s	
V _r	inserção	1200 1111/3	
T_{i}	Tempo para inserir um componente	0,31s	
T_c	Tempo para carregar um componente	0,31s	
T_f	Tempo para troca de uma ferramenta	1 s	
$F_x F_y$	Coordenadas cartesianas do centro	(50,50) mm	
1 x 1 y	médio do alimentador de ferramentas	· · · ·	
Cx_n, Cy_n	Coordenadas cartesianas da posição	(200,30)-(710,30) mm espaçados a cada 10	
Cx_n, Cy_n	dos componentes no alimentador	mm em x	
	Coordenadas cartesianas do ponto (0,0)		
Bx,By	da PCB em relação à origem da	(400,200) mm	
	máquina		

Na Tabela 6, a coluna da esquerda apresenta os nomes das variáveis consideradas. Na coluna central são descritos o significado físico de cada parâmetro e na coluna direita são apresentados os valores medidos.

A fim de tornar a análise o mais realista possível, o programa de montagem gerado pela meta-heurística foi executado pela máquina e o tempo total de montagem foi comparado ao tempo de montagem gerado pelo software da máquina. Foram abordados dois casos, os quais são apresentados na Tabela 7.

Tabela 7: Otimização de um caso real

Caso	Resultado da	Resultado do algoritmo híbrido (s)		Diferença	Aumento do tempo disponível para
	máquina (segundos)	Média (s)	Desvio padrão	percentual	montagem (mensal)
Caso 1	37,14	33,57	2,35	- 10,6 %	3,18%
Caso 2	75,08	72,16	3,61	-4,0%	0,84%

Na Tabela 7, a primeira coluna à esquerda descreve o nome do caso testado. Foram escolhidos dois casos, o Caso 1 corresponde a uma placa pequena (apenas trinta itens), o que corresponde a 30% do tempo total de produção no mês, porém representa apenas 17% no faturamento da empresa. O Caso 2 corresponde a uma placa média com 72 itens montados. Nesse caso, o produto consome 21% do tempo de produção, porém representa 31% do faturamento mensal. Na segunda coluna é apresentado o tempo de montagem, utilizando o programa gerado pelo software da máquina. Na terceira coluna é mostrado o tempo médio de montagem e o respectivo

desvio padrão, considerando cem execuções do programa de montagem gerado pela meta-heurística. A quarta coluna exibe a diferença percentual entre os tempos de montagem das duas colunas anteriores. A quinta coluna, apresenta quanto tempo disponível no mês foi poupado, utilizando o novo programa de montagem.

Pode-se observar, em ambos os casos, um pequeno ganho em relação ao programa apresentado pela máquina. Isso ocorre devido às características da máquina insersora modelada. A máquina apresenta apenas duas ferramentas na cabeça insersora. Quanto menor o número de pinças insersoras disponíveis, menor é o percentual de otimização, fato que foi constatado ao longo dos testes.

Para a fábrica, em questão, os produtos pequenos, além de apresentarem um baixo valor agregado, correspondem a mais de 60% dos itens montados pela máquina. Para esses itens especificamente, é muito interessante reduzir o tempo montagem o quanto for possível, diminuindo o custo de produção do item, além de liberar mais tempo para montagem de produtos de maior valor agregado. Considerando que o faturamento está limitado pela capacidade de produção da máquina, somente essas duas otimizações permitem elevar o faturamento médio da empresa em 4%.

5 Conclusões

Este trabalho apresentou a aplicação de um algoritmo híbrido ao problema do ciclo de montagem de uma máquina insersora, que considerou quatro subproblemas da área de otimização combinatória, associados ao problema principal do ciclo de montagem de uma máquina insersora automática de componentes multicabeça. Os quatro problemas considerados são: a alocação de componentes no alimentador, a sequência de inserção dos componentes em uma placa de circuito impresso, o plano de carga dos componentes e a otimização do conjunto de ferramentas. O algoritmo híbrido utiliza conceitos das meta-heurísticas Busca Tabu e algoritmos genéticos, sendo objetivo do algoritmo minimizar o tempo total de montagem, elevando a produtividade da máquina.

O problema é definido pela literatura como NP-Difícil, o que motiva o uso de heurísticas e metaheurísticas para a sua abordagem. Encontra-se, na literatura, soluções propostas por diversos autores aplicadas a diversos tipos de máquina, onde modelos específicos foram desenvolvidos e meta-heurísticas foram aplicadas. Grande parte dos trabalhos não contemplam todos os subproblemas associados, concentrando-se em subproblemas específicos.

No estudo do modelo são apresentados os quatro subproblemas associados ao problema principal e como esses foram inseridos no modelo proposto. A construção do modelo visou à abrangência, assim, é genérico e contempla um grande número de configurações mecânicas de máquinas insersoras *multi-head*.

A arquitetura do modelo foi apresentada, visando elucidar como ocorreu o encadeamento entre a Busca Tabu e o algoritmo genético. A validação do algoritmo foi feita particionando o problema principal em subproblemas. Foram testadas instâncias com resultados conhecidos, utilizadas por outros autores, o que permitiu verificar a eficiência da implementação.

Nos experimentos, máquinas reais foram modeladas e problemas reais foram testados no modelo, utilizando a técnica computacional apresentada. Os resultados foram comparados com os fatos reais, testando as soluções geradas em uma máquina insersora real. Soluções de boa qualidade foram geradas e a técnica proposta foi capaz de reduzir custos operacionais e elevar a produtividade da máquina.

Referências

- [1] CRAMA, Y.; KLUNDERT, J. Van de; SPIEKSMA, F. C. R. Production planning problems in printed circuit board assembly. *Discrete Applied Mathematics*, v. 123, n. 1-3, p. 339-361, nov. 2002.
- [2] AYOB, M. Optimisation of surface mount device placement machine in printed circuit board assembly. *The School of Computer Science and Information Technology*. Nottingham: University of Nottingham, 2005.
- [3] CRAMA, Y. Throughput rate optimization in the automated assembly of printed circuit boards. *Annals of Operations Research*, New Jersey, USA, v. 26, n.1-4, p. 455–480, dec. 1990.

- [4] VIANA, G. V. R. Metaheurísticas e programação paralela em otimização combinatória. Fortaleza: UFC Edições, 1998.
- [5] GUPTA, S. K.; RAJAGOPAL, D. Sheet metal bending: forming part families for generating shared pressbrake setups. *Journal of Manufacturing Systems*, Dearborn, Michigan, USA, v. 21, n. 5, p. 392-350, jul. 2002.
- [6] BOUSSAID, I.; LEPAGNOT, J.; SIARRY, P. A survey on optimization metaheuristics. *Information Sciences*, Paris, v. 237, n. 11, p. 82-117, mar. 2013.
- [7] AYOB, M.; KENDALL, G. A new dynamic point specification approach to optimise surface mount placement machine in printed circuit board assembly. *The School of Computer Science and Information Technology* University of Nottingham, 2002.
- [8] GROOVER, M. F. Automation, production systems, and computer-integrated manufacturing. Prentice Hall, Englewood Cliffs, 808 p., 2007.
- [9] KING, J. R. Machine-component grouping in production low analysis: an approach using a rank order clustering algorithm. *International Journal of Production Research*, Milton Park, Inglaterra, v. 18, n.2, p. 213-232, abr. 1980.
- [10] GUPTA, S. K.; RAJAGOPAL, D. Sheet metal bending: forming part families for generating shared press brake setups. *Journal of Manufacturing Systems*, Atlanta, USA, v.21, n. 5, p. 329-350, out. 2002.
- [11] GLOVER, F. *Tabu search a tutorial*. USA: center of applied artificial intelligence-University of Colorado, 1990.
- [12] GLOVER, F.; LAGUNA, M. Tabu search. [S.l.]: Kluwer Academic Publishers, 382 p., 1997.
- [13] STEPANENKO, S. Global optimization methods based on tabu search. Südwestdeutscher Lerlag für Hochschutschriften, 204 p., 2009.
- [14] HOLLAND, J. H. Adaptation in natural and artificial systems. Michigan: University of Michigan Press, Ann Arbor, 1975.
- [15] REEVES, C. Genetic algorithms. In: GLOVER, F. KOCHENBERGER, G. A. *Handbook of Metaheuristics*, Kluwer Academic Publishers, 2003.
- [16] SPANOS, A. C.; PONI, S. T.; TATSIOPOULOS, I. P. A new Hybrid Parallel Genetic Algorithm for the Job-Shop Scheduling Problem. *International Transactions in Operational Research*, Chichester, Inglaterra, v.21, n.3, p. 479-499, 2014.
- [17] BLAZEWICZ, J.; KOVALYOV, M. Y. The complexity of two group scheduling problems. *Journal of Scheduling*, Nova Iorque, v. 5, n. 6, p. 477- 485, nov/dez. 2002.
- [18] XHAFA, F. Metaheuristics for Scheduling in Industrial and manufacturing Applications. Springer-Verlag Berlin Heidelberg, 2008.
- [19] LARUSIC, J; ABRAHAM, P. The asymmetric bottleneck travelling salesman problem: algorithms, complexity and empirical analysis. *Computers & Operations Research*, Atlanta, USA, v. 43, p. 20-35, mar. 2014.