

OTIMIZAÇÃO DO PROCESSO DE INSERÇÃO AUTOMÁTICA DE COMPONENTES ELETRÔNICOS EMPREGANDO A TÉCNICA DE TIMES ASSÍNCRONOS

Cesar Scarpini Rabak

Jaime Simão Sichman

Laboratório de Técnicas Inteligentes (LTI)

Escola Politécnica (EP)

Universidade de São Paulo (USP)

Resumo

Este trabalho apresenta um sistema computacional que busca otimizar o processo de inserção automática de componentes eletrônicos com a máquina Panasert AVK, baseada na técnica de times assíncronos (A-Teams). É feita a formulação do problema de inserção e apresentada a arquitetura do sistema. Por último, são apresentados os resultados obtidos e conclusões inferidas, demonstrando a viabilidade da utilização desta técnica nesta classe de problemas.

Palavras-chave: otimização, inteligência artificial, times assíncronos.

Abstract

This paper presents a computational system which aims to optimize the automatic insertion process of electronic components with the Panasert AVK machine, based on asynchronous teams (A-Teams). A formulation of the insertion problem is proposed and the architecture of the system is presented. Finally, the obtained results and inferred conclusions are presented, showing the viability of this technique when applied in this class of problems.

Keywords: optimization, artificial intelligence, asynchronous teams.

1. Introdução

Máquinas inseridoras de componentes são utilizadas na indústria eletrônica moderna para a montagem automática de placas de circuito impresso. Essas placas formam a parte principal dos equipamentos eletrônicos tanto de entretenimento (tais como rádios, televisores, vídeo cassetes), como de informática (tais como microcomputadores, monitores de vídeo, máquinas de FAX, etc.). A fase de montagem automatizada é uma área onde existem oportunidades para se obter potencialmente grandes economias. No Brasil, em particular, as máquinas de montagem automática (das quais as máquinas inseridoras são um caso particular) têm preço relativamente alto, o que torna o seu custo de operação uma componente importante na matriz de custos de produção dos equipamentos eletrônicos.

O objetivo da otimização desse processo é minimizar o tempo total de inserção, aumentando assim a utilização da máquina inseridora. A otimização é obtida escolhendo-se a ordem apropriada de inserção dos componentes na placa, o que corresponde a resolver uma instância do *problema do caixeiro viajante* (“*Traveling Salesman Problem*” ou abreviadamente TSP) (Lawler *et al.*, 1985), e a colocação mais apropriada dos componentes nos escaninhos, que pode ser enunciado como o *problema de alocação quadrática* (“*Quadratic Assignment Problem*” ou abreviadamente QAP) (Lawler, 1963). Tanto um como outro problema são conhecidos pela dificuldade de solução devido à explosão combinatória das soluções, sendo caracterizados como problemas do tipo NP-difícil (Garey & Johnson, 1979). Esses problemas são caracterizados pela ausência de algoritmos que sejam exatos e rápidos. Os mais rigorosos são lentos ou consomem muitos recursos computacionais, enquanto os heurísticos não são muito robustos, sendo sensíveis à instância do problema: dão bons resultados para alguns casos e falham fragorosamente em outros casos. Assim, no contexto deste trabalho, o termo otimização é empregado indistintamente para descrever o processo de busca da melhor solução para este problema, seja esta exata ou aproximada.

Neste trabalho apresentamos um sistema para otimizar o tempo total de inserção da máquina inseridora de componentes axiais Panasert AVK (Panasonic, 1996), fabricada pela empresa Panasonic Manufacturing Equipment Division, utilizando a técnica denominada Times Assíncronos (“*Asynchronous Teams*” ou abreviadamente A-Teams) (Talukdar *et al.*, 1996).

2. Inserção Automática de Componentes Eletrônicos em Placas de Circuito Impresso

Placas de circuito impresso consistem de placas de suporte não condutor (geralmente plásticos como fenolite ou epóxi), com espessura entre 1,5 a 2,0 mm, contendo trilhas de material condutor (geralmente cobre na sua superfície). Cada placa tem definida pelo seu projeto as posições onde componentes eletrônicos, tais como resistores, capacitores, diodos, transistores ou circuitos integrados, deverão ser montados para que ela execute a sua função no sistema a que será destinada.

Devido à multitudine e diferentes tecnologias com que são fabricados, os componentes eletrônicos adquirem uma grande variedade de formato, tamanho, embalagem. Por outro lado, componentes eletrônicos fisicamente idênticos quanto aos seus aspectos externos podem ser diferentes eletricamente (resistores de valor ôhmico diferente, ou transistores de mesmo encapsulamento mas com características elétricas diferentes, por exemplo). Essa variedade torna a montagem dessas placas uma operação complexa: certas tecnologias de componentes tornam a montagem manual de certos componentes praticamente impossível, enquanto outros tipos de componentes somente podem ser montados manualmente a custos

toleráveis pela indústria no presente estado da tecnologia. São necessárias, em geral, duas etapas na montagem dessas placas: uma manual e outra automatizada. Dependendo do tipo de produto a que a placa de circuito impresso se destina, essa etapa automatizada pode chegar a 100% da fase de inserção.

2.1 Máquinas Insersoras

A indústria eletrônica dispõe de máquinas automáticas que são capazes de efetuar a inserção de certos tipos (classes) de componentes, permitindo que esse processo de montagem seja mecanizado. Essas máquinas podem variar desde robôs de uso geral, empregados para manusear os componentes eletrônicos (quando são providos de pinças e acessórios especiais), até máquinas especializadas, construídas especialmente para esse fim. Essas últimas são denominadas *máquinas insersoras* de componentes eletrônicos.

Existem máquinas insersoras muito distintas que atendem tanto às necessidades da tecnologia específica dos componentes que inserem, como às diferentes abordagens de resolução de problemas tecnológicos que surgem na construção desse tipo de equipamentos.

Entretanto, pode-se observar que todas as máquinas insersoras executam cinco operações fundamentais:

1. *posicionamento para retirada* de um componente de um alimentador;
2. *retirada do componente* do alimentador corrente;
3. *transporte do componente* desde o alimentador até a placa de circuito impresso;
4. *posicionamento para colocação* do componente na placa de circuito impresso;
5. *inserção do componente* na placa de circuito impresso.

O passo 3 pode envolver algum processamento intermediário, como por exemplo, rotação do componente, preparação dos fios (corte e/ou dobra), etc.

Em algumas máquinas, existe apenas um único alimentador de componentes. Neste caso, na fase de preparação os componentes são arranjados em uma única fita de alimentação, em uma ordem predefinida correspondente à sua colocação pela insersora na placa de circuito impresso. Em outra classe de máquinas, os componentes a serem usados na montagem são disponibilizados em alimentadores individuais, um para cada tipo de componente. Neste outro caso, a fita de alimentação já faz parte da forma de condicionamento do componente providenciada pelo fabricante. Esses alimentadores são chamados de *escaninhos* e podem ser montados em conjuntos dispostos tanto de forma linear como circular. Nestas máquinas, o posicionamento desses conjuntos para a retirada de um componente de um alimentador acarreta a movimentação física de uma parte da máquina que contém estes escaninhos. Como esta parte é relativamente pesada, o tempo da operação torna-se mais demorado. O número de escaninhos de uma máquina desse tipo torna-se uma de suas características, já que limita o número máximo de tipos de componente que é possível colocar em uma placa em uma dada operação de inserção.

Ao classificar as máquinas insersoras, utilizamos o termo *ciclo de máquina* para designar a série de operações fundamentais necessárias para levar a cabo a retirada e inserção de um componente. As máquinas são diferenciadas pelos mecanismos utilizados para efetuar essas operações. Dentre as características utilizadas para classificar as máquinas insersoras, a mais importante noção é a da *concorrência*, na qual se distinguem duas categorias:

1. **seqüenciais:** as cinco operações fundamentais descritas acima são executadas seqüencialmente, um ciclo consiste claramente dos passos fundamentais realizados naquela ordem;
2. **concorrentes:** duas ou mais operações fundamentais podem ser realizadas concorrentemente, um mesmo ciclo pode consistir de operações fundamentais referentes à retirada de um componente e à inserção do componente previamente retirado.

Máquinas concorrentes podem ter ferramentas especializadas para retirada e posicionamento de componentes, com mecanismos de transporte entre os dois, ou múltiplas cabeças de inserção, capazes de realizar as operações de retirada e inserção. As próprias operações fundamentais, inclusive, podem ser mais ou menos concorrentes: por exemplo, no posicionamento para colocação da placa de circuito impresso, que é tipicamente realizado por uma mesa XY que a movimenta, os movimentos nos dois eixos X e Y podem também ser concorrentes ou seqüenciais.

A categoria de uma máquina afeta claramente o cálculo de seu tempo de ciclo da máquina. No caso de uma máquina totalmente seqüencial, este tempo será a soma das durações de cada operação fundamental. No caso de uma máquina concorrente, será necessário definir melhor o tempo de ciclo, verificando as operações concorrentes e os momentos em que ocorrem dependências entre elas. Por exemplo, se em uma determinada máquina o posicionamento da placa pode ser feita em paralelo com a retirada do componente do alimentador, o tempo de ciclo conterà um termo com o máximo entre esses dois tempos, uma vez que a inserção na placa requer que *ambas* as operações já tenham ocorrido; em outras palavras, é necessário estudar as precedências das operações fundamentais dessas máquinas para formular os seus tempos de ciclo.

2.2 A Máquina Inersora Panasert AVK

A inersora Panasert AVK é uma máquina de fabricação da Panasonic para inserção de componentes *through hole* axiais, isto é, aqueles que têm seus fios alinhados com seu eixo longitudinal (tais como diodos, resistores, capacitores, supressores de transientes, etc.), em placas de circuito impresso que tenham furos para acomodar esses fios e receber solda pelo lado oposto àquele por onde esses componentes tenham sido posicionados na placa. Conseqüentemente, para efetuar uma inserção de um componente a AVK precisa também dobrar esses fios, cortá-los e ao término da operação de inserção dobrá-los novamente pela parte inferior da placa para estabilizar o componente até que este seja soldado em uma fase posterior do processo produtivo.

Os blocos principais da inersora AVK são: uma única cabeça de inserção de componentes, um mandril que se move entre o ponto de retirada de componentes (fixo na máquina) e o ponto de inserção (na área da placa) transportando o componente do escaninho para a cabeça de inserção, dois conjuntos de escaninhos para alojar os componentes e uma mesa XY para movimentar a placa de circuito impresso sob a cabeça de inserção conforme mostra a Figura 1. Uma unidade de controle comanda todo o conjunto, interpretando o programa de inserção para cada placa. Essa unidade de controle tem capacidade para 8 programas e uma memória total de 2000 posições. Assim pode-se ter desde um único programa com 2000 inserções até 8 programas cuja soma não exceda a memória total. Outrossim, caso já estejam armazenados oito programas na memória, o restante da memória fica indisponível mesmo que a quantidade de memória por eles ocupada não exceda a memória total.

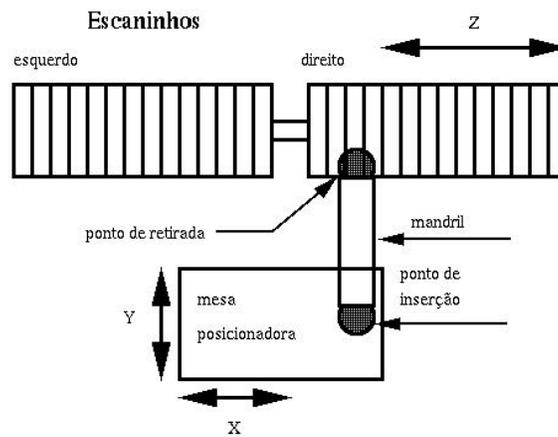


Figura 1 – Blocos Principais da Inseridora AVK

Os componentes eletrônicos a inserir podem ser alojados em carretéis (considerado forma opcional pelo fabricante) ou em dispensadores feitos de caixas de papelão (porta-fitas), denominados de *flat pack*. Neste último caso, os componentes são dispostos de forma equidistante em fitas de papel ou material plástico flexível e acondicionados nessas caixas de forma similar a formulários contínuos. Essas caixas por sua vez são alojadas em escaninhos providos na AVK. As fitas de componentes saindo desses escaninhos são conduzidas por guias de alumínio fazendo que elas façam um giro de 90 graus até atingir certas fendas que sujeitam as fitas de componentes para que o mandril possa sacar os componentes quando estiver alinhado com a fenda, isto é, esse determinado componente é aquele pronto para inserção.

Os escaninhos estão divididos em dois grupos, um denominado *esquerdo* “ZL” e outro denominado *direito* “ZR”, e se movimentam por meio de servomotores. Cada um destes grupos contém 60 fendas distintas.

A AVK pode receber fitas de componentes tendo duas larguras: 26 mm ou 52 mm. Quando trabalha com componentes de 52 mm, cada quatro porta-fitas de componentes ocupam seis fendas de escaninhos para acomodar a largura maior. Neste caso, o operador da máquina pode escolher livremente qual(is) fenda(s) que deverá(ão) ficar vaga(s). Dessa forma, no limite cada grupo pode operar ou com 40 componentes nos escaninhos ocupados com componentes de 52 mm (80 ao todo) ou com 60 componentes nos escaninhos ocupados com componentes de 26 mm (120 ao todo). O movimento da placa de circuito impresso é realizado pela mesa XY com movimentos independentes e simultâneos nos dois eixos.

Esta máquina é uma máquina *concorrente*, pois na AVK o posicionamento do conjunto de escaninhos e o movimento da mesa XY (e nesta em cada eixo independentemente) são operações concorrentes. Por outro lado, as operações de retirada e transporte do componente e a inserção são aquelas que ultimadamente dão a *cadência* da máquina; em particular, a AVK pode ser adquirida em duas versões, uma com cadência de 5 componentes por segundo e outra com 2,5 componentes por segundo. Essa cadência é definida pelo fabricante para as seguintes condições ditas normais: a distância (em X ou Y) da posição anterior para a próxima inserção seja menor ou igual a 30 mm, a diferença de medida de largura de dobra dos fios dos componentes seja menor ou igual a 5 mm e a troca de escaninho não ultrapasse um adjacente.

O fato da mesa XY ter seus eixos independentes implica que para um dado deslocamento, o tempo gasto nesse deslocamento será determinado pela maior distância (X ou Y), sendo portanto dado pela *distância de Chebychev*, dada por $\max(|x_2 - x_1|, |y_2 - y_1|)$.

Para o usuário, programar a insersora significa gerar um plano de atribuição de componentes a cada escaninho, e um programa de inserção, indicando em que posição cada componente vai ser inserido. O programa deve incluir também a largura da dobra, a altura de inserção, o giro (em múltiplos de 90 graus, apenas) e se necessário um *offset* para ângulos não múltiplos de 90 graus.

2.3 Análise dos Tempos de Operação da AVK

Conhecendo as operações fundamentais da máquina insersora AVK, precisamos determinar o seu tempo de ciclo de máquina, ou seja, o tempo necessário para efetuar a inserção de um componente na placa. Vamos utilizar a seguinte notação:

- n : o número total de componentes a ser inseridos;
- i : índice para os ciclos da máquina, ou seja execução de uma instrução de inserção da máquina vindo da linha (de programa) $i - 1$ para a linha i ;
- c_1 : tempo de cadência da máquina (constante que indica o tempo mínimo de ciclo);
- c_2 : tempo de carga de uma placa na mesa XY (constante);
- $x(i)$: tempo para deslocar a mesa XY da posição x_{i-1} para posição x_i ;
- $y(i)$: tempo para deslocar a mesa XY da posição y_{i-1} para posição y_i ;
- $z(i)$: tempo para deslocar o conjunto de escaninhos da posição z_{i-1} para posição z_i ;
- $w(i)$: tempo para modificar a abertura da pinça que dobra os fios dos componentes da largura w_{i-1} para a largura w_i .

O tempo de duração de um ciclo i vale:

$$\tau(i) = \max(c_1, x(i), y(i), z(i), w(i)) \quad (1)$$

O tempo total de inserção de uma placa será então:

$$T_o = \sum_{i=1}^n \tau(i) + c_2 \quad (2)$$

O tempo de inserção e o deslocamento da cabeça do ponto de retirada dos componentes até o ponto de inserção são tão pequenos frente aos outros tempos que ficam absorvidos no tempo c_1 .

3. Otimização

Neste capítulo, abordamos o problema de inserção automática de componentes eletrônicos sob o prisma de um problema de otimização, formulando matematicamente o problema de otimização a ser solucionado.

3.1 Análise Qualitativa do Problema de Otimização de Inserção

Analisando-se a operação da insersora e a equação de tempo total de inserção de uma placa de circuito impresso como exemplificado na equação (2), fica claro que a **seqüência** de inserção influi no tempo total de inserção, uma vez que uma escolha infeliz pode levar a mesa posicionadora a fazer grandes movimentos entre inserções seguidas, onerando o tempo total de inserção. Por outro lado, vê-se que a atribuição dos tipos de componentes aos escaninhos para uma dada seqüência de inserção também influi no tempo total, pois se entre duas inserções o escaninho anterior àquele contendo o componente a inserir estiver muito afastado deste último, a insersora terá seu tempo de inserção dominado pelo tempo de troca de escaninhos. Como já observamos na seção anterior, dado que o conjunto de escaninhos é o bloco (móvel) com maior massa na insersora, é de se esperar que esse movimento da máquina seja mais lento.

Portanto, o que se deve buscar, em última análise, é o melhor “casamento” entre os tempos obrigatórios devidos aos deslocamentos da mesa posicionadora com os tempos obrigatórios devidos às trocas de escaninhos, de modo a se ter as operações concorrentes com tempos os mais próximos possíveis.

Nota-se também que a escolha da melhor atribuição dos componentes aos escaninhos é condicionada pela seqüência de inserção na placa, e por outro lado a melhor seqüência de inserção pode ser diferente do mínimo percurso (planar) na placa entre os pontos de inserção devido ao custo de troca de escaninhos (Figura 1). A melhor seqüência de inserção é condicionada portanto tanto pela atribuição dos componentes aos escaninhos, quanto pela seqüência de inserção.

A determinação do menor percurso, ou seja, da melhor seqüência de inserção levando em conta os custos (tempos) dos ciclos, conforme a equação (1), pode ser formulada como um problema do tipo do TSP, enquanto a atribuição dos componentes aos escaninhos que minimize o tempo total de inserção pode ser formulada como um problema do tipo do QAP.

3.2 Formulação do Problema de Otimização de Inserção

A função objetivo a otimizar (minimizar) é o tempo total de inserção de uma determinada placa, matematicamente:

$$\min f(p, q) = \sum_{i=1}^n \tau(i) + c_2 \quad (3)$$

onde p é um percurso na placa, q uma atribuição de componentes aos escaninhos e $\tau(i)$ o tempo de ciclo da máquina para uma operação de inserção i dados p e q , conforme a equação (1).

Sejam as seguintes definições:

- n : número total de inserções na placa ($n \leq 2000$);
- m : número de tipos de componentes distintos a serem utilizados nas placas ($m \leq 120$);
- v_{ij} : variável de decisão que representa o percurso da posição i para a posição j (isto é o movimento em x e y) na placa de circuito impresso ($0 \leq i, j \leq n$);

z_{kl} : variável de decisão que representa a colocação do tipo de componente k no escaninho l ;

$t(i)$: função que tem como entrada uma posição i na placa e retorna o tipo de componente que deve ser inserido nesta posição.

O problema é então minimizar a quantidade:

$$T = \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^m \sum_{l=1}^m c_{ijkl} v_{ij} z_{t(i)k} z_{t(j)l} \quad (4)$$

sujeita a:

$$v_{ii} = 0, \quad 1 \leq i \leq n \quad (5)$$

$$v_{ij} \in \{0,1\}, \quad 1 \leq i, j \leq n \quad (6)$$

$$\sum_{i=1}^n v_{ij} = 1, \quad 1 \leq j \leq n \quad (7)$$

$$\sum_{j=1}^n v_{ij} = 1, \quad 1 \leq i \leq n \quad (8)$$

$$u_i - u_j + n v_{ij} \leq n - 1, \quad 2 \leq i, j \leq n \quad (9)$$

$$z_{kl} \in \{0,1\}, \quad 1 \leq k, l \leq n \quad (10)$$

$$\sum_{k=1}^m z_{kl} = 1, \quad 1 \leq l \leq m \quad (11)$$

$$\sum_{l=1}^m z_{kl} = 1, \quad 1 \leq k \leq m \quad (12)$$

onde:

c_{ijkl} = é tempo necessário a uma inserção, igual ao tempo de duração de ciclo da máquina $\tau(i)$;

$v_{ij} = 1$ se e somente se existe movimento da posição i para a posição j ;

$z_{kl} = 1$ se e somente se o componente no escaninho l é k ;

u_i, u_j são números reais arbitrários, diferentes entre si.

A função objetivo (4) acumula o tempo total de inserção. As equações (7) a (9) são as restrições do TSP (Reinelt, 1994), as equações (11) a (12) são restrições do QAP (Pardalos *et al.*, 1994). Note-se que esses dois problemas são bem imbricados, não havendo uma maneira de se desacoplar o TSP do QAP, para se tentar resolvê-los individualmente.

Esta formulação é similar à encontrada por Leipälä & Nevalainen (1989) ao analisar uma máquina Panasert RH e àquela relatada por Walas & Askin (1984) ao estudar uma prensa puncionadora com um magazine rotativo de ferramentas. Nossa formulação difere da encontrada por Broad *et al.* (1996), pois a máquina estudada neste caso (Panasonic MPa) possui escaninhos tanto na parte da frente como na parte de trás do porta placas (mesa XY). Assim, a cabeça tem que efetuar um movimento mais complexo, necessitando levar em conta no equacionamento os arcos de retorno da cabeça a um dos dois conjuntos de escaninhos.

3.3 Abordagem ao Problema de Otimização

Como podemos observar pela formulação acima, o problema de otimização da AVK é um problema de otimização discreta, que somente pode ser abordado por algoritmos heurísticos, uma vez que pelo tamanho do TSP (até 2000 cidades) e do QAP (120) gerados, não é exequível o uso de enumeração total. O que se necessita, então, é uma maneira de organizar os algoritmos para que operem com maior sinergia possível, permitindo que cooperem entre si, obtendo soluções que não seriam encontradas por cada um deles isoladamente (Souza & Talukdar, 1993).

Neste trabalho, optou-se por utilizar a técnica de times assíncronos (A-Teams). Trata-se de uma nova técnica para resolução de problemas baseada na utilização simultânea e assíncrona de diversos agentes, formados por algoritmos que cooperam entre si. Tal técnica é descrita na seção seguinte.

4. Times Assíncronos (A-Teams)

Um time assíncrono (A-Team) é um arranjo de algoritmos (também denominados *agentes*) de forma que possam cooperar para atingir um objetivo desejado pelo seu projetista, sem que haja necessidade de uma coordenação prévia entre esses algoritmos. A única comunicação entre esses algoritmos se dá através de soluções compartilhadas, armazenadas em repositórios denominados *memórias*. O fato de não requererem coordenação permite que se organizem A-Teams onde os tempos de processamento dos vários agentes sejam bem distintos, possibilitando assim que se possam combinar algoritmos mais velozes que produzem soluções mais fracas com aqueles que, embora tenham como saída soluções de melhor qualidade, são normalmente mais lentos.

A arquitetura de uma solução utilizando A-Teams é construída empregando-se apenas dois elementos básicos, agentes e memórias (Talukdar *et al.*, 1996), conforme a Figura 2:

- **Agentes** são unidades autônomas, cujo sistema de controle é totalmente autocontido. Não aceitam nenhuma instrução de seleção ou escalonamento de outros agentes ou qualquer outro tipo de elemento de controle externo. O agente tem três componentes: um operador (o algoritmo que gera a solução), um sistema de comunicação que estabelece como ele vai escrever e ler nas memórias compartilhadas, e um sistema de controle que determina quando e como vai trabalhar;
- **Memórias** são coleções de soluções que podem ser lidas ou depositadas (geradas) pelos agentes. Os agentes lêem as memórias para ter uma solução inicial a partir da qual podem gerar novas soluções, que por outro lado tornam-se disponíveis para os outros agentes que têm acesso a estas memórias.

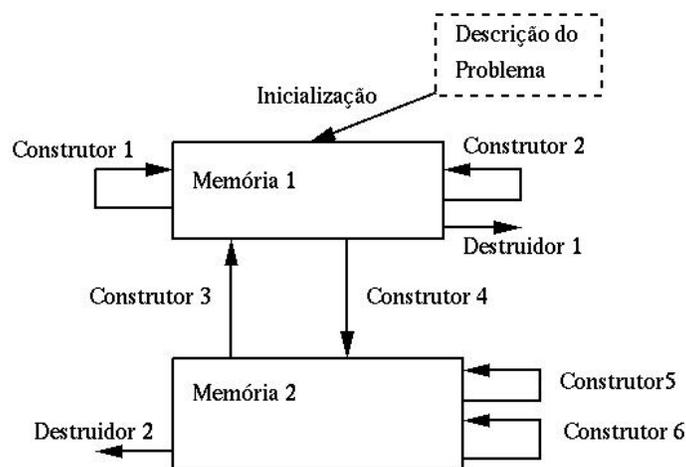


Figura 2 – Arquitetura de um A-Team

Os agentes podem ser de dois tipos: *construtores* e *destruidores*. Construtores são aqueles que colocam novas soluções (tentativas) nas memórias. Destruidores apagam soluções tentativas das memórias eliminando assim as menos apropriadas. Os destruidores podem ainda ser empregados para eliminar padrões de produção de soluções indesejadas tais como seqüências de soluções repetidas.

A-Teams podem ser vistos como uma rede de memórias e agentes. A propriedade fundamental dessa rede é o fato dos seus elementos serem *fortemente cíclicos*. Usando grafos para representar A-Teams, temos os arcos representando os agentes e as memórias representadas pelos vértices (Kang & Talukdar, 1997).

Soluções utilizando A-Teams já foram testadas em uma variedade de problemas de simulação (Talukdar *et al.*, 1996), e apresentam o seguinte comportamento em comum:

1. *Diversidade*: a qualidade das soluções melhora com a faixa de habilidades dos agentes construtores;
2. *Escala*: há pouca, se alguma, penalidade por excesso na habilidade dos construtores. Ao contrário, a escalabilidade parece ser o caso comum, já que a qualidade das soluções invariavelmente pode ser melhorada pela adição de agentes, tanto construtores como destruidores;
3. *Modularidade*: adicionar agentes autônomos a fluxos de dados fortemente cíclicos é relativamente fácil, não importando se os agentes são grandes ou pequenos, gerais ou especializados;
4. *Dualidade*: destruição apta pode compensar por construção inepta e vice-versa;
5. *Tamanho da População*: a qualidade das soluções tende a melhorar com o aumento do tamanho das populações de soluções, embora haja uma tendência à saturação;
6. *Paralelismo*: a velocidade de produção das soluções melhora conforme mais processadores (ou computadores) sejam adicionados até que haja processadores suficientes para todos os agentes trabalhar em paralelo todo o tempo.

A técnica de A-Teams tem sido pesquisada desde 1990, quando foi feita a primeira aplicação com sucesso, e tem sido empregada na solução de problemas de otimização como o próprio TSP, projeto de arranha-céus, robôs por demanda para o ônibus espacial, diagnóstico e controle de sistemas de energia elétrica e escalonamento de trens. Uma descrição mais completa com as devidas referências pode ser encontrada em Talukdar *et al.* (1996).

4.1 O Protocolo de Projeto de A-Teams

Ainda não se dispõe de uma teoria abrangente para orientação no processo de análise e síntese de A-Teams. Os passos propostos a seguir derivam da experiência acumulada pelos experimentadores da técnica (Peixoto & Souza, 1994):

1. *Escolha do Problema*

Determinar em qual classe de problemas de otimização a solução buscada se encaixa.

2. *Decomposição em subproblemas*

A decomposição não precisa ser hierárquica. Os subproblemas não precisam ser disjuntos ou distintos. Eles podem ter alguma sobreposição, ou alguma interação mais complexa. Os subproblemas devem ser escolhidos de modo que possam ser conectados por agentes em laços fechados, onde cada subproblema possa ser usado para construir soluções para o próximo subproblema no laço.

3. *Atribuição de uma ou mais memórias para cada subproblema*

O propósito das memórias é guardar uma população das soluções tentativas de seu subproblema. Populações maiores levam a melhores soluções, mas os benefícios caem rapidamente. Populações de tamanho moderado têm desempenho esperado tão bom quanto as de muito maior tamanho.

4. *Seleção dos algoritmos ou operadores para cada subproblema*

Quanto maior a faixa das capacidades dos algoritmos, melhores serão as soluções encontradas. Não há necessidade de serem uniformes seja no tamanho ou na cobertura. Alguns podem ser grandes, outros pequenos, alguns gerais, outros especializados. O mais importante é a variedade de abordagens.

5. *Transformação de cada algoritmo em um agente autônomo*

Um agente pode ser entendido tendo três componentes: um operador, um sistema de comunicação, e um sistema de controle.

6. *Formação de destruidores*

Deve haver um balanço entre construção e destruição de soluções tentativas, caso contrário as memórias rapidamente ficarão saturadas. Destruidores têm mais duas funções: tornar as soluções que caem no seu espaço de saída quase inacessíveis aos construtores e filtrar padrões de construção indesejados, reconhecendo-os e eliminando-os.

7. *Montagem dos agentes e memórias em fluxos de dados fortemente cíclicos*

8. *Testes e modificação dos fluxos de dados*

O processo de resolução tem como primeiro passo a inicialização das memórias com populações iniciais de soluções tentativas, ativação dos agentes e monitoramento das mudanças nas populações de soluções. Se as soluções convergem lentamente, deve-se retornar ao passo 4. Se mesmo assim a convergência for lenta, deve-se reiniciar o passo 2.

4.2 Adequação da Técnica ao Problema de Otimização de Inserção

A adequação da técnica de A-Teams ao problema da inserção automática de componentes eletrônicos fica agora nítida. Trata-se de um problema de otimização complexo, que mesmo quando dividido em subproblemas somente pode ser solucionado por algoritmos heurísticos, que são sabidamente sensíveis à instância do problema. No caso de A-Teams, pode-se combinar um número de algoritmos para tentar cobrir uma faixa maior de casos. Ela é ainda atrativa pelo fato de ter simplicidade na formulação, permitindo que uma vez que se tenha escolhido uma ou mais representações para o problema ou subproblemas, e eleito os algoritmos mais promissores, já se possa trabalhar nas soluções.

Do ponto de vista da implementação, a técnica permite a reutilização de algoritmos já desenvolvidos e testados por outros pesquisadores. É evidente que deve-se proceder à etapa de transformação destes algoritmos em agentes. Tal esforço, todavia, é muito inferior àquele que seria despendido para criar os algoritmos desde o início. Um outro efeito positivo é o de permitir uma heterogeneidade em termos de linguagens de programação utilizadas no desenvolvimento dos agentes. Um último ponto a ressaltar é a extrema modularidade da técnica: na ausência de uma solução aceitável, basta tentar adicionar novos agentes, e com esforço reduzido, verificar se uma solução melhor é obtida.

5. O Sistema OPTIMA

O sistema OPTIMA (Rabak, 1999), como mostra a Figura 3, é composto de um A-Team projetado para resolver o problema geral de otimização de inserção de componentes da AVK.

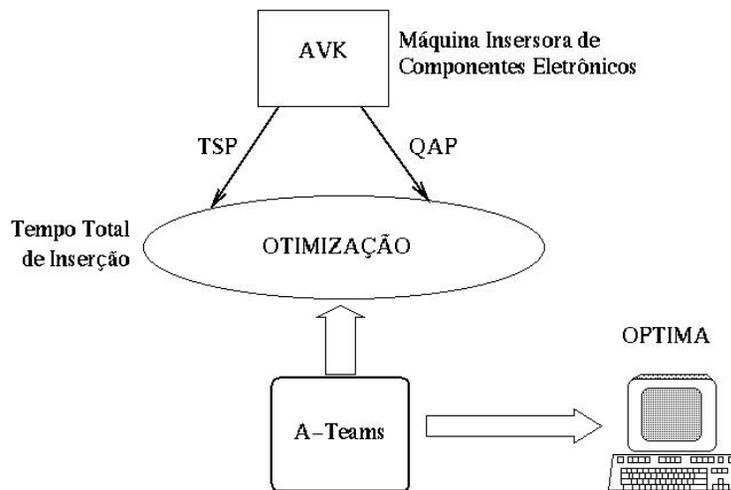


Figura 3 – Visão Geral do Sistema OPTIMA

5.1 Arquitetura do Sistema OPTIMA

A implementação do OPTIMA ficou muito flexível, permitindo que facilmente se incorporem mais agentes, necessitando-se apenas que seja feita uma pequena adaptação,

como descrito a seguir. Note-se que devido aos agentes serem programas autônomos, o OPTIMA permite que se tenha agentes programados em linguagens diferentes, simplificando ainda mais a integração no sistema.

Basicamente, na arquitetura do OPTIMA, utilizaram-se dois agentes que resolvem o TSP e outros dois que resolvem o QAP. Tais agentes se comunicam através de uma memória única, que contém as soluções parciais obtidas por cada um destes algoritmos, cujo detalhamento é mostrado a seguir.

5.1.1 Agentes

O tipo de TSP que se formula ao resolver o problema de otimização da inserção automática na AVK é um TSP simétrico pois a distância (custo) de um nó i para um nó j é a mesma de j para i (ou seja $d_{ij} = d_{ji}$ para $\forall i, j \in 1, 2, \dots, n$), e também se verifica a desigualdade triangular, pois têm-se que $d_{ik} \leq d_{ij} + d_{jk}$ para três i, j, k quaisquer onde d_{ab} é a distância entre a e b (Lawler *et al.*, 1985).

Estamos utilizando no OPTIMA dois algoritmos para resolver o TSP, sendo uma implementação do algoritmo Lin-Kernighan, empregando movimentos 3-opt como elemento básico e número limitado em avanço de submovimentos (Reinelt, 1994), e uma implementação do algoritmo *greedy* (Rardin, 1998).

O QAP formulado para analisar o problema de otimização da inserção automática na AVK é assimétrico, já que a matriz de fluxo representa o número de trocas (saltos) entre escaninhos, e a troca de um escaninho para outro é condicionada pela direção do percurso de inserção na placa. Assim, é possível haver fluxo somente de um escaninho para um consecutivo, por exemplo. É também do tipo euclidiano, uma vez que a matriz de distâncias é simétrica e a desigualdade triangular também se verifica.

Os algoritmos utilizados para resolver o QAP nesta implementação foram o CRAFT (Armour & Buffa, 1963) e o HGW (West, 1983), ambos algoritmos heurísticos por melhoria.

Os algoritmos escolhidos resolvem instâncias canônicas dos subproblemas TSP e QAP. Para fazer uso deles, é necessário adaptar cada algoritmo para que possa receber dados (da instância) do problema da AVK. Para a obtenção dos subproblemas TSP e QAP temos que processar a definição do problema, dada inicialmente no formato de arquivo de programa de inserção da insersora AVK, para obter alguns parâmetros dos problemas, como a matriz de distância para o TSP e as matrizes de distância e fluxo para o QAP.

A matriz de distâncias para o TSP é obtida empregando-se a equação (1), e considerando os $n(n - 1)/2$ custos para os possíveis pares de inserção, dado que a operação da máquina faz com ela tenha custos (tempos) simétricos.

No caso do QAP temos de considerar as matrizes de distâncias e de fluxo. A matriz de distâncias sempre será uma tabela com diagonal nula indicando o tempo de trocas entre escaninhos, sendo monotonicamente crescente para escaninhos mais distantes entre si. Em geral esta matriz é fixa para a máquina, exceto se os operadores efetuarem ajustes na insersora que modifiquem suas velocidades ou qualquer outro parâmetro que influa no tempo de troca de escaninhos.

A matriz de fluxo será uma função do caminho que a máquina insersora vai percorrer na placa, dando como resultado quantas vezes ocorrem trocas entre dois determinados escaninhos. A diagonal pode ser não nula, quando as inserções sucessivas ocorrem com componentes do mesmo tipo, usando-se o mesmo escaninho. Como esses casos não

contribuem com custo para a solução do QAP, zera-se essas posições na extração das matrizes, tornando-as adequadas para os algoritmos de QAP.

Os algoritmos devem então estar munidos de um escalonador (*scheduler*) que busque instâncias do problema para processar, ative-os e execute critérios de parada, e um *seletor* para escolher as soluções da memória que irá modificar (ou eventualmente destruir se for um agente destruidor).

O destruidor monta uma tabela das soluções existentes e as ordena (inversamente) pelo tempo total calculado usando o modelo da máquina AVK. Quando o número de soluções excede um valor predeterminado pelo usuário, o destruidor apaga aquelas que tenham os maiores tempos totais de inserção, eliminando assim as menos promissoras e abrindo espaço na memória do A-Team. Outra função do destruidor é identificar soluções repetidas e apagá-las da memória.

5.1.2 Memória

No OPTIMA a memória é um diretório cujo nome passa a ser o da corrida de otimização, e onde as soluções são armazenadas em arquivos.

Devido a não ser necessário utilizar algoritmos que necessitem de representações parciais (como por exemplo o deconstrutor proposto por Souza (1993) ou o Held-Karp (Held & Karp, 1970) para o TSP), que necessitariam de diferentes memórias para armazená-las, a arquitetura do A-Team para o OPTIMA permanece com uma única memória e os agentes, descritos na seção anterior, modificam as soluções ali depositadas, conforme mostra a Figura 4.

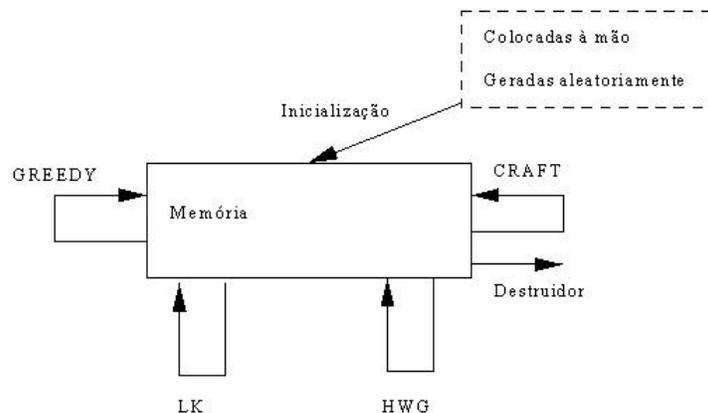


Figura 4 – Arquitetura do Sistema OPTIMA

5.2 Interface com o Usuário

Na implementação do OPTIMA, os agentes são programas que são lançados em função das escolhas efetuadas numa Interface Homem Máquina (IHM), conforme mostra a Figura 5. Neste painel, o usuário pode determinar o número de instâncias que deseja ter na semente automática, o problema a tratar, etc. Embora a arquitetura do OPTIMA use uma única memória, para cada corrida cria-se um diretório com nome diferente, de modo a possibilitar a análise e resultados de parametrizações diversas.

O sistema está munido de um visualizador, mostrado na Figura 6, que permite verificar o percurso de inserção na placa de circuito impresso descrita pelo arquivo de entrada do problema para as várias soluções depositadas pelos agentes na memória do A-Team do OPTIMA. Pode-se ainda verificar o tempo de inserção, o número de trocas de escaninhos, e as distâncias percorridas nas direções X e Y.

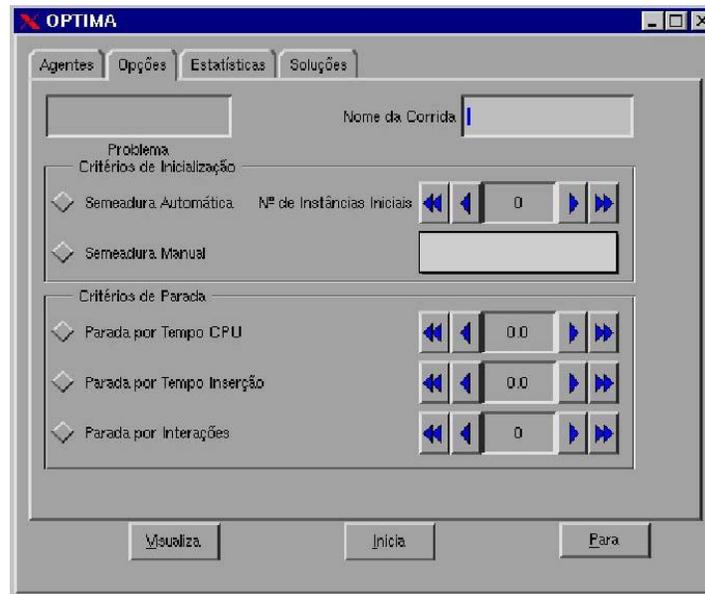


Figura 5 – IHM do OPTIMA. Painel de opções

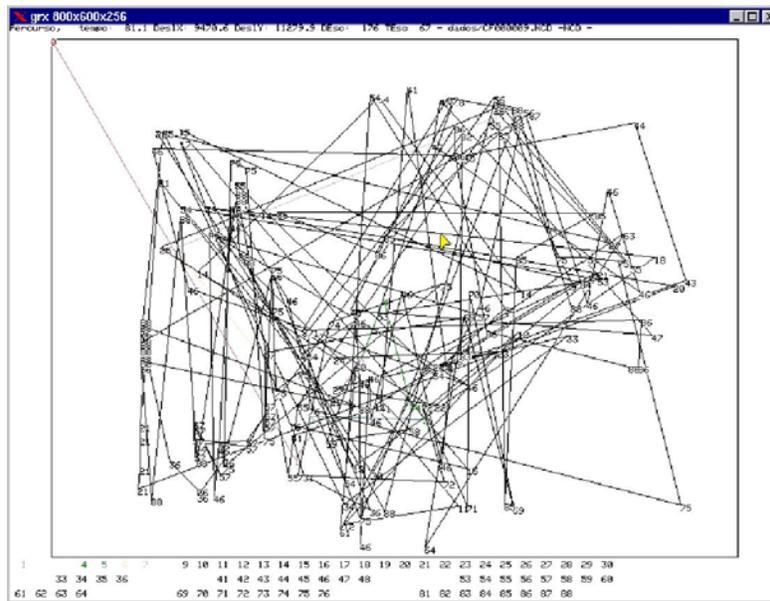


Figura 6 – IHM do OPTIMA. Visualizador

O usuário pode iniciar uma corrida de otimização no OPTIMA usando dois critérios de inicialização: semente de soluções iniciais e reutilização de soluções de uma outra corrida.

Quando é escolhida a semente de soluções na inicialização do A-Team, soluções iniciais são geradas aleatoriamente designando atribuições de tipos de componentes aos escaninhos e seqüências de inserção, de modo a criar uma população mínima para os agentes iniciarem seu trabalho. O sistema permite que se escolha o número inicial dessas soluções iniciais.

Na reutilização, inicia-se o sistema com os resultados de uma otimização anterior, sem modificar os arquivos existentes na memória.

Quanto ao critério de parada, podem ser utilizadas três estratégias distintas:

- limitar o tempo total de processamento do OPTIMA nos computadores escolhidos para a corrida;
- estabelecer o tempo total de inserção na insersora AVK (do problema), sendo que o OPTIMA para ao encontrar a primeira solução com tempo menor ou igual ao lançado nesse parâmetro;
- limitar o número de ciclos dos agentes, de modo a parar seu processamento quando esse limite é atingido.

6. Resultados

A seguir apresentamos os resultados obtidos a partir da otimização de oito problemas reais, e de dois problemas criados sinteticamente para poder avaliar a qualidade das soluções por inspeção no visualizador.

6.1 Problemas Reais

Os problemas reais foram cedidos por um fabricante de placas de circuito impresso em regime OEM, e representam placas típicas que são montadas com a AVK. A configuração de tais problemas está descrita na Tabela 1.

Tabela 1 – Casos de Teste Reais

Placa	Componentes	Escaninhos
CF00000	237	21
CF00003	194	66
CF00005	113	60
CF00007	183	93
CF00011	149	14
CF00013	120	48
CF00015	185	32
CF00019	104	37

Como visto anteriormente, o número de componentes indica o tamanho do TSP, e o de escaninhos o tamanho do QAP, a serem resolvidos para otimizar a inserção nessas placas.

Na Tabela 2, são exibidos os resultados para estes casos de teste. Nas colunas desta tabela, indicam-se respectivamente o tempo inicial com a configuração do fabricante, o tempo de inserção obtido com o OPTIMA, utilizando apenas os agentes para resolver o TSP, a redução percentual correspondente, o tempo de inserção obtido com a adição dos algoritmos para resolver o QAP, sua contribuição para a redução percentual correspondente, e finalmente a redução de tempo percentual total em relação ao tempo de inserção do fabricante.

Tabela 2 – Resultados Iniciais Obtidos para os Casos de Teste Reais

Placa	Fabricante [s]	TSP [s]	% TSP	TSP + QAP [s]	% QAP	% Total
CF000001	81.97	80.19	2.17	80.19	0.00	2.17
CF000003	81.25	72.53	10.74	72.30	0.31	11.01
CF000005	45.60	42.22	7.40	42.09	0.31	7.69
CF000007	77.53	69.96	9.76	69.50	0.66	10.36
CF000011	55.09	52.35	4.97	52.29	0.10	5.07
CF000013	53.93	46.31	14.12	46.28	0.08	14.18
CF000015	68.48	65.05	5.01	64.96	0.13	5.14
CF000019	42.72	38.70	9.41	38.46	0.61	9.96

Uma primeira análise desses dados mostra que o tempo de inserção obtido com o OPTIMA foi inferior ao fornecido pelo fabricante em todos os casos. Além disso, observou-se que ao tratar algumas instâncias o OPTIMA foi mais bem sucedido do que ao tratar outras, sendo que a redução média de tempo de inserção para todas as placas foi da ordem de 7,8%.

Contudo, devemos ressaltar que essas placas, por estarem em produção já há bastante tempo, são também produto de um procedimento de otimização manual, envolvendo várias horas de trabalho de técnicos. Em particular, as placas com tempos de inserção mais otimizados correspondem àquelas de produção mais freqüente pelo fabricante, com conseqüente aumento de sintonia no programa de inserção.

Tabela 3 – Resultados Intermediários para os Casos de Teste Reais

Placa	Inicial [s]	OPTIMA [s]	% Total
CF000001	123.06	80.19	34.8
CF000003	107.49	72.30	32.7
CF000005	61.52	42.09	31.6
CF000007	102.16	69.50	32.0
CF000011	74.89	52.29	30.2
CF000013	68.16	46.28	32.1
CF000015	98.54	64.96	34.1
CF000019	55.76	38.46	31.0

Para esclarecer esse ponto de vista, montamos a Tabela 3 onde exibem-se o tempo inicial correspondente ao tempo médio obtido pelo embaralhamento aleatório das permutações das linhas do programa e da atribuição dos escaninhos, o tempo obtido pelo sistema, e a redução percentual do tempo de inserção. Esse tempo inicial pode ser encarado como o tempo de

inserção que seria obtido por um operador iniciante procurando fazer o primeiro programa correto para inserção da placa, usando, por exemplo, a ordem de aparecimento dos componentes em uma lista de material e lançando as coordenadas dos componentes, valores de largura de dobra, etc. Comprova-se nesta tabela que as reduções percentuais obtidas com o sistema são respeitáveis, habilitando-o a produzir soluções de qualidade sem a necessidade da concentração de esforços de especialistas disponíveis na indústria.

Outra questão que a Tabela 2 ressalta é a aparente pouca contribuição dos algoritmos de QAP para a melhoria da solução. Numa primeira análise, fomos levados a pensar que a qualidade das soluções obtidas dos algoritmos utilizados não fosse suficiente, chegando a considerar a hipótese de lançar mão de mais algoritmos. Contudo, a análise da operação individual dos mesmos indicava que quando calculavam os custos pelo produto das matrizes de fluxo e distância achavam melhorias bem mais altas, sendo que em alguns casos estas ultrapassavam os 20%.

A análise dessa questão indica que quando se extrai essas matrizes da forma exposta na seção anterior, não se considera a informação referente à oportunidade de aproveitar o tempo devido às operações concorrentes. Para tentar elucidar essa questão, modificou-se o algoritmo CRAFT para que ele calculasse o impacto das permutações usando a equação (1) do tempo de duração de ciclo da insersora ao invés de utilizar o custo calculado pelas matrizes de fluxo e de distância. Os resultados desses testes encontram-se na Tabela 4. Vê-se que a utilização do algoritmo adaptado dessa forma consegue melhorar o tempo de inserção em 11,1%, em média.

Tabela 4 – Resultados Finais para os Casos de Teste Reais

Placa	Fabricante [s]	OPTIMA [s]	% Total
CF000001	81.97	77.55	5.4
CF000003	81.25	69.12	14.9
CF000005	45.60	40.62	10.9
CF000007	77.53	66.18	14.6
CF000011	55.09	49.68	9.8
CF000013	53.93	44.50	17.5
CF000015	68.48	64.46	5.9
CF000019	42.72	37.88	11.33

6.2 Problemas Sintéticos

Adicionalmente aos problemas reais, decidiu-se gerar alguns problemas que foram denominados sintéticos. O objetivo de avaliar a solução fornecida pelo OPTIMA para tais problemas foi o de testar a robustez da abordagem, dadas algumas configurações onde a solução ótima era conhecida a priori (por exemplo, quando a solução ótima requer apenas uma troca de escaninho por ciclo), e possibilitar que o resultado pudesse ser visualizado por mínima inspeção, ou seja, pela simples verificação dos resultados no visualizador. No problema PB000001, estabeleceu-se que cada tipo de componente (escaninho) seria inserido uma única vez, enquanto no PB000003, os 30 componentes se repetem, mas se arranjados numa ordem adequada permitiriam a inserção com não mais que uma troca de escaninho por inserção.

Tabela 5 – Casos de Teste Sintéticos

Problema	Componentes	Escaninhos
PB000001	120	120
PB000003	120	30

A configuração destes problemas está descrita na Tabela 5. Na Tabela 6, são exibidos os resultados para estes casos de teste sintéticos. Nesta tabela, indicam-se o tempo inicial obtido pelo embaralhamento aleatório de uma solução predefinida, o tempo de inserção obtido com o OPTIMA, o tempo mínimo teórico, que neste caso é a configuração mínima definida pelo traçado sintético, e o percentual de melhoria obtido, neste caso já considerando o uso do algoritmo CRAFT modificado.

Tabela 6 – Resultados Finais para os Casos de Teste Sintéticos

Placa	Inicial [s]	OPTIMA [s]	Lim. Inferior [s]	% Total
PB000001	80.15	47.4	42.7	40.8
PB000003	79.86	47.8	42.5	40.1

Observa-se também que o OPTIMA conseguiu reduzir relativamente bem os tempos de inserção em relação à média das instâncias aleatórias, ficando porém mais de 10% acima do limite inferior ou da possível solução ótima. Entretanto, como o objetivo principal do OPTIMA é o de obter boas soluções num tempo de processamento razoável, podemos considerar o resultado acima como sendo de boa qualidade.

7. Conclusão

Como resultado deste trabalho, pode-se afirmar que a técnica de A-Teams mostra-se apropriada para abordar problemas de otimização similares aos da otimização da máquina insersora AVK.

Os resultados mostram que a abordagem de combinar o QAP com o TSP consegue obter uma melhora adicional nos tempos de inserção de 4,7% em média em relação a uma abordagem que utiliza exclusivamente o TSP.

Como uma vantagem no uso desta técnica, podemos ressaltar sua inerente modularidade. O fato dos agentes nos A-Teams serem autônomos facilita a modularização desde a concepção até a implementação, sendo adaptada ao desenvolvimento incremental. Uma vez determinado o problema e escolhidos os algoritmos, já se podem iniciar os experimentos, através da construção de um sistema mínimo, e ir incrementalmente ampliando-o conforme as necessidades. Em particular, no caso do OPTIMA, pode-se experimentar a modificação de um algoritmo para resolução do QAP, para especializá-lo mais e melhorar o desempenho do sistema.

O uso de A-Teams para solução de QAPs, mesmo puros, é uma segunda contribuição deste trabalho, já que não há na literatura consultada e na página WWW da QAPBIB (Burkard, Karih & Rendl, 1991) nenhuma menção ao uso dessa técnica. Há, portanto, uma nova e interessante linha de pesquisas nessa área, mormente por ser um problema que tem se

mostrado bem difícil de resolver. Uma possibilidade viável é a construção de A-Teams específicos para a resolução de instâncias do QAP, visando realizar uma avaliação da técnica. Poderiam-se reduzir os limites inferiores conhecidos ao tratar instâncias dos problemas para os quais não se conheçam ainda os valores ótimos, de forma análoga ao trabalho realizado por Souza (1993) para o TSP.

Agradecimentos

O segundo autor é financiado parcialmente pelo CNPq, processo 301041/95-4. Os autores também gostariam de agradecer os valiosos comentários do Prof. Carlos Eduardo Ferreira (IME/USP).

Referências Bibliográficas

- (1) Armour, G.C. & Buffa, E.S. (1963). A Heuristic Algorithm and Simulation Approach to the Relative Location of Facilities. *Management Science*, **9**, 294-309.
- (2) Broad, K.; Mason, A.; Rönnqvist, M. & Frater, M. (1996). Optimal Robotic Component Placement. *Journal of the Operational Research Society*, **47**, 1343-1354.
- (3) Burkard, R.E.; Karish, S.R. & Rendl, F. (1991). QAPLIB-A Quadratic Assignment Problem Library. *European Journal of Operational Research*, **55**, 115-119.
- (4) Garey, M.R. & Johnson, D.S. (1979), *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco, Freeman.
- (5) Held, M. & Karp, R.M. (1970). The Traveling-Salesman Problem and Minimum Spanning Trees. *Operations Research*, **18**, 1138-1162.
- (6) Kang, T.C. & Talukdar, S. (1997). Overview on Designing Autonomous Agents Teams (A-Teams). Apresentado na Conference on Management and Control of Production and Logistics MCPL'97, Campinas.
- (7) Lawler, E.L. (1963). The Quadratic Assignment Problem. *Management Science*, **9**, 586-599.
- (8) Lawler, E.L. *et al.* (1985). *The Traveling Salesman Problem*. John Wiley & Sons, Chichester.
- (9) Leipälä, T. & Nevalainen, O. (1989). Optimization of the Movements of a Component Placement Machine. *European Journal of Operational Research*, **38**, 167-177.
- (10) Panasonic (1996). *Catálogo Panasert Electronic Component Placement/Insertion Machines*. Matsushita Electric Industrial Co., Ltd., **27**.
- (11) Pardalos, P.; Rendl, F. & Wolkowicz, H. (1994). The Quadratic Assignment Problem: A Survey and Recent Developments. **In: Proceedings of the DIMACS Workshop on Quadratic Assignment Problems** [edited by Pardalos, P.; Rendl, F.; Wolkowicz, H.], DIMACS Series in Discrete Mathematics, **16**, 1-42.
- (12) Peixoto, H. & Souza, P. (1994). Uma Metodologia de Especificação de Times Assíncronos. Technical Report DCC-94-12, Unicamp, Campinas.

-
- (13) Rabak, C.S. (1999). Otimização do Processo de Inserção Automática de Componentes Eletrônicos Empregando a Técnica de Times Assíncronos. Dissertação de Mestrado, Escola Politécnica da Universidade de São Paulo, Departamento de Engenharia Elétrica, São Paulo, SP.
 - (14) Rardin, R.L. (1998). *Optimization in Operations Research*. Prentice Hall, Inc., Upper Saddle River, NJ.
 - (15) Reinelt, G. (1994). *The Traveling Salesman Computational Solutions for TSP Applications*. Springer-Verlag, Berlin.
 - (16) Souza, P. & Talukdar, S. (1993). Asynchronous Organizations For Multi-Algorithm Problems. ACM Symposium on Applied Computing, Indianapolis.
 - (17) Souza, P. (1993). Asynchronous Organizations for Multi-Algorithm Problems. Ph. D. Dissertation, Carnegie Mellon University, Dept. of Electrical and Computer Engineering, Pittsburgh, PA.
 - (18) Talukdar, S.; Baerentzen, L.; Gove, A. & Souza, P. (1996). Asynchronous Teams: Cooperation Schemes for Autonomous Agents. Engineering Design Center, Carnegie Mellon University.
 - (19) Walas, R.A. & Askin, R.G. (1984). An algorithm for NC Turret Punch Press Tool Location and Hit Sequencing. *IEE Transactions*, **16**(3), 280-287.
 - (20) West, D.H. (1983). Algorithm 608 Approximate Solution of the Quadratic Assignment Problem. *ACM Transactions on Mathematical Software*, **9**(4), 461-466.