

OTIMIZAÇÃO DO PROJETO DE REDES URBANAS BASEADO NO PROBLEMA DE STEINER

Luiz Carlos de Abreu Rodrigues e Hideson Alves da Silva

CPGEI – CEFET/PR – Centro Federal de Educação Tecnológica do Paraná, Av. Sete de Setembro 3165, 80230-901, Curitiba, PR, Brasil

Resumo: Este trabalho apresenta uma proposta para a otimização do projeto de vários tipos de redes urbanas, como, por exemplo, redes de telecomunicações, que freqüentemente vêm sendo representadas em sistemas GIS, onde a rede pode ser modelada como um problema de *Steiner*. Propõe-se que este problema de otimização seja resolvido em duas etapas, onde, na primeira etapa, executa-se o pré-processamento do problema, reduzindo a complexidade computacional do mesmo, e, numa segunda etapa, utiliza-se a Busca Tabu para a geração de soluções deste problema.

Abstract: This paper presents an approach to solve network design optimization problems, e.g. telecommunication networks, which most frequently rely on GIS information. These networks can be modeled as Steiner problems with the solution of the optimization problem in two steps. First a preprocessing of the Steiner problem is performed, aiming at a reduction in the computational complexity of the problem. The second step is to solve the resulting optimization problem using Tabu Search.

Key words: Networks, Steiner problem, Tabu Search, Preprocessing, GIS.

1. INTRODUÇÃO

Apresentamos uma proposta para a otimização do projeto de redes urbanas, tais como redes de telecomunicações, distribuição de energia elétrica, distribuição de gás natural e águas pluviais, entre outras. Este

problema é modelado como um problema de *Steiner*, como é chamado na literatura¹, que corresponde, no nosso caso, a uma rede com topologia do tipo estrela, composta por nós de demanda (clientes) e nós de Steiner. Os nós de Steiner são os pontos usados para as ligações com a rede e, no caso de uma cidade, todas as junções de ruas e pontos onde há equipamentos são considerados nós de Steiner. Assim, o problema de Steiner consiste em definir uma rede com custo mínimo, definindo-se quais nós de Steiner estarão ativos (ou seja, que farão parte da rede), quais serão as ligações entre nós de Steiner ativos, compondo o *backbone* da rede, e a que nó de Steiner estará ligado cada nó de demanda, restringindo-se que cada nó de demanda seja conectado somente em um nó de *Steiner* ativo.

A complexidade computacional do problema de Steiner, classificado como *NP-hard*², favoreceu o surgimento de diversas propostas heurísticas para solucioná-lo, havendo diversas abordagens usando Busca Tabu na literatura²⁻⁹. A Busca Tabu, proposta por Glover¹⁰, se caracteriza como uma heurística de busca através de soluções vizinhas da solução vigente, explorando estruturas de memória para inibir certos movimentos (ou soluções). As soluções vizinhas correspondem ao conjunto de soluções que podem ser geradas a partir de uma solução vigente.

O principal argumento para o uso de abordagens heurísticas, que não garantem a identificação da solução ótima, é que, para problemas de grandes dimensões (como ocorre com problemas reais de redes urbanas), nem sempre é possível encontrar a solução ótima do problema, principalmente quando há uma limitação no tempo de processamento. Assim, neste casos, prioriza-se a identificação de boas soluções, tentando-se encontrar a melhor solução possível em um período de tempo (ou número de iterações) definido. Neste trabalho estuda-se o impacto de um procedimento de pré-processamento na redução da complexidade computacional do problema de Steiner. Ou seja, realizou-se uma comparação dos resultados da execução da Busca Tabu, com e sem a execução do pré-processamento.

O resultado deste estudo da otimização do projeto de redes urbanas é um algoritmo que pode ser implementado e utilizado em sistemas comerciais de geoprocessamento, particularmente, utilizando-se das informações destes sistemas comerciais. Ao mesmo tempo, empresas que trabalham com redes (telecomunicações, elétrica, gás natural etc.), instaladas em regiões urbanas, necessitam de sistemas de apoio para seus projetos de redes.

Este trabalho está assim organizado: na seção 2 é apresentado o algoritmo de Busca Tabu que foi implementado; na seção 3 são apresentados os movimentos implementados nesta abordagem para tratar o problema de Steiner; na seção 4 é apresentado o procedimento de pré-processamento; na seção 5 são apresentados os resultados preliminares deste trabalho, ainda em andamento, e na seção 6 são apresentadas conclusões e trabalhos futuros.

2. ALGORITMO DE BUSCA TABU

O algoritmo de Busca Tabu apresentado nesta seção baseia-se nos procedimentos apresentados por Glover e Laguna¹⁰. A cada nova iteração, gera-se um nova solução vigente, vizinha da solução anterior. No problema de *Steiner*, associa-se um custo a cada aresta e nó de *Steiner* “ativos” (que fazem parte da solução). A cada solução vizinha é definido um valor, calculado pela função objetivo, tomando-se como nova solução vigente, entre todas as soluções vizinhas da solução vigente, aquela com melhor valor da função objetivo (solução de menor custo). Se não existem movimentos (que geram soluções vizinhas) que melhorem a solução vigente, o algoritmo de Busca Tabu escolhe o movimento que “piora menos” o valor da função objetivo da solução (até então) vigente.

A única estrutura de memória usada, nesta implementação, para restringir a geração de movimentos ou soluções vizinhas, foi a Lista Tabu. Na Lista Tabu são armazenados os últimos movimentos executados pelo algoritmo, tendo a função de inibir a ocorrência de ciclos e/ou que a busca fique presa em pontos de mínimo (ou máximo) local. Assim, o último movimento executado é armazenado na Lista Tabu e fica proibido (torna-se “tabu”) por um determinado número de iterações. O “Critério de Aspiração” é usado para determinar quando um movimento será executado, mesmo sendo classificado como proibido por estar na Lista Tabu. Normalmente, utiliza-se o Critério de Aspiração se a solução formada a partir de um movimento proibido for melhor que a melhor solução encontrada até o momento.

A busca por soluções em uma certa vizinhança é chamada “intensificação” porque, quanto mais promissora a região, mais intensa a busca nesta região. Mas quando a busca fica “presa” em uma região (de mínimos ou máximos locais), a busca deve ser deslocada para uma região ainda não explorada. O procedimento de geração de uma nova solução inicial, mudando a região de busca, é chamado de “diversificação”, oferecendo novas opções de soluções do problema. A diversificação permite que o algoritmo faça uma busca mais abrangente em todo o universo de soluções possíveis do problema. A geração de uma nova solução inicial pode ser feita aleatoriamente ou utilizando-se informações das melhores soluções encontradas até o momento, chamada de “memória adaptativa” por Berger *et al*³.

Na figura 1 é apresentado o algoritmo básico de Busca Tabu. Nesta figura, identificou-se a solução vigente por s , a melhor solução vizinha por s' , a melhor solução encontrada após a geração de uma (nova) solução inicial é identificada por s^* e a melhor solução encontrada entre todas as soluções geradas é identificada por s^{best} . Nesta implementação, o algoritmo será executado até que o critério de parada da diversificação seja atingido. Foi

definido, como critério de parada, o número máximo de 1000 iterações sem que a solução s^{best} seja melhorada. O critério de parada da intensificação foi definido como 20 iterações, ou seja, se forem executadas 20 iterações sem que o valor de s^* seja melhorado, será realizada uma diversificação. Optou-se, nesta etapa do trabalho, por gerar as soluções iniciais aleatoriamente.

```

1 Enquanto o critério de parada da
  diversificação não é atingido,
  faça:
2   Gerar uma solução inicial (s)
3   Se primeira vez
4      $s^* = s$ 
5 Enquanto o critério de parada da
  intensificação não é atingido,
  faça:
6   Gerar a vizinhança de s E
   selecione a melhor solução
   vizinha ( $s'$ )
7    $s = s'$ 
8   Se  $s'$  é melhor que  $s^*$  então:
9      $s^* = s'$ 
10  Se  $s^*$  é melhor que  $s^{\text{best}}$  então:
11     $s^{\text{best}} = s^*$ 
12 Retornar  $s^{\text{best}}$ 

```

Figure 1. Algoritmo Básico de Busca Tabu

3. MOVIMENTOS NO PROBLEMA DE STEINER

Na implementação do problema de *Steiner* foram adotados dois movimentos básicos, para gerar soluções vizinhas: *i*) exclusão (ou desativação) de um nó de *Steiner*; *ii*) inclusão (ou ativação) de um nó de *Steiner*. Dependendo destes movimentos é que serão definidas as ligações entre arestas, ligando os nós de demanda aos nós de *Steiner* ativos e ligando estes últimos entre si.

A partir das figuras 2 e 3 são apresentadas as implicações de um movimento de exclusão. Na figura 2 é apresentado um exemplo de problema de Steiner, onde os nós de Steiner são identificados por retângulos (com números de 1 a 5) e os nós de demanda são identificados por círculos (com letras de *a* até *g*). Nesta figura é possível diferenciar as ligações entre nós de Steiner e a ligação de cada nó de demanda a um único nó de Steiner.

Na figura 3 é apresentada uma solução vizinha à solução do exemplo da figura 2, onde houve a exclusão do nó 3, que tornou-se inativo. Este

movimento de exclusão implicou na redefinição da conexão entre nós de Steiner para que o *backbone* fosse restabelecido, e, também, os nós de demanda *d* e *e* foram reconectados a outros nós de *Steiner*. Todas as novas ligações, feitas em consequência da exclusão do nó 3, foram representadas na figura 3 por linhas tracejadas.

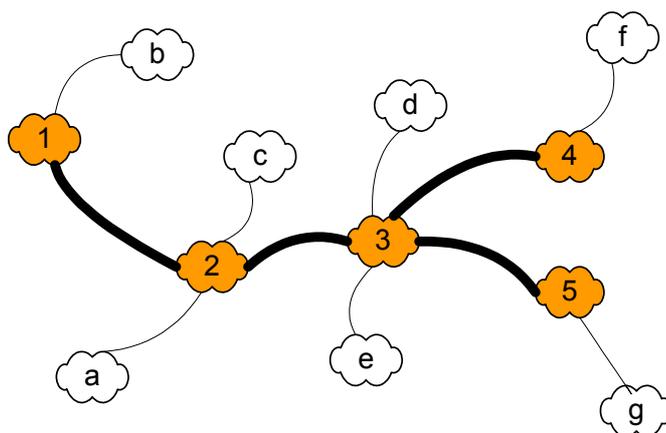


Figure 2. Exemplo de um problema de Steiner

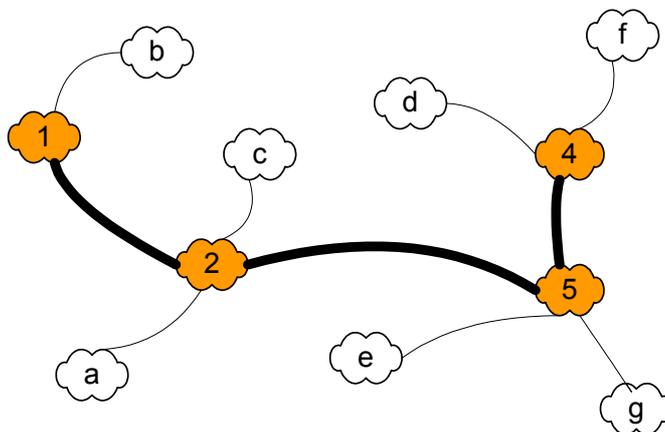


Figure 3. Nova Solução após eliminar o nó 3

Na figura 3 verifica-se que é necessário que as conexões entre nós de *Steiner* sejam reavaliadas para que sejam encontradas as melhores conexões, recompondo o *backbone*. Porém, esta reavaliação das conexões envolve apenas os nós de *Steiner* que estavam ligados ao nó excluído. Os nós de demanda que ficaram sem conexão nesta operação, também devem ser

reavaliados para serem conectados aos nós de *Steiner* que permaneceram ativos.

A recomposição do *backbone* será mais complexa se houver a inclusão (ativação) de um novo nó de *Steiner*, quando comparado ao movimento de exclusão, porque todos os nós de *Steiner* e nós de demanda deverão ser reavaliados. Isto ocorre porque o nó de *Steiner* incluído pode ser uma melhor opção de conexão para qualquer um dos nós do problema. Por isso, nesta implementação, adotou-se a estratégia de só considerar os movimentos de inclusão, na geração de soluções vizinhas, se nenhum movimento de exclusão gerar uma solução melhor que a solução vigente.

Além disso, foram adotadas duas formas para calcular o custo da solução: *i*) utilizando-se o algoritmo A^* (ou *branch and bound*), onde são avaliadas todas as conexões de nós, encontrando as arestas que retornam uma solução com o menor custo possível¹¹; e *ii*) utilizando-se o algoritmo *2-opt*⁷. O algoritmo *2-opt* é mais simples que o A^* , consistindo em trocar duas arestas não adjacentes da rota corrente x pelo único par de arestas que não pertence à rota e resulta em uma nova rota x' . Cada solução x tem uma vizinhança associada $N(x)$ que contém todas as soluções x'' que podem ser geradas pela aplicação de um movimento *2-opt* à solução x . O melhor movimento admissível em $N(x)$ é selecionado e a busca vai iterativamente de uma solução a outra até que algum critério de parada seja satisfeito. Na implementação deste trabalho, o algoritmo *2-opt* é usado na geração de soluções vizinhas a partir de um movimento de inclusão. Todas as soluções vizinhas geradas a partir de um movimento de exclusão e a melhor solução vizinha, obtida a partir de um movimento de inclusão, têm sua solução calculada pelo algoritmo A^* . Isto é feito porque, dado um conjunto de nós de *Steiner* ativos, o algoritmo A^* é capaz de identificar as arestas, a serem ativadas, que geram a solução de menor custo, o que o algoritmo *2-opt* não é capaz de garantir.

4. PRÉ-PROCESSAMENTO

A função de pré-processamento visa reduzir a complexidade do problema, eliminando informações redundantes do problema, sem que as características originais sejam alteradas. Ou seja, a solução ótima pode ser encontrada utilizando-se a instância original ou a reduzida, após a execução do pré-processamento. O papel do pré-processamento é o de identificar nós de *Steiner* e arestas que:

Necessariamente não farão parte da solução ótima, podendo ser eliminados do problema;

Necessariamente farão parte da solução ótima, não precisando serem tratados como variáveis de decisão;

Podem ser juntados compondo uma nova aresta que contenha um ou mais nós de Steiner e arestas.

No pré-processamento, o grau de cada nó de Steiner é identificado pelo número de arestas adjacentes (conectadas a ele). Além disso, todo nó de Steiner conectado a um nó de demanda foi chamado de nó terminal, porque estará *necessariamente* ativo. Os testes implementados, baseados em Uchoa, Aragao e Ribeiro^{12,13} foram:

- Teste NTD1 – Todo nó de grau 1 que não é terminal e sua aresta adjacente podem ser eliminados, sendo *necessariamente* inativos;
- Teste NTD2 – Todo nó não terminal u de grau 2 e suas duas arestas (u,v) e (u,w) adjacentes podem ser substituídos por uma única aresta (v,w) de custo $c(u,v) + c(u,w)$;
- Teste TD1 – Toda aresta adjacente a um nó terminal de grau 1 será necessariamente ativa, assim como o nó conectado a este terminal;
- Teste SD – Dada uma aresta (u,v) , conectada aos nós u e v . Identificando-se o custo do menor “caminho” entre estes dois nós como $B(u,v)$, tal que $B(u,v) < c(u,v)$, então a aresta (u,v) é redundante, devendo ser necessariamente inativa;
- BD3 – Dado um nó não terminal u de grau 3, adjacente aos nós v , w e z . Se:

$$\min\{B(v,w) + B(v,z); B(w,v) + B(w,z); B(z,v) + B(z,w)\} \leq c(u,v) + c(u,w) + c(u,z)$$

então pode-se mostrar que o nó u é no máximo de grau 2. Neste caso, se o nó u é no máximo de grau 2, as arestas (u,v) , (u,w) e (u,z) podem ser substituídas pelas arestas: (v,w) de custo $c(u,v) + c(u,w)$; (v,z) de custo $c(u,v) + c(u,z)$; e (w,z) de custo $c(u,w) + c(u,z)$. Apesar deste teste manter o número de arestas igual, há a possibilidade de o teste SD eliminar uma ou mais das novas arestas geradas a partir deste teste. O leitor interessado deve consultar Uchoa, Aragao e Ribeiro^{12,13} para encontrar outros testes e detalhes sobre os testes aqui apresentados.

As figuras 4 e 5 são apresentadas, ao final do artigo, para ilustrar o efeito do pré-processamento num problema de Steiner. A figura 4 representa o problema B1, apresentado no OR-Library¹⁴, com 50 nós de Steiner, 63 arestas e 9 nós terminais (nós de demanda), estes últimos identificados pela cor preta. Na figura 5 é apresentado o resultado da execução do pré-processamento, onde observa-se uma sensível redução no número de nós e arestas do problema.

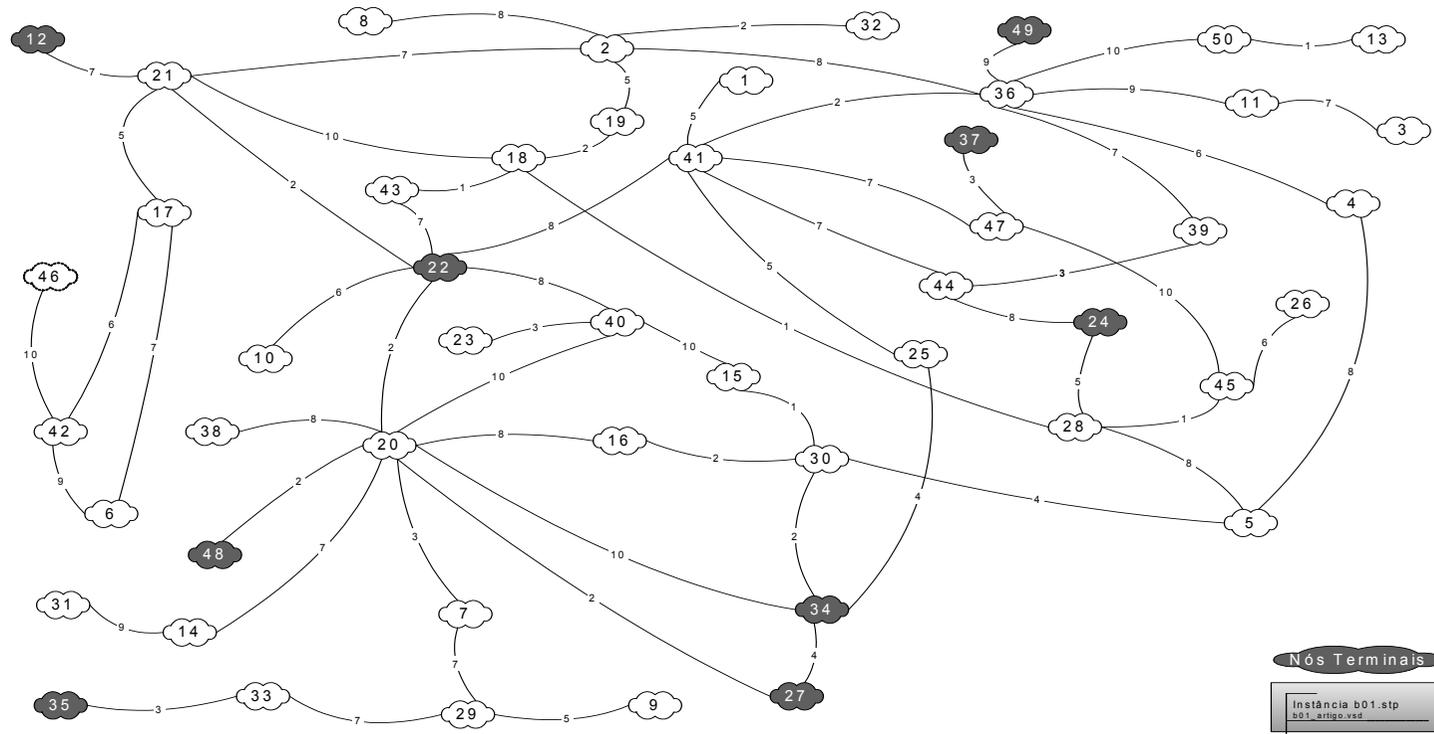


Figure 4. Nós de Steiner e arestas propostos no problema B01¹⁵.

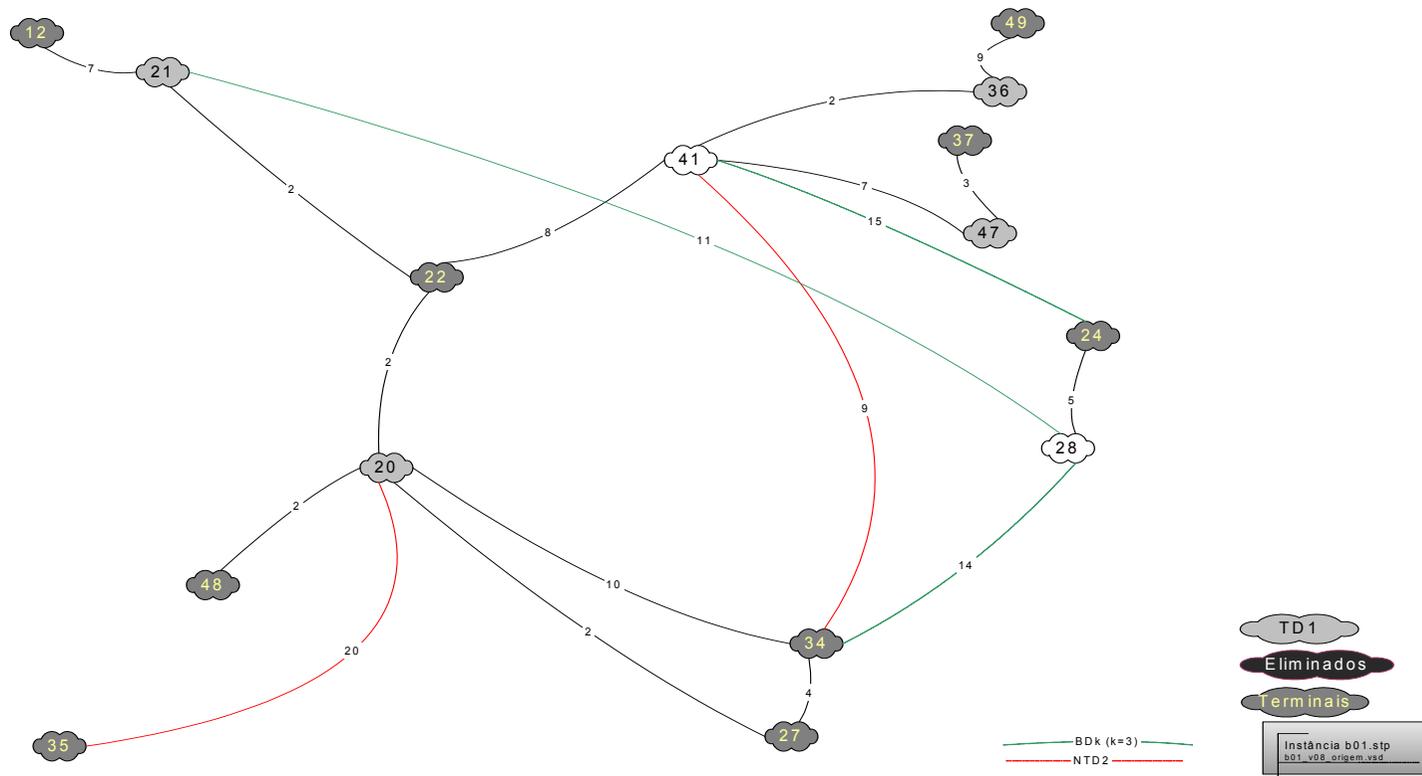


Figure 5. Nós de Steiner e arestas remanescentes do problema B1 após o pré-processamento.

5. RESULTADOS

Nesta seção são apresentados dois tipos de resultados. Na tabela 1, são apresentados dados dos problemas extraídos da OR-Library¹⁴ e os resultados do pré-processamento. N_i , A_i e T_i indicam, respectivamente, o número de nós de Steiner, arestas e nós terminais dados pelos problemas. N_p e A_p indicam, respectivamente, o número de nós de Steiner e de arestas após o pré-processamento para cada um dos problemas. “% N” e “% A” indicam, respectivamente, o percentual de redução no número de nós de Steiner e de arestas após o pré-processamento. N_{pt} indica o número de nós de Steiner que serão necessariamente ativos, após o pré-processamento, para cada um dos problemas, contando o nós terminais.

Tabela 1. Resultados do pré-processamento.

Nome	N_i	A_i	T_i	N_p	% N	A_p	% A	N_{pt}
B01	50	63	9	15	70	19	69,8	13
B02	50	63	13	18	64	25	60,3	16
B03	50	63	25	28	44	39	38,1	28
B04	50	100	9	28	44	64	36	9
B05	50	100	13	30	40	66	34	14
B06	50	100	25	39	22	86	14	26
B07	75	94	13	19	74,7	31	67,0	16
B08	75	94	19	23	69,3	35	62,8	20
B09	75	94	38	46	38,7	64	31,9	43
B10	75	150	13	48	36	118	21,3	16
B11	75	150	19	47	37,3	118	21,3	20
B12	75	150	38	59	21,3	129	14	40
B13	100	125	17	23	77	42	66,4	19
B14	100	125	25	39	61	63	49,6	31
B15	100	125	50	61	39	90	28	53
B16	100	200	17	65	35	155	22,5	18
B17	100	200	25	60	40	142	29	27
B18	100	200	50	74	26	168	16	51

Na tabela 2, são identificados, o valor da solução ótima (opt), o número médio de iterações e o valor médio da solução, com seus respectivos desvios padrões, encontrados sem (I_s e S_s) e com o pré-processamento (I_p e S_p), após 100 repetições de cada uma das instâncias.

Tabela 2. Comparação na solução de problemas.

Nome	opt	Is	Ss	Ip	Sp
B16	127	218,7 ± 271,8	132,4 ± 2,7	395,7 ± 408,9	127,3 ± 0,9
B18	218	135,5 ± 159,1	218,0 ± 0,2	49,2 ± 57,4	218,0 ± 0,2
C01	85	154,8 ± 62,9	122,0 ± 17,3	353,8 ± 325,1	87,4 ± 0,6

6. CONCLUSÕES E TRABALHOS FUTUROS

Os resultados preliminares, usando instâncias do OR-Library¹⁴, deixam claro que a abordagem proposta (uso da Busca Tabu com pré-processamento) é promissora. O próximo passo será testá-la como uma ferramenta de um sistema GIS, no projeto de redes urbanas. Também será preciso estudar valores adequados para os critérios de parada da diversificação e da intensificação. Além disso, a geração de soluções iniciais poderão se beneficiar das informações do pré-processamento, o que ainda não ocorre hoje.

7. REFERÊNCIAS

1. S. Chopra e M. R. Rao, "On the Steiner Tree Problem I & II", *Mathematical Programming* **64**, 209-246 (1994).
2. C. H. Chu, G. Premkumar e H. Chou, "Digital Data Network Design using genetic algorithms", *European Journal of Operational Research*, 140-158 (1999).
3. D. Berger, B. Gendrom e J. Y. Potvin, "Tabu Search for a Network Loading Problem", *Institute for Systems Research*, Technical Report 99-23, 1999.
4. J. Xu, S. Y. Chiu e F. Glover, "Using Tabu Search to solve the Steiner tree-star problem in telecommunication networks design", *Telecommunication Systems* **6**, 117-127 (1996).
5. C.C.Ribeiro, M.C.Souza, "Tabu Search for the Steiner Problem in Graphs", *Networks* **36**, 138-146 (2000).
6. J. Xu, S. Y. C. Chiu, F. Glover, "Tabu Search Heuristics for Designing a Steiner Tree based Digital Line Network", *University of Colorado*, Technical Report, 1995.
7. J. Xu, S. Y. C. Chiu, F. Glover, "Optimizing a ring-based private line telecommunication network using tabu search", *University of Colorado*, Technical Report, 1998.
8. M. P. Bastos, C. C. Ribeiro, "Reactive Tabu Search with Path-Relinking for the Steiner Problem in Graphs", *Essays and Surveys in Metaheuristics*, 39-58 (2001).
9. F. Glover, "Tabu Search – Part I", *ORSA Journal of Computing* **1**, 190-206 (1989).
10. F. Glover e M. Laguna, *Tabu Search*, Ed. Kluwer Academic (1997).
11. J. Pearl, *Heuristics: Intelligent search strategies for computer problem solving*, (Addison Wesley, 1984).
12. E. Uchoa, M. P. Aragao, C. C. Ribeiro, "Preprocessing Steiner Problems from VLSI Layout", *Networks* **40**, 38-50 (2002).
13. E. Uchoa, *Algoritmos para Problemas de Steiner com Aplicações em Projeto de Circuitos VLSI*, PUC-Rio, Tese de doutorado, 2001.
14. J. E. Beasley, "OR-Library", *Journal of the Operational Research Society* **41** (1990).