

UMA COMPARAÇÃO DO PROCESSO DE LOW ANNEALING COM HIGH ANNEALING APLICADO AO POSICIONAMENTO DE CÉLULAS

Renato Hentschke¹, Diogo Fiorentin¹, Ricardo Reis¹

<renato,dfiorent,reis>@inf.ufrgs.br

¹ – Instituto de Informática - UFRGS - Universidade Federal do Rio Grande do Sul
CP15064 - Campus do Vale
CEP 915011-970 - Porto Alegre – Brasil

ABSTRACT

This article compares the *low-annealing* with *high-annealing* process applied to the physical synthesis cell placement. We observe, through cost variation graphics that the Simulated Annealing algorithm, with high initial temperatures spends unnecessary iterations and CPU time in the destruction of the initial solution. The execution data shows that both *low* and *high* annealing reach equivalent solutions, but the *high-annealing* process uses 100 more iterations for that. By the graphic and results analysis, we conclude that the best approach is to use the initial solution and improve it slowly.

RESUMO

Este artigo faz uma comparação do processo de *low-annealing* e *high-annealing* aplicado para o posicionamento de células em síntese física. Observa-se, através de gráficos de variação do custo, que o algoritmo de *Simulated Annealing* com altas temperaturas iniciais mostra variar o custo negativamente de forma desnecessária, desperdiçando iterações e consequentemente tempo de CPU. Os dados de execução mostram que os dois processos atingem soluções equivalentes, sendo que usa-se mais iterações no processo de *high-annealing*. Observa-se nos gráficos, que o aproveitamento da solução inicial combinado com uma convergência sempre positiva, porém lenta, pode levar aos melhores resultados.

UMA COMPARAÇÃO DO PROCESSO DE LOW ANNEALING COM HIGH ANNEALING APLICADO AO POSICIONAMENTO DE CÉLULAS

Renato Hentschke¹, Diogo Fiorentin¹, Ricardo Reis¹

<renato,dfiorent,reis>@inf.ufrgs.br

¹ – Instituto de Informática - UFRGS - Universidade Federal do Rio Grande do Sul
CP15064 - Campus do Vale
CEP 915011-970 - Porto Alegre – Brasil

RESUMO

Este artigo faz uma comparação do processo de *low-annealing* e *high-annealing* aplicado para o posicionamento de células em síntese física. Observa-se, através de gráficos de variação do custo, que o algoritmo de *Simulated Annealing* com altas temperaturas iniciais mostra variar o custo negativamente de forma desnecessária, desperdiçando iterações e consequentemente tempo de CPU. Os dados de execução das mostram que os dois processos atingem soluções equivalentes, sendo que usa-se mais iterações no processo de *high-annealing*. Observa-se que o aproveitamento da solução inicial combinado com uma convergência sempre positiva, porém lenta, pode levar aos melhores resultados.

1. INTRODUÇÃO

A etapa de posicionamento é de significativa importância para a síntese física do circuito. Seu papel é de definir a posição das células ao longo do espaço no circuito integrado. Desta forma, o posicionamento influi diretamente no roteamento, definindo o gerenciamento e o tamanho mínimo das conexões.

Existem vários algoritmos para posicionamento na literatura, porém há pouco consenso de qual o melhor. As publicações de algoritmos para posicionamento costumam ter suas próprias métricas, dificultando que haja dados comprovando qual a melhor estratégia de posicionamento.

Os algoritmos de posicionamento existentes na literatura se dividem em dois grupos: construtivos e iterativos. Os algoritmos construtivos devem gerar um posicionamento a partir do *netlist* de entrada somente. Os iterativos, por outro lado, partem de um posicionamento inicial, gerado por algoritmos construtivos.

Entre os algoritmos iterativos, se destacam as meta-heurísticas *Simulated Annealing*, Algoritmos Genéticos e Busca Tabu. Todas elas se caracterizam por necessitar de muito tempo de processamento (em geral, ordens de grandeza maiores que algoritmos construtivos). Porém,

observa-se em [1] que as meta-heurísticas, dado o tempo de processamento necessário, podem atingir soluções melhores.

Especificamente, existem diversos trabalhos dedicados para otimizar o tempo de CPU requerido pelo *Simulated Annealing* [2][3]. Entre as técnicas mais importantes, destacam-se as *schedules* de temperatura adaptativas. A função da temperatura é fornecer ao algoritmo um certo grau de liberdade para aceitar transformações que piorem o estado atual. Quanto mais alta a temperatura, maior é a liberdade dada.

Em um processo de *Simulated Annealing*, as modificações aleatórias (perturbações) impostas ao estado atual do algoritmo são aceitas de acordo com uma probabilidade de aceitação. Esta probabilidade é variável de acordo com a temperatura.

Baseado na temperatura inicial dos algoritmos, existem duas técnicas possíveis de *Simulated Annealing*: *high-annealing* (altas temperaturas iniciais) e *low-annealing* (baixas temperaturas iniciais). O uso de altas temperaturas iniciais faz com que haja uma destruição da solução inicial, tornando o processo independente do algoritmo de posicionamento inicial empregado. Iniciar por baixas temperaturas pode ser uma boa maneira, porém, de economizar iterações.

O trabalho de Aart [3] mostra uma técnica de encontrar a temperatura inicial adaptada para cada circuito. A temperatura é encontrada com base em uma probabilidade inicial alvo. Em [3], esta técnica é usada para *high-annealing* (probabilidade inicial maior que 0,4).

Além da determinação da temperatura inicial, deve haver uma preocupação com a variação da temperatura ao longo do processo. Neste artigo, considera-se como ponto final do processo a temperatura zero, sendo que ela é atingida depois de um número pré-determinado de iterações.

Este artigo inicia um trabalho de comparação entre os dois processos. O objetivo é avaliar as possibilidades e requisitos necessários ao processo de *low-annealing* combinado com diversos algoritmos construtivos. Ao

mesmo tempo, deseja-se avaliar o processo de *high-annealing* em função da variação de temperatura ao longo do processo.

Este artigo está organizado da seguinte maneira. A sessão 2 explica a metodologia de comparação empregada. A sessão 3 apresenta o fluxo de posicionamento empregado. A sessão 4 discute posicionamento com *low-annealing*, enquanto que a sessão 5 discute posicionamento com *high-annealing*. Finalmente, a sessão 6 apresenta as conclusões e trabalhos futuros.

2. METODOLOGIA DE COMPARAÇÃO

Os algoritmos são comparados segundo as seguintes variáveis:

- Estimativa de tamanho dos fios.
- Estimativas de Congestionamento
- Tempo de CPU
- Roteabilidade no Tropic [4]

Para estimar o tamanho dos fios, usa-se o método do semiperímetro. Para o congestionamento, usa-se o método da Densidade Máxima por Bounding-Boxes [7].

No gerador de *layout* do Tropic3, o espaço dedicado ao roteamento varia conforme a complexidade do circuito, havendo canal de roteamento de altura variável entre as bandas. Assim, quanto melhor for o posicionamento, menor será a área de roteamento, aumentando, portanto, a densidade (transistores por milímetro quadrado). Assim, a comparação de roteabilidade no Tropic é feita pela densidade do *layout*. Os dados são obtidos a partir da execução do Tropic com um parâmetro especial para considerar o posicionamento de uma ferramenta externa.

3. FLUXO DE POSICIONAMENTO

O processo de posicionamento estudado neste trabalho é feito em três etapas: posicionamento inicial, pós-processamento do posicionamento inicial e posicionamento iterativo. O posicionamento inicial é realizado pelos seguintes algoritmos:

Algoritmo A	Aleatório
Algoritmo B	Plic-Plac[5]
Algoritmo C	Cluster Growth C com 20 níveis de profundidade.
Algoritmo D	Particionamento em Quadratura, usando o particionamento de Fiducia-Mateyses [6]
Algoritmo E	Particionamento em Bissecção de Bandas, com particionamento de Fiducia-Mateyses.

Resultados experimentais mostram que o algoritmo aleatório e o Plic-Plac são os mais rápidos (tempos

menores que 1 segundo para circuitos com até 5500 células). A qualidade de posicionamento gerado mostra que os algoritmos baseados em particionamento são os conduzem a melhores resultados. Há, portanto, um compromisso de tempo de execução com qualidade. O algoritmo Cluster Growth possui um tempo de execução e qualidade moderados.

A próxima etapa é o pós-processamento do posicionamento inicial. É utilizado um algoritmo guloso, com função de custo especial para equalização do tamanho das bandas. O objetivo é que haja um equilíbrio do tamanho das bandas do posicionamento inicial, evitando áreas vazias.

Por fim, utiliza-se o *Simulated Annealing* para iterar sobre o posicionamento inicial. Para encontrar a temperatura inicial, usa-se o mecanismo adaptativo de Aart [3], que encontra a temperatura inicial apropriada para o *benchmark* sendo posicionado. Este mecanismo exige que haja uma probabilidade inicial desejada. Em seu artigo, Aart sugere uma probabilidade inicial de 0,95.

São consideradas neste artigo duas metodologias para encontrar a probabilidade inicial: *low-annealing* e *high-annealing*. A metodologia de *low-annealing* trabalha com baixas probabilidades iniciais (0,02, por exemplo). A metodologia de *high-annealing* usa probabilidades maiores que 0,4. Neste artigo, trabalha-se com 0,4, que é suficiente para quebrar a estrutura inicial do posicionamento.

A variação da temperatura é feita através de uma curva de terceiro grau com ponto de inflexão variável. No processo de *high-annealing*, é desejado que a curva tenha rápida convergência, para diminuir a temperatura inicial rapidamente. A figura 2.a mostra dois exemplos de curvas adequadas para *high-annealing*. No processo de *low-annealing*, esta curva deve ser mais próxima de uma reta conectando a temperatura inicial (na figura 2: t_i) com a temperatura final (que vale zero), como mostra a figura 2.b.

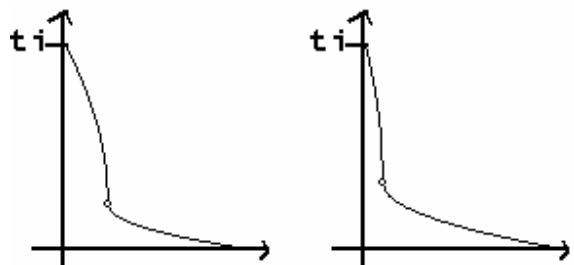


Figura 2.a – Variações de temperatura adequadas para *high-annealing*.

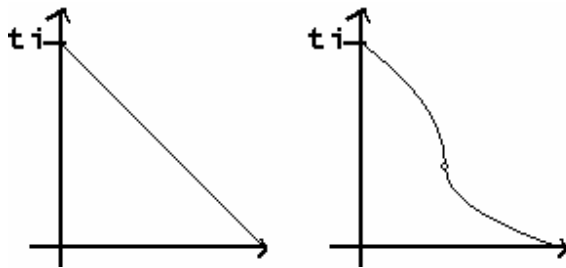


Figura 2.b – Variações de temperatura adequadas para *low-annealing*

4. POSICIONAMENTO COM LOW-ANNEALING

O objetivo de *low-annealing* é manter o resultado obtido como solução inicial e trabalhar sobre ele sem que haja perda da estrutura inicial. Assim, iterações usadas para a destruição da solução inicial são economizadas.

Estudam-se, nesta sessão, alguns fluxos de execução. São consideradas combinações de algoritmos construtivos com *low-annealing*. É feita uma comparação, para o mesmo circuito, de possíveis probabilidades iniciais. A probabilidade inicial tem um efeito diferente para cada algoritmo. No caso de algoritmos como a quadratura, onde o *wirelength* (comprimento total dos fios) atingido é melhor, a probabilidade inicial deve ser menor. Nos casos onde o algoritmo é semelhante ao aleatório, a temperatura inicial deve ser mais alta.

Por fim, serão feitos alguns testes para diferentes circuitos, mantendo a mesma temperatura inicial e algoritmo para posicionamento inicial.

4.1. VARIANDO A TEMPERATURA INICIAL

Quatro experimentos são realizados nesta sessão, sobre o benchmark ISCAS'89 C1908_3x3 (783 células):

- Low-Annealing* com probabilidade inicial de 0,2 (que é alta)
- Low-Annealing* com probabilidade inicial de 0,02
- Low-Annealing* com probabilidade inicial de 0,002
- Low-Annealing* com probabilidade inicial de 0,0002

Observa-se no gráfico da figura 1, porém, que o efeito final no WL não é determinado somente pela temperatura no circuito C1908. Há uma forte dependência do algoritmo de posicionamento inicial. A partir dos dados da figura 1, observa-se que a probabilidade de 0,2 não é adequada para *low-annealing*, por ser muito alta. A probabilidade de 0,02 ainda é alta para os circuitos com melhor posicionamento inicial. Porém, os circuitos com pior posicionamento inicial, gerados pelos algoritmos A e B, tem a sua melhor otimização (em relação as outras probabilidades). Isto

significa que a probabilidade de 0,02 ainda não foi suficiente para quebrar a estrutura inicial do posicionamento. A probabilidade de 0,002 apresenta resultados ruins para os circuitos menos otimizados e resultados ótimos para os circuitos mais otimizados (com exceção da quadratura). Em outras palavras, uma probabilidade de 0,002 é muito baixa para recuperar um posicionamento inicial ruim, mas é suficientemente baixa para melhorar um bom posicionamento inicial. No caso da probabilidade de 0,0002, há uma vantagem razoável dos circuitos otimizados também, porém percebe-se que a temperatura é baixa demais, prendendo-se com mais facilidade em máximos locais.

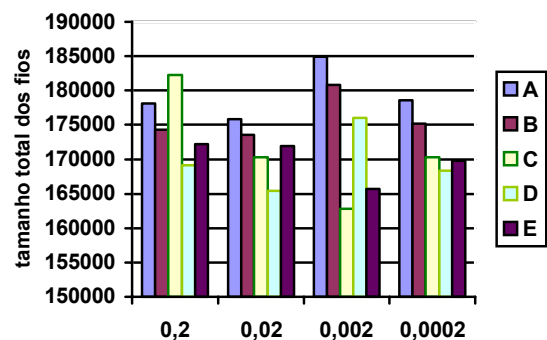


Figura 1 – Variação do tamanho dos fios de acordo com a temperatura inicial aplicada

Ao analisar a curva de variação do custo (figura 2) de cada um dos métodos descritos, observa-se uma convergência gulosa dos algoritmos com baixa temperatura, enquanto que a probabilidade inicial de 0,02 mostra uma variação de custo sempre descendente.

A figura 2 mostra graficamente a evolução do custo de cada uma das execuções usando o algoritmo aleatório como inicial. Observe como a execução com probabilidade inicial de 0,2 começa aumentando o custo inicial, mas consegue convergir adequadamente no final. As demais temperaturas iniciais, porém, convergem imediatamente para soluções melhores. Quanto menor a temperatura inicial, mais brusca é a convergência, aproximando-se de um algoritmo guloso. A variação ideal ocorre quando a curva aproxima-se de uma reta, havendo uma variação uniforme do custo. Desta forma, o algoritmo está mais habilitado a evitar de ficar preso em máximos locais (como os algoritmos gulosos).

4.2. RESULTADOS GERAIS PARA LOW ANNEALING

A tabela 1 refere-se à execução de *Low Annealing* sobre diversos circuitos benchmarks ISCAS'89 com a quadratura de algoritmo construtivo, seguida de pós-processamento. Por fim, usa-se Simulated Annealing,

com probabilidade inicial de 0,02 e 300 iterações. O software Tropic [4] é utilizado para geração do *layout* e roteamento.

Tabela 1 – Resultados de Low Annealing

Benchmark	Num Cells	WL	CPU	Cong.	Densidade Tropic
C499	364	89344	161	31	68K
Bw	468	94829	219	27	63K
C1908	783	207652	588	39	58K
Mult	1408	577351	1625	49	63K
Alu4_4x4	2523	3811192	6142	265	-
C53	3249	986913	7318	154	43K
Alu4_2x2	4844	4837485	18056	183	-

Observe, na tabela 1, que a coluna de densidade reflete a roteabilidade do circuito. A tendência natural é que circuitos mais complexos tenham menor densidade devido ao aumento da necessidade de conexões. A exceção é o circuito Mult que trata-se de um multiplicador e, por isto, é mais regular que o bloco de lógica aleatória C1908. Por ser mais regular, é mais simples de posicionar. Outra exceção é o bloco C53, que é maior que o Alu4_4x4, porém menos complexo. Na coluna de densidade, onde aparece um '-' significa que o software Tropic não foi capaz completar o roteamento. Isto ocorre devido a demasiada demanda por conexões.

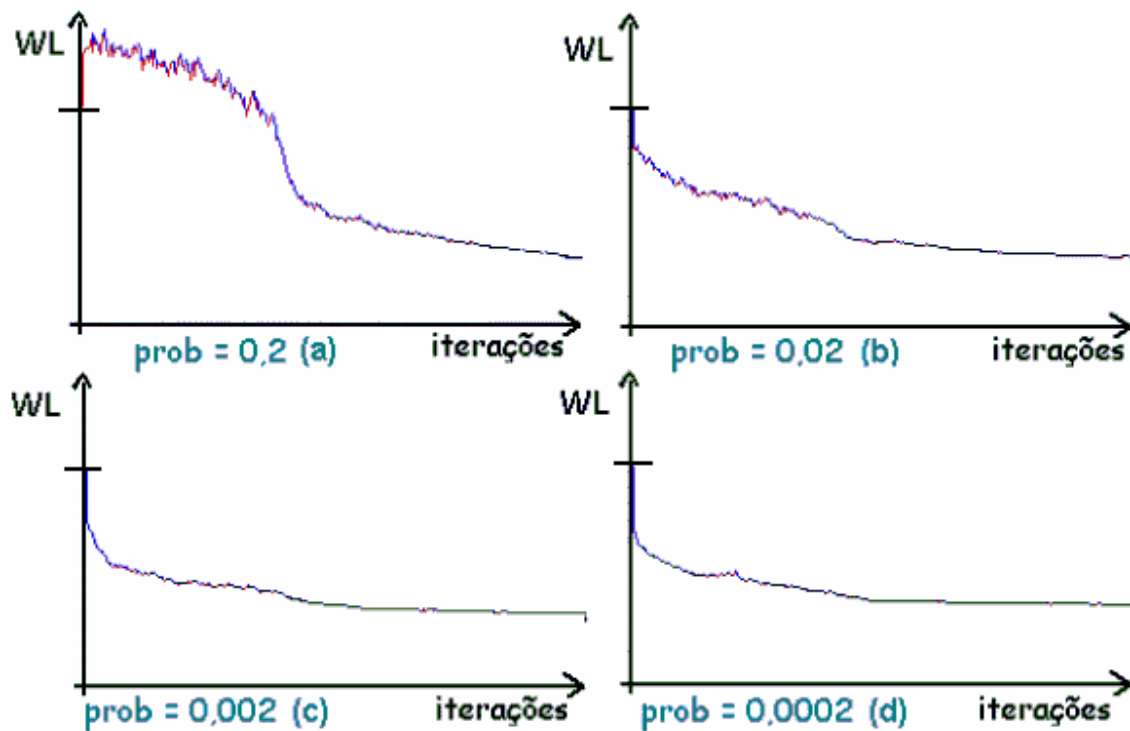


Figura 2 – Variação do custo em processos de *low-annealing* com diferentes probabilidades iniciais

5. POSICIONAMENTO COM HIGH-ANNEALING

O processo de *high-annealing* é diferente do *low-annealing* por ser totalmente independente da solução inicial. O processo inicia em altas temperaturas, fazendo com que a solução inicial seja totalmente destruída. A figura 3 mostra a variação do custo de todas as soluções. Repare como o custo inicial cresce demasiadamente, mostrando que a solução inicial é totalmente descartada.

Note que a curva de variação do custo é diferente para cada posicionamento inicial. Quanto mais otimizado é o algoritmo, maior é a destruição realizada nas altas temperaturas. O *Cluster Growth*, para este *benchmark*,

apresentou a melhor solução e, por isto, teve o seu posicionamento mais afetado pelo *Simulated Annealing*. A conclusão que pode ser tirada aqui é que há uma forte correlação entre a otimização da solução inicial com o tamanho da variação negativa no custo, para uma probabilidade inicial fixa (no caso: 0,4). Comparando com o gráfico da figura 2, conclui-se que o tamanho da variação negativa pode ser controlado com mudança na temperatura inicial. Esta observação é a primeira idéia para o desenvolvimento de uma *schedule* de temperatura totalmente adaptativa, que possa manter a estrutura inicial e convergir equilibradamente, o que será realizado como trabalho futuro.

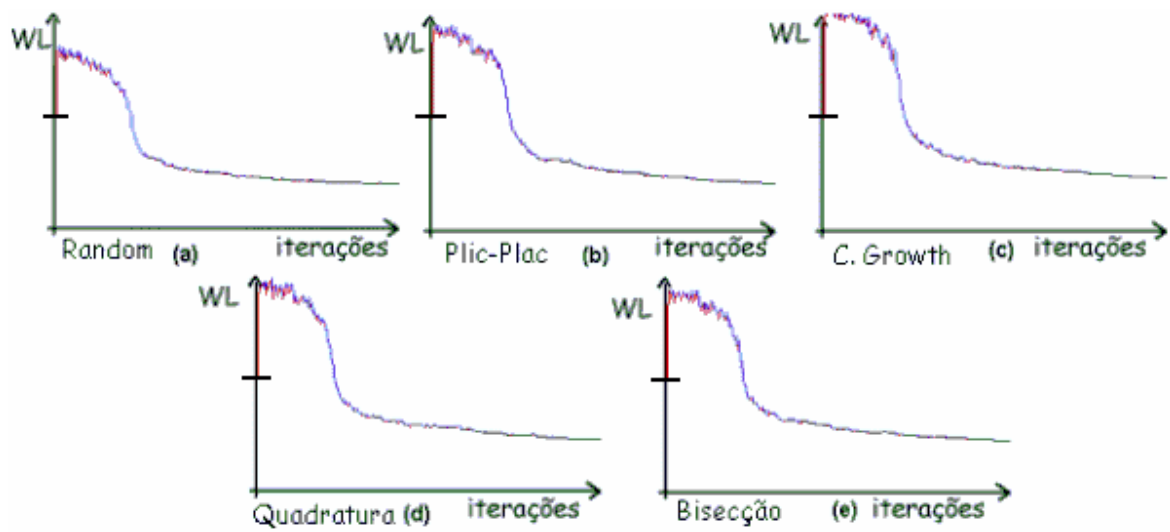


Figura 3 – Variação do custo para *high-annealing*

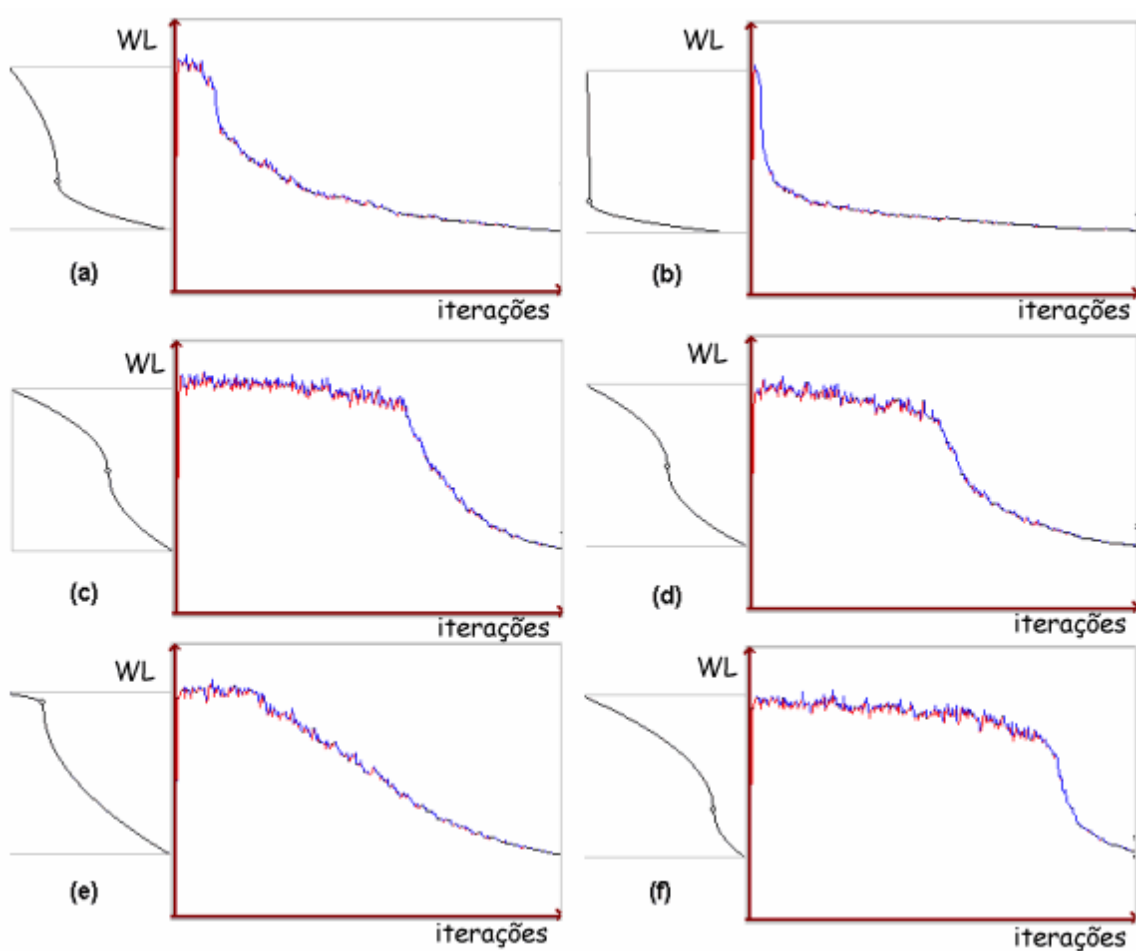


Figura 4 – Curva de variação do custo acompanha curva de variação da temperatura, em *high-annealing*

Uma característica muito interessante do *high-annealing* é como a variação do custo acompanha a variação da temperatura. Observe a figura 4, que mostra alguns exemplos para o circuito C1908, posicionado inicialmente com o Plic-Plac. A figura 4 mostra claramente, nos exemplos *c,d e f*, que o processo de *high-annealing* pode desperdiçar muitas iterações no começo do processo, sendo que *schedules* como nos exemplos *a e b* podem apresentar resultados melhores em menos tempo.

As altas temperaturas têm a função de tornar o algoritmo independente da solução inicial, fugindo de qualquer eventual máximo local. Observando as figuras, nota-se que ficar muito tempo em altas temperaturas, porém, é inútil.

5.1. RESULTADOS GERAIS PARA HIGH ANNEALING

A fim de comparar o processo de *high-annealing* com *low-annealing*, foi feito um experimento com 400 iterações (em *low-annealing* – tabela 1 – usou-se 300 iterações) e 3 repetições por célula (em *low-annealing* usou-se 3 repetições também). Os resultados estão na tabela 2.

Tabela 2 – Resultados para high-annealing

Benchmark.	Num Cells	Wirelength	Cong.	Densidade Tropic
C499	364	87590	32	67K
Bw	468	95376	32	63K
C1908	783	198976	35	57K
Mult	1408	529440	46	62K
Alu4 4x4	2523	3637810	262	-
C53	3249	986373	133	49K
Alu4 2x2	4844	4659951	187	-

Os resultados mostram que a complexidade e o tamanho do circuito prejudicam o desempenho dos algoritmos de posicionamento. Quanto maior o circuito, menor a densidade atingida, pois há um aumento na demanda por conexões.

Comparando-se os dados com a tabela 1, nota-se que os resultados são praticamente equivalentes. O *low-annealing* ganha em alguns casos, como o circuito C499 e C1908, enquanto que perde em outros casos, sempre com uma pequena diferença. Esta diferença é fruto do caráter não-determinístico dos algoritmos. Com isto, observa-se que o processo de *high-annealing* levou mais tempo (400 iterações, contra 300) para atingir os mesmos resultados.

6. CONCLUSÕES E TRABALHOS FUTUROS

Resumindo as conclusões obtidas ao longo do artigo, o processo de *low-annealing* é mais guloso ao baixar a

temperatura. Apesar de mais guloso, o processo está apto a escapar de máximos locais através de uma lenta convergência. Esta convergência pode ser obtida através da escolha correta da temperatura inicial. É proposto como trabalho futuro que se encontre esta temperatura com um processo automático.

Os gráficos da figura 3 mostraram que *high-annealing* perde tempo destruindo a solução inicial. Os mesmos gráficos mostram também que, quanto mais otimizado o circuito inicial, maior é a destruição realizada (fixando-se a temperatura inicial). Assim, é possível controlar a variação do custo pela temperatura inicial.

Por fim, os resultados das tabelas 1 e 2 mostram que o uso de *low-annealing* pode, de fato, economizar iterações por aproveitar a solução inicial.

6. REFERÊNCIAS

- [1] HENTSCHE R., FIORENTIN D, MORAES F., REIS R. **Comparação de Posicionamento com Simulated Annealing e Quadratura no Gerador Automático de Macro-células Tropic**. Aceito para publicação no Iberchip 2003.
- [2] SU, Lixin; BUNTINE, Wray; NEWTON, Richard. **Learning as Applied To Stochastic Optimization for Standard-Cell Placement**. In IEEE Transactions on Computer-Aided Design of Integrated Circuits And Systems, Vol. 20, nº 4, April 2001.
- [3] AARTS E. H. L. and LAARHOVEN P.J.M.. **A New Polynomial-Time Cooling Schedule**. In Proceedings of ICCAD 1985.
- [4] MORAES, Fernando Gehm; ROBERT, Michel; AUVERGNE, Daniel. **A Virtual CMOS Library Approach for Fast Layout Synthesis**. In: VLSI: Systems on a Chip. Lisbon, 1999, p. 415-426
- [5] HENTSCHE R., REIS R. **Plic-Plac: a Novel Constructive Algorithm for Placement**. Aceito para publicação no ISCAS 2003, Bangkok, Tailândia
- [6] SHERWANI, Naveed A. **Algorithms for VLSI Physical Design Automation**. Third Edition, Kluwer Academic Publishers, 1998.
- [7] YANG X, KASTNER R, SERRAFZADEH M. **Congestion Estimation During Top-Down Placement**. IEEE Transactions on CAD and Systems, Jan 2002.