

Raqueline Azevedo Medeiros Santos

*Estudo e desenvolvimento de uma estratégia
de diversificação para melhoria de
performance da metaheurística Busca Tabu
usando Reconexão de Caminhos: estudo de
caso no problema de Localização de
Facilidades*

Natal – RN

Novembro / 2007

Raqueline Azevedo Medeiros Santos

*Estudo e desenvolvimento de uma estratégia
de diversificação para melhoria de
performance da metaheurística Busca Tabu
usando Reconexão de Caminhos: estudo de
caso no problema de Localização de
Facilidades*

Trabalho de Conclusão de Curso apresentado com a finalidade de obter o título de Bacharel em Ciência da Computação da Universidade Federal do Rio Grande do Norte.

Orientador:

Prof. D.Sc. Dario José Aloise

CURSO DE CIÊNCIA DA COMPUTAÇÃO
DEPARTAMENTO DE INFORMÁTICA E MATEMÁTICA APLICADA
CENTRO DE CIÊNCIAS EXATAS E DA TERRA
UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE

Natal – RN

Novembro / 2007

Trabalho de Conclusão de Curso sob o título “*Estudo e desenvolvimento de uma estratégia de diversificação para melhoria de performance da metaheurística Busca Tabu usando Reconexão de Caminhos: estudo de caso no problema de Localização de Facilidades*”, defendida por Raqueline Azevedo Medeiros Santos e aprovada em 30 de novembro de 2007, em Natal, Rio Grande do Norte, pela banca examinadora constituída pelos professores:

Prof. D.Sc. Dario José Aloise
Departamento de Engenharia de Produção - UFRN
Orientador

Prof. M.Sc. Luiz Amorim Carlos
Departamento de Informática e Matemática Aplicada -
UFRN

Profa. M.Sc. Cláudia Aparecida Cavaleiro Francisco
Departamento de Engenharia de Produção - UFRN

Agradecimentos

Como uma boa potiguar
E para não parecer clichê
É por meio destes versos
Que venho agradecer...

Aos meus pais, minha família
Que deram o suporte para minha educação
Que me ensinaram o valor da vida
Sempre com amor e atenção

Ao meu orientador, Prof. Dario
Que neste trabalho me guiou
Por sua atenção e paciência
Pelo seu tempo e tudo que me ensinou

Aos meus colegas de classe
Também não poderia esquecer
Que durante esses anos me acompanharam
Por sua amizade, quero agradecer

E para, enfim, terminar
Aquele que tenho grande apreço
Porque sem ele nada existiria
Ao Papai do Céu eu agradeço.

“As pessoas têm estrelas que não são as mesmas.”

Antoine de Saint-Exupéry

Resumo

Este trabalho descreve uma nova estratégia para melhoria de performance da metaheurística Busca Tabu com o intuito de explorar ainda mais o espaço de busca através do uso de procedimentos de Reconexão de Caminhos. Para isso, vários procedimentos de Reconexão de Caminhos são aplicados em pontos gerados simultaneamente pela estratégia de Diversificação por Reinício da Busca Tabu. Assim, é introduzido o funcionamento da Busca Tabu e suas principais características, assim como, a estratégia de Reconexão de Caminhos e suas variações. A implementação dessa nova estratégia é aplicada ao problema de Localização de Facilidades de p -medianas. Dessa forma, os resultados computacionais obtidos são utilizados para validar essa nova estratégia fazendo uma comparação com outra estratégia presente na literatura.

Palavras-chave: metaheurística, busca tabu, reconexão de caminhos, diversificação por reinício, localização de facilidades, p -medianas.

Abstract

This work describes a new strategy to obtain a better performance of the Tabu Search metaheuristic with the purpose of exploring the search space through the usage of Path Relinking procedures. For this purpose, various Path Relinking procedures are applied in points simultaneously generated by the Restart Diversification strategy of the Tabu Search. This way it is introduced how the Tabu Search works showing its key features and also the Path Relinking strategy and its variations. The implementation of this new strategy is applied to the p -median Location Facilities problem. Therefore, the computational results obtained are utilized to validate this new strategy making a comparison with another strategy of the literature.

Keywords: metaheuristic, tabu search, path relinking, restart diversification, location facilities, p -median.

Sumário

Lista de Figuras

Lista de Tabelas

Introdução	p. 11
1 Busca Tabu	p. 14
1.1 Espaço de Busca e Estrutura de Vizinhaça	p. 15
1.2 Lista Tabu	p. 15
1.3 Critério de Aspiração	p. 17
1.4 Um algoritmo simples	p. 17
1.4.1 Exemplo	p. 18
1.5 Critério de Parada	p. 20
1.6 Intensificação e Diversificação	p. 21
1.7 Tendências	p. 21
2 Reconexão de Caminhos	p. 22
2.1 Exemplo	p. 24
2.2 Outras estratégias e variações	p. 25
3 Localização de Facilidades	p. 27
3.1 Problema de p -medianas	p. 27
3.1.1 Modelo Matemático	p. 28
3.1.2 Exemplo	p. 29

3.1.3	Outras considerações	p. 29
4	Metodologia	p. 31
4.1	Diversificação por Reinício	p. 31
4.2	Estratégia de Aloise <i>et al</i>	p. 32
4.3	Uma estratégia mais robusta	p. 33
5	Experimentos Computacionais	p. 35
5.1	Algoritmo baseado na troca de facilidades	p. 35
5.1.1	Procedimento de Teitz e Bart	p. 36
5.1.2	Implementação de Whitaker	p. 37
5.2	Resultados	p. 39
5.2.1	Diversificações	p. 39
5.2.1.1	Critério de diversificação	p. 39
5.2.2	Validação da estratégia mais robusta	p. 39
5.2.3	Aplicando a intensificação	p. 43
	Considerações Finais	p. 45
	Referências	p. 46

Lista de Figuras

1	Realizando movimentos não aprimorantes para escapar de ótimos locais	p. 14
2	Solução inicial	p. 16
3	Realização da troca entre a tarefa 7 e a tarefa 5	p. 16
4	Estrutura da lista tabu	p. 16
5	Exemplo de Busca Tabu no problema de sequenciamento de tarefas .	p. 19
6	Caminho gerado utilizando Reconexão de Caminhos	p. 22
7	Ilustração da Reconexão de Caminhos	p. 23
8	Problema do Caixeiro Viajante	p. 24
9	Ilustração da Reconexão de Caminhos aplicada ao problema do Caixeiro Viajante	p. 25
10	Problema de p -medianas	p. 29
11	Diversificação por Reinício	p. 31
12	Estratégia de Aloise <i>et al</i>	p. 32
13	Explosão de diversificações	p. 33
14	Grafo aplicado ao problema de 2-medianas com facilidades = {B,E} e clientes = {A,C,D}	p. 36
15	Média de sucessos da aplicação da Reconexão de Caminhos em função do número de iterações não aprimorantes antes da diversificação . . .	p. 42
16	Número de sucessos da aplicação da Reconexão de Caminhos em função da diversificação realizada	p. 42

Lista de Tabelas

1	Descrição dos clientes do grafo da Figura 14	p. 37
2	Porcentagem das tentativas de Reconexão de Caminhos com sucesso .	p. 40
3	Tempo médio de execução (em segundos)	p. 43
4	Média das soluções	p. 44

Introdução

O termo “metaheurística” surgiu em 1986 no mesmo artigo que introduziu a Busca Tabu por Fred Glover.

Uma metaheurística é, na verdade, uma estratégia que guia e modifica outras heurísticas para produzir soluções além daquelas que são normalmente geradas numa busca pelo ótimo local. (GLOVER e LAGUNA, 1993, p. 17).

Ou, como afirma Glover (1996a), as metaheurísticas, na sua definição original, são métodos que orquestram a interação entre procedimentos de melhoras locais e estratégias de alto nível para criar um processo capaz de escapar de um ótimo local e realizar uma busca mais robusta no espaço de soluções.

Dessa forma, as metaheurísticas são bastante utilizadas em problemas de otimização combinatória NP-difíceis. Esses problemas caracterizam-se pela inviabilidade da exploração de todas as possíveis soluções em tempo polinomial. Assim, a metaheurística permite explorar apenas uma parte desse espaço de soluções, evidenciando regiões que visem o aprimoramento da solução.

Durante as últimas décadas, diversas metaheurísticas foram surgindo. Como podemos ver na literatura, algumas das principais são: Algoritmos Genéticos (1970), *Simulated Annealing* (1983), Busca Tabu (1986), GRASP (*Greedy Randomized Adaptive Search Procedure*) (1989), VNS (*Variable Neighborhood Search*) (1997), entre outras.

Segundo Glover e Laguna (1993), as metaheurísticas podem ser classificadas de acordo com três fatores:

1. o uso de memória adaptativa;
2. o tipo de exploração de vizinhança utilizado;
3. o número de soluções correntes levadas de uma iteração para a próxima.

A metaheurística Busca Tabu, por exemplo, emprega memória adaptativa; utiliza algum tipo de busca sistemática na vizinhança que permite selecionar o próximo mo-

vimento e/ou aprimorar a solução corrente; e pode tanto mover de uma solução para outra, a cada iteração, como também, ter uma abordagem baseada em população.

Além das metaheurísticas, estratégias auxiliares foram sendo desenvolvidas com o objetivo de permitir a obtenção de melhores soluções durante a busca. A Reconexão de Caminhos (*Path Relinking*) é uma dessas estratégias. Ela permite gerar soluções explorando caminhos entre uma solução corrente e uma solução elite, através de movimentos que introduzem atributos da solução elite na solução corrente.

No decorrer deste trabalho estaremos propondo uma nova estratégia de utilização da Reconexão de Caminhos na metaheurística Busca Tabu com o objetivo de introduzir melhorias na busca de uma boa solução. A idéia é explorar ainda mais o espaço de busca através do uso da Reconexão de Caminhos em pontos gerados simultaneamente pela estratégia de Diversificação por Reinício da Busca Tabu. A validação da nossa estratégia é feita através da sua aplicação no problema de localização de facilidades de p -medianas.

Este trabalho encontra-se dividido da seguinte forma:

O capítulo 1 descreve a metaheurística Busca Tabu, suas principais características, tais como, a estrutura da lista tabu, a exploração de vizinhança, conceitos de intensificação e diversificação. Além de explicar seu funcionamento através de um algoritmo simples.

No capítulo 2, apresenta-se o funcionamento da estratégia de Reconexão de Caminhos através de exemplos e discute-se um pouco sobre os métodos encontrados na literatura e suas variações.

Em seguida, no capítulo 3, fala-se um pouco sobre o problema de localização de facilidades indicando os principais problemas existentes, em especial, o problema de p -medianas, que será utilizado para validar nossa estratégia.

A metodologia empregada é descrita no capítulo 4. Mas antes, tem-se uma breve explicação da técnica de Diversificação por Reinício, que é utilizada pela Busca Tabu, e apresenta-se uma estratégia de melhoria, desenvolvida por Aloise *et al* (2006), a qual se baseia nesse trabalho.

O capítulo 5 mostra os experimentos computacionais realizados para o problema de p -medianas, assim como os resultados obtidos. Além disso, explica como foi desenvolvido o nosso algoritmo e quais as teorias que o fundamentaram. Depois disso, é feita uma comparação entre as duas estratégias que serão apresentadas no decorrer

deste trabalho.

Por fim, tem-se a conclusão e sugestões para trabalhos futuros.

1 Busca Tabu

A Busca Tabu é um método de busca local proposto por Glover em 1986. De acordo com Glover e Laguna (1997), a filosofia da Busca Tabu está na exploração de uma coleção de princípios de resolução inteligente de problemas. Dessa forma, uma das principais características desse método ou metaheurística, é que ele escapa de ótimos locais permitindo a realização de movimentos não aprimorantes. Isso significa que essa técnica abre espaço no conjunto de soluções percorrendo-o de forma mais abrangente.

A Figura 1, a seguir, mostra como a realização de movimentos não aprimorantes permite que a metaheurística escape de ótimos locais.

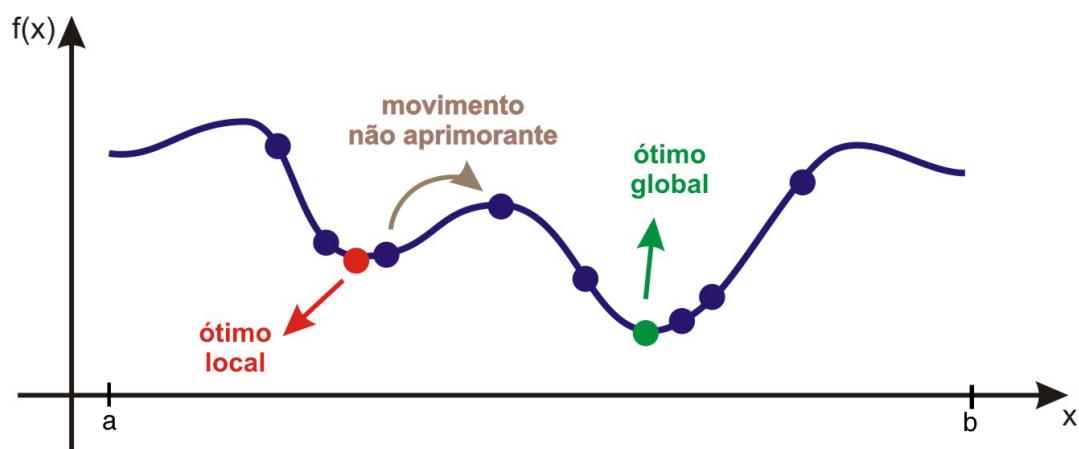


Figura 1: Realizando movimentos não aprimorantes para escapar de ótimos locais

A Busca Tabu é muito utilizada em problemas de otimização combinatória em várias áreas, tais como, telecomunicações, distribuição de energia, exploração mineral, localização de facilidades, entre outras.

A seguir, veremos uma descrição dos principais itens que compõem essa metaheurística, mostrando seu funcionamento. Em adição, apresentaremos como essa técnica pode ser estruturada através de um algoritmo simples e, também, como são aplicados os conceitos de intensificação e diversificação.

1.1 Espaço de Busca e Estrutura de Vizinhança

Segundo Gendreau (2003), os dois elementos básicos de qualquer Busca Tabu está na definição do seu espaço de busca e da estrutura de vizinhança.

O espaço de busca pode ser definido, simplesmente, como todas as possíveis soluções que podem ser visitadas durante a busca. Por exemplo, o espaço de busca pode ser o conjunto de soluções viáveis para o problema dado. “É importante notar que nem sempre é uma boa idéia restringir o espaço de busca apenas para soluções viáveis. Em muitos casos, é desejável que a busca também caminhe por soluções inviáveis” (GENDREAU, 2003, p. 41).

Já a estrutura de vizinhança, é definida por uma transformação local a ser aplicada, a cada iteração, na solução corrente. Dessa forma, dada a solução corrente S , podemos definir $N(S)$ (a vizinhança de S) como um subconjunto do espaço de busca composto pelas soluções obtidas pela aplicação da transformação em S . Segundo Glover e Laguna (1993), cada solução $S' \in N(S)$ pode ser alcançada diretamente por S através de uma operação, que é chamada de movimento.

Portanto,

[...] escolher o espaço de busca e a estrutura de vizinhança é o passo mais crítico na construção de uma heurística de Busca Tabu. Pois é nesse estágio que devemos fazer o melhor uso do entendimento e conhecimento do problema que temos em mãos. (GENDREAU, 2003, p. 42).

1.2 Lista Tabu

Como vemos em Gendreau (2003) e em Glover e Laguna (1997), a Busca Tabu utiliza uma memória de curto prazo, também chamada de lista tabu. Essa lista impede que o algoritmo gere ciclos durante a busca, já que ela armazena os movimentos recentes, impedindo que sejam executados novamente.

Vamos considerar o problema do sequenciamento de tarefas (*jobs*), como sugerido em Glover e Laguna (1993), que consiste em determinar a melhor sequência para o processamento de várias tarefas numa máquina. Vamos supor que a solução inicial para o nosso problema é a solução mostrada pela Figura 2.

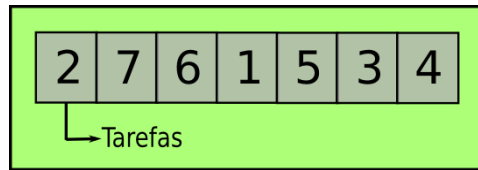


Figura 2: Solução inicial

A ordem na Figura 2 indica que a tarefa 2 será executada primeiro, seguida pela tarefa 7 e, assim, sucessivamente. Nesse problema, o vizinho de uma solução é obtido pela troca de posições de duas tarefas, como é ilustrado na Figura 3. Portanto, a vizinhança será formada por um total de 21 soluções, que podem ser obtidas através dessa operação de troca.

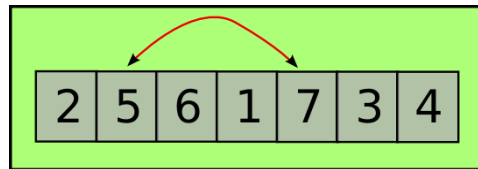


Figura 3: Realização da troca entre a tarefa 7 e a tarefa 5

A troca de posições numa solução pode ser representada por um par de elementos. E, assim, como uma forma de prevenir que durante a busca sejam repetidos os mesmo movimentos de troca tentados num passado recente, classificaremos como tabu, somente para exemplificar, os três movimentos mais recentes, ou seja, cada movimento estará definido como tabu durante 3 iterações. Como realizar a troca das tarefas 7 e 5 é a mesma coisa que trocar 5 por 7, ambas possuirão a mesma representação: (5,7). Então, a estrutura da lista tabu para esse exemplo é ilustrada pela Figura 4.

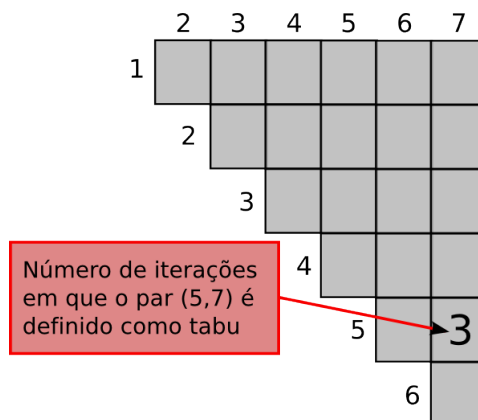


Figura 4: Estrutura da lista tabu

Cada célula da estrutura contém o número de iterações que faltam para que o movimento correspondente possa ser realizado novamente. Assim, caso a célula (2,3)

tenha o valor zero, então as tarefas 2 e 3 estão livres para trocar de posições. Porém, se a célula (2,5) possuir o valor 2, por exemplo, então, as tarefas 2 e 5 não poderão trocar de posições pelas próximas 2 iterações.

Como podemos ver, a implementação da lista tabu pode ser bastante simples. Entretanto, segundo Gendreau (2003), podemos ter algumas variações, como a utilização simultânea de várias listas tabus e, também, a variação no seu tamanho, já que nem sempre listas tabus de tamanho fixo impedem a geração de ciclos durante a busca.

1.3 Critério de Aspiração

Apesar de geralmente impedir a geração de ciclos, a lista tabu pode impedir que soluções ainda não visitadas sejam excluídas da busca, ou seja, quando um movimento está marcado como tabu, mesmo que ele resulte numa solução de menor custo, ele não poderá ser realizado. Então, para solucionar esse problema, utiliza-se o critério de aspiração.

A função desse critério é revogar a condição de tabu para um certo movimento, ou seja, ele permite que um movimento, considerado tabu, seja realizado, caso a solução por ele gerada seja melhor que a melhor solução já encontrada. Portanto, o critério de aspiração pode evitar a perda de boas soluções (GLOVER e LAGUNA, 1997).

1.4 Um algoritmo simples

A seguir, como descrito em Gendreau (2003), temos o Algoritmo 1, descrevendo uma metaheurística de Busca Tabu. Trata-se de um algoritmo simples (não incluindo nenhuma estratégia de diversificação nem de intensificação) para termos uma idéia de como funciona essa metaheurística.

Vamos supor que estamos tentando minimizar uma função $f(S)$ e que a cada iteração o algoritmo seleciona a melhor solução.

Utilizaremos a seguinte notação:

- $S \rightarrow$ a solução corrente;
- $S^* \rightarrow$ a melhor solução encontrada;
- $f^* \rightarrow$ valor de S^* ;

- $N(S) \rightarrow$ a vizinhança de S ;
- $\tilde{N}(S) \rightarrow$ é o subconjunto admissível de $N(S)$ (que não é tabu ou é permitido pelo critério de aspiração).

Algoritmo 1 *Busca_Tabu*

```

1: {Inicialização}
2: Construa uma solução inicial  $S_0$ 
3:  $S \leftarrow S_0$ ;
4:  $f^* \leftarrow f(S_0)$ ;
5:  $S^* \leftarrow S_0$ ;
6:  $T \leftarrow \emptyset$ ; {Inicialize a Lista Tabu como vazia}
7:
8: {Busca}
9: while critério de parada não é satisfeito do
10:   Selecione  $S$  em  $\operatorname{argmin}[f(S')]$ ; onde  $S' \in \tilde{N}(S)$ 
11:   if  $f(S) < f^*$  then
12:      $f^* \leftarrow f(S)$ 
13:      $S^* \leftarrow S$ 
14:   end if
15:   Armazene como tabu o movimento atual em  $T$ 
16: end while

```

Então, como podemos perceber, o algoritmo é constituído de duas partes: inicialização e busca. Na inicialização, o algoritmo constrói uma solução inicial que, na maioria das vezes, é gerada aleatoriamente. Depois disso, ele inicializa a lista tabu e os valores de S , f^* e S^* .

Na fase de busca, a cada iteração o algoritmo seleciona a melhor solução na sua vizinhança para continuar a busca a partir dela (linha 10). E depois, na linha 11, ele verifica se a nova solução é melhor que a melhor solução. Caso seja, ele atualiza o valor da melhor solução. Bom, até esse ponto, isso não difere em nada de um método de busca local. Entretanto, se analisarmos melhor, vemos que o algoritmo não seleciona a nova solução em toda a vizinhança e, sim, num subconjunto dela. Esse subconjunto vai depender da lista tabu e do critério de aspiração. No final da iteração, indicado pela linha 15, o algoritmo armazena como tabu o movimento realizado.

1.4.1 Exemplo

Seguindo a idéia exposta pelo Algoritmo 1, temos a seguir um exemplo aplicado ao problema de sequenciamento de tarefas. Apesar de termos definido esse problema anteriormente, segue uma descrição mais formal, como vemos em Glover e Laguna

(1993). O objetivo desse problema é encontrar uma determinada sequência de execução de tarefas numa máquina de forma a minimizar o tempo de processamento. Nesse problema, cada tarefa j ($j = 1, \dots, n$) possui um tempo de processamento p_j , um tempo máximo para sua execução d_j e uma penalidade w_j . Dessa forma, o objetivo é minimizar

$$T = \sum_{j=1}^n w_j [C_j - d_j]^+$$

onde C_j é o tempo de término do processamento da tarefa j , ou seja, $C_j = p_j + C_{j-1}$; e, para um dado valor v , $[v]^+ = \max(0, v)$.

Na Figura 5 mostramos a execução de 3 iterações do algoritmo de Busca Tabu para um conjunto de 4 tarefas com $p = [6, 4, 8, 2]$, $d = [9, 12, 15, 8]$ e $w_j = 1$ para $j = 1, \dots, 4$.

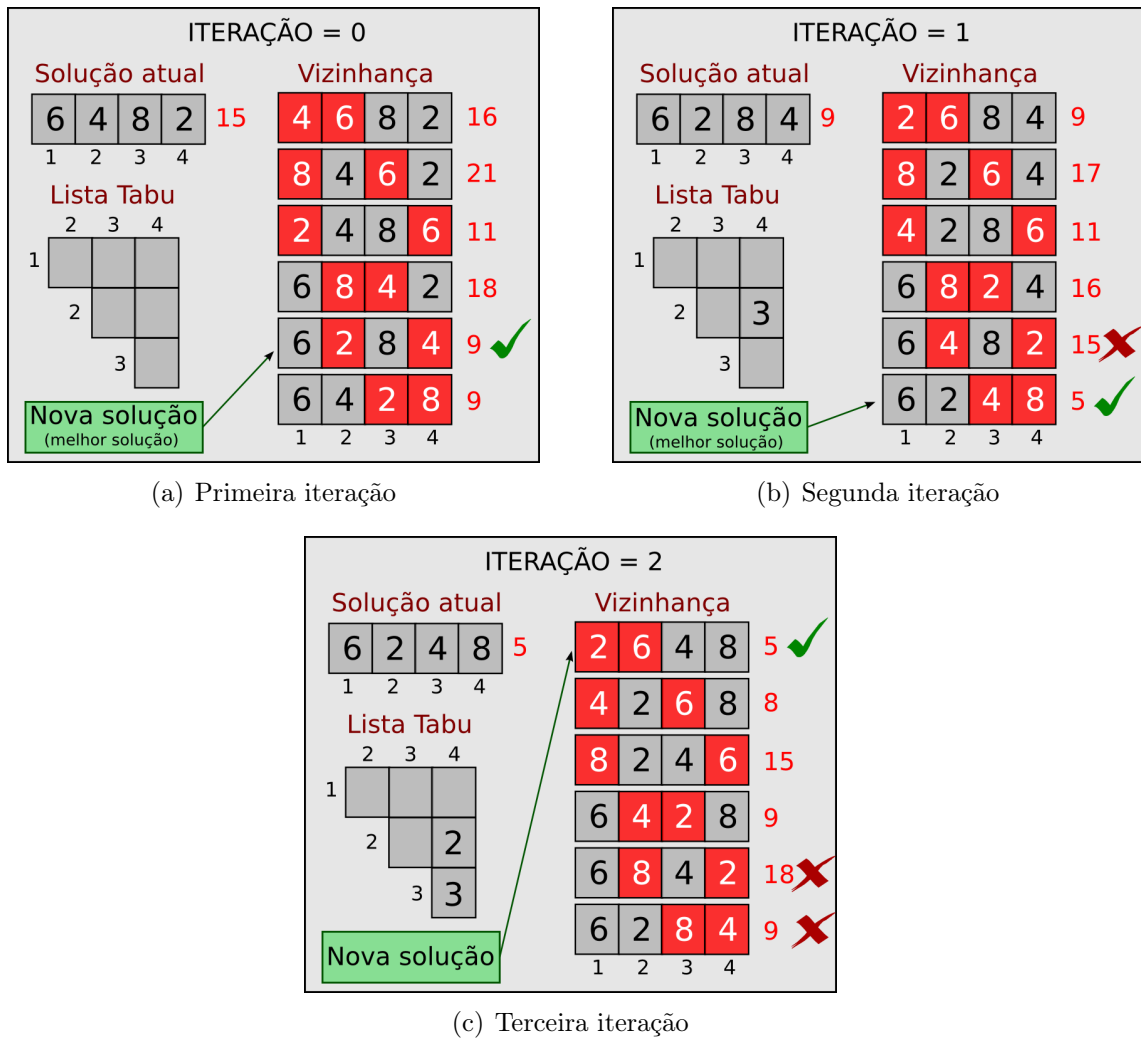


Figura 5: Exemplo de Busca Tabu no problema de sequenciamento de tarefas

Ao contrário do que vimos na seção 1.2, a representação da lista tabu indica quais

as posições que foram trocadas e não as tarefas. Isso mostra o quão subjetivo tornam-se as implementações de metaheurísticas, já que diversas maneiras podem ser empregadas para a resolução de um mesmo problema.

Como podemos ver na primeira iteração, ilustrada pela Figura 5(a), a partir da solução inicial, que possui $custo = 15$, é gerada a sua vizinhança realizando a troca de posições das tarefas. Consequentemente, a solução $[6, 2, 8, 4]$, que possui o menor custo, é selecionada e seu movimento é armazenado na lista tabu (a troca entre as posições 2 e 4). Nesse caso, essa solução também é atualizada como a melhor solução até o momento.

Na segunda iteração, ver Figura 5(b), a solução corrente é a melhor solução que foi selecionada da vizinhança da iteração anterior. Vemos que a solução $[6, 4, 8, 2]$ está marcada como tabu. Mas, vemos que a solução $[6, 2, 4, 8]$ tem menor custo ($custo = 5$) e é, então, atualizada como a solução corrente e, também, como melhor solução.

Na Figura 5(c), identificamos a última iteração do exemplo. Podemos observar o decremento do valor da célula $(2, 4)$ da lista tabu, assim como, o movimento $(3, 4)$ que foi armazenado como tabu na iteração anterior. Mais uma vez, geramos a vizinhança e selecionamos a melhor opção. Finalmente, o processo termina obtendo a solução $[6, 2, 4, 8]$ com $custo = 5$, encontrada na segunda iteração, como a melhor solução encontrada durante a execução do algoritmo.

1.5 Critério de Parada

Como vimos no Algoritmo 1 da seção anterior, a Busca Tabu é executada até que um certo critério de parada seja satisfeito. Segundo Gendreau(2003), os critérios de parada mais comuns utilizados em Busca Tabu são:

- depois de um certo número de iterações (ou um certo tempo de processamento);
- depois de um certo número de iterações sem melhora no valor da função objetivo (critério mais utilizado);
- quando o valor da função objetivo alcança um valor limite especificado previamente.

1.6 Intensificação e Diversificação

Como toda metaheurística, a Busca Tabu possui estratégias para realizar intensificação e diversificação. As estratégias de intensificação são baseadas na pesquisa cuidadosa de regiões consideradas promissoras. Dessa forma, uma estratégia típica é reiniciar a busca com a melhor solução encontrada e fixar os componentes mais atrativos.

Já a diversificação, segundo Gendreau (2003), é um mecanismo algorítmico que tenta forçar a busca por áreas não exploradas no espaço de busca. Na Busca Tabu, a fase de diversificação geralmente é implementada utilizando uma memória de longo prazo (GLOVER, 1996). Essa memória pode, por exemplo, armazenar os atributos menos utilizados durante a busca.

Duas estratégias principais de diversificação são geralmente utilizadas em Busca Tabu:

- Diversificação Contínua (*Continuous Diversification*) → adiciona na função objetivo um componente referente a frequência dos atributos, ou seja, insere informação de memória na avaliação do custo dos movimentos;
- Diversificação por Reinício (*Restart Diversification*) → força atributos raramente utilizados na solução atual, e reinicia a busca a partir da solução obtida.

No capítulo 4, entraremos em mais detalhes sobre a técnica de Diversificação por Reinício, que será utilizada na implementação do algoritmo.

1.7 Tendências

As pesquisas com Busca Tabu lidam com várias técnicas para melhorar a busca, tornando-a mais efetiva através da inclusão de métodos para explorar melhor a informação disponível durante a busca (GENDREAU, 2003).

DREAU, 2006, p. 58). E, permitindo a inclusão de atributos de soluções de alta qualidade, ela gera a possibilidade de encontrar soluções melhores que, muito provavelmente, não seriam encontradas por uma busca normal.

Como afirmam Glover, Laguna e Marti (2003), essa técnica foi chamada de Reconexão de Caminhos, tanto pela geração virtual de um novo caminho entre soluções previamente conectadas por uma série de movimentos executados durante a busca, quanto pela geração de um caminho entre soluções previamente conectadas a outras soluções, mas que não estão conectadas entre si. A Figura 7 mostra dois caminhos hipotéticos ligando a solução A à solução B. A linha contínua indica o caminho original produzido por uma operação normal de um procedimento de busca que produz uma série de movimentos levando de A para B. Já a linha pontilhada, indica o procedimento de Reconexão de Caminhos. Os caminhos são diferentes porque a seleção do movimento durante uma busca normal não sabe onde a solução B se encontra até que ela seja finalmente alcançada, ou seja, ela simplesmente segue uma trajetória em que os passos intermediários são determinados por alguma função de avaliação. Por exemplo, uma forma comumente usada é selecionar o movimento que minimiza (ou maximiza) o valor da função objetivo. Entretanto, durante a Reconexão de Caminhos, o principal objetivo é incorporar atributos da solução guia e ao mesmo tempo avaliar o valor da função objetivo. Além do mais, as duas soluções já são previamente conhecidas.

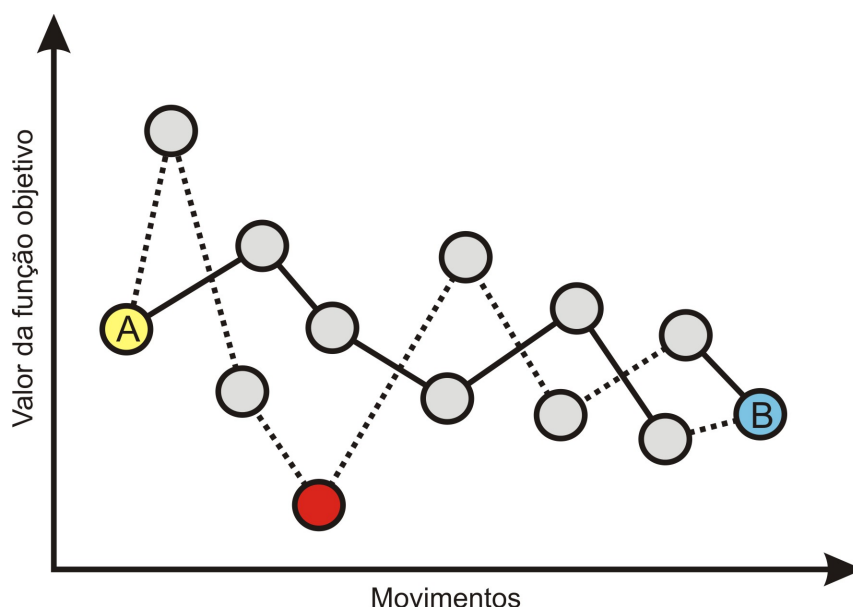


Figura 7: Ilustração da Reconexão de Caminhos

Dessa forma, é possível que a realização do *Path Relinking* encontre soluções melhores, como podemos ver, na Figura 7, a solução em destaque alcançada pelo caminho

pontilhado. Entretanto, ele pode, também, resultar em soluções que não sejam melhores que a solução inicial e a solução guia, mas que podem se tornar valiosos pontos de acesso para, a partir delas, encontrar melhores soluções. Por isso, é vantajoso examinar soluções vizinhas ao longo do caminho gerado pelo *path relinking* e, também, guardar as soluções de alta qualidade encontradas, que podem servir como ponto de partida para a continuação da busca (GLOVER, LAGUNA e MARTI, 2003).

2.1 Exemplo

Vamos, então, analisar o funcionamento de um procedimento de Reconexão de Caminhos para o seguinte exemplo, aplicado ao problema do caixeiro viajante.

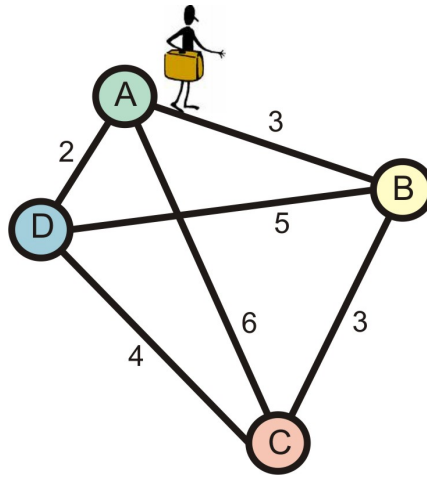


Figura 8: Problema do Caixeiro Viajante

No problema do caixeiro viajante desejamos encontrar um caminho que passe por cada uma das N cidades (nesse exemplo temos quatro cidades: A , B , C e D) apenas uma vez, e que tenha o menor custo; onde o custo do caminho é a soma dos custos das distâncias percorridas entre as cidades.

Para ilustrar o procedimento de Reconexão de Caminhos, partiremos da solução inicial $DCAB$ (custo = $\overline{DC} + \overline{CA} + \overline{AB} + \overline{BD} = 4 + 6 + 3 + 5 = 18$) e, utilizaremos a solução $ACBD$ (custo = 16) como solução guia.

Incluindo os atributos da solução guia na solução inicial, obtemos três novas soluções, como podemos ver na Figura 9, a seguir.

Dessa forma, a solução $ACDB$ é obtida pela inclusão da cidade A na primeira posição; a solução $DCBA$ é obtida pela inclusão da cidade B na terceira posição; e a solução $BCAD$ é obtida pela inclusão da cidade D na última posição.

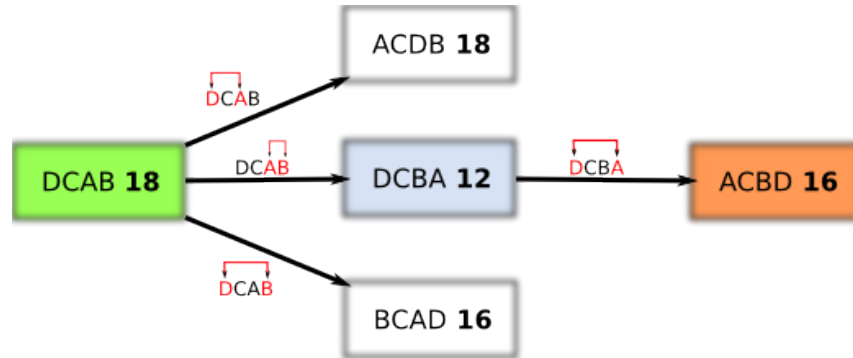


Figura 9: Ilustração da Reconexão de Caminhos aplicada ao problema do Caixeiro Viajante

Além disso, vemos que as setas, em cima das sequências de letras, indicam as trocas realizadas para a inserção das cidades nas posições solicitadas a partir da solução inicial e de acordo com a solução guia. Consequentemente, a solução de menor custo é escolhida (*DCBA*, com custo 12).

Em seguida, só é necessário mais um passo: realizando a troca da primeira posição com a última posição já obtemos a solução guia, caracterizando o fim do procedimento de Reconexão de Caminhos. Portanto, o caminho encontrado passa pelas soluções: $DCAB \rightarrow DCBA \rightarrow ACBD$. Além disso, vemos que, nesse caso, a Reconexão de Caminhos encontrou uma solução melhor que a solução guia.

2.2 Outras estratégias e variações

Outra possibilidade na Reconexão de Caminhos é considerar a combinação de atributos de um conjunto de soluções guias. Segundo Glover, Laguna e Marti (2003), essa técnica é conhecida como *Multiparent Path Relinking* e permite a geração de caminhos no espaço de busca de forma a reconectar pontos previamente conhecidos, mas de uma maneira não alcançada pelo histórico da busca anterior.

A Reconexão de Caminhos também se beneficia de uma estratégia denominada *tunnelling strategy*, que encoraja a utilização de uma diferente estrutura de vizinhança durante a busca, ou seja, permite a realização de movimentos que levam a soluções inviáveis, por exemplo.

Além disso, temos uma variação da Reconexão de Caminhos que usa vizinhanças construtivas (*constructive neighborhoods*) para criar soluções a partir de um conjunto de outras soluções. Nesse caso, as soluções guias pertencem a um conjunto de soluções elites, mas a solução inicial pode ser uma solução parcial (incompleta) ou, até mesmo,

uma solução vazia. O uso dessa técnica,

[...] permite que a solução inicial se mova entre as soluções guias, através de um caminho que progressivamente introduz elementos contidos nas soluções guias, ou que são avaliados como atrativos baseados na composição desse conjunto de soluções. (GLOVER, LAGUNA e MARTI, 2003, p. 7).

Portanto, a Reconexão de Caminhos muitas vezes pode ser interpretada como um método evolucionário em que as soluções são geradas através da combinação dos elementos de outras soluções, como cita Ho e Gendreau (2006).

3 *Localização de Facilidades*

Muitos problemas do cotidiano envolvem a decisão sobre locais para instalação de serviços e de meios de transportes. Por exemplo, qual a melhor localização para a instalação de postos de serviços, hospitais, fábricas, centros de distribuição de mercadorias, armazéns, antenas de telefonia celular, etc [...] (LORENA, 2004).

Problemas de localização tratam de decisões sobre onde localizar facilidades, considerando clientes que devem ser servidos, de forma a otimizar um certo critério.

É notável que os modelos para os problemas de localização de facilidades podem diferir com relação a sua função objetivo, a métrica de distância aplicada, o número e o tamanho das facilidades a serem alocadas, entre outros vários fatores. Dependendo da aplicação específica, a inclusão e consideração desses diversos fatores na formulação do problema pode levar a vários modelos de localização diferentes. (HALE e MOBERG, 2003, p. 22).

Como afirma Hale e Moberg (2003), um tipo de função objetivo predominante nos modelos para problemas de localização de facilidades é o *minisum*. A classe de problemas *minisum* tenta localizar o menor número de novas facilidades tal que a condição de distância pré-estabelecida para as facilidades existentes não seja violada. Um dos problemas mais referenciados dessa classe é o problemas das p -medianas.

A seguir, veremos uma breve descrição do problema de p -medianas, o qual será tratado neste trabalho.

3.1 Problema de p -medianas

Um dos mais conhecidos problemas de localização de facilidades é o problema de p -medianas. É um problema clássico que, segundo Beasley (1985), consiste em localizar p facilidades (denominadas medianas) em uma rede, de modo a minimizar a soma das distâncias de cada nó de demanda à sua mediana mais próxima. Esse

problema tem grande importância prática como, por exemplo, na localização de antenas de telecomunicação e/ou de roteadores *wireless*.

O problema de p -medianas é um problema de otimização combinatória e foi reconhecido como um problema NP-difícil por Cournuejols, Fisher e Nemhauser em 1977 (ALP, ERKUT e DREZNER, 2003).

De acordo com Reese (2005), podemos, então, defini-lo como: dado um grafo ou uma rede $G = (V, E)$, encontrar $V_p \subseteq V$ tal que $|V_p| = p$ e a soma das distâncias dos vértices em $\{V \setminus V_p\}$ para o vértice mais próximo em V_p seja minimizada.

3.1.1 Modelo Matemático

Como podemos ver em (SENNE, LORENA e PEREIRA, 2003), o problema de p -medianas pode ainda ser modelado como o seguinte problema de programação inteira binária:

$$\min \sum_{i=1}^n \sum_{j=1}^n d_{ij} x_{ij}$$

sujeito à:

$$\sum_{i=1}^n x_{ij} = 1, \forall j \quad (3.1)$$

$$\sum_{i=1}^n x_{ii} = p \quad (3.2)$$

$$x_{ij} \leq x_{ii}, \forall i, j \quad (3.3)$$

$$x_{ij} \in \{0, 1\}, \forall i, j \quad (3.4)$$

onde:

- $[d_{ij}]_{n \times n}$ é uma matriz simétrica de custos (distâncias), com $d_{ii} = 0, \forall i$;
- p é o número de medianas a serem localizadas;
- n é o número de nós da rede, com $N = \{1, \dots, n\}$;
- $[x_{ij}]_{n \times n}$ é uma matriz de alocação, com $x_{ij} = 1$ se o nó j é alocado à mediana i e $x_{ij} = 0$ caso contrário; $x_{ii} = 1$ se o nó i é uma mediana e $x_{ii} = 0$, caso contrário.

A restrição 3.1 garante que cada nó j é alocado a apenas um nó i , o qual deve ser uma mediana. A restrição 3.2 determina o número exato de medianas a ser localizado.

A restrição 3.3 garante que um nó j não seja alocado a um nó i que não é mediana e a restrição 3.4 corresponde às condições de integralidade.

3.1.2 Exemplo

Para entendermos melhor o problema de p -medianas, vamos analisar o problema de localizar 1(uma) facilidade na rede da Figura 10.

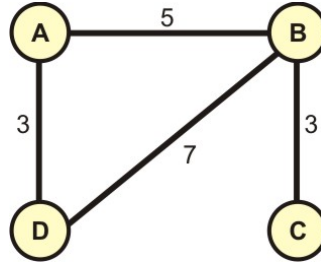


Figura 10: Problema de p -medianas

Seja $custo(n)$ a função definida pela soma das distâncias de todos os nós(usuários) para a possível facilidade n . Então,

- $custo(A) = \overline{BA} + \overline{CA} + \overline{DA} = 5 + (5 + 3) + 3 = 16$
- $custo(B) = \overline{AB} + \overline{CB} + \overline{DB} = 5 + 3 + 7 = 15$
- $custo(C) = \overline{AC} + \overline{BC} + \overline{DC} = (5 + 3) + 3 + (3 + 7) = 21$
- $custo(D) = \overline{AD} + \overline{BD} + \overline{CD} = 3 + 7 + (7 + 3) = 20$

Como o nó B tem o menor custo, então ele é a solução ótima para esse problema, ou seja, B é a facilidade escolhida.

3.1.3 Outras considerações

De acordo com Rolland, Schilling e Current (1996), Hakimi provou em 1965 que, para instâncias de rede desse problema, existe uma solução ótima, tal que, todas as suas facilidades se encontram em nós pertencentes a rede. Essa descoberta foi o que permitiu a formulação dessa versão do problema de p -medianas numa rede como um problema de programação inteira, como foi mostrado anteriormente. Entretanto, sabemos que se trata de um problema NP-difícil levando-nos a fazer a utilização de heurísticas e/ou metaheurísticas para tentar solucioná-lo em tempo hábil.

Dessa forma, veremos, no decorrer deste trabalho, como aplicaremos a metaheurística Busca Tabu para a resolução desse problema.

4 Metodologia

Como já foi dito anteriormente, o propósito deste trabalho é introduzir melhorias que podem ser derivadas da inclusão de estratégias de diversificação em métodos de Busca Tabu usando procedimentos de Reconexão de Caminhos. A seguir, vamos discutir qual a nossa metodologia. Iremos descrever o funcionamento da estratégia de Diversificação por Reinício; em seguida, falaremos um pouco da idéia proposta por Aloise *et al* (2006); e, por fim, iremos expor a nossa proposta.

4.1 Diversificação por Reinício

A estratégia de Diversificação por Reinício, já introduzida no capítulo 2 deste trabalho, consiste em forçar atributos raramente utilizados na solução corrente (ou na melhor solução encontrada) e reiniciar a busca desse ponto. Assim, vamos acompanhar, através da Figura 11, a execução hipotética de um algoritmo de Busca Tabu com a aplicação dessa estratégia.

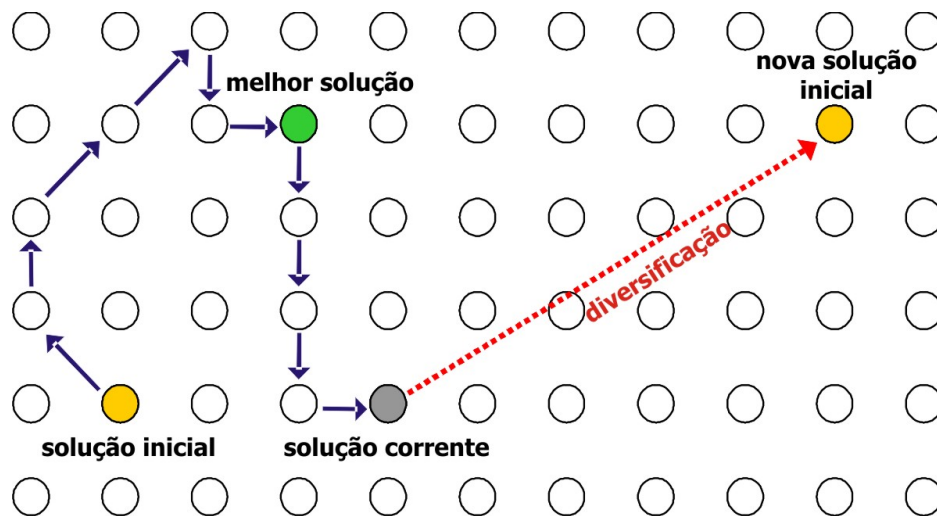


Figura 11: Diversificação por Reinício

Então, o algoritmo de Busca Tabu começa a partir de uma solução inicial e vai

procurando, a cada iteração, por melhores soluções na vizinhança a partir da solução corrente. A solução identificada como melhor solução, na Figura 11, é a melhor solução encontrada até o momento. Consequentemente, quando um determinado critério é satisfeito, a Diversificação por Reinício é aplicada na solução corrente, ou seja, os atributos que estão considerados como raramente utilizados são inseridos na solução corrente, gerando uma nova solução (que, muito provavelmente, se encontra bem afastada da solução corrente). Por sua vez, a busca continuará a partir da nova solução gerada pela diversificação.

É importante observar que, quando a diversificação é realizada, a exploração de uma vasta região no espaço de busca é negligenciada, embora o algoritmo possa voltar depois para explorar essa região.

4.2 Estratégia de Aloise *et al*

A proposta apresentada em Aloise *et al* (2006) surgiu, justamente, pela observação dessa região que fica, muitas vezes, inexplorada durante a busca. Dessa forma, essa estratégia assegura que a região negligenciada pela diversificação seja explorada através da aplicação da Reconexão de Caminhos, como é ilustrado na Figura 12.

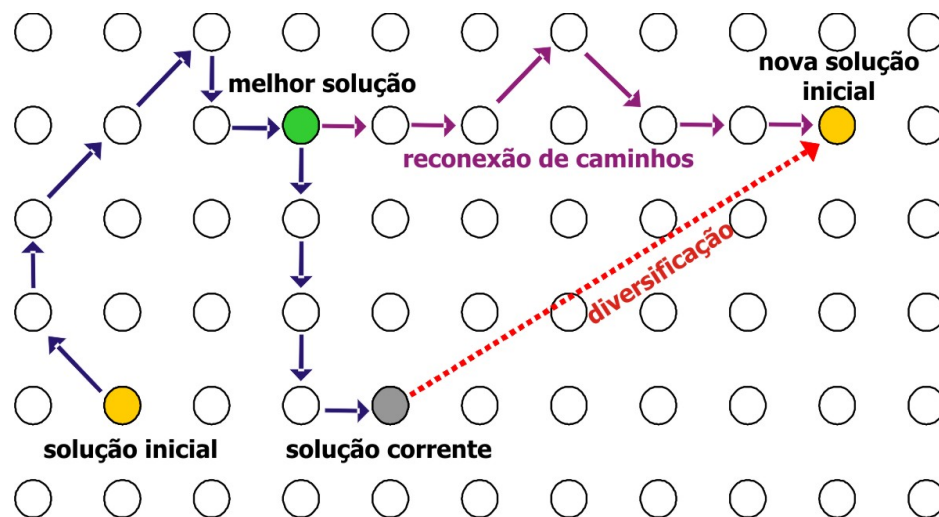


Figura 12: Estratégia de Aloise *et al*

Logo após ser realizada a diversificação, um procedimento de Reconexão de Caminhos é realizado partindo da melhor solução encontrada até aquele momento e utilizando como solução guia, a solução obtida pela diversificação. Depois disso, a busca é reiniciada a partir da solução diversificada.

Os resultados apresentados em Aloise *et al* (2006) mostraram bons indícios da existência de soluções boas (melhores do que as melhores já encontradas) no caminho obtido pela realização da Reconexão de Caminhos.

4.3 Uma estratégia mais robusta

Como poderemos observar, a seguir, nossa proposta é, na verdade, um aprimoramento da estratégia de Aloise *et al*. Ao invés de realizarmos apenas uma diversificação e aplicarmos a Reconexão de Caminhos, propomos a realização do que chamamos de “uma explosão de diversificações”, ou seja, fazemos várias diversificações, mais especificamente, três diversificações, e aplicamos a Reconexão de Caminhos em cada uma delas. Continuamos partindo da melhor solução encontrada e utilizando cada uma das soluções diversificadas como solução guia em cada procedimento de Reconexão de Caminhos. Esse processo é ilustrado pela Figura 13, onde podemos verificar que a busca continua a partir de uma das soluções diversificadas.

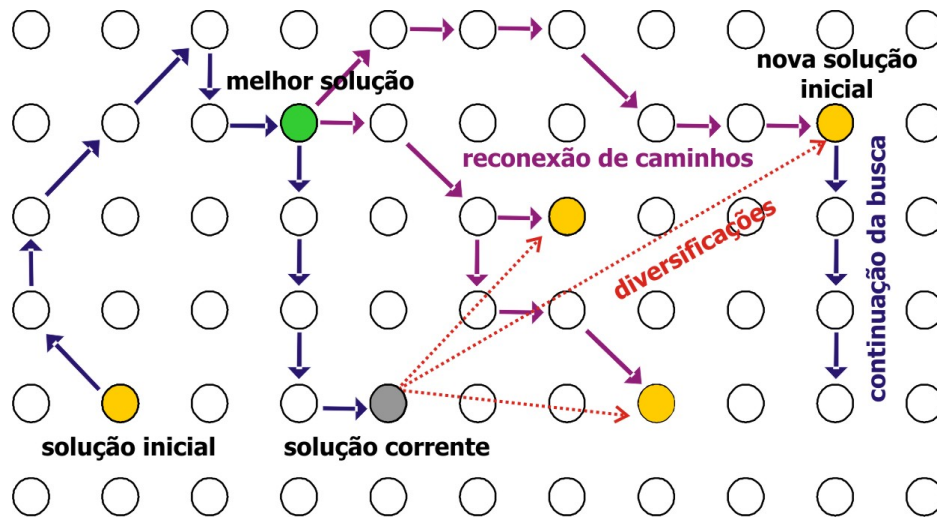


Figura 13: Explosão de diversificações

Portanto, a nossa idéia é explorar ainda mais o espaço de busca através do uso de procedimentos de Reconexão de Caminhos em outros pontos gerados simultaneamente pela estratégia de Diversificação por Reinício.

Além de fazermos uma comparação com a estratégia de Aloise *et al* (2006) com o intuito de validarmos nossa estratégia, faremos uma intensificação no algoritmo da seguinte maneira: durante a realização da busca, a melhor solução dos procedimentos de Reconexão de Caminhos é armazenada; ao final, quando o critério de parada for satisfeito, verificamos qual a melhor solução (comparando com a melhor obtida pelas

Reconexões de Caminhos) e aplicamos a Reconexão de Caminhos entre a melhor solução e a solução corrente; depois disso, a busca continua a partir da melhor solução encontrada por esse procedimento de Reconexão de Caminhos até que um certo número de iterações não aprimorantes seja alcançado.

Essa estratégia se assemelha com o procedimento da metaheurística *GRASP*, que gera uma lista restrita de candidatos formada com os melhores elementos (RESENDE e RIBEIRO, 2003). Entretanto, na metaheurística *GRASP* a solução na lista restrita é escolhida aleatoriamente e, na nossa estratégia, escolhemos sempre a melhor solução.

A seguir, iremos mostrar como foi implementada essa estratégia para o problema de p -medianas e os resultados obtidos.

5 *Experimentos Computacionais*

O algoritmo de Busca Tabu para o problema de p -medianas foi implementado utilizando uma abordagem baseada em troca de facilidades, como a utilizada por Hansen e Mladenovic (1997). Essa abordagem, é descrita detalhadamente por Resende e Werneck (2003), como veremos na seção 5.1. Em seguida, na seção 5.2, mostraremos os resultados obtidos, discutindo ainda, mais alguns detalhes de implementação. Também faremos um comparativo entre os resultados obtidos pelas duas estratégias previamente mostradas.

5.1 Algoritmo baseado na troca de facilidades

Primeiramente, vamos estabelecer a seguinte notação. F é o conjunto das facilidades em potencial e U é o conjunto de clientes que devem ser servidos pelas facilidades. Dessa forma, temos, $n = |U|$, $m = |F|$ e p , o número requerido de facilidades (no nosso caso, temos $m = n$, já que todos os clientes são possíveis candidatos a facilidades).

Além disso, usaremos u para identificar um cliente genérico e f para uma facilidade genérica. O custo de termos um cliente u associado a uma facilidade f é $d(u, f)$, que nada mais é do que a distância entre eles (sempre não-negativa). Uma solução S é qualquer subconjunto de F com p elementos.

Todo cliente u é associado à sua facilidade mais próxima $f \in S$ (ou seja, a facilidade que minimiza $d(u, f)$). Essa facilidade será denotada, $\phi_1(u)$. Nosso algoritmo também precisa acessar a segunda facilidade mais próxima, que será denotada por $\phi_2(u)$. Para simplificar a notação, abreviaremos $d(u, \phi_1(u))$ para $d_1(u)$ e $d(u, \phi_2(u))$ para $d_2(u)$.

Facilidades que são candidatas à inserção serão chamadas f_i (por definição $f_i \notin S$). E facilidades candidatas à remoção serão denotadas por f_r (por definição, $f_r \in S$).

Nesse modelo, todos os valores de $\phi_1(u)$ e $\phi_2(u)$ para uma dada solução S podem ser computados em $O(pn)$: para cada um dos n clientes, encontramos as distâncias

para cada uma das p facilidades e escolhemos a menor.

5.1.1 Procedimento de Teitz e Bart

Introduzido por Teitz e Bart em 1968, o procedimento padrão de busca local para o problema de p -medianas é baseado na troca de facilidades. Para cada facilidade $f_i \notin S$, o procedimento determina que facilidade $f_r \in S$ produz uma melhor solução caso f_i e f_r sejam trocadas (ou seja, se f_i for inserida e f_r removida da solução). Caso exista, a melhor troca é realizada e a busca continua a partir da nova solução.

A implementação desse procedimento pode ser feita em $O(pmn)$. Para determinar a primeira e a segunda facilidades mais próximas temos um algoritmo $O(pn)$. Depois, para cada par (f_i, f_r) , temos que calcular o valor da função *profit* que resultará da substituição de f_i por f_r . O valor dessa função expressa o efeito causado pela troca no valor final da solução, pois, sabemos que podem ocorrer ganhos e/ou perdas em sua realização.

Podemos melhorar o entendimento dessa função estabelecendo dois casos:

1. Se a facilidade associada a u não é f_r (facilidade a ser removida), o cliente mudará para f_i se, e somente se, f_i estiver mais próxima;
2. Se u estiver associado a f_r , o cliente deve ser associado, ou a $\phi_2(u)$ (a segunda facilidade mais proxima) ou a f_i (a facilidade a ser inserida), de acordo com a que estiver mais perto.

Para ilustrar esses casos, observemos o grafo da Figura 14.

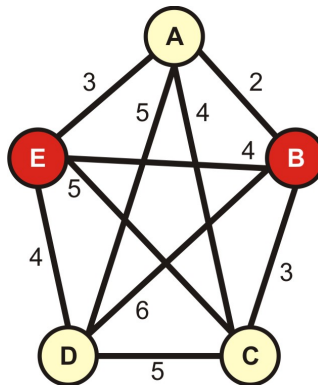


Figura 14: Grafo aplicado ao problema de 2-medianas com facilidades = $\{B, E\}$ e clientes = $\{A, C, D\}$

Então, geramos a Tabela 1, abaixo, para o conjunto de clientes com suas facilidades (ϕ_1 e ϕ_2) e suas respectivas distâncias (d_1 e d_2).

	A	C	D
$\phi_1(u)$	B	B	E
$d_1(u)$	2	3	4
$\phi_2(u)$	E	E	B
$d_2(u)$	3	5	6

Tabela 1: Descrição dos clientes do grafo da Figura 14

Supondo agora que desejamos fazer a troca da facilidade B pela facilidade em potencial C , ou seja, $f_r = B$ e $f_i = C$. Para o cliente D , caímos no primeiro caso, pois, a facilidade a ser removida não é a facilidade associada ao cliente D , ou seja, $f_r \neq \phi_1(D)$. Nesse caso, queremos saber apenas se $d(D, f_i) < d_1(D)$. Como $5 \not< 4$, então, o cliente D mantém a sua facilidade.

Para o cliente A , temos o segundo caso, pois, a facilidade a ser removida é a facilidade associada ao cliente A , ou seja, $f_r = \phi_1(A)$. Nesse caso, devemos calcular: $\min(d_2(A), d(A, f_i))$. Como $\min(3, 4) = 3$, teríamos então que substituir a facilidade associada a A por sua segunda facilidade mais próxima (caso a troca fosse realmente realizada).

Então, a função *profit* pode ser definida da seguinte forma:

$$profit(f_i, f_r) = \sum_{u: \phi_1(u) \neq f_r} \max\{0, [d_1(u) - d(u, f_i)]\} - \sum_{u: \phi_1(u) = f_r} [\min\{d_2(u), d(u, f_i)\} - d_1(u)]$$

O primeiro somatório refere-se aos clientes que não têm f_r como facilidade ($f_r \neq \phi_1(u)$). Eles não possuem nada a perder com a troca. Já o segundo somatório refere-se aos clientes que podem perder ou ganhar com a troca ($f_r = \phi_1(u)$). Essa expressão pode ser calculada em $O(n)$ para cada par de candidatos. Como temos p candidatos para remover e $m - p$ candidatos à inserção, teremos que chamar a função $p(m - p)$ vezes, então temos $O(pm)$. Portanto, cada iteração no algoritmo de busca leva $O(pnm)$. No nosso caso, como temos $m = n$ então temos $O(pn^2)$.

5.1.2 Implementação de Whitaker

Uma implementação mais eficiente do procedimento de Teitz e Bart foi feita por Whitaker em 1983. Essa implementação foi utilizada por Hansen e Mladenovic em

1997, como foi dito no início deste capítulo. Entretanto, há uma pequena diferença nas duas implementações: enquanto que Whitaker prefere a estratégia de realizar logo a primeira troca que resulta na melhora da solução (*first improvement*), Hansen e Mladenovic avaliam todas as trocas e escolhem a mais proveitosa (*best improvement*). Como dissemos anteriormente, utilizamos para o problema de p -medianas a estratégia de Hansen e Mladenovic, ou seja, *best improvement*.

Dessa forma, a implementação de Whitaker permite encontrar o melhor candidato a ser removido, dado um certo candidato a ser inserido, em $\Theta(n)$. O Algoritmo 2, a seguir, apresenta a função *findOut*, descrito por Resende e Werneck (2003). Ela recebe como entrada um candidato a inserção (f_i) e retorna f_r , a melhor facilidade para realização da troca, assim como, o valor do *profit*.

Algoritmo 2 *findOut*(S, f_i, ϕ_1, ϕ_2)

```

1:  $gain \leftarrow 0$ ;
2: for all ( $f \in S$ ) do
3:    $netloss(f) \leftarrow 0$ ;
4: end for
5: for all  $u \in S$  do
6:   if  $d(u, f_i) \leq d_1(u)$  then {ganho se  $f_i$  está mais próximo de  $u$ }
7:      $gain \leftarrow^+ [d_1(u) - d(u, f_i)]$ 
8:   else {perda se a facilidade associada a  $u$  é removida}
9:      $netloss(\phi_1(u)) \leftarrow^+ \min\{d(u, f_i), d_2(u)\} - d_1(u)$ ;
10:  end if
11: end for
12:  $f_r \leftarrow \operatorname{argmin}_{f \in S} \{netloss(f)\}$ ;
13:  $profit \leftarrow gain - netloss(f_r)$ ;
14: return ( $f_r, profit$ );

```

Nesse caso, a função *profit* é decomposta em dois componentes: *gain* e *netloss*. O componente *gain* avalia, para todos os clientes, aqueles que se beneficiam com a inserção de f_i na solução, ou seja, os clientes que estão mais próximos de f_i do que suas próprias facilidades. Assim, a diferença de suas distâncias é o valor que será reduzido do custo da solução final, caso f_i seja inserida. As linhas 6 e 7 do pseudocódigo calculam o valor de *gain*.

Ao contrário do componente *gain*, o componente *netloss* avalia, para todos os clientes, aqueles que não se beneficiam com a inserção de f_i na solução. Se a facilidade associada a u for removida, então ele deverá se associar ou a $\phi_2(u)$ ou a f_i . Em ambos os casos, o custo da solução com a inserção de f_i irá aumentar ou permanecer constante. A linha 9 do pseudocódigo calcula o valor de *netloss*.

Portanto, dado que a função *findOut* é executada em $O(n)$, cada iteração será executada em $O(mn)$, já que precisamos chamar a função *findOut* para cada um dos $m - p$ candidatos à inserção.

5.2 Resultados

Os testes foram realizados nas mesmas instâncias utilizadas por Aloise *et al* (2006), que foram retiradas da OR-Library (BEASLEY, 1985). Além disso, a implementação do algoritmo foi feita na linguagem C++, usando a versão 4.0 do compilador g++. E os experimentos foram realizados num Intel Core 2 Duo 2.0 GHz com 1Gb de memória RAM rodando o MAC OS X 10.4.10.

5.2.1 Diversificações

As diversificações no algoritmo são realizadas da seguinte forma. Seja n o número de elementos (nós) de uma determinada instância para o problema de p -medianas. Uma solução válida é composta por p elementos (p facilidades ou medianas). Então, quando o critério de diversificação é satisfeito, uma determinada quantidade de elementos, pertencente ao conjunto formado pelos $n - p$ elementos, são inseridos na solução corrente, de acordo com suas frequências (memória de longo prazo), substituindo os elementos mais frequentes naquela solução. Após cada diversificação, a memória de longo prazo é reinicializada para evitar que os mesmos elementos sejam utilizados nas diversificações seguintes.

5.2.1.1 Critério de diversificação

Com relação ao critério de diversificação, o algoritmo diversifica a busca depois de 30, 60 ou 90 iterações serem realizadas sem que haja melhora na melhor solução encontrada desde a última diversificação.

5.2.2 Validação da estratégia mais robusta

Nesse caso, como o nosso algoritmo realiza três diversificações, uma delas insere $p/4$ elementos e as outras duas $p/8$ elementos cada uma. Depois de realizadas as três diversificações, a Busca Tabu continua a partir da primeira solução diversificada obtida.

Na Tabela 2 podemos ver os resultados obtidos, lembrando que as colunas E1 referem-se aos resultados obtidos utilizando a estratégia de Aloise *et al*, ou seja, fazendo apenas uma diversificação. As colunas E2 são os resultados obtidos pela nossa estratégia, usando a explosão de diversificações. Dessa forma, para cada instância, cada uma das condições foi executada 20 vezes usando como critério de parada a realização de cinco diversificações.

Os resultados medem a frequência com que a execução da Reconexão de Caminhos encontrou uma melhor solução do que a melhor solução encontrada até aquele momento.

Instâncias			Número de iterações não aprimorantes antes da diversificação					
			30 iterações		60 iterações		90 iterações	
Nome	n	p	E1	E2	E1	E2	E1	E2
Pmed29	600	120	5,26%	4,60%	5,38%	5,38%	4,55%	10,23%
Pmed30	600	200	3,26%	4,26%	2,15%	5,49%	5,43%	4,35%
Pmed33	700	70	4,76%	5,00%	3,70%	7,55%	13,95%	25,00%
Pmed34	700	140	3,23%	3,23%	3,95%	2,35%	2,41%	1,19%
Pmed37	800	80	2,35%	4,71%	6,90%	7,87%	6,82%	5,56%
Pmed40	900	90	3,00%	3,00%	1,00%	4,00%	4,00%	8,00%
Média:			3,64%	4,13%	3,85%	5,44%	4,69%	9,05%

Tabela 2: Porcentagem das tentativas de Reconexão de Caminhos com sucesso

Para entender melhor os resultados, vamos dar uma olhada nos algoritmos das duas implementações.

Algoritmo 3 E1

```

1: inicialização()
2: while número_diversificações < 5 do
3:     if atingiu número de iterações não aprimorantes then
4:         {Diversificação}
5:         solução_div = diversifica(p/4, solução_corrente)
6:
7:         {Reconexão de Caminhos}
8:         melhor_sol_path = path_relinking(melhor_solução, solução_div)
9:
10:        solução_corrente = solução_div
11:        número_diversificações = número_diversificações + 1
12:    else
13:        busca_tabu()
14:    end if
15: end while

```

O Algoritmo 3 mostra como é feita a diversificação. Podemos observar que o algoritmo pára após a realização de cinco diversificações, como vemos na linha 2. Quando atingimos um certo número de iterações não aprimorantes (30, 60 ou 90), é hora do algoritmo realizar a diversificação procedido da execução do procedimento de Reconexão de Caminhos, identificados, respectivamente, nas linhas 5 e 8. Dessa forma, toda vez que esse procedimento retorna uma solução melhor que a melhor solução isso será contabilizado para obtermos os resultados ao final da execução.

Algoritmo 4 E2

```

1: inicialização()
2: while número_diversificações < 5 do
3:     if atingiu número de iterações não aprimorantes then
4:         {Primeira diversificação}
5:         solução_div1 = diversifica(p/4, solução_corrente)
6:         melhor_sol_path1 = path_relinking(melhor_solução, solução_div1)
7:
8:         {Segunda diversificação}
9:         solução_div2 = diversifica(p/8, solução_corrente)
10:        melhor_sol_path2 = path_relinking(melhor_solução, solução_div2)
11:
12:        {Terceira diversificação}
13:        solução_div3 = diversifica(p/8, solução_corrente)
14:        melhor_sol_path3 = path_relinking(melhor_solução, solução_div3)
15:
16:        solução_corrente = solução_div1
17:        número_diversificações = número_diversificações + 1
18:    else
19:        busca_tabu()
20:    end if
21: end while

```

Já no Algoritmo 4, temos a realização de três diversificações seguidas, cada uma, por um procedimento de Reconexão de Caminhos. Depois, a busca continua a partir da solução da primeira diversificação. É claro que podemos escolher qualquer uma das 3 soluções obtidas pela diversificação ou, até mesmo, a melhor delas. Mas, podemos notar que essa decisão não influencia muito na obtenção dos resultados e, por outros testes realizados, continuamos obtendo sempre um padrão parecido como o mostrado na Tabela 2. Nesse caso, contabilizamos nos nossos resultados quando pelo menos uma das três soluções obtidas no procedimento de Reconexão de Caminhos for melhor que a melhor solução encontrada. Dessa forma, podemos fazer uma comparação entre os resultados obtidos pelos dois algoritmos.

A partir da análise dos resultados, vemos que a nossa estratégia obteve, em geral,

uma melhor frequência de reconexões de caminho com sucesso, como podemos observar através do gráfico da Figura 15, que mostra as médias obtidas pelos dois algoritmo.

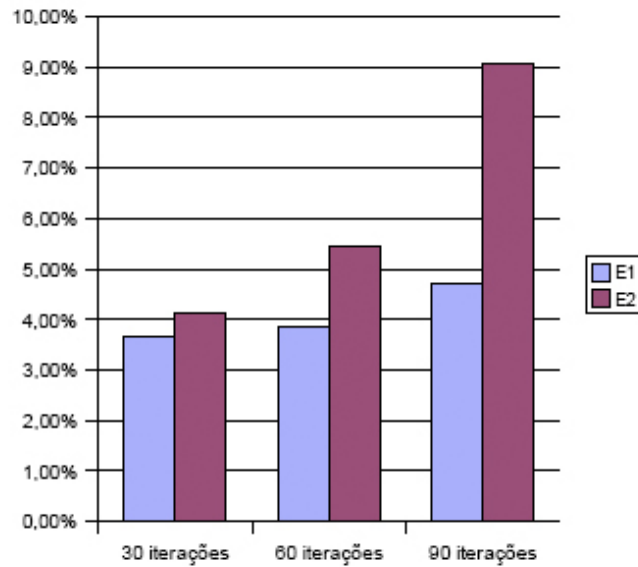


Figura 15: Média de sucessos da aplicação da Reconexão de Caminhos em função do número de iterações não aprimorantes antes da diversificação

Além disso, observamos que a Reconexão de Caminhos apresenta uma maior taxa de sucesso quando a condição para a diversificação é menos restrita, ou seja, à medida que aumenta o número de iterações não aprimorantes antes da diversificação, a taxa de sucesso é maior. Como explica Aloise *et al* (2006), isso ocorre porque à medida que a busca é realizada, mais informações úteis são obtidas sobre os elementos e, portanto, uma solução diversificada mais confiável pode ser obtida.

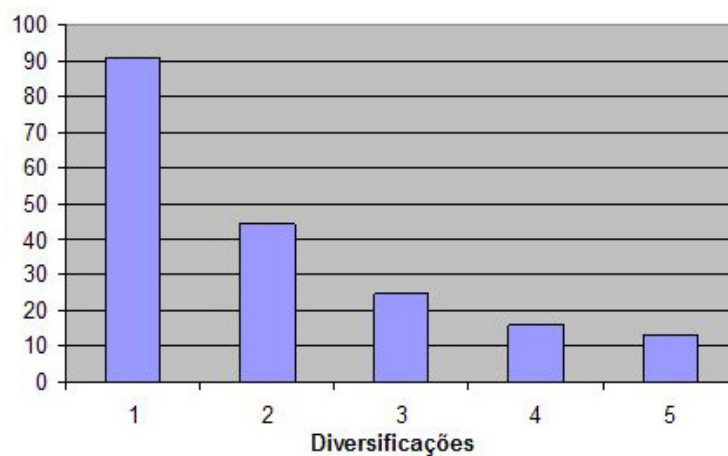


Figura 16: Número de sucessos da aplicação da Reconexão de Caminhos em função da diversificação realizada

Através da Figura 16, acima, notamos o mesmo efeito observado por Aloise *et al*

(2006), indicando que a maior parte dos sucessos se encontram na primeira diversificação realizada. Isso já era esperado, pois, no decorrer da busca, a cada iteração se torna mais difícil encontrar uma solução melhor que a melhor já encontrada.

Com isso, vemos que nossa estratégia apresenta-se melhor do que a estratégia de Aloise *et al.* Esse resultado é um indicativo muito importante para que o algoritmo com a aplicação da intensificação obtenha melhores resultados como apresentaremos na próxima seção.

Outra informação importante, com relação aos resultados obtidos, é o tempo de execução que é mostrado pela Tabela 3, a seguir.

Instâncias			Número de iterações não aprimorantes antes da diversificação					
			30 iterações		60 iterações		90 iterações	
Nome	n	p	E1	E2	E1	E2	E1	E2
Pmed29	600	120	106	91	153	318	191	415
Pmed30	600	200	293	643	539	1065	170	1233
Pmed33	700	70	70	154	97	179	67	127
Pmed34	700	140	175	247	270	455	382	703
Pmed37	800	80	97	168	159	262	148	208
Pmed40	900	90	153	235	240	454	253	336
Média:			149	256	243	455	202	504

Tabela 3: Tempo médio de execução (em segundos)

Podemos confirmar que houve em média um aumento de duas vezes no tempo de execução da nossa estratégia com relação a estratégia de Aloise *et al.* Portanto, vemos que o procedimento de Reconexão de Caminhos em conjunto com a Diversificação por Reinício tem grande importância na execução do algoritmo.

5.2.3 Aplicando a intensificação

Como explicado no capítulo anterior, a intensificação no algoritmo é feita da seguinte forma: a melhor solução de todos os procedimentos de Reconexão de Caminhos é armazenada e quando o algoritmo “termina” obtemos uma melhor solução ($\min(\text{melhor solução das Reconexões de Caminhos, melhor solução da Busca Tabu})$) e aplicamos a Reconexão de Caminhos entre ela e a solução corrente; depois, aplicamos a Busca Tabu na melhor solução encontrada nessa Reconexão de Caminhos até que um certo número de iterações não aprimorantes seja alcançado.

Dentre os vários testes realizados, os resultados mostrados na Tabela 4, a seguir,

foram os melhores obtidos e seu algoritmo possui a seguinte configuração:

- cada uma das três diversificações insere $p/8$ elementos;
- para cada instância, cada uma das condições foi executada 20 vezes usando como critério de parada a realização de cinco diversificações (da mesma forma como é mostrado pelo Algoritmo 4);
- o algoritmo diversifica a busca depois de 30, 60 ou 90 iterações serem realizadas sem que haja melhora na solução;
- depois de realizadas as três diversificações, a Busca Tabu continua a partir da melhor das três soluções diversificadas;
- o algoritmo pára depois de 30, 60 ou 90 iterações sem que haja melhora na solução, após ser realizada a intensificação.

Fazendo apenas uma ressalva: o Algoritmo E1 só realiza uma diversificação, como é mostrado no Algoritmo 3, e não aplicamos nenhum processo de intensificação.

Instância	Ótimo	30 iterações		60 iterações		90 iterações	
		E1	E2	E1	E2	E1	E2
Pmed29	3033	3035,05	3034,35	3035,1	3033,95	3034,45	3034,25
Pmed30	1989	1991,5	1992,25	1991,3	1990,95	1991,05	1990,35
Pmed33	4700	4701,5	4701,3	4700,75	4700,45	4700,2	4700,45
Pmed34	3013	3014,55	3014,85	3014,35	3014,2	3014,3	3013,85
Pmed37	5057	5058,5	5058	5058,55	5058,3	5058,45	5057,85
Pmed40	5128	5130,6	5130,4	5130,9	5130,25	5130,95	5130,75

Tabela 4: Média das soluções

Diante dos resultados obtidos, vemos que o Algoritmo E2 obteve, no geral, melhores resultados que o Algoritmo E1, o que demonstra que a implementação da nossa estratégia pode implicar numa melhora significativa da solução.

Como sabemos, vários problemas considerados NP-difíceis tem aplicações práticas em indústrias, por exemplo. E, apesar de observarmos um “pequena” melhora, podemos obter reduções de custos consideráveis, de acordo com o problema ao qual está sendo aplicado.

Considerações Finais

Atualmente, vemos que existe uma grande aplicação de metaheurísticas nos problemas que até hoje não se encontraram soluções polinomiais, como os problemas NP-difíceis. A maioria desses problemas têm grande aplicação prática no nosso cotidiano. É por isso que surge a necessidade de criar algoritmos e estratégias que possam encontrar boas soluções num tempo considerável.

Durante o decorrer deste trabalho mostramos uma estratégia que melhora a performance da metaheurística Busca Tabu usando a Reconexão de Caminhos. E podemos afirmar que a aplicação dessa nova estratégia pode ser bastante promissora. Além disso, a partir dela podemos desenvolver e verificar diversos outros aprimoramentos para torná-la ainda melhor como, por exemplo, verificar a possibilidade de reiniciar a busca, depois de cada diversificação, com a melhor solução obtida entre as melhores dos procedimentos de Reconexão de Caminhos; utilizar outras estratégias de Reconexão de Caminhos, como mostrado na seção 2.2; e, aplicar essa estratégia em outros problemas disponíveis na literatura.

Referências

- ALOISE, D. J. et al. Using path relinking to improve a tabu search diversification strategy. In: CLAIO - CONGRESSO LATINO-IBEROAMERICANO DE INVESTIGACIÓN OPERATIVA, 13., 2006, Montevideo. *Anais...* Montevideo, 2006.
- ALP, O.; ERKUT, E.; DREZNER, Z. An efficient genetic algorithm for the p-median problem. In: *Annals of Operations Research*. Boston: Kluwer Academic Publishers, 2003. p. 21–42.
- BEASLEY, J. *OR-Library*. Disponível em: <<http://people.brunel.ac.uk/~mastjjb/jeb/orlib/pmedinfo.html>>. Acesso em: 14 jun. 2007.
- BEASLEY, J. A note on solving large p-median problems. *European Journal of Operational Research*, n. 21, p. 270–273, 1985.
- DREZNER, Z. On the conditional p-median problem. *Computers and Operations Research*, v. 22, n. 5, p. 525–530, 1995.
- GENDREAU, M. An introduction to tabu search. In: GLOVER, F.; KOCHENBERGER, G. (Ed.). *Handbook of metaheuristics*. Boston: Kluwer Academic Publishers, 2003. p. 37–54.
- GLOVER, F. Tabu search and adaptive memory programming - advances, applications and challenges. In: BARR, R. S.; HELGASON, R. V.; KENNINGTON, J. L. (Ed.). *Interfaces in computer science and operations research*. Boston: Kluwer Academic Publishers, 1996a. p. 1–75.
- GLOVER, F. Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research*, n. 13, p. 533–549, 1996b.
- GLOVER, F. Multi-start and strategic oscillation methods - principles to exploit adaptive memory. In: LAGUNA, M.; GONZALES-VELARDE, J. L. (Ed.). *Computing tools for modeling, optimization and simulation: Interfaces in Computer Science and Operations research*. Boston: Kluwer Academic Publishers, 2000. p. 1–24.
- GLOVER, F.; LAGUNA, M. Tabu search. In: REEVES, C. (Ed.). *Modern Heuristic Techniques for Combinatorial Problems*. Oxford: Blackwell Scientific Publications, 1993. p. 70–150.
- GLOVER, F.; LAGUNA, M. *Tabu Search*. Boston: Kluwer Academic Publishers, 1997.
- GLOVER, F.; LAGUNA, M.; MARTI, R. Scatter search and path relinking: Advances and applications. In: GLOVER, F.; KOCHENBERGER, G. (Ed.). *Handbook of Metaheuristics*. Boston: Kluwer Academic Publishers, 2003. p. 219–249.

- HALE, T. S.; MOBERG, C. R. Location science research: A review. In: *Annals of Operations Research*. Boston: Kluwer Academic Publishers, 2003. p. 21–35.
- HANSEN, P.; MLADENOVIC, N. Variable neighbourhood search for the p-median. *Location Science*, n. 5, p. 207–226, 1997.
- HO, S. C.; GENDREAU, M. Path relinking for the vehicle routing problem. *Journal of Heuristics of Heuristics*, v. 12, n. 1-2, p. 55–72, 2006.
- LORENA, L. A. N. *Problema de Localização de Facilidades de Máxima Cobertura*. Disponível em: <http://www.lac.inpe.br/~lorena/marcos/public_html/arsig2/maxcobARSIG2.htm>. Acesso em: 13 jun. 2007.
- LORENA, L. A. N. *Problemas de localização e transportes integrados a sistemas de informações geográficas*. Disponível em: <<http://www.comciencia.br/200404/reportagens/10.shtml>>. Acesso em: 12 jun. 2007.
- LORENA, L. A. N.; SENNE, E. L. F. Abordagens complementares para problemas de p-medianas. *Rev. Produção*, v. 13, n. 3, p. 42–51, 2003.
- REESE, J. *Methods for solving the p-median problem an annotated bibliography*. Disponível em: <<http://lagrange.math.trinity.edu/tumath/research/reports/report96.pdf>>. Acesso em: 10 jun. 2007.
- REEVES, C.; YAMADA, T. Genetic algorithms, path relinking and the flowshop sequencing problem. *Evolutionary Computation*, v. 6, p. 45–60, 1998.
- RESENDE, M. G. C.; RIBEIRO, C. C. Greedy randomized adaptive search procedures. In: GLOVER, F.; KOCHENBERGER, G. (Ed.). *Handbook of Metaheuristics*. Boston: Kluwer Academic Publishers, 2003. p. 219–249.
- RESENDE, M. G. C.; RIBEIRO, C. C. Grasp with path relinking: recent advances and applications. In: IBARAKI, T. (Ed.). *Metaheuristics: Progress as real problem solvers*. Boston: Kluwer Academic Publishers, 2005. p. 29–63.
- RESENDE, M. G. C.; WERNECK, R. F. *A Fast Swap-based Local Search Procedure for Location Problems*. AT&T Labs Research, 2003.
- ROLLAND, E.; SCHILLING, D. A.; CURRENT, J. R. An efficient tabu search procedure for the p-median problem. *European Journal of Operational Research*, p. 329–342, 1996.
- SENNE, E. L. F.; LORENA, L. A. N.; PEREIRA, M. A. Um algoritmo branch-and-price para problemas de localização de p-medianas. In: SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL, 35., 2003, Natal. *Anais...* Natal, 2003.
- TEITZ, M. B.; BART, P. Heuristic methods for estimating the generalized vertex median of a weighted graph. *Operations Research*, v. 16, n. 5, p. 955–961, 1968.