

Implementações do Simulated Annealing para o problema do Caixeiro Viajante Simétrico

Universidade Federal do Paraná
Tópicos em Inteligência Artificial
Dr^a Aurora Pozo

Elaboração do Relatório: Ademir Roberto Freddo, Robison Cris Brito

Implementações:

Trabalho 1: Ademir Roberto Freddo, Robison Cris Brito

Trabalho 2: André Leonardo Richardz, Eros Moreira de Carvalho,
Gabriel Silva Ramos.

Trabalho 3: Ana Maria Pereira, Bruno Silva de Oliveira

Resumo

Este documento apresenta três implementações de *Simulated Annealing* para o problema do Caixeiro Viajante. Os trabalhos são denominados respectivamente Trabalho1, Trabalho2, Trabalho3 e possuem diferentes características na implementação do algoritmo. Também são apresentadas outras características como geração de estados iniciais e escolha/calibragem dos parâmetros. Além disso, são relatados testes de cada algoritmo, em duas situações: mesmos estados e parâmetros (utilizando baixas temperaturas), mesmos estados e diferentes parâmetros (utilizando altas temperaturas). Os resultados são ilustrados em gráficos para a comparação dos trabalhos.

1. Introdução

O algoritmo *Simulated Annealing* (SA), atualmente, é utilizado como um componente da meta-heurística, principalmente quando aplicado em algoritmos isolados. O SA surgiu do algoritmo denominado Metrópolis [4] e Kirkpatrick [3] sugeriu a utilização deste algoritmo no ambiente computacional. O SA simula o resfriamento de um conjunto de átomos aquecidos (recozimento). Essa simulação corresponde a um processo térmico de liquidação de um cristal em alta temperatura. A temperatura sofre uma lenta e gradativa diminuição até um estado de energia ou ponto solidificação mínima. O SA é considerado o mais velho das meta-heurísticas, e certamente, um dos primeiros algoritmos a encontrar alternativas para escapar do mínimo local. A ideia básica é permitir que soluções piores que a atual sejam selecionadas para exploração da vizinhança. Escapando assim, do mínimo local. A probabilidade de ser selecionado um estado pior vai diminuindo ao longo da busca devido ao fato da redução de temperatura.

O algoritmo possui uma estrutura padrão que pode ser modificada, além dos parâmetros iniciais que devem ser calibrados antes da execução. Essas peculiaridades geram diversas maneiras de implementação e execução do algoritmo. Este documento apresenta 3 implementações de SA realizada por diferentes grupos. O objetivo deste trabalho é apresentar as características e resultados de cada implementação. As implementações foram testadas utilizando o problema do Caixeiro Viajante

Simétrico[8]. A Seção 2 deste artigo apresenta o problema do Caixeiro Viajante a ser utilizado nos testes das implementações. A fundamentação do algoritmo Simulated Annealing é apresentada na Seção 3. A Seção 4 apresenta as 3 implementações do SA. Na Seção 5 são apresentados os testes dos algoritmos utilizando os mesmos parâmetros e estados iniciais para todos os trabalhos. Na Seção 6 também são apresentados os resultados com mesmos estados iniciais porém com diferentes parâmetros. Por último, na Seção 7 são relatadas as conclusões.

2. Problema do Caixeiro Viajante

O problema do caixeiro viajante é um problema clássico de otimização combinatória. Suponha que um caixeiro tenha de visitar n cidades diferentes, iniciando e encerrando sua viagem na primeira cidade. A ordem de visita às cidades não importa e cada uma delas pode ir diretamente a qualquer outra. Resumindo, o problema do caixeiro viajante consiste em descobrir a rota que torna mínima a viagem total.

Para exemplificar, se tivermos quatro cidades A, B, C e D, uma rota que o caixeiro deve considerar poderia ser: saia de A e daí vá para B, dessa vá para C, e daí vá para D e então volte a A. Esta é uma possibilidade, mas existem outras seis rotas, entre elas: ABCDA, ABDCA, ACBDA, ACDBA, ADBCA, ADCBA.

Para os testes foram utilizadas as seguintes instâncias de dados [9]: Dantzig42 (42 cidades), Fri26 (26 cidades), gr48 (48 cidades) e hk48 (cidades). Todas as instâncias são simétricas.

3. Fundamentação Teórica do *Simulated Annealing*

Esse processo foi baseado na idéia moldagem de metais e vidros, que ficam menos flexíveis conforme a temperatura vai diminuindo. Quanto ao processo de busca, esse algoritmo é o resultado da combinação de duas estratégias: caminhos aleatórios e melhoramento iterativo. Na primeira fase da busca, a tendência de melhorar o resultado é baixa, entretanto, faz com que o espaço de busca seja explorado, esse comportamento é alterado, juntamente com a temperatura, fazendo o resultado convergir no final do processamento do algoritmo a um mínimo local. A escolha de um estado pior depende de duas variáveis: a diferença de custo entre o estado atual e o próximo estado e a Temperatura atual. Quanto maior a diferença entre o estado atual e o próximo estado, menor é a possibilidade dele ser selecionado, e quanto maior a temperatura, maior a possibilidade de um estado pior ser escolhido [1,2].

A estratégia do SA é permitir movimentos ruins em altas temperaturas. Os movimentos ruins são permitidos por uma distribuição de probabilidade [5]:

$$p(\Delta E) = e^{-\Delta E / T}$$

A medida que a temperatura diminui, os movimentos ruins são reduzidos.

3.1 Algoritmo

No Quadro 1 é apresentado o algoritmo do SA.

<p>Parâmetros:</p> <p style="padding-left: 40px;">T0 (temperatura inicial),</p> <p style="padding-left: 40px;">Tf (temperatura final),</p> <p style="padding-left: 40px;">L(numero de iteracoes),</p> <p style="padding-left: 40px;">α (redução de temperatura)</p> <p>T ← T0 ; (temperatura atual recebe temperatura inicial)</p> <p>S0 ← gera solução inicial; S ← S0; (solução atual)</p> <p>S* ← S0; (melhor solução obtida até o momento)</p>

```

enquanto  $T > T_f$  faça (temperatura alta)
  para  $cont \leftarrow 1$  até  $L(T)$  faça (iterações para equilíbrio)
     $S' \leftarrow$  seleciona uma solução vizinha de  $S$ 
     $\Delta custo \leftarrow custo(S') - custo(S)$ ;
    se  $\Delta custo < 0$ 
       $S \leftarrow S'$ ;
    senão
      Numero_Random  $\leftarrow$  Rand[0..1];
      se Numero_Random  $< \exp(-\Delta custo/T)$ 
         $S \leftarrow S'$ ;
      se ( $S < S^*$ )  $S^* \leftarrow S$  (guardar o melhor estado)
    fim para  $T \leftarrow \alpha T$  (reduz a temperatura)
fim-enquanto
Retorne  $S$  (melhor solução)

```

Quadro 1 – Algoritmo SA

A primeira parte do algoritmo apresenta os parâmetros necessários, entre eles:

- **Temperatura Inicial:** de preferência alta para a aceitação de valores ruins;
- **Temperatura Final:** relacionar com o tamanho do problema, iterações, tempo;
- **Número de iterações:** atingir o equilíbrio em uma temperatura;
- **Taxa de redução de temperatura:** quanto se quer explorar determinadas regiões.

A escolha dos valores iniciais dos parâmetros são cruciais para o sucesso desse algoritmo. Os parâmetros são formados pela Temperatura Inicial do processo, pela Temperatura Final ou ponto de parada, Número de Iteração para selecionar o novo estado a cada valor de temperatura, esse número de iteração também é chamado de Ponto de Equilíbrio, e ainda o índice de resfriamento de temperatura. Esses valores bem calibrados aumentam a chance de localizar um estado ótimo global.

O algoritmo inicia gerando um estado inicial (randomicamente ou heurísticamente construído) e inicia um parâmetro chamado Temperatura, explicado posteriormente. Então, a cada iteração do processo, é selecionado randomicamente uma amostra dos estados vizinhos da solução atual, os quais podem se tornar estados atuais de busca, isso se o custo do estado vizinho for inferior ao custo do estado atual. Se o estado vizinho for maior que o estado atual, então será aplicado a fórmula da probabilidade desenvolvida por Boltzmann: $\exp(-(\text{custoDoSucessor} - \text{CustoEstadoAtual}) / \text{Temperatura})$. O valor retornado por essa fórmula será comparado com um número gerado aleatoriamente entre 0 e 1, obtido na iteração. Se o valor retornado pelo algoritmo de Boltzmann for maior que o número aleatório, o estado vizinho mesmo possuindo custo superior ao estado atual, será selecionado como estado atual, para ter exploração de seus vizinhos.

A temperatura T é decrementado a cada iteração do processo de busca, por isso, no início do processamento a possibilidade de ser escolhido estados posteriores piores que o atual é grande, no andamento da busca, as possibilidades vão diminuindo, com a tendência de terminar com buscas locais [6,7].

Para o índice de resfriamento, são aconselhados valores entre 0 e 1, entretanto, os valores ideais para o índice de resfriamento podem modificar durante o processo de busca, com o objetivo de balancear entre a diversificação e a intensificação. Por exemplo, no início da busca, a Temperatura pode ser linearmente reduzido, convergindo para um mínimo local até o final da busca. Porém, existem variantes desse algoritmo que utilizam processos distintos de resfriamento e reaquecimento, possibilitando constantes alternâncias entre diversificação e intensificação.

Valores como Fator de Resfriamento, número de iteração e Temperatura inicial são características específicas de cada problema. No SA, o estado a ser escolhido depende das características atuais dos

parâmetros, já que o processo é classificado como sem memória, entretanto, o uso de memória em SA pode beneficiar o processo de busca.

4. Trabalhos de SA desenvolvidos

Nesta seção serão apresentados os trabalhos desenvolvidos de cada grupo.

4.1 Trabalho 1

O primeiro trabalho foi desenvolvido utilizando a linguagem de programação Java para Desktop. O algoritmo utilizado no trabalho foi o mesmo apresentado na Seção 3.1 (Quadro 1). A Figura 1 mostra o fluxograma. Percebe-se que o algoritmo possui dois laços de iteração. O primeiro possui como ponto de parada o valor da temperatura final, ou seja quando a temperatura inicial atinge a temperatura final. O segundo laço executa o número de iterações para a redução da temperatura. Após um determinado número de teste, a temperatura sofre uma taxa de redução.

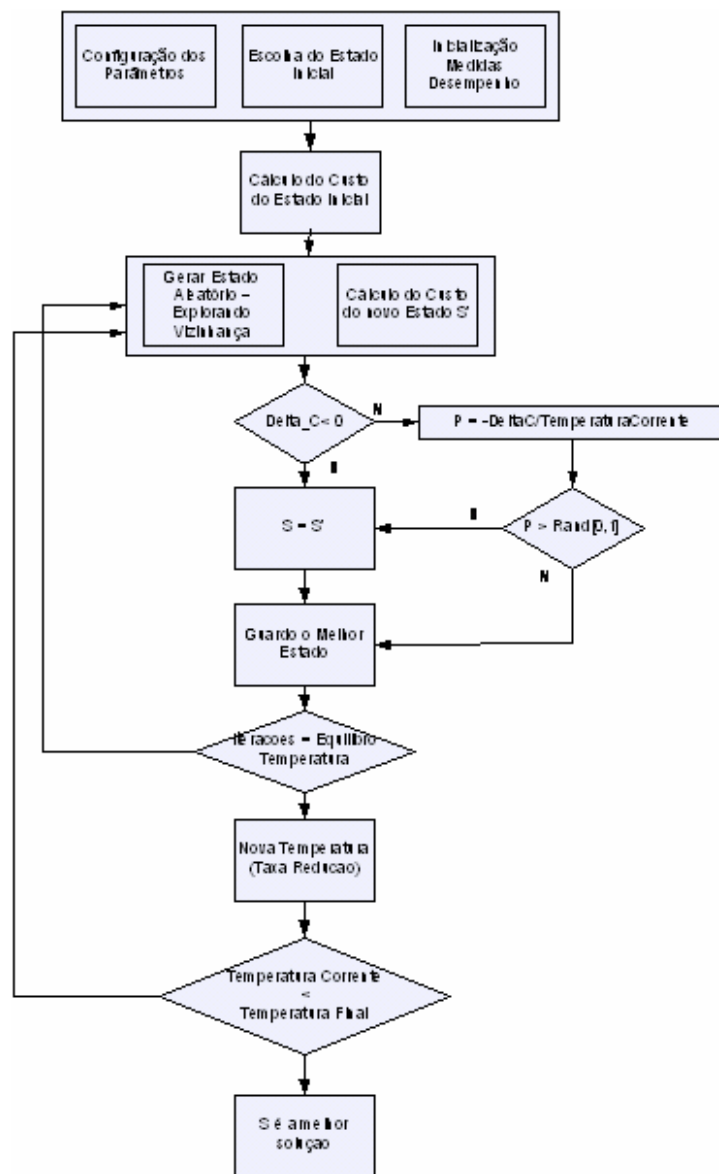


Figura 1 – Fluxograma Trabalho 1

Geração dos Parâmetros

A partir dos estados iniciais gerados, utilizou-se estes estados para a escolha dos melhores parâmetros. Para isso, foi criado um programa que testou os seguintes parâmetros: temperatura final, temperatura inicial, redução de temperatura e número de iterações para redução de temperatura. Este último somente foi utilizado pelo Trabalho1 devido às características do algoritmo. A seguir a definição dos parâmetros do algoritmo 1.

Temperatura

Todos os testes para geração dos parâmetros de temperatura iniciaram-se com temperatura inicial em 100. Para cada estado gerado na Seção 5.1 foram realizadas 3 execuções com uma taxa de redução de temperatura de 0.8%.

Temperatura Final

A Figura 2 ilustra o gráfico gerado para a Temperatura Final para uma instância de dados. O eixo X corresponde a Temperatura e no eixo Y estão os custos. O gráfico mostra que quando a temperatura final ultrapassa o valor 10 e a medida que diminui, o custo também diminui. Isto ocorre porque a busca realiza uma maior intensificação no espaço de busca. Porém, depois de uma certa melhora quando a temperatura final está baixa, o resultado se mantém fixo devido a probabilidade de aceitação de outros estados com custo superior.

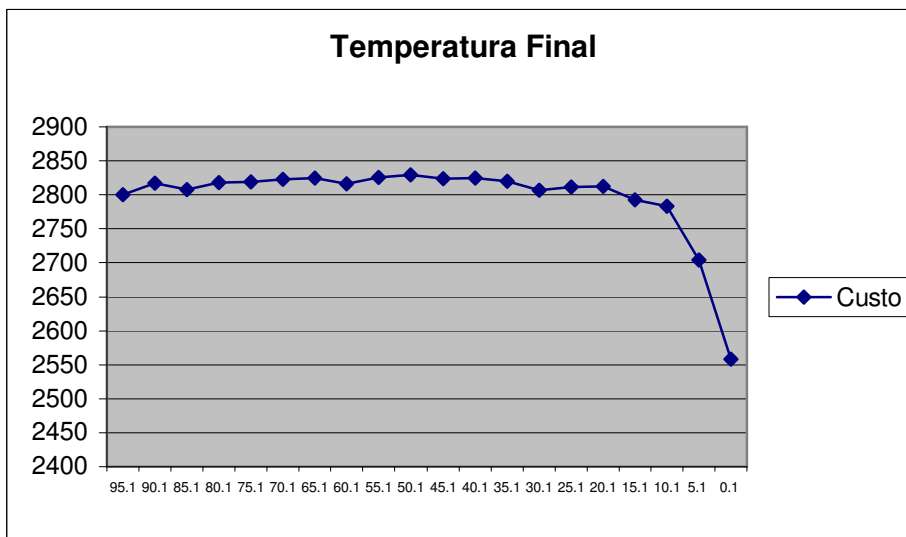


Figura 2 – Gráfico Temperatura Final

Temperatura Inicial

Para escolha do parâmetro de temperatura inicial foram calculadas as médias de 5 execuções em cada um dos estados gerados na seção 5.1 e utilizou-se como 0.8% como taxa de redução de temperatura e 0.1 como temperatura final. A Figura 3 apresenta o gráfico com o melhor valor encontrado para Temperatura Inicial. No gráfico é ilustrado que a melhor temperatura, ou seja a reduziu mais o custo foi de valor 30.

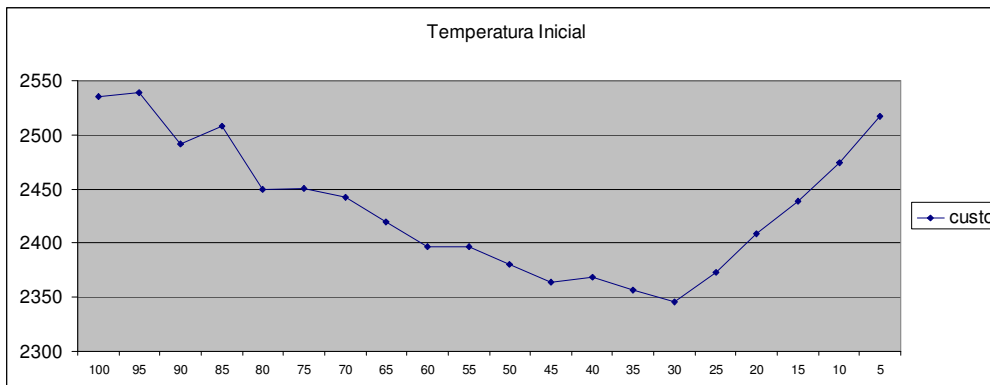


Figura 3 – Gráfico Temperatura Inicial

Número de Iterações

Outro parâmetro trabalhado foi o número de iterações. Este parâmetro indica qual é o número de iterações necessários para aplicar uma taxa de redução de temperatura. Em todas as instâncias testadas chegou-se a um valor de 60% da matriz, mais precisamente em uma instância com 40 cidades o valor adequado de iterações é de 24. Isso indica que após 24 iterações deve-se aplicar a taxa de redução de temperatura a temperatura inicial. A Figura 4 mostra o gráfico gerado.

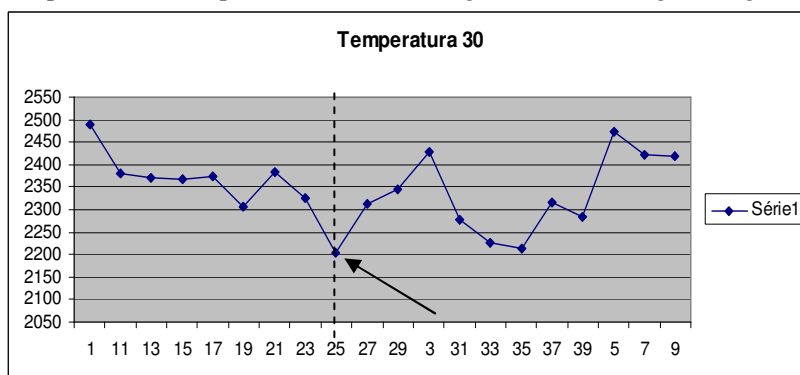


Figura 4 – Gráfico Número de Iterações

Taxa de Redução de Temperatura

Para a taxa de redução de temperatura o programa realizou cinco execuções por estados e taxa de redução foi variada de 3 até 0.1. A Figura 5 mostra que os melhores resultados foram alcançados com uma taxa de redução de 0,2.

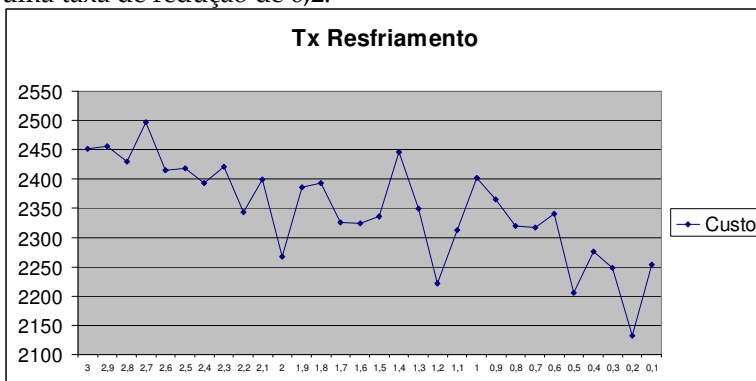


Figura 5 – Gráfico Redução de Temperatura

Após realizados estes testes, seguem os parâmetros utilizados pelo Trabalho1:

- Temperatura Final: 0.1
- Temperatura Inicial: 30
- Número de Iterações: 60% do tamanho da matriz
- Taxa de Redução de Temperatura: 0.2

Estes mesmos parâmetros também serão utilizados pelos Trabalho2 e Trabalho3 para testes comparativos na Seção 5.

Geração dos Estados Iniciais

Para a geração desses 10 estados iniciais foi implementado um programa que importa as bases de dados e gera 50.000 estados iniciais. Destes 50.000 estados foram escolhidos os cinco melhores de cada instância. O objetivo do programa foi encontrar estados iniciais de boa qualidade. O sexto estado é um estado seqüencial. Os últimos quatro estados foram gerados aleatoriamente. Esses estados são considerados de qualidade razoável. Resumindo, os estados iniciais correspondem a 5 estados de boa qualidade, um estado seqüencial e 4 estados de qualidade razoável.

4.2 Trabalho 2

O Trabalho 2 foi desenvolvido na Linguagem C. O algoritmo é uma implementação do SA simples, ou seja, adota-se o primeiro operador que resulte em resultados melhores. Não avalia-se toda a vizinhança a procura do melhor operador. Isso faz com que o número de iterações seja igual ao número de nós avaliados. Outro detalhe é a não utilização de uma taxa de redução de temperaturas. Optou-se por criar um cronograma (tabela) de temperaturas, o qual foi preenchido com uma função linear decrescente, calculada com base nas temperaturas final e inicial.

Os melhores parâmetros foram identificados a partir da observação da quantidade de soluções ruins que o algoritmo optava por trocar. Então, ajustou-se os parâmetros para que este número ficasse entre 10% e 20% do total de iterações do algoritmo

O Quadro 2 apresenta o algoritmo. A Figura 6 ilustra o fluxograma do trabalho.

```
Inicializa Cronograma com uma função decrescente de temperaturas
Solucao <- Solucao_Inicial
Custo <- Calcula_Custo(Solucao)
Operador <- primeiro operador disponível e ainda não aplicado
enquanto não esgotar operadores E não fizer MAXIMO_ITERACOES faça
    Aplica_Operador(Operador, Solucao)
    Custo_Novo <- Calcula_Custo(Solucao)
    se Custo_Novo >= Custo então
        Temperatura <- Consulta(Cronograma)
        Probabilidade <- e^(Diferenca_Custos / Temperatura)
        se Numero_aleatorio < Probabilidade
            Custo <- Custo_Novo
            Operador <- primeiro operador disponível e ainda não aplicado
        senão
            Desaplica_Operador(Operador, Solucao)
            Operador <- proximo operador disponível ainda não aplicado
    senão
        Custo <- Custo_Novo
        Operador <- primeiro operador disponível
fim_enquanto
```

Quadro 2 – Algoritmo do Trabalho 2

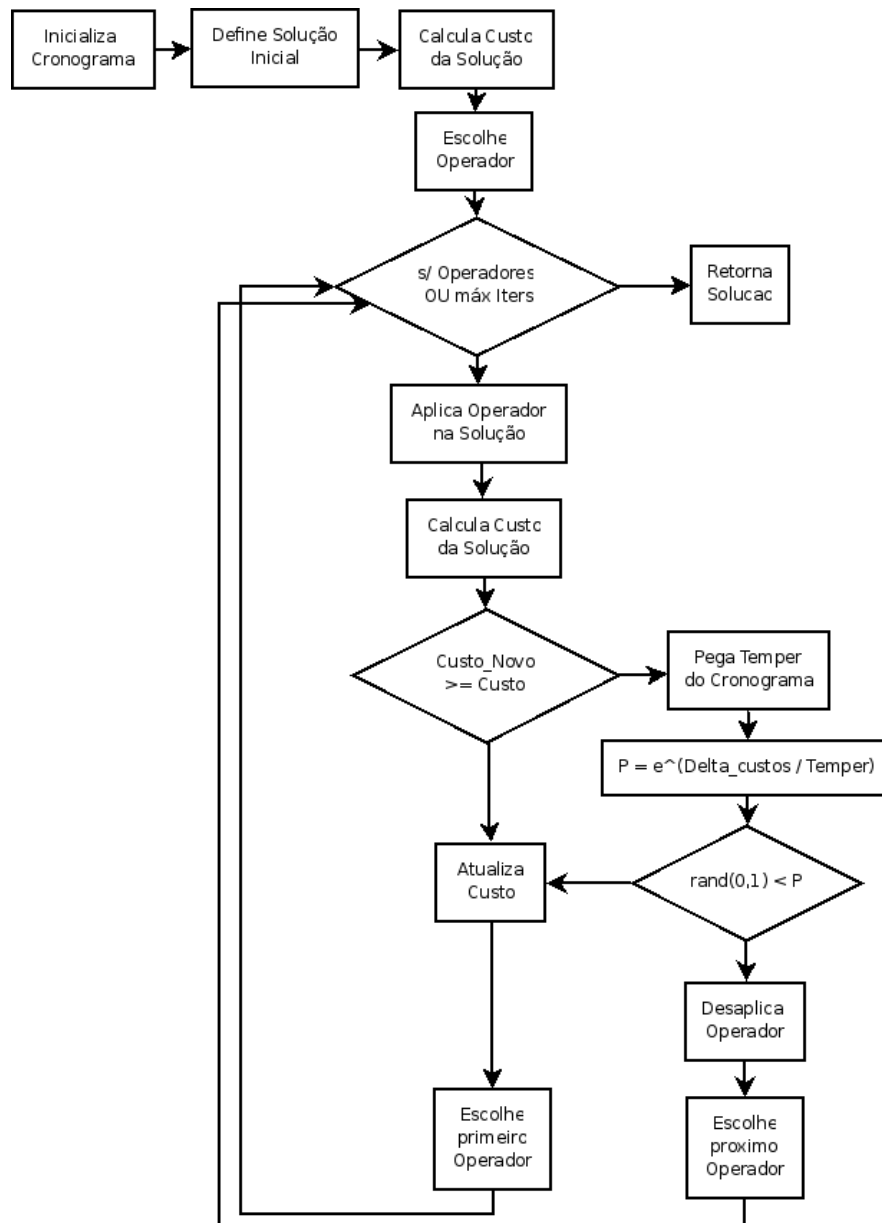


Figura 6 – Fluxograma do Trabalho 2.

4.3 Trabalho 3

O Trabalho 3 foi desenvolvido em C++. Os estados iniciais eram gerados aleatoriamente a partir da especificação de uma cidade de partida. Outra opção de geração dos estados iniciais era construir uma solução através da Busca Gulosa. Em todos os testes realizados inicialmente (inclusive utilizando a Busca Gulosa) utilizou-se a mesma cidade de partida para que as comparações realizadas pudessem ter a mesma base e, portanto, mais precisa sobre as melhorias encontradas.

O algoritmo possui apenas um laço de iteração. Esse laço é controlado pela temperatura e pelo número de iterações. Os parâmetros foram determinados pela avaliação do comportamento da função custo ao

longo das iterações do algoritmo para diferentes valores dos parâmetros. Observou-se assim que aumentando o número de iterações o algoritmo tende a resultados mais promissores. Entretanto, aumentando-se muito o número de iterações, quer por aumento da temperatura inicial ou por diminuição da velocidade de redução da temperatura, observamos que a melhora nos resultados não é significativa. O Quadro 3 apresenta o algoritmo. A Figura 7 ilustra o Fluxograma do Algoritmo.

```

SimulatedAnnealing
  nóAtual = CriarNóInicial()
  distância = Custo(nóAtual)
  enquanto (tatual < tfinal) e (repetições < limite) faça
    proximoNó = GerarPróximoResultado(nóAtual)
    delta = Custo(proximoNó) - distância
    se (delta < 0)
      nóAtual = proximoNó
      repetições = 0
      distância = distância + delta
    senão
      com probabilidade exp(-delta/tatual), faça
        nóAtual = proximoNó
        repetições = 0
        distância = distância + delta
    fim se
    tatual = tatual * fatorRedução
    repetições = repetições + 1
  fim enquanto
  retorna nóAtual
fim
  
```

Quadro 3 – Algoritmo do Trabalho 3.

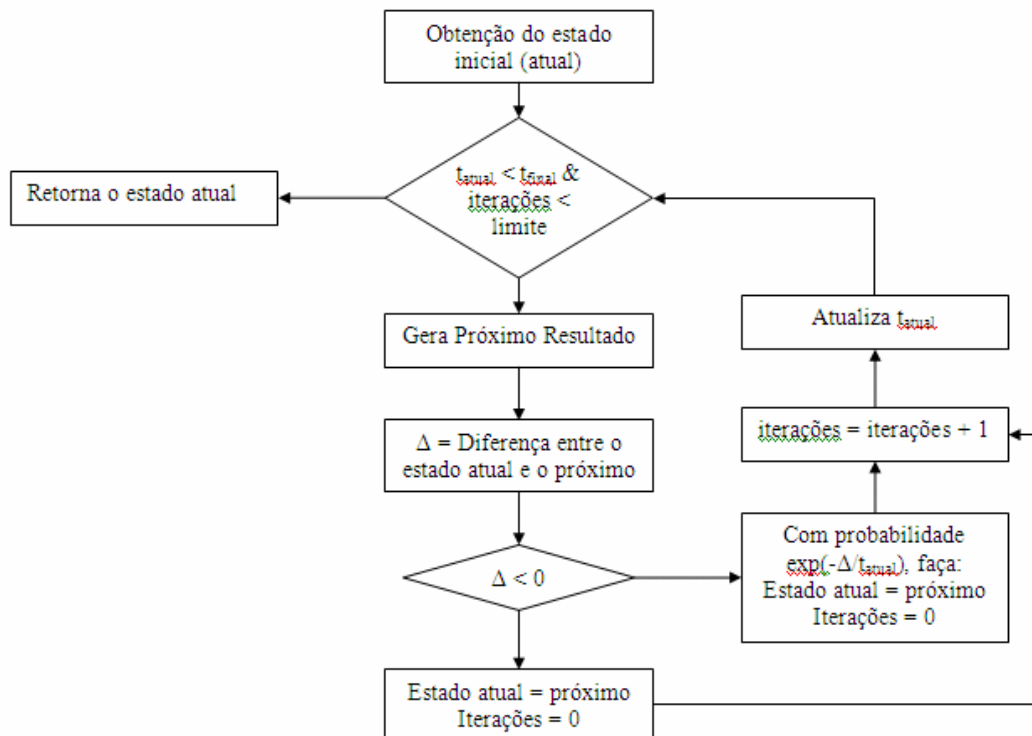


Figura 7 – Fluxograma do Trabalho 3

5. Resultados e Comparações Comuns

Nesta seção utilizou-se parâmetros e estados em comum. O objetivo desta seção é comparar os trabalhos utilizando os mesmo estados iniciais e mesmo parâmetros. Utilizou-se uma temperatura inicial baixa nesses testes.

5.1 - Geração dos Estados Iniciais

Os testes foram realizados a partir de dez estados iniciais gerados pelo Trabalho1. A seguir as tabelas (Tabela 1, Tabela 2, Tabela 3 e Tabela 4) referente aos estados iniciais e custos de cada instância [9]. Cada cidade é representada por um número. A cidade inicial é identificada por zero. O Custo Inicial corresponde a soma das distâncias entre as cidades, saindo da cidade zero e retornando a mesma.

	Estados Iniciais	Custo Inicial
1	0,39,4,2,8,26,25,27,29,24,5,33,34,38,37,40,3,6,7,36,35,32,31,30,23,9,11,12,13,14,15,16,18,17,19,21,10,22,20,28,41,1,	1178
2	0,41,3,24,25,32,34,38,36,4,33,39,37,7,9,23,26,28,20,18,19,16,17,15,14,13,12,21,22,10,11,8,5,35,30,31,27,29,6,2,1,40,	1184
3	0,39,38,1,3,8,30,32,33,29,9,22,16,12,14,13,15,17,18,19,20,28,25,5,6,24,26,7,10,11,21,27,23,2,37,36,34,31,35,4,40,41,	1197
4	0,4,35,39,37,38,40,2,6,24,27,25,30,31,29,33,3,5,23,28,32,36,34,8,9,21,20,19,18,17,16,15,14,13,12,11,10,22,26,7,1,41,	1200
5	0,1,37,36,31,32,28,27,20,16,19,18,17,15,13,14,12,10,23,9,8,24,29,5,4,39,2,26,6,3,25,21,22,11,7,33,34,38,35,30,40,41,	1204
6	0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,	699
7	0,6,14,12,22,19,28,24,32,30,37,34,1,40,5,3,11,15,20,17,25,27,35,8,16,13,21,18,26,23,31,39,2,41,7,10,36,38,9,4,29,33,	2332
8	0,37,3,7,5,8,13,10,18,15,23,20,28,25,33,41,39,4,1,9,14,12,19,16,24,22,32,29,34,40,2,21,26,31,36,6,17,35,38,27,30,11	2436
9	0,41,38,4,1,9,6,39,2,10,18,15,22,20,28,25,32,30,37,35,40,14,11,19,16,24,21,29,26,34,31,5,3,8,12,23,36,33,27,13,7,17,	2438
10	0,9,17,15,22,19,27,35,3,41,7,5,12,18,20,28,36,33,40,38,4,1,8,6,13,11,16,14,24,21,29,26,34,31,39,2,10,25,23,30,32,37	2217

Tabela 1 – Instância Dantzig – 42 cidades

	Estados Iniciais	Custo Inicial
1	0,2,9,12,14,4,3,5,1,13,11,10,16,19,15,7,6,8,18,17,20,21,25,22,23,24,	1242
2	0,12,20,16,17,18,15,8,7,6,9,14,4,3,5,2,1,13,11,10,19,25,21,22,23,24,	1272
3	0,13,11,12,10,9,16,15,18,8,6,3,5,1,14,2,4,7,19,17,20,25,21,22,23,24,	1276
4	0,11,10,14,9,7,8,12,13,2,1,4,5,3,6,15,19,17,20,16,18,23,22,25,21,24,	1305
5	0,1,14,12,2,13,11,10,17,25,16,8,7,6,3,4,5,9,18,19,15,20,21,22,23,24,	1306
6	0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25	1140
7	0,20,17,22,8,2,11,19,16,6,4,12,21,14,5,3,13,25,18,24,23,10,15,9,7,1,	2287
8	0,11,20,14,25,5,2,10,7,13,24,18,4,15,3,9,6,19,8,12,23,17,21,22,16,1,	2558
9	0,24,22,12,17,8,6,11,16,25,19,5,21,1,3,15,9,18,23,2,14,20,13,4,10,7,	2625
10	0,19,5,21,1,24,3,15,9,18,23,2,8,7,12,17,20,25,11,14,22,6,16,13,10,4,	2976

Tabela 2 – Instância Fri – 26 cidades

	Estados Iniciais	Custo Inicial
1	0,47,25,32,7,9,30,11,4,21,8,26,16,31,20,13,5,10,28,12,27,6,17,24,2,45,42,40,18,3,29,44,46,39,1,34,37,19,38,41,36,23,14,15,35,22,33,43,	8049
2	0,12,10,8,21,35,28,15,6,45,24,1,37,36,19,38,39,41,34,29,44,42,46,17,47,25,7,30,14,32,11,23,9,4,20,26,31,16,13,5,27,40,33,22,2,18,3,43,	8430
3	0,43,27,5,6,10,12,33,3,29,18,42,45,17,2,22,25,47,35,15,28,46,40,24,1,41,19,37,34,44,39,38,36,14,23,9,11,4,30,32,7,8,20,26,16,31,21,13,	8432
4	0,10,35,15,47,33,46,42,40,43,6,27,5,20,8,13,21,32,4,11,9,30,7,31,26,16,1,44,24,18,3,2,36,19,29,37,34,41,38,39,23,14,25,17,12,28,22,45,	8449
5	0,10,47,28,40,43,24,17,42,3,18,2,33,22,44,46,45,15,27,6,35,21,5,7,8,13,20,16,26,31,36,9,11,32,4,30,14,23,1,34,41,37,29,19,38,39,25,12,	8460
6	0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47	19837
7	0,28,10,17,26,46,25,15,16,27,36,8,6,33,5,14,22,31,3,23,13,20,12,1,29,18,7,44,34,43,2,32,11,40,47,9,38,30,35,45,4,42,39,24,41,19,21,37,	21000
8	0,34,24,3,23,32,43,4,41,40,1,21,30,38,29,19,31,47,46,36,28,35,44,16,26,45,37,5,25,42,2,33,22,8,7,27,6,39,11,20,9,17,15,12,14,13,18,10,	19767
9	0,34,24,3,23,32,43,4,41,40,1,21,30,38,29,19,31,47,46,36,28,35,44,16,26,45,37,5,25,42,2,33,22,8,7,27,6,39,	19767

	11,20,9,17,15,12,14,13,18,10,	
10	0,22,31,39,20,47,37,28,30,29,18,45,17,27,34,43,15,35,25,32,24,13,40,12,42,14,41,11,8,33,6,21,3,4,19,1,9,10,26,46,7,44,16,5,2,38,36,23,	22086

Tabela 3 – Instância GR – 48 cidades

	Estados Iniciais	Custo Inicial
1	0,41,43,26,36,18,1,10,47,25,27,11,28,31,24,35,33,5,7,6,46,9,39,17,37,13,21,12,8,29,22,2,4,32,14,23,34,30,19,16,40,42,20,38,44,45,3,15,	17366
2	0,10,16,34,40,2,4,25,14,20,32,42,47,11,9,39,21,17,13,35,37,8,29,22,27,28,31,24,6,7,33,5,46,12,23,30,19,3,45,44,38,43,26,36,18,1,15,41,	17400
3	0,15,34,16,40,11,4,32,25,28,27,35,21,12,13,2,39,33,5,7,6,46,9,22,29,8,24,31,37,17,30,19,3,45,44,38,43,26,36,41,10,23,14,18,1,47,20,42,	17485
4	0,1,10,15,23,27,24,11,6,12,40,25,14,20,42,47,28,21,37,31,13,2,32,4,22,29,8,9,17,46,7,5,35,33,39,16,34,19,30,3,45,44,38,43,26,41,36,18,	17488
5	0,14,40,16,47,11,30,21,13,9,35,6,37,17,46,7,5,33,39,31,12,23,28,27,8,29,22,4,24,2,25,20,32,42,1,10,15,34,19,3,41,45,44,38,43,26,36,18,	17619
6	0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,	48170
7	0,13,4,42,21,41,2,10,19,39,18,8,47,1,37,16,36,46,5,14,34,6,44,3,43,32,11,31,33,12,27,25,40,22,23,38,20,28,29,45,17,26,15,35,24,30,9,7,	51538
8	0,34,26,16,43,15,24,31,41,13,40,29,21,38,20,36,28,18,45,17,33,23,12,3,2,39,47,46,22,30,11,19,27,8,10,9,25,7,14,44,5,4,42,6,35,32,1,37,	53054
9	0,37,34,44,16,43,32,24,31,41,13,22,29,21,11,30,39,38,28,19,27,36,8,17,20,9,18,25,6,33,23,15,7,5,4,2,3,40,1,10,47,45,26,35,42,46,14,12,	51614
10	0,21,41,3,10,19,39,11,40,1,8,38,16,37,46,18,17,6,5,15,35,44,43,33,2,22,31,23,32,4,12,20,29,45,27,34,42,24,25,14,30,28,9,36,26,13,47,7,	48621

Tabela 4 – Instância HK – 48 cidades

5.2 Geração dos Melhores Parâmetros

Para esses testes foram utilizados os parâmetros gerados pelo Trabalho1. A Seção 4.1 explica a geração.

A partir dos estados iniciais apresentados na seção anterior, para cada trabalho foram obtidos os seguintes resultados. Para os estados iniciais identificados de 1 a 10, cada trabalho apresentou o custo final, número de iterações e avaliações realizados. O custo final corresponde a soma das distâncias do estado final encontrado. A profundidade da árvore é o número de iterações. O número de nós expandidos e avaliados correspondem as avaliações.

Para os resultados abaixo, todos os grupos utilizaram os mesmos parâmetros. Seguem os parâmetros:

- Temperatura Final: 0.1
- Temperatura Inicial: 30
- Número de Iterações: 60% do tamanho da matriz
- Taxa de Redução de Temperatura: 0.2%

5.3 Resultados Obtidos

As Tabelas 5, 6, 7 e 8 apresentam os resultados obtidos para instância pelos Trabalhos 1, 2 e 3. Lembrando que os resultados foram obtidos utilizando-se os mesmos estados iniciais e mesmos parâmetros enfatizando uma temperatura baixa.

	Trabalho 1			Trabalho 2			Trabalho 3		
	Custo Final	Iterações	Avaliações	Custo Final	Iterações	Avaliações	Custo Final	Iterações	Avaliações
1	1109	68765	71205	1113	999999	999999	1178	4	4
2	973	59272	71205	1072	999999	999999	1184	4	4
3	1086	49575	71205	1171	999999	999999	1197	4	4
4	1080	60255	71205	1147	999999	999999	1200	4	4
5	1093	61281	71205	1194	999999	999999	1204	4	4
6	699	41109	71205	699	44	44	699	4	4
7	1951	54263	71205	2233	999999	999999	1178	4	4

8	1828	38382	71205	2243	999999	999999	1184	4	4
9	1880	41905	71205	2369	999999	999999	1197	4	4
10	1628	59782	71205	2013	999999	999999	1200	4	4

Tabela 5- Resultado instância Dantzig 42 – parâmetros idênticos

	Trabalho 1			Trabalho 2			Trabalho 3		
	Custo Final	Iterações	Avaliações	Custo Final	Iterações	Avaliações	Custo Final	Iterações	Avaliações
1	1224	4553	42750	1242	999999	999999	1238	4	4
2	1252	3998	42750	1276	999999	999999	1276	4	4
3	1216	4453	42750	1276	76	76	1276	4	4
4	1134	4433	42750	1305	171	171	1305	4	4
5	1277	5011	42750	1306	999999	999999	1306	4	4
6	1003	4439	42750	1140	999999	999999	1140	4	4
7	1999	4887	42750	2287	999999	999999	2267	4	4
8	1854	4386	42750	2558	999999	999999	2267	4	4
9	1858	4398	42750	2625	215	215	2527	4	4
10	2322	4382	42750	2976	202	202	2741	4	4

Tabela 6- Resultado instância Fri 26 – parâmetros idênticos

	Trabalho 1			Trabalho 2			Trabalho 3		
	Custo Final	Iterações	Avaliações	Custo Final	Iterações	Avaliações	Custo Final	Iterações	Avaliações
1	6738	22161	71250	7554	999999	999999	8049	4	4
2	7381	44522	71250	7379	999999	999999	8269	4	4
3	7504	53343	71250	8160	999999	999999	8432	4	4
4	8078	58640	71250	8215	999999	999999	8449	4	4
5	7584	61066	71250	8206	999999	999999	8460	4	4
6	14468	31566	71250	18256	999999	999999	18970	4	4
7	15127	46719	71250	20954	999999	999999	20651	4	4
8	15958	52760	71250	18875	999999	999999	19767	4	4
9	15579	53604	71250	18785	999999	999999	19767	4	4
10	16144	54602	71250	21163	999999	999999	21427	4	4

Tabela 7- Resultado instância Gr48 – parâmetros idênticos

	Trabalho 1			Trabalho 2			Trabalho 3		
	Custo Final	Iterações	Avaliações	Custo Final	Iterações	Avaliações	Custo Final	Iterações	Avaliações
1	16512	7708	71250	16512	228	228	17366	4	4
2	16366	12914	71250	16284	284	284	17400	4	4
3	16790	7285	71250	16571	111	111	17485	4	4
4	15619	8940	71250	15783	245	245	17488	4	4
5	15443	5437	71250	15443	288	288	17619	4	4
6	36636	18781	71250	37161	484	484	47828	4	4
7	35462	4590	71250	35343	730	730	51437	4	4
8	39182	39349	71250	36557	954	954	51734	4	4
9	37172	5365	71250	36401	773	773	51220	4	4
10	38228	8532	71250	36083	723	723	48276	4	4

Tabela 8- Resultado instância Hk48 – parâmetros idênticos

Além das diferenças na obtenção do custo final, as tabelas apresentam algumas características dos algoritmos como número de iterações e avaliações. O algoritmo 1 apresenta diferenças entre iterações e avaliações pelo fato de possuir dois laços. Um para iterações e outro para redução de temperatura, mais precisamente, a temperatura somente é reduzida após um determinado número de iterações.

5.4 Gráficos comparativos das soluções

A seguir os gráficos comparativos das soluções obtidas de cada trabalho.

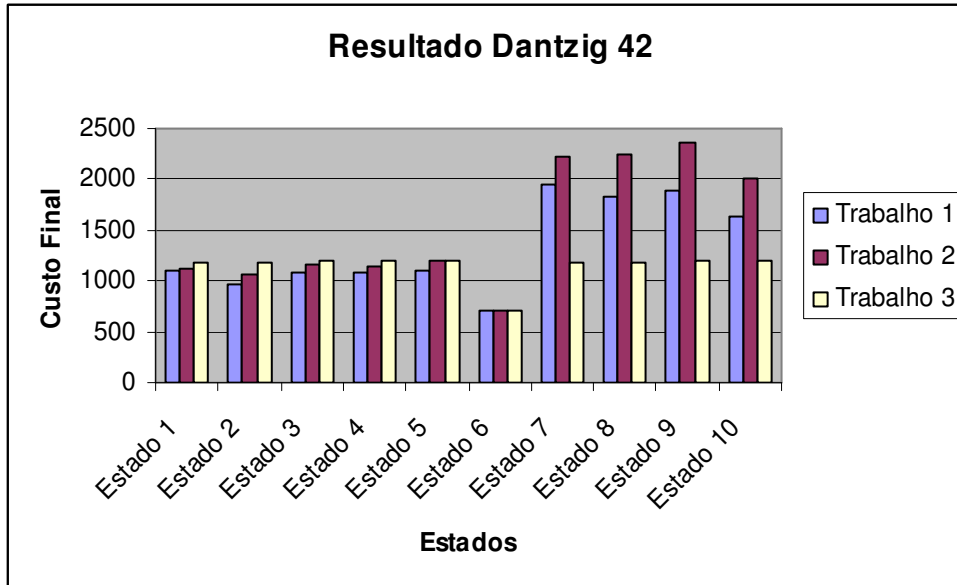


Figura 8 - Gráfico Instância Dantzig 42.

A Figura 5 ilustra os resultados para a instância Dantzig 42. No gráfico verifica-se que o trabalho 1 obteve os melhores resultados para os 6 estados iniciais, porém para os 4 últimos o trabalho 3 obteve melhores resultados. Lembre-se que os 6 primeiros estados foram estados com estado inicial razoável. Porém os 4 últimos são considerados estados com custo alto.

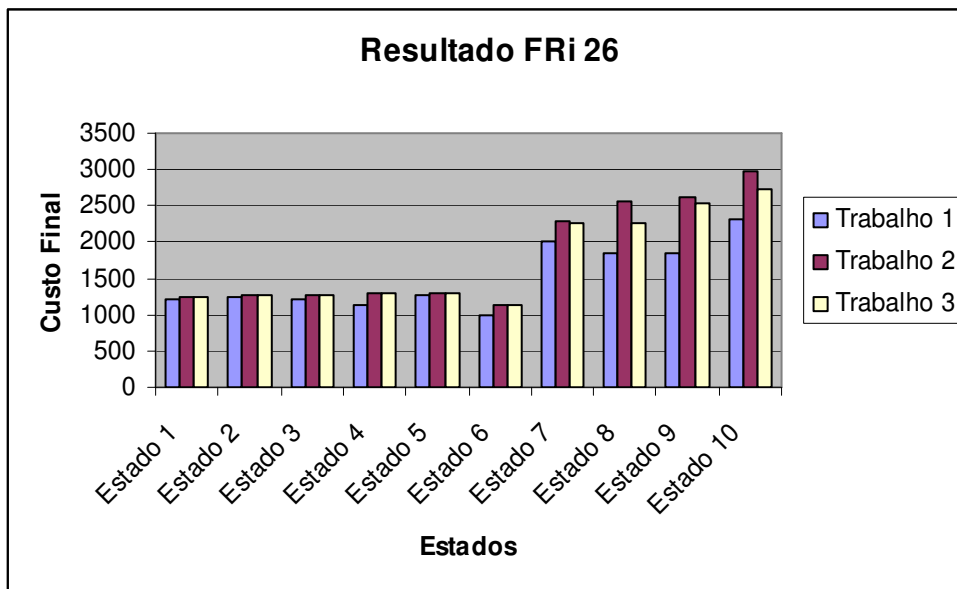


Figura 9 - Gráfico Instância Fri 26

Na Figura 9 e Figura 10 são apresentados os gráficos para a instância Fri26 e Gr48 respectivamente. Observa-se que o trabalho 1 obteve melhores resultados para as duas instâncias em todos os estados.

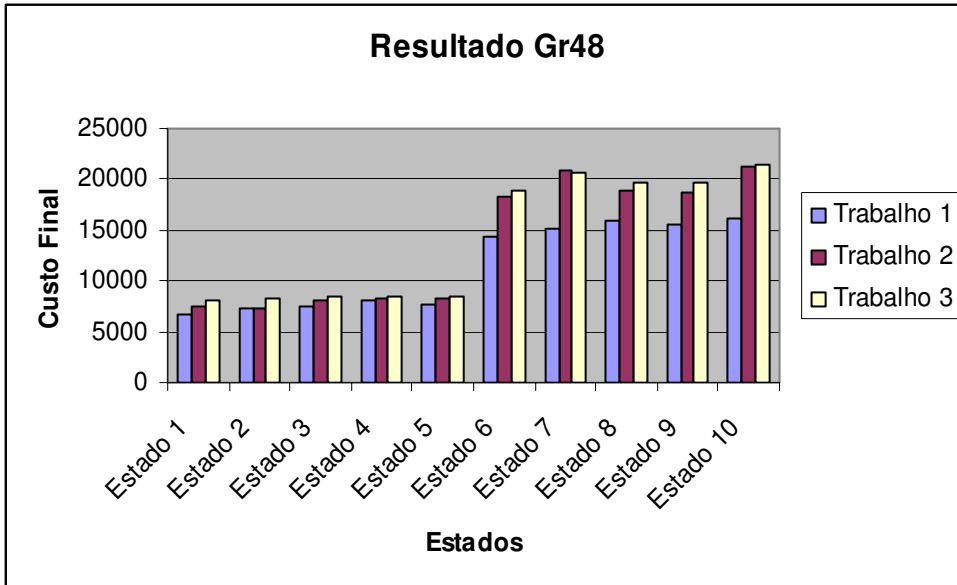


Figura 10 - Gráfico Instância Gr48

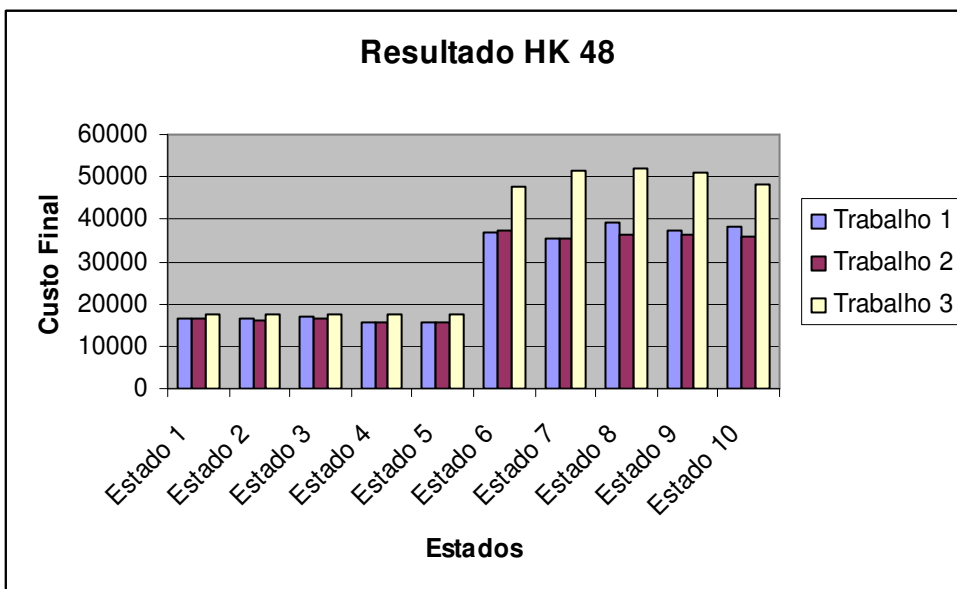


Figura 11 - Gráfico Instância Hk48

A Figura 11 ilustra os resultados da instância Hk8. Observa-se um equilíbrio entre os 3 trabalhos nos resultados dos 6 primeiros estados. Porém nos 4 últimos estados o trabalho 2 obteve melhores resultados.

6. Resultados e Comparações Individuais de cada grupo

Nesse capítulo, cada trabalho realizou seus testes, ou seja, escolheu os parâmetros que melhor se adaptaram ao seu algoritmo. Mesmo assim, para análise comparativa foram utilizados os mesmo estados iniciais da Seção 5.

6.1 Parâmetros do Trabalho 1

- Temperatura Final: 1
- Temperatura Inicial: 500
- Número de Iterações: 60% do tamanho da matriz
- Taxa de Redução de Temperatura: 0,2%

6.2 Parâmetros do Trabalho 2

- Temperatura Final: 1;
- Temperatura Inicial: 100000;
- Número de Iterações: (obs.: não utilizamos este parâmetro);
- Taxa de Redução de Temperatura: Utilizou-se a seguinte função decrescente de temperaturas: $t(x) = \text{MAX_ITER}/((x+1)*10)$, onde MAX_ITER = 1000000 e 'x' é o número da iteração;

6.3 Parâmetros do Trabalho 3

- Temperatura Final: 10.0
- Temperatura Inicial: 500.0
- Número de Iterações: 100
- Taxa de Redução de Temperatura: 0.9999

6.4 Resultados Obtidos

As Tabelas 9, 10, 11 e 12 apresentam os resultados obtidos em cada teste.

	Trabalho 1			Trabalho 2			Trabalho 3		
	Custo Final	Iterações	Avaliações	Custo Final	Iterações	Avaliações	Custo Final	Iterações	Avaliações
1	1109	73724	74520	1071	100194	100194	1016	39119	39119
2	973	73802	74520	1080	100089	100089	1036	39119	39119
3	1086	73020	74520	1121	100046	100046	1066	39119	39119
4	1080	70505	74520	1062	100097	100097	1091	38755.8	38755.8
5	1093	41109	74520	1130	100192	100192	1074	38890.8	38890.8
6	699	54263	74520	699	100046	100046	1063	39066.3	39066.3
7	1951	38382	74520	1946	100292	100292	1099	39119	39119
8	1828	41905	74520	1960	100376	100376	1064	39031.7	39031.7
9	1880	59782	74520	1913	100430	100430	1066	39119	39119
10	1628	41109	74520	1740	100266	100266	1109	39119	39119

Tabela 9- Resultado instância Dantzig 42 – parâmetros diferentes

	Trabalho 1			Trabalho 2			Trabalho 3		
	Custo Final	Iterações	Avaliações	Custo Final	Iterações	Avaliações	Custo Final	Iterações	Avaliações
1	1224	71402	74520	1204	5321	5321	1132	39119	39119
2	1241	71942	74520	1343	100044	100044	1070	39119	39119
3	1216	12118	74520	1226	15740	15740	1080	39119	39119
4	1134	73622	74520	1134	20794	20794	1096	39119	39119
5	1155	73523	74520	1279	100075	100075	1081	39119	39119
6	1003	72676	74520	1077	100065	100065	1076	39119	39119
7	1882	73177	74520	1831	100078	100078	1098	39119	39119
8	1854	73901	74520	2128	100191	100191	1063	39119	39119
9	1785	73361	74520	1746	21050	21050	1074	39119	39119
10	2300	69400	74520	2300	100054	100054	1027	38941.9	38941.9

Tabela 10- Resultado instância Fri26 – parâmetros diferentes

	Trabalho 1			Trabalho 2			Trabalho 3		
	Custo Final	Iterações	Avaliações	Custo Final	Iterações	Avaliações	Custo Final	Iterações	Avaliações
1	6738	67124	74520	6985	100269	100269	7026	24885.8	24885.8
2	6564	70175	74520	7379	100298	100298	6776	24551	24551
3	7215	42409	74520	8117	100299	100299	7229	25163.8	25163.8
4	6512	72746	74520	8149	100075	100075	6567	25883.7	25883.7
5	7584	72769	74520	7584	100183	100183	6779	25483.8	25483.8
6	14468	72810	74520	13835	100721	100721	6882	25308	25308
7	14891	71897	74520	16672	100621	100621	6734	25146.2	25146.2
8	15511	71740	74520	17404	100436	100436	7210	24652.1	24652.1
9	15579	72525	74520	17404	100436	100436	7112	24717.8	24717.8
10	15257	72750	74520	21163	100833	100436	6883	25081.8	25081.8

Tabela 11- Resultado instância Gr48 – parâmetros diferentes

	Trabalho 1			Trabalho 2			Trabalho 3		
	Custo Final	Iterações	Avaliações	Custo Final	Iterações	Avaliações	Custo Final	Iterações	Avaliações
1	15512	79154	74520	15691	4065	4065	15808	16670.5	16670.5
2	14366	60837	74520	16009	7900	7900	15884	16958.7	16958.7
3	14790	47002	74520	16641	5562	5562	15670	17118.8	17118.8
4	15518	48363	74520	15456	3845	3845	15865	17016.7	17016.7
5	14443	52644	74520	15443	4192	4192	14751	18606	18606
6	15166	64828	74520	34256	6155	6155	15559	17613.4	17613.4
7	35744	56372	74520	35107	9922	9922	15469	18447.8	18447.8
8	20443	69942	74520	35109	18124	18124	15717	18147.7	18147.7
9	33113	56252	74520	32142	10317	10317	15627	17845.1	17845.1
10	38844	45682	74520	35980	8214	8214	16244	15778.6	15778.6

Tabela 12- Resultado instância Hk48 – parâmetros diferentes

6.5 Comparações dos Resultados

A seguir os gráficos comparativos de cada trabalho.

7. Conclusões

A seguir as conclusões de cada trabalho.

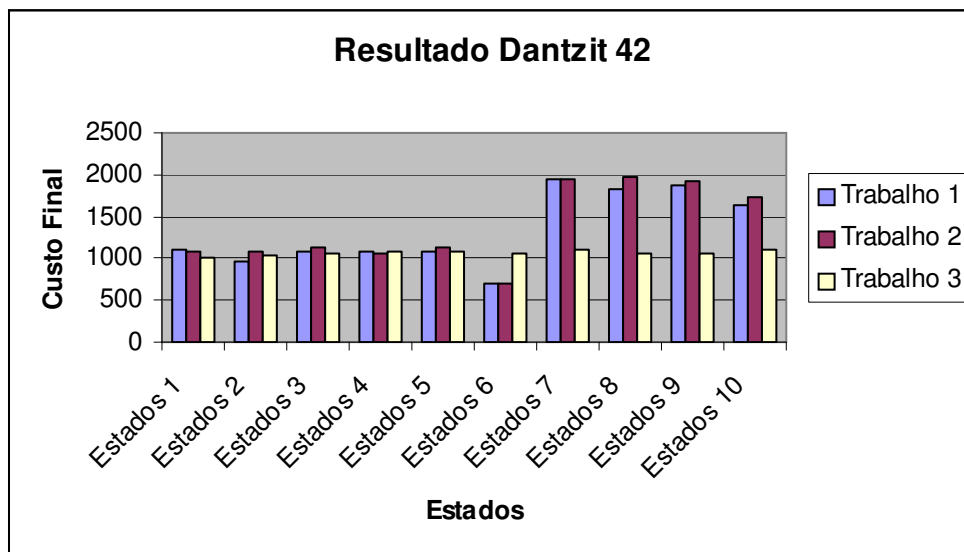


Figura 12 - Gráfico Instância Dantzig 42 – parâmetros diferentes

A Figura 12 ilustra resultado para a instância Dantzig 42. Observa-se que houve um equilíbrio nos primeiros estados, porém o Trabalho 3 obteve resultados excelentes nos 4 últimos estados, onde o estado inicial não é bom.

A Figura 13 ilustra resultado para a instância Fri 26. Assim como na figura 12, houve um equilíbrio nos primeiros estados, porém a superioridade do Trabalho 3 foi maior para os 4 últimos estados.

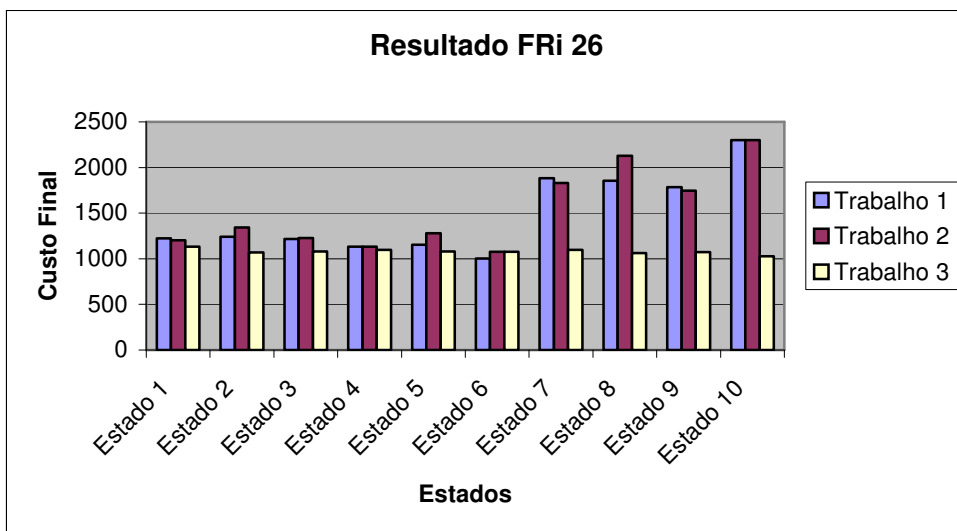


Figura 13 - Gráfico Instância Fri 26 – parâmetros diferentes

Para a instância GR48 o Trabalho 1 obteve melhores resultados (com pouca diferença) nos 5 melhores estados. Porém os últimos 5 estados o Trabalho 3 foi melhor.

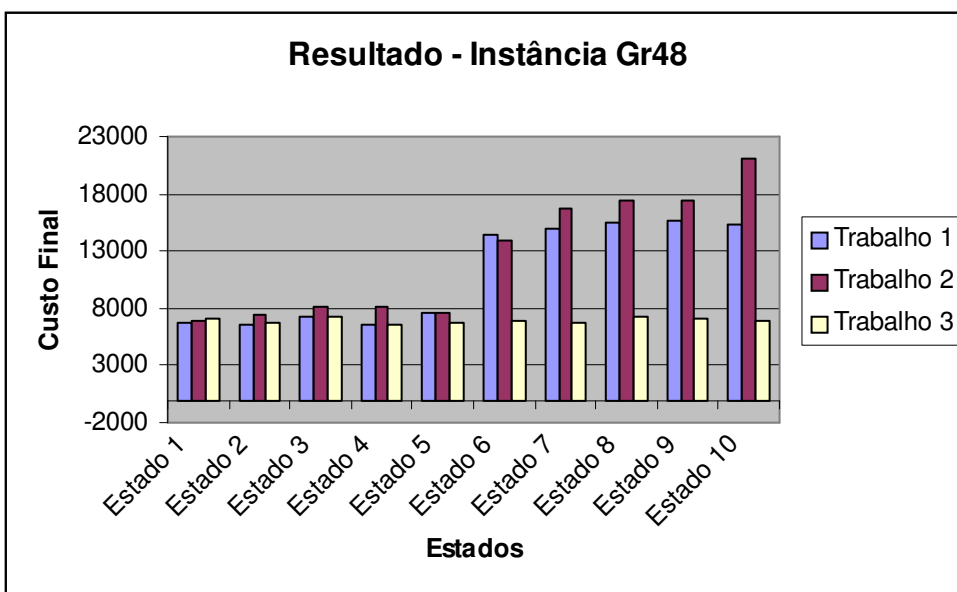


Figura 14 - Gráfico Instância Gr 48 – parâmetros diferentes

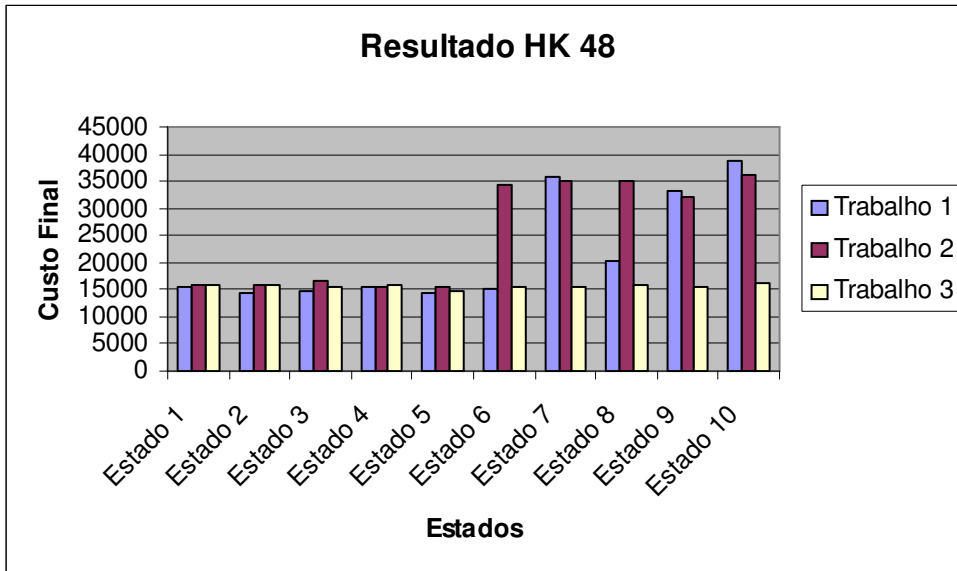


Figura 15- Gráfico Instância Hk 48 – parâmetros diferentes

Para os testes demonstrados acima observou-se uma regularidade na obtenção de melhores resultados entre o Trabalho 1 e Trabalho 3. Em algumas instâncias um Trabalho é superior a outro.

7.1 Conclusões do Trabalho 1

“A principal vantagem do Simulated Annealing é a tentativa de convergir para uma solução ótima. Como desvantagem, relaciona-se a o tempo de processamento e a calibragem dos parâmetros.

A escolha da taxa de redução de temperatura, temperatura inicial e final é muito importante. Escalas de temperatura (taxa de redução) que permite um resultado bom (ótimo global) não são muito adotadas em aplicações por serem lentas.

O SA não garante que a solução encontrada seja a melhor possível. Depende do estado inicial e do problema onde está sendo aplicado. Os parâmetros mais adequados para uma dada aplicação podem ser estabelecidos por experimentação.

O SA é semelhante ao Hill Climbing quando a temperatura está estabilizada. Se o SA não aceita piores estados, então é Hill Climbing. O SA é ótimo e completo quando o resfriamento é suave com muitas iterações para o equilíbrio de temperatura.”

7.2 Conclusões do Trabalho 2

“Observou-se que um mesmo cronograma de temperatura pode não funcionar bem para todos os problemas. Usando uma curva decrescente senoidal, não conseguimos bons resultados para os problemas hk48 e fri26. O melhor cronograma que obtivemos foi com uma função decrescente retilínea. Um fator que foi usado para considerar se um cronograma estava bom ou não foi a quantidade de soluções ruins que o algoritmo optava por trocar. Julga-se que, se o algoritmo fazia X iterações, então em torno de 10% a 20% dessas iterações deveriam optar pela solução modificada, ao invés de manter a solução corrente.”

7.3 Conclusões do Trabalho 3

“Se o algoritmo utilizar um maior fator de redução da temperatura é mais eficaz que uma Temperatura Inicial alta. Para grandes valores de Fator de Redução, aumentar a Temperatura Inicial não melhora o resultado. Usar a busca gulosa para geração de estados iniciais geralmente piora o resultado nas primeiras iterações. O algoritmo girou em torno de 86,9% do ótimo. “

7.4 Conclusões Gerais

O Trabalho 1 possui como principal diferença para os demais trabalhos a utilização de dois laços de temperatura. O laço principal é controlado apenas pela temperatura, enquanto que um laço intermediário é utilizado para realizar um número de iterações. Isso caracteriza a existência de mais um parâmetro no Trabalho 1 chamado de iterações. Esse parâmetro informa qual é o número de iterações necessários para aplicar uma taxa de redução de temperatura. O Trabalho 2 e 3 são muito semelhantes. Possuem apenas um laço de repetição que é controlado pela temperatura e por um número de iterações. O Trabalho 2 possui uma agenda de temperatura que pode ser aplicado.

Os testes realizados sempre com os mesmos estados iniciais mostraram a evolução dos trabalhos. Na seção 5 foram apresentados testes utilizando uma temperatura inicial baixa. Para a maioria dos estados o Trabalho 1 obteve melhores resultados. Apenas para a instância Dantzig 42 o Trabalho 3 obteve melhores resultados para os últimos estados. Na seção 6 cada trabalho utilizou os parâmetros que considera melhor. As temperaturas iniciais adotadas foram altas, porém os resultados mostraram uma regularidade na redução dos custos entre o Trabalho 1 e Trabalho 3 com uma superioridade do Trabalho 3 nos estados iniciais considerados “ruins” (estado 6 à 10). Porém nos primeiros estados (1 à 5) o Trabalho 1 obteve pequena diferença na redução de custos para o Trabalho 3. Isso deve-se ao fato do Trabalho 1 intensificar a busca local com utilização de dois laços de repetição no algoritmo.

Outra conclusão obtida é relacionada a redução dos custos. Quando o estado inicial é considerado bom (os cinco primeiros estados) a redução de custos não é alta. Porém quando os estados iniciais são considerados ruins, a redução é bastante significativa. As Figuras 16 e 17 mostram as reduções de custos para os estados nas bases Dantzig 42 e Fri26. Foi realizada uma média entre todos os trabalhos referente a diferença entre o custo inicial e custo final gerado.

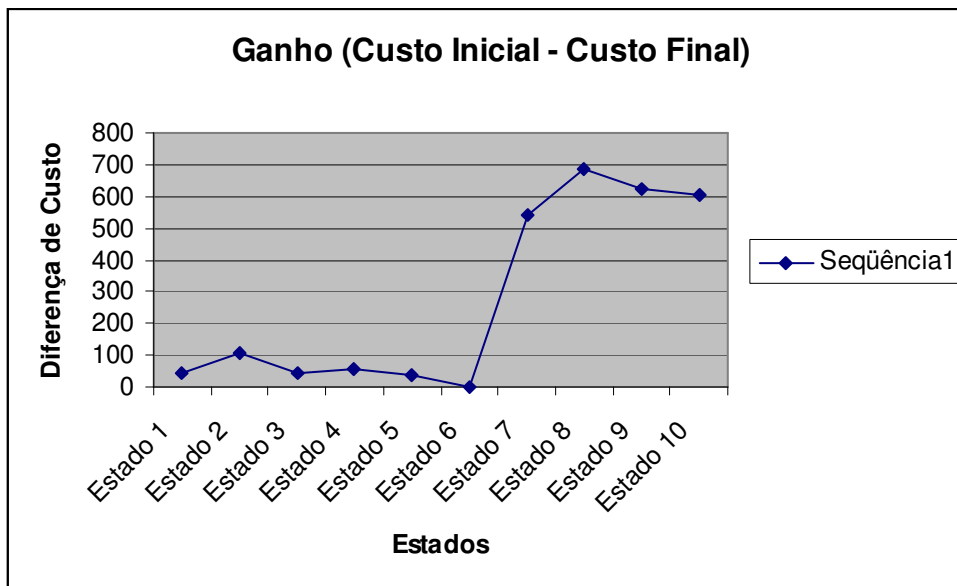


Figura 16 – Redução de custos base Dantzig 42

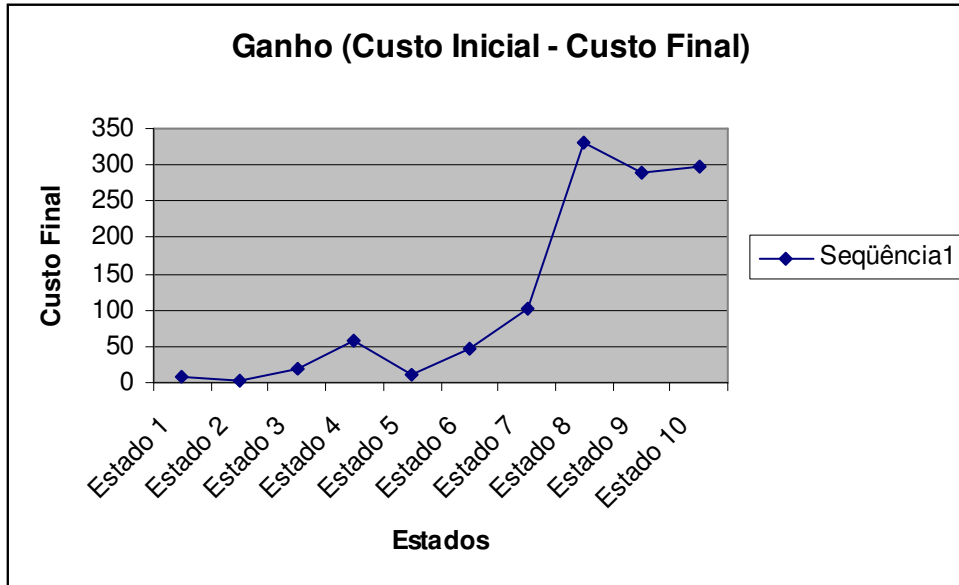


Figura 17 – Redução de custos base Fri 26

O SA em todos os algoritmos trabalha com diversificação quando a temperatura corrente é alta. A medida que a temperatura diminui, a diversificação também diminui e o SA passa a intensificar a busca. Isso ocorre devido a probabilidade de aceitar estados considerados “ruins”.

Referências

- [1] AARTS E. e KORST J., *Simulated Annealing and BoltzmannMachine*, John Wiley, 1989. DOWSLAND, K.A. *Simulated Annealing*, In Reeves, C.R. (ed), *Modern Heuristic Techniques for Combinatorial Problems*, Blackwell Scientific Publications, 20-69, 1993.
- [2] Cerny, V., "Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm", *J. Opt. Theory Appl.*, 45, 1, 41-51, 1985.
- [3] Kirkpatrick, S., C. D. Gelatt Jr., M. P. Vecchi, "Optimization by Simulated Annealing", *Science*, 220, 4598, 671-680, 1983.
- [4] Metropolis, N., A. Rosenbluth, M. Rosenbluth, A. Teller, E. Teller, "Equation of State Calculations by Fast Computing Machines", *J. Chem. Phys.*, 21, 6, 1087-1092, 1953.
- [5] Press, Wm. H., B. Flannery, S. Teukolsky, Wm. Vetterling, *Numerical Recipes*, 326-334, Cambridge University Press, New York, NY, 1986.
- [6] Simulated Annealing. Disponível em: <http://mathworld.wolfram.com/SimulatedAnnealing.html>.
- [7] About Simulated Annealing. Disponível em: http://home.att.net/~srschmitt/sa_demo/SA-demo_1.html.
- [8] Traveling salesman problem (TSP) using Simulated Annealing. Disponível em: <http://www.math.uu.nl/people/beukers/anneal/anneal.html>.
- [9] TSPLIB. Disponível em: <http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsplib.html>.