

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA
DEPARTAMENTO DE ENGENHARIA DE SISTEMAS

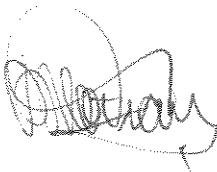
PROBLEMAS DE ROTEAMENTO DE VEÍCULOS
VIA METAHEURÍSTICA TABU

Este exemplar corresponde à
redação final da tese defendida
por Vitória M. Miranda Pureza
e aprovada pela Comissão Julgadora
em 20/08/90.

por: Vitória M. M. Pureza *et al.*

Orientador: Paulo Morelato França

/90



Tese de Mestrado apresentada à
Faculdade de Engenharia Elétrica da
Universidade Estadual de Campinas

Agosto de 1990

Este trabalho contou com o apoio financeiro do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

Dedicado a Mário, Joana, Júlio, Emília e Tonzé

AGRADECIMENTOS

Aos profs. Paulo Morelato França, Jurandir Fernando R. Fernandes, Antônio Galvão Novaes e Pedro Luis Dias Peres.

Ao prof. Anésio dos Santos Jr.

Aos colegas Aguinaldo R. de Campos Jr., Antonio Sergio Ando, Aurelio Leite, Felipe Martins Müller, Lúcia Filomena Guimarães e Naasson Alcântara.

A Miguel Rozsas e Gorgônio Araújo.

A Eduardo Moscoso.

A Nabil Jorge Miguel e Zbgniew Koperczak.

A Ana Cristina Müller e Cláudia Ayres Lemonache.

A Ana Clelia Ferreira, Jorge Vicente Lopes da Silva e Sílvio Henrique Brunoro Grilo.

A Sílvio Luiz Lima Nogueira.

À Faculdade de Administração e à Faculdade de Ciência da Computação da Universidade de Ottawa, que gentilmente cederam seus recursos computacionais.

Agradecimentos especiais a Sergio Jachtchenco.

RESUMO

O Problema de Roteamento de Veículos (PRV) consiste basicamente em definir rotas eficientes para uma frota de veículos que deve entregar quantidades de bens a um conjunto de clientes. Vários métodos têm sido propostos para tal tarefa, mas devido ao esforço computacional requerido, problemas de maior porte (50 clientes ou mais) são resolvidos por algoritmos aproximados.

Dentre estes algoritmos aproximados, abordamos os métodos de melhoria de rotas. Estes métodos são caracterizados pela geração de uma solução inicial factível, seguida da aplicação de mecanismos de busca que alteram a solução inicial. Estes mecanismos promovem essencialmente a melhoria da função objetivo em direção a um mínimo local. Neste ponto, dada a falta de movimentos de melhoria, o algoritmo pára.

Apesar do desempenho excelente deste métodos, observa-se uma limitação fundamental. Sendo o PRV um problema combinatório e portanto não convexo, o ótimo local obtido pode não ser o ótimo global. Consequentemente, a qualidade da solução final depende drasticamente da solução de partida.

Várias técnicas foram elaboradas com vistas à superação da otimalidade local. A maioria delas recomeça o processo de busca a partir de soluções iniciais diferentes ou atrasa a obtenção do ponto ótimo.

Outra maneira de lidar com tais limitações é através da aplicação da estratégia de Busca Tabu. Ao invés de evitar ótimos locais, a Busca Tabu os supera, permitindo assim a continuidade das explorações.

Neste trabalho apresentamos um estudo da aplicação da técnica de Busca Tabu ao

PRV. Um algoritmo dotado de mecanismos de Busca Tabu foi utilizado para a resolução de vinte problemas caracterizados pela existência (ou ausência) de certas restrições temporais. Procedemos a várias análises do comportamento do algoritmo e comparações com outros métodos heurísticos. Os resultados indicaram ser a Busca Tabu uma ferramenta poderosa na resolução de problemas combinatórios.

CONTEÚDO

Página

I. O Problema de Roteamento de Veículos.....	1
I.1. Introdução.....	1
I.2. Métodos Exatos.....	7
I.3. Métodos Aproximados.....	8
I.3.1. Métodos de Construção de Rotas.....	8
I.3.2. Métodos de Duas Fases.....	11
I.3.3. Métodos de Otimização Incompleta.....	12
I.3.4. Métodos de Melhoria de Rotas.....	13
I.4. Limitações dos métodos "hill climbing".....	15
II. A Busca Tabu : Fundamentos.....	18
II.1. Introdução.....	18
II.2. Definições e Nomenclatura.....	19
II.3. A Busca Tabu e seus Elementos Básicos.....	21
II.4. Formas de Lista Tabu.....	25
II.5. Níveis de Aspiração.....	29
II.5.1. Implementação.....	30
II.6. Listas Tabu e Oscilação Estratégica.....	37
II.6.1. Aplicações em Pós-otimização.....	38
II.6.2. Utilização Alternada de Métodos Opostos.....	39
II.7. Funções de Memória de Médio e Longo Prazo.....	41

III. A Busca Tabu Aplicada a Problemas de Roteamento de Veículos.....	44
III.1. Introdução.....	44
III.2. Descrição do Algoritmo.....	47
III.3. A Lista Tabu.....	58
III.3.1. Especificação dos atributos do movimento.....	58
III.3.2. Composição da Lista Tabu.....	60
IV. Desenvolvimento do Trabalho - Composição da Lista Tabu.....	62
IV.1. Introdução.....	62
IV.2. Atributo Economia.....	63
IV.3. Atributo Custo.....	65
IV.4. Atributo Configurações Parciais da Solução.....	69
IV.5. Atributo Arestas Adicionadas e Eliminadas.....	73
V. Resultados Computacionais e Conclusões.....	75
V.1. Introdução.....	75
V.2. Características dos Problemas Tratados.....	76
V.3. Análises Preliminares.....	78
V.3.1. Sensibilidade à Solução de Partida (atributo custo).....	79
V.3.2. Determinação da Faixa de Parâmetros (atributo aresta).....	80
V.3.3. Escolha do Atributo.....	90
V.4. Comportamento do Algoritmo.....	92
V.4.1. O Processo de Busca.....	92

V.4.2. Até o Primeiro Mínimo Local.....	97
V.4.3. Após o Primeiro Mínimo Local - Fase Tabu.....	98
V.4.4. Ciclagem.....	100
V.5. Contribuição da Busca Tabu.....	101
V.6. Desempenho do Algoritmo com Relação a Outros Métodos Heurísticos..	103
V.7. Redução do Número de Veículos.....	105
V.8. Resultados Gerais.....	107
V.9. Perspectivas.....	110
Referências Bibliográficas.....	111

CAPÍTULO I

O PROBLEMA DE ROTEAMENTO DE VEÍCULOS

I.1 - INTRODUÇÃO

A alocação e roteamento de veículos com vistas à distribuição de bens ou serviços é um elemento essencial de qualquer sistema logístico. Em função do alto custo envolvido, existe uma necessidade cada vez maior de encontrar maneiras mais eficientes e econômicas de realizar estas operações. A complexidade das operações logísticas que são geradas dentro das sociedades modernas tem impulsionado o desenvolvimento científico de métodos matemáticos e sistemas computacionais de apoio à tomada de decisões. Soluções que no passado eram selecionadas com facilidade e segurança por profissionais experientes, hoje necessitam do apoio de sistemas específicos.

O reconhecimento da eficácia desses sistemas computacionais quando aplicados a casos reais, tem atraído a atenção tanto dos responsáveis pela gestão de sistemas logísticos como daqueles que desenvolvem tais ferramentas. Assim, nas duas últimas décadas, a ciência da logística tem mostrado um impressionante desenvolvimento. As decisões afetas especificamente à alocação e roteamento de veículos foram englobadas por uma área de pesquisa conhecida como Problemas de Roteamento de Veículos (PRV).

Nesta tese é focado um dos problemas de roteamento de veículos, cujo objetivo é a determinação da configuração de rotas que veículos devem percorrer afim de satisfazer a demanda de clientes por um conjunto de produtos. Em geral os problemas reais impõem restrições sobre veículos (capacidade, tipo, horários, tamanho de frota) e sobre clientes (demanda, horários, preferências), que devem ser atendidas pelos sistemas computacionais. Além disso, esses sistemas devem buscar soluções factíveis que otimizem um (ou mais)

critérios considerados relevantes pelo usuário; por exemplo, minimização de custos, maximização da satisfação dos clientes, etc. Assim sendo, existem inúmeras situações que caracterizam problemas de roteamento de veículos, cada qual definida por conjuntos de restrições e objetivos.

Em sua forma mais simples, o PRV assume a existência de uma única garagem, onde todas as rotas devem começar e terminar dentro de um horizonte de tempo, normalmente definido pela jornada de trabalho. Considera-se uma frota homogênea, ou seja, todos os veículos possuem as mesmas características no que diz respeito ao tipo e à capacidade. Cada veículo estabelece uma única rota, e os clientes a ele designados devem ter demandas tais que a capacidade total não seja excedida. Assume-se que informações relativas aos clientes, isto é, suas posições geográficas e demandas, são conhecidas. Exige-se que todos sejam atendidos uma única vez e que suas demandas sejam totalmente satisfeitas.

Na maioria dos problemas encontrados na prática, entretanto, existem modificações e restrições adicionais que o PRV básico ignora. Por exemplo:

1. Cada veículo pode operar em mais de uma rota, desde que o tempo total gasto nestas rotas seja menor que um dado tempo T .
2. O problema pode envolver operações de entrega e coleta em várias combinações.
3. São consideradas outras atividades que consomem tempo, que não o transporte; em geral, o descarregamento e carregamento de veículos ou a prestação de serviço propriamente dita.

4. Assumem-se *janelas de tempo* para veículos, isto é, períodos determinados no horizonte de tempo em que a atividade é realizada. Geralmente em sua especificação são considerados horários de almoço e paradas para descanso.

5. Assumem-se *janelas de tempo* para clientes. São definidos intervalos dentro dos quais os clientes podem ser atendidos, ou seja, o serviço ou descarregamento de bens deve ser iniciado e/ou completado dentro dos limites do intervalo.

6. Estabelecem-se restrições com respeito à estrutura das rotas, como distância máxima total por rota, preferência por formatos de rotas, preferência por rotas com pontos de demanda próximos uns dos outros, etc.

Estas restrições são facilmente incorporadas ao PRV básico. Seguem-se outras, cuja inclusão exige alterações mais profundas em sua estrutura:

7. Existência de mais de uma garagem

Se cada garagem possui sua própria frota e a cada frota for designada uma área geográfica de atuação, então tem-se múltiplos problemas independentes a resolver. Se por outro lado, for permitido a cada veículo sair de uma garagem, realizar sua rota, terminar em outra garagem e a partir daí, iniciar nova rota, então tratam-se de problemas interdependentes e cada garagem não pode ser tratada isoladamente.

8. Nível de serviço

O tempo requerido para satisfação das necessidades do cliente é uma medida de nível de serviço, e a configuração das rotas para um dado período é, por definição, uma

aproximação ou uma ordem arbitrariamente imposta. Aproximações são normalmente utilizadas em atividades de distribuição em que se caracterizam visitas regulares a clientes fixos com demandas típicas. Caso contrário, a ordem de atendimento pode ser determinada pela ordem de recebimento de pedidos. É possível também designar prioridades para clientes de acordo com a urgência de cada um.

9. Múltiplos tipos de mercadorias

A designação de clientes e, por conseguinte, a configuração das rotas deve considerar a compatibilidade entre quantidades de tipos de mercadoria requeridos e a capacidade do veículo. Ou seja, tem-se uma combinação de problema de roteamento e problema da mochila.

10. Múltiplos tipos de veículos

Esta situação está geralmente associada à distribuição de múltiplos tipos de mercadorias, especificamente nos casos em que alguns destes tipos requerem transporte especial (por exemplo, compartimentos refrigerados) e outros não. A complexidade implícita na designação de clientes é, portanto, acrescida da necessidade de observar restrições não só com respeito à capacidade mas também ao tipo de veículo.

11. Informações acerca do número de clientes, suas demandas e localizações, dadas de forma probabilística.

Em inúmeras aplicações, o problema de planejamento de médio e longo prazo para aquisição de uma frota de veículos tem de ser resolvido a partir de informações vagas, na melhor das hipóteses probabilísticas, sobre o que será requerido destes veículos.

Em relação à escolha de um critério para orientar a definição das rotas dos veículos, a maioria dos problemas de interesse busca representá-los como uma *combinação* entre a *maximização* da soma das prioridades de clientes que podem ser atendidos por uma frota V de veículos e a *minimização* do número total de veículos (custo fixo) e/ou distância (ou tempo) total percorrida (custo variável). Seguem-se algumas formulações:

1. Minimizar o número total de veículos utilizados, dado que todos os clientes sejam roteados. Se neste caso o horizonte de tempo é desconsiderado, tem-se um problema com características similares ao problema da mochila generalizado.
2. Minimizar a distância (ou tempo) total percorrida, dado que todos os clientes sejam roteados. Se a frota consistir de um único veículo e se o horizonte de tempo for desconsiderado, tem-se o problema do Caixeiro Viajante.
3. Minimizar a distância (ou tempo) total percorrida, para o menor número de veículos compatível com os requerimentos dos clientes.
4. Se veículos com diferentes capacidades podem ser escolhidos, encontrar a combinação ótima de veículos, tal que o custo total seja minimizado.
5. Para um dado conjunto de clientes cujas demandas e posições geográficas não são determinísticas, mas dadas de forma probabilística, encontrar a solução dos problemas

1 e 4.

Nas situações em que o número de veículos é fixado previamente, existe a possibilidade de, para as restrições e objetivos estabelecidos, não existir solução factível.

Nestes casos, existem naturalmente duas alternativas:

p1. Utilizar mais veículos

e/ou

p2. adiar o serviço para alguns clientes além do nível de serviço estabelecido, ou transferí-lo para o próximo período.

Isto sugere a redefinição do problema, cujo objetivo, desta vez pode ser o de minimizar:

s1. O número de veículos extras

e/ou

s2. O número (ou soma ponderada) de clientes não servidos no período atual

e/ou

s3. A distância total (ou tempo) percorrida.

Nestes casos, a função critério do PRV pode ser uma combinação linear de (s1), (s2) e (s3).

I.2 MÉTODOS EXATOS

A classificação de algoritmos em termos de sua complexidade é feita utilizando-se o conceito de limitação polinomial. Algoritmos polinomiais são aqueles em que o número de operações elementares necessário para a obtenção da solução ótima de um dado problema, é limitado, no pior caso, por uma função polinomial do tamanho do problema. Problemas resolvidos por algoritmos polinomiais pertencem à Classe-P e em função de tais características, sua otimalidade é obtida com eficiência. Por outro lado, problemas para os quais não se conhecem algoritmos polinomiais que os resolvam são classificados como *NP-completos* ou "*NP-hard*". São considerados complexos e de difícil tratamento.

Um número substancial de métodos exatos foi elaborado com vistas à resolução de problemas de roteamento. Levantamentos detalhados a este respeito podem ser encontrados em Christofides [5], Christofides et al [7], Watson-Gandy e Foulds [44] e Bodin et. al [2]. Entretanto, a complexidade do problema torna tais métodos aplicáveis apenas para pequenos problemas. O PRV é um problema NP-completo; especificamente, os algoritmos exatos disponíveis requerem um número de passos computacionais que cresce como uma função exponencial do número de clientes que precisam ser visitados. A consideração de restrições torna sua resolução ainda mais difícil; a existência de janelas de tempo, por exemplo, os torna problemas "NP-hard". Savelsbergh [35] mostrou que neste caso, mesmo a obtenção de uma solução factível quando o número de veículos é fixado, é um problema NP-completo. Não é surpresa, portanto, que o maior VRP resolvido

otimamente envolvesse apenas 53 clientes [5].

I.3 - MÉTODOS APROXIMADOS

Em vista do esforço computacional requerido, e até por razões de viabilidade, problemas de maior porte são tratados por métodos aproximados, os quais ao longo do tempo têm sido elaborados e modificados com vistas a um crescente aumento de eficiência.

Com o objetivo de apresentar as principais heurísticas já propostas, será aqui adotada a seguinte classificação:

1. Métodos de Construção de Rotas
2. Métodos de Duas Fases
3. Métodos de Otimização Incompleta
4. Métodos de Melhoria de Rotas

I.3.1 - Métodos de Construção de Rotas

Muitas das idéias que levaram à elaboração da maioria dos métodos de construção de rotas têm sua origem no trabalho de Clarke e Wright [8]. O método de Clarke e Wright assume, como solução de partida, n rotas distintas constituídas por um único cliente (tal configuração é considerada infactível pois é geralmente impraticável designar um veículo

da frota para cada cliente). Isto implica que a distância percorrida ao se servir um dado cliente i é dada por $d(0,i) + d(i,0)$, onde 0 é o índice da garagem e $d(.)$ a distância medida por uma métrica definida. O algoritmo avalia, então, a economia em distância que resulta de todas as possíveis combinações de rotas. Por exemplo, uma rota que combina o nó i e o nó j partiria da garagem para o nó i , indo diretamente para o nó j , seguido de um retorno à garagem. A economia resultante desta combinação é dada por:

$$s(i,j) = d(0,i) + d(0,j) - d(i,j)$$

Esquemáticamente:

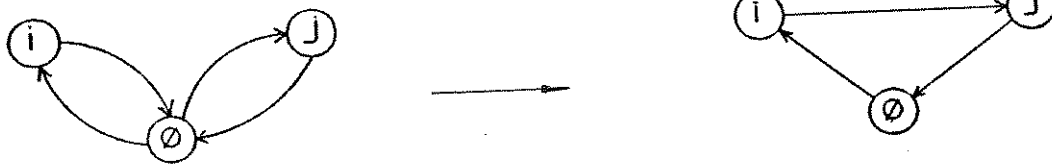


Figura I.1

Como a expressão indica, a combinação de nós em uma única rota elimina dois arcos que conectam respectivamente os nós i e j à garagem; entretanto, é necessário adicionar o arco que conecta, por sua vez, os nós i e j . Uma dada combinação de rotas é passível de realização se ela resultar em economia positiva em distância. A este respeito, Clarke e Wright sugeriram uma estratégia gulosa, isto é, a cada estágio é escolhida a combinação factível que provoca a maior economia.

A despeito de sua simplicidade, o método mostra-se eficiente, ou seja, rapidamente se encontra uma solução com características de ótimo local. Entretanto, a qualidade das

soluções diminuí progressivamente com o tamanho do problema e com a inclusão de restrições. Inúmeras variações foram propostas com vistas à superação das principais limitações do método básico. Gaskell [15] e Yellow [47] modificaram o cálculo de economia $s(i,j)$, introduzindo um parâmetro cuja variação produz rotas com diferentes formatos. Gaskell também sugeriu incluir distâncias médias dos nós às garagens. Golden, Magnanti e Nguyen [23] consideraram o cálculo de economia apenas entre nós que estejam próximos uns dos outros. Mole e Jameson [31] utilizaram um critério mais elaborado que inclui dois parâmetros exógenos e incorpora um processo de refinamento de soluções. Finalmente, muitos autores têm desenvolvido estratégias "look ahead" que buscam avaliar as consequências de escolher uma economia específica antes de efetivamente aceitá-la [25,28,40,41].

Além do critério de expansão de rotas, algoritmos de construção podem ser classificados de acordo com a ordem desta expansão, ou seja, se as rotas são construídas paralela ou sequencialmente. Algoritmos sequenciais constroem uma rota por vez até que todos os clientes sejam inseridos. Algoritmos paralelos constroem um número de rotas simultaneamente, por sua vez, gerado livremente ou pré-fixado. Versões paralelas da heurística de Clarke e Wright são encontradas em Christofides et al [7] e Christofides [5]. M. Solomon [36] utilizou uma versão paralela de uma variante da heurística de Clarke e Wright, especificamente o critério de Gaskell e Yellow.

I.3.2 - Métodos de Duas Fases

Nos métodos de duas fases, as operações de designação de clientes a veículos e o roteamento propriamente dito não são feitos simultaneamente, mas sim em duas fases distintas. Na primeira etapa, são aplicados critérios convenientes para a formação de grupos de clientes, cujas restrições podem ser satisfeitas por um único veículo. A definição dos grupos reduz o problema original a m problemas menores e independentes (m Caixeiros Viajantes). A segunda fase do método trata então de construir uma rota para cada grupo de clientes através de algum algoritmo adequado.

Diferentes técnicas para a fase 1 foram propostas. Tyagi [42] apresentou um método que agrupa pontos de demanda baseado no conceito do vizinho mais perto. Ou seja, clientes são adicionados sequencialmente, cada adição correspondendo ao cliente mais próximo do último a ser adicionado. Christofides et al. [7], utilizaram uma regra similar, porém em ambas as etapas incluíram um algoritmo de refinamento de soluções. O algoritmo da "varredura" de Gillett e Miller [16] ordena os clientes de acordo com o desvio angular com relação a uma direção arbitrária, centrada na garagem. A seguir, uma varredura radial os designa aos veículos até que alguma das restrições seja violada. Neste ponto, uma nova rota é iniciada e o processo se repete até que todos os clientes sejam roteados.

I.3.3 - Métodos de Otimização Incompleta

Alguns métodos de otimização incompleta poderiam ser propriamente classificados como métodos de duas fases. Entretanto, em função de sua forte conexão com formulações de programação inteira do problema, destacam-se como uma categoria separada.

Christofides et al. [7] desenvolveram procedimentos heurísticos para o PRV que são modificações de algoritmos exatos de busca em árvore. Fisher e Jaikumar [11,13] trabalharam, por sua vez, com a resolução de problemas de designação generalizada e subproblemas de Caixeiro Viajante. Seu método heurístico é um desenvolvimento da técnica de decomposição de Benders descrita em artigo anterior [12]. Cullen et. al [9] partiram de uma formulação de "set partitioning" do PRV como base para uma heurística que alterna entre subproblemas de designação e localização, sempre que melhorias adicionais forem possíveis.

Quase todos os métodos heurísticos sugeridos para a resolução do PRV são construtivos, na medida que, em um dado estágio, existe pelo menos uma rota incompleta que será estendida nos estágios posteriores até tornar-se uma rota completa. Entretanto, uma vez que um método construtivo é usado para produzir uma solução para o PRV, esta solução pode ser melhorada pela aplicação de técnicas de otimização local que mantêm a factibilidade da solução. De fato, métodos de refinamento de soluções têm sido notadamente utilizados, o que sugeriu a seguinte classificação:

I.3.4 - Métodos de Melhoria de Rotas

Os métodos de melhoria de rotas podem ser vistos como uma aplicação específica de uma classe de procedimentos heurísticos utilizados na resolução de diversos problemas de otimização, particularmente os combinatórios - as heurísticas "hill climbing". Estas heurísticas são caracterizadas por duas etapas básicas:

- 1) Geração de uma solução inicial.
- 2) Utilização de mecanismos que alterem a solução inicial, de forma a produzir melhores resultados.

Após a obtenção da solução de partida, são promovidas transformações factíveis em sua configuração, através da aplicação sistemática de operações previamente especificadas. Estas aplicações visam essencialmente a melhoria da função objetivo a partir da solução inicial, até que se atinja um mínimo local, ou seja, até o estágio em que nenhuma melhoria adicional é possível.

No caso específico de métodos de melhoria de rotas, a solução inicial é geralmente obtida através de heurísticas clássicas de construção de rotas, duas fases ou mesmo otimização incompleta. Neste ponto reside o grande atrativo destes métodos, qual seja, a possibilidade de obterem soluções que superem de forma significativa os resultados dos procedimentos clássicos.

Com este objetivo, vários mecanismos de melhoria de rotas têm sido propostos e

dentre estes, possivelmente o mais conhecido seja a heurística de troca de arestas r -OPT [29,30]. Em tais procedimentos são realizadas operações de substituição de r arestas não adjacentes, ou seja, r ligações (não adjacentes) definidas entre pares de nós, que fazem parte da solução corrente. Para cada rota gerada na etapa anterior, são computadas as aplicações factíveis destas operações (no caso definindo-se os grupos de arestas eliminadas e adicionadas), e escolhida aquela que provoca o maior ganho na função objetivo.

Esquemáticamente:



Figura I.2 - Substituição das arestas A - D, B - E e C - F por A - B, C - D e E - F ($r=3$)

Este procedimento é repetido até que nenhuma melhoria seja possível pela substituição de r arestas; diz-se, então, que a rota resultante é r -ótima. Para n clientes, considerações de todas as possíveis trocas envolvem um esforço computacional da ordem de n^r , um fato que na prática tem limitado o valor de r a 2 ou 3. Face a esta limitação, Christofides e Eilon [6] sugeriram um algoritmo onde operações 2-OPT e 3-OPT são realizadas alternadamente. Uma variação promissora conhecida como Or-OPT [33] examina apenas trocas de arestas que inserem pequenas cadeias de nós em novas

localizações da rota. Stewart [38] menciona que Or-OPT tem uma performance tão boa quando o 3-OPT com apenas 5% das comparações.

Seguindo esta mesma orientação gulosa, Waters [43] propôs para o PRV uma combinação de procedimentos de melhoria de rotas que envolve a remoção e posterior inserção de clientes de uma parte da solução para outra. A idéia central é que pela remoção, espaços factíveis sejam criados nas rotas, nos quais os clientes removidos possam ser inseridos.

Wren e Holliday [46] sugeriram combinações mais complexas que incluem não só a remoção e inserção de clientes como também procedimentos de redução do número de veículos, através de tentativas de redistribuição generalizada e utilização de um mesmo veículo em rotas distintas.

M. Dror e L. Levy [10] propuseram três métodos heurísticos de melhoria de rotas com vistas à resolução de problemas de roteamento de inventário. Estas heurísticas promovem operações de troca de nós, cuja aplicação, em contraste com métodos de troca de arestas, não está limitada a sistemas de uma única rota.

I.4 - LIMITAÇÕES DOS MÉTODOS "HILL CLIMBING"

Não obstante o excelente desempenho que várias das heurísticas "hill climbing" apresentam, em se tratando de problemas combinatórios, ainda persistem graves limitações. Em virtude da não convexidade do problema, o ótimo local obtido pode não ser o ótimo

global. Paralelamente, deduz-se que sob tal orientação, a qualidade da solução final depende drasticamente da solução de partida, o que pode comprometer a robustez do método.

Em outras palavras, heurísticas "hill climbing" são caracterizadas tanto pela habilidade em atingir ótimos locais como pela inabilidade de sair deles. A otimalidade local é, portanto, um aspecto crucial que deve ser trabalhado e se possível transcendido pelas abordagens heurísticas.

Com esta finalidade várias estratégias têm sido propostas. A "aleatorização controlada", por exemplo, assume duas formas. A primeira delas é a conhecida "reinicialização aleatória", ou seja, uma nova solução de partida é obtida através de um processo de geração que inclui um elemento aleatório. A segunda versão clássica é a "perturbação aleatória", um procedimento que ao invés de reinicializar o processo, gera periodicamente uma série de movimentos aleatórios que resultam na alteração dos passos que a busca normalmente seguiria.

"Simulated Annealing" [3,27], um refinamento da estratégia de perturbação aleatória, tem sido anunciada como uma nova e poderosa metodologia para a resolução de problemas combinatórios. Seu nome resulta do intento em relacionar a abordagem ao processo físico de "annealing", isto é, uma técnica de redução da temperatura de um material ao seu nível mais baixo de energia. De acordo com a analogia à resolução de problemas combinatórios, foi postulado que o caminho para um estado ótimo deve começar longe da otimalidade. Neste estado inicial, movimentos de piora da função objetivo são aceitos com uma probabilidade relativamente alta, que gradualmente diminui com o tempo. No entanto, tal

analogia não é de certo óbvia e nem sempre justificada. Em particular, a definição de uma função objetivo que pode ser comparada a uma função energia, pode causar dificuldades.

Uma outra maneira de contornar limitações de otimalidade local é pela aplicação da estratégia de Busca Tabu [17,19,20,21]. Em contraste às técnicas anteriores, a Busca Tabu não utiliza recursos de aleatorização, nem induz o progresso lento rumo a um mínimo local. Ela promove no procedimento heurístico a *continuação das explorações* em busca de melhores soluções, sem que este se confunda com a ausência de movimentos de melhoria e sem que retorne ao ótimo local de onde emergiu.

O objetivo deste trabalho é o estudo da aplicação da Busca Tabu a contextos de Roteamento de Veículos. Isso foi feito a partir da assimilação dos fundamentos teóricos, seguida de implementação computacional, análise dos resultados e proposição de novos conceitos.

CAPÍTULO II

A BUSCA TABU : FUNDAMENTOS

II.1 - INTRODUÇÃO

A Busca Tabu [17,19,20,21,22] é uma estratégia utilizada na resolução de problemas de otimização combinatória, cuja aplicação a heurísticas "hill-climbing" viabiliza a superação das limitações de otimalidade local. Originada no final dos anos 70 como um método de resolução de problemas reais de "covering", graças a sua adaptabilidade foi posteriormente aplicada a outros contextos que vão desde "scheduling" até projeto de circuitos integrados e planejamento espacial ("layout"). Estudos computacionais envolvendo problemas combinatórios clássicos como o Caixeiro Viajante [20,21] e "Graph Coloring" [24], entre outros, têm evidenciado a habilidade desta estratégia em produzir soluções de melhor qualidade com um menor esforço computacional, geralmente dominando métodos alternativos.

Neste capítulo são introduzidos os conceitos fundamentais e elaborações da estratégia de Busca Tabu.

II.2 - DEFINIÇÕES E NOMENCLATURA

Conforme apresentado no capítulo anterior, heurísticas "hill climbing" se notabilizam pela utilização de mecanismos que alteram a configuração de uma solução completa. Um importante aspecto deste mecanismo é que com a promoção destas alterações, institui-se um processo iterativo de busca. Ou seja, é gerada uma sequência de soluções factíveis através da aplicação sistemática de operações, ou num sentido mais amplo, de *movimentos*, que provocam a maior melhoria na função objetivo. No caso específico dos métodos de melhoria de rotas apresentados, estes movimentos são caracterizados pela adição e eliminação de arestas (r-OPT (1965,1973), Or-OPT (1976)) ou pela remoção e inserção de nós (heurísticas de Wren e Holliday (1972), M. Dror e L. Levy (1986) e Waters (1987)).

O movimento representa, portanto, a transição de uma solução para outra, e conforme será visto neste capítulo, a forma como movimentos são descritos, tem um papel de extrema importância na estratégia de Busca Tabu. Por esta razão, introduziremos algumas formalizações acerca deste conceito.

Seja X a região factível de soluções, definida pelas restrições impostas ao sistema. Seja S o conjunto total de movimentos cuja aplicação a estas soluções factíveis produz soluções também factíveis. O processo de busca é estabelecido pela aplicação de movimentos $s \in S$ a soluções $x \in X$, gerando soluções $s(x) \in X$. Define-se $X(s)$ como o conjunto de soluções $x \in X$ às quais pode-se aplicar o movimento $s \in S$. De igual forma,

define-se $S(x)$ como o subconjunto de S cujos movimentos podem ser aplicados a uma dada solução factível x .

Esquemáticamente:

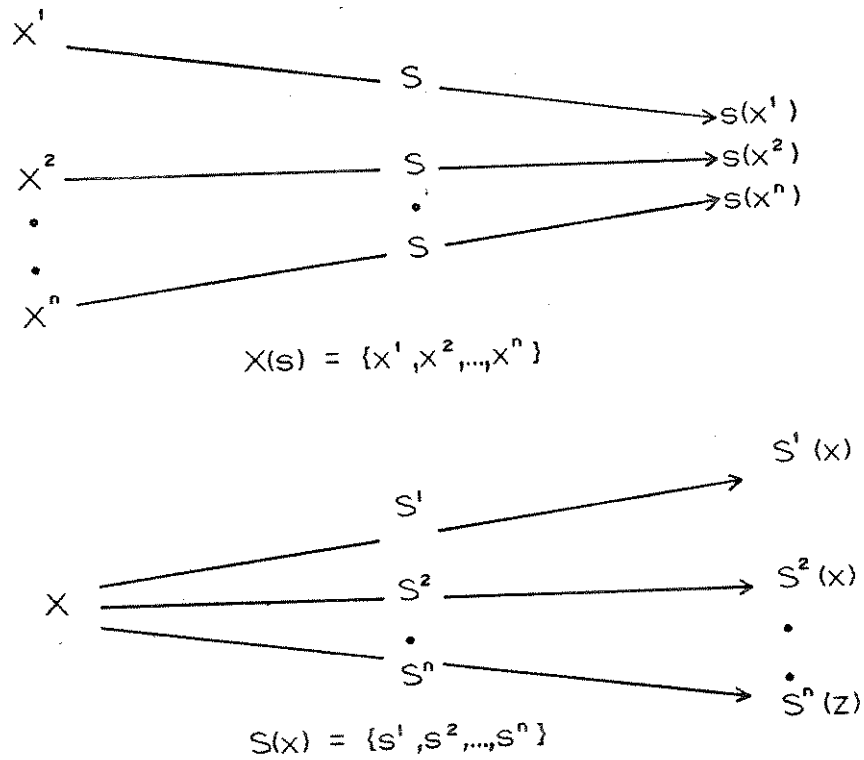


Figura II.1

O movimento s consiste, portanto, de um mapeamento definido em um conjunto $X(s)$ de X .

II.3 - A BUSCA TABU E SEUS ELEMENTOS BÁSICOS

Consideremos que partindo de uma solução inicial, um procedimento heurístico hipotético realize movimentos que visem a melhoria da função objetivo, até que se atinja um mínimo local. Se desejarmos a continuidade do processo de busca, o primeiro passo seria naturalmente realizar um movimento de piora, digamos o menos pior, de forma a se emergir deste mínimo local. Se a partir daí, reavaliarmos os movimentos disponíveis e efetivarmos o que promove a maior melhoria na função objetivo, existe a possibilidade de que tal movimento seja exatamente o reverso do previamente realizado, o que implicaria no retorno ao mínimo. A avaliação seguinte indicaria novamente o primeiro movimento e assim, no processo de busca, se seguiria uma sequência de um único movimento e sua reversão. Esta situação ilustra o fenômeno da ciclagem, capaz de comprometer de forma irremediável a exploração em busca de melhores soluções. Necessita-se, portanto, de um elemento restritivo na busca, de forma a impedir a efetivação não só de movimentos de reversão mas também de todo e qualquer movimento que gere soluções-tentativa anteriormente encontradas.

Entretanto, se considerarmos a evolução do processo de busca a partir de uma dada solução como uma soma de alterações em sua configuração, é de se supor que com sucessivas alterações, a possibilidade de um retorno a esta solução diminua. Isto sugere que a restrição feita previamente a certos movimentos possa ser relaxada ao longo das iterações. Mais do que uma mera possibilidade, tal relaxação é desejável na medida em que estes movimentos, não oferecendo risco de ciclagem e podendo ser novamente utilizados, dão

flexibilidade ao método, permitindo-lhe um espectro maior de opções.

Os aspectos acima referidos constituem os dois elementos básicos da Busca Tabu; a restrição da busca pela classificação de certos movimentos como proibidos (ou seja, como tabu) e sua liberação através de uma função de curto prazo, que provoca um "esquecimento" estratégico.

Sua implementação consiste na criação de um subconjunto T de S cujos elementos são chamados *movimentos tabu*. Os elementos de T são determinados por uma função não-markoviana que utiliza informação histórica do processo de busca até t iterações passadas, onde t pode ser fixo ou variável. De acordo com a natureza do problema, T pode ser representado por uma *lista de itens* ou por um conjunto de condições tabu (por exemplo, inequações lineares ou relações lógicas).

Para T apropriadamente determinado e uma função de avaliação ÓTIMO, o processo pode ser descrito como se segue:

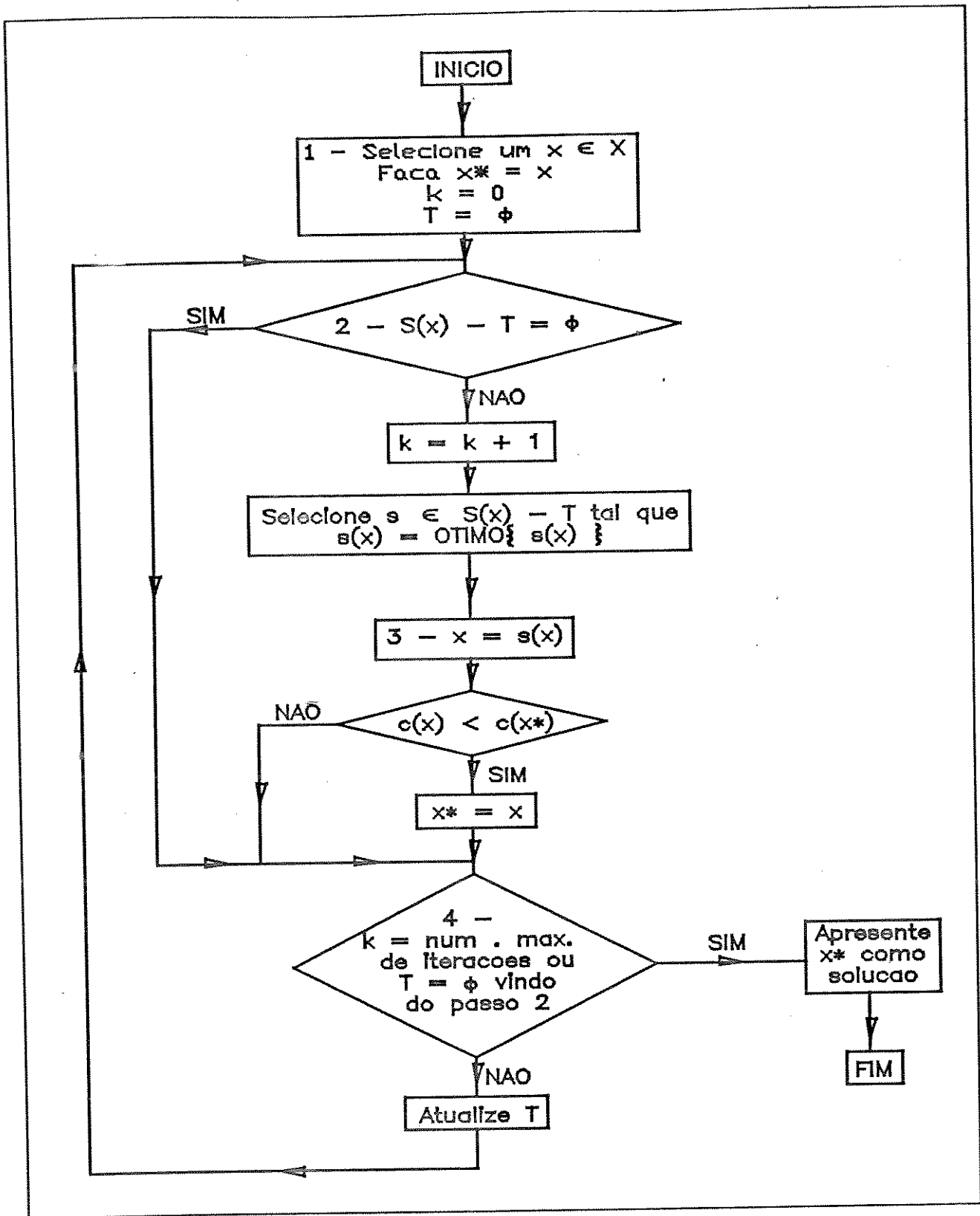


Figura II.2

Alguns aspectos desta versão devem ser comentados:

1. É evidente que sendo T o agente de restrição da busca, sua composição e a forma como é atualizado têm um papel fundamental na geração de soluções.

2. Não há qualquer referência à condição de otimalidade local, exceto implicitamente onde o ótimo local pode vir a atualizar a melhor solução previamente encontrada.

3. A função ÓTIMO, ao contrário das funções de avaliação de heurísticas "hill climbing", escolhe não o movimento que promove a maior melhoria na função objetivo, mas sim o que chamaremos de "melhor" movimento, em função das novas condições presentes. Uma escolha natural para ÓTIMO seria a de selecionar $s_k(x)$ tal que:

$$c(s_k(x)) = \text{Min} \{ c(s(x)) \mid s \in S(x) - T \}$$

Nos casos em que T e S podem ser expressos como um requerimento que satisfaça um conjunto de inequações (tais como limitantes nas variáveis), a solução $s(x)$ pode ser obtida pela resolução de um problema de otimização linear, como os encontrados em aplicações de programação inteira que utilizam métodos de programação linear como subrotina.

Assim, a cada execução do passo 2 será escolhido um movimento **não tabu** que forneça a maior melhoria na função objetivo, ou no caso de falta de tais movimentos, o que promova a menor piora. Nos casos em que $S(x) - T$ é grande e definido de forma itemizada, pode parecer adequado que ÓTIMO possua mecanismos para a redução desta

região. Aqui não nos referimos à minimização do esforço computacional no processo de identificação do $c(s(x))$ mínimo, mas às razões estratégicas de alguns métodos (e.g. "Simulated Annealing" [3,27]) que buscam promover um progresso lento rumo a um mínimo local sob a premissa de que tal procedimento, apropriadamente regulado, fará o ótimo local mais parecido com o ótimo global.

Neste aspecto a Busca Tabu tem uma diretiva mais agressiva. Isso se deve a duas considerações. Primeiro, a de que vários problemas de otimização podem ser resolvidos otimamente pela efetivação do melhor movimento a cada passo. Como exemplo pode-se citar problemas de fluxo de redes de custo mínimo resolvidos através de movimentos baseados em sucessivos caminhos mínimos. Se em tais procedimentos escolhermos movimentos que não os melhores, o resultado pode ser a geração de soluções inferiores. Por outro lado, outros métodos isentos de tal exigência não são prejudicados e muitas vezes, têm seu processo de busca acelerado pela escolha das melhores avaliações.

A outra consideração é a de que a otimalidade local não mais representa uma barreira para a continuidade do processo e, portanto, não há razões para manter a busca em regiões menos atraentes. Pelo contrário, deve-se estimular prioritariamente a exploração de novas regiões onde existe efetivamente a promessa de melhores soluções.

II.4 - FORMAS DE LISTAS TABU

Conforme dito anteriormente, uma preocupação básica da Busca Tabu é proibir a reversão de quaisquer movimentos feitos nas últimas t iterações. Segundo esta orientação,

podemos definir a lista T da seguinte maneira:

$$(1) \quad T = \{ s^{-1} : s = s_h \text{ para } h > k-t \}$$

onde k é o índice da iteração e s^{-1} o inverso do movimento (ou seja, $s^{-1}(s(x)) = x$).

Entretanto, o impedimento de movimentos de reversão pode não ser suficiente para prevenção de ciclagem. Considerando-se que geralmente deseja-se impedir o retorno a um *estado-solução* previamente visitado, onde o melhor movimento não tabu disponível é o mesmo que antes, faz-se necessário incluir o movimento em si. Uma representação conveniente seria:

$$(2) \quad T = T_1 \cup T_2$$

onde $T_1 = \{ s^{-1}_h : h > k-t_1 \}$ e $T_2 = \{ s_h : h > k-t_2 \}$

Em ambas as definições está inserido o procedimento de atualização mencionado no passo 3 da versão apresentada. Consiste em fazer $T = T - s^{-1}_{k-t_1} + s^{-1}_k$ (e $T = T - s_{k-t_2} - s_k$), onde os sinais - e + representam as operações de eliminação e adição de elementos a um conjunto. Por convenção, quando $k \leq t_1$ (e $k \leq t_2$), a referência a $s^{-1}_{k-t_1}$ (e s_{k-t_2}) é desconsiderada.

A forma de atualização indicada se baseia na premissa de que a probabilidade de ciclagem é inversamente proporcional à distância da solução-tentativa x àquela solução

prévia. Se a distância é medida em termos do número de movimentos feitos desde que a solução prévia foi visitada e considerando-se que nenhum movimento de reversão seja permitido, então uma lista T devidamente caracterizada e suficientemente dimensionada deve ser capaz de impedir ciclagens. Isso não significa que o tamanho de T deva ser grande já que desta forma reduziríamos o espectro de escolhas e conseqüentemente, o acesso a soluções promissoras. Faz-se necessário um compromisso entre estes dois aspectos, de maneira a determinar uma faixa de valores de t que permita ao método encontrar soluções desejáveis ao mesmo tempo em que previne a ciclagem.

As formas apresentadas de representação de T são raramente utilizadas na prática. Nos contextos em que considerações de dominância são relevantes, a prevenção de s^{-1} deve também implicar na prevenção de outros movimentos dominados por s^{-1} . Além disso, por razões de economia-de memória e facilidade de processamento, é geralmente desejável representar os movimentos apenas por uma faixa parcial de seus *atributos*, ou seja, seus elementos descritivos.

Como exemplo, considere a resolução de um Problema de Programação Inteira, na qual a transição entre soluções seja efetuada pela incrementação ou decrementação unitária do valor de uma dada variável. Uma forma de representação do movimento adotaria como atributo, o par ordenado (v_1, v_2) , onde v_1 é o vetor-solução correspondente à solução π ; e v_2 o vetor-solução correspondente à solução $s(x)$. A lista de pares (v_2, v_1) seria utilizada com vistas à prevenção de reversões de movimento já que estes pares caracterizam s^{-1} . A representação do movimento atinge seu nível restritivo mais baixo, na

medida em que cada elemento da lista define a reversão de um único movimento.

Uma representação mais simples de movimento adotaria como atributos os incrementos ou decrementos associados às variáveis cujos valores foram efetivamente alterados com a aplicação. Neste caso, movimentos seriam descritos por pares ordenados (i, e_i) onde i é o índice da variável x_i e e_i o incremento ou decremento aplicado à x_i ($e_i \in \{-1, 1\}$). A lista tabu pode ser então definida pelos pares $(i, -e_i)$. Supondo que uma variável x_i tenha seu valor atual (in)decrementado na k -ésima iteração, então **todos** os movimentos que impliquem na sua (de)incrementação serão classificados como tabu nas próximas t iterações.

Sob tal representação, cada elemento de T não estará associado a um único movimento, mas potencialmente poderá englobar conjuntos C_h de movimentos que compartilham os mesmos atributos. A lista T é então mais convenientemente representada da seguinte forma:

$$(3) \quad T = \cup C_h : h > k-t \text{ (onde } s_h^{-1} \in C_h)$$

Um ponto que merece ênfase diz respeito à atualização de T nas situações em que $S(x) \cap T$ é vazio, isto é, quando todos os movimentos disponíveis são tabu. Seguindo o mesmo critério de prioridade, é retirado da lista o menor nº de elementos, na sequência dos mais antigos para os mais novos. Isso permitirá que algum(ns) movimento(s) obtenha(m) o status não-tabu e a busca tenha continuidade.

II.5 - NÍVEIS DE ASPIRAÇÃO

A forma de representação (3) de T possui, conforme apresentado, características bastante atraentes, tanto a nível computacional quanto estratégico. Entretanto, ao proibir conjuntos de movimentos em função de certos atributos, são desprezadas opções que poderiam vir a fornecer estados-solução até então não encontrados.

Com o objetivo de superar esta limitação e promover ganhos em flexibilidade na escolha de movimentos é que se torna necessário um mecanismo capaz de reconhecer situações em que é possível a superação do status tabu. A questão é como defini-lo, mantendo-se a garantia da não ocorrência de ciclagem. A este respeito, vejamos com maior detalhe algumas formas de prevenção.

A maneira mais natural de se impedir a ciclagem consiste numa simples análise da solução resultante da aplicação do movimento. Caso tenha sido gerada anteriormente, evitar sua efetivação certamente evitará a ciclagem. Trata-se, portanto, de um impedimento direto e explícito de reincidência de uma dada solução. No entanto, seria necessário manter todas as soluções encontradas até o momento e verificá-las, o que implicaria em maior utilização de memória e esforço de processamento. Isso desestimula a utilização desta representação para a maioria dos problemas. Requer-se então um critério mais adequado às limitações computacionais, e se possível, que se aproxime da óbvia eficiência desta primeira estratégia.

Para este fim, consideremos mais detalhadamente um novo conceito de movimento cuja identificação depende não só de s mas também do próprio x ; a ele chamaremos de

movimento de solução específica. Sejam os seguintes níveis de estratégia para prevenção de ciclagem, os quais envolvem proibir o movimento de solução específica $x \rightarrow s(x)$ caso:

- a) o movimento s tenha sido aplicado a x previamente.
- b) o movimento s^{-1} tenha sido aplicado a $s(x)$ previamente.

No critério (a) reconhece-se a intenção de impedir a repetição do movimento, o que na prática não funciona bem. Sob esta orientação, por exemplo, é possível ocorrer a reversão de um movimento tão logo ele seja feito.

O critério (b), por outro lado, visa atingir a reversão do movimento, o que por sua vez, é compatível com a premissa assumida para prevenção de ciclagem e corresponde à diretriz básica da Busca Tabu. Sob o respaldo desta condição, permite-se que um movimento tabu $x \rightarrow s(x)$ seja efetuado caso $s(x) \rightarrow x$ não tenha ocorrido previamente, com boa garantia de prevenção de ciclagem. A utilização da condição (a) juntamente com a condição (b) funciona como uma aproximação da primeira estratégia, já que, desta forma, estaremos prevenindo tanto a reversão como a repetição do movimento.

II.5.1 - Implementação

Vamos nos valer de um exemplo para explicar os aspectos de implementação de níveis de aspiração, aplicados à estratégia de proibição dos movimentos reversos,

representada pelo critério (b).

Suponhamos que $c(x)$ assumam valores inteiros definidos numa faixa $q = 1, \dots, U$. Defina-se, então, $Best(q)$ como o menor valor de $c(x)$ que pode ser encontrado pela reversão de um movimento prévio que produziu $s(x)$, tal que $c(s(x)) = q$. Inicialmente $Best(q) = U + 1$ para todo q .

Quando um movimento de solução específica $x \rightarrow s(x)$ é efetivado, atualiza-se $Best(q)$ para $q = c(s(x))$, segundo a regra $Best(q) = \text{Min}(Best(q), c(x))$. Em iterações subsequentes, caso um movimento tabu que produza $s(x)$ seja apontado como candidato à efetivação, verifica-se se o mesmo satisfaz a seguinte condição:

$$(C1) \quad c(s(x)) < Best(c(x))$$

Em caso afirmativo, diz-se que o movimento atingiu o nível de aspiração necessário para a superação de seu status tabu. Em outras palavras, foi possível verificar que a efetivação do movimento não implica em risco de ciclagem.

Para uma melhor compreensão do seu funcionamento, consideremos que num dado processo de busca tenham sido realizados os seguintes movimentos de solução específica:

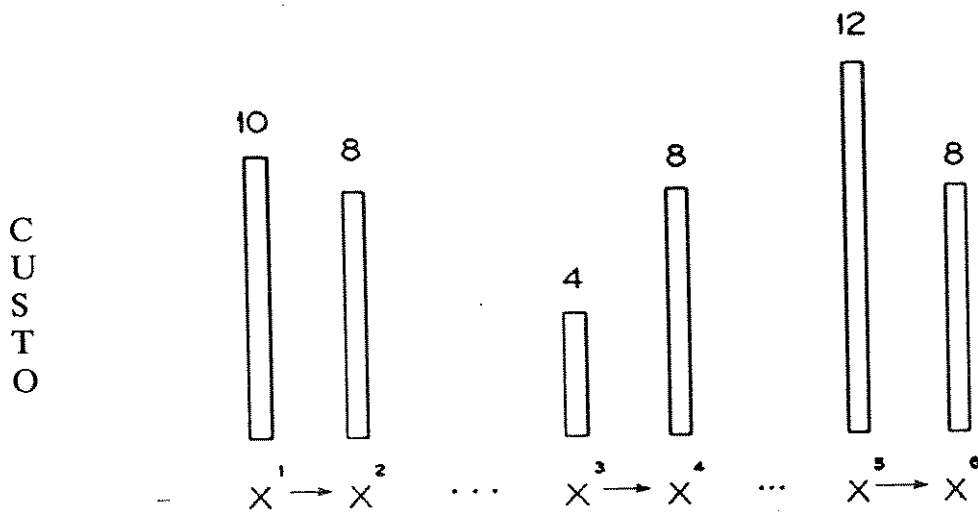


Figura II.3

Segundo a regra de atualização apresentada, $Best(8)=4$.

Consideremos que, a seguir, a função de avaliação aponte o seguinte movimento como o melhor candidato a ser efetivado:

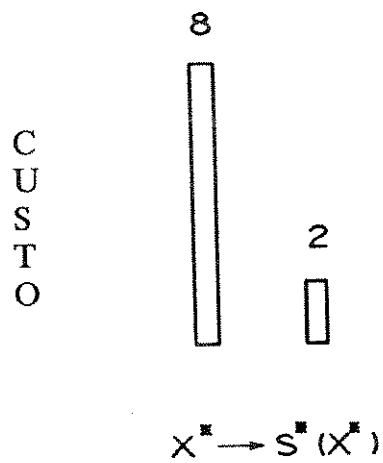


Figura II.4

Apesar de não representar a reversão de nenhum dos movimentos prévios, suponhamos que este movimento compartilhe todos os atributos associados a alguma destas reversões. Naturalmente, ele seria classificado como movimento tabu. No entanto, a verificação da condição (C1) indica que o nível de aspiração foi atingido ($c(s^*(x^*)) < Best(c(x^*))$) e assim é permitido ao movimento a superação de seu status tabu e, conseqüentemente, sua efetivação.

Neste exemplo, a utilização de níveis de aspiração manteve a informação de que até aquele momento todas as soluções a partir das quais foram geradas soluções com custo igual a 8, tinham o custo mínimo de 4. Portanto, garante-se que não haverá reversão a partir da solução de custo 8 se a solução resultante tiver custo menor que 4. Acima deste mínimo, entretanto, não se guardam informações, de forma que um movimento tabu como:

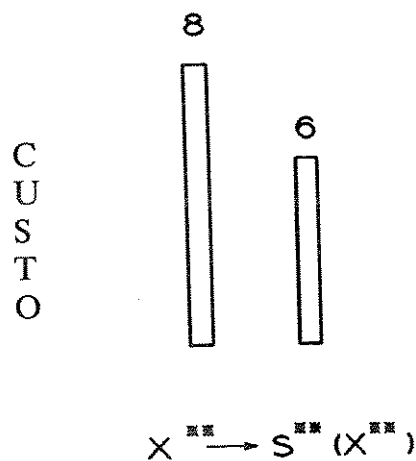


Figura II.5

seria rejeitado, apesar de seu reverso não ter sido realizado anteriormente. A não superação do estado tabu significa apenas que o risco de ciclagem não foi eliminado.

Vamos analisar a implementação de níveis de aspiração considerando os movimentos diretos, ou seja sua aplicação à estratégia (a) previamente definida. A forma correspondente de (a) seria definir $Best(q)$ para todo $q = c(x)$, como o menor valor de $c(s(x))$ que pode ser obtido por todos os movimentos prévios $x' \rightarrow s(x)$ tal que $c(x') = c(x)$. $Best(q)$ é inicializado conforme indicado na abordagem anterior, porém sua atualização segue a regra $Best(q) = \text{Min}(Best(q), c(s(x)))$. A condição de superação do estado tabu é idêntica à apresentada anteriormente, ou seja $c(s(x)) < Best(c(x))$.

Observemos, a seguir, duas seqüências de movimentos onde ambas as estratégias de nível de aspiração são aplicadas:

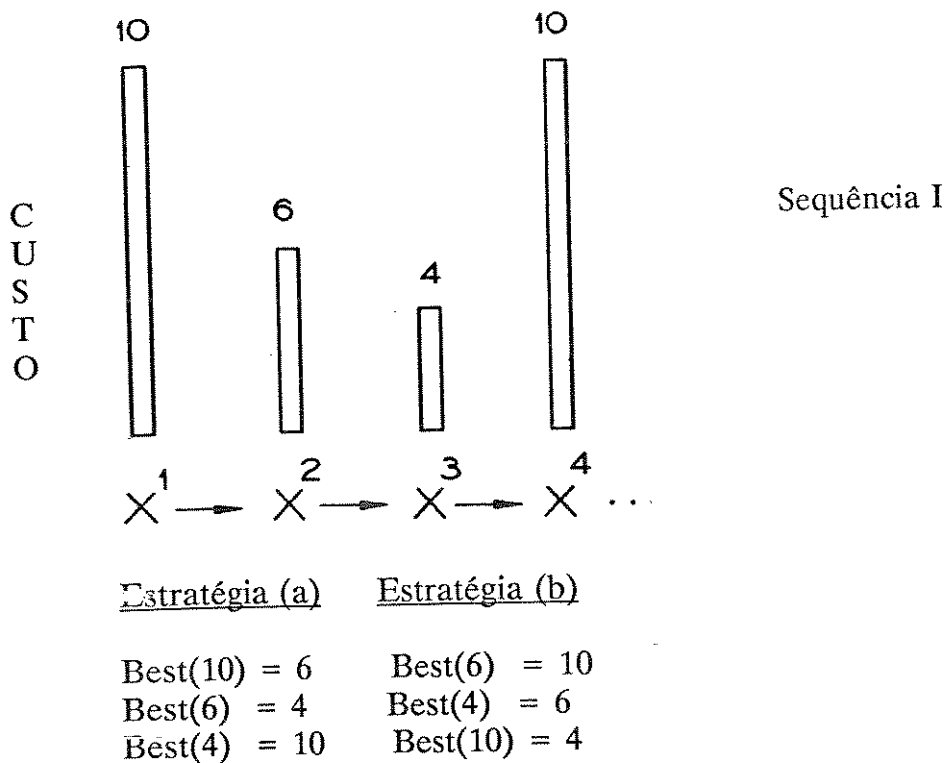
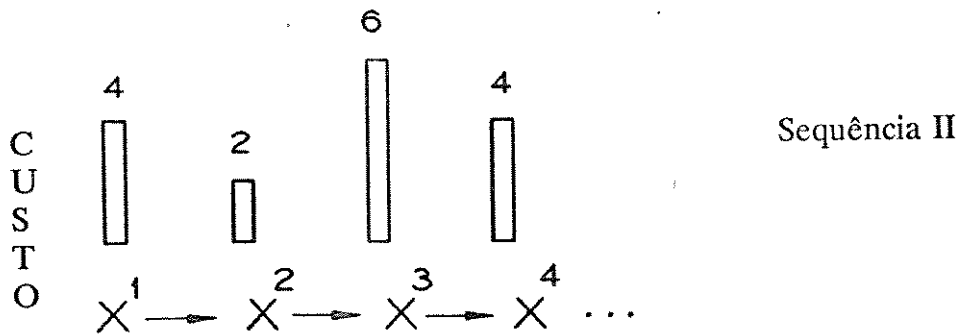


Figura II.6



(II) Estratégia (a) Estratégia (b)

$$\text{Best}(4) = 2 \qquad \text{Best}(2) = 4$$

$$\text{Best}(2) = 6 \qquad \text{Best}(6) = 2$$

$$\text{Best}(6) = 4 \qquad \text{Best}(4) = 6$$

Figura II.7

Na sequência I, suponha que a partir de x^4 , fosse indicado um movimento tabu que, se efetivado, resultaria numa solução de custo 4 (risco de reversão). Graças à estratégia (b) o movimento não seria realizado, já que $\text{Best}(10)=4$ não permite a satisfação da condição (C1). Na sequência II, caso o movimento candidato tabu gerasse uma solução de custo 2 (risco de repetição), o movimento seria rejeitado pela intervenção da estratégia (a), que forneceu $\text{Best}(4) = 2$.

Estes exemplos sugerem a utilização simultânea de (a) e (b), que por apresentarem a mesma condição de superação do estado tabu podem ser combinadas facilmente num mesmo procedimento. Neste caso:

$$\text{Best}(q) = \text{Min} (\text{Best}(q), c(x), c(s(x))) \text{ para todo } q = c(x), c(s(x))$$

Para $q = c(x^0)$ (onde x^0 é a solução de partida), toma-se $c(x) = U$. Para $q = c(x^f)$

(onde x^f é a última solução do processo iterativo), faz-se $c(s(x)) = U$.

Uma forma interessante de utilização dos níveis de aspiração consiste em defini-los apenas com respeito aos movimentos correntemente tabu. Para tal, cria-se uma lista tabu T_p para cada atributo do movimento $a_p(s,x)$. A cada lista é designado um comprimento t_p , definido segundo razões hierárquicas dentre os atributos. O estado tabu é, então, estabelecido, associando-se a estas listas uma função de aspiração $A(e)$, definida para todo $e = a_p(s,x)$.

Quando um movimento $x \rightarrow s(x)$ é realizado, o valor de $A(e)$ para todo $e = a_p(s,x)$ é atualizado de acordo com os critérios apresentados anteriormente. Ou seja:

$$A(e) = \text{Min} (A(e), c(x)) \text{ ou}$$

$$A(e) = \text{Min} (A(e), c(x), c(s(x))) \text{ (caso se opte pela forma combinada)}$$

$A(e)$ é inicialmente igual a $U + 1$ para todo $e = a_p(s(x))$ e posteriormente assume este valor sempre que e deixa de pertencer a todas as listas tabu.

O estado tabu do movimento pode ser então definido como uma função do estado de seus atributos. Dado $a_p(s,x)$, este atributo terá "passe livre" se $c(s(x))$ for menor que o nível de aspiração requerido para o atributo. Em outras palavras, $a_p(s,x)$ recebe "passe livre" se $c(s(x)) < A(e)$ para $e = a_p(s,x)$. Como os elementos e fora da lista têm $A(e) = U + 1$, então automaticamente recebem o seu passe. Ao movimento s é permitido então superar seu estado tabu, se todos ou um certo número de atributos de suficiente importância, tiverem "passe livre".

A versão aqui apresentada leva em consideração a necessidade de discriminação dos

atributos de acordo com a influência que cada um, individualmente, exerce na qualidade das soluções geradas. Tal discriminação se reflete na atribuição de diferentes tamanhos de lista, diferentes níveis de aspiração e diferentes pesos quando na determinação do status do movimento. Sob esta perspectiva, obtém-se uma atuação mais precisa dos mecanismos restritivos.

II.6 - LISTAS TABU E OSCILAÇÃO ESTRATÉGICA

A Busca Tabu, além de sua já conhecida atuação em termos de prevenção de ciclagem e, por conseguinte, na continuação da busca, apresenta um aspecto muitas vezes despercebido, porém de singular importância - *a indução de comportamentos de busca*.

Nas aplicações até então apresentadas, estes efeitos assumem sua forma mais básica, manifestando-se numa sucessão de soluções na qual o valor da função objetivo oscila entre o decréscimo e o crescimento. Tal oscilação resulta da combinação da orientação agressiva da busca (e que estimula o uso de algoritmos com estratégias gulosas) e do elemento restritivo representado pelas listas tabu.

No entanto, vários estudos têm considerado a utilização de listas tabu adicionais com vistas à oscilação deliberada de *outros* parâmetros. Conforme será visto a seguir, a aplicação destes recursos é vantajosa na medida que torna possível uma variedade de análises e expressivos ganhos estratégicos.

II.6.1 - Aplicações em pós-otimização

Tomemos como exemplo, o Problema de Roteamento de Veículos, no qual se deseja minimizar a distância total percorrida, sujeito à satisfação das restrições dos clientes. Em situações reais, existe interesse não só em se determinar a configuração de rotas de menor custo, mas também a configuração de rotas de custo mínimo para um certo número de veículos próximo do valor ótimo. A razão deste interesse é a necessidade de ter soluções disponíveis quando certos parâmetros são alterados, tais como a quebra de um ou mais veículos, a falta de alguns motoristas, etc.

Neste caso, uma segunda lista tabu é utilizada para forçar o aumento e diminuição do número de veículos, gerando um padrão de busca semelhante ao balanço de um pêndulo, em ambos os lados do número correntemente associado a um ótimo local. Esta forma de controle é útil mesmo quando se deseja determinar a configuração das rotas para um único número de veículos; a indução de oscilações em torno do número ótimo pode revelar oportunidades de obtenção de configurações mais atraentes.

Outro importante estímulo para a utilização de uma estratégia oscilante é que ela permite a descoberta de boas soluções para *condições perturbadas*. O tratamento formal de problemas de roteamento envolve a rígida satisfação de restrições, quando na prática pode ser possível alterar ou relaxar algumas restrições. Em particular, o acesso a resultados que violam ligeiramente certas restrições, permite ao usuário reconsiderar sua relevância. É possível aceitar uma configuração de rotas que dê um bom resultado mas que não

satisfaça todas as restrições de forma exata. Por exemplo, uma boa solução pode incluir uma rota que exceda o tempo máximo de rota em apenas alguns minutos, ou um veículo que chega ligeiramente mais tarde do que o requerido por um dado cliente.

II.6.2 - Utilização alternada de métodos opostos

Vários tipos de procedimentos de busca heurística têm sua "imagem especular", conforme exemplificado pelas classificações de métodos como "primais ou duais", "construtivos ou destrutivos", "interiores ou exteriores", etc. Melhor que selecionar apenas um tipo de busca de tal classificação, ganhos estratégicos são obtidos ao se considerar sua utilização *alternada*. Estratégias opostas podem ser tipicamente organizadas de tal maneira que cada ponto de cruzamento entre elas represente um ponto de otimalidade local (relativo a uma dada orientação de busca). Seja, por exemplo, uma função de avaliação $E(x)$ de uma solução tentativa x (parcial ou completa), expressa na forma $E(x) = aF(x) + bO(x)$, onde $F(x)$ denota uma medida de factibilidade e $O(x)$ uma medida de otimalidade. O propósito de definir $F(x)$ e $O(x)$ nesta representação é de permitir que uma "diferenciação qualitativa" entre factibilidade e otimalidade seja implementada pelo ajuste de a e b , como uma característica dinâmica do processo.

A natureza deste ajuste pode ser ilustrada pela referência aos procedimentos heurísticos "inside out" e "outside in". Uma abordagem "outside in" começa a uma distância da região factível e altera os valores das variáveis (como uma regra, unidirecionalmente),

até que a factibilidade seja atingida. Em contraste, uma abordagem "inside out" começa dentro da região factível (após uma inicialização, se requerido) e muda o valor das variáveis até que qualquer possibilidade de melhoria implique na saída desta região.

Manipulando a e b, e usando listas tabu simples para impedir ciclagem, obtém-se um mecanismo conveniente para progredir além do domínio destes dois procedimentos - adotando-se um método flexível de abordagem e então cruzando além das fronteiras factíveis em quaisquer dos lados. Os parâmetros podem ser usados para governar a profundidade de penetração além ou para dentro da região factível antes de determinar a trajetória de volta à periferia. No espírito da perspectiva de Inteligência Artificial, esta manipulação de pesos oferece uma oportunidade de empregar uma estratégia de aspiração adaptativa, restringindo as faixas de parâmetros em torno de valores que escondem boas soluções e, então, alargando as faixas depois de um período em que soluções melhores não foram localizadas. Tal manipulação dinâmica de a e b permite funções muito mais sutis, que não são facilmente expressas em forma matemática.

Esquemáticamente:

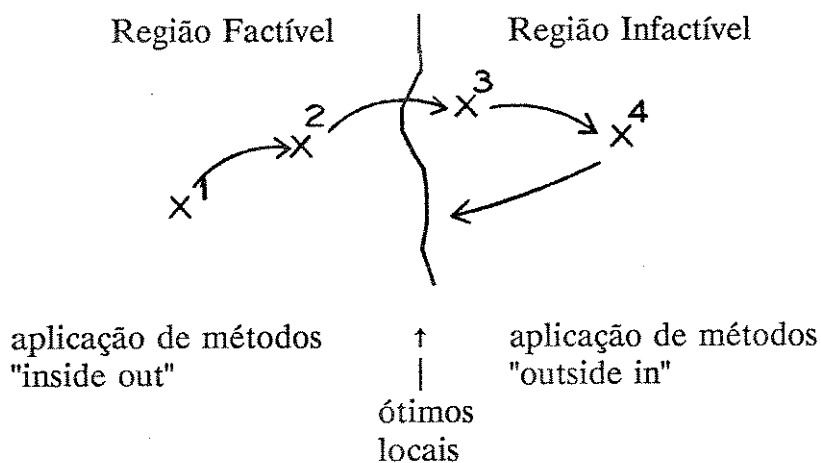


Figura II.8

Listas tabu que não apenas proíbem reversões de movimento mas que também promovem tais cruzamentos e retornos, oferecem uma maneira efetiva de superar limitações de otimalidade local de buscas padrão e de encontrar soluções melhoradas. Cabe aqui observar que a Busca Tabu foi originalmente concebida como um dispositivo para implementação deste tipo de estratégia [18].

II.7 - FUNÇÕES DE MEMÓRIA DE MÉDIO E LONGO PRAZO

Além de funções de curto prazo representadas pelas listas tabu, a Busca Tabu também emprega funções de memória de médio e longo prazo. O objetivo destas funções é de promover respectivamente a *intensificação regional* e a *diversificação global* da busca.

Memórias de médio prazo operam, guardando e comparando características de um certo número de melhores soluções- tentativas geradas durante um período específico de busca. Características que são comuns a todas ou à grande maioria destas soluções são tomadas como um atributo regional de boas soluções. O método procura, então, novas soluções que exibam estas características, impondo restrições ou penalizações aos movimentos disponíveis.

Esta estratégia é particularmente útil para a resolução de problemas grandes. Sob uma orientação gulosa como a da Busca Tabu, existe uma tendência à geração de boas soluções. Boas soluções, por sua vez, tendem a incorporar apenas um subconjunto de

elementos de decisão (variáveis, arestas, etc.) Por exemplo, no contexto de Problemas de Roteamento, para grafos moderadamente densos, o número de arestas diferentes incorporadas nas rotas em qualquer solução é geralmente apenas uma fração do conjunto total de arestas. Portanto, após um certo número inicial de iterações, o método pode eliminar todas as arestas ainda não incorporadas em qualquer rota e então dedicar-se ao problema resultante, consideravelmente menor. Já que a execução das iterações é agora mais rápida, a busca pode examinar muito mais alternativas em um dado período de tempo, além de focar possibilidades que parecem bem mais atraentes.

Funções de memória de longo prazo, cujo objetivo é o de diversificar a busca, empregam princípios que são o reverso daqueles de memórias de médio prazo. Ao invés de estimular a busca em regiões que contêm boas soluções previamente encontradas, a função de memória de longo prazo, guia o processo para regiões que marcadamente contrastam com aquelas examinadas até então.

Esta abordagem difere daqueles métodos que buscam diversidade através da geração de uma série de pontos de partida aleatórios, e que, portanto, não possuem mecanismos que lhes permitam acumular conhecimento. O objetivo é a criação de um critério de avaliação que pode ser usado por um processo de busca heurística, especificamente feito para produzir **deliberadamente** novos pontos de partida. Sob esta orientação, estes critérios penalizam as características que memórias de longo prazo consideram como frequentes em execuções prévias do processo de busca.

Uma aplicação do PRV fornece, novamente, uma ilustração conveniente. Uma

forma simples de memória de longo prazo neste sistema é a contagem do número de vezes que cada aresta aparece nas rotas geradas. Uma penalização proporcional a esse número favorece a geração de "boas" soluções de partida, que tenderão a evitar tais arestas usadas mais comumente no passado. A diversificação alcançada por este critério pode ser ampliada, mantendo-se as penalidades por um certo período após o início da heurística incorporada com a busca tabu. As penalidades são então eliminadas e a busca tabu procede de acordo com seu critério de avaliação normal. Este mesmo tipo de procedimento pode ser usado para continuar diretamente do ponto atual de busca para uma nova região, sem recorrer à geração de uma nova solução.

CAPÍTULO III

A BUSCA TABU APLICADA A PROBLEMAS DE ROTEAMENTO DE VEÍCULOS

III.1 - INTRODUÇÃO

O Problema de Roteamento de Veículos (PRV) e suas variantes, sendo um problema combinatório, mostra-se adequado à aplicação da técnica de Busca Tabu. Estudos preliminares da utilização da estratégia no tratamento de problemas do Caixeiro Viajante (PCV) [20,21], e aplicações a problemas de roteamento básico [45], apresentaram resultados bastante promissores.

Em ambos os trabalhos, a Busca Tabu foi aplicada à heurística 2-OPT e o desempenho das versões geradas comparado à 3-OPT. No que diz respeito ao PCV, a utilização de listas tabu simples e estruturas de níveis de aspiração, produziu custos de solução inferiores aos do método 3-OPT padrão, na maioria dos problemas testados. Quanto ao PRV, foi obtido desempenho equivalente entre os dois algoritmos em problemas de até 50 clientes, utilizando-se tão somente mecanismos de Busca Tabu simples.

Estes resultados evidenciam a habilidade da Busca Tabu em guiar heurísticas relativamente simples, de forma a lhes permitir superar o desempenho de procedimentos normalmente superiores. No contexto específico de Problemas de Roteamento, vemos tais evidências como um estímulo à sua utilização em extensões que caracterizem aplicações práticas de interesse.

Uma extensão de grande importância, por exemplo, consiste na inclusão de restrições temporais sobre uma parte ou no conjunto total de clientes. Estas restrições, comumente chamadas de janelas de tempo, definem um período dentro do qual os clientes

devem ser atendidos, ou seja, no qual o serviço ou descarregamento de bens deve ser iniciado e/ou completado.

Na presença de janelas de tempo, os custos de roteamento consideram não apenas a distância (ou tempo) total, mas podem também incluir o tempo de espera, resultante da chegada do veículo a um dado cliente antes do início de sua janela de tempo. Sob uma perspectiva computacional, a existência de tais restrições confere aos problemas uma complexidade muito maior [36].

Por representarem situações reais, naturalmente encontradas nas mais diversas áreas, problemas de roteamento assim caracterizados formam um importante segmento, conhecido como Problemas de Roteamento de Veículos com Janelas de Tempo (PRVJT).

Não obstante sua importância, pouco tem sido dedicado ao desenvolvimento de técnicas para o tratamento de problemas que como o PRVJT, envolvem considerações não somente espaciais mas também temporais.

Reconhecendo tais limitações, resolvemos abordar neste trabalho variações do PRV caracterizadas basicamente pela inclusão de restrições temporais. Especificamente, os problemas aqui tratados serão discriminados pela existência ou não de horizontes de tempo, de tempos de serviço e de janelas de tempo para clientes. Tal discriminação sugeriu a divisão dos problemas tratados nos seguintes grupos:

- i. Problemas *sem* horizonte de tempo, *sem* tempo de serviço e clientes *sem* janelas de tempo.

- ii. Problemas *com* horizonte de tempo, *com* tempo de serviço e clientes *sem* janelas de tempo.
- iii. Problemas *com* horizonte de tempo, *com* tempo de serviço e clientes *com* janelas de tempo.

Os problemas de roteamento aqui abordados têm como objetivo a obtenção da configuração de rotas que minimiza a distância total percorrida para o menor número de veículos compatível com os requerimentos dos problemas.

III.2 - DESCRIÇÃO DO ALGORITMO

Em linhas gerais, o algoritmo é constituído das seguintes etapas:

1. Geração de uma solução inicial.
2. Procedimentos de melhoria de rotas a partir da solução inicial, compostos de:
 - 2a) Sequência de operações até um mínimo local.
 - 2b) Sequência de operações a partir do mínimo local encontrado em 2a.

A seleção do algoritmo foi então estabelecida a partir das características dos grupos de problemas tratados. Em função da existência de restrições temporais, utilizou-se na etapa 1, o algoritmo de construção sequencial de rotas de Marius Solomon [36]. Idealizado para problemas de roteamento com janelas de tempo, este algoritmo possui mecanismos de verificação de factibilidade temporal, os quais, dada a sua simplicidade, foram incorporados na etapa posterior.

Na segunda etapa, adotou-se um procedimento baseado na heurística de Moshe Dror e Larry Levy [10], originalmente concebida para problemas de roteamento de estoques. Esta heurística, ao introduzir o conceito de troca de nós em contraste aos métodos de troca de arestas, generalizou a técnica de melhoria de rotas r-OPT, até então limitada a sistemas de uma única rota.

Etapa 1 - GERAÇÃO DE UMA SOLUÇÃO INICIAL

HEURÍSTICA SEQUENCIAL DE CONSTRUÇÃO DE ROTAS [36]

1. Seja r a rota correntemente sendo construída. Faça $r=0$.

2. Inicialização das rotas:

Faça $r = r + 1$

Escolha o nó inicial da rota r , através de um dos seguintes critérios:

(a) o cliente não roteado mais distante da garagem

(b) o cliente não roteado com o menor prazo de atendimento

3. Seja $(i_0, i_1, i_2, \dots, i_m)$ a rota em questão, com $i_0 = i_m =$ garagem (nó 0) e $d(i, j)$ a distância correspondente ao caminho de menor custo entre quaisquer dois nós distintos i e j . Para cada cliente não roteado, obtenha a posição de inserção factível que provê a melhor avaliação segundo o critério:

$$c_1(i(u), u, j(u)) = \text{Min} [c_1(i_{p-1}, u, i_p)], \quad p = 1, \dots, m$$

tal que

$$c_1(i, u, j) = \alpha_1 c_{11}(i, u, j) + \alpha_2 c_{12}(i, u, j), \quad \alpha_1 + \alpha_2 = 1$$

$$\alpha_1, \alpha_2 \geq 0$$

e

$$c_{11}(i, u, j) = d(i, u) + d(u, j) - \mu d(i, j), \quad \mu \geq 0;$$

$$c_{12}(i, u, j) = b_{ju} - b_j,$$

onde b_j é o tempo atual de início do serviço no cliente j e b_{ju} o tempo de

início do serviço no cliente j , ajustado pela inserção de u na rota.

4. Dentre todos os clientes não roteados, obtenha u^* segundo o critério:

$c_2(i(u^*), u^*, j(u^*)) = \text{ótimo} [c_2(i(u), u, j(u))]$, para todo u roteado e factível onde

$$c_2(i, u, j) = \delta d(0, u) - c_1(i, u, j), \quad \delta \geq 0;$$

Insira o cliente u^* na rota r entre $i(u^*)$ e $j(u^*)$. Atualize os tempos de início de serviço de todos os clientes, a partir de $j(u^*)$ e a capacidade restante do veículo.

5. Se todos os clientes tiverem sido inseridos, páre.

Se ainda há clientes a serem inseridos, verifique se a capacidade do veículo foi excedida ou se não existem mais posições factíveis na rota r . Neste caso, vá para o passo 2 e inicialize uma nova rota. Caso contrário vá para o passo 3 e continue as inserções.

O critério utilizado no passo 4 busca maximizar os ganhos de inserção de uma cliente na rota parcial que está sendo contruída, em oposição à servi-lo numa rota direta. Se estabelecermos, por exemplo, $\mu = \alpha_1 = \delta = 1$ e $\alpha_2 = 0$, então $c_2(i, u, j)$ representa a economia em distância ao se servir o cliente u na mesma rota que serve os clientes i e j , em oposição ao serviço direto e individual de u . Se por outro lado, $\alpha_1 = 0$ e $\alpha_2 = 1$ (mantendo-se os valores dos outros parâmetros), $c_2(i, u, j)$ representará a economia em tempo. Outros efeitos podem ser obtidos alterando-se o valor de parâmetros diferentes de zero; por exemplo, o incremento de δ de 1 para 2 funciona como uma penalização sobre rotas compostas de um único cliente. Como visto, não são considerados os custos fixos

resultantes da aquisição e manutenção dos veículos, nem penalidades sobre tempos de espera.

O trabalho de Marius Solomon apresenta dois outros critérios de inserção cujos detalhes podem ser obtidos na referência [36]. O primeiro deles visa minimizar uma medida de distância e tempo totais; o outro leva em consideração a urgência de cada cliente em ser servido.

Etapa 2 - PROCEDIMENTO DE MELHORIA DE ROTAS

Antes da descrição do algoritmo convém explorar algumas diferenças e semelhanças conceituais entre procedimentos de melhoria de rotas e as heurísticas de troca de arestas r-OPT.

Conforme já mencionado no capítulo 1, o método r-OPT consiste na transformação de uma dada rota, através da substituição de r de suas arestas por um conjunto de r arestas não pertencentes à solução. A escolha destes conjuntos segue o critério de minimização de distâncias, isto é, a cada iteração são selecionados dentre os grupos disponíveis aqueles que promovem a maior economia positiva. Uma rota é dita r-ÓTIMA se é impossível obter soluções a um custo inferior trocando-se quaisquer conjuntos de r de suas arestas por outro conjunto de r arestas.

No exemplo abaixo é apresentada uma aplicação da heurística 2-OPT. O termo de economia resultante, neste caso, é obtido pela expressão $OS(A,D;B,C) = d(A,D) + d(B,C)$

- $d(A,C) - d(B,D)$.



Figura III.1

Uma transformação r altera, em graus variados, a sequência em que está ordenado um dado conjunto de nós, e é portanto orientada para problemas de uma só rota, do tipo do Caixeiro Viajante simples. Sua aplicação a sistemas de m rotas, $m > 1$, exige a conversão das m rotas em uma rota equivalente. Tal equivalência é obtida pela agregação das rotas originais, separadas por "nós fantasmas" cujas coordenadas são idênticas às da garagem. Esquemáticamente:

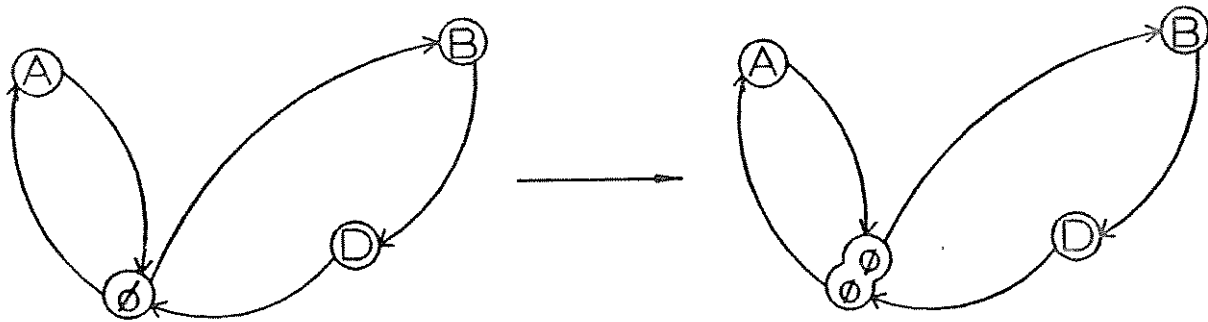


Figura III.2

Vejam os um exemplo da aplicação da heurística 2-OPT a sistemas com 2 rotas, na qual foi efetuada a troca das arestas C - 0 e G - H por C - G e H - 0.

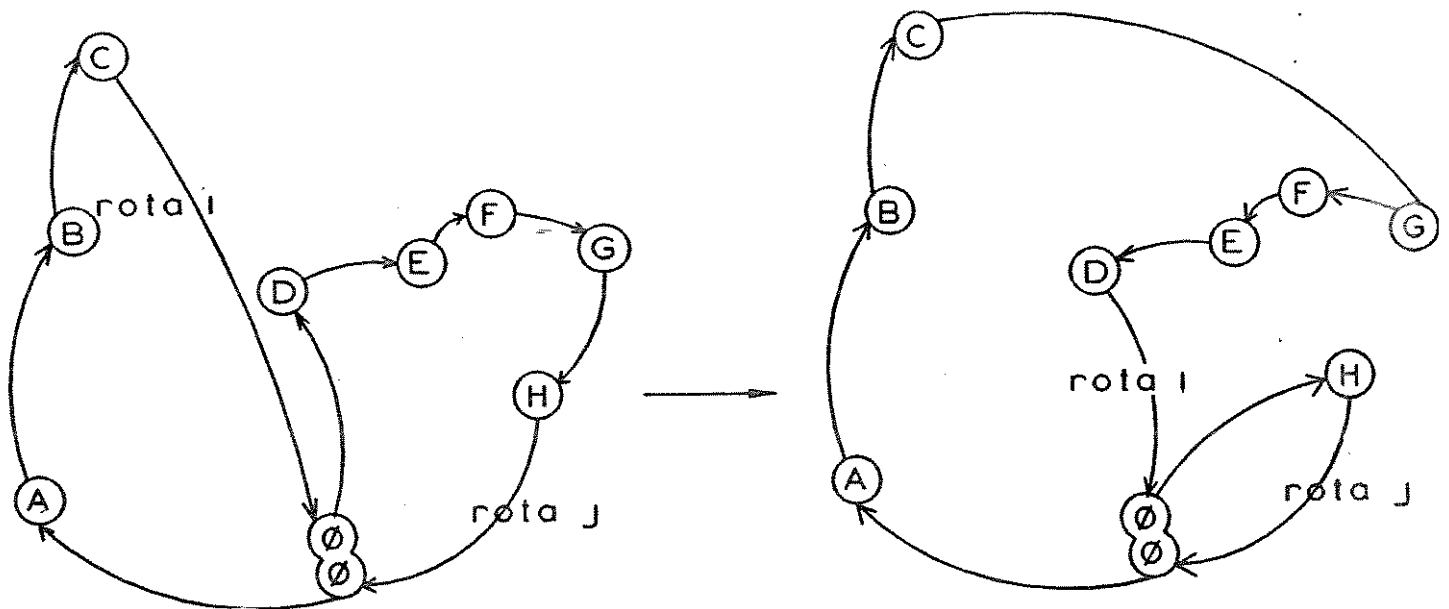


Figura III.3

Com o movimento, todos os nós da rota j, exceto "H", seriam transferidos para a rota i. Nota-se que com a transferência, inverteu-se a sequência original na qual o segmento estava ordenado. Se este problema incluir restrições temporais como janelas de tempo, ou

mesmo restrições de capacidade, é pouco provável que movimentos que alterem tão extensamente a configuração das rotas como o apresentado, sejam factíveis. Portanto, não é conveniente a aplicação de heurísticas r-OPT a problemas sujeitos a estas restrições, especialmente se em tais problemas as rotas forem compostas por um número razoável de nós. Neste caso, o mecanismo de busca estará sujeito a ocorrência de inúmeros movimentos deste tipo.

O procedimento de intercâmbio de nós de Dror e Levy, por outro lado, tem como principal atrativo a aplicabilidade a sistemas de m rotas. A estrutura original é mantida de maneira que a transformação de uma solução em outra é obtida por meio de operações inter-rotas. Especificamente, utilizam-se operações de inserção de um único nó e troca simples de dois nós.

Esquemáticamente:

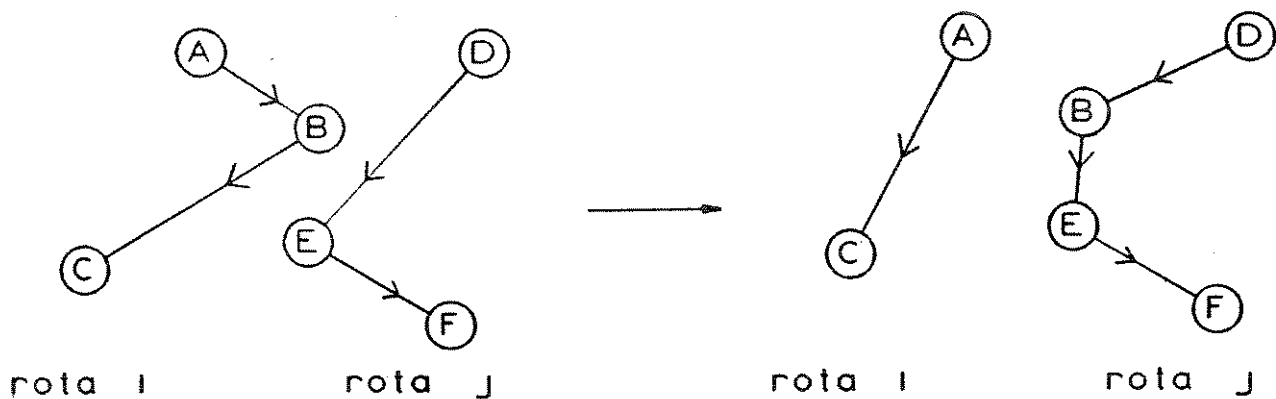


Figura III.4 - Inserção do nó B da rota i na rota j entre os nós D e E

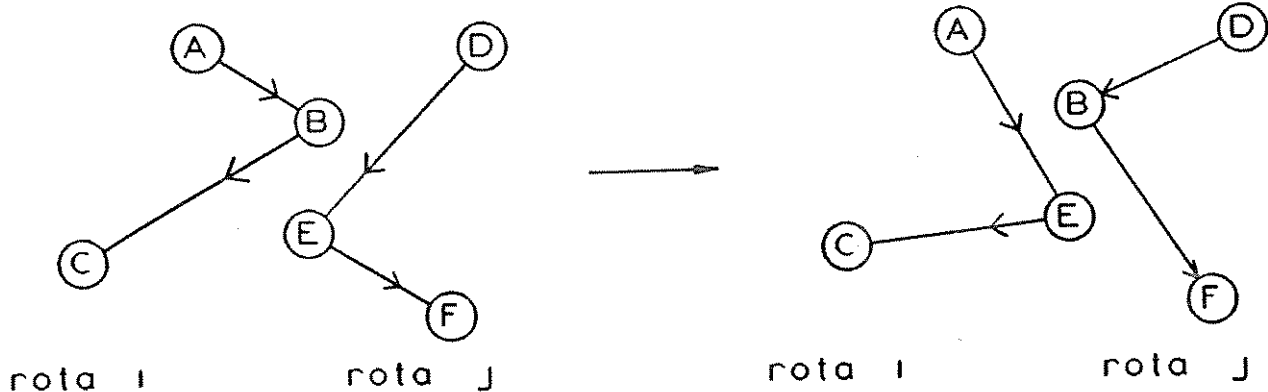


Figura III.5 - Troca do nó B da rota i pelo nó E da rota j

As diretrizes deste método são, entretanto, essencialmente as mesmas das heurísticas r-opt. A escolha das operações é igualmente guiada pelo critério da máxima redução na distância total. Ou seja, definimos de forma similar uma função de avaliação que corresponde ao termo de economia (OS), resultante da aplicação das operações apresentadas. No exemplo da figura III.4, o OS correspondente à inserção do nó B entre os nós D e F é dado por:

$$OS(A,B,C;D,E) = d(A,B) + d(B,C) + d(D,E) - d(A,C) - d(D,B) - d(B,E)$$

No caso da troca do nó B pelo nó E, temos:

$$OS(A,B,C;D,E,F) = d(A,B) + d(B,C) + d(D,E) + d(E,F) - d(A,E) - d(E,C) - d(D,B) - d(B,F)$$

Assim como as heurísticas r -opt, podemos definir como condição de otimalidade local, a situação em que nenhuma redução de distância é possível através da efetivação de operações de inserção ou troca de nós. Um conjunto de rotas que atinge esta situação é chamada *troca-dupla ótima*.

É oportuno tecer alguns comentários sobre esta heurística que combina dois procedimentos clássicos em problemas de roteamento: a inserção e troca de nós.

O conceito de otimalidade local de uma solução x^* alcançada por uma heurística específica se traduz na inexistência de melhores soluções, dentre aquelas que podem ser alcançadas a partir de x^* , através da aplicação dos mecanismos de busca da heurística. Assim, para uma heurística de inserção de nós está configurado um ótimo local quando todas as tentativas de inserção de nós na melhor solução corrente não provocarem um OS positivo.

Porém, o mínimo local pode ser superado se a heurística incorporar um outro (ou outros) mecanismo de busca, como por exemplo, a troca de dois nós. O novo ótimo local se caracterizaria, desta maneira, pela inexistência de OS positivo nos dois procedimentos utilizados.

Estes aspectos sugerem o uso destes dois mecanismos incorporados num mesmo procedimento heurístico, o que é sugerido no algoritmo guloso de Dror e Levy. Eles utilizaram heurísticas r -OPT e de intercâmbio de arestas (2-OPT e 3-arc) com vistas à alteração da configuração de cada rota individualmente, sempre que se atinja a otimalidade local através do procedimento de intercâmbio de nós. Desta forma, mesclam-se transformações inter e intra rotas.

Como se pode perceber, esta mescla de operações é um mecanismo particularmente importante quando se utilizam métodos sujeitos a limitações de otimalidade local. Mesmo em procedimentos isentos de tais limitações, é interessante adotar este mecanismo, na medida em que confere uma orientação agressiva ao processo de busca. Ao contrário do algoritmo original, aqui será utilizada apenas a heurística 2-OPT para a promoção de transformações intra-rotas. Esta heurística será aplicada em cada rota sempre que se atingir um mínimo local, assim definido segundo as operações de intercâmbio de nós.

A seguir será apresentado o algoritmo de Dror e Levy modificado, em duas fases.

ALGORITMO GULOSO DE DROR E LEVY MODIFICADO

Fase I

Nesta primeira fase, são realizados movimentos associados a termos de economia estritamente positivos. Isto implica que o valor da função objetivo decresce monotonicamente, ou seja, o custo de solução em qualquer iteração é sempre menor que os obtidos nas iterações anteriores. Não há risco de obtenção de soluções previamente encontradas (ciclagem) e conseqüentemente não há necessidade de verificar o status dos movimentos candidatos.

1. Inicialização da fase I:

Seja x^* a solução inicial obtida na etapa anterior.

Faça $T = \phi$.

2. Calcule todos os termos de economia OS resultantes de operações de inserção e troca de nós. Mantenha numa lista os termos associados a operações que resultam em configurações factíveis. Se a lista estiver vazia, vá para o passo 9.
3. Ordene a lista de termos OS em ordem decrescente.
4. Se o primeiro termo OS for positivo, efetue a transformação correspondente, adicione a reversão do movimento à lista tabu e vá para o passo 2.
5. Aplique a heurística 2-OPT a cada uma das rotas. Caso seja realizado algum movimento, adicione a reversão do movimento à lista tabu e vá para o passo 2.

Fase II

Na segunda fase serão utilizadas listas tabu de forma a permitir a realização de operações inter-rotas que promovem tanto a melhoria como a piora do valor da função objetivo.

6. Inicialização da fase II:
Seja LIMIT o número de iterações desejado. Faça IT= contador de iterações = 0. Atualize x^* , fazendo-o igual à última solução encontrada no passo anterior.
7. Se IT=LIMIT vá para o passo 9.
Caso contrário, tome o primeiro termo OS da lista ordenada de operações de

CAPÍTULO IV

DESENVOLVIMENTO DO TRABALHO - COMPOSIÇÃO DA LISTA TABU

intercâmbio de nós que corresponda a um movimento não tabu. Faça $IT=IT+1$ e efetue o movimento. Seja x_{it} a solução resultante. Se $c(x_{it}) < c(x^*)$ então faça $x^* = x_{it}$. Adicione a reversão do movimento à lista T. Caso não haja movimentos não-tabu, libere o movimento mais antigo da lista e repita o passo. Se o movimento inter-rotas realizado estiver associado a um OS negativo, aplique a heurística 2-OPT a cada uma das rotas. Caso seja realizado algum movimento intra-rota, adicione sua reversão à lista tabu.

8. Calcule todos os termos de economia resultantes de operações de inserção e troca de nós, mantendo numa lista os termos correspondentes às operações que resultam em configurações factíveis. Se a lista estiver vazia, vá para o passo 9. Caso contrário, vá para o passo 7.
9. Apresente x^* como a melhor solução obtida no processo iterativo.

III.3 - A LISTA TABU

III.3.1 - Especificação dos atributos do movimento

A caracterização de atributos de movimento, ou seja, a seleção de uma faixa parcial de seus elementos descritivos é o primeiro passo para se estabelecer a composição da lista tabu. Existem maneiras diferentes de definir tais elementos, o que torna esta tarefa um pouco delicada, ainda mais levando em conta sua importância no processo de busca, e

consequentemente na qualidade das soluções obtidas. Como uma primeira proposição, consideremos o movimento como resultado de operações sobre nós e arestas.

O fato de serem utilizados elementos de naturezas distintas poderia, a princípio, sugerir para cada tipo de operação diferentes definições de atributos de movimento, e consequentemente, diferentes listas tabu. No que diz respeito à técnica 2-OPT, suas características tornam a descrição do movimento bastante simples e direta. Segundo a orientação de Glover [20,21], para definir este tipo de movimento é necessário identificar apenas as arestas eliminadas e adicionadas. Nas operações de intercâmbio de nós, entretanto, tal caracterização requer a utilização de uma função mais complexa, que forneça um número maior de informações. Estas informações geralmente constituem-se da operação efetuada (troca ou inserção), dos nós envolvidos (inclusive os nós vizinhos afim de especificar a posição na rota), das rotas envolvidas, etc.

No entanto, o estudo cuidadoso das operações mostra que é possível reduzi-las a processos de eliminação e adição de arestas, semelhantes aos usados nas técnicas 2-OPT. Por exemplo, na figura III.6, a troca do nó B da rota i pelo nó E da rota j pode ser vista como a eliminação das arestas (A,B), (B,C), (D,E) e (E,F) e a adição de (A,E), (E,C), (D,B) e (B,F):

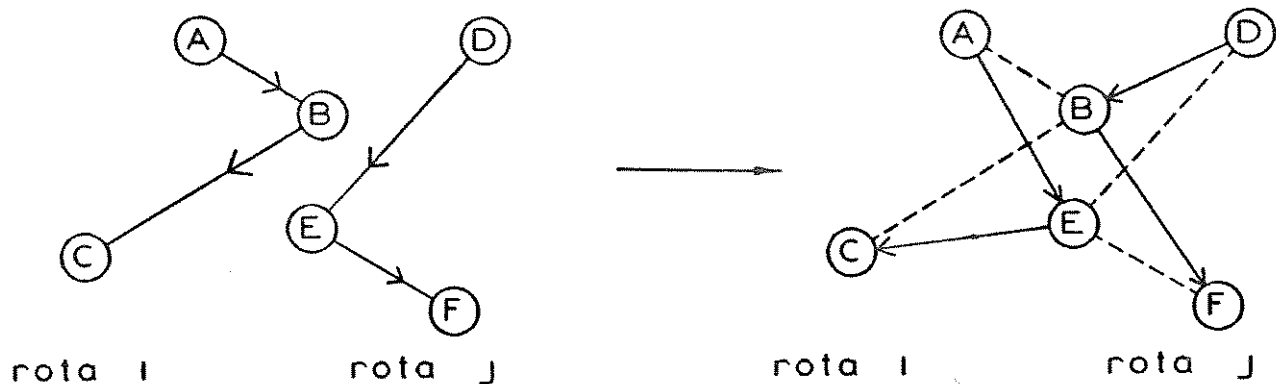


Figura III.6

Em vista disso, podemos definir os atributos do movimento como *Arestas Eliminadas* e *Arestas Adicionadas*. Além de evitar os requerimentos computacionais que acompanham a utilização de funções mais complexas, eliminam-se os problemas que resultariam de diferentes definições de movimento e diferentes listas tabu. E mais importante, para o algoritmo apresentado, tais funções tornariam muito difícil a formalização adequada de reversão do movimento.

III.3.2 - Composição da Lista Tabu

Como já mencionado no capítulo anterior, a prevenção de ciclagem exige a utilização de mecanismos que impeçam a reversão de movimentos realizados previamente. A reversão do movimento corresponde à reversão dos passos que compõem a operação realizada, ou seja, consiste na eliminação das arestas adicionadas e adição das arestas

eliminadas. Sugere-se, portanto, a criação de duas listas tabu; na primeira delas (DEL) serão armazenados os identificadores das arestas adicionadas e na segunda (AD), os identificadores das arestas eliminadas, nas últimas t iterações. O elemento restritivo atua nestas listas estabelecendo quais arestas, definidas sob um determinado atributo, recebem um caráter proibitivo segundo o outro atributo, nas t iterações seguintes. Ou seja, arestas eliminadas (adicionadas) numa dada operação estão sujeitas a restrições no que diz respeito à sua adição (eliminação) nas próximas t iterações. Uma aresta é dita tabu se a efetivação do movimento promover sua inclusão numa das listas, sendo que ela já pertence à outra lista.

O status do movimento é então definido como se segue:

Seja o conjunto $E(s)$ composto das arestas eliminadas e adicionadas em um dado movimento s . Seja também $E_t(s)$ um subconjunto de $E(s)$ tal que seus elementos são tabu. Diz-se que o movimento s é tabu, se e somente se o número de elementos de $E_t(s)$ for maior que L , onde L é um inteiro menor ou igual a soma das arestas eliminadas e adicionadas.

Desta forma definimos o status do movimento através da avaliação do status das arestas envolvidas na operação. Um dado movimento é classificado como tabu, se o número de arestas assim classificados exceder um certo limitante previamente definido. A razão da utilização de uma faixa de valores de limitante deve-se à má qualidade das soluções obtidas ao se adotar a orientação mais restritiva de conferir o status tabu a um movimento, se pelo menos uma de suas arestas envolvidas for tabu.

IV.1 - INTRODUÇÃO

No capítulo II foram apresentadas três formas de representação de listas tabu. Dentre elas, foi apontada como a mais conveniente aquela em que cada elemento está associado *não a* um único movimento, mas a conjuntos de movimentos que compartilham certas características. Tal representação foi estabelecida a partir de considerações de dominância e com vistas à minimização do esforço computacional no que concerne à conservação de memória e facilidade de processamento.

No entanto, não havia motivos aparentes para a utilização imediata desta representação mais sofisticada, de forma que, como uma abordagem inicial, nos restringimos ao objetivo básico de proibição de ciclagem. Ou seja, o passo inicial para a determinação da composição da lista tabu foi o de estabelecer atributos que representassem unicamente o movimento e sua reversão. A partir da análise do desempenho da representação simples das listas tabu é possível considerar a necessidade e uso de outras representações.

IV.2 - ATRIBUTO ECONOMIA

O algoritmo de melhoria de rotas utilizado associa a cada opção de movimento, uma medida de *economia em distância*, sobre a qual será baseada a escolha. Já que nosso principal objetivo foi o de definir funções de mínima complexidade, a primeira idéia foi a de utilizar estas economias resultantes como atributos do movimento. Sua implementação consistiu na criação de *duas listas tabu*, uma delas representando os *movimentos diretos* realizados nas ultimas t iterações e a outra, os *movimentos reversos* associados. Ou seja, de acordo com os atributos adotados, a cada iteração foi armazenada na primeira lista, a economia resultante da aplicação da operação e na segunda, o negativo deste valor. Assim, para todo movimento candidato à efetivação, verificou-se a existência da economia a ele associada em alguma das duas listas. Em caso afirmativo, o movimento foi classificado como tabu.

O desempenho desta composição foi então analisado em problemas de 10 a 60 nós do grupo iii, e conforme sugerido por Glover [20,21], adotou-se o tamanho de ambas as listas igual a sete. Foram verificados os seguintes resultados:

1) *Ocorrência de proibição de movimentos que não gerariam estados-solução prévios*

Isto indica que a economia, por si só, não é suficiente para caracterizar um único movimento; ou seja, mesmo os problemas tratados, cujos dados seguem uma distribuição uniforme, estão sujeitos à existência de movimentos diferentes com a mesma economia.

2) *Ocorrência de ciclagem*

O fato de serem utilizados problemas pequenos e com restrições adicionais de

janela de tempo, resulta em poucas opções de movimento, que tendem a se repetir no curso das iterações. Uma utilização conveniente das listas tabu arcaria com a responsabilidade de classificar dentre estes, aqueles movimentos cuja realização implicaria no retorno a soluções prévias. Verificou-se, entretanto, a ocorrência de ciclagem em algum estágio, para todos os problemas. Uma análise mais detalhada forneceu as seguintes razões:

2a. Tamanho insuficiente das listas

e/ou

2b. Incapacidade de reconhecimento de movimentos reversos equivalentes

A utilização do atributo economia não fornece mecanismos que permitam distinguir situações em que a reversão de um dado movimento ocorre através de uma combinação de dois outros movimentos, e vice-versa. Desta forma foi possível retornar a uma solução prévia, o que a priori, não implicaria em ciclagem, caso a lista associada ao primeiro atributo (movimento direto) atuasse convenientemente. Entretanto, na maioria dos problemas, em algum momento, esta lista não teve o tamanho suficiente para manter os movimentos proibidos sob seu escopo, e a ciclagem ocorreu.

O item 2b. foi considerado uma limitação cuja superação, ao contrário de 2a., implica em alterações mais profundas na estrutura da lista. Pelo que foi exposto acima, talvez fosse possível prevenir a ciclagem apenas encontrando-se o tamanho ideal das listas. Entretanto, nos pareceu mais interessante investir em definições de atributos mais satisfatórias.

IV.3 - ATRIBUTO CUSTO

A incapacidade de se impedir, sob certas circunstâncias, a reincidência de estados-solução, sugeriu a utilização de um atributo, neste sentido mais poderoso, já que está diretamente associado às próprias soluções - *o custo total*.

A lista tabu ficou limitada a um único campo onde seriam armazenados os *custos das soluções* a cada iteração. Para os mesmos problemas testados, observou-se que movimentos reversos eram permitidos; entretanto, se dentro do escopo da lista, nenhum retornou a soluções prévias. Ou seja, no intervalo entre a realização do movimento e sua reversão, outros movimentos se seguiram e alteraram, em diferentes regiões, a configuração das rotas. Por outro lado, a lista tornou-se plenamente capaz de identificar movimentos reversos equivalentes, de forma que movimentos que não seriam proibidos segundo o atributo economia, receberam o status tabu. Uma limitação óbvia diz respeito à incapacidade de distinção entre diferentes soluções com o mesmo custo. Pelo fato de não ter sido verificada nenhuma ocorrência deste tipo, resolveu-se optar, por hora, em permitir que a lista impedisse a realização de tais movimentos.

No que diz respeito à qualidade da solução final, não se evidenciou, apesar das observações acima, superioridade do atributo custo com relação à economia. Em alguns problemas, obteve-se melhores resultados utilizando-se custo; em outros, o inverso ocorria. A esperada superioridade do custo total, baseada no fato de que soluções prévias estão impedidas de reincidir, não se verifica em situações em que tal reincidência mostra-se conveniente, como será visto no exemplo a seguir.

Seja a sequência de soluções x^1 , x^2 e x^3 .

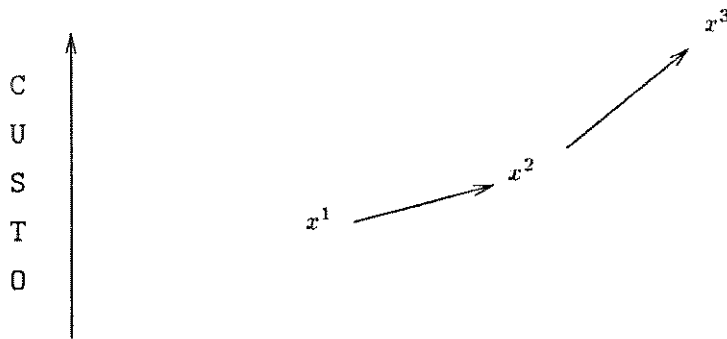


Figura IV.1

1) Pode ser aplicado à solução x^3 , um movimento de piora ou melhoria. No segundo caso, a melhoria incidirá no custo total de x^3 , ou seja, a solução resultante x^* terá custo $c(x^*) = c(x^3) - OS^3$. Um próximo movimento de melhoria reduzirá este custo para $c(x^{**}) = c(x^*) - OS^*$.

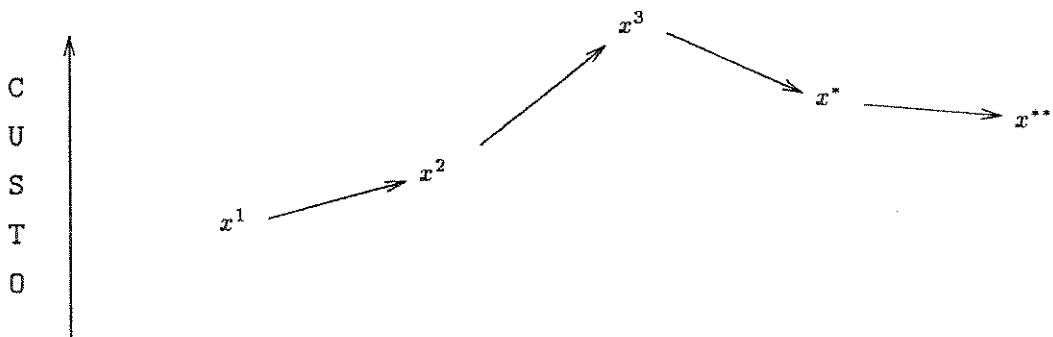


Figura IV.2

2) Por outro lado, se a partir de x^3 o movimento resultar no retorno a x^1 , necessariamente será realizado um movimento de piora. Supondo que a lista mantenha sob seu escopo os movimentos que levariam à geração de x^2 e x^3 , será selecionado um outro movimento.

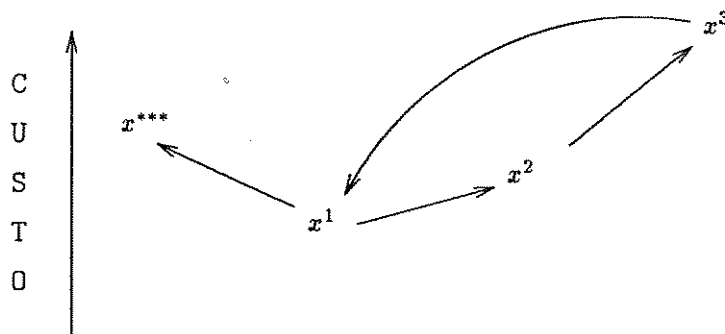


Figura IV.3

Neste caso, a piora incidirá no custo total de x^1 , isto é, $c(x^{***}) = c(x^1) - OS^1$.

Em alguns problemas, geralmente caracterizados por um número razoável de movimentos consecutivos de piora, as soluções x^{***} (obtidas sob o atributo economia) tenderam a ser mais baratas que x^{**} (obtidas sob o atributo custo). Esta tendência resultou nos casos em que o atributo custo foi superado.

Apesar do atributo economia ter apresentado um desempenho comparável ao custo de solução, o fato de não distinguir movimentos reversos equivalentes, nos fez optar, na etapa seguinte, por abandoná-lo. Mesmo quando o comprimento das listas pareceu ser

suficiente, não foi possível evitar a ocorrência de ciclagem e por conseguinte, o comprometimento da busca.

Uma nova bateria de testes foi então realizada, utilizando-se apenas o atributo custo. Foram aplicados a problemas dos grupos i e ii, que sendo exemplos clássicos, têm relatadas na literatura soluções obtidas por outros métodos heurísticos. Desta forma, foi possível ter uma idéia do desempenho do algoritmo proposto.

Sendo a ciclagem um aspecto de extrema importância no processo de busca, desejou-se inicialmente definir uma relação entre o tamanho do problema e o tamanho mínimo de lista para sua prevenção. Foi implementado um mecanismo que calcula sempre que um movimento candidato é classificado com tabu, o número de iterações entre a obtenção da solução recorrente em questão - e que o movimento candidato geraria, se efetivado - e a solução atual. Nos testes que envolveram todos os problemas dos grupos i e ii, foi adotado um tamanho de lista infinito (de forma a permitir as medições) e 500 iterações.

No que diz respeito à determinação da relação tamanho de lista/tamanho do problema, os resultados foram inconclusivos; entretanto foi possível verificar a necessidade de tamanhos de lista muito grandes, na ordem de 200 a 400 posições. Quanto à qualidade da solução final, observou-se que com relação à melhor solução apresentada na referência [7] para cada problema, houve ligeira superação em três casos, e nos restantes uma piora significativa, oscilando na faixa dos 0.8 a 5.5%.

Algumas conclusões podem ser extraídas destes resultados:

Pelo fato dos problemas tratados terem distribuição uniforme, são pouco frequentes

os casos de soluções diferentes com o mesmo custo. Em vista disso, pode-se dizer que a utilização do atributo custo promove essencialmente o impedimento de soluções previamente obtidas, de forma que praticamente está sendo adotada a primeira estratégia, elaborada com vistas à prevenção de ciclagem e apresentada no capítulo II. Esta estratégia exige o registro e checagem de todas as soluções, de forma que descartamos sua utilização em função dos requerimentos computacionais envolvidos nestas operações. Supôs-se que superada esta dificuldade, tivéssemos à disposição um mecanismo eficiente para empreender o processo de busca. Entretanto, apesar da ciclagem ter sido prevenida, foram obtidos resultados insatisfatórios no que diz respeito ao custo da solução final. Ou seja, o algoritmo não foi capaz de produzir soluções de qualidade, apesar da atuação precisa dos elementos restritivos.

Tais conclusões sugeriram que a Busca Tabu não se restringisse à prevenção de ciclagem como vinha sendo feito, mas que fosse utilizada com um propósito adicional (já mencionado por Glover) de obtenção de melhores soluções. Com o objetivo de promover níveis superiores de "esperteza", novos atributos passaram a ser considerados.

IV.4 - ATRIBUTO CONFIGURAÇÕES PARCIAIS DA SOLUÇÃO

Com o objetivo de analisar possíveis ganhos estratégicos através da proibição não apenas da reversão do movimento em questão, mas de conjuntos de movimento, definiu-se como atributo as configurações parciais da solução. Como configuração parcial entende-se

as regiões das rotas efetivamente alteradas com a operação. Sob esta diretiva é possível elaborar diferentes composições de lista que visam impedir a geração de soluções em que sejam verificadas tais configurações.

Glover menciona a aplicação da Busca Tabu a problemas de Caixeiro Viajante, utilizando a heurística 2-OPT e adotando como atributos cada uma das quatro arestas envolvidas [20,21]. Sempre que um movimento candidato implicar na adição de alguma das arestas previamente eliminadas ou na eliminação de alguma das arestas previamente adicionadas, estará passível de proibição. Se assumirmos que a relevância de cada aresta no que diz respeito à ciclagem é inversamente proporcional ao número de arestas envolvidas na operação, é razoável, como neste caso, utilizá-las individualmente como atributos do movimento.

Em se tratando de operações de inserção ou troca de nós, entretanto, ocorre implicitamente, a substituição de três e quatro arestas respectivamente; ou seja, seis ou oito arestas são manipuladas a cada movimento. De acordo com a premissa acima, a proibição de um movimento apenas porque ele resulta numa solução envolvendo uma ou duas arestas classificadas como tabu, é excessivamente restritiva e pode incorrer no alto risco de serem descartadas soluções promissoras. Portanto, ao se utilizar configurações parciais, buscou-se designar como atributos não uma aresta específica, porém grupos de arestas. Para este fim, foram elaboradas três diferentes composições de lista, cada uma representando níveis restritivos próprios.

Configuração A

A configuração é definida pelas arestas que foram *eliminadas* pela aplicação de um dado movimento. Como em toda operação estão envolvidas duas rotas k e l , foram sugeridas, a partir desta definição, as seguintes composições de lista:

i- *As arestas eliminadas em k ;*

As arestas eliminadas em l

ii- *As arestas eliminadas em k e l*

Caso um movimento implique na *adição das arestas eliminadas em k ou/e l* , será classificado como tabu.

A composição ii, como pode-se notar, indica uma estratégia menos restritiva que a de i, já que é necessário a ocorrência simultânea de duas situações para que se verifique o status tabu.

Configuração B

A configuração é definida pelas arestas *eliminadas e adicionadas* em cada rota pela aplicação do movimento. Sua implementação consiste na criação de uma lista tabu, que numa dada operação envolvendo as rotas k e l , deve armazenar:

iii- *As arestas eliminadas e adicionadas em k;*

As arestas eliminadas e adicionadas em l

Um movimento é classificado como tabu se implicar na *adição das arestas eliminadas e na eliminação das arestas adicionadas em k ou l.*

Antes de proceder à implementação, entretanto, foram verificadas limitações nestas configurações que resultaram em reincidência de estados-solução. Para um dado problema de 30 nós do grupo iii (já resolvido segundo o atributo custo), acompanhou-se passo a passo, a atuação de cada uma das configurações de lista. Observou-se que dado o seguinte movimento:

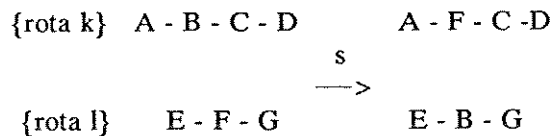


Figura IV.4

não haveria qualquer impedimento por parte dos três mecanismos restritivos para que a sequência:

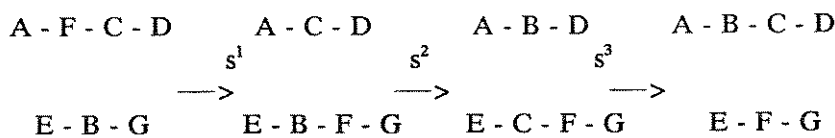


Figura IV.5

fosse realizada.

Através deste exemplo, verifica-se que assim como o atributo economia, estas composições de lista são incapazes de reconhecer movimentos reversos equivalentes. As composições i e ii não reconhecem que com o movimento s^3 , não apenas estariam presentes as arestas B - C, C - D e E - F, mas também A - B e F - G (o que conferiria o status tabu a s^3).

A composição iii, apesar de estruturalmente diferente, apresentou a mesma limitação. Sua aplicação a outros exemplos indicou ser esta a composição mais suscetível à reincidência de solução e ciclagem, seguida de ii e i. Aparentemente esta hierarquia segue o nível restritivo adotado, o que levou à elaboração de uma quarta composição de lista.

IV.5 - ATRIBUTO ARESTAS ADICIONADAS E ELIMINADAS

Utilizando-se os mesmos atributos de movimento apresentados na composição iii, foram criadas *duas listas tabu AD e DEL* para o armazenamento das *arestas eliminadas e adicionadas*, respectivamente. Ao contrário das outras composições, cada elemento das listas consiste de uma *única* aresta, o que confere um nível restritivo adicional a este mecanismo.

Um movimento é então classificado como tabu se sua efetivação implicar na *adição de arestas eliminadas OU na eliminação de arestas adicionadas* em pelo menos uma das rotas. Pode-se considerar dignos de proibição os movimentos em que se verifica a ocorrência de

pelo menos uma aresta tabu. Entretanto, observou-se que desta forma o procedimento seria rigoroso demais. Definiu-se, então, um parâmetro L correspondente ao número máximo de arestas tabu permitidas numa dada operação. Todo movimento cujo número de arestas proibidas for superior a L é classificado como tabu. Aplicando-se uma variação paramétrica de L sobre uma faixa conveniente de valores é possível definir a fronteira entre a severidade que impede configurações promissoras e a complacência que gera ciclagem.

Os testes iniciais foram aplicados a 4 problemas-teste dos grupos i e ii, de diferentes tamanhos, estruturas e características temporais. Ao invés de 500 iterações, a busca foi limitada a 500 movimentos de intercâmbio de nós. Foi assumido o tamanho das listas (t) igual a sete e valores de L entre 0 e 6. Entretanto, mesmo sob tais níveis restritivos, ocorreu ciclagem o que sugeriu o aumento de t , e subsequentemente sua variação paramétrica. Os testes foram então repetidos para tamanhos de lista 10, 20 30 e 40, assim definidos a partir dos resultados apresentados por J.A.G. Willard [45]. Segundo este autor, o tamanho de lista mínimo entre 20 e 40 parece ser apropriado para a maioria dos problemas por ele abordados, cujos tamanhos variavam entre 6 e 100 (clientes).

Os resultados obtidos definiram uma faixa de L e t onde as melhores soluções foram encontradas, por sua vez, utilizada na aplicação do algoritmo aos problemas restantes. O desempenho do atributo aresta foi, então, comparado ao do atributo custo quanto à qualidade de solução e tempo necessário para sua obtenção. Desta forma foi possível definir dentre estes, o atributo mais adequado. Definido o atributo, procedemos à análise de vários aspectos do comportamento do algoritmo e a comparações dos resultados com os já apresentados por outros métodos heurísticos.

CAPÍTULO V

RESULTADOS COMPUTACIONAIS E CONCLUSÕES

V.1 - INTRODUÇÃO

Neste capítulo são apresentados os resultados computacionais de interesse, obtidos ao longo do desenvolvimento do trabalho e dispostos em ordem cronológica.

Os custos representam a distância total percorrida pelo conjunto de rotas que compõem as soluções de menor custo. Estas são consideradas as melhores soluções, independente do tempo requerido para sua obtenção. Os tempos de CPU foram computados a partir da leitura ou geração de dados de entrada e consistem do tempo necessário para a obtenção da melhor solução encontrada no curso das iterações. Estabeleceu-se um número fixo de 500 ou 600 iterações (sob o atributo CUSTO) e 500 ou 600 movimentos de intercâmbio de nós (sob o atributo ARESTA), computados a partir do primeiro mínimo local.

O algoritmo foi implementado em linguagem Pascal nas versões disponíveis dos computadores utilizados. Para os problemas dos grupos i e ii, a implementação foi feita em Pascal SVS, sendo executado no supermicro Digirede MC 68020 (UNIX). Para problemas do grupo iii, o algoritmo foi, por sua vez, implementado em Pascal Berkeley e sua execução realizada na workstation Sun MicroSystem 3/60M - SunOS 4.0.3 (UNIX).

V.2 - CARACTERÍSTICAS DOS PROBLEMAS TRATADOS

A tabela abaixo resume as principais características dos problemas, para aos quais foi aplicada a implementação de Busca Tabu proposta.

GRUPO	PROBLEMA	TAMANHO	ESTRUTURA	CAPACIDADE	HORIZONTE DE TEMPO	TEMPO DE SERVIÇO	JANELA DE TEMPO
I	P1	50	uniforme	160	-	0	-
	P2*	75	uniforme	140	-	0	-
	P3	100	uniforme	200	-	0	-
	P4	150	uniforme	200	-	0	-
	P5*	120	agrupado	200	-	0	-
	P6	100	agrupado	200	-	0	-
II	P7*	50	uniforme	160	200	10	-
	P8	75	uniforme	140	160	10	-
	P9	100	uniforme	200	230	10	-
	P10	150	uniforme	200	200	10	-
	P11	120	agrupado	200	720	50	-
	P12*	100	agrupado	200	1040	90	-
III	P13	50	uniforme	100	800	120	300
	P14	50	uniforme	100	800	120	120
	P15	100	uniforme	200	1200	120	300
	P16	100	uniforme	200	1200	120	210
	P17	150	uniforme	300	1400	120	300
	P18	150	uniforme	300	1400	120	210
	P19	200	uniforme	300	1400	120	400
	P20	200	uniforme	300	1400	120	600

* - Problemas-teste

Tabela V.1

Cada problema é definido pelos seguintes dados de entrada:

1. Número de clientes (tamanho do problema)
2. Coordenadas retangulares de cada nó
3. Demandas dos clientes
4. Capacidade dos veículos
5. Tempos de serviço (grupos ii e iii)
6. Horizonte de tempo (grupos ii e iii)
7. Janelas de tempo para clientes (grupo iii)

Grupos i e ii

Os dados de entrada dos problemas dos grupos i e ii podem ser encontrados em [6] e [7]. São tidos como problemas difíceis e tradicionalmente utilizados na análise do desempenho de algoritmos.

Grupo iii

O tamanho, capacidade dos veículos e o horizonte de tempo de cada problema foram previamente especificados. Além disso, estabeleceu-se valores máximos para as coordenadas retangulares dos nós, demandas dos clientes e tempos de serviço. Estes dados foram então gerados dentro do intervalo definido por estes valores máximos. A cada conjunto de informações assim determinadas, foram especificados dois valores máximos de janela de tempo. Com isso, definiu-se pares de problemas que diferem entre si somente no que diz respeito às janelas de tempo dos clientes. Portanto, para este grupo, os tempos de

serviço e janelas de tempo apresentados na tabela anterior, correspondem aos valores máximos que estes parâmetros podem assumir.

As janelas de tempo foram assumidas como o período em que o serviço deve ser iniciado e completado. Não existem restrições ou penalidades sobre tempos de espera e não são considerados tempos de carregamento na garagem, nem paradas para descanso.

V.3 - ANÁLISES PRELIMINARES

Uma das etapas mais importantes deste trabalho consistiu de análises preliminares do comportamento do algoritmo sob os atributos CUSTO e ARESTA. Conforme mencionado no capítulo IV, os resultados obtidos permitiram a elaboração de conceitos essenciais à utilização eficiente da técnica de Busca Tabu, tais como a determinação da composição da lista, faixas de trabalho convenientes, etc.

Dado o grande número de problemas, estas análises foram aplicadas a quatro problemas-teste, selecionados dentre os grupos i e ii. A seleção buscou um subconjunto de problemas que representasse as várias características destes grupos, tais como tamanho, estrutura e existência (ou ausência) de restrições temporais. Os resultados obtidos foram, então, estendidos aos problemas restantes. As seções seguintes, resumem estas análises preliminares.

V.3.1 - Sensibilidade à solução de partida (atributo CUSTO)

Como já mencionado, heurísticas "hill climbing" são fortemente dependentes da solução inicial. Este fato pode ser visto como uma limitação, já que confere pouca robustez a estes métodos. Existe, portanto, interesse em determinar o grau de sensibilidade das soluções finais da Busca Tabu com relação às soluções de partida. Com este objetivo, os problemas-teste foram resolvidos (sob o atributo CUSTO), variando-se os parâmetros que definem critérios de construção das soluções iniciais. Os resultados para os critérios de construção C1, C2, C3 e C4 são dados na tabela abaixo (os números em cada célula representam Custo e Tempo (s)):

PROBLEMA	C1	C2	C3	C4
P2	860 * 463	860 * 463	879 256	893 196
P5	1062 * 281	1062 * 282	1069 5755	1069 5753
P7	573 465	573 465	567 * 624	567 * 624
P12	877 6238	872 2551	866 * 5507	868 868

*melhor solução do problema

- C1 - economia em *distância* ao se servir clientes em rotas já existentes, em oposição ao serviço individual
- C2 - economia em *tempo* ao se servir clientes em rotas já existentes, em oposição ao serviço individual
- C3 - critério C1, com penalização sobre a existência de rotas compostas de um único cliente
- C4 - critério C2, com penalização sobre a existência de rotas compostas de um único cliente

Tabela V.2

A diferença percentual entre a melhor e a pior solução de cada problema oscilou entre 0.7 a 3.8%. Estes resultados indicam que mesmo em algoritmos dotados de mecanismos de Busca Tabu, a solução de partida pode ter um papel relevante na qualidade das soluções geradas. Dentre os quatro critérios de construção, o algoritmo foi mais eficiente (custo de solução) com a adoção de C1 ou C2.

Os pares de resultados idênticos obtidos para os problemas p2, p5 e p7, advêm da geração de uma mesma solução de partida, apesar de terem sido adotados critérios diferentes. Não se verificou nenhuma situação em que, partindo de diferentes soluções iniciais, tenham sido obtidas soluções idênticas.

V.3.2 - Determinação da faixa de parâmetros (atributo ARESTA)

Com relação ao atributo ARESTA, existe um número razoável (virtualmente infinito) de valores que os parâmetros exógenos L (número máximo de arestas tabu toleráveis em um movimento) e t (comprimento da lista tabu) podem assumir. Com vistas à determinação de faixas de trabalho favoráveis à geração das melhores soluções, analisou-se o comportamento do algoritmo, variando-se estes parâmetros. Definimos como faixa inicial, L entre 0 e 6 e t igual a 10, 20, 30 e 40. Aplicando estes testes aos problemas p2, p5, p7 e p12, obtivemos os seguintes resultados (os números em cada célula representam

Custo e a melhor solução está ressaltada):
Mov. inter-rotas Tempo(s)

L t	10	20	30	40
0	890 0 29	881 293 682	890 0 25	893 0 25
1	886 35 96	869 195 417	877 33 99	890 0 25
2	854 35 95	851 62 155	867 218 489	880 397 947
3	861 10 49	860 19 72	868 131 294	863 460 1020
4	886 5 40	864 79 191	865 56 167	864 74 224
5	886 5 42	866 27 84	865 432 917	865 483 1012
6	886 5 40	866 43 115	871 43 115	866 239 523

Tabela V.3 - Problema P2

L t	10	20	30	40
0	1052 108 1266	1095 94 1323	1091 374 8644	1103 6 155
1	1090 30 430	1055 486 5815	1056 105 1501	1066 250 3996
2	1092 13 210	1049 148 1776	1051 253 3248	1050 92 1337
3	1062 24 410	1049 427 5395	1049 438 6069	1074 256 3849
4	1062 16 308	1061 136 1738	1050 371 4714	1049 236 2927
5	1062 18 297	1062 16 264	1054 75 885	1050 143 1652
6	1062 26 354	1062 40 476	1054 148 1593	1049 462 5160

Tabela V.4 - Problema P5

t	10	20	30	40
L				
0	596 36 122	578 348 1161	604 0 23	604 0 21
1	597 12 61	573 212 695	574 371 1178	598 4 37
2	596 10 50	573 321 1020	578 284 960	571 460 1557
3	596 10 51	571 296 944	571 345 1178	573 261 881
4	596 10 51	573 250 762	573 218 693	571 138 488
5	596 10 51	595 38 139	572 89 296	572 433 1405
6	604 2 32	596 11 51	572 308 986	572 375 1265

Tabela V.5 - Problema P7

L	t	10	20	30	40
0		873 51 831	874 108 1934	879 104 3089	883 0 109
1		872 87 1318	870 73 1170	867 131 2183	874 152 3482
2		870 67 1060	867 433 6678	868 371 6524	871 477 9000
3		871 59 901	866 453 6166	867 150 2268	867 499 7579
4		871 31 545	870 90 1298	868 353 4789	867 440 6192
5		871 29 531	871 40 673	872 127 2011	867 386 5335
6		870 67 973	870 38 639	869 59 903	866 214 2933

Tabela V.6 - Problema P12

No que diz respeito ao parâmetro L, os melhores resultados foram obtidos para os valores intermediários 2, 3 e 4. Tomando-se as 4 soluções mais baratas obtidas para cada problema (utilizando o tempo de CPU como critério de desempate entre custos idênticos), verificou-se que para p2, p5 e p7, todas foram obtidas sob algum destes valores. O problema p12, por outro lado, apresentou seus melhores resultados para L igual a 6, 3 e 5. No entanto, uma avaliação das melhores soluções seguintes, indica custos apenas ligeiramente maiores para L igual a 2, 3 e 4.

A partir de L igual a 5, verificou-se na maioria dos problemas a ocorrência de ciclagem e, como resultado, soluções de baixa qualidade. Resultados ainda piores foram

obtidos para L menor que 2, desta vez em função de suas condições excessivamente restritivas.

No que diz respeito ao tamanho de lista t , foi verificada ciclagem para todos os valores de L , com t igual a 10. Como resultado as soluções tendem a ser caras. Quanto aos outros tamanhos de lista, constatamos que seu efeito sobre a qualidade das soluções está vinculado ao valor do parâmetro L . Observou-se que para a faixa intermediária de L , as melhores soluções foram obtidas de forma praticamente homogênea para t igual a 20, 30 e 40. Para L maior que 4, por outro lado, melhores resultados foram obtidos com o aumento do tamanho da lista. Comportamento oposto verificou-se para L menor que 2.

Os dois últimos resultados são decorrentes da combinação L e t em questão, cujo efeito foi o de minimizar as limitações já mencionadas destes extremos. Ou seja, tamanhos pequenos de lista devem ser assumidos para condições muito restritivas ($L < 2$) e tamanhos grandes para condições muito relaxadas ($L > 4$).

Conforme mencionado por Glover [20,21], estratégias gulosas como a deste algoritmo, tendem a gerar boas soluções que geralmente compartilham elementos de decisão, como certas arestas ou grupos de arestas. Em situações restritivas (L pequeno), após um primeiro aparecimento nos estágios iniciais da busca, estas arestas são usualmente proibidas de serem incorporadas até que obtenham o status de elementos livres. Durante este período, o método não tem escolha senão incorporar arestas que não foram incluídas até então, por não fornecerem soluções atraentes. Para listas pequenas, estes efeitos são

minorados e soluções melhores encontradas.

Aparentemente, condições pouco restritivas (risco de ciclagem) fornecem soluções de melhor qualidade do que condições severas.

Estes resultados sugeriram uma faixa de L entre 2 e 4 e t igual a 20, 30 e 40, a ser aplicada aos problemas restantes. Não constatamos nenhuma relação entre os valores destes dois parâmetros e características dos problemas, tais como tamanho ou existência de restrições temporais.

A seguir, a representação gráfica dos resultados das tabelas V.3 a V.6.

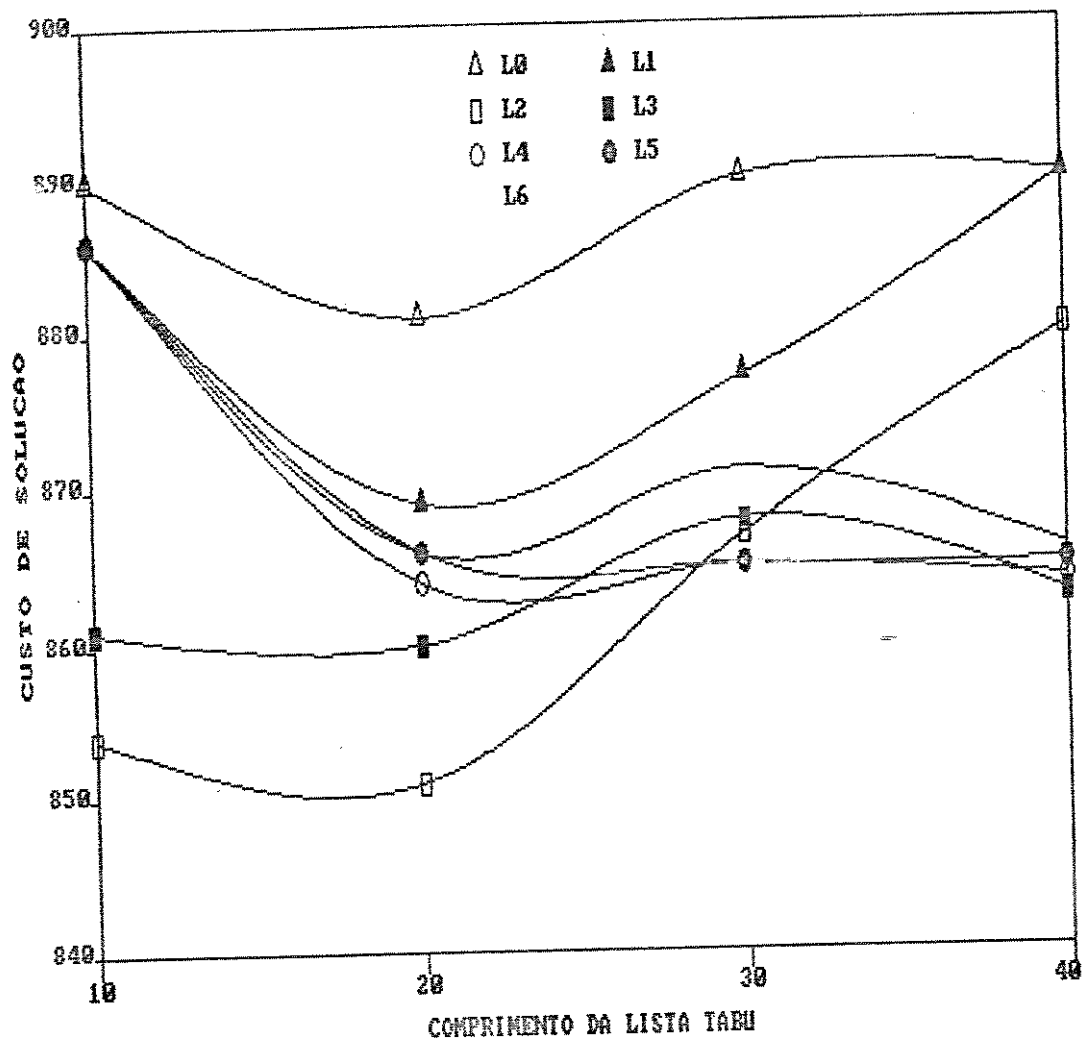


Figura V.1 - Problema P2

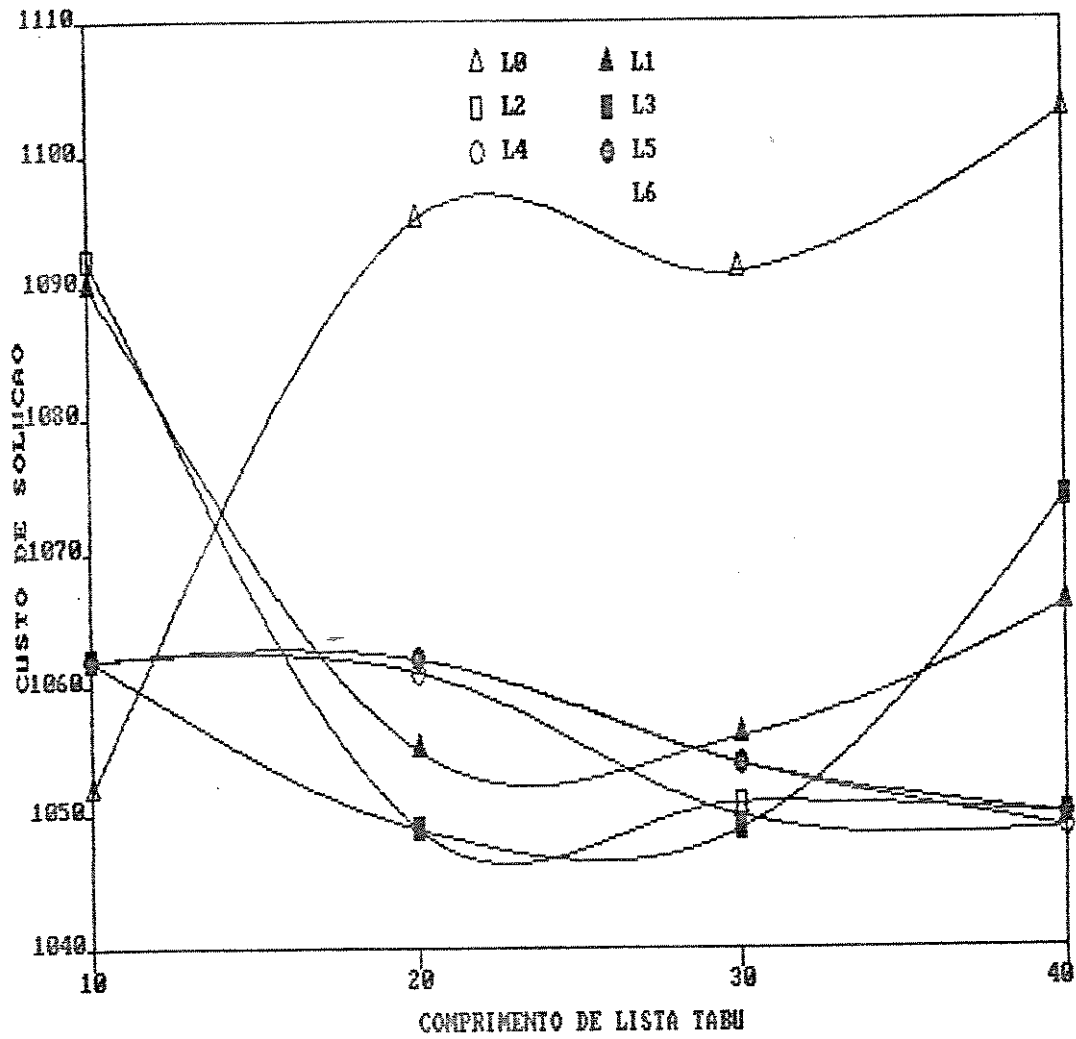


Figura V.2 - Problema P5

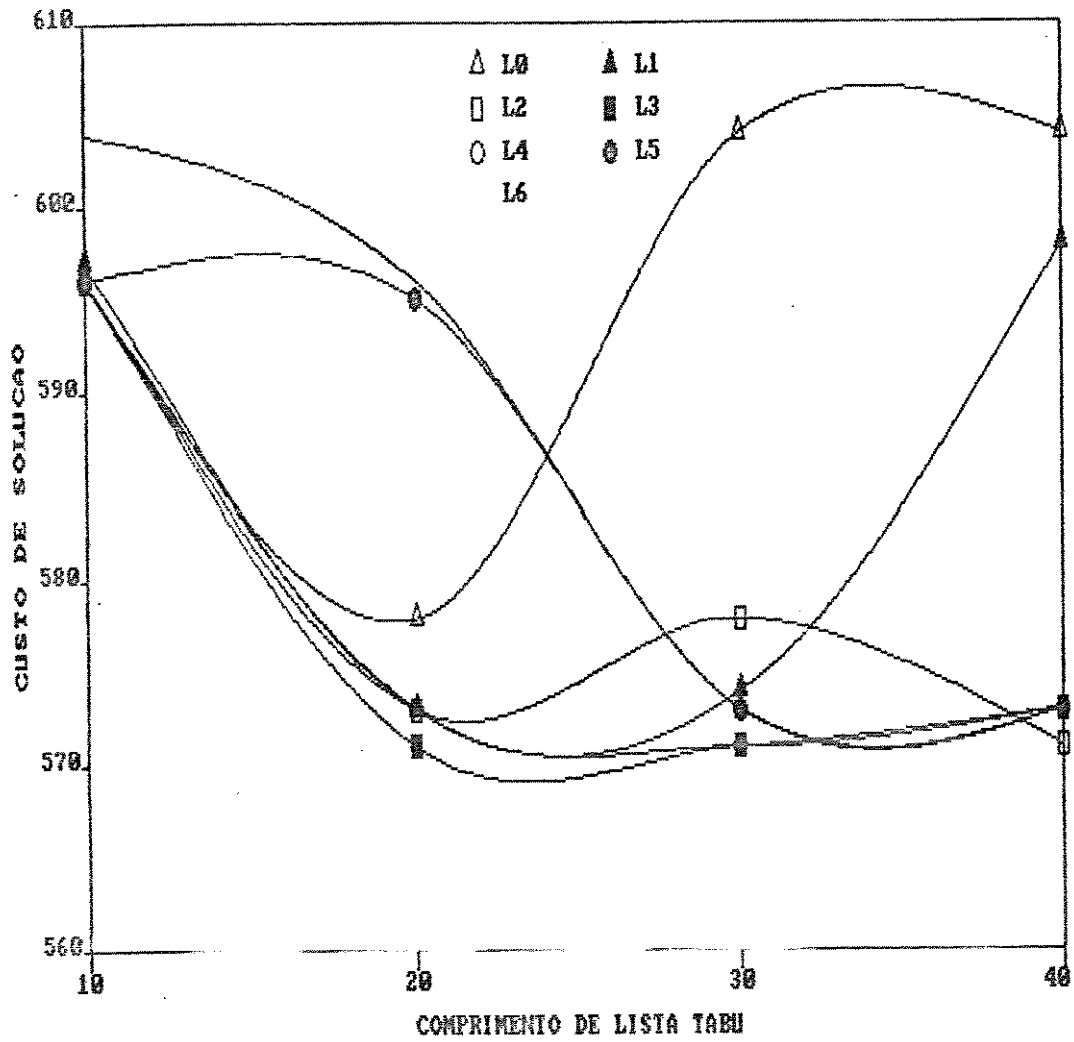


Figura V.3 - Problema P7

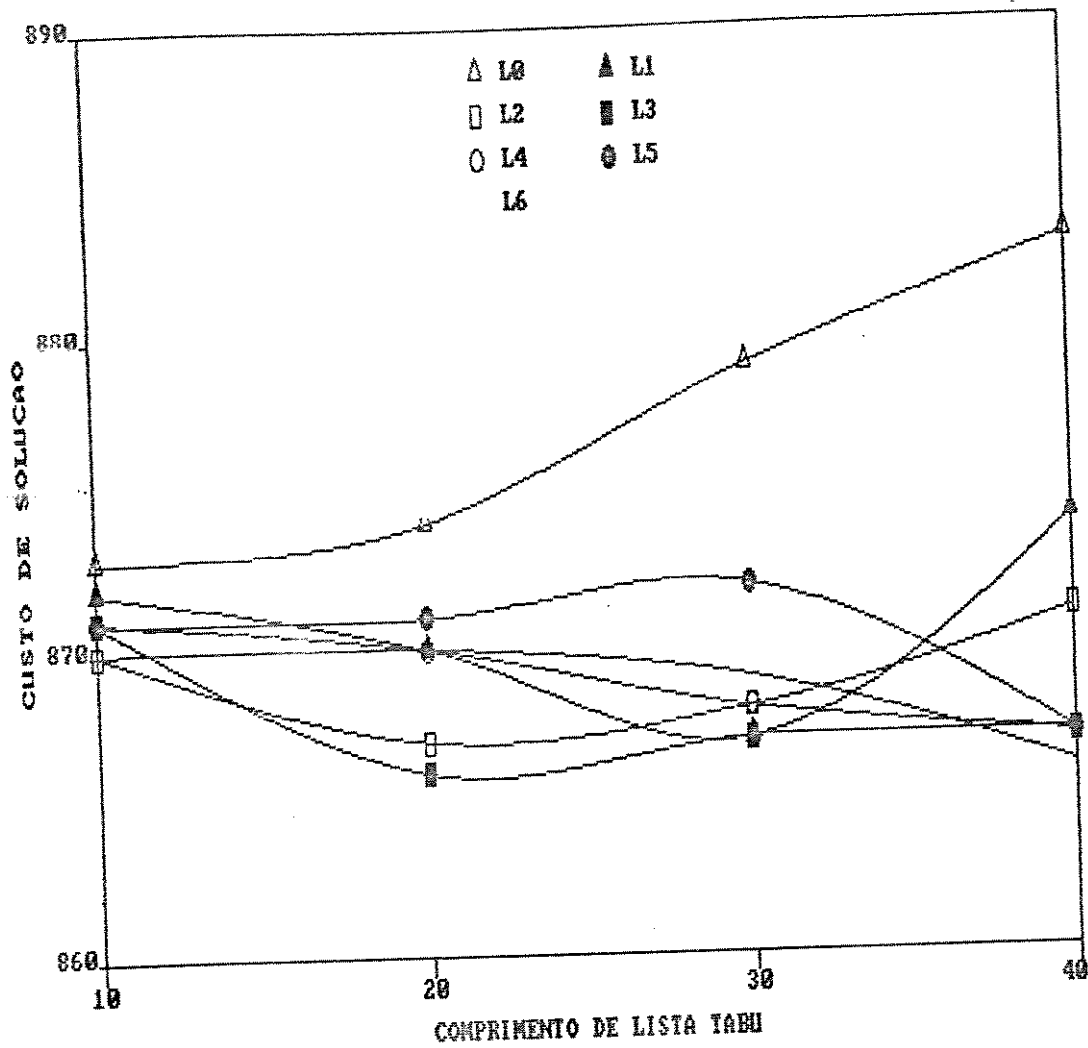


Figura V.4 - Problema P12

V.3.3 - Escolha do atributo

Com vistas à determinação do atributo mais efetivo, problemas dos grupos i e ii foram resolvidos, utilizando-se os atributos CUSTO e ARESTA. A tabela a seguir resume

os resultados obtidos (os números em cada célula representam Custo):
Mov. inter-rotas Tempo (s)
ou Iteração

PROBLEMA	ATRIBUTO	
	ARESTA	CUSTO
P1	* 535 487 973	553 97 -
P2	* 851 62 155	860 216 463
P3	* 855 197 1796	858 337 -
P4	* 1081 500 7351	1115 209 -
P5	* 1049 148 1776	1062 16 281
P6	* 826 336 3527	833 481 -
P7	571 460 1557	* 567 185 624
P8	* 929 393 2772	938 108 -
P9	* 887 364 7245	917 533 -
P10	* 1227 596 19376	1239 197 -
P11	* 1631 136 3501	1638 126 -
P12	* 866 453 6166	* 866 433 5507

*melhor solução

Tabela V.7

O atributo ARESTA apresentou desempenho superior quanto à qualidade das soluções geradas em dez dos doze problemas tratados. Nestes casos, os custos das soluções minimizaram os resultados correspondentes ao outro atributo de 0,3 a 3,3%. Apenas no problema p7, CUSTO conseguiu apresentar um melhor desempenho, e ainda assim, em apenas 0,7%.

Apesar de não se dispor da maioria dos tempos de CPU para CUSTO, as melhores soluções foram geralmente encontradas antes da metade do número de iterações designado. Este fato não deve ser considerado uma vantagem, mas sim uma limitação do atributo. Considerando que as soluções obtidas foram geralmente inferiores, tem-se aí configurado um processo de busca menos eficiente, no que diz respeito à exploração de regiões novas e atraentes. O atributo ARESTA, por outro lado, caracterizou-se pela descoberta contínua destas regiões, já que a maioria de suas melhores soluções foi encontrada nos últimos estágios da busca.

Estes resultados determinaram a escolha do atributo ARESTA, a ser utilizado nas etapas posteriores deste trabalho.

V.4 - COMPORTAMENTO DO ALGORITMO

V.4.1 - O processo de busca

A aplicação da técnica Tabu a um dado algoritmo resulta em comportamentos de

busca típicos, que podem ser percebidos pela análise da progressão do processo iterativo. Como exemplo, segue-se uma descrição detalhada do processo de busca apresentado pelo problema p2:

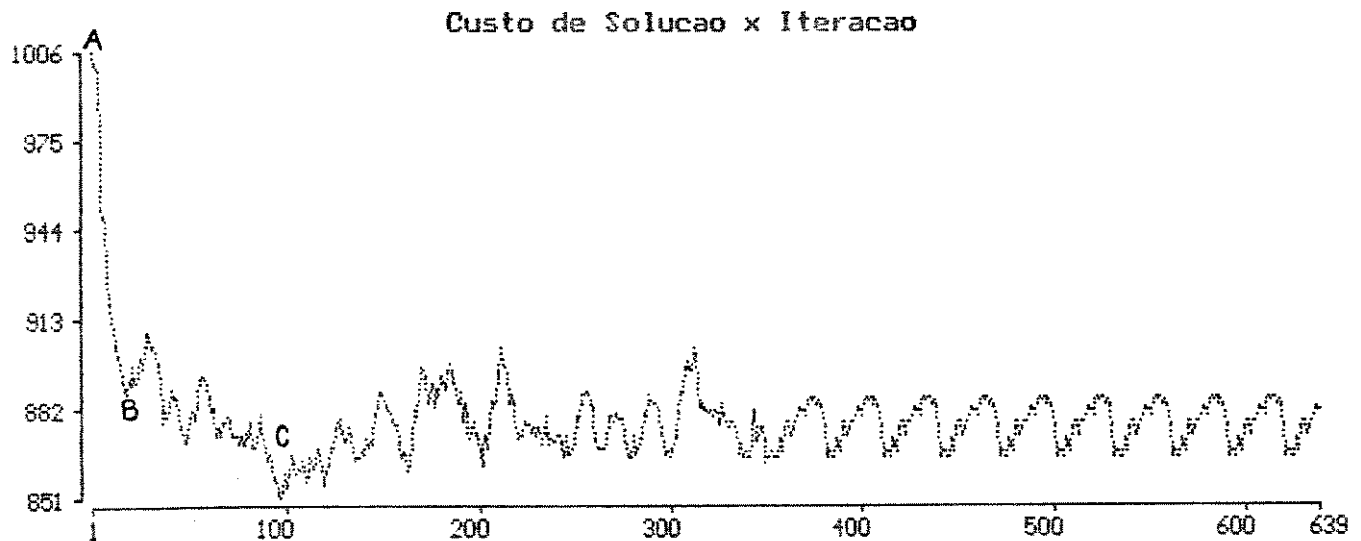


Figura V.5 - Problema P2

Observa-se que os custos de solução decrescem rapidamente, a partir do custo inicial de 1006 (ponto A). Há um decréscimo contínuo até a iteração 16, onde se obtém um custo de 888 (ponto B). Este ponto representa o primeiro mínimo local e, portanto, marca o início da fase tabu. Após um curto período, onde se verifica uma certa tendência ao aumento do custo de solução, obtém-se, a partir da iteração 40, dois mínimos que o

reduzem para 862. Durante as 25 iterações seguintes, observa-se certa oscilação. Por volta da octogésima iteração, a busca alcança uma região de boas soluções e inicia um processo contínuo de redução de custos. Na iteração 96, obtém-se a melhor solução do processo iterativo (ponto C), com custo de 851. A partir daí, o algoritmo mostra-se incapaz de produzir soluções de melhor qualidade. Os custos oscilam entre 853 e 896, até que na iteração 360, inicia-se um processo de ciclagem.

Nas seções seguintes, são investigados alguns aspectos dos processos de busca, apresentados pelos problemas-teste e por dois problemas do grupo iii (p13 e p19). Seguem-se as figuras correspondentes a estes processos:

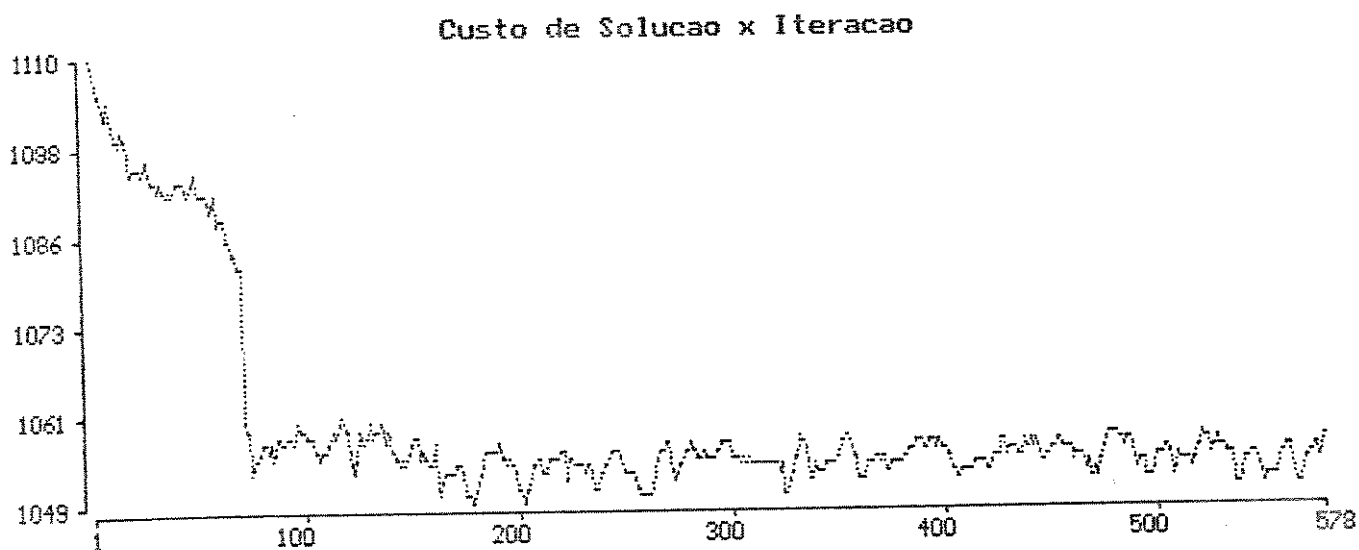


Figura V.6 - Problema P5

Custo de Solucao x Iteracao

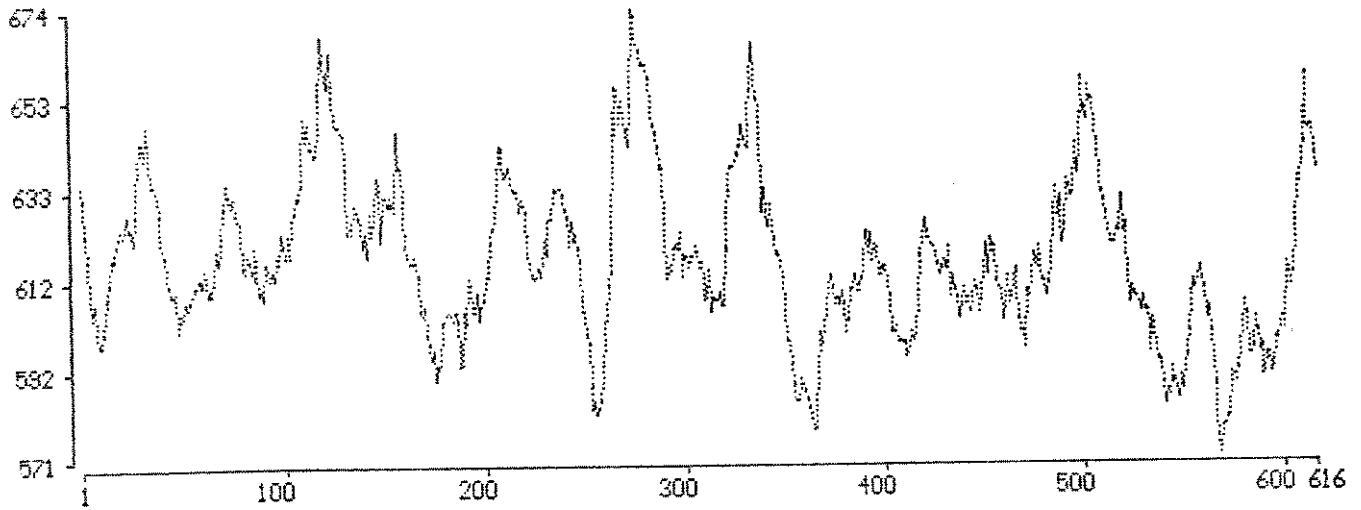


Figura V.7 - Problema P7

Custo de Solucao x Iteracao

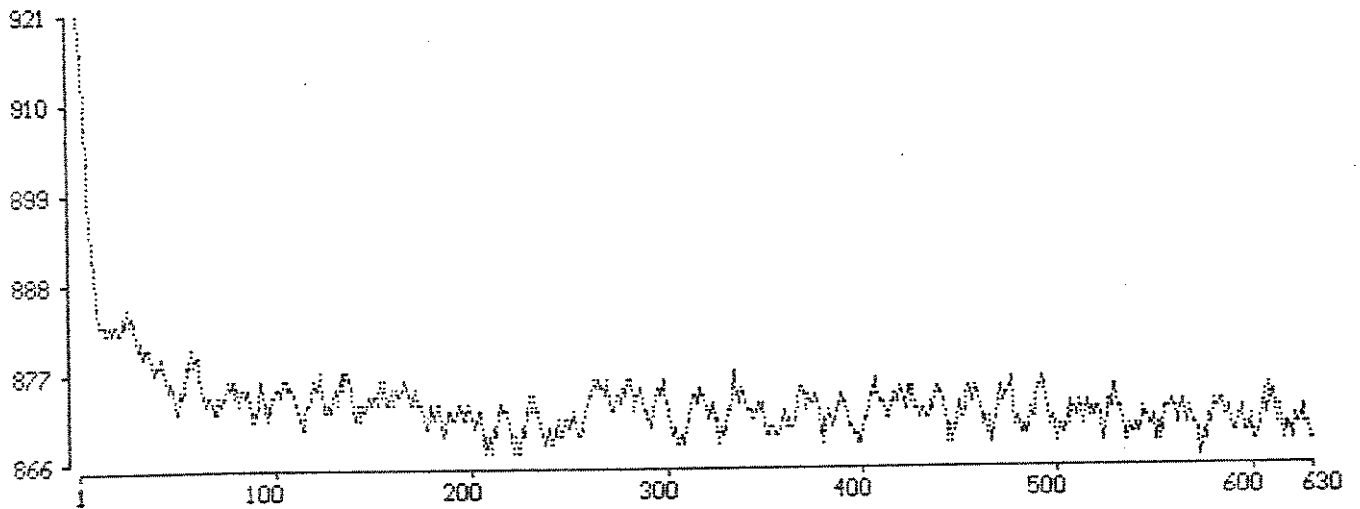


Figura V.8 - Problema P12

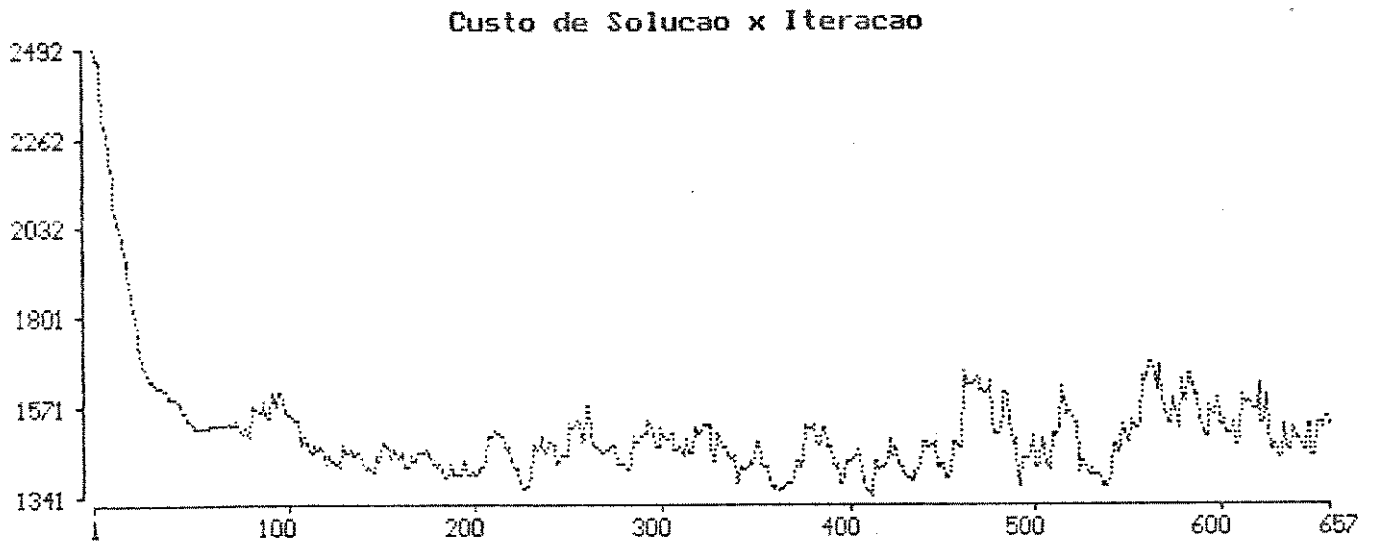


Figura V.9 - Problema P13

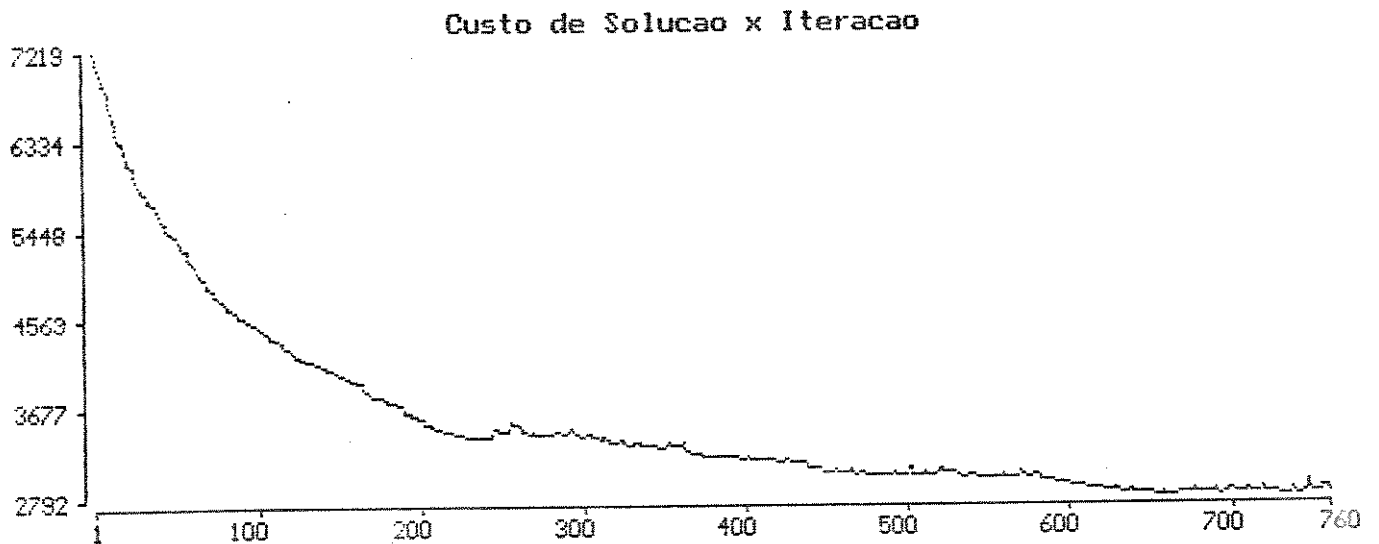


Figura V.10 - Problema P19

V.4.2 - Até o primeiro mínimo local

Os problemas-teste apresentaram de 4 a 16 iterações entre a geração da solução de partida e o primeiro mínimo local, ou seja, uma pequena parcela do número de iterações total. Comportamento oposto foi verificado para os problemas p13 e p19. De fato, foram necessárias 55 iterações para p13 e 234 para p19 atingirem o primeiro mínimo local.

Uma das hipóteses para tal comportamento envolve duas considerações. A primeira delas diz respeito à heurística de construção utilizada. Verifica-se que a aplicação de uma heurística sequencial de construção da solução de partida é particularmente eficiente à nível intra-rota. Entretanto, no que tange à solução total, observa-se certo desbalanceamento entre as rotas. Dada a construção sequencial e a estratégia gulosa empregadas, as rotas iniciais possuem um número maior de clientes, ao contrário das últimas que tendem a ser sub-utilizadas. Normalmente isso significa soluções resultantes que não correspondem a mínimos locais com relação a movimentos inter-rotas.

A segunda consideração diz respeito aos graus de restrição destes grupos de problemas. Verificou-se que nos estágios iniciais desta etapa, existe uma pequena parte do espaço de movimentos (factíveis e infactíveis) associada a economias relativamente grandes. Grupos i e ii, por suas características pouco restritivas, têm grande chance de que estes movimentos sejam também factíveis. Havendo tal disponibilidade e considerando a estratégia gulosa do algoritmo, estes movimentos serão naturalmente escolhidos. Como

grandes economias são obtidas, atinge-se o primeiro mínimo local em um pequeno número de iterações.

Problemas do grupo iii, por outro lado, caracterizam-se por um número menor de movimentos que resultam em configurações factíveis. Como são poucos os movimentos associados a economias relativamente grandes, existe maior probabilidade destes serem infactíveis. Consequentemente, o processo de busca necessita de um número maior de iterações para atingir um mínimo local. É interessante observar que sob as condições restritivas deste grupo, a solução de partida tende a ser ainda mais pobre do que a obtida nos grupos i e ii. Em função disso, o número de iterações tende a crescer com o número de rotas (tamanho do problema), pois neste caso, o desbalanceamento entre as rotas é ainda mais evidente.

V.4.3 - Após o primeiro mínimo local - Fase Tabu

Um aspecto de interesse consiste em se determinar as contribuições à qualidade da solução final, de faixas específicas de iterações. A verificação de um comportamento típico, permite estabelecer um número mínimo de iterações ao qual o processo de busca deve ser submetido para a obtenção de certos graus de redução de custo. Foram obtidos os seguintes resultados:

PROBLEMA	FAIXA DE ITERAÇÕES	$\frac{(Z_f - Z_m) * 100}{Z_f}$
P2	16 - 48	2,3
	48 - 96	4,2
P5	11 - 80	4,5
	80 - 163	4,6
	163 - 178	4,8
P7	4 - 10	1,0
	10 - 176	2,3
	176 - 256	3,5
	256 - 364	4,1
	364 - 570	5,5
P12	10 - 96	1,3
	96 - 184	1,5
	184 - 208	1,9
P13	55 - 79	1,4
	79 - 183	8,1
	183 - 225	10,9
	225 - 413	12,2
P19	234 - 400	7,0
	400 - 467	12,9
	467 - 600	15,2
	600 - 657	17,3

Z_f = Custo da melhor solução encontrada na faixa de iterações
 Z_m = Custo do primeiro mínimo local

Tabela V.8

As faixas de cada problema são regiões definidas por dois mínimos locais. O primeiro deles é o mínimo de menor custo da faixa anterior e o segundo, o próximo mínimo com custo menor que o do primeiro. A primeira iteração da primeira faixa de cada problema corresponde, portanto, a do primeiro mínimo local. A última iteração da última faixa corresponde a da melhor solução encontrada.

Neste conjunto de problemas, não se evidenciou a existência de faixas de iterações em que fosse possível generalizar o grau de redução com relação ao custo do primeiro mínimo local. Não obstante, verifica-se uma tendência ao aumento do grau de redução com o número de iterações nos problemas-teste. Comportamento oposto foi observado para p13 e p19.

As melhores soluções dos problemas-teste tanto foram encontradas nos estágios iniciais da busca (p2 e p5) como no seu final (p7 e p12). Os problemas p13 e p19, por outro lado, apresentaram seus melhores resultados depois da segunda metade do processo iterativo. É interessante observar que problemas com o mesmo comportamento, no que diz respeito à etapa de obtenção da melhor solução, são caracterizados pela presença (ou ausência) de restrições temporais.

V.4.4 - Ciclagem

Dentro das faixas de parâmetros L e t selecionada, foi constatada a ocorrência de ciclagem, em alguns problemas. As faixas em que isso ocorreu reforçam os resultados obtidos na análise preliminar. Ou seja, verificou-se ciclagem para valores grandes de L e tamanhos de lista igual a 20, e até 30. É interessante observar que o ponto de manifestação de ciclagem parece estar vinculado ao tamanho de lista. Quanto maior a lista, mais tardiamente a ciclagem se manifestou.

Problemas de menor porte mostraram-se os mais suscetíveis à ocorrência de

ciclagem. Devido ao menor número de movimentos disponíveis, estes tendiam a se repetir no curso das iterações, e sob condições relaxadas, eram efetivamente realizados.

V.5 - CONTRIBUIÇÃO DA BUSCA TABU

A tabela abaixo apresenta a contribuição da Busca Tabu à qualidade da solução final e os tempos de CPU requeridos:

PROBLEMA	$\frac{(Z_s - Z_t) * 100}{Z_s}$	$\frac{(Z_m - Z_t) * 100}{Z_m}$	TEMPO (s)
P1	12,3	8,6	973
P2	15,5	4,2	155
P3	18,8	4,3	1796
P4	16,7	4,6	7351
P5	5,4	4,8	1776
P6	8,6	6,8	3527
P7	10,0	5,5	1557
P8	2,8	2,5	2772
P9	12,5	7,5	7245
P10	7,2	2,9	19376
P11	3,3	0,5	3501
P12	5,9	1,9	6166
P13	46,2	11,5	367
P14	48,8	16,2	289
P15	57,9	10,6	1810
P16	57,1	18,8	1844
P17	41,8	20,4	2357
P18	56,7	19,6	3191
P19	61,3	17,3	8323
P20	59,5	7,7	9907

Z_s = Custo da solução de partida

Z_m = Custo do primeiro mínimo local

Z_t = Custo da melhor solução obtida na fase tabu

Tabela V.9

Os resultados indicam que na maioria dos problemas, a Busca Tabu teve um papel fundamental na qualidade das soluções geradas.

Do conjunto total, os problemas do grupo iii foram os grandes beneficiados. Com relação aos custos do primeiro mínimo, verificam-se economias de 15% em média. Não constatamos relação entre os ganhos obtidos e o tamanho do problema, o que sugere que sua utilização seja estendida a problemas ainda maiores, sem detrimento da eficiência do método. As altas economias com relação à solução de partida obtidas neste grupo, reafirmam a ineficiência da heurística de construção de rotas.

No que diz respeito aos tempos de processamento, deve ser considerado que o algoritmo proposto incorpora apenas os elementos da Busca Tabu simples, conforme definida por Glover [20,21]. Não foram utilizados mecanismos de níveis de aspiração ou funções de memória de médio e longo prazo. Além disso, o cálculo das inserções e trocas de nós é exaustivo e não descarta operações com pequenas chances de serem selecionadas. Outra característica desta implementação é que o movimento escolhido é sempre o melhor. Estas observações explicam o grande tempo computacional requerido.

V.6 - DESEMPENHO DO ALGORITMO COM RELAÇÃO A OUTROS MÉTODOS HEURÍSTICOS

A tabela V.10 apresenta uma comparação dos custos de solução dos problemas dos grupos i e ii obtidos pelo algoritmo proposto com os de outros sete métodos heurísticos, já publicados (os números em cada célula representam a distância total percorrida):

PROBLEMA	A	B	C	D	E	F	G	H
P1	585	575	*524	534	547	*524	556	536
P2	900	910	865	871	883	857	855	851 ♦*842
P3	886	882	851	851	851	*833	860	855
P4	1204	1259	1079	1064	1093	*1014	1089	1081
P5	1079	1100	1266	1092	1066	-	1052	*1049
P6	831	879	937	*816	827	*816	834	826
P7	619	599	*560	*560	565	*560	577	571 ♦*560
P8	976	969	933	924	969	*916	939	929
P9	973	999	888	*885	915	*885	918	887
P10	1426	1289	1230	1217	1245	-	*1210	1227
P11	1634	1590	1776	1608	1612	-	*1552	1631
P12	877	883	949	878	876	-	874	*866

* melhor solução

♦ solução obtida a partir de diferente solução de partida —

A - Método de Clarke & Wright [8]

B - Método de Mole & Jameson [31]

C - Método da Varredura [16]

D - Algoritmo de otimização incompleta [7]

E - Algoritmo de duas fases [7]

F - Algoritmo de Fisher e Jaikumar [13]

G - Algoritmo PSA3 de Altinkemer e Gavish [1]

H - Implementação de Busca Tabu [34]

Tabela V.10

O algoritmo proposto encontrou a melhor solução dos problemas P2, P5, P7 e P12.

Estas soluções superaram o melhor custo obtido por outros métodos em até 1,5%.

Quanto aos problemas para os quais o algoritmo não forneceu a melhor solução, verificou-se um ótimo desempenho. No que diz respeito aos métodos A, B, C, D, E e G, ele parece ser o mais regular em termos de qualidade de solução. Em sete problemas, o

algoritmo obteve custos que não excederam 2,3% em relação à melhor solução encontrada pelos outros métodos. Em apenas um problema (p11), obtivemos um custo 5,1% superior à melhor solução.

Quanto ao método F, encontramos custo até 2,6% superiores às soluções disponíveis. Em um problema (p4), foi obtido um custo de solução que excedeu a melhor solução em 6,6%.

V.7 - REDUÇÃO DO NÚMERO DE VEÍCULOS

Conforme apresentado no capítulo III, a estratégia de construção da solução inicial visa obter soluções de baixo custo variável (distância ou tempo). Não são incluídos custos resultantes da aquisição e manutenção de veículos, de maneira que não existe qualquer penalização sobre rotas adicionais.

No entanto, os custos fixos resultantes de uma rota adicional geralmente superam a economia em distância que esta venha a apresentar, especialmente se os veículos mantêm-se ociosos após terem completado suas rotas. Na prática, soluções que utilizam menos veículos são geralmente as mais atraentes.

Apesar de não terem sido utilizados mecanismos que promovessem (diretamente) a redução de veículos, foi possível verificá-la nas etapas subsequentes à geração da solução inicial. Em sete dos oito problemas do grupo iii, reduziu-se de uma a duas rotas da solução de partida. Em seis destes problemas, a redução de pelo menos uma rota foi

alcançada durante a fase tabu. Em quatro problemas, a melhor solução obtida também minimizava o número de rotas.

V.8 - RESULTADOS GERAIS

O objetivo deste trabalho é o estudo da aplicação da estratégia de Busca Tabu a contextos de roteamento de veículos. Para tal, foi elaborado um algoritmo que incorpora os mecanismos de Busca Tabu simples. Este algoritmo foi aplicado a vinte problemas de diferentes tamanhos, estruturas e níveis de restrição temporal. Seguem-se os principais resultados e conclusões:

1. Foi observado o desempenho do algoritmo sujeito a diferentes soluções de partida. Verificamos que mesmo com a utilização de mecanismos de Busca Tabu, a solução de partida pode ter papel relevante sobre a qualidade das soluções geradas.

2. Analisamos o desempenho do algoritmo quando submetido a diferentes composições de lista tabu. Constatamos que a utilização de uma composição de lista definida por atributos parciais do movimento (atributo ARESTA) - e que portanto proíbe um conjunto maior de movimentos - é mais eficiente em termos de qualidade de solução, do que composições que proíbem exatamente a reincidência de soluções (atributo CUSTO).

3. Durante o estudo da aplicação do atributo ARESTA, definimos faixas de trabalho dos parâmetros L (número de arestas tabu toleráveis em um movimento) e t (tamanho das listas tabu), favoráveis à geração das melhores soluções. Estas faixas apontam os valores intermediários de L (2,3 e 4) e t igual a 20, 30 e 40. Constatamos que estes valores definem

as fronteiras entre condições excessivamente restritivas (verificadas para $L < 2$) e condições muito relaxadas (verificadas para $L > 4$). Observamos que pequenos valores de L devem ser acompanhados de listas grandes; grandes valores de L , de listas pequenas. Tal procedimento minimiza os efeitos prejudiciais destas condições extremas.

4. Observamos ciclagem para todo L com t igual a 10, em função das condições relaxadas impostas por este último parâmetro. Em alguns problemas verificamos ciclagem para valores grandes de L e t igual a 20 e 30. Nestes casos, quanto maior a lista, mais tardiamente ocorria ciclagem. Problemas de menor porte foram os mais suscetíveis devido a menor disponibilidade de movimentos.

5. Não constatamos etapas da fase tabu (faixas de iterações) onde ocorram reduções específicas de custos.

6. Verificamos que a Busca Tabu promoveu nos 20 problemas tratados, reduções de 8,8% em média, em relação aos custos do primeiro mínimo local. Os problemas com janela de tempo foram os mais beneficiados; nestes casos, a média de ganhos obtidos cresce para 15%. Não constatamos relação entre as reduções obtidas nesta etapa e o tamanho do problema ou faixas de janelas de tempo. Isso sugere que a aplicação da estratégia a problemas maiores e mais restritos, não afetará os níveis de ganhos atingidos. Obtivemos tempos de processamento grandes, em função do cálculo exaustivo dos movimentos disponíveis.

7. O desempenho do algoritmo nos problemas dos grupos i e ii, foi comparado a outros métodos heurísticos. Verificamos que em três destes problemas, o algoritmo superou a melhor solução obtida pelos outros métodos em até 1,5%. Quanto aos problemas para os quais o algoritmo não forneceu a melhor solução, ele se mostrou um dos métodos mais regulares em termos de qualidade de solução, apresentando custos que, em sua maioria, não excederam 2,6% em relação às melhores soluções.

8. Mesmo sem possuir mecanismos que promovam diretamente a redução do número de veículos (rotas), o algoritmo reduziu de uma a duas rotas da solução inicial, em sete problemas dos grupo iii. Em seis dos problemas, a redução de pelo menos uma rota foi obtida na fase tabu.

V.9 - PERSPECTIVAS

1. O cálculo exaustivo dos movimentos de intercâmbio de nós, resultou em tempos de CPU excessivos. Acreditamos que a aplicação de heurísticas simples com vistas à redução do volume de inserções e trocas, pode levar a tempos de processamento menores, sem efeitos sobre a qualidade das soluções. Sugerimos, portanto, o desenvolvimento e utilização destas heurísticas.

2. Sob a perspectiva da sugestão anterior, sugerimos que com a redução dos tempos computacionais, sejam aplicados os mecanismos de Busca Tabu que neste trabalho não puderam ser explorados. Em especial, acreditamos que a utilização de funções de memória de médio e longo prazo e oscilação estratégica, pode trazer ganhos substanciais à qualidade das soluções.

REFERÊNCIAS BIBLIOGRÁFICAS

1. Altinkemer, K. & Gavish, B., "Parallel Savings Based Heuristics for the Delivery Problem", *Ops. Res* (a ser publicado).
2. Bodin, L., Golden, B., Assad, A. & Ball, M., "The State of the Art in the Routing and Scheduling of Vehicles and Crews", *Computers and Ops. Res.*, Vol. 9, pp 63-212 (1983).
3. Cerny, V., "Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm", *Journal of Optimization Theory and Applications*, Vol. 45, pp 41-52 (1985).
4. Cheshire, L. M., Malleson, A.M. & Naccache, P. F., "A Dual Heuristic for Vehicle Scheduling", *J. Opnl. Res. Soc.*, Vol. 33, pp 51-61 (1982).
5. Christofides, N., "The Vehicle Routing", in "The Traveling Salesman Problem" editado por Lawler, E. L., Lenstra, J. K., Rinnooy Kan, A. H. G., eds., pp 431-448, North-Holland (1985).
6. Christofides, N. & Eilon, S., "An Algorithm for the Vehicle Dispatching Problem", *Opnl. Res. Quart.*, Vol. 20, pp 309-318 (1969).
7. Christofides, N., Mingozzi, A. & Toth, P., "The Vehicle Routing Problem", *Combinatorial Optimization*, pp 315-338 (1979).

8. Clarke, G. & Wright, J., "Scheduling of Vehicles from a Central Depot to a Number of Delivery Points", *Ops. Res.*, Vol. 12, pp 568-581 (1964).

9. Cullen, F., Jarvis, J. & Ratliff, H. D., "Set Partitioning based Heuristic for Interactive Routing", *Networks*, Vol. 11, pp 125-143 (1981).

10. Dror, M. & Levy, L., "A Vehicle Routing Improvement Algorithm Comparison of a "Greedy" and a Matching Implementation for Inventory Routing", *Computers & Ops. Res.*, Vol. 13, pp 33-45 (1986).

11. Fisher, M. L., "Lagrangian Relaxation Method for Integer Programming Problems", *Management Science*, Vol. 27, pp 1-18 (1981).

12. Fisher, M. L. & Jaikumar, R., "A Decomposition Algorithm for Large-scale Vehicle Routing", *Decision Science Working Paper 78-11-05*, University of Pennsylvania (1978).

13. Fisher, M. L. & Jaikumar, R., "A Generalized Assignment Heuristic for Vehicle Routing", *Networks*, Vol. 11, pp 109-124 (1981)

14. Foster, B. A. & Ryan, D. M., "An Integer Programming Approach to the Vehicle Scheduling Problem", *Opnl. Res. Quart.*, Vol. 27, pp 367-384 (1976).

15. Gaskel, T. J., "Bases for Vehicle Fleet Scheduling", *Opnl. Res. Quart.*, Vol. 18, pp 281-295 (1967).
16. Gillet, B. E. & Miller, L. R., "A Heuristic Algorithm for the Vehicle-dispatch Problem", *Ops. Res.*, Vol. 22, pp 340-349, (1974).
17. Glover, F., "Future Paths for Integer Programming and Links to Artificial Intelligence", *Computers & Ops. Res.*, Vol. 13, pp 533-549 (1986).
18. Glover, F., "Heuristics for Integer Programming Using Surrogate Constraints", *Decision Sciences*, Vol. 8, pp 156-166 (1977).
19. Glover, F. & Greenberg, H. J., "New Approaches for Heuristic Search: A Bilateral Linkage with Artificial Intelligence", *European Journal of Opnl. Res.*, Vol. 39, pp 119-130 (1987).
20. Glover, F., "Tabu Search", Part I, *ORSA Journal on Computing*, Vol. 1, n. 3, pp 190-206 (1989).
21. Glover, F., "Tabu Search", Part II, *ORSA Journal on Computing*, Vol. 2, n. 1 (1989).

22. Glover, F. & McMillan, C, "The General Employee Scheduling Problem: An Integration of MS and AI", *Comput. & Ops. Res.*, Vol.13, pp 563-573 (1986).
23. Golden, B., Magnanti, T. & Nguyen, H., "Implementing Vehicle Routing Algorithms", *Networks*, Vol. 7, pp 113-148 (1977).
24. Hertz, A & de Werra, D., "Using Tabu Search Techniques for Graph Coloring", *Computing* 39, pp 345-351 (1987).
25. Holmes, R. A. & Parker, R. G., "A Vehicle Scheduling Procedure Based upon Savings and a Solution Perturbation Scheme", *Ops. Res. Quart.*, Vol. 27, pp 83-91 (1976).
26. Horowitz, E. & Sahni, S., "Fundamentals of Computer Algorithms", Computer Science Press Inc (1978).
27. Kirkpatrick, S., Gellatt Jr., C. D. & Vecchi, M. P., "Optimization by Simulated Annealing", *Science*, Vol. 220, pp 671-680 (1983).
28. Knowles, K., "The Use of a Heuristic Tree-search Algorithm for Vehicle Routing and Scheduling", *Opln. Res. Conf.*, Exeter, England (1967).

29. Lin, S., "Computer Solutions of the TSP", Bell System Tech. J., Vol. 44, pp 2245-2270 (1965).
30. Lin, S. & Kerningham, B. W., "An Effective Heuristic Algorithm for the TSP", Ops. Res., Vol. 21, pp 498-516 (1973).
31. Mole, R. & Jameson, S., "A Sequential Route Building Algorithm Employing a Generalized Savings Criterion", Opnl. Res. Quart., Vol. 27, pp 503-511 (1976).
32. Nelson, M. D., Nygard, K. E., Griffin, J. H. & Shreve, W. E., "Implementation Techniques for the Vehicle Routing Problem", Computers & Ops. Res., Vol. 12, pp 273-283 (1985).
33. Or, I., "Traveling Salesman Type Combinatorial Problems and their Relation to the Logistics of Regional Blood Banking", Tese de Doutorado, Dept. of Industrial Engineering and Management Sciences, Northwestern University (1976).
34. Pureza, V. & França, P., "Vehicle Routing Problems via Tabu Search Metaheuristic (submetido a Annals of Operations Research).
35. Savelsbergh, M., "Private Communication by Alexander Rinoooy Kan" (1984).

36. Solomon M. M., "An Algorithm for the Vehicle Routing and Scheduling Problems with Time Window Constraints", *Ops. Res.*, Vol. 35, pp 254-265 (1987).
37. Spaccamela, A. M. & Kan, A. H. G. R & Stougie L., "Hierarchical Vehicle Routing Problems", *Networks*, Vol. 14, pp 571-586 (1984).
38. Stewart, W. & Golden, B., "New Algorithms for Deterministic and Stochastic Vehicle Routing Problems", Tese de Doutorado, University of Maryland (1981).
39. Stewart, W. R. Jr., "Accelerated Branch Exchange Heuristics for Symmetric Traveling Salesman Problems", *Networks*, Vol. 17, pp 423-437 (1987).
40. Tillman, F. & Cochran, H., "A Heuristic Approach for Solving the Delivery Problem", *J. Ind. Engng.*, Vol. 19, pp 354-358 (1968).
41. Tillman, F. & Hering, R. D., "A Study of a Look-ahead Procedure for Solving the Multiple Delivery Problem", *Transportation Res.*, Vol. 5, pp 225-229 (1971).
42. Tyagi, M. S., "A Practical Method for Truck Dispatching Problem", *J. Ops. Res. Soc. Japan*, Vol. 10, pp 76-92 (1968).

43. Waters, C. D. J., "A Solution Procedures for the Vehicle Scheduling Problem Based on Iterative Route Improvement", *J. Opnl. Res. Soc.*, Vol. 38, pp 833-839 (1987).
44. Watson-Gandy, C. D. T. & Foulds, L. R., "The Vehicle Scheduling Problem: A Survey", *New Zealand Ops. Res.*, Vol. 9, pp 73-91 (1981).
45. Willard, J. A. G., "Vehicle Routing using r-Optimal Tabu Search", Tese de Mestrado, "Imperial College of Science, Technology and Medicine - University of London (1989).
46. Wren, A & Holliday A., "Computer Scheduling of Vehicles from One or More Depots to a Number of Delivery Points", *Opnl. Res. Quart.*, Vol. 23, pp 333-344 (1972).
47. Yellow, P. C., "A Computational Modification of the Savings Method of Vehicle Scheduling", *Opnl. Res. Quart.*, Vol. 21, pp 281 (1970).

V.9 - PERSPECTIVAS

1. O cálculo exaustivo dos movimentos de intercâmbio de nós, resultou em tempos de CPU excessivos. Acreditamos que a aplicação de heurísticas simples com vistas à redução do volume de inserções e trocas, pode levar a tempos de processamento menores, sem efeitos sobre a qualidade das soluções. Sugerimos, portanto, o desenvolvimento e utilização destas heurísticas.

2. Sob a perspectiva da sugestão anterior, sugerimos que com a redução dos tempos computacionais, sejam aplicados os mecanismos de Busca Tabu que neste trabalho não puderam ser explorados. Em especial, acreditamos que a utilização de funções de memória de médio e longo prazo e oscilação estratégica, pode trazer ganhos substanciais à qualidade das soluções.

REFERÊNCIAS BIBLIOGRÁFICAS