



**GUSTAVO PEREIRA ROSA**

***SIMULATED ANNEALING* APLICADO AO  
PROBLEMA GERAL DE DIMENSIONAMENTO DE  
LOTES E PROGRAMAÇÃO DA PRODUÇÃO**

**LAVRAS - MG  
2010**

**GUSTAVO PEREIRA ROSA**

***SIMULATED ANNEALING* APLICADO AO PROBLEMA GERAL DE  
DIMENSIONAMENTO DE LOTES E PROGRAMAÇÃO DA PRODUÇÃO**

Monografia de graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do curso de Ciência da Computação para obtenção do título de Bacharel em Ciência da Computação.

Orientador:

Prof. Dr. Cláudio Fabiano Motta Toledo

**LAVRAS - MG  
2010**

**GUSTAVO PEREIRA ROSA**

***SIMULATED ANNEALING* APLICADO AO PROBLEMA GERAL DE  
DIMENSIONAMENTO DE LOTES E PROGRAMAÇÃO DA PRODUÇÃO**

Monografia de graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do curso de Ciência da Computação para obtenção do título de Bacharel em Ciência da Computação.

APROVADA em \_\_\_\_ de \_\_\_\_\_ de \_\_\_\_\_

Dr. Joaquim Quinteiro Uchôa UFLA

Dr. Wilian Soares Lacerda UFLA

Prof. Dr. Cláudio Fabiano Motta Toledo

Orientador

**LAVRAS - MG**

**2010**

*A meu pai Eduardo.*

*A minha mãe Claudete.*

*A minha namorada Jéssica.*

*A minha irmã Gabrielle.*

DEDICO.

## **AGRADECIMENTOS**

Agradeço ao meu professor, orientador e amigo Cláudio, que me deu a oportunidade e o conhecimento para realizar este trabalho.

Agradeço a minha família por me apoiar em todos os momentos e que foram os responsáveis por me fazer chegar até aqui.

A minha namorada, pelo carinho e amor que sente por mim e que sempre me deu forças quando precisei.

A todos os outros professores do DCC – UFLA que, durante toda a graduação, conseguiram fazer com que eu aprendesse muito.

A meus amigos de curso, comp2007/1, que se manteve unida do primeiro ao último período.

E agradeço a todos que tornaram possível a realização deste trabalho.

## RESUMO

Este trabalho propõe uma metaheurística, denominada *Simulated Annealing*, para solucionar o Problema Geral de Dimensionamento de Lotes e Programação da Produção (PGDLPP) com e sem máquinas paralelas, e com penalização para demandas não atendidas. Um conjunto de instâncias, baseadas em parâmetros utilizados na literatura, é solucionado pelo método proposto e comparado com um método exato que serve como base para avaliação das soluções. Os resultados revelam o melhor desempenho obtido pela metaheurística proposta em comparação com outras metaheurísticas encontradas na literatura.

Palavras-chave: *Simulated Annealing*. Programação da produção. Dimensionamento de lotes. Metaheurística.

## **ABSTRACT**

This work proposes a metaheuristic named, Simulated Annealing, to solve the General Lot sizing and Scheduling Problem (GLSP) with and without parallel machines, and with penalties for demand shortcoming. A set of instances, based on parameters used in the literature, is solved by the proposed method and it is compared with an exact method which is used as a benchmark to evaluate the solutions. The results report the better performance found by the proposed metaheuristic in comparison to other metaheuristic found in the literature.

**Keywords:** Simulated Annealing. Scheduling. Lot sizing. Metaheuristic.

## LISTA DE ILUSTRAÇÕES

Figura 1 Variação dos micro-períodos .....	21
Figura 2 Matriz de sequências .....	23
Figura 3 Algoritmo <i>Greedy-Mod</i> .....	25
Figura 4 Método para atualizar $X_{ij}$ , $K_{it}$ e $D_j$ .....	26
Figura 5 Algoritmo <i>Simulated Annealing</i> .....	28
Figura 6 Algoritmo de ajuste adaptativo da Temperatura Inicial .....	29
Figura 7 Inserção de produto .....	30
Figura 8 Remoção de produto .....	30
Figura 9 Troca de produto .....	31
Figura 10 Operação de reparo da sequência .....	31
Figura 11 Desvios médios dos diferentes métodos <i>Simulated Annealing</i> para instancias de máquinas simples .....	37
Figura 12 Tempos médios dos diferentes métodos <i>Simulated Annealing</i> para instâncias de máquinas simples .....	38
Figura 13 Vitórias dos métodos <i>Simulated Annealing</i> sobre o AMPL .....	42
Figura 14 Desvios médios dos diferentes métodos <i>Simulated Annealing</i> para instâncias de máquinas paralelas P1, P2 e P3 .....	42
Figura 15 Desvios médios dos diferentes métodos <i>Simulated Annealing</i> para instâncias de máquinas paralelas P4, P5 e P6 .....	43
Figura 16 Tempos médios dos diferentes métodos <i>Simulated Annealing</i> para instâncias de máquinas paralelas P1, P2 e P3 .....	44
Figura 17 Tempos médios dos diferentes métodos <i>Simulated Annealing</i> para instâncias de máquinas paralelas P1, P2 e P3 .....	44
Figura 18 Desvios médios dos diversos métodos para instâncias de máquinas simples .....	47
Figura 19 Desvios médios dos diversos métodos para instâncias de máquinas paralelas P1 e P2 .....	50
Figura 20 Desvios médios dos diversos métodos para instâncias de máquinas paralelas P4, P5 e P6 .....	50

## LISTA DE TABELAS

Tabela 1	Parâmetros para geração de instâncias com uma única máquina.....	32
Tabela 2	Parâmetros para geração de instâncias com mais de uma máquina ...	32
Tabela 3	Resultados obtidos para o PGDLPP com máquina simples em S1.....	34
Tabela 4	Resultados obtidos para o PGDLPP com máquina simples em S2.....	34
Tabela 5	Resultados obtidos para o PGDLPP com máquina simples em S3.....	35
Tabela 6	Resultados obtidos para o PGDLPP com máquina simples em S4.....	36
Tabela 7	Resultados obtidos para o PGDLPP com máquina s paralelas em P1	38
Tabela 8	Resultados obtidos para o PGDLPP com máquina s paralelas em P2	39
Tabela 9	Resultados obtidos para o PGDLPP com máquina s paralelas em P3	39
Tabela 10	Resultados obtidos para o PGDLPP com máquina s paralelas em P4 .....	40
Tabela 11	Resultados obtidos para o PGDLPP com máquina s paralelas em P5 .....	40
Tabela 12	Resultados obtidos para o PGDLPP com máquina s paralelas em P6 .....	41
Tabela 13	Resultados obtidos para o PGDLPP com os diversos métodos para S1 e S2.....	46
Tabela 14	Resultados obtidos para o PGDLPP com os diversos métodos para S3 e S4.....	46
Tabela 15	Resultados obtidos para o PGDLPP com os diversos métodos em P1 e P2.....	48
Tabela 16	Resultados obtidos para o PGDLPP com os diversos métodos em P4 e P5.....	48
Tabela 17	Resultados obtidos para o PGDLPP com os diversos métodos em P6 .....	49

## SUMÁRIO

1	INTRODUÇÃO.....	10
1.1	Contextualização .....	10
1.2	Objetivos do Trabalho.....	10
1.3	Metodologia .....	11
1.4	Estrutura do Trabalho .....	12
2	REFERENCIAL TEÓRICO.....	13
3	METODOLOGIA .....	20
3.1	Modelo matemático para o PGDLPP.....	20
3.2	O método proposto .....	23
3.2.1	Representação da solução e o método <i>Greedy-Mod</i> .....	23
3.2.2	Pseudocódigo .....	26
3.2.3	Movimentos de busca.....	30
4	RESULTADOS COMPUTACIONAIS .....	32
4.1	Definição das instâncias.....	32
4.2	Avaliação do método proposto.....	33
4.3	<i>SimulatedAnnealing</i> x Outros métodos.....	45
5	CONCLUSÕES .....	54
6	REFERÊNCIAS BIBLIOGRÁFICAS .....	55

## 1 INTRODUÇÃO

### 1.1 Contextualização

Este trabalho propõe a aplicação de uma abordagem heurística, chamada *Simulated Annealing*, para resolver o Problema Geral de Dimensionamento de Lotes e Programação da Produção (PGDLPP). Trata-se de um problema de programação da produção com a finalidade de encontrar a melhor distribuição e a melhor quantidade de produtos como forma de minimizar os custos de produção.

Heurística é a denominação para um algoritmo que fornece soluções sem uma garantia formal de sua qualidade. O *Simulated Annealing* se encaixa nesse contexto como uma busca local probabilística fundamentada numa analogia com a termodinâmica. *Annealing* é o processo físico que consiste em aquecer um sólido a altas temperaturas e depois resfriá-lo lentamente até se atingir uma estrutura molecular estável, ou seja, minimizar a energia livre do sólido. Em problemas de otimização, é possível fazer uma analogia a esse processo de forma que o algoritmo aqueça uma solução (estado físico) e do processo de resfriamento encontre uma função com custo mínimo (energia).

Os métodos heurísticos têm mostrado grande eficiência ao resolver problemas de produção, principalmente quando a complexidade do problema envolvido é elevada. Com base nisso surgiu a motivação para elaborar um trabalho com o objetivo de aplicar o método *Simulated Annealing* ao PGDLPP.

### 1.2 Objetivos do Trabalho

O principal objetivo do presente trabalho é aplicar o algoritmo *Simulated Annealing* ao Problema Geral de Dimensionamento de Lotes e Programação da Produção (PGDLPP) com máquinas paralelas e resolvê-lo em diversas instancias do

problema. Métodos de resfriamento para o algoritmo são avaliados e comparados a outros métodos utilizados para solucionar o PGDLPP.

Objetivos específicos deste trabalho:

- Estudo do Problema Geral de Dimensionamento de Lotes e Programação da Produção (PGDLPP) e avaliação de instâncias existentes na literatura;
- Avaliar o desempenho de diversos métodos de resfriamento para o algoritmo *Simulated Annealing*;
- Comparar os resultados com outros métodos heurísticos aplicados ao PGDLPP.

### 1.3 Metodologia

O trabalho aqui apresentado pode ser classificado, quanto a sua natureza, como pesquisa aplicada ou tecnológica, pois objetiva a aplicação de conhecimentos prévios para buscar melhores resultados na resolução do Problema Geral de Dimensionamento de Lotes e Programação da Produção (PGDLPP). Quanto aos objetivos, classifica-se como do tipo exploratória, uma vez que visa à descoberta de práticas que melhorem o problema estudado. Com o objetivo principal de otimizar as soluções para o problema em questão, o trabalho também se classifica como pesquisa operacional quanto ao procedimento. Trata-se de uma pesquisa em laboratório já que ocorre a possibilidade de se controlar as variáveis que possam intervir no experimento. Os testes foram realizados em um computador com processador Intel® Core™ i5 CPU 750, 2.67 GHz e 2GB de RAM. Foi utilizada a linguagem de programação Java na implementação dos algoritmos executados no sistema operacional Windows 7 Professional build 7600 64-bit.

#### **1.4 Estrutura do Trabalho**

O presente capítulo contextualiza o trabalho introduzindo o problema estudado, os objetivos a serem alcançados e a metodologia de pesquisa adotada. O Capítulo 2 apresenta o referencial teórico necessário para um melhor entendimento desta monografia. O Capítulo 3 apresenta o problema e o método heurístico abordado. O Capítulo 4 relata os resultados obtidos e uma comparação com outras heurísticas da literatura. Por fim, o Capítulo 5 apresenta as conclusões do trabalho.

## 2 REFERENCIAL TEÓRICO

### 2.1 Problemas de dimensionamento de lotes e programação da produção

Segundo Karimi *et al.* (2003), o planejamento da produção é uma atividade que consiste no melhor uso dos recursos de forma a satisfazer a produção em um determinado horizonte de tempo. O planejamento da produção envolve decisões importantes e podem ser classificadas como sendo de longo, médio e curto prazo. Nas decisões de longo prazo, objetiva-se antecipar necessidades agregadas e tomar decisões estratégicas. Decisões de médio prazo envolvem planejamentos importantes como estabelecer quantidades a serem produzidas, otimizar fatores relacionados à produção e satisfazer restrições relacionadas ao problema. No curto prazo, são tomadas decisões do dia-a-dia do processo produtivo.

Decisões de programação da produção e dimensionamento de lotes estão diretamente relacionados ao problema de identificar quando e quanto de um ou vários produtos devem ser produzidos em uma ou mais máquinas. Os problemas de produção estão relacionados com o planejamento da produção, envolvendo decisões como escalonamento de tarefas, alocação de recursos e minimização de custos, levando em consideração as demandas e prazos a serem atendidos.

Existem na literatura vários problemas que envolvem programação da produção e dimensionamento de lotes. Podemos citar alguns como *Economic Lot Scheduling Problem* (ELSP) (RAZA et al, 2008), o *Capacitated Lot Sizing Problem* (CLSP) (KARIMI et al, 2003), *Discrete Lot Sizing and Scheduling Problem* (DLSP) (FLEISCHMANN, 1990) e o *General Lot Sizing and Scheduling Problem* (GLSP) (FLEISCHMANN et al, 1997) que será o foco deste trabalho.

Ainda de acordo com Karimi (2003), diversas características podem ser relevantes na classificação de problemas de dimensionamento de lotes: horizonte de planejamento, número de produtos e capacidade disponível. O horizonte de planejamento é o tempo para o qual a produção será projetada e pode ser finito ou infinito. O problema pode possuir apenas um ou múltiplos produtos a serem

produzidos. Restrições de capacidade, que limitam a produção, podem existir sendo o problema então classificado como capacitado ou não capacitado.

A demanda dos produtos está relacionada com o que deve ser produzido ao longo do tempo, onde o horizonte de planejamento é segmentado em períodos e as demandas devem ser atendidas em cada período. Neste caso, denomina-se demanda estática aquela que é igual para todos os períodos do horizonte de planejamento, ou dinâmica caso a demanda por período seja diferente. Custos de configuração ou ajuste (*setup*) de produtos nas máquinas podem ser dependentes dos itens produzidos anteriormente ou independentes destes, ou seja, podem ser dependentes ou não da sequência de produção. Nas abordagens mais próximas de sistemas de produção reais, os custos de *setup* são dependentes da sequência de produção.

O número de máquinas disponíveis para a produção e atendimento das demandas pode ser maior que um, tornando o problema mais complexo, uma vez que há maiores possibilidades de organização da produção.

Conforme descrito em Toledo (2008), os problemas de dimensionamento de lotes, apesar de estarem sujeitos a variações, mantém algumas características comuns como: horizonte de planejamento dividido em períodos e demandas que podem variar de um período para o outro. Esses problemas têm como objetivo minimizar custos tanto de configuração da máquina quanto de estoque do produto. A demanda existente deve ser atendida, ou seja, deve ser produzida uma quantidade que, se somada ao estoque existente para determinado produto, seja maior ou igual à demanda estabelecida naquele período. Restrições de capacidade podem levar à produção antecipada e à estocagem, visto que a capacidade de um período pode não comportar a produção de toda a demanda do produto neste período. Problemas de dimensionamento de lotes em máquinas paralelas, embora não muito citados na literatura, apresentam considerável complexidade. O número de possibilidades de planejamento aumenta de acordo com o número de máquinas presentes.

A maioria dos problemas de dimensionamento de lotes e programação da produção é difícil de ser resolvido já que um problema desse tipo com restrição de

capacidade é NP-Difícil segundo Bitran e Yanasse (1982). Problema de dimensionamento de lotes e programação da produção com vários itens (multi-item) também foi provado ser NP-Difícil por Chen e Thizy (1990). Para resolvê-los, vários métodos foram desenvolvidos, tanto métodos exatos quanto métodos chamados heurísticos ou metaheurísticos.

## 2.2 Métodos exatos e métodos heurísticos

Os métodos exatos são aqueles que, sob as mesmas condições, resultam sempre na mesma solução. Um exemplo de método exato, proposto por Land e Doig (1960), é o *Branch and Bound* (BB) que enumera sistematicamente todas as soluções candidatas, onde várias outras soluções são descartadas devido ao uso de limitantes superiores e inferiores. O uso desse método pode ser visto em Fleischmann (1990) que utiliza BB junto com a Relaxação Lagrangiana para resolver o DLSP. Barany *et al.* (1984) e Leung *et al.* (1989) utilizam BB junto com planos de corte para resolver o CLSP.

Todavia, métodos exatos não costumam ser utilizados em problemas de dimensionamento de lotes e programação da produção, pois, devido a complexidade do problema, não retornam uma solução factível dentro de um tempo razoável. De acordo com Cormen *et al.* (2002), alguns problemas nem mesmo possuem algoritmos exatos e, dentre os que possuem, o tempo computacional pode tornar inviável a utilização do método.

Segundo Bueno (2009), métodos heurísticos são algoritmos exploratórios que buscam resolver problemas. Geralmente não envolvem a implementação computacional de um conhecimento especializado. Por este motivo, muitas vezes, esses métodos são classificados como “busca cega”. Tendo como ponto de partida uma solução viável, baseiam-se em sucessivas aproximações direcionadas a um ponto ótimo. Logo, estes métodos costumam encontrar as melhores soluções possíveis para o problema e não soluções exatas. Segundo Michalewicz e Fogel (2004), heurísticas

podem ser baseadas em observações da natureza, como analogias, em regras básicas ou então pelo senso comum. Para Jans e Degraeve (2007), a principal razão para o uso de heurísticas é a flexibilidade e capacidade para lidar com problemas complexos.

Em Fraga (2006) pode-se encontrar uma divisão dos métodos heurísticos em heurísticas construtivas, heurísticas de refinamento e metaheurísticas. Há ainda as heurísticas híbridas que combinam duas ou mais heurísticas com a finalidade de ampliar o efeito na busca de melhores soluções.

Uma heurística construtiva gera uma solução de forma incremental, ou seja, elemento a elemento, de modo que a cada iteração, um novo elemento é escolhido para fazer parte da solução. A escolha do elemento varia de acordo com a função de avaliação para cada problema abordado. Destaca-se entre elas, a forma gulosa onde o elemento melhor classificado, de acordo com a função de avaliação, é selecionado. As soluções geradas por um algoritmo de construção gulosa geralmente são melhores que as obtidas de forma aleatória. (STUTZLE e DORIGO, 2000).

Heurísticas de refinamento partem de uma solução inicial e tentam encontrar uma solução melhor em sua vizinhança. Ao encontrar, a solução corrente é substituída pela nova e a busca continua até que não exista mais uma solução melhor na vizinhança da solução corrente, ou até atingir algum critério de parada (FRAGA, 2006). Este tipo de heurística está fortemente relacionado com a escolha de uma estrutura de vizinhança para seu bom desempenho. No entanto, os problemas comuns dessas heurísticas estão em convergir para ótimos locais e ficarem presas nessas soluções.

A palavra metaheurística vem da junção da palavra grega "*heurisken*" que significa procurar, descobrir, com o sufixo "meta" que significa além, mais longe (BLUM e ROLI, 2003). Uma boa definição para metaheurística é dada por Ribeiro (2002), "As metaheurísticas são procedimentos destinados a encontrar uma boa solução, eventualmente a ótima, consistindo na aplicação, em cada passo, de uma heurística subordinada, a qual tem que ser modelada para cada problema específico". As metaheurísticas se destacam das heurísticas pelo fato de serem capazes de escapar de

ótimos locais. Isso permite que sejam aplicadas a diversos tipos de problemas. Exemplos de metaheurísticas são Busca Tabu, *Simulated Annealing*, Algoritmos Genéticos, Colônia de Formigas, entre outras.

### 2.3 Exemplos de métodos heurísticos

Fleischmann e Meyr (1997) propõem um modelo para a resolução do PGDLPP utilizando a heurística baseada em buscas locais denominada *Threshold Accepting*. Em Meyr (2002) estende-se o PGDLPP com tempo de preparação para o contexto de máquinas paralelas, utilizando *Simulated Annealing* e *Threshold Accepting*.

*Simulated Annealing* (SA) é uma técnica descrita por Kirkpatrick (KIRKPATRICK, 1983) originada de conceitos da mecânica estatística no qual ele aplica o algoritmo de Metropolis (METROPOLIS et al, 1953) para a resolução de problemas de otimização. O algoritmo e os resultados obtidos podem ser encontrados em Kirkpatrick (1983).

Segundo Izquierdo (2000), *annealing* consiste no processo térmico de aquecimento de um material a uma alta temperatura, até torná-lo líquido, seguido de uma lenta e gradativa diminuição de sua temperatura, podendo assim atingir um estado sólido mais estável. A cada configuração atingida, o material apresenta um nível de energia. O processo termina quando o ponto de solidificação é atingido, apresentando uma configuração de energia mínima. Em otimização combinatória, pode-se definir um processo similar onde é possível encontrar, dentre um espaço de soluções, uma configuração com custo minimizado. Os algoritmos desenvolvidos utilizando este mesmo conceito são denominados “*Simulated Annealing*” ou Recozimento Simulado.

Desta forma, *Simulated Annealing* é um método iterativo que combina técnicas de busca local e randomização que, quando aplicado a um problema de otimização, procura evitar a parada prematura em um ótimo local. Isto é possível, pois, após o aquecimento, soluções ruins também são aceitas seguindo uma função de probabilidade de acordo com a temperatura atual.

Busca Tabu é uma metaheurística que tem sido aplicada com sucesso em diversos problemas de otimização combinatória. Este método usa exploração reativa e memória flexível para guiar um procedimento de busca em vizinhança no processo de solução. Através da exploração reativa, determina-se uma direção de busca com base nas propriedades da solução corrente e do histórico da busca (GLOVER, 1989, 1990, GLOVER e LAGUNA, 1997). A memória flexível é formada por estruturas de memória de curto e longo prazo que armazenam o histórico da busca. A memória de curto prazo armazena atributos de soluções visitadas anteriormente em um curto espaço de tempo, ou seja, mais recentes. Estes atributos são armazenados em uma lista tabu para impedir o retorno a soluções visitadas. A memória de longo prazo contém um histórico amplo de soluções e seus respectivos atributos encontrados durante o processo de busca. É utilizada em estratégias de diversificação e intensificação da busca.

Para Zorzalet al (2006) A palavra “tabu” da expressão Busca Tabu não se refere a qualquer tipo de consideração de natureza sobrenatural ou moral. Ela se refere às restrições colocadas para guiar o processo de busca e que operam em vários aspectos como, por exemplo, excluir alternativas de busca classificadas como “proibidas”, transladar avaliações modificadas e avaliar probabilidades de seleção. Essas restrições são criadas pela referência a estruturas de memória que são desenvolvidas para este propósito específico.

Em 1975, J. Holland introduziu os algoritmos genéticos (HOLLAND, 1992) com o objetivo de desenvolver sistemas artificiais que se comportem de forma semelhante ao processo evolutivo das espécies (CONCILIO, 2000). Inspirados na teoria da evolução e nos processos de seleção natural, de acordo com Filho (2000) citado por Neto et al (2007), os algoritmos genéticos (AG) são algoritmos de busca e otimização que consideram cada indivíduo como sendo candidato a integrante do resultado ótimo. Os AGs diferem dos métodos determinísticos por utilizarem meios probabilísticos para convergência do resultado, trabalham com parâmetros codificados e avaliam cada indivíduo isoladamente através de um paralelismo implícito

(MARQUES NETO et al, 2007). A cada geração, descartam-se os piores indivíduos e os melhores permanecem transmitindo suas qualidades. O algoritmo utiliza o conhecimento adquirido anteriormente para formar uma nova população mais próxima do resultado esperado (MITCHELL, 1998).

Em Toledo et al (2009) é apresentada uma codificação do Algoritmo Genético para resolver o PGDLPP que, ao final, é comparado com outras metaheurísticas obtendo os melhores resultados. O AG, a codificação e os resultados podem ser vistos em Toledo et al (2009).

## **2.4 Conclusão**

Neste capítulo, foram expostos os conceitos sobre problemas de dimensionamento de lotes e programação da produção, assim como definições de métodos exatos e métodos heurísticos. Essa base teórica é fundamental para o melhor entendimento dos capítulos subsequentes.

### 3 METODOLOGIA

#### 3.1 Modelo matemático para o PGDLPP

Toledo et al (2009) apresentaram o modelo matemático para o PGDLPP considerando máquinas paralelas e penalização das demandas não atendidas. Esse modelo será detalhado nesta seção. A formulação proposta segue a modelagem descrita por Meyr (2002) para o PGDLPP com máquinas paralelas. A principal diferença está na inserção de variáveis que acumulam as demandas não atendidas para cada produto. Essas variáveis são incluídas na equação de balanço de estoque no primeiro período de produção e também são penalizadas na função objetivo do modelo.

O problema é modelado dividindo o horizonte de planejamento em  $T$  macro-períodos. Por sua vez, cada macro-período  $t$  possui um número fixo de micro-períodos  $S$ . Um aspecto interessante neste tipo de modelagem é que o tamanho de cada micro-período  $s$  varia de forma proporcional ao tamanho do lote do produto a ele atribuído. Assim, as variáveis de dimensionamento dos lotes e de atribuição de produtos às linhas e períodos estão indexadas por produtos e micro-períodos. O modelo assume que um único produto é atribuído e produzido em cada micro-período.

A Figura 1 apresenta um exemplo dessa situação, supondo três produtos e três micro-períodos em cada macro-período em uma linha L1. O tamanho dos micro-períodos  $s_1, s_2, s_3, s_4, s_5$  e  $s_6$  variam de forma proporcional à quantidade produzida. Por exemplo, o lote em  $s_2$  de P2 é maior que o lote de P2 em  $s_4$  na Figura 1. Assim, os micro-períodos  $s_1, s_4$  e  $s_6$  ocupam 1 unidade de tempo,  $s_2$  ocupa 2 unidades e  $s_5$  ocupa 3 unidades de tempo.

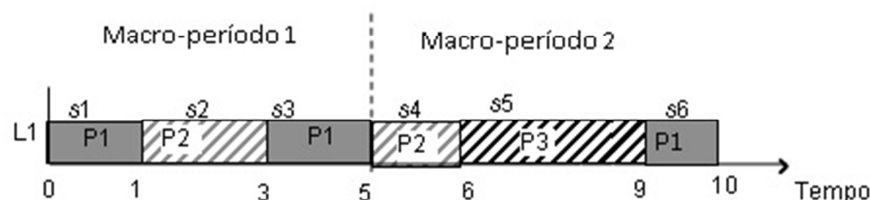


Figura 1 - Variação dos micro-períodos (Toledo et al, 2009)

Um total de  $J$  produtos e  $L$  linhas são considerados. Abaixo são listados os demais parâmetros do problema:

- $C_t$ : Capacidade em unidades de tempo disponível no macro-período  $t$ .
- $TP_{lj}$ : Tempo de processamento do produto  $j$  na linha  $l$ .
- $Min_j$ : Lote mínimo do produto  $j$ .
- $H_j$ : Custo de estoque do produto  $j$ .
- $CT_{ij}$ : Custo de troca do produto  $i$  para o produto  $j$ .
- $D_{it}$ : Demanda do produto  $i$  no macro-período  $t$ .
- $I_{j0}$ : Estoque inicial do produto  $j$  no início do horizonte de tempo.
- $Y_{lj,0}$ : 1, se o produto  $j$  está ajustado inicialmente para a linha  $l$ ; 0 caso contrário.
- $M$ : Penalização por unidade de demanda não atendida.

As variáveis do modelo são apresentadas a seguir:

- $I_{jt} \geq 0$ : Estoque do produto  $j$  ao final do macro-período  $t$ .
- $q_{ljs} \geq 0$ : Quantidade do produto  $j$  produzido no micro-período  $s$  da linha  $l$ .

- $q_{0j} \geq 0$ : Quantidade de demanda do produto  $j$  que não foi produzida.
- $y_{l,j,s}$ : 1 Se o produto  $j$  é atribuído à linha  $l$  no micro-período  $s$ ; 0 caso contrário.
- $z_{l,i,j,s} \geq 0$ :  $z_{l,i,j,s} = 1$  se há troca do produto  $i$  para  $j$  no micro-período  $s$  da linha  $l$ ;  $z_{l,i,j,s} = 0$ , caso contrário.

Modelo matemático para o PGLDPP com máquinas paralelas e penalização de demandas:

$$\text{Minimize } \sum_{j=1}^J \sum_{t=1}^T H_j l_{jt} + \sum_{i=1}^J \sum_{j=1}^J \sum_{l=1}^L \sum_{s=1}^{T.S} CT_{ij} Z_{l,i,j,s} + M \sum_{j=1}^J q_j^0 \quad (1)$$

$$l_{j1} = l_{j,0} + q_j^0 + \sum_{l=1}^L \sum_{s=1}^S q_{ljs} - D_{j1} \quad j = 1, \dots, J \quad (2a)$$

$$l_{jt} = l_{j,t-1} + \sum_{l=1}^L \sum_{s=(t-1)S+1}^{T.S} q_{ljs} + q_j^0 - D_{j1} \quad j = 1, \dots, J, t = 1, \dots, T \quad (2b)$$

$$\sum_{j=1}^J \sum_{s=(t-1)S+1}^{T.S} TP_{lj} q_{ljs} \leq C \quad l = 1, \dots, L, t = 1, \dots, T \quad (3)$$

$$q_{ljs} \leq \frac{C_t}{TP_{lj}} y_{ljs} \quad l = 1, \dots, L, j = 1, \dots, J, s = 1, \dots, T.S \quad (4)$$

$$q_{ljs} \geq \text{Min}_j (y_{ljs} - y_{l,j,s-1}) \quad l = 1, \dots, L, j = 1, \dots, J, s = 1, \dots, T.S \quad (5)$$

$$\sum_{j=1}^J y_{ljs} = 1 \quad l = 1, \dots, L, s = 1, \dots, T.S \quad (6)$$

$$z_{l,i,j,s} \geq y_{l,i,s-1} + y_{l,j,s} - 1 \quad l = 1, \dots, L, i = 1, \dots, J, j = 1, \dots, J, s = 1, \dots, T.S \quad (7)$$

A função objetivo minimiza os custos de estoque e os custos de troca envolvendo produtos. Os custos de troca foram considerados como dependentes da sequência dos produtos, mas independentes das linhas. As demandas não atendidas são acumuladas nas variáveis  $q_j^0$  e penalizadas na função objetivo. O estoque de cada macro-período é determinado pelas equações (2a) e (2b). Observe que a demanda não atendida é acumulada na equação de balanço de estoque do primeiro período (2a). Isso permite que as demandas não atendidas nos demais períodos continuem satisfazendo a

restrição de balanço de estoque já que são acumuladas em  $q_j^0$ . Considera-se que essas demandas foram produzidas em um período  $t=0$  com capacidade ilimitada. A restrição (3) garante que a capacidade disponível dentro do período  $t$  não seja violada. A capacidade disponível por macro-período é a mesma para todas as linhas, mas cada linha pode ter diferentes tempos de processamento por produto ( $TP_{1,j}$ ). O tamanho máximo do lote de um produto em um micro-período é dado pela restrição (4). Essa restrição também impõe que nada seja produzido, caso não ocorra atribuição do produto ao micro-período ( $y_{1,j,s} = 0$ ). A restrição (5) assegura que um lote mínimo será produzido quando ocorre efetivamente a atribuição de um novo produto a um micro-período ( $y_{1,j,s} = 1$ ). Por outro lado, se o mesmo produto é atribuído a dois micro-períodos consecutivos, não necessariamente haverá produção na segunda atribuição. A equação (6) permite que somente um produto seja produzido em um micro-período. As trocas de produtos são determinadas pela restrição (7).

## 3.2 O método proposto

### 3.2.1 Representação da solução e o método *Greedy-Mod*

Uma possível solução para o problema apresentado é dada por linhas na forma de matriz. Essa matriz contém informações sobre a sequência de produtos do tipo  $L \times T$ , onde cada entrada  $(l \times t)$ , com  $l \in L$  e  $t \in T$ , representa uma sequência de produtos a serem produzidos. A representação dessa matriz pode ser vista na Figura 2.

	T1			T2		
L1	P3	P1	P2	P2	P3	
L2	P2			P3	P1	

Figura 2 - Matriz de sequências

As sequências de produtos possuem duas restrições nas quais duas entradas consecutivas não podem ser ocupadas pelo mesmo produto, com exceção para a última entrada de um macro-período e a primeira entrada do próximo macro-período, e

também não pode haver uma entrada sem produto entre duas entradas ocupadas em um mesmo macro-período. Essa representação foi adotada por Fleischmann e Meyr (1997) para o PGDLPP com uma única linha e estendida para múltiplas linhas para este trabalho.

Uma vez conhecida a sequência de produção e, conseqüentemente os custos de troca da solução, é necessário estabelecer as quantidades a serem produzidas. Para isso utiliza-se um método determinístico guloso chamado *Greedy-Mod* o qual será executado pelo método *Simulated Annealing*. O método aqui utilizado é uma adaptação para o PGDLPP com máquinas paralelas do método *Greedy-Mod* proposto por Fleischmann e Meyr (1997) para o PGDLPP com máquinas simples. O algoritmo *Greedy-Mod* percorre todos os períodos, seguindo do último para o primeiro, atribuindo inicialmente o lote mínimo dos produtos. Em seguida, aloca o restante da capacidade disponível nas máquinas, em cada período, aos respectivos lotes de produtos com demanda a ser satisfeita. Isso é feito selecionando os produtos pela ordem decrescente dos valores  $H_j/TP_{lj}$  (custo de estoque/tempo de processamento). Ao final, o algoritmo realiza uma etapa chamada de desagregação na qual, para cada período  $t$ , a quantidade total de um produto a ser produzida é dividida igualmente e alocada para os lotes referentes a este produto em  $t$ .

A Figura 3 apresenta o pseudocódigo para o algoritmo *Greedy-Mod* e os parâmetros utilizados estão listados abaixo:

- $X_{ljt}$  - armazena o lote do produto  $j$  produzido na linha  $l$  no macro-período  $t$ .
- $Y_{ljt}$  - número de lotes do produto  $j$  no macro-período  $t$  na linha  $l$ .
- $D_j$  - demanda acumulada (total) do produto  $j$ .
- $AC_{jt}$  - demanda acumulada do produto  $j$  até o macro anterior ao  $t$ .

```

Greedy-Mod
t=T
Enquanto (t≥1) faça
  //Garantindo lote mínimo
  Para toda linha l
    Para todo produto j com (Yljt>0)
      p = mínimo(Minj * Yljt ; Dj - ACjt); //Calcula a parcela a ser produzida
      Atualiza(Xljt, Klt, Dj,p);
      Se Klt< 0 então
        Penalizar capacidade violada
    fim do Para
  fim do Para
  //Evitando penalização de demandas não atendidas
  Para todo produto j
    Q = Dj - LSjt;
    Para toda linha l com (Yljt>0 e Q > 0)
      p = Mínimo(Dj - ACjt ; Klt/TPlj ; Q);
      Atualiza(Xljt, Klt, Dj,p);
      Q = Q - p;
    fim do Para
  fim do Para
  //Distribui demandas em uma ordem gulosa
  Para toda linha l e produto j com (Yljt> 0) em ordem decrescente de Hj/TPlj
    p = Mínimo(Dj - ACjt ; Klt/TPlj);
    Atualiza(Xljt, Klt, Dj,p);
  fim do Para
  t = t-1
fimdo Enquanto
qj0 = Dj, para todo j.
//Distribui as demandas dos macro-período para os micro-períodos
Para todo l,j,t e s pertencente a St
  Se Yljt> 0 então
    Xljs = Xljt / Yljt;
  Senão
    Xljs = 0;
fim do Para
fim do método

```

Figura 3 - Algoritmo *Greedy-Mod*

- $LS_{jt}$  - limite superior estimado considerando que toda a capacidade disponível fosse utilizada para produzir o produto  $j$  em períodos anteriores ao  $t$ .

$$\begin{cases} LS_{jt} = 0; & \text{para } t = 1 \\ LS_{jt} = LS_{j(t-1)} + \sum_{l=1}^L \frac{\text{Minimo}(1, Y_{ljt}) * K_{lt}}{TP_{lj}}; & \text{para } t > 1 \end{cases} \quad (8)$$

- $X_{ljs}$  – lote do produto  $j$  produzido na linha  $l$  no micro-período  $s$ .
- $Q_j$  – quantia de demanda do produto  $j$  não alocada pelo método de decodificação.

A figura 4 abaixo demonstra como são atualizadas as quantias  $X_{ljs}$ ,  $K_{lt}$  e  $D_j$  quando um lote  $p$  é alocado.

**Atualiza**( $X_{ljs}$ ,  $K_{lt}$ ,  $D_j$ ,  $p$ )

$X_{ljs} = X_{ljs} + p$ ; //Adiciona à parcela, a quantia a ser produzida

$K_{lt} = K_{lt} - p * TP_{lj}$ ; //Subtrai a capacidade para produzir a quantia  $p$

$D_j = D_j - p$ ; //Nova demanda do produto que deverá ser atendida.

**Fim Atualiza**

Figura 4 – Método para atualizar  $X_{ljs}$ ,  $K_{lt}$  e  $D_j$

### 3.2.2 Pseudocódigo

O método heurístico *Simulated Annealing* foi adaptado para o PGDLPP e o pseudocódigo do método é descrito na Figura 5. O método inicia uma temperatura  $T$ , uma temperatura  $T_0$  elevada e uma solução  $S$ . A partir dessa solução será gerada outra solução  $S'$ . Essa nova solução pertence à vizinhança de  $S$ , sendo obtida aplicando-se movimentos locais em  $S$ . Esses movimentos são do tipo inserir lote em um período, deletar lote, trocar lotes entre períodos diferente, etc. Após a escolha de um vizinho, as quantidades a serem produzidas são calculados em relação a nova sequência de produção resultante, isto é feito chamando-se o método *Greedy-Mod*. O vizinho é

então avaliado e, caso seja melhor que a melhor solução encontrada até então, a melhor solução é atualizada com esse novo vizinho.

Há uma probabilidade de escolha de uma solução pior para escapar de ótimos locais em função da temperatura. Nesta função de probabilidade, toma-se um valor aleatório entre 0 e 1. Caso este valor seja menor que  $e^{-\Delta/T}$ , com  $\Delta$  sendo a diferença entre o valor da solução vizinha  $S'$  e o valor da solução corrente  $S$ , uma solução piorada é aceita.

Esse processo é repetido um determinado número de vezes para cada temperatura, o que equivale ao número de iterações necessário para se atingir o equilíbrio térmico em uma dada temperatura. Nestas iterações, a temperatura é reduzida gradativamente e o processo é reaquecido, caso seu valor chegue a um limiar fixado em 0,001.

Os parâmetros de controle são a razão de resfriamento  $\alpha$  (utilizada na expressão de resfriamento (9)) o número de iterações máxima para cada temperatura SAMax e a temperatura inicial  $T_0$ .

$$T_k = \alpha * T_{k-1} \quad (9)$$

Nos testes realizados também foram usadas duas outras expressões de resfriamento, sendo elas:

$$T_k = \frac{T_{k-1}}{1 + \gamma \sqrt{T_{k-1}}}, \forall k \geq 1 \quad (10)$$

onde  $T_k$  representa a temperatura na iteração  $k$  e  $\gamma$  uma constante tal que  $0 < \gamma < 1$ .

Outra forma de resfriamento utilizada é:

$$T_k = \begin{cases} \beta T_{k-1} & \text{se } k = 1 \\ \frac{T_{k-1}}{1 + \gamma T_{k-1}} & \text{se } k > 1 \end{cases} \quad (11)$$

onde  $\gamma = \frac{T_0 - T_{k-1}}{(k-1)T_0 T_{k-1}}$ , sendo  $T_0$  a temperatura inicial,  $T_k$  a temperatura na iteração  $k$  e  $\beta$  um parâmetro para corrigir a imperfeição do resfriamento ( $0 < \beta < 1$ ).

```

SimulatedAnnealing ()
  SAMax ← 40
  α ← 0.95
  Inicialize uma solução S
  S* ← S
  T0 ← TemperaturaInicial(S, θ, μ, SAMax, 1000)
  T ← T0
  greedy-mod(S), f(S)
  f(S*) ← f(S)
  Enquanto (tempoTotal ainda não atingido)
    Enquanto(número de iterações < SAMax)
      Gere um vizinho S' ∈ N(S)
      greedy-mod(S')
      Δ ← f(S') - f(S)
      Se (Δ < 0)
        S ← S'
        Se (f(S) < f(S*))
          S* ← S
      Senão
        Tome x ∈ [0, 1]
        Se (x < e-Δ/T)
          S ← S'
      Fim se
    Fim Enquanto
  T ← T * α
  Se (T < 0.001)
    T ← T0
  Fim Enquanto
Fim SimulatedAnnealing

```

Figura 5 - Algoritmo *Simulated Annealing*

Todos os três métodos com as expressões de resfriamento relacionadas acima operam de forma adaptativa na determinação da temperatura inicial realizando uma simulação do *Simulated Annealing*. Essa simulação é realizada pelo método *TempereturaInicial()* no pseudocódigo da Figura 5.

Um pseudocódigo para o referido método é apresentado na Figura 6. O método parte de uma solução S e de uma temperatura qualquer definida pelo usuário. Nesta

etapa inicial, são contados quantos vizinho dessa solução são aceitos em SAMax iterações do método com a dada temperatura.

Se o número de vizinhos aceitos for alto, definido como 95% dos vizinhos, então a temperatura corrente é retornada como sendo a temperatura inicial para o método principal *Simulated Annealing*. Caso o número de vizinhos aceitos não atinja o valor mínimo determinado, a temperatura aumenta por uma taxa de 10% da temperatura corrente. Esse processo segue até que se obtenha o número de vizinhos aceitos desejados para uma determinada temperatura.

A figura 5 ilustra o algoritmo deste processo de ajuste da temperatura inicial, onde  $\theta$  é a taxa de aumento da temperatura,  $\mu$  é o taxa de vizinhos aceitos e  $T_0$  é a temperatura inicial para o método.

```

TemperaturaInicial (S,  $\theta$ ,  $\mu$ , SAMax,  $T_0$ )
  T  $\leftarrow$   $T_0$ 
  Continua  $\leftarrow$  TRUE
  Enquanto (Continua)
    Aceitos  $\leftarrow$  0
    Enquanto(número de iterações < SAMax)
      Gere um vizinho  $S' \in N(S)$ 
       $\Delta \leftarrow f(S') - f(S)$ 
      Se ( $\Delta < 0$ )
        Aceitos  $\leftarrow$  Aceitos + 1
      Senão
        Tome  $x \in [0, 1]$ 
        Se ( $x < e^{-\Delta/T}$ )
          Aceitos  $\leftarrow$  Aceitos + 1
        Fim se
      Fim Enquanto
    Fim Enquanto
    Se (Aceitos >  $\mu * SAMax$ )
      Continua  $\leftarrow$  FALSE
    Senão
      T  $\leftarrow$  T *  $\theta$ 
    Fim se
  Fim Enquanto
Fim TemperaturaInicial

```

Figura 6 - Algoritmo de ajuste adaptativo da Temperatura Inicial

### 3.2.3 Movimentos de busca

Os movimentos de busca na vizinhança de uma solução são classificados como movimentos de sequência no qual alteram a ordem dos produtos na matriz de sequência das linhas. Os movimentos propostos são listados a seguir, onde os parâmetros  $l$ ,  $t$ ,  $p$  e  $s$  são, respectivamente, linha, macro-período, produto e micro-período:

1. **Inserção de produto:** Insere um novo produto  $p$  em um micro-períodos selecionados aleatoriamente;
2. **Remoção de produto:** Remove um produto  $p$  de um micro-períodos selecionados aleatoriamente;
3. **Troca de produto:** Troca dois produtos  $p$  e  $p'$  localizados em dois micro-períodos  $s$  e  $s'$  selecionados aleatoriamente, onde:
  - a.  $s$  e  $s'$  podem estar na mesma linha e macro-período;
  - b.  $s$  e  $s'$  podem estar em um mesmo macro-período, porém em linhas diferentes;
  - c.  $s$  e  $s'$  podem estar em linhas e macro-períodos diferentes.

As figuras a seguir exemplificam os movimentos de inserção, remoção e troca de produtos respectivamente.

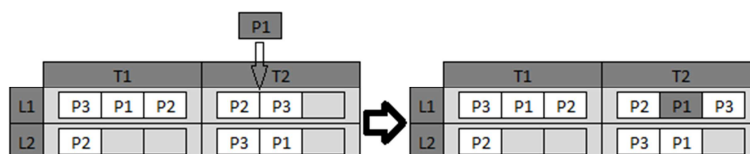


Figura 7 - Inserção de produto

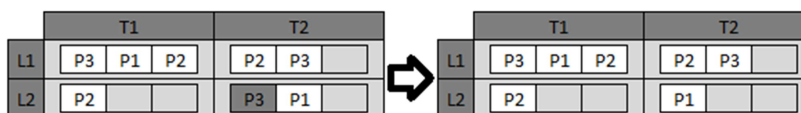


Figura 8 - Remoção de produto

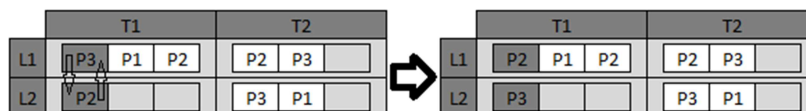


Figura 9 - Troca de produto

Após a realização de algum dos movimentos de sequência citados acima, pode haver duas entradas consecutivas, em um mesmo macro-período, preenchidas com o mesmo produto. Isso não pode ocorrer, então é realizada uma operação de reparo conforme ilustrado na figura 10. O reparo elimina a duplicação do produto na representação da solução obtida.

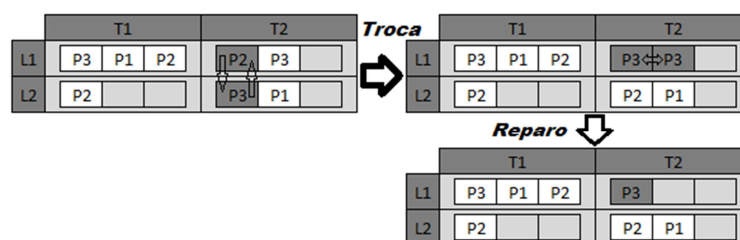


Figura 10 - Operação de reparo da sequência

### 3.3 Conclusão

Neste capítulo, foi apresentado o modelo matemático para o PGDLPP envolvendo máquinas paralelas e o pseudocódigo do método *Simulated Annealing*. Para se entender como o método heurístico foi aplicado ao problema, foi demonstrado a representação da solução utilizada, o uso do *Greedy-Mod* para distribuição das quantidades a serem produzidas e como são feitos os movimentos de busca.

## 4 RESULTADOS COMPUTACIONAIS

### 4.1 Definição das instâncias

As instâncias utilizadas para a avaliação de desempenho do método proposto estão divididas em dois grupos: instâncias com máquinas simples, ou seja, uma única máquina e instâncias com máquinas paralelas, isto é, mais de uma máquina.

O conjunto de instâncias com máquinas simples contém quatro subconjuntos denominados S1, S2, S3 e S4, para os quais foram gerados um total de 10 instâncias em cada subconjunto. Os parâmetros utilizados na geração das instâncias foram os mesmos apresentados por Haase (1996) e podem ser vistos na Tabela 1, onde U(%) é a porcentagem de utilização da capacidade disponível.

Tabela 1 - Parâmetros para geração de instâncias com uma única máquina

	Máquinas	Períodos	Micro-períodos	Produtos	Capacidade	U(%)
<b>S1</b>	1	6	4	4	200	80
<b>S2</b>	1	6	4	4	200	90
<b>S3</b>	1	5	4	4	200	90
<b>S4</b>	1	5	5	5	250	90

Já o conjunto de instâncias com máquinas paralelas é dividido em seis subconjuntos, chamados de P1, P2, P3, P4, P5 e P6, contendo cinco instâncias cada um deles. A tabela 2 apresenta os parâmetros utilizados para a geração destas instâncias.

Tabela 2 - Parâmetros para a geração de instâncias com mais de uma máquina.

	Máquinas	Períodos	Micro-períodos	Produtos	Capacidade	U(%)
<b>P1</b>	2	6	4	4	200	80
<b>P2</b>	3	6	4	4	200	80
<b>P3</b>	4	6	4	4	200	80
<b>P4</b>	2	6	8	8	400	80
<b>P5</b>	3	6	8	8	400	80
<b>P6</b>	4	6	8	8	400	80

Todas as instâncias foram codificadas usando a linguagem de modelagem matemática AMPL e solucionadas através de um método exato disponível no pacote de *solvers* CPLEX (*branch & CUT*) com tempo de execução limitado em 1 hora.

#### 4.2 Avaliação do método proposto

O método *Simulated Annealing* foi executado dez vezes para cada uma das instâncias com um tempo de execução limitado a de 30 minutos, ou seja, metade do tempo estabelecido ao método exato. Estas execuções foram realizadas para três variações do método:

- SA1: método *Simulated Annealing* com expressão de resfriamento utilizando a equação (9);
- SA2: método *Simulated Annealing* com expressão de resfriamento utilizando a equação (10);
- SA3: método *Simulated Annealing* com expressão de resfriamento utilizando a equação (11).

Ao término das dez execuções realizadas para cada instância, foi calculada a média dos custos das soluções encontradas. Esta média foi utilizada para a avaliação e comparação dos métodos propostos.

Os resultados para instâncias com apenas uma máquina são apresentados nas tabelas 3, 4, 5 e 6. As tabelas apresentam a média das soluções encontradas em determinada instância. Abaixo de cada célula é mostrado o valor do desvio em relação à solução obtida pelo AMPL. Esse desvio é calculado conforme a seguinte equação:

$$Desvio_{AMPL} = \frac{100 (solução_{Analisada} - solução_{MetodoExato})}{solução_{MetodoExato}} \quad (12)$$

Assim, quanto menor o desvio, melhor o desempenho do algoritmo analisado em relação ao método exato. Desvios iguais a zero significam que o valor encontrado foi igual ao resultado obtido pelo método exato. Desvios negativos significam que o

método analisado atingiu soluções melhores que o método exato. Os resultados, encontrados pelo método, exato marcados com \* significam que foi encontrado o valor ótimo para determinada instância, caso contrario, o valor exibido é a melhor solução encontrada ao final do tempo de execução dado ao método.

As tabelas também apresentam o tempo, considerando a média das 10 execuções, gasto pelos métodos propostos para encontrar sua melhor solução.

Tabela 3 - Resultados médios obtidos para o PGDLPP com máquina simples em S1

Instancias	AMPL	SA1		SA2		SA3	
		Fitness	Tempo	Fitness	Tempo	Fitness	Tempo
S1-0	1658*	1658	0,66	1658	4,38	1658	7,42
		0		0		0	
S1-1	1421*	1421	0,69	1421	1,47	1421	0,63
		0		0		0	
S1-2	1700*	1703	1,33	1703	16,45	1703	3,60
		0,17		0,17		0,17	
S1-3	1459*	1471	0,84	1471	1,29	1471	1,43
		0,82		0,82		0,82	
S1-4	1543*	1543	1,05	1543	6,95	1543	3,40
		0		0		0	
S1-5	1402*	1402	0,67	1402	10,78	1402	7,77
		0		0		0	
S1-6	1477*	1480	1,49	1480	20,52	1480	6,88
		0,20		0,20		0,20	
S1-7	1463*	1463	0,82	1463	3,88	1463	1,16
		0		0		0	
S1-8	1480*	1487	1,11	1487	8,15	1487	6,68
		0,47		0,47		0,47	
S1-9	1626*	1628	7,85	1628	43,39	1628	9,47
		0,12		0,12		0,12	

Tabela 4 - Resultados médios obtidos para o PGDLPP com máquina simples em S2

Instancias	AMPL	SA1		SA2		SA3	
		Fitness	Tempo	Fitness	Tempo	Fitness	Tempo
S2-0	1729*	1729	3,06	1729	12,71	1729	4,62
		0		0		0	
S2-1	1758*	1758	0,83	1758	1,87	1758	0,93

		0		0		0	
<b>S2-2</b>	2162*	2162	1,87	2162	15,05	2162	6,87
		0		0		0	
<b>S2-3</b>	1747*	1747	2,91	1747	32,34	1747	5,16
		0		0		0	
<b>S2-4</b>	1638*	1638	0,26	1638	2,95	1638	0,17
		0		0		0	
<b>S2-5</b>	1534*	1539	0,75	1539	6,31	1539	0,94
		0,32		0,32		0,32	
<b>S2-6</b>	1533*	1533	0,69	1533	3,04	1533	0,54
		0		0		0	
<b>S2-7</b>	1822*	1822	1,11	1822	16,86	1822	4,09
		0		0		0	
<b>S2-8</b>	1863*	1863	0,41	1863	2,09	1863	1,34
		0		0		0	
<b>S2-9</b>	1751*	1751	0,73	1751	5,07	1751	0,81
		0		0		0	

Tabela 5 - Resultados médios obtidos para o PGDLPP com máquina simples em S3

Instancias	AMPL	SA1		SA2		SA3	
		Fitness	Tempo	Fitness	Tempo	Fitness	Tempo
<b>S3-0</b>	1269*	1279	0,45	1279	4,88	1279	1,99
		0,78		0,78		0,78	
<b>S3-1</b>	1491*	1491	0,67	1491	3,95	1491	0,32
		0		0		0	
<b>S3-2</b>	1338*	1338	0,24	1338	3,17	1338	0,23
		0		0		0	
<b>S3-3</b>	1530*	1530	1,50	1530	16,64	1530	0,57
		0		0		0	
<b>S3-4</b>	1663*	1663	1,15	1663	10,47	1663	1,50
		0		0		0	
<b>S3-5</b>	1542*	1542	0,14	1542	0,78	1542	0,31
		0		0		0	
<b>S3-6</b>	1553*	1553	1,18	1553	13,76	1553	0,31
		0		0		0	
<b>S3-7</b>	1518*	1518	0,55	1518	2,24	1518	0,29
		0		0		0	
<b>S3-8</b>	1526*	1526	0,53	1526	4,87	1526	0,48
		0		0		0	

<b>S3-9</b>	1441*	1441	0,84	1441	8,35	1441	0,52
		0		0		0	

Tabela 6 - Resultados médios obtidos para o PGDLPP com máquina simples em S4

Instancias	AMPL	SA1		SA2		SA3	
		Fitness	Tempo	Fitness	Tempo	Fitness	Tempo
<b>S4-0</b>	1833	1833	3,51	1833	16,83	1833	10,51
		0		0		0	
<b>S4-1</b>	1986	1986	3,68	1986	13,03	1986,2	36,33
		0		0		0,01	
<b>S4-2</b>	1825	<b>1805</b>	12,17	<b>1805</b>	84,84	<b>1808</b>	72,26
		-1,09		-1,09		-1,09	
<b>S4-3</b>	1674	1779	8,14	1779	37,45	1779	27,16
		6,27		6,27		6,27	
<b>S4-4</b>	1974	1974	3,02	1974	12,35	1974	8,51
		0		0		0	
<b>S4-5</b>	1994	1994	4,96	1994	19,23	1994	45,65
		0		0		0	
<b>S4-6</b>	1790	1790	2,59	1790	10,76	1790	14,33
		0		0		0	
<b>S4-7</b>	1786	<b>1777</b>	0,73	<b>1777</b>	6,21	<b>1777</b>	6,67
		-0,50		-0,50		-0,50	
<b>S4-8</b>	2150	<b>2139</b>	1,93	<b>2139</b>	9,69	<b>2139</b>	7,48
		-0,51		-0,51		-0,51	
<b>S4-9</b>	1782	1782	1,53	1782	7,22	1782	3,52
		0		0		0	

A Tabela 3 mostra que o método, em suas três variações, foi capaz de encontrar o valor ótimo em 5 das 10 instâncias geradas e conseguiu valor factível nas demais com desvio abaixo de 1% em relação ao método exato. Nas Tabelas 4 e 5 os métodos obtiveram 9 soluções ótimas e 1 factível. Os desvios, considerando as soluções factíveis, ficaram também abaixo de 1%. Já na Tabela 6, o resultado se mostra mais satisfatório, pois os métodos encontraram 9 soluções ótimas, exceto o método SA3 que encontrou 8 soluções ótimas. Há três instâncias nas quais os métodos apresentam desvios negativos, ou seja, soluções melhores que aquelas encontradas pelo AMPL. Isso ocorreu nas instâncias S4-2, S4-7 e S4-8.

A partir destes resultados foi estabelecido o gráfico da Figura 11. Este gráfico ilustra os desvios médios dos resultados, ou seja, a média dos desviosAMPL calculados pela Equação (12), de cada variação do *Simulated Annealing* em relação a cada grupo de instâncias.

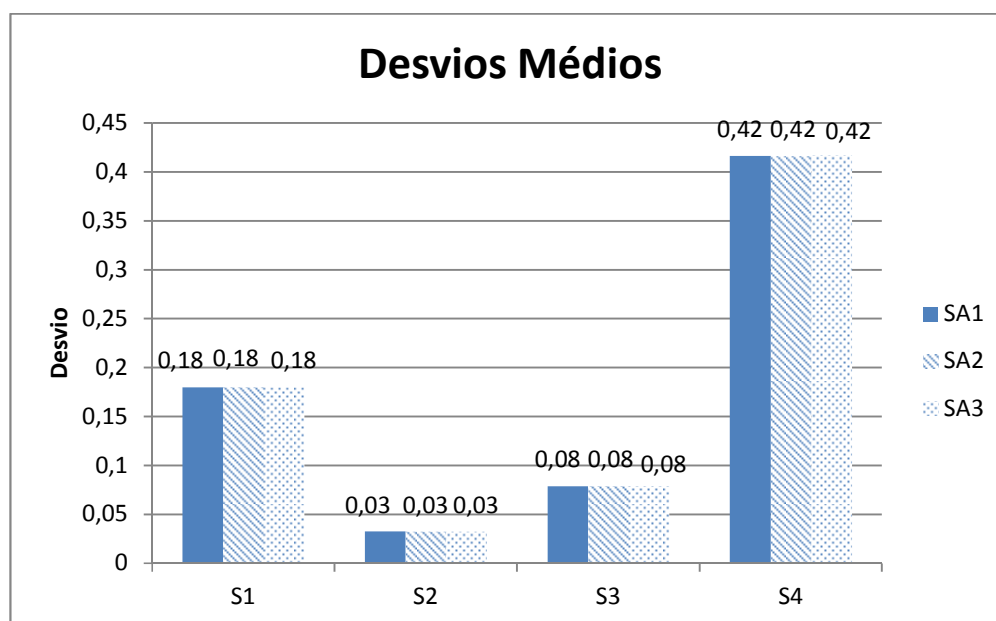


Figura 11 - Desvios médios dos diferentes métodos *Simulated Annealing* para instâncias de máquinas simples

Avaliando apenas os desvios médios não é possível dizer qual método obteve o melhor desempenho, já que os três apresentam resultados bastante semelhantes. Desta forma, foi necessário estabelecer o gráfico da Figura 12 que ilustra os tempos médios gasto por cada método para se encontrar o melhor resultado. Analisando esse gráfico, fica fácil perceber a superioridade do método SA1 que obteve tempos menores na maioria dos grupos de instancias.

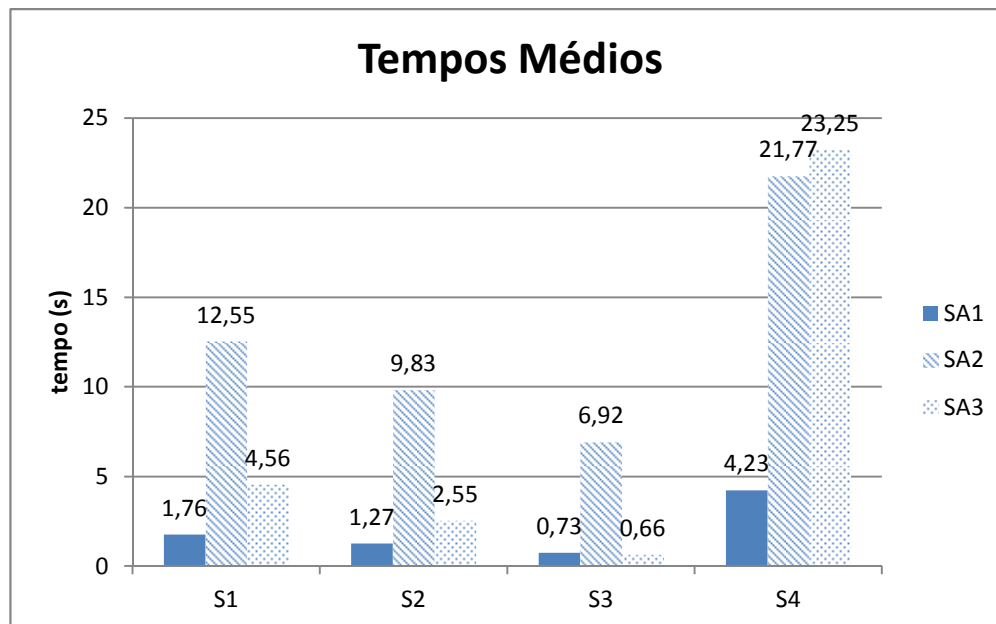


Figura 12 - Tempos médios dos diferentes métodos *Simulated Annealing* para instâncias de máquinas simples

Desta forma, considerando os dois aspectos avaliados, o método SA é o mais indicado na resolução de instâncias do PGDLPP com máquinas simples. Isso se deve ao fato dele obter a mesma qualidade de solução em um tempo de convergência inferior aos demais métodos, exceto pelo subconjunto S3.

Os resultados obtidos nas execuções para máquinas paralelas podem ser vistos nas tabelas 7 a 12. Devido à complexidade das instâncias, os resultados exibidos pelo AMPL são as melhores soluções encontradas ao final do tempo de execução dado ao método.

Tabela 7 - Resultados médios obtidos para o PGDLPP com máquinas paralelas em P1

Instancias	AMPL	SA1		SA2		SA3	
		Fitness	Tempo	Fitness	Tempo	Fitness	Tempo
<b>P1-0</b>	1765,54	<b>1737,47</b>	34,24	<b>1737,47</b>	126,92	1737,59	285,99
		-1,59		-1,59		-1,58	
<b>P1-1</b>	1752,46	<b>1750,30</b>	20,71	<b>1750,30</b>	38,27	1781,32	358,67
		-0,12		-0,12		1,64	
<b>P1-2</b>	1598,30	1862,62	0,78	1862,62	8,35	1862,62	0,86
		16,53		16,53		16,53	
<b>P1-3</b>	1498,54	1554,97	14,52	1554,97	82,26	1565,16	260,35
		3,76		3,76		4,44	
<b>P1-4</b>	1493,31	1556,87	36,73	1556,87	126,66	1572,90	276,80
		4,25		4,25		5,32	

Na tabela 7 os resultados se apresentam favoráveis ao método exato, onde obteve vitória sobre 3 das 5 instâncias em comparação com os métodos SA1 e SA2 e 4 vitórias sobre o método SA3.

Tabela 8 - Resultados médios obtidos para o PGDLPP com máquinas paralelas em P2

Instancias	AMPL	SA1		SA2		SA3	
		Fitness	Tempo	Fitness	Tempo	Fitness	Tempo
<b>P2-0</b>	1002,83	1040,62	12,28	1040,62	39,34	1103,58	421,95
		3,76		3,76		10,04	
<b>P2-1</b>	1161,31	<b>1144,68</b>	5,73	<b>1144,68</b>	29,22	1213,59	374,56
		-1,43		-1,43		4,50	
<b>P2-2</b>	1625,10	1669,42	10,50	1669,42	27,18	1674,24	195,63
		2,72		2,72		3,02	
<b>P2-3</b>	1544,75	<b>1518,80</b>	14,20	<b>1518,80</b>	99,67	1648,23	252,22
		-1,67		-1,67		6,69	
<b>P2-4</b>	1163,30	<b>1163,30</b>	3,57	<b>1163,30</b>	23,68	1323,18	517,18
		-8,59		-8,59		13,74	

Tabela 9 - Resultados médios obtidos para o PGDLPP com máquinas paralelas em P3

Instancias	AMPL	SA1		SA2		SA3	
		Fitness	Tempo	Fitness	Tempo	Fitness	Tempo
P3-0	420,38	420,38	2,09	420,38	27,92	592,03	427,18
		8,96		8,96		40,83	
P3-1	496,04	496,04	1,20	496,04	5,27	732,22	392,93
		0		0		47,61	
P3-2	523	523	0,41	523	0,55	937,88	507,73
		0		0		79,32	
P3-3	538,70	538,70	0,28	538,70	0,24	723,20	338,10
		3,60		3,60		34,24	
P3-4	850,27	<b>850,27</b>	1,37	<b>850,27</b>	22,99	1052,77	321,65
		-1,71		-1,71		23,81	

Os resultados das Tabelas 8 e 9 demonstram principalmente que o método SA3 foi o mais ineficiente, pois não conseguiu superar as demais abordagens. SA1 e SA2 conseguiram resultados satisfatórios, superando o método exato em 3 das 5 instâncias no subconjunto P2 e praticamente igualando os resultados do método exato em P3.

Tabela 10 - Resultados médios obtidos para o PGDLPP com máquinas paralelas em P4

Instancias	AMPL	SA1		SA2		SA3	
		Fitness	Tempo	Fitness	Tempo	Fitness	Tempo
P4-0	5819,75	4112,13	492,53	<b>4094,69</b>	601,30	4635,74	382,84
		-29,34		-29,64		-20,34	
P4-1	5560,99	<b>4019,17</b>	512,48	4056,48	455,44	4634,27	374,45
		-27,72		-27,05		-16,66	
P4-2	4371,45	<b>3490,14</b>	416,11	3525,65	508,45	4116,97	542,98
		-20,16		-19,34		-5,82	
P4-3	5043,77	<b>3893,62</b>	451,34	3906,63	558,38	4476,02	536,08
		-22,80		-22,54		-11,25	
P4-4	4536,87	<b>4013,69</b>	378,49	4025,01	475,13	4649,80	326,61
		-11,53		-11,28		2,48	

Tabela 11 - Resultados médios obtidos para o PGDLPP com máquinas paralelas em P5

Instancias	AMPL	SA1		SA2		SA3	
		Fitness	Tempo	Fitness	Tempo	Fitness	Tempo
P5-0	5730,18	3576,56	543,05	<b>3571,09</b>	473,46	4691,73	330,46
		-37,58		-37,67		-18,12	
P5-1	5364,30	3694,25	241,19	<b>3682,14</b>	358,39	4735,60	425,31
		-31,13		-31,35		-11,72	
P5-2	6375,30	4009,87	417,62	<b>3944,04</b>	499,29	5040,98	474,54
		-37,10		-38,13		-20,92	
P5-3	7045,37	3889,73	483,75	<b>3879,90</b>	571,79	4993,36	638,66
		-44,79		-44,92		-29,12	
P5-4	6666,00	3892,75	518,65	<b>3866,54</b>	588,90	5095,22	576,34
		-41,60		-41,99		-23,56	

Tabela 12 - Resultados médios obtidos para o PGDLPP com máquinas paralelas em P6

Instancias	AMPL	SA1		SA2		SA3	
		Fitness	Tempo	Fitness	Tempo	Fitness	Tempo
P6-0	5679,64	<b>3457,68</b>	435,91	3464,85	416,89	5216,57	484,20
		-39,12		-38,99		-8,15	
P6-1	7713,07	3399,29	538,50	<b>3363,80</b>	480,34	5312,73	419,96
		-55,92		-56,38		-31,12	
P6-2	7806,11	3306,85	485,71	<b>3293,70</b>	364,98	5178,12	435,28
		-57,63		-57,80		-33,66	
P6-3	7836,66	3454,77	487,51	<b>3445,20</b>	241,41	5314,66	254,19
		-55,91		-56,03		-32,18	
P6-4	7130,80	3338,07	331,92	<b>3299,24</b>	584,99	5022,52	463,82
		-53,18		-53,73		-29,56	

Por outro lado, o método exato encontra mais dificuldade em determinar boas soluções nas instâncias mais complexas apresentadas nas Tabelas 10, 11 e 12. Os métodos propostos superam o método exato em todas as instâncias dos subconjuntos P4, P5 e P6.

SA consegue os melhores resultados em P4, enquanto SA2 é superior no subconjunto P5. No subconjunto P6, SA2 também obteve os melhores resultados exceto pela instância P6-0.

Desta forma, todas as três variações do método *Simulated Annealing* conseguem resultados melhores que o método exato na maioria das instâncias, onde SA3 surge como a abordagem menos eficiente. Um gráfico com as vitórias dos métodos sobre o AMPL está ilustrado no gráfico da Figura 13.

O número de vitórias dos métodos heurísticos demonstra que o método exato encontra dificuldades para retornar boas soluções a medida que a complexidade das instâncias aumenta. Os gráficos de média dos desvios médios podem ser vistos NAS Figuras 14 e 15.

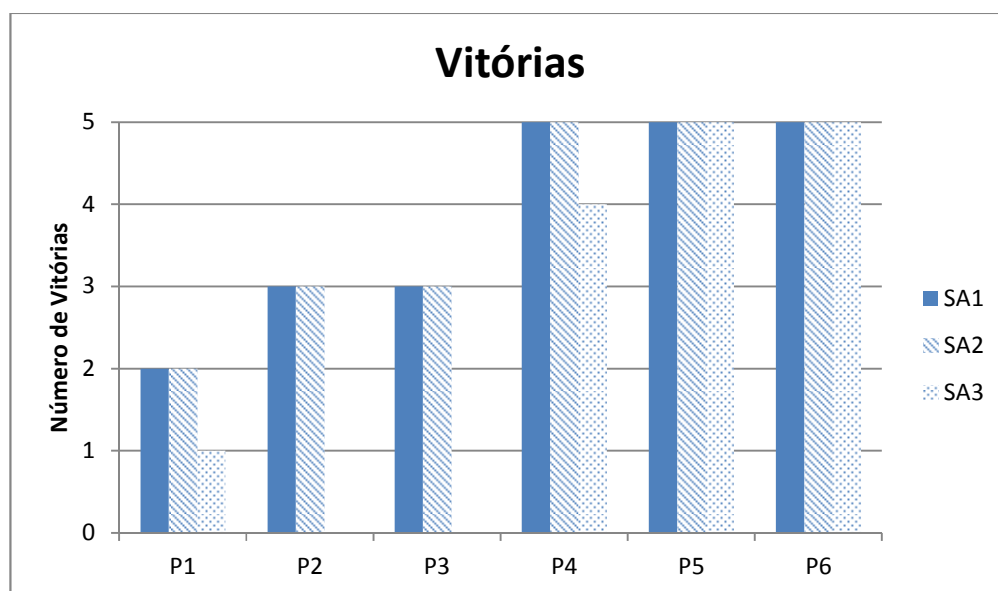


Figura 13 - Vitórias dos métodos *Simulated Annealing* sobre o AMPL

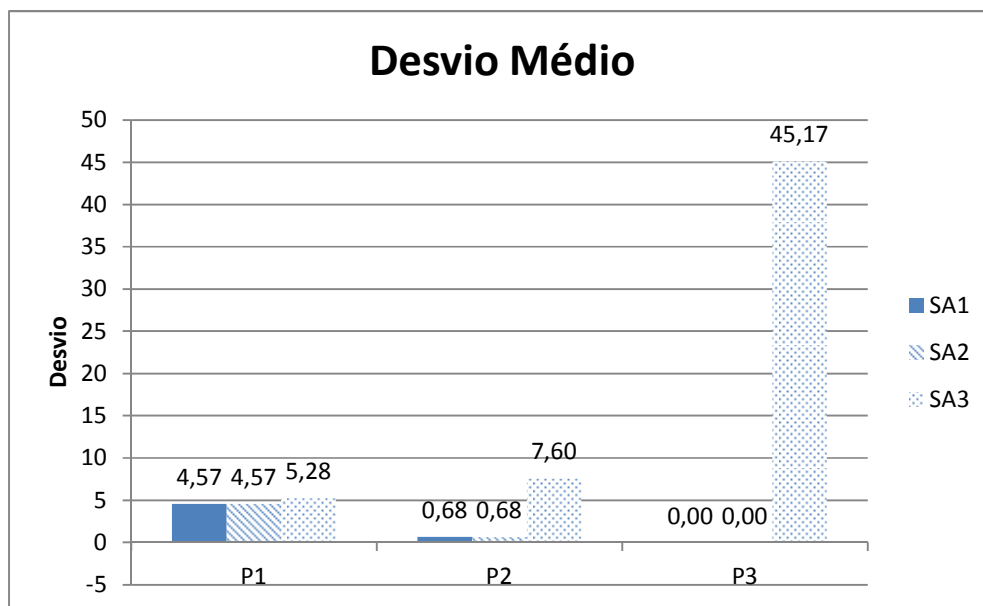


Figura 14 - Desvios médios dos diferentes métodos *Simulated Annealing* para instâncias de máquinas paralelas P1, P2 e P3

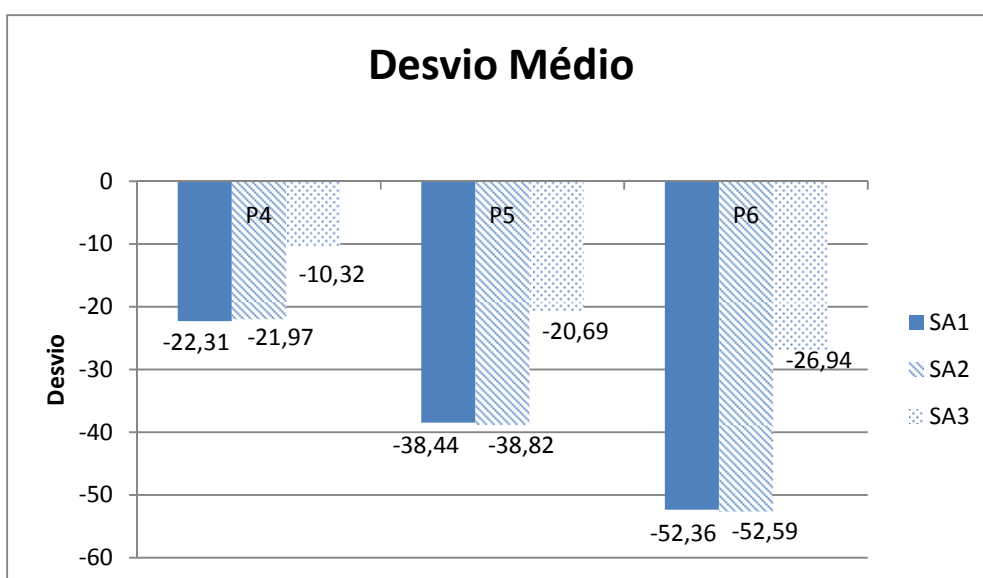


Figura 15 - Desvios médios dos diferentes métodos *Simulated Annealing* para instancias de máquinas paralelas P4, P5 e P6

As figuras 14 e 15 indicam que SA1 e SA2 apresentam, em média, desempenho bastante semelhantes, mas há uma certa superioridade de SA2 nas instâncias mais complexas. Por outro lado, SA3 se confirma como a abordagem menos indicada. Os tempos médios podem ser vistos nas Figuras 16 e 17.

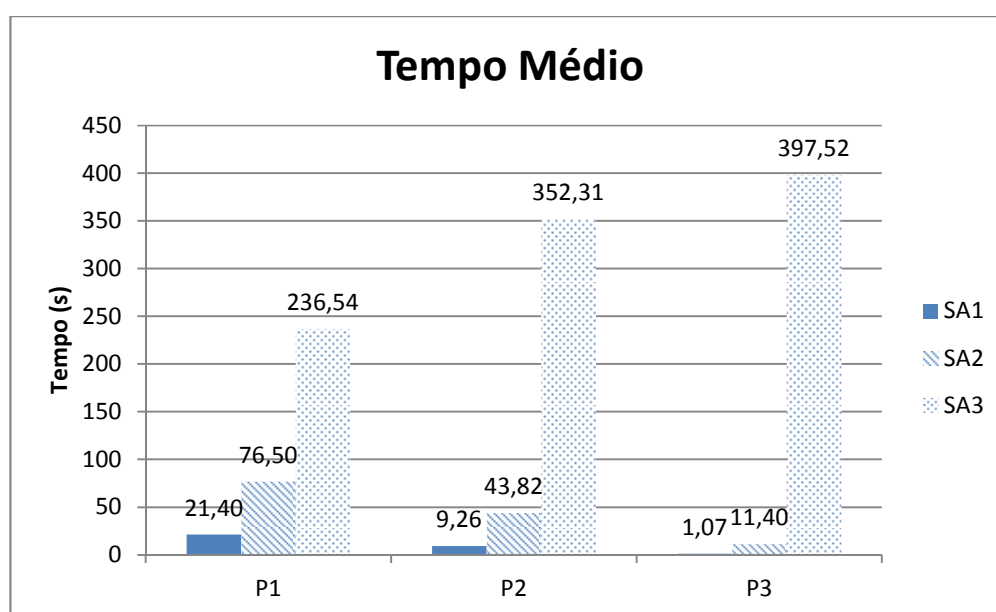


Figura 16 - Tempos médios dos diferentes métodos *Simulated Annealing* para instâncias de máquinas paralelas P1, P2 e P3

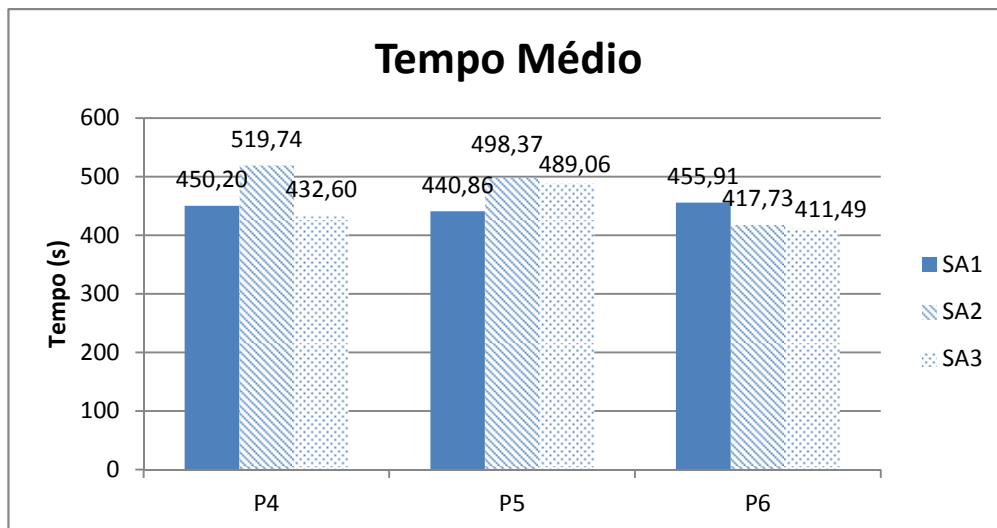


Figura 17 - Tempos médios dos diferentes métodos *Simulated Annealing* para instâncias de máquinas paralelas P4, P5 e P6

Os gráficos das Figuras 16 e 17 demonstram que SA1 continua sendo a abordagem com convergência mais rápida para os melhores resultados em praticamente todos os subconjuntos, onde a única exceção foi o subconjunto P6 (figura 17).

Os resultados nos permitem concluir que há um desempenho mais satisfatório quando utilizado a expressão de resfriamento (9) para instâncias mais simples. Porém, para instâncias mais complexas, o método SA2, que utiliza a expressão de resfriamento (10), se mostra a melhor escolha, pois, apesar de ter um tempo médio um pouco mais elevado, retorna os melhores resultados conforme a complexidade aumenta.

#### 4.3 *Simulated Annealing* x Outros métodos

Nesta seção, os métodos *Simulated Annealing* propostos são comparados com o método exato AMPL e com outras heurísticas presentes na literatura. As heurísticas usadas para comparação são:

- Busca Tabu (*Tabu Search*);
- Algoritmo Genético

Ambas propostas em Toledo et. al. (2009) para solucionar o PGDLPP com máquinas simples e máquinas paralelas.

Inicialmente, as Tabelas 13 e 14 apresentam essa comparação através dos resultados de testes realizados para instâncias com máquina simples. Nestes testes, para cada instância, os métodos foram executados dez vezes durante 30 minutos. O método *Simulated Annealing* escolhido para execuções com uma máquina utiliza a expressão de resfriamento (9), pois apresentou melhores resultados, como visto na seção anterior.

Tabela 13 - Resultados médios obtidos para o PGDLPP com os diversos métodos para S1 e S2

Instancias	AMPL	SA1	TS	AG	Instancias	AMPL	SA1	TS	AG
<b>S1-0</b>	1658	1658	1658	1658	<b>S2-0</b>	1729	1729	1737	1746
		0	0	0			0	0,46	0,98
<b>S1-1</b>	1421	1421	1421	1421	<b>S2-1</b>	1758	1758	1758	1758
		0	0	0			0	0	0
<b>S1-2</b>	1700	1703	1700	1700	<b>S2-2</b>	2162	2162	2162	2162
		0,17	0	0			0	0	0
<b>S1-3</b>	1459	1471	1459	1459	<b>S2-3</b>	1747	1747	1747	1748,2
		0,82	0	0			0	0	0,06
<b>S1-4</b>	1543	1543	1543	1551	<b>S2-4</b>	1638	1638	1638	1638
		0	0	0,51			0	0	0
<b>S1-5</b>	1402	1402	1402	1402	<b>S2-5</b>	1534	1539	1534	1534
		0	0	0			0,3260	0	0
<b>S1-6</b>	1477	1480	1480	1486,5	<b>S2-6</b>	1533	1533	1533	1533
		0,20	0,20	0,64			0	0	0
<b>S1-7</b>	1463	1463	1463	1463	<b>S2-7</b>	1822	1822	1822	1833,8
		0	0	0			0	0	0,64
<b>S1-8</b>	1480	1487	1487	1487	<b>S2-8</b>	1863	1863	1863	1864,7
		0,47	0,47	0,47			0	0	0,09
<b>S1-9</b>	1626	1628	1628	1633,4	<b>S2-9</b>	1751	1751	1751	1751
		0,12	0,12	0,45			0	0	0

Tabela 14 - Resultados médios obtidos para o PGDLPP com os diversos métodos para S3 e S4

Instancias	AMPL	SA1	TS	AG	Instancias	AMPL	SA1	TS	AG
------------	------	-----	----	----	------------	------	-----	----	----

<b>S3-0</b>	1269	1279	1269	1269	<b>S4-0</b>	1833	1833	1837	1861,5
		0,78	0	0			0	0,21	1,55
<b>S3-1</b>	1491	1491	1491	1491	<b>S4-1</b>	1986	1986	1986	1989,6
		0	0	0			0	0	0,18
<b>S3-2</b>	1338	1338	1338	1338	<b>S4-2</b>	1825	<b>1805</b>	1823	1827,2
		0	0	0			-1,09	-0,10	0,12
<b>S3-3</b>	1530	1530	1530	1530	<b>S4-3</b>	1674	1779	1674	1681,8
		0	0	0			6,27	0	0,46
<b>S3-4</b>	1663	1663	1663	1667,8	<b>S4-4</b>	1974	1974	1974	1974
		0	0	0,28			0	0	0
<b>S3-5</b>	1542	1542	1542	1542	<b>S4-5</b>	1994	1994	1995,5	2011,3
		0	0	0			0	0,07	0,86
<b>S3-6</b>	1553	1553	1553	1553	<b>S4-6</b>	1790	1790	1790	1793,3
		0	0	0			0	0	0,18
<b>S3-7</b>	1518	1518	1522	1532,1	<b>S4-7</b>	1786	<b>1777</b>	<b>1777</b>	1784,8
		0	0,26	0,92			-0,50	-0,50	-0,06
<b>S3-8</b>	1526	1526	1526	1526	<b>S4-8</b>	2150	<b>2139</b>	2175	2195,4
		0	0	0			-0,51	1,16	2,11
<b>S3-9</b>	1441	1441	1441	1441	<b>S4-9</b>	1782	1782	1788	1788,7
		0	0	0			0	0,33	0,37

Com os resultados gerados é possível obter o gráfico da Figura 18 que ilustra os desvios médios dos diferentes métodos executados para cada conjunto de instancias.

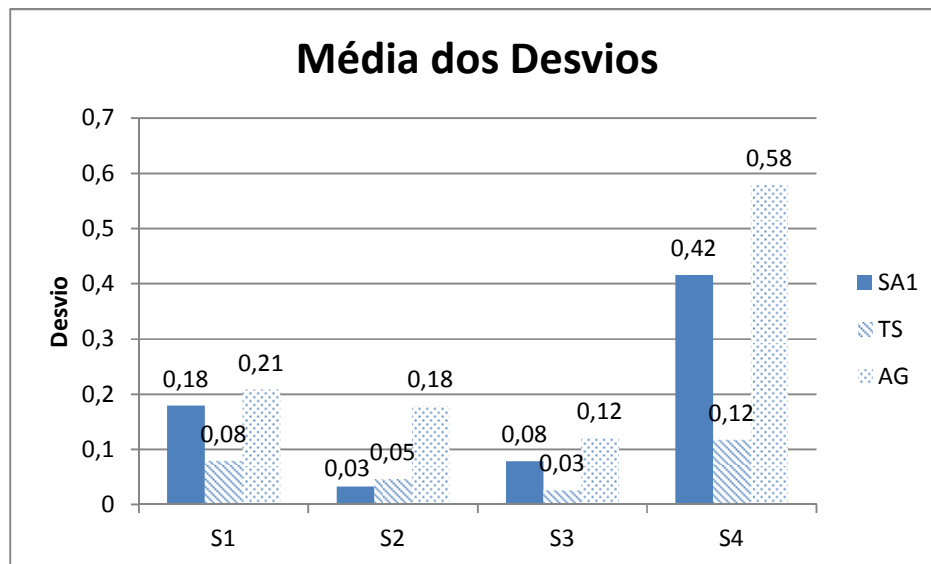


Figura 18 - Desvios médios dos diversos métodos para instâncias de máquinas simples

Para instâncias com máquinas simples o método SA1 foi ligeiramente melhor que a TS apenas em S2. Para as demais instâncias, a Busca Tabu se mostrou mais eficiente.

As Tabelas 15, 16 e 17 apresentam os resultados dos testes realizados para instâncias com máquinas paralelas. Nestes testes, para cada instância os métodos foram executados dez vezes, durante 1 hora em cada execução. O método *Simulated Annealing* escolhido para execuções com mais de uma máquina utiliza a expressão de resfriamento (10), pois apresentou os melhores resultados, como visto na seção anterior.

As instâncias do tipo P3 não foram avaliadas para os métodos Busca Tabu e Algoritmo Genético, não apresentando, portanto, dados tabelados.

Tabela 15 - Resultados médios obtidos para o PGDLPP com os diversos métodos em P1 e P2

Instancias	AMPL	SA2	TS	AG	Instancias	AMPL	SA2	TS	AG
<b>P1-0</b>	1765,55	<b>1737,47</b>	1804,6	2166,6	<b>P2-0</b>	1002,83	1040,62	1315,23	1669,6
		-1,59	2,21	22,71			3,76	31,15	66,48
<b>P1-1</b>	1752,46	<b>1750,30</b>	1805,91	2123,3	<b>P2-1</b>	1161,31	<b>1144,68</b>	1661,19	1793,4
		-0,12	3,04	21,16			-1,43	43,04	54,42
<b>P1-2</b>	1598,30	1862,62	1630,04	1846,7	<b>P2-2</b>	1625,10	1669,42	1905,95	2601,4
		16,53	1,98	15,54			2,72	17,28	60,07
<b>P1-3</b>	1498,54	1554,97	1633,95	1867,4	<b>P2-3</b>	1544,75	<b>1518,80</b>	1867,02	2276,9
		3,76	9,03	24,61			-1,67	20,86	47,39
<b>P1-4</b>	1493,32	1556,87	1642,65	1868,3	<b>P2-4</b>	1163,30	<b>1163,30</b>	1601,98	1932,9
		4,25	10,00	25,11			-8,59	37,70	66,15

Tabela 16 – Resultados médios obtidos para o PGDLPP com os diversos métodos em P4 e P5

Instancias	AMPL	SA2	TS	AG	Instancias	AMPL	SA2	TS	AG
<b>P4-0</b>	5819,75	<b>4094,69</b>	4785,40	4869,10	<b>P5-0</b>	5730,18	<b>3571,09</b>	5142,95	4711,40
		-29,64	-17,77	-16,33			-37,67	-10,24	-17,77
<b>P4-1</b>	5560,99	<b>4056,48</b>	4706,85	4653,40	<b>P5-1</b>	5364,30	<b>3682,14</b>	5036,63	4913,70
		-27,05	-15,35	-16,32			-31,35	-6,10	-8,39
<b>P4-2</b>	4371,45	<b>3525,65</b>	4221,44	4052,30	<b>P5-2</b>	6375,30	<b>3944,04</b>	5593,09	5183,60
		-19,34	-3,43	-7,30			-38,13	-12,26	-18,69
<b>P4-3</b>	5043,77	<b>3906,63</b>	4.480,20	4380,5	<b>P5-3</b>	7045,37	<b>3879,90</b>	5326,12	5326,4
		-22,54	-11,17	-13,15			-44,92	-24,40	-24,39
<b>P4-4</b>	4536,87	<b>4025,01</b>	4636,19	4411,8	<b>P5-4</b>	6666,00	<b>3866,54</b>	5417,91	5552,1
		-11,28	2,18	-2,75			-41,99	-18,72	-16,71

Tabela 17 – Resultados médios obtidos para o PGDLPP com os diversos métodos em P6

Instancias	AMPL	SA2	TS	AG
<b>P6-0</b>	5679,64	<b>3464,85</b>	6154,35	4711,40
		-38,99	8,35	-17,04
<b>P6-1</b>	7713,07	<b>3363,80</b>	6206,19	4913,70
		-56,38	-19,53	-36,29
<b>P6-2</b>	7806,11	<b>3293,70</b>	6122,44	5183,60
		-57,80	-21,56	-33,59
<b>P6-3</b>	7836,66	<b>3445,20</b>	6110,87	5552,1
		-56,03	-22,02	-29,15
<b>P6-4</b>	7130,80	<b>3299,24</b>	5846,67	5530,83
		-53,73	-18,01	-22,43

Em P1 (Tabela 15), não há uma diferença de desempenho significativa entre *Simulated Annealing* e Busca Tabu. Em P2 (Tabela 15), o método proposto consegue melhores resultados em 3 das 5 instâncias, e seu desvio médio é menor que 1% contra mais de 30% para os outros métodos. No entanto, para todas as outras instâncias, o método *Simulated Annealing* passa a obter soluções muito melhores que as demais abordagens. Os gráficos das Figuras 19 e 20 ilustram a média dos desvios obtidos pelos métodos em questão e possibilitam comparar o desempenho dos mesmos.

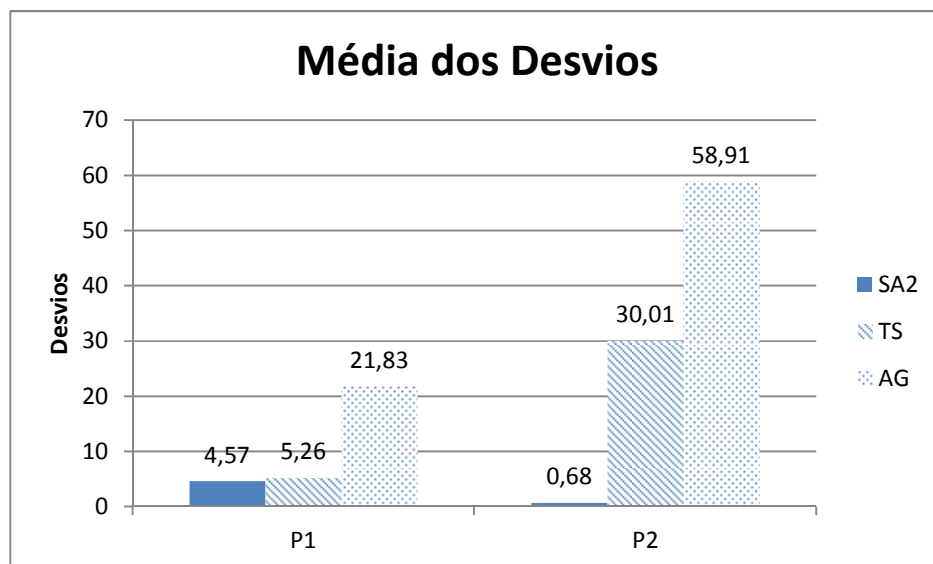


Figura 19 - Desvios médios dos diversos métodos para instâncias de máquinas paralelas P1 e P2

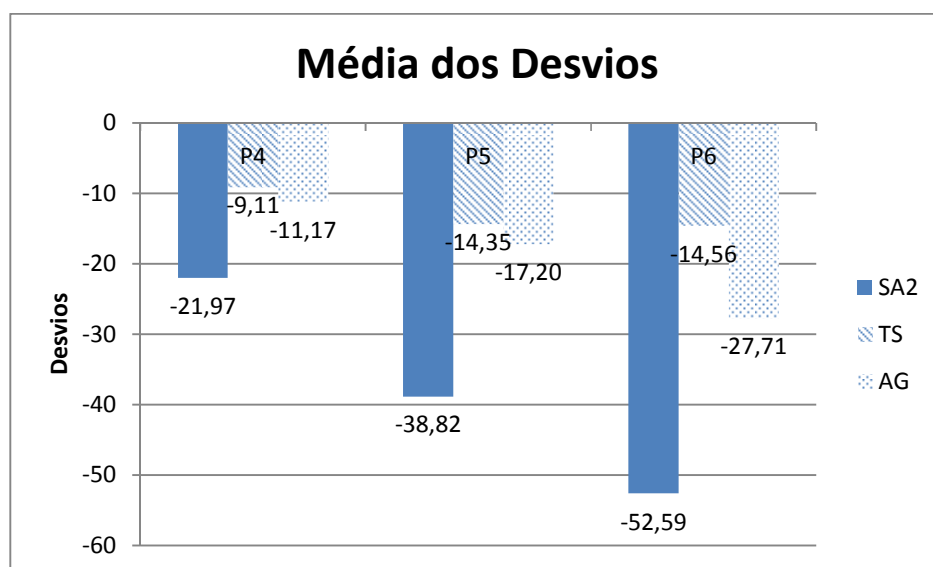


Figura 20 - Desvios médios dos diversos métodos para instâncias de máquinas paralelas P4, P5 e P6

#### 4.4 Conclusão

Os resultados nos permitem concluir que, para problemas com máquinas simples, o método heurístico Busca Tabu é a melhor escolha quando comparado ao SA. Já para problemas com máquinas paralelas, o método heurístico *Simulated Annealing* SA2 obtém os melhores resultados. Logo, a função de resfriamento dada pela equação (10) foi a escolha mais apropriada para resolução de instâncias mais complexas.

Observando a Figura 21, pode-se ver que SA1, utilizando a Equação (9), possui um decaimento exponencial de sua temperatura e assim pode abranger todo o espaço de soluções das instâncias simples. Para instâncias de máquinas paralelas, o espaço de soluções é muito grande e SA2, que possui um resfriamento mais rápido e em temperaturas mais baixas o resfriamento é mais lento, se comporta melhor, uma vez que é reaquecido poucas vezes e consegue buscar melhores soluções em uma determinada vizinhança já que em temperaturas mais baixas, as chances de se aceitar uma solução de piora é muito menor. Por sua vez, SA3, que utiliza a expressão de resfriamento da Equação (11), trabalha em temperaturas mais altas com um resfriamento muito lento e assim com alta probabilidade de aceitar uma solução pior, o que não é o ideal para casos mais complexos.

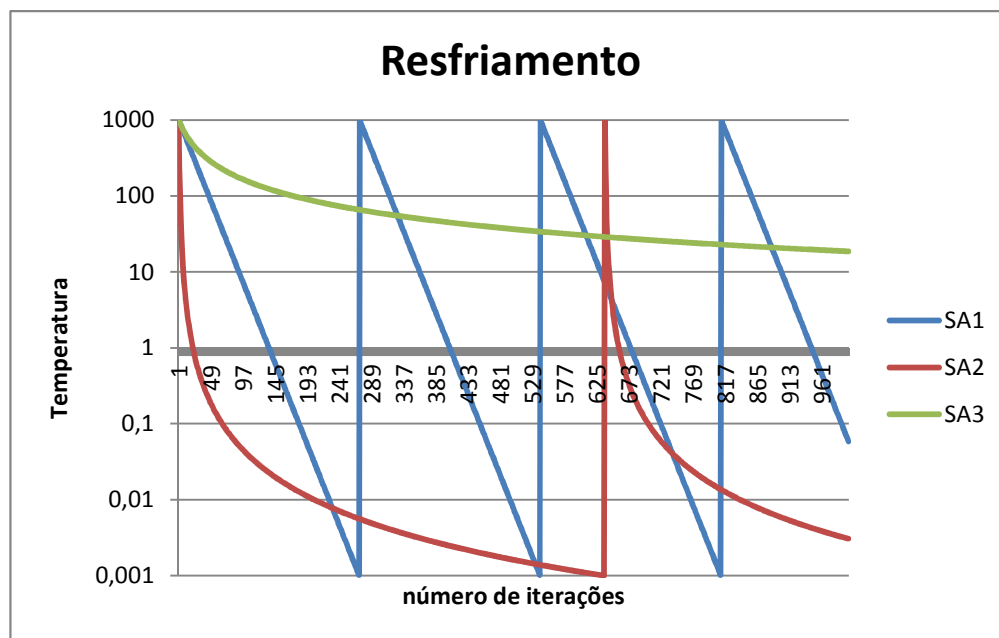


Figura 21 – Gráfico de descaimento da temperatura para as três funções de resfriamento

## 5 CONCLUSÕES

O principal objetivo deste trabalho é o estudo e proposta de uma abordagem do *Simulated Annealing* capaz de resolver o PGDLPP. Os objetivos propostos incluem o estudo e resolução do problema original e sua extensão para cenários com várias máquinas trabalhando em paralelo. Foram utilizadas instâncias do PGDLPP e, a partir destas, realizados os testes de desempenho do *Simulated Annealing*. Os resultados obtidos pelo *Simulated Annealing* foram comparados com métodos exatos e heurísticos.

A fim de encontrar a melhor configuração para o método abordado, foram utilizadas três equações de resfriamento. Destas, conclui-se que, para instâncias simples, ao utilizar a Equação (9) de resfriamento exponencial, obtém-se resultados mais satisfatórios. Porém, para se obter melhores resultados em instâncias complexas, faz-se necessário o uso da Equação (10) que se ajusta melhor a esse tipo de instância.

Em comparação com outros métodos, os resultados apresentados mostram que o *Simulated Annealing* é um algoritmo bastante eficiente na resolução do PGDLPP, principalmente para instâncias mais complexas. Isto aumenta a viabilidade de sua aplicação em problemas reais, pois estes normalmente apresentam um alto grau de complexidade.

Trabalhos futuros poderão estudar outras equações de resfriamento para o método a fim de encontrar melhores resultados.

## 6 REFERÊNCIAS BIBLIOGRÁFICAS

BARANY, I., VAN ROY, T. J., WOLSEY, L. A., **Strong formulations for multi-item capacitated lot sizing**, *Management Science*, 30(10):1255–61, 1984.

BITRAN, G. R., YANASSE, H. H. **Computational Complexity of Capacitated Lot Size Problem**. *Management Science*, n. 28, p. 1178-1186, 1982.

BLUM, C. e ROLI, A.. **Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison**. *ACM Computing Surveys*, v. 35, n. 3, p. 268–308, 2003.

BUENO, F., **Métodos Heurísticos**, Araranguá, Instituto de Física de São Carlos, IFSC, 2009.

CHEN, W. H. e THIZY, J. M., **Analysis of relaxations for the multi-item capacitated lot-sizing problem**, *Annals of Operations Research*, 26, 29-72, 1990.

CONCILIO, R.. **Contribuições à solução de problemas de escalonamento pela aplicação conjunta de computação evolutiva e otimização com restrições**. Dissertação de Mestrado, Campinas, SP, Unicamp, 2000.

CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L e STEIN, C.. **Algoritmos: Teoria e Prática**. Tradução da Segunda Edição Americana por Vandenberd D. de Souza. Campus, Rio de Janeiro, 2002.

FLEISCHMANN, B., **The discrete lot sizing and scheduling problem**, *European Journal of Operational Research* 44, 337–348, 1990.

FLEISCHMANN, B., MEYR, H., **The general lotsizing and scheduling problem**, *OR Spektrum*, 19, 11-21, 1997.

FRAGA, M. C. P.. **Uma metodologia híbrida Colônia de Formigas – Busca Tabu – Reconexão por Caminhos para resolução do problema de Roteamento de Veículos com Janela de Tempo**. Dissertação de Mestrado, Belo Horizonte, MG, CEFET-MG, 2006.

GLOVER, F., **Tabu search: Part I**, *ORSA Journal on Computing*, 1 (3), 190-206, 1989.

GLOVER, F., **Tabu search: Part II**, *ORSA Journal on Computing*, 2 (1), 4-32, 1990.

GLOVER, F., LAGUNA, M., *Tabu search*, Boston: Kluwer Publishers, 1997.

HAASE, K., **Capacitated lot-sizing with sequence dependent setup costs**, *OR Spektrum*, 8, 51-59, 1996.

HOLLAND, J. H.. **Adaptation in Natural and Artificial Systems**, 2a edição, MIT Press, 1992.

IZQUIERDO, V. B., **Uma proposta de especificação formal e fundamentação teórica para *Simulated Annealing***, Porto Alegre, Instituto de Informática, UFRGS, 2000.

JANS, R., DEGRAEVE, Z., **Meta-heuristics for dynamic lot sizing: A review and comparison of solution approaches**, *European Journal of Operational Research* 177, p. 1855–1875, 2007.

KARIMI, B.; GHOMI, S. M. T. F.; WILSON, J. M., **The capacitated lot sizing problem: a review of models and algorithms**. OMEGA, v. 31, p. 365–378, 2003.

KIRKPATRICK, S., GELLAT, C. D., VECCHI, M. P., **Optimization by *Simulated Annealing***. Science, 1983.

LAND, A. H., DOIG, A. G., **An Automatic Method for Solving Discrete Programming Problems**, Econometrica, Vol.28, p. 497-520, 1960.

LEUNGE J., MAGNANTI, T., VACHANI, R.. **Facets and algorithms for capacited lot sizing**, Mathematical Programing 45, 1989.

MARQUES NETO, M. C.; BARBOSA, F. ; FREITAS, C. ; BAGANO, P.. **Uma Ferramenta Baseada em Algoritmos Genéticos para a Geração de Tabela de Horário Escolar**. In: Sétima Escola Regional de Computação Bahia-Sergipe, 2007, Vitória da Conquista, 2007.

METROPOLIS, N. et al.. **Equation of State Calculation by Fast Computer Machines**. The Journal of Chemical Physics, New York, v. 1, n. 6, p. 1087-1092, 1953.

MEYR, H., **Simultaneous lot sizing and scheduling on parallel production lines**, European Journal of Operational Research 39, p. 277–292, 2002.

MICHALEWICZ, Z., FOGEL, D. B., **How to Solve It: Modern Heuristics**, 2nd ed., XVIII, 554 p. 174 illus., Hardcover, 2004.

MITCHELL, M.. **An introduction to genetic algorithms**. 1ª edição. Londres: MIT Press., 1998.

RAZA, A. S., AKGUNDUZ, A.. **A comparative study of heuristic algorithms on Economic Lot Scheduling Problem.** Computers & Industrial Engineering, Volume 55, Issue 1, 2008.

RIBEIRO, C. C.. **Metaheuristics and applications.** In Advanced School on Artificial Intelligence, Portugal, 2002.

STUTZLE, T. e DORIGO, M.. **The Ant Colony Optimization Metaheuristic: Algorithms, Applications, and Advances.** Technical Report IRIDIA-2000-32, 2000.

TOLEDO, C. F. M. **Problema Conjunto de Dimensionamento de Lotes e Programação da Produção.** Tese de Doutorado, Campinas, SP, UNICAMP, 2005.

TOLEDO, C. F. M., FRANÇA, P. M., ROSA, K. A. **Evaluating Genetic Algorithms with Different Population Structures on a Lot Sizing and Scheduling Problem.** In: 23rd Annual ACM Symposium on Applied Computing, Fortaleza, p. 1777-1781, 2008a.

TOLEDO, C. F. M., FRANÇA, P. M., MORABITO, R., KIMMS, A. **Multi-population genetic algorithm to solve the synchronized and integrated two-level lot sizing and scheduling problem.** International Journal of Production Research, p. 1-23, 2008b.

TOLEDO, C. F. M., FRANÇA, P. M., FERREIRA, J. E. **Meta-Heuristic Approaches for a Soft Drink Industry Problem.** In: 13th IEEE International Conference on Emerging Technologies and Factory Automation, Hamburg, Alemanha, p. 1384-1391, 2008c.

TOLEDO, C. F. M., FERREIRA, J. E., SIMEONE, F., ROSA, G. P.. **Metaheurísticas Aplicadas ao Problema Geral de Dimensionamento de Lotes e Programação da Produção.** In: XLI Simpósio Brasileiro de Pesquisa Operacional, João Pessoa, 2009.

ZORZAL, E. R. ; SILVA, L. F.; CARDOSO, A.; KIRNER, C.; JÚNIOR, E. L.; YAMANAKA, K.. **Associando Realidade Virtual e o Algoritmo de Busca Tabu para o Problema de Carregamento de Veículos.** In: II Workshop de Aplicações de Realidade Virtual - WARV2006, Pernambuco, 2006.