

## 064/2009 - UMA BUSCA LOCAL ITERATIVA APLICADA AO PROBLEMA DE STEINER EUCLIDIANO EM $\mathcal{R}^N$

**Vinícius Leal do Forte**

COPPE/UFRJ – Programa de Engenharia de Sistemas e Computação  
Caixa Postal 68511. CEP: 21941-972, Rio de Janeiro, RJ

[vlforte@cos.ufrj.br](mailto:vlforte@cos.ufrj.br)

**Nelson Maculan**

COPPE/UFRJ – Programa de Engenharia de Sistemas e Computação  
Caixa Postal 68511. CEP: 21941-972, Rio de Janeiro, RJ

[maculan@cos.ufrj.br](mailto:maculan@cos.ufrj.br)

**José André de Moura Brito**

Instituto Brasileiro de Geografia e Estatística – COMEQ/DPE/IBGE  
Av. República do Chile, 500 – 10º andar, CEP: 23031-170, Rio de Janeiro, RJ

[jose.m.brito@ibge.gov.br](mailto:jose.m.brito@ibge.gov.br)

**Flávio Marcelo Tavares Montenegro**

Instituto Brasileiro de Geografia e Estatística – COMEQ/DPE/IBGE  
Av. República do Chile, 500 – 10º andar, CEP: 23031-170, Rio de Janeiro, RJ

[flavio.montenegro@ibge.gov.br](mailto:flavio.montenegro@ibge.gov.br)

**Marcelo Lisboa Rocha**

Curso de Ciência da Computação - Universidade Federal do Tocantins  
Av. NS 15, S/N – ALCNO 14 - Bloco 02 – Sala 21, CEP: 77020-210, Palmas – TO

[marcelolisboarocha@yahoo.com.br](mailto:marcelolisboarocha@yahoo.com.br)

### Resumo

Neste trabalho, aplicamos uma metaheurística de Busca Local Iterativa (ILS) para a resolução do Problema de Steiner Euclidiano em  $\mathcal{R}^n$ . Este problema de alta complexidade computacional consiste em encontrar a menor árvore que interconecta um conjunto de pontos dados pertencentes ao  $\mathcal{R}^n$ , podendo-se acrescentar pontos extras que reduzam o tamanho total da árvore. O algoritmo ILS desenvolvido baseia-se nos chamados vetores topologia de Smith, que são estruturas concebidas originalmente para enumerar árvores de Steiner. Resultados computacionais demonstrativos da boa performance do método são apresentados e comparados àqueles obtidos pelos melhores métodos da literatura.

**Palavras-Chaves:** Problema de Steiner Euclidiano; Vetores Topologia; Metaheurísticas; ILS

### Abstract

In this work, we apply an Iterated Local Search (ILS) metaheuristic to solve the Euclidean Steiner Tree Problem in  $\mathcal{R}^n$ . This high computational complexity problem consists in finding the smallest tree that connects a set of given points in  $\mathcal{R}^n$ , allowing extra

points to be added that reduce the total length of the tree. The ILS we have implemented is based on the Smith's topology vectors, which are structures conceived originally to enumerate Steiner trees. Computational results showing the good performance of the proposed method are presented and compared to that ones provided by the best methods found in the literature.

**Keywords:** Euclidean Steiner Problem; Topology Vectors; Metaheuristics; ILS

## 1. INTRODUÇÃO

O presente trabalho trata do estudo e resolução do Problema de Steiner Euclidiano (PSE) em  $\mathcal{R}^n$ . Dados  $p$  pontos no espaço euclidiano  $n$ -dimensional, o PSE consiste em determinar a menor árvore que os interconecta, sendo permitido o uso de pontos extras, caso sua inclusão reduza o comprimento total da árvore[3]. Tais pontos são chamados pontos de Steiner. Detalhes sobre a parte histórica deste problema podem ser encontrados em [5]. Este problema pertence à classe NP-difícil, se considerarmos a métrica Euclidiana, e NP-completo com a sua discretização[2].

Devido a particularidades geométricas do problema no plano, tanto soluções exatas como heurísticas são encontradas com eficiência para  $n=2$ . Porém, as mesmas particularidades não se verificam para  $n > 2$ , de forma que a obtenção de uma solução exata, pelos algoritmos conhecidos atualmente [17], requer um tempo de processamento proibitivamente alto mesmo para instâncias relativamente pequenas, de cerca de 20 pontos dados. Assim sendo, têm-se recorrido, no caso  $n>2$ , ao desenvolvimento de algoritmos heurísticos e metaheurísticos. Tais algoritmos, até o presente momento, são entretanto ainda relativamente poucos, destacando-se entre estes um algoritmo GRASP [15], o qual tem obtido as melhores soluções para instâncias de porte médio (com cerca de algumas centenas de pontos).

Neste trabalho, apresentamos um conjunto de algoritmos implementados a partir do estudo da metaheurística ILS (Iterated Local Search), com o objetivo de também encontrar soluções de boa qualidade para o Problema de Steiner Euclidiano em  $\mathcal{R}^n$ . Tais algoritmos incorporam, em sua busca local, uma estrutura advinda do processo de enumeração do algoritmo exato de Smith [17].

A apresentação deste trabalho está dividida da seguinte forma: na segunda seção temos a definição do problema de Steiner, bem como as suas propriedades e características fundamentais. A seção 3 traz uma descrição do algoritmo exato proposto na literatura e relaciona as principais metaheurísticas propostas para encontrar soluções viáveis de boa qualidade. Na seção 4, apresentamos os aspectos principais da metaheurística ILS, e em seguida, descrevemos com detalhes o algoritmo proposto para o Problema de Steiner Euclidiano em  $\mathcal{R}^n$ . Finalmente, na última seção, temos a apresentação dos resultados computacionais para instâncias geradas aleatoriamente e suas comparações com outros algoritmos propostos na literatura.

## 2. O PROBLEMA DE STEINER EUCLIDIANO EM $\mathcal{R}^n$

Dados  $p$  pontos no espaço euclidiano  $n$ -dimensional, deve-se encontrar a menor árvore que os interconecta, considerando distâncias euclidianas, sendo permitida a inclusão de pontos extras (pontos de Steiner).

Algumas propriedades referentes à árvore obtida como solução do problema foram demonstradas por Gilbert e Pollak[3], quais sejam:

- (1) Para  $p$  pontos dados, o número máximo de pontos de Steiner é  $p - 2$ .
- (2) O número de ligações (grau) de um ponto de Steiner é 3.
- (3) As ligações provenientes de um ponto de Steiner perfazem, mutuamente, ângulos de  $120^\circ$ , e pertencem a um plano.

Introduzimos aqui algumas observações relativas às estruturas usadas na resolução do problema:

- O conjunto de ligações na árvore é chamado de topologia. Caso todos os pontos de Steiner dessa árvore tenham grau igual à 3, dizemos que ela possui uma topologia de Steiner.
- Se a inserção ou remoção de pontos de Steiner em uma árvore com topologia de Steiner não reduz o tamanho da árvore, então esta é uma árvore de Steiner.
- A menor das árvores de Steiner é a árvore mínima de Steiner, que por sua vez, corresponde à solução exata do problema.
- Quando consideramos o número de pontos de Steiner,  $s$ , igual à  $(p-2)$ , a topologia de Steiner é chamada de cheia. A figura 1 mostra algumas árvores de Steiner.

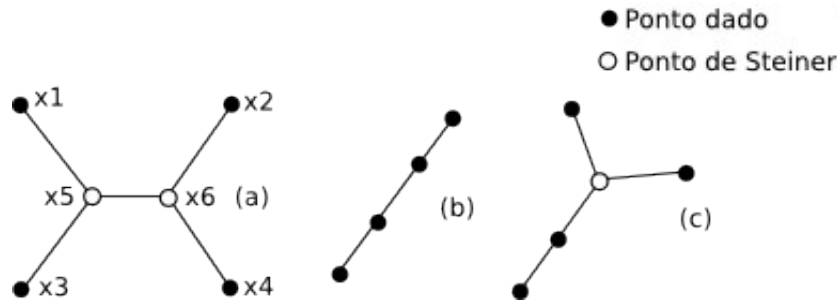


Figura 1 – Árvores de Steiner.

Para obter a menor árvore de uma dada topologia, temos que minimizar a soma das distâncias entre os  $p$  pontos no  $\mathbb{R}^n$  e os pontos de Steiner. Inicialmente, como a localização no  $\mathbb{R}^n$  dos pontos de Steiner não está definida, suas coordenadas serão as variáveis do problema. Com base na figura 1(a), temos o seguinte problema de otimização:

$$\text{Minimizar } \|x^1 - x^5\| + \|x^3 - x^5\| + \|x^2 - x^6\| + \|x^4 - x^6\| + \|x^5 - x^6\|$$

Apesar de a função convexa considerada na minimização não ser diferenciável podemos obter o seu mínimo através de um algoritmo proposto por Smith em [17].

Mesmo havendo um método eficiente,  $O(np)$ , para a minimização de uma topologia de Steiner, ainda resta encontrar a menor de todas as árvores de Steiner dentre todas as topologias de Steiner possíveis para os  $p$  pontos dados. Em [3] é apresentada uma expressão que fornece o número de topologias possíveis para  $p \geq 3$  pontos dados e  $s$  pontos de Steiner, qual seja:

$$C_{p,s+2} = \frac{(p+s-2)!}{s! 2^s}$$

Utilizando um argumento apresentado em [5], podemos considerar todas as topologias de Steiner como sendo cheias, já que toda árvore não-cheia é uma árvore com pontos de Steiner degenerados, ou seja, o tamanho de uma de suas ligações é nulo. Com isso, é suficiente considerar somente topologias de Steiner cheias na resolução do problema.

Todavia, mesmo considerando apenas as topologias cheias, temos que  $s = p - 2$ , o que implica, por sua vez, que a expressão acima é reduzida apenas a  $(2p-5)!!$  [11], onde  $(.)!!$  denota o duplo fatorial. A partir desta expressão, considerando-se uma instância de apenas 10 pontos, temos que o número de topologias de Steiner cheias é  $15!! = 1.3.5.7.9.11.13.15 = 2,027,025$ . Desta forma, devido ao rápido crescimento do número de estruturas a serem verificadas, um algoritmo de força bruta para encontrar a árvore mínima de Steiner torna-se inviável, à medida que o número de pontos dados aumenta. Esse aspecto combinatório torna o problema intratável por algoritmos exatos, sendo entretanto adequado o

uso de métodos heurísticos na tentativa de se encontrar soluções sub-ótimas de boa qualidade.

O critério mais usual para avaliar a qualidade de uma solução obtida por um algoritmo heurístico, consiste em utilizar a razão de Steiner  $\rho$ , que corresponde à razão entre o tamanho da árvore mínima de Steiner, obtida pelo algoritmo, e a árvore geradora mínima para os  $p$  pontos dados, que pode ser obtida em tempo  $O(p^2)$ .

Na próxima seção, descreveremos o funcionamento do único algoritmo exato proposto na literatura para a solução do problema de Steiner euclidiano em  $\mathfrak{R}^n$ .

### 3. O MÉTODO DE SMITH

Smith [17] propôs um algoritmo exato para resolver o problema de Steiner euclidiano em  $\mathfrak{R}^n$ . Esse algoritmo constrói a árvore mínima de Steiner realizando uma enumeração de todas as árvores de Steiner possíveis para uma particular distribuição de pontos dados. A enumeração é feita com auxílio de uma estrutura chamada vetor topologia, que representa uma topologia de Steiner cheia de forma biunívoca. Ou seja, dada uma topologia de Steiner, podemos representá-la de forma única com um vetor topologia, como também podemos obter uma única topologia de Steiner através de um vetor topologia.

Durante a enumeração, são realizados cortes baseados em estruturas parciais do vetor topologia, objetivando-se minimizar o número de topologias a serem consideradas no processo de enumeração. Uma topologia é minimizada a cada nó da enumeração pela resolução de  $n$  sistemas de equações lineares iterativamente, com o objetivo de obter a árvore de Steiner.

#### 3.1 VETORES TOPOLOGIA

A estrutura denominada vetor topologia, como o nome indica, consiste em um vetor de  $(p-3)$  componentes, sendo associado a cada posição  $i$  o valor  $a_i$ , tal que  $1 \leq a_i \leq 2i+1$ .

Obtemos um vetor topologia construtivamente a partir de um vetor nulo, o qual representa uma topologia trivial envolvendo 3 pontos selecionados dentre os  $P$  pontos dados, que denotamos respectivamente por  $x_1, x_2$  e  $x_3$ , e um ponto de Steiner  $p+1$ . As 3 arestas formadas também são numeradas respectivamente de 1 a 3, de acordo com os índices dos pontos dados aos quais se ligam.

Em seguida, incluímos um quarto ponto dado,  $x_4$ , fazendo a conexão deste com o ponto de Steiner  $p+2$ , que é inserido em uma das 3 ligações da topologia anterior. Esta inclusão dará origem a duas novas ligações: do quarto ponto ao ponto  $p+2$  (numerada por 4) e deste ao ponto  $p+1$  (numerada por 5). Este processo é então repetido para incluir, seqüencialmente, todos os demais  $p-4$  pontos dados. No caso geral, para  $i > 4$ , a inserção do ponto  $x_i$  dá origem a duas novas arestas, ambas ligadas ao ponto de Steiner  $p+i-2$ : a aresta  $2i-4$ , ligada ao ponto dado  $x_i$ , e a aresta  $2i-3$ , ligada a um ponto de Steiner.

A cada inserção de um ponto dado, o número associado a cada ligação escolhida para a inserção do ponto de Steiner é armazenado no vetor topologia, na posição relativa à quantidade de pontos inseridos.

Na figura 2 podemos observar as etapas de construção do vetor topologia para uma topologia de Steiner cheia com cinco pontos dados.

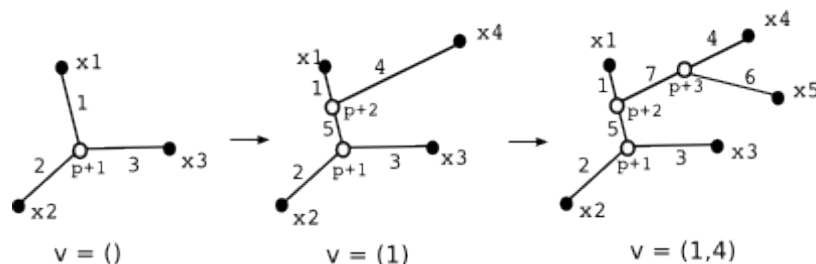


Figura 2 – Construção de um vetor topologia com cinco pontos dados.

### 3.2 MINIMIZAÇÃO DAS TOPOLOGIAS

No método de Smith, uma topologia é modelada através de um sistema de molas ideais [17]. Ao resolver este sistema iterativamente, produz-se uma solução de equilíbrio (configuração de energia potencial mínima) que, por sua vez, corresponde a uma árvore de tamanho mínimo.

A equação a seguir fornece as coordenadas de cada um dos  $k$  pontos de Steiner, considerando a  $i$ -ésima iteração do processo iterativo.

$$x_k^{(i+1)} = \frac{\sum_{j|(j,k) \in V} \frac{x_j^{(i+1)}}{\|x_j^{(i)} - x_k^{(i)}\|}}{\sum_{(j,k) \in V, j \in S} \frac{1}{\|x_k^{(i)} - x_j^{(i)}\|}}, p+1 \leq k \leq 2p-2$$

Sendo  $S$  o conjunto dos pontos de Steiner e  $V$  a topologia de Steiner representada pelo vetor topologia. Smith[17] provou que, no final do processo iterativo, o sistema converge para uma única solução.

### 3.3 ENUMERAÇÃO DAS TOPOLOGIAS

Nesta fase do algoritmo, um vetor topologia é construído componente a componente, até que se alcance o número máximo de  $(p-3)$  componentes. A cada elemento inserido, o vetor topologia resultante é minimizado, segundo o processo acima.

A minimização de estruturas parciais (vetores topologia com menos de  $(p-3)$  componentes) é fundamental para acelerar o processo de enumeração. Se uma estrutura parcial, após a minimização, apresenta um comprimento total maior que aquele da menor árvore encontrada até o momento, então qualquer árvore cujo vetor topologia tenha esta estrutura parcial como prefixo também possuirá um comprimento maior que a menor árvore de Steiner encontrada até então. Dessa forma, a enumeração não é continuada com esta estrutura parcial, e um procedimento *backtraking* é realizado para eliminar famílias de soluções ruins associadas a ela.

## 4. MÉTODOS HEURÍSTICOS

O algoritmo apresentado na seção anterior consome um tempo computacional muito elevado para encontrar soluções exatas mesmo para instâncias contendo poucos pontos, chegando a tornar-se inviável para instâncias com mais de 20 pontos.

Devido à alta complexidade computacional envolvida na resolução exata do Problema de Steiner Euclidiano em  $\mathfrak{R}^n$ , diversos métodos heurísticos têm sido desenvolvidos para solucioná-lo, produzindo, em geral, soluções de boa qualidade em um tempo computacional viável. A maior parte destes algoritmos é aplicável em casos particulares do  $\mathfrak{R}^2$  [19] (ver também o algoritmo exato de Cockayne et al.[1]) e  $\mathfrak{R}^3$  [13][14] [16]. Acrescenta-se, ainda, que poucas heurísticas foram desenvolvidas com objetivo de abordar pontos em dimensões maiores ou iguais a 3, destacando-se, dentre estas: o método de relaxação dinâmica estendida [12], o qual simula a evolução dinâmica de uma película de sabão sob tensão de forças superficiais e tem se mostrado a heurística mais rápida e entre

todas as propostas para o problema mesmo para instâncias grandes; uma metaheurística simulated annealing [8], cujas perturbações são similares aos movimentos utilizados em um método proposto por Thompson em [18] para resolver o problema de encontrar árvores filogenéticas, cuja solução pode ser encarada como uma árvore mínima de Steiner [5]; metaheurísticas tais como, algoritmos genéticos [10], algoritmo de otimização microcanônica [11] e GRASP [15], desenvolvidas para administrar buscas locais baseadas em vetores topologia [11] têm sido aplicadas com sucesso, encontrando soluções subótimas de boa qualidade, além de um número expressivo de ótimos globais, destacando-se o GRASP.

A seguir, apresentaremos com mais detalhes o processo de busca local baseado em vetores topologia utilizado nas metaheurísticas citadas anteriormente, tendo em vista que o mesmo foi usado na metaheurística ILS implementada neste trabalho.

#### 4.1 BUSCA LOCAL BASEADA EM VETORES TOPOLOGIA

Heurísticas de busca local têm sido amplamente aplicadas, com êxito, a problemas combinatórios[4]. Partindo de uma solução inicial, a busca local iterativamente a substitui por uma solução melhor pertencente a sua vizinhança, segundo o objetivo do problema[4]. A vizinhança de uma solução é formada por todas as soluções que podem ser obtidas através da modificação (movimento, perturbação, ...) desta solução.

Nos trabalhos [10] e [11], Montenegro e Maculan desenvolveram uma busca local para o Problema de Steiner Euclidiano em  $\mathbb{R}^n$  baseada na estrutura de vetor topologia usada no processo de enumeração do algoritmo exato de Smith [17]. A escolha desta estrutura é atribuída à unicidade na representação de cada árvore de Steiner possível (para uma dada instância) por um vetor topologia, possibilitando, desta forma, a busca da árvore mínima de Steiner no espaço dos vetores topologia.

A busca local proposta neste trabalho consiste dos seguintes passos principais: dado um vetor topologia  $v$ , cuja razão de Steiner associada à árvore de Steiner é  $\rho$ , é escolhida aleatoriamente uma posição  $i$  desse vetor e o valor  $v(i)$  é alterado, criando-se assim um vetor  $v'$  vizinho de  $v$ . Após a minimização de  $v'$ , usando o método de minimização de topologias apresentado na seção 2, verificamos se  $\rho' < \rho$ , em qual caso efetuamos o movimento  $v \leftarrow v'$  e  $\rho \leftarrow \rho'$  e repetimos o processo novamente e, em caso contrário, escolhemos um novo par  $(i, v(i))$  e tentamos novamente. Após um critério de parada ser satisfeito (normalmente o número máximo de iterações sem melhora de  $\rho$ ) interrompemos a busca local.

Como a quantidade de valores possíveis em cada posição  $i$  do vetor topologia é  $2i+1$ , temos que o número de vizinhos que podem ser obtidos na busca é igual a  $\sum_{i=1}^{p-3} 2i+1$ , ou seja, é de ordem  $O(p^2)$ .

Porém, como toda busca local, esta tende a ficar “presa” em mínimos locais, sendo necessária a aplicação de uma metaheurística para evitar que a heurística pare na presença do primeiro mínimo local que encontrar. O pseudo-código relacionado à busca local em vetores topologia é apresentado na figura 3.

```

Busca Local ( v□, rho )
v = v□
Para j até maximo_vizinhos Faça
    i = rand(1 , p-3) //rand(a,b)=gera um número pseudo-aleatório entre a e b//
    v'(i) = rand(1, 2i+1)
    aux = v(i)
    v(i) = v'(i)
    rho' = Minimiza(v)
    Se rho'<rho
        j = 0
        rho = rho'
    Senão
        v(i) = aux
    Fim-Se
Fim- Para
Fim- Busca Local

```

Figura 3 - Pseudo-Código da Busca Local em Vetores Topologia

## 5. METODOLOGIA PROPOSTA

Nesta etapa do texto iremos detalhar o algoritmo que aplicamos na resolução do Problema de Steiner Euclidiano em  $\mathbb{R}^n$ , bem como introduzir em aspectos gerais a metaheurística ILS utilizada no algoritmo.

### BUSCA LOCAL ITERATIVA (ILS)

A Busca Local Iterativa (ILS) é uma metaheurística recente. Inicialmente denominada como *Large-step Markov chain* [9], foi criada por Otto e Felten em 1991 e popularizada em 2002, quando da publicação de um capítulo escrito por Lourenço, Martin e Stützle no livro *Handbook of Metaheuristics* [7].

Após uma fase de inicialização onde uma solução inicial é construída, a Busca Local Iterativa aplica iterativamente o procedimento de busca local, que alterando a solução encontrada (mínimo local) através do procedimento de perturbação, utiliza essa solução para inicializar a busca local da próxima iteração, se esta for aceita pelo procedimento de aceitação. A figura 4 mostra o pseudo-código deste algoritmo.

```

Busca Local Iterativa ( )
s□ = Inicialização( )
s = Busca_local( s□ )
Repita
    s' = Perturbação( s )
    s'*=Busca_local( s' )
    s = Aceitação( s , s'* )
Até ( critério de parada )

```

Figura 4 – Pseudo-código básico da metaheurística ILS

A justificativa para o uso do procedimento de perturbação sobre sucessivas buscas locais é dada através do seguinte argumento: Seja  $S$  o conjunto de todas as soluções viáveis para um problema e  $S^*$ , tal que  $S^* \subseteq S$ , o conjunto dos mínimos locais obtidos pela aplicação da busca local a diferentes elementos de  $S$ . Temos que, dado  $s \in S$ , uma solução viável para o problema, encontramos um mínimo local  $s^* \in S^*$  através da aplicação da busca local à  $s$  e obtemos  $s''$  realizando uma perturbação em  $s^*$  que ao ser utilizada como solução inicial para a busca local é retornado o mínimo local  $s'^*$ . Com isso, se  $s^* \neq s'^*$  dizemos que  $s^*$  e  $s'^*$  são

vizinhos no conjunto de mínimos locais  $S^*$ . Dessa forma, se iterativamente aplicarmos o procedimento de perturbação e a busca local como realizado neste parágrafo estaremos encontrando vizinhos em  $S^*$ .

Este argumento mostra que o ILS realiza uma busca local nos mínimos do problema, indo muito além da simples aplicação da heurística e, conseqüentemente, retornando resultados, em geral, mais próximos do valor ótimo.

Para que o algoritmo realize a busca local no conjunto dos mínimos locais de forma eficiente é necessário que a perturbação não seja muito grande, nem muito pequena. Se esta for muito pequena é possível que a solução perturbada encontre novamente o mínimo local, cuja perturbação a gerou, e se a perturbação for muito grande, o algoritmo torna-se equivalente à metaheurística de reinicialização aleatória[7], onde a busca local é executada inúmeras vezes com soluções iniciais distribuídas aleatoriamente sobre o espaço de soluções.

Uma solução  $s'^*$  só é utilizada como solução inicial para a busca local da iteração seguinte se for aceita pelo procedimento de aceitação, o qual pode ser definido de diferentes formas compreendidas entre dois critérios extremos de aceitação. O primeiro critério corresponde à aceitação de mínimos locais que sejam melhores que a melhor solução obtida até a atual iteração. O segundo, representando o extremo oposto do primeiro critério, consiste em sempre aceitar o mínimo local obtido na iteração como solução inicial da iteração seguinte. Outros critérios podem ser desenvolvidos entre esses dois extremos, podendo-se incorporar informações históricas com intuito de variar gradativamente entre os critérios extremos, com o objetivo de intensificar (primeiro critério) ou diversificar (segundo critério) a busca por soluções menores.

O funcionamento deste algoritmo pode ser esclarecido pela observação da figura 5.

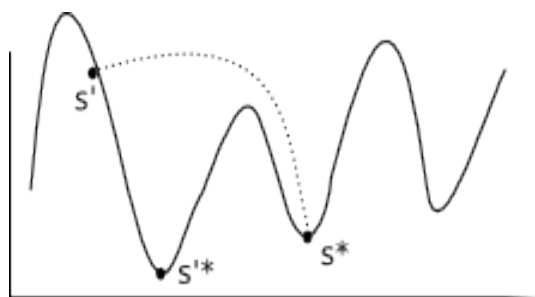


Figura 5 – Iteração básica do ILS. Uma solução  $s^*$  é perturbada e a busca local é aplicada a sua perturbação  $s'$ , encontrando uma solução menor  $s'^*$ .

#### **BUSCA LOCAL ITERATIVA (ILS) APLICADA AO PROBLEMA DE STEINER EUCLIDIANO EM $\mathbb{R}^n$**

Nesta seção, descreveremos com detalhes como cada um dos procedimentos envolvidos no ILS foram desenvolvidos em sua aplicação na resolução do Problema de Steiner Euclidiano em  $\mathbb{R}^n$ .

O procedimento de inicialização constrói uma solução inicial derivada da árvore geradora mínima (AGM), construída para os pontos dados, através da inserção de pontos de Steiner. Considerando apenas um ponto da árvore geradora mínima uma única vez, se o mesmo tiver mais de duas ligações com outros pontos dados, então duas ligações são selecionadas e um ponto de Steiner é inserido à estrutura. Esta inserção é realizada fazendo-se com que o novo ponto de Steiner receba as duas ligações escolhidas e, depois, conectando-o ao ponto dado através de uma nova aresta. Assim, substituindo-se as duas ligações escolhidas



previamente por uma única ligação, o ponto dado tem seu grau reduzido de uma unidade. Este processo é repetido até que o ponto dado tenha apenas uma ligação (grau um). A construção, então, é continuada a partir de outro ponto da AGM que possua duas ou mais ligações (grau maior que um), e prossegue até que todos os pontos dados na estrutura resultante possuam grau um. Na figura 6 é realizado o procedimento de inicialização para um conjunto de pontos e sua AGM.

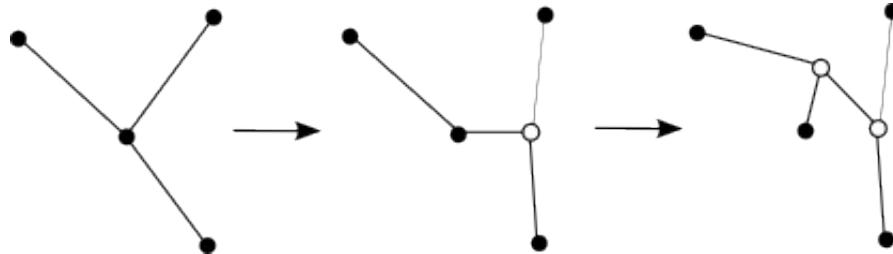


Figura 6 – Inserção de pontos de Steiner em uma Árvore Geradora Mínima.

A busca local utilizada em nosso algoritmo é a mesma apresentada na seção 4.1, ou seja, uma busca local baseada em vetores topologia, a qual tem sido usada com sucesso em metaheurísticas aplicadas a instâncias de tamanho médio (até cerca de 250 pontos dados) [11] [15].

As perturbações são realizadas de forma aleatória. Fazendo  $v \rightarrow v'$  e para  $k \in [1, (p-3)]$ , escolhemos aleatoriamente com distribuição uniforme  $k$  valores inteiros no intervalo  $[1, p-3]$ , formando com esses elementos o conjunto  $\{\pi_1, \pi_2, \dots, \pi_k\}$  das  $k$  posições a serem modificadas no vetor topologia  $v'$  para que este seja uma perturbação de  $v$ . Em seguida, substituímos, para cada  $\pi_i \in \{\pi_1, \pi_2, \dots, \pi_k\}$ , o valor  $v'(\pi_i)$  por um valor escolhido aleatoriamente, com distribuição uniforme, no intervalo  $[1, (2\pi_i+1)]$ , tal que após a substituição  $v'(\pi_i) \neq v(\pi_i)$ . Como consequência, temos uma perturbação  $v'$  de tamanho  $k$  do vetor topologia  $v$ . O pseudo-código referente a este processo de perturbação pode ser visto na figura 7.

```

v' = Perturbação(v, k)
v' = v
Para i = 1 até k Faça
    h(i) = rand(1, p-3) // h(i) ≠ h(j) ∀ i, j ∈ {1, ..., k}
    v'(h(i)) = rand(1, 2(h(i))+1) // v(h(i)) ≠ v'(h(i))
Fim-Para
Fim Perturbação

```

Figura 7 – Pseudo-código do procedimento de perturbação.

O procedimento de aceitação, responsável por possibilitar ou não a mudança da região de busca no algoritmo, foi desenvolvido de duas formas diferentes, levando à implementação de dois algoritmos (ILS\_1 e ILS\_2), que diferem somente neste procedimento.

Para ambos os procedimentos, um conjunto E foi criado para armazenar os cinco vetores topologias associados às melhores soluções obtidas até o momento. Toda vez que um vetor é aceito, este substitui o vetor associado à pior solução do conjunto E. Este conjunto é denominado na literatura como conjunto elite.

Ainda contemplando os dois procedimentos de aceitação, se um vetor topologia  $v''$  possuir após a minimização o valor  $\rho_{v''}$  menor que o valor  $\rho_{v^*}$  para  $v^*$ , então o vetor topologia  $v''$  é aceito, substituindo o vetor do conjunto E com maior  $\rho$ .

No algoritmo ILS\_1, se o vetor não obedece o critério citado acima, então para que

este vetor topologia,  $v''$ , seja aceito é preciso que  $\rho_{v''} \in [\mu - 2\sigma, \mu + 2\sigma]$ , onde  $\mu$  é a média dos valores de  $\rho$  pertencentes ao conjunto E e  $\sigma$  o desvio padrão dos valores de  $\rho$  pertencentes a E. Uma vez que um valor é aceito, o conjunto E é atualizado e, com isso, os valores de  $\mu$  e  $\rho$  também.

Para o algoritmo ILS\_2 foi aplicado o mesmo critério de aceitação usado na metaheurística simulated annealing[6] e também nos primeiros trabalhos relacionados ao ILS[9]. Tal critério consiste na aceitação de uma solução sob uma determinada probabilidade, ou seja, para  $v$  e  $v^*$  vetores topologias, o vetor  $v''$  será aceito se para um valor  $r$ , obtido aleatoriamente com distribuição uniforme no intervalo  $(0,1]$ , for igual a:

$$e^{\frac{-(\rho_{v''} - \rho_{v^*})}{t}} \geq r$$

onde  $t$  simboliza metaforicamente a temperatura do processo de resfriamento de metal[6], que em nosso algoritmo recebe inicialmente o valor  $t=1$  e gradativamente é reduzido, resolvendo a equação  $t = \alpha t$  a cada iteração, para  $\alpha=0,99$ .

## 6. RESULTADOS COMPUTACIONAIS

Nesta seção iremos apresentar alguns resultados obtidos pela aplicação dos dois algoritmos ILS implementados na resolução do Problema de Steiner Euclidiano, considerando instâncias de pequeno e médio porte. Também comparamos os resultados com outros algoritmos citados no texto, como o algoritmo de Otimização Microcanônica ( $\mu O$ ) [11], o GRASP [15] e o algoritmo exato para instâncias pequenas.

Os algoritmos foram implementados em linguagem C e compilados com GCC 4.2 em um computador dotado de processador Xeon Quad 3.2 Ghz, com 16 Gb de memória RAM, executando o sistema operacional Ubuntu 8.04.

Com base em alguns teste preliminares, o número máximo de vizinhos para a busca local foi escolhido como  $5p$ , sendo  $p$  a quantidade de pontos dados. A quantidade de valores do vetor topologia a serem modificados na perturbação, ou seja, o valor de  $k$  foi escolhido como  $\lfloor \frac{p}{2} \rfloor - 1$ . Armazenamos no conjunto elite, E, somente cinco vetores topologia.

Para testar os algoritmos foram utilizados seis tipos de instâncias com: 8, 9, 10 e 11 pontos pertencentes ao  $\mathbb{R}^3$ , 10 pontos pertencentes ao  $\mathbb{R}^4$  e 10 pontos pertencentes ao  $\mathbb{R}^5$ . Foram consideradas as mesmas 1000 instâncias de cada tipo geradas, em [11], com os pontos distribuídos aleatoriamente em um cubo, ou hipercubo, unitário.

A tabela seguinte apresenta o valor médio de  $\rho$ , o seu desvio padrão, o tempo de CPU e o número de acertos em relação à solução ótima para cada algoritmo aplicado ao conjunto de instâncias.

p		8 Pontos em $\mathbb{R}^3$	9 Pontos em $\mathbb{R}^3$	10 Pontos em $\mathbb{R}^3$	11 Pontos em $\mathbb{R}^3$	10 Pontos em $\mathbb{R}^4$	10 Pontos em $\mathbb{R}^5$
Exato	Média $\rho$	0,946761	0,946397	0,946758	0,946831	0,92778	0,91163
	Desvio Padrão	0,019507	0,017754	0,017531	0,016693	0,018336	0,018687
	Tempo de CPU	1,4	7,8	43,0	240,0		
$\mu O$	Média $\rho$	0,947507	0,947494	0,948521	0,948724	0,92951	0,91314
	Desvio Padrão	0,019637	0,017884	0,017828	0,016434	0,018773	0,019077
	Tempo de CPU	6,963	9,4704	12,065	14,76816	13,1142	14,0616
	Nº de Acertos	908	845	766	742	751	741
GRASP	Média $\rho$	0,947015	0,947387	0,947571	0,948109		
	Desvio Padrão	0,019563	0,018268	0,021365	0,016261		
	Tempo de CPU	13,48	16,755	20,823	23,873		

	Nº de Acertos	931	902	878	885		
Multstart soap-film	Média $\rho$	0,94756	0,94745	0,94809	0,94795	0,92910	0,91285
	Desvio Padrão	0,019704	0,17973	0,017639	0,015999	0,018556	0,019192
	Tempo de CPU	0,50348	0,57830	0,94809	0,75340	0,81175	0,96380
	Nº de Acertos	64	15	2	4	119	357
ILS_1	Média $\rho$	0,94676	0,9464	0,94677	0,94691	0,92781	0,91165
	Desvio Padrão	0,019507	0,017745	0,01753	0,015837	0,018325	0,018692
	Tempo de CPU	13,13	23,944	41,6	59,84	45,6	49,056
	Nº de Acertos	1000	999	996	974	992	991
ILS_2	Média $\rho$	0,94683	0,94657	0,94682	0,94702	0,92789	0,91169
	Desvio Padrão	0,019488	0,017761	0,01753	0,01589	0,018377	0,018698
	Tempo de CPU	6,9648	12,744	22,5376	32,8	25,0592	26,7168
	Nº de Acertos	988	966	981	941	975	980

Tabela 1 – Quadro comparativo com a média do valor das soluções, seu desvio padrão, o tempo de CPU em segundos e o número de solução ótimas encontradas.

Já que os resultados das heurísticas concorrentes foram aqueles diretamente reportados de [15], foi necessário realizar um escalamento do tempo de CPU, pois cada algoritmo foi executado em máquinas com poder de processamento diferentes. Utilizamos um escalamento que multiplica o tempo de CPU dos algoritmos implementados neste trabalho pela razão entre o clock do processador usado neste experimento (3200 Mhz) e o clock do processador usado nos demais experimentos (200 Mhz).

Para o algoritmo GRASP, não apresentamos os dados referentes às instâncias de 10 pontos para dimensões 4 e 5, já que estes não constam da publicação original[15].

Observando a tabela 1 podemos concluir que os algoritmos propostos neste trabalho produziram resultados bem melhores que as metaheurísticas concorrentes, encontrando valores ótimos na maioria dos casos e tanto o valor médio de  $\rho$  quanto o desvio padrão bem próximos dos valores exatos correspondentes.

Comparando as duas metaheurísticas ILS implementadas neste trabalho, observamos que o algoritmo ILS\_1 obteve resultados melhores que o algoritmo ILS\_2, no entanto, com tempo de CPU maior que os obtidos pelas outras heurísticas. Mesmo o algoritmo ILS\_2 não tendo apresentado uma performance qualitativamente tão boa quanto a do ILS\_1, aquele ainda encontra soluções melhores que as das outras heurísticas em um tempo próximo ao utilizado pela heurística GRASP.

Obtivemos também alguns resultados preliminares para instâncias aleatórias de tamanho médio que permitem inferir um funcionamento razoável do ILS também para instâncias desse porte. Por exemplo, para instâncias com 50 e 100 pontos em dimensão 3, obtivemos, utilizando o algoritmo ILS\_1, razões de Steiner médias de  $\rho = 0,95538$  e de  $\rho = 0,95617$ , respectivamente. Para instâncias de mesmos tamanhos, o algoritmo de melhor performance em instâncias de médio porte encontrado na literatura [15] obteve razões médias de  $\rho = 0,940796$  e de  $\rho = 0,948964$ , respectivamente. Em que pese os conjuntos de instâncias usados nos testes preliminares que realizamos serem diferentes daqueles usados em [15], estes resultados indicam uma performance do ILS\_1, em termos de qualidade, já razoavelmente próxima às dos melhores algoritmos da literatura.

Tais resultados, no entanto, ainda precisam ser melhorados, através de testes mais exaustivos para ajuste dos parâmetros de busca, antes de serem comparados de maneira direta e sistemática com as instâncias tratadas na literatura. O ajuste dos parâmetros deve contemplar não apenas a qualidade da solução, mas também o tempo de processamento, que, no momento, ainda está elevado. No exemplo anterior, o tempo de processamento demandado pelo ILS\_1 foi, em média, da ordem de 124,44 segundos, para instâncias com 50 pontos, e de 885,51 segundos, para instâncias com 100 pontos. Apesar disso, tendo em conta o nível de

qualidade já alcançado nos testes preliminares, acreditamos que o algoritmo ILS se mostra bastante promissor no tratamento de instâncias também de médio porte, podendo vir a alcançar um padrão de performance semelhante àquele referente às instâncias menores.

Com os resultados obtidos, podemos concluir que as metaheurísticas ILS desenvolvidas neste trabalho representam alternativas viáveis para encontrar soluções sub-ótimas de boa qualidade para o Problema de Steiner Euclidiano em  $\mathbb{R}^n$ . Em trabalhos futuros, pretendemos explorar diferentes alternativas para o procedimento de perturbação, bem como desenvolver um método híbrido considerando a metaheurística desenvolvida neste trabalho e o método de Smith, além da possível inclusão de características presentes em outras metaheurísticas.

## 7. REFERÊNCIAS BIBLIOGRÁFICAS

- [1]Cockayne, E. J. e Hewgill, D. E. (1986). Exact Computation of Steiner Minimal Trees in the Plane. Inf. Process. Lett., 22, pp. 151-156
- [2]Garey, M. R. ,Graham, R. L. e Johnson, D. S. (1977). The Complexity of Computing Steiner Minimal Trees. Siam J. Appl. Math., 32, pp. 835-859
- [3]Gilbert, E. N. e Pollak, H. O. (1968). Steiner Minimal Trees. SIAM Journal on Applied mathematics, 16, pp. 323-345
- [4]Glover, F. e Laguna, M. (1997). Tabu Search. Kluwer Academic Publishers. Boston.
- [5]Hwang, F. K. , Richards, D. S. e Winter, P. (1992). The Steiner Tree Problem. In: Annals of Discrete Mathematics, 53, North-Holland, Amsterdam.
- [6]Kirkpatrick, S. , Gellat, D. C. e Vecchi, M. P. (1983). Optimization by Simulated Annealing. Science, 220, pp. 671-680
- [7]Lourenço, H. R. , Martin, O. e Stützle T. (2002). Iterated Local Search. In F. Glover and G. Kochenberger, editors, Handbook of Metaheuristics, volume 57 of International Series in Operations Research & Management Science, pages 321-353. Kluwer Academic Publishers, Norwell, MA.
- [8]Lundy, M. (1985). Applications of the Annealing Algorithm to Combinatorial Problems in Statistics. Biometrika, 72, pp. 191-198
- [9]Martin, O. , Otto, S.W. e Felten, E.W. (1991). Large-step Markov chains for the traveling salesman problem. Complex Systems, 5, pp. 299-326
- [10]Montenegro, F. M. T. e Maculan, N. (2000). A Genetic Algorithm for the Euclidean Steiner Problem in  $\mathbb{R}^n$ . In: Proc. X Cong. Ibero-latinoamericano Inv. Oper. CLAIO, Cidade do México.
- [11]Montenegro, F. M. T. (2001). Heurísticas Para o Problema de Steiner Euclidiano em  $\mathbb{R}^n$ . Tese de D. Sc., COPPE - Universidade Federal do Rio de Janeiro.
- [12]Montenegro, F. M. T. , Maculan, N. , Plateau, G. e Boucher, P. (2001). New Heuristics for the Euclidean Steiner Problem in  $\mathbb{R}^n$ . In: C. Ribeiro and P. Hansen, editors, Essays and Surveys in Meyaheuristics, volume 15 of Operations Research/Computer Science Interfaces series, pages 509-524. Kluwer Academic Publishers, Boston.
- [13]Pereira, M. (1998). Proposta e Avaliação para o Problema de Steiner Geométrico em Duas e Três Dimensões. Dissertação de M. Sc., UFPE.
- [14]Ravada, S. e Sherman, A. T. (1994). Experimental Evaluation of a Partitioning Algorithm for the Steiner Tree Problem in  $\mathbb{R}^2$  and  $\mathbb{R}^3$ . Networks, 24, pp. 409-415

- [15]**Rocha, M. L. (2008)**. Aplicações de Algoritmos Paralelos e Híbridos para o Problema de Árvore de Steiner Euclidiana no  $\mathbb{R}^n$ . Tese de Doutorado. COPPE - Universidade Federal do Rio de Janeiro.
- [16]**Smith, J. M. , Weiss, R. e Patel, M. (1995)**. An  $O(N^2)$  Heuristic for Steiner Minimal Trees in  $E^3$ . Networks, 25, pp. 273-289
- [17]**Smith, W. D. (1992)**. How to Find Steiner Minimal Tree in Euclidean d-Space. Algorithmica, 7, pp. 137-178
- [18]**Thompson, E. A. (1973)**. The Method of Minimum Evolution. Annals of Human Genetics, 36, pp. 333-340
- [19]**Zachariasen, M. (1999)**. Local Search for the Steiner Tree Problem in the Euclidean Plane. European Journal of Operational Research, 119, pp. 282-300