

Um algoritmo do tipo *Beam Search* para alocação de células a centrais de telefonia celular

Cassilda Maria Ribeiro (UNESP-SP/Brasil) - cassilda@feg.unesp.br
• Departamento de Matemática, Av. Ariberto Pereira da Cunha, 333, Pedregulho, 12516-410, Guaratinguetá-SP
Aníbal Tavares de Azevedo (UNICAMP-SP/Brasil) - anibal.azevedo@fca.unicamp.br,
Rodolfo Florence Teixeira Junior (UFSCAR – SP/Brasil) - rodolfo.florence@ufscar.br

RESUMO O problema de atribuição de células as centrais de telefonia móvel é um problema combinatório do tipo NP-completo. Isto significa que a alternativa para a solução deste tipo de problema é a utilização de métodos heurísticos, pois eles permitem que se encontre uma solução de boa qualidade num tempo computacional bastante satisfatório. Neste artigo é apresentado um algoritmo de *Beam Search*, para resolver o Problema de Atribuição de Células às Centrais de Telefonia Móvel. São também apresentadas algumas modificações deste algoritmo, que permitem sua implementação em paralelo. Foram feitos testes computacionais com problemas considerados grandes obtendo-se boas soluções. Observou-se que as versões modificadas do algoritmo, em muitos casos, produziram soluções de melhor qualidade.

Palavras-chave Otimização Combinatória. Problema de Atribuição. *Beam Search*.

ABSTRACT *This paper presents the optimization problem of assigning cells to switches in a cellular mobile network. This problem is known as an NP-Complete problem, which means the alternative to solving this problem is using heuristic methods, which can lead to good feasible solutions to problems of a certain size in satisfactory computational time. This paper proposes a Beam Search method to solve this problem. Some modifications are also presented, which enable a parallel approach. Results show that the proposed method achieves good solutions for large-scale instances. It also can be noted that the algorithm modification produced better solutions in most cases.*

Keywords *Combinatorial Optimization. Assignment problem. Beam Search.*

1. INTRODUÇÃO

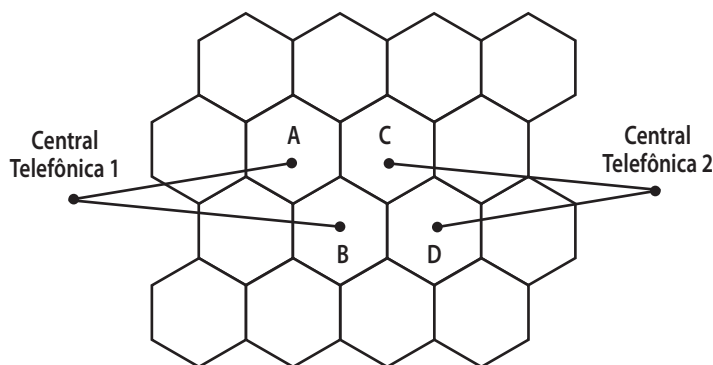
O crescimento prodigioso das telecomunicações nos últimos anos faz com que ela esteja cada vez mais presente na sociedade moderna. A popularização de um dos equipamentos mais extraordinários de todos os tempos, o telefone celular, acirrou a competição entre as operadoras do serviço, fazendo com que a busca por uma rede de telefonia celular cada vez mais eficiente e com capacidade de atender a demanda sempre crescente, seja imprescindível.

Numa rede de telefonia celular móvel os serviços são oferecidos por regiões. Essas regiões são chamadas zonas de cobertura e são divididas em pequenas áreas geográficas denominadas de células. Então, pode-se dizer que as células são as unidades básicas do sistema de telefonia celular típica. Cada célula é responsável por cobrir (oferecer serviços) a um certo número de assinantes. Para tanto, em cada célula existe uma estação básica (antena) que faz a comunicação entre os assinantes pertencentes à mesma célula. Além disso, cada célula está conectada a uma central de telefonia móvel (central de comutação), que é a responsável por fazer as ligações entre os assinantes de duas células diferentes.

As células possuem, por razões de cálculos, o formato hexagonal, formando algo parecido com uma colméia de abelhas. Cada central de telefonia móvel tem capacidade de comutar um determinado número de assinantes. Assim, várias células podem ser conectadas a uma mesma central, bastando que se respeite a capacidade máxima da mesma. Contudo, uma célula não pode estar conectada a mais de uma central ao mesmo tempo.

Na Figura 1 é apresentado um exemplo onde as células A e B estão conectadas à Central Telefônica 1 e as células C e D estão conectados à central telefônica 2.

Figura 1 – Divisão do sistema de telefonia celular em células.



Fonte: Elaborado pelos autores.

As antenas instaladas em cada célula utilizam canais de rádio para fazer a comunicação entre seus assinantes e, para evitar interferências, duas células adjacentes utilizam grupos de canais de rádios diferentes. Os telefones celulares possuem transmissores de baixa potência e as estações básicas também transmitem em baixa potência. Isto é necessário para que o consumo de energia do telefone celular, que normalmente são alimentados por bateria, seja relativamente baixo. Em resumo, baixa potência significa poder usar baterias menores e foi isso que viabilizou os telefones celulares. Além do mais, como os transmissores têm baixa potência, as transmissões da estação básica e dos telefones dentro da célula não ultrapassam os limites de transmissão. Assim as células não vizinhas podem usar um mesmo grupo de canais de rádio (frequência) e estas frequências podem ser reutilizadas de maneira intensa em toda a cidade.

Quando um usuário em comunicação se move de uma célula para outra, a estação básica (antena) da nova célula tem a responsabilidade de manter a comunicação deste usuário, para tanto ela deve alocar para ele um novo canal de rádio. A transferência de comunicação de uma célula para outra é chamada (em inglês) de *handoff*.

O mecanismo que envolve a transferência de um usuário entre duas estações básicas (células) ocorre quando o nível do sinal recebido pelo usuário fica muito fraco. Existem dois tipos de *handoff*. No exemplo da figura 1, quando o usuário se move da célula A para a B, o *handoff* é chamado de simples porque as duas células estão conectadas à mesma central telefônica de comutação, e o custo induzido para se fazer à transferência de A para B não é elevado. Por outro lado, quando o usuário se move da célula A para a célula C, o *handoff* é dito ser complexo. O custo induzido para se fazer esta transferência é alto, pois as duas centrais de comutação (1 e 2) devem estar ativas durante o processo de *handoff* e a base de dados contendo informações sobre os assinantes deve ser atualizada. Como o custo do *handoff* complexo é muito maior que o *handoff* simplex seria natural pensar que deveria se aumentar a área de abrangência das centrais telefônicas de maneira a diminuir a quantidade de *handoff* complexo. Isto, contudo, fere a idéia do conceito de telefonia celular (RODRIGUES, 2000) que consiste exatamente na substituição do transmissor único de alta potência que é responsável pela cobertura de uma grande área, como é feito na telefonia fixa, por vários transmissores de baixa potência, onde cada um é responsável pela cobertura de uma pequena área (célula) da área total permitindo, como já foi dito anteriormente, uma redução no consumo de energia, o que implica na utilização de baterias pequenas.

Neste artigo é apresentado um algoritmo do tipo *Beam Search*, para resolver o Problema de Atribuição de Células às Centrais de Telefonia Móvel. O *Beam Search* aqui apresentado é um algoritmo de enumeração implícita, que tem uma estrutura do tipo árvore de decisões.

Este artigo está organizado da seguinte forma: a seção 2 apresenta o problema de atribuição de células a centrais de telefonia celular, sua formulação matemática bem como alguns algoritmos da literatura utilizados na sua resolução; a seção 3 apresenta detalhadamente o *Beam Search* proposto; a seção 4 apresenta duas modificações realizadas no *Beam Search* com intuito de melhor o desempenho; a seção 5 apresenta os Testes Computacionais e Resultados Obtidos; e, por fim, a seção 6 expõe as Conclusões.

2. O PROBLEMA DE ATRIBUIÇÃO DE CÉLULAS A CENTRAIS

Uma das tarefas mais importantes no planejamento de redes de telefonia celular móvel é a atribuição de células a centrais telefônicas de comutação de modo ótimo, isto é, encontrar uma atribuição de células a centrais que respeite certas restrições e minimize o custo total de operação. Na minimização do custo total de operação, dois fatores devem ser considerados. O primeiro deles é o custo de cabeamento, isto é o custo da ligação entre os assinantes das células e a central de comutação a qual elas estão conectadas. Este custo depende da distância entre a célula e sua central. O segundo fator é o custo gerado pelo *handoff* entre células.

Neste trabalho, entende-se por atribuição de células às centrais, a determinação de qual central de comutação será a responsável pela comunicação dos assinantes de uma célula com as demais células do sistema de telefonia móvel. Como esta comunicação utiliza a antena existente em cada célula, muitas vezes fala-se em atribuição de antenas às centrais de comutação no lugar de se falar em atribuição de células às centrais.

O problema de se atribuir n células a m centrais numa rede de telefonia celular é um problema NP-completo (ABUAMARA *et al.*, 2006), logo ainda não existe um método exato capaz de achar a solução desse problema em tempo polinomial. Somente o método de busca exaustiva garantiria que a solução desse problema é ótima. Mas, para isso seria necessário examinar todo o espaço de soluções possíveis, o que é inviável, pois o número de soluções possíveis é enorme. Por

exemplo, para uma rede com 150 células e quatro centrais telefônicas, rede esta considerada de tamanho médio, seria necessário examinar (4150) soluções possíveis. Utilizando-se um computador capaz de examinar uma solução a cada 1ns, seriam necessários $6,46 \times 10^{73}$ anos. Devido a essa dificuldade, invariavelmente os métodos utilizados na resolução deste problema são as heurísticas. Na literatura existem diversos tipos de heurísticas, e as chamadas meta-heurísticas tem sido as mais utilizadas, veja, por exemplo, Abuamara *et al.* (2006); Dianati *et al.* (2003); Houeto e Pierre (2002); Quintero e Pierre (2003). Outros autores desenvolveram heurísticas do tipo Enumeração Implícita, entre eles: Merchant e Sengupta (1995); Saha *et al.* (2000). Menon e Gupta (2004) fizeram uma heurística mista que utiliza técnicas de Programação Linear e *Simulated Annealing*; e Salomão (2005) resolveu este problema utilizando Técnicas de Programação Linear com Geração de Colunas.

2.1. Formulação Matemática

Antes de apresentar a formulação matemática do problema, deve-se esclarecer que neste trabalho considerou-se que os custos de cabeamento e *handoff* serão oferecidos pela operadora e, portanto, eles entram como dados do problema. Uma descrição mais completa sobre o cálculo do *handoff* pode ser vista em Alonso *et al.* (1992).

A formulação do problema descrito como um problema de programação inteira quadrática é dada a seguir:

Sejam n células que devem ser atribuídas a m centrais de comutação. É assumido que a localização das células e das centrais é fixa e conhecida. Sejam H_{ij} e H'_{ij} respectivamente os custos por unidade de tempo dos *handoffs* simples e complexo que ocorrem entre as células i e j ($i, j = 1, \dots, n$). Como já foi dito acima, é assumido que os custos de *handoffs* são conhecidos e proporcionais à frequência de *handoffs* que ocorrem entre as células i e j . Seja C_{ik} o custo de cabeamento por unidade de tempo entre a célula i e a central de comutação k ($i = 1, \dots, n; k = 1, \dots, m$). Seja λ_i o número de chamadas que a célula i faz por unidade de tempo e seja M_k a capacidade da central k de efetuar chamadas por unidade de tempo. O objetivo deste problema consiste em atribuir cada célula a uma central de modo a minimizar o custo total por unidade de tempo. O custo total por unidade de tempo tem dois componentes: o custo do *handoff* por unidade de tempo e o custo de cabeamento entre as células e as centrais. A otimização deste problema deve ser feita de modo que a capacidade máxima de cada central não seja violada e que cada célula seja atribuída a uma única central.

A formulação matemática do problema descrito acima foi feita utilizando variáveis inteiras e com função objetivo não-linear. Para tanto foram definidas as seguintes variáveis:

$$x_{ik} = \begin{cases} 1 & \text{se a célula } i \text{ for tribuída a central } k \\ 0 & \text{caso contrário} \end{cases}$$

Considerando que cada célula só pode ser atribuída a uma única central, tem-se a seguinte restrição dada pela equação (1).

$$\sum_{k=1}^m x_{ik} = 1 \text{ para } i = 1, \dots, n \quad (1)$$

A restrição de capacidade das centrais de efetuarem chamadas é dada pela equação (2).

$$\sum_{i=1}^n \lambda_i x_{ik} \leq M_k \text{ para } k = 1, \dots, m \quad (2)$$

O custo total de cabeamento é dado por: $\sum_{i=1}^n \sum_{k=1}^m C_{ik} x_{ik}$.

O custo de *handoff* simples ocorre se duas células estão conectadas à mesma central e o *handoff* complexo ocorre se elas estiverem conectadas a centrais diferentes. Observe então que, ao se considerar todas as centrais, tem-se que: $\sum_{k=1}^m x_{ik} x_{jk} = 1$ se as células i e j são conectadas à mesma central de comutação e $\sum_{k=1}^m x_{ik} x_{jk} = 0$ se elas estão conectadas a centrais distintas.

Deste modo, os custos totais de *handoff* simples e complexo são dados respectivamente pelas equações (3) e (4).

$$\sum_{i=1}^n \sum_{j=1}^n H_{ij} \sum_{k=1}^m x_{ik} x_{jk} \quad (3)$$

$$\sum_{i=1}^n \sum_{j=1}^n H_{ij} \left(1 - \sum_{k=1}^m x_{ik} x_{jk}\right) \quad (4)$$

O custo total da rede é dado pela soma dos custos de cabeamento mais os custos de *handoff*, e pode ser escrito pela equação (5).

$$\sum_{i=1}^n \sum_{k=1}^m C_{ik} x_{ik} + \sum_{i=1}^n \sum_{j=1}^n H_{ij} \sum_{k=1}^m x_{ik} x_{jk} + \sum_{i=1}^n \sum_{j=1}^n H'_{ij} \left(1 - \sum_{k=1}^m x_{ik} x_{jk}\right) \quad (5)$$

Segundo Hedible e Pierre (2000), o custo do *handoff* simples H_{ij} pode ser considerado irrelevante quando comparado ao custo do *handoff* complexo H'_{ij} . Fazendo-se então: $h_{ij} = H'_{ij} - H_{ij}$, substituindo-a em (5) e desconsiderando a parte constante, a função objetivo a ser minimizada torna-se a apresentada pela equação (6).

$$\sum_{i=1}^n \sum_{k=1}^m C_{ik} x_{ik} + \sum_{i=1}^n \sum_{j=1}^n h_{ij} \left(1 - \sum_{k=1}^m x_{ik} x_{jk}\right) \quad (6)$$

Deste modo o problema a ser resolvido é o apresentado em (7). Observe que ele é um problema não linear com variáveis inteiras.

$$\begin{aligned} \text{Minimizar} \quad & \sum_{i=1}^n \sum_{k=1}^m C_{ik} x_{ik} + \sum_{i=1}^n \sum_{j=1}^n h_{ij} - \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^m h_{ij} x_{ik} x_{jk} \\ \text{sujeito a} \quad & \begin{cases} \sum_{k=1}^m x_{ik} = 1 \text{ para } i = 1, \dots, n \\ \sum_{i=1}^n \lambda_i x_{ik} \leq M_k \text{ para } k = 1, \dots, m \\ x_{ik} = 0 \text{ ou } 1 \text{ para } i = 1, \dots, n \text{ e } k = 1, \dots, m \end{cases} \end{aligned} \quad (7)$$

Em muitos artigos da literatura o problema (7) tem sido transformado em um problema linear com variáveis inteiras, através da substituição das equações $\sum_{k=1}^m x_{ik} x_{jk} = 1$ e $\sum_{k=1}^m x_{ik} x_{jk} = 0$ por um conjunto de restrições lineares. Veja por exemplo: Menon (2004), Merchand e Sengupta (1995), Salomão (2005). A desvantagem de se transformar o problema não linear em um problema linear é o aumento considerável do número de variáveis. Assim um problema que já é muito grande, torna-se maior ainda.

O algoritmo proposto neste trabalho utiliza a formulação não linear do problema.

3. O ALGORITMO DO TIPO *BEAM SEARCH*

O algoritmo do *Beam Search* é um método do tipo Enumeração Implícita para resolver problemas de Otimização Combinatória. Pode-se dizer que ele é uma adaptação do método de Branch and Bound onde somente os nós mais promissores de cada nível da árvore de decisões (atribuições) são guardados na memória para serem visitados, enquanto que os demais nós são descartados permanentemente. Como uma grande parte dos nós da árvore de atribuições é descartada, isto é, somente alguns poucos nós são selecionados para serem analisados, o tempo de execução do método é polinomial com relação ao tamanho do problema. Em resumo, pode-se dizer que o *Beam Search* é uma técnica de busca em árvore que em cada nível da árvore é analisado um número fixo de nós e, por conseguinte, um número fixo de soluções. O número de nós analisados em cada nível é chamado de largura da busca e é denotado por β .

O *Beam Search* foi usado pela primeira vez pela comunidade de Inteligência Artificial para tratar problemas de reconhecimento de fala (LOWERRE, 1976). A literatura fornece várias aplicações deste método em problemas de sequenciamento da produção (SABUNCUOGLU; BAYIZ, 1999; CROCE; T'KINDT, 2002; FOX, 1983; OW; MORTON, 1988; VALENTE; ALVES, 2005).

3.1. Árvore com todas as soluções possíveis

Antes de apresentar o algoritmo proposto será utilizado um exemplo para mostrar como seria a árvore de decisões (atribuições) se fossem enumeradas todas as soluções possíveis.

Seja $n = 4$, o número de células (antenas) que devem ser atribuídas a $m = 2$ centrais de comutação. Cada célula (antena) A_i tem capacidade de realizar um número fixo de chamadas λ_i por unidade de tempo. Assim, o número de chamadas cobertas pela antena A_1 é λ_1 , e λ_2 é o número de chamadas cobertas pela antena A_2 e assim sucessivamente. Essas chamadas serão realizadas pela central de comutação C_k , se a célula A_i estiver conectada à central C_j . Cada uma dessas centrais tem uma capacidade M_k , de efetuar chamadas. Então M_1 é a capacidade da central C_1 , M_2 é a capacidade da central C_2 e assim sucessivamente. Cada célula só pode estar conectada a uma única central. A variável é x_{ik} , onde $x_{ik} = 1$ significa que a célula i está conectada a central k e $x_{ik} = 0$ caso contrário. As Tabelas 1, 2, 3 e 4, mostradas a seguir, apresentam as informações sobre custo de cabeamento, custo de *handoff*, número de chamadas de cada antena e capacidade das centrais, para o exemplo apresentado.

Tabela 1 – Custo de Cabeamento C_{ik} entre as antenas e as centrais.

| Antenas | Central 1 | Central 2 |
|---------|-----------|-----------|
| 1 | 0 | 0 |
| 2 | 4 | 4 |
| 3 | 4 | 4 |
| 4 | 8 | 8 |

Fonte: Elaborado pelos autores.

Tabela 2 – Custo de *handoff* h_{ij} entre as antenas.

| Antena/ Antena | 1 | 2 | 3 | 4 |
|-------------------|---|----|---|----|
| 1 | 0 | 3 | 2 | 2 |
| 2 | 3 | 0 | 1 | 51 |
| 3 | 2 | 1 | 0 | 4 |
| 4 | 2 | 51 | 4 | 0 |

Fonte: Elaborado pelos autores.

Tabela 3 – Número de chamadas λ_i para cada antena i .

| Antenas | Nº chamadas |
|---------|-------------|
| 1 | 4 |
| 2 | 4 |
| 3 | 4 |
| 4 | 4 |

Fonte: Elaborado pelos autores.

Tabela 4 – Capacidade M_k das centrais.

| Central | Capacidade |
|---------|------------|
| 1 | 10 |
| 2 | 10 |

Fonte: Elaborado pelos autores.

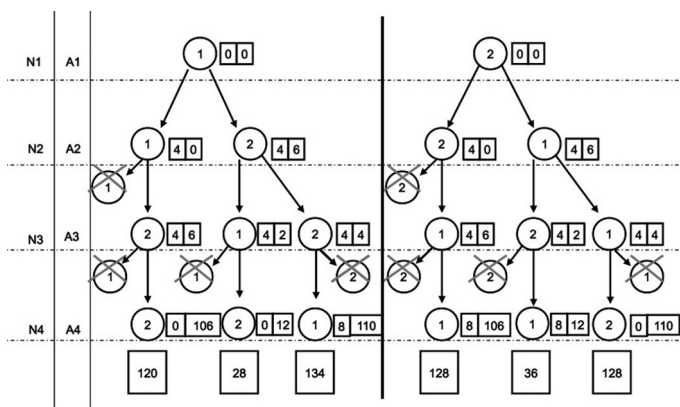
Para se construir a árvore de decisões, é necessário estabelecer as seguintes definições:

- (D.1) A árvore é construída por nível e em cada nível i é feita a atribuição da i -ésima Antena a cada uma das Centrais.
- (D.2) Os nós que estão no nível 1 da árvore são chamados de nós semente porque cada um deles vai gerar uma sub-árvore de decisões.
- (D.3) A cada nível i , ao se realizar a atribuição da i -ésima Antena a uma central k , são contabilizados dois custos: custo fixo de cabeamento e custo de *handoff* considerando todas as $(i-1)$ atribuições anteriores.

Tendo em vista que a árvore possui n níveis (D.1), onde n é o número de antenas, uma solução completa, com a atribuição de todas as n antenas à todas as m centrais só será obtida ao se definir as atribuições até o nível n .

A Figura 2 mostra como é a árvore com todas as soluções possíveis para esse problema. Os números das centrais estão representados dentro dos nós, o número da antena está representado ao lado pela letra A, então, A1 representa a antena 1, e os níveis estão representados pela letra N, por exemplo N1 representa o nível 1, N2 o nível 2 e assim para os demais níveis. Os nós de cada nível da árvore representam uma atribuição. Começa-se a árvore fixando uma central, por exemplo, a central C_1 e atribuindo a ela uma célula (antena), por exemplo, A1. Assim, o nó 1, do nível N1, significa que a célula A1 foi atribuída a central C_1 , isto é, $x_{11} = 1$. O nó desta atribuição inicial é chamado de nó semente. Uma vez feita a atribuição no nó semente segue-se fazendo as demais atribuições, sempre levando em conta as atribuições que já foram feitas nos nós anteriores. Note que o número de nós sementes é igual ao número de centrais m . Logo cada nó semente dará origem a uma sub-árvore. A árvore completa terá então m sub-árvores sendo uma para cada central.

Figura 2 – Árvore de soluções factíveis considerando o nó semente C_1 no nível N1 (sub-árvore da esquerda) e C_2 no nível N1 (sub-árvore da direita).



Fonte: Elaborado pelos autores.

Na Figura 2, os nós marcados com uma cruz fornecem uma atribuição inviável e por esta razão eles são automaticamente eliminados do processo. É importante frisar que o número total de soluções é $m^n = 2^4 = 16$, mas dentre estas apenas 6 são viáveis, justificando a aplicação de um pro-

cedimento de busca em árvore. Como em cada nível da árvore é feita a atribuição de uma antena a todas as centrais, assim no nível N1 é feita, por exemplo, a atribuição da antena A_1 a cada uma das centrais (nós sementes). No nível N2 é feita a atribuição da antena A_2 a cada uma das centrais; no nível N3 é feita a atribuição da antena 3 e no nível N4 é feita a atribuição da antena 4. Então, observe na Figura 2, que o nível N2 tem 4 nós decorrentes da atribuição da antena A_2 às centrais 1 e 2, na sub-árvore da esquerda e às centrais 2 e 1 na segunda sub-árvore da direita. No nível N3 têm-se seis nós, decorrentes da atribuição da antena A_3 respectivamente às centrais 2, 1 e 2 na sub-árvore da esquerda e as centrais 1, 2 e 1 na sub-árvore da direita. Ao lado de cada nó da Figura 2 também estão representados, dentro do pequeno retângulo, respectivamente os custos fixos de cabeamento e custos parciais de *handoff* relativa à atribuição efetuada no nó. Por exemplo, no nível N3, tem-se a atribuição da antena A_3 à central 2. Neste caso, o custo de cabeamento entre A_3 e C_2 é 4 e o custo parcial de *handoff* é 6 porque tem-se o *handoff* entre A_3A_1 , A_1A_3 , A_3A_2 e A_2A_3 .

Em cada nível são calculados somente os custos parciais da solução que está sendo obtida, pois como já foi dito a solução só será completa quando se gerar os nós do último nível da árvore. Deste modo, nos nós do nível N1 só se tem o custo de cabeamento (custo linear), no nível N2 já se têm os custos lineares mais os custos do *handoff* do nó semente até os nós do nível N2, e assim no último nível, tem-se custos lineares mais custos de *handoff* do nó semente até os nós do nível Nn. À medida que se vai descendo na árvore, o custo vai se completando pela decisão tomada. Assim, o ramo da árvore contendo os nós A_1C_1 , A_2C_1 , A_3C_2 , e A_4C_2 indica que foram feitas as seguintes atribuições: $x_{11} = 1$, $x_{21} = 1$, $x_{32} = 1$, $x_{42} = 1$.

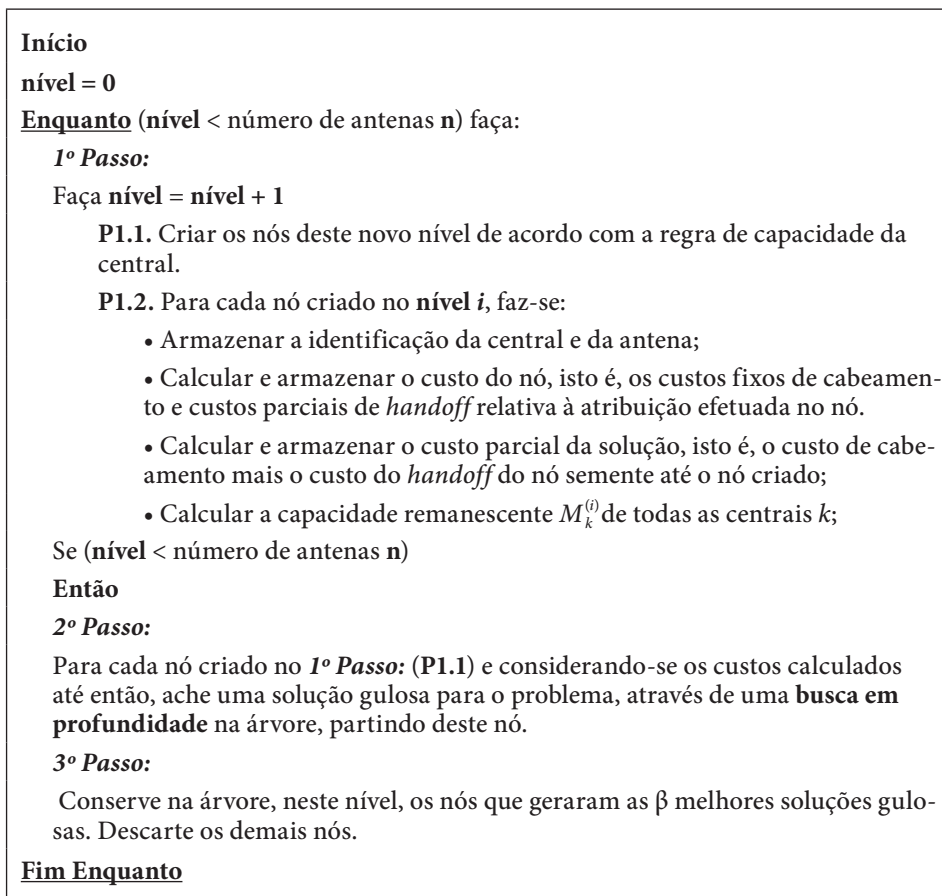
3.2. Árvore de soluções gerada pelo *Beam Search*

O algoritmo proposto não cria todos os nós da árvore como foi feito no exemplo mostrado e por esta razão não é feita uma enumeração completa de todas as soluções factíveis. Isso evita o crescimento exponencial da árvore. O algoritmo segue algumas regras para se criar os nós. Essas regras também visam impedir a criação de soluções inviáveis.

Em cada nó da árvore são armazenadas as seguintes informações: (1) Identificação da Central; (2) Identificação da Antena (célula); (3) Custos fixos de cabeamento e custos parciais de *handoff* relativos à atribuição efetuada no nó; (4) Capacidade remanescente da central que ele representa. A capacidade remanescente da central $M_k^{(i)}$ é calculada subtraindo-se da capacidade inicial da central M_k , a quantidade de chamadas λ_i referente às demandas das antenas que já foram alocadas a esta central. (5) Custo parcial da solução, isto é, custo de cabeamento mais *handoff* do nó semente até o nó em questão.

Durante o processo de construção da árvore de solução, o primeiro critério para se criar ou não um nó no nível i , é verificar a capacidade remanescente $M_k^{(i)}$ de uma central. Um nó só será criado (atribuição da antena A_i) no nível i se $\lambda_i \leq M_k^{(i)}$, isto é se o número de chamadas λ_i cobertas pela célula (antena) A_i for menor ou igual à capacidade remanescente $M_k^{(i)}$ da central C_k . Se $M_k^{(i)} < \lambda_i$, todos os ramos que seriam originados neste nó não vão existir, pois a célula i não poderá ser atribuída à central k . Depois de passar pelo critério da capacidade da central, o nó só permanecerá na árvore de solução se ele passar pelo critério da largura da busca β . Por exemplo, se a largura de busca escolhida pelo usuário for $\beta = 2$, a cada nível, só vão permanecer na árvore, os nós que gerarem as duas menores soluções, calculadas pela técnica do algoritmo guloso. Assim, para $\beta = 2$, de cada nó só sairão dois galhos, e serão aqueles que geraram as soluções gulosas de menor custo. O algoritmo utilizado para se construir a árvore é mostrado na Figura 3.

Figura 3 – Algoritmo para obtenção da árvore de decisões.



Fonte: Elaborado pelos autores.

Inicialmente, o algoritmo encontra-se no **nível zero** de solução, pois as atribuições começarão a serem feitas agora. A seguir faz-se **nível = 1** e começam a serem criados os nós (**P1.1**) deste nível. Os nós do **nível 1** são os nós sementes e é criado um nó semente para cada central. Em (**P1.2**), para cada nó criado em (**P1.1**) é atribuída uma célula (antena) do vetor de células levando-se em conta a capacidade da central. A seguir calculam-se os custos de cabeamento e *handoff* de cada um desses nós.

A busca em profundidade realizada no **2º Passo**, é feita utilizando um algoritmo do tipo guloso e ela visa escolher, no **3º Passo**, quais os nós, do **nível *i***, devem permanecer na árvore de soluções factíveis, de modo a respeitar a largura de busca β . Por exemplo, se $\beta = 2$, em cada nível vão permanecer somente dois nós. O processo de criação da solução gulosa é semelhante ao processo de criação da árvore de soluções apresentado na Figura 2, só que aqui a cada passo escolhe-se a atribuição que gera a melhor solução parcial do nó, isto é, vão fazer parte da solução os nós com menor custo. A busca em profundidade é apresentada na Figura 4.

Figura 4 – Busca em Profundidade do Passo2 - Algoritmo Guloso.

Para cada nó j criado no **1º passo (P1.1)** fazer uma arborescência em profundidade, a partir deste nó, descendo até o último nível, isto é:

- Seja um nó j criado no **nível i** com custo de solução parcial S_j , calculada até o **nível i** .
- Escolha para fazer parte desta solução o nó p do **nível $i+1$** que é viável e que gere a menor solução parcial S_p , onde $S_p = S_j + \text{custo fixo de cabeamento} + \text{custos parciais de handoff}$ relativa à atribuição efetuada no nó p .
- Repetir este processo a partir do nó p , descendo até o último nível. Somente após descer no último nó do ramo é que se terá o custo total de uma solução gulosa.

Fonte: Elaborado pelos autores.

Como já foi dito anteriormente, as soluções gulosas servem para escolher dentre os nós, do **nível i** que foram criados no **1º Passo**, aqueles que vão permanecer na árvore. Essa escolha é feita da seguinte maneira: Ordenar os nós criados no **nível i** , em ordem crescente, de acordo com os resultados dos custos das soluções gulosas que cada um deles gerou. Entre os nós do **nível i** escolher para permanecer na árvore os β nós que geraram as soluções gulosas de menor custo, Cortar os demais nós do nível i , isto é: Se largura de busca é $\beta=2$, só vão continuar a fazer parte da árvore os dois nós que geraram as duas soluções gulosas de menor custo. O valor da menor solução gulosa calculada no **2º Passo**, também é usado como limitante superior (corte) para se gerar ou não os outros nós da árvore, no 1º passo. Este limitante superior deve ser atualizado à medida que forem encontradas soluções gulosas menores. Com isto, têm-se dois critérios de corte para se criar ou não os nós da árvore. O primeiro deles é a capacidade das centrais e o segundo é o custo da menor solução gulosa, isto é, se o custo parcial da solução no nó que acabou de ser criado, for maior que o custo da melhor solução gulosa, este nó deve ser excluído da solução.

Depois de terminada a Busca em Profundidade deve-se verificar se o **nível** da árvore de decisões é igual ao número de antenas, Se este for o caso, o algoritmo termina, pois já se chegou ao final da árvore. Senão, o algoritmo volta ao **1º passo** para gerar os nós do próximo nível.

Figura 5 – Passos da execução do *Beam Search* para o Exemplo da seção 3.1.

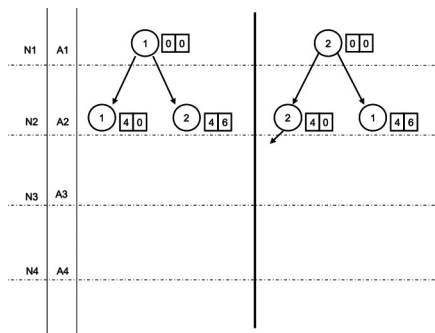


Figura 5 (a)

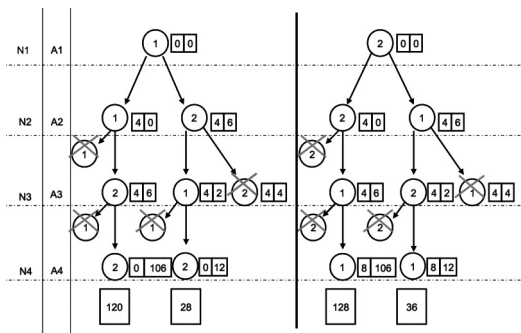


Figura 5 (b)

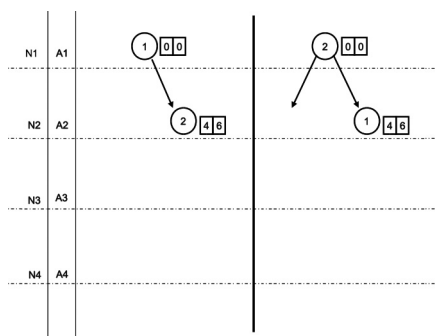


Figura 5 (c)

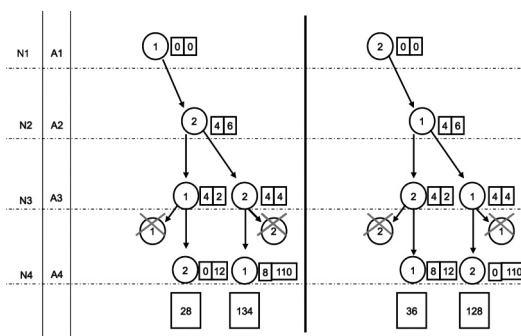


Figura 5 (d)

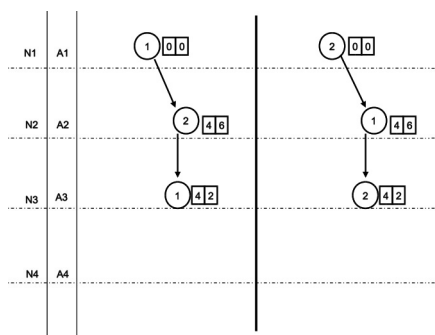


Figura 5 (e)

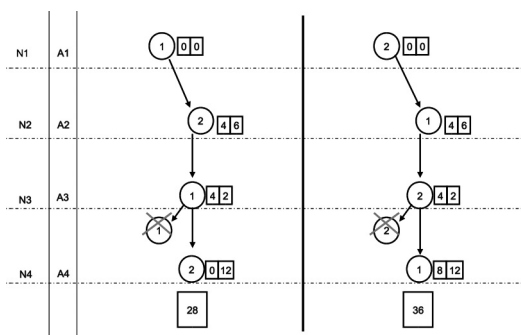


Figura 5 (f)

Fonte: Elaborado pelos autores.

As Figuras 5 (a)-(f), apresentam a resolução detalhada do Exemplo da Seção 3.1 pelo algoritmo do *Beam Search*. No exemplo é considerada largura de busca igual à $\beta = 2$. Os passos a serem seguidos para se resolver o exemplo são:

- (i) Construir os nós do nível N1 (um nó para cada central). Os custos de cabeamento e *handoff*, neste nível são iguais a zero (veja os custos nas Tabelas 1 e 2).
- (ii) No nível N2 inicialmente são criados 4 nós, pois a partir da atribuição feita no nível N1, existem 4 atribuições possíveis para a antena 2 a saber: A_2C_1 , A_2C_2 , na sub-árvore da esquerda e A_2C_2 , A_2C_1 na sub-árvore da direita. Esta situação está descrita na Figura 5(a) e corresponde ao 1º Passo do *Beam Search*. Como a largura da busca é dois, somente dois desses nós vão permanecer na árvore.
- (iii) Para escolher quais nós do nível N2 vão permanecer na árvore, é realizada uma busca em profundidade, até o último nível, para cada uma das 4 atribuições feita no nível N2.

Seja por exemplo o nó 2 do nível N2, da sub-árvore da esquerda, da Figura 5(b). As atribuições feitas até este nó são: A_1C_1 no nível N1 e A_2C_2 no nível N2 com custo de cabeamento de 4 e *handoff* parcial 6. Faltam as atribuições das antenas A3 no nível N3 e A4 no nível N4. A antena A3 pode ser atribuída à central C1 com custos de cabeamento e *handoff*, respectivamente de 4 e 2 (nó 1 do nível N3), e atribuída à central C2 com custos de cabeamento e *handoff*, respectivamente de 4 e 4 (nó 2 do nível N3). Como se trata de um algoritmo guloso, escolhe-se atribuir a antena A3 à central C1, pois o custo parcial desta solução é menor. Descarta-se então o nó 2 do nível N3. Continuando as atribuições a partir do nó 1 do nível N3, devido à restrição de capacidade, a antena A4 só pode ser atribuída à central C2 e o custo desta atribuição é de zero para cabeamento e 12 para o *handoff*. Aplicando-se este procedimento para todos os 4 nós do nível N2, serão encontradas 4 soluções gulosas, onde serão escolhidas as duas com menores custos. Tal situação corresponde ao 2º Passo do *Beam Search*.

A Figura 5(c) mostra que os nós escolhidos para ficar na árvore, nível N2, (3º Passo) foram: nó 2, na sub-árvore esquerda e nó 1 na sub-árvore direita. A seguir são criados o nós do nível N3 (1º Passo), e depois uma busca gulosa a partir desses nós (Figura 5(d)), para se escolher quais nós do nível N3 vão permanecer na árvore. A Figura 5(e) mostra os nós escolhidos do nível N3 (3º Passo). A Figura 5(f) mostra a arborescência (1º Passo) dos nós da Figura 5(e) e a nova solução gulosa (2º Passo). No final, obtêm-se duas soluções: A_1C_1 , A_2C_2 , A_3C_1 , A_4C_2 com custo de 28 e A_1C_2 , A_2C_1 , A_3C_2 , A_4C_1 com custo de 36. Adota-se então a solução de menor custo entre as duas.

4. MODIFICAÇÕES NO *BEAM SEARCH*

É possível melhorar o desempenho do *Beam Search* fazendo-se algumas modificações, por exemplo estabelecendo uma ordem para se fazer a atribuição das antenas, de acordo com um critério heurístico. O exemplo da seção 3.1 seguiu a ordem numérica das antenas. A seguir são apresentadas duas dessas modificações.

4.1. *Beam Search* com ordenação por *handoff* e custo de cabeamento (BSHC)

Aqui se escolheu como critério para fazer a atribuição das antenas, a ordem decrescente do custo estimado de *handoff* e de cabeamento, ou seja, para cada antena A_i foi calculado o custo estimado de *handoff* e cabeamento CEHC dado pela Eq. (8). A Tabela 5 mostra esses cálculos para o exemplo da Seção 3.1.

$$CEHC_i = \sum_{j=1}^n h_{ij} + \sum_{k=1}^m C_{ik} \quad (8)$$

Tabela 5 – Definição da ordem de alocação das antenas por meio do CEHC.

| Antena/ Antena | A1 | A2 | A3 | A4 |
|-------------------|----|----|----|----|
| CEHC | 7 | 63 | 15 | 73 |
| Ordem | 4 | 2 | 3 | 0 |

Fonte: Elaborado pelos autores.

4.2. *Beam Search* empregando sub-problemas (BSHCSP)

Uma outra modificação consiste em aplicar o algoritmo de *Beam Search* em cada sub-árvore separadamente. A vantagem disso é que, ao invés de se resolver um problema grande, resolve-se m subproblemas menores e com isso diminui-se o risco de se eliminar uma sub-árvore completa, quando se aplica a largura da busca b , eliminando assim a região onde se encontra a solução ótima. Veja, por exemplo, o caso quando se está resolvendo um problema com 4 centrais e deseja-se utilizar uma largura de busca $b = 2$. Quando se aplica um único *Beam Search*, para toda a árvore, no nível N1 permanecem somente dois nós sementes. Duas sub-árvores inteiras serão cortadas e com isso pode-se eliminar a vizinhança que contém a solução ótima. Quando se aplica um *Beam Search* em cada sub-árvore, isto não ocorre porque a largura de busca $b = 2$ vai ser aplicada por sub-árvore. Pode-se dizer que este método preserva melhor a noção de busca na vizinhança. Mas a grande vantagem deste método é que ele pode ser executado em paralelo. Na implementação em paralelo, cada processador irá executar um *Beam Search* em uma sub-árvore. Deste modo, se o problema tem quatro centrais, ter-se-ia quatro processadores e cada um desses processadores iria executar um *Beam Search* em uma sub-árvore. Esta implementação permite obter uma grande redução do tempo computacional, pois em cada sub-árvore tem-se um problema de busca totalmente independente, o que dispensa a comunicação entre os processadores proporcionando um *speed-up* (SCHENDEL, 1984) e uma eficiência elevada.

5. TESTES COMPUTACIONAIS

A seguir são apresentados os testes computacionais realizados com o algoritmo do *Beam Search* empregando o custo de cabeamento e *handoff* para ordenar a atribuição das antenas e utilizando duas versões do algoritmo, a saber: problema completo e sub-problemas. Também foi rodada uma versão do *Beam Search* (BS) aplicado ao problema completo e que faz a atribuição das antenas utilizando o critério numérico, tal como no exemplo da seção 3.1. Foram rodadas 47 instâncias com dimensões variadas. Começou-se com problemas de tamanho considerado pequeno (15 antenas e 2 centrais), até problemas de tamanho considerado grande (200 antenas e 4 centrais). As instâncias são as mesmas de Merchant e Sengupta (1995). Os algoritmos foram implementados em Matlab 7.0, numa máquina com processador Intel 2.0 GHz, 1G de RAM.

A Tabela 6 mostra, para cada uma das instâncias rodadas os valores de função objetivo obtidos com as três versões do *Beam Search*: o BS, BSHC e o BSHCSP. Na primeira coluna está o número da instância, a seguir o número de células e centrais da instância e depois, a Função Objetivo. Em todas as versões a largura de busca é igual ao número de centrais.

Tabela 6 – Resultados obtidos com os diferentes métodos de *Beam Search*.

| Instância | Células/ Centrais | F.O BSHC | F.O BSHC SP | F.O BS | Instância | Células/ Centrais | F.O BSHC | F.O BSHC SP | F.O BS |
|-----------|----------------------|-------------|----------------|-----------|-----------|----------------------|-------------|----------------|-----------|
| 1 | 15/2 | 130.91 | 130.91 | 134.61 | 25 | 125/2 | 2914.48 | 2914.48 | 2914.48 |
| 2 | 15/3 | 124.62 | 124.62 | 131.90 | 26 | 125/3 | 2217.93 | 2217.93 | 2213.78 |
| 3 | 15/4 | 101.23 | 101.23 | 100.83 | 27 | 125/4 | 2965.73 | 2965.73 | 2959.14 |
| 4 | 15/5 | 105.77 | 105.77 | 106.44 | 28 | 125/5 | 1943.16 | 1943.15 | 2041.405 |
| 5 | 30/2 | 341.22 | 341.22 | 328.28 | 29 | 150/2 | 4007.00 | 4007.00 | 4007.00 |
| 6 | 30/3 | 348.34 | 345.53 | 345.53 | 30 | 150/3 | 3393.07 | 3393.07 | 3247.26 |
| 7 | 30/4 | 285.99 | 285.99 | 285.99 | 31 | 150/4 | 2787.55 | 2787.55 | 2744.26 |
| 8 | 30/5 | 250.03 | 250.03 | 246.75 | 32 | 150/5 | 3122.70 | 3122.70 | 3177.71 |
| 9 | 45/2 | 975.89 | 975.88 | 960.33 | 33 | 175/2 | 5122.59 | 5122.59 | 5122.59 |
| 10 | 45/3 | 555.80 | 555.80 | 555.80 | 34 | 175/3 | 5566.67 | 5564.08 | 4994.93 |
| 11 | 45/4 | 524.85 | 524.85 | 529.01 | 35 | 175/4 | 3229.75 | 3229.74 | 3365.60 |
| 12 | 45/5 | 441.95 | 441.95 | 427.95 | 36 | 175/5 | 2523.86 | 2523.86 | 2532.85 |
| 13 | 60/2 | 921.91 | 921.91 | 921.91 | 37 | 200/2 | 5782.84 | 5782.84 | 5782.84 |
| 14 | 60/3 | 711.88 | 711.88 | 767.25 | 38 | 200/3 | 4371.26 | 4371.26 | 4371.26 |
| 15 | 60/4 | 737.73 | 737.73 | 739.89 | 39 | 200/4 | 4393.75 | 4393.75 | 4393.75 |
| 16 | 60/5 | 725.41 | 725.41 | 696.06 | 40 | 200/5 | 3768.43 | 3768.43 | 3792.84 |
| 17 | 75/2 | 1537.67 | 1537.67 | 1537.67 | 41 | 225/2 | 6771.95 | 6771.95 | 6892.52 |
| 18 | 75/3 | 995.41 | 995.41 | 995.41 | 42 | 225/3 | 4960.16 | 4960.16 | 4956.89 |
| 19 | 75/4 | 1427.84 | 1427.84 | 1444.14 | 43 | 225/4 | 5384.48 | 5384.48 | 5375.05 |
| 20 | 75/5 | 967.10 | 967.09 | 973.093 | 44 | 225/5 | 5630.72 | 5624.69 | 5398.15 |
| 21 | 100/2 | 1736.03 | 1736.03 | 1736.03 | 45 | 250/2 | 8075.77 | 8075.77 | 8075.77 |
| 22 | 100/3 | 1383.43 | 1383.43 | 1401.56 | 46 | 250/3 | 8624.23 | 8624.23 | 8714.41 |
| 23 | 100/4 | 1441.01 | 1441.01 | 1448.91 | 47 | 250/4 | 6101.35 | 6088.19 | 6199.61 |
| 24 | 100/5 | 1493.69 | 1489.90 | 1466.66 | | | | | |

Fonte: Dados da pesquisa.

Pode-se ver na Tabela 6 que as heurísticas que utilizaram o critério de ordenação em função do custo forneceram melhores resultados que o BS simples. Na comparação entre o BS simples com o BSHC e o BSHCSP, o BS simples apresentou melhor resultado em 14 casos; em 17 casos as heurísticas BSHC e o BSHCSP apresentaram melhores resultados; e nos demais casos os resultados foram iguais. Na comparação entre as heurísticas com critério de ordenação por custo, o BSHCSP apresentou melhor resultado que o BSHC em 4 casos. Nos demais, eles foram iguais.

6. CONCLUSÃO

As heurísticas do tipo *Beam Search* não necessitam de uma solução inicial e sempre encontram uma solução viável. Uma das vantagens dessas heurísticas é que elas não geram todos os ramos da árvore, seu tempo de execução é, portanto, polinomial. A busca em profundidade permite que se encontrem boas soluções, pois ela considera o custo da atribuição e a capacidade das centrais. Os resultados obtidos com as três heurísticas mostraram que elas são relativamente equivalentes, com um ligeiro ganho para as que utilizaram a ordem de atribuição considerando o custo de *handoff* e cabeamento. O *Beam Search* que considera sub-problemas (BSHCSP) pode ser facilmente implementado em paralelo. Para tanto, deve-se colocar um processador para trabalhar em cada sub-árvore. Cada processador vai então buscar a melhor solução de cada uma dessas sub-árvores.

A eficiência de um algoritmo paralelo depende de uma distribuição equilibrada das tarefas que são executadas em cada processador e do volume de comunicações efetuada entre os processadores. No caso do BSHCSP, as sub-árvores possuem o mesmo tamanho e são totalmente independentes, com isso as tarefas a serem executadas pelos processadores estão equilibradas e não necessitam de comunicação entre eles, então é possível obter uma drástica redução do tempo computacional, redução esta da ordem de até 80% do tempo despendido no algoritmo sequencial, porque, por exemplo, no caso de cinco processadores o tempo de execução em paralelo seria aproximadamente o tempo de execução sequencial dividido pelo número de processadores.

Uma característica do *Beam Search* é que durante o processo de busca existe a garantia de que a mesma solução não será avaliada duas vezes, isto porque não ocorrem duplicações de soluções. Isto é, ao se utilizar, em todas as sub-árvores, sempre a mesma ordem de atribuição das antenas.

REFERÊNCIAS BIBLIOGRÁFICAS

ALONSO, E.; MEIER-HELLSTERN, S.; POLLINI, G. P. Influence of Cell Geometry on Handover and Registration Rates in Cellular and Universal Personal Telecommunications Networks, **8th ITC Specialist Seminar on universal Communication**, Genova, Italy, 1992.

ABUAMARA, M. H.; SAIT, S.; SUBHAN, A. A Heuristics Based Approach for Cellular Mobile Network Planning. **The International Wireless Communications and Mobile Computing Conference (IWCMC06)**, July, Vancouver, Canada. 2006

DELLA CROCE, F.; T'KINDT, V. A Recovering Beam Search Algorithm for the One-Machine Dynamic Total Completion Time Scheduling Problem, **Journal of the Operational Research Society**, vol 54, pp. 1275-1280, 2002.

DIANATI, M.; NAIK, S.; SHEN, X.; KARRAY, F. A Genetic Algorithm Approach for Cell to Switch Assignment in Cellular Mobile Networks. **Proc. 2003 Canadian Workshop on Infor. Theory**, pp. 159-162, Waterloo, Ontario, Canada, May 18-21.

FOX, M. S. **Constraint-Directed Search: A case Study of Job-Shop Scheduling**, PhD. thesis, Carnegie-Mellon University, USA, 1983.

HEDIBLE, C.; PIERRE, S. Genetic algorithm for the assignment of cells to switches in personal communication networks, Electrical and Computer Engineering, **Canadian Conf. on**, vol. 2, pp. 1077 -1081, 2000.

- HOUETO, F.; PIERRE, S. Assigning cells to switches in cellular mobile networks using taboo search, **IEEE Transactions on Systems, Man and Cybernetics, Part B**, vol. 32 n. 3, pp. 351–356, 2002.
- LOWERRE, B. T. **The HARPY Speech Recognition System**, PhD. thesis, Carnegie-Mellon University, USA, 1976.
- MENON, S.; GUPTA, R. Assigning cells to switches in cellular networks by incorporating a pricing mechanism into simulated annealing, **IEEE Transactions on Systems, Man, and Cybernetics, Part B**, vol. 34, n. 1, p. 558-565, 2004.
- MERCHANT, A.; SENGUPTA, B. Assignment of cells to switches in PCS networks, **IEEE/ACM Trans. Networking**, vol. 3, pp. 521-526, Oct, 1995.
- OW, P. S.; MORTON, T. E. Filtered Beam Search in Scheduling, **International Journal of Production Research**, vol. 26, pp. 35-62, 1988.
- QUINTERO, A.; PIERRE, S. Assigning cells to switches in cellular mobile networks: a comparative study. **Computer Communications**, v. 26, n. 9, p. 950-960, 2003.
- RODRIGUES, M. E. C. **Técnicas de Traçado de Raios em Três Dimensões para Cálculo de Campos em Ambientes Interiores e Exteriores**. Dissertação de Mestrado apresentada ao Departamento de Engenharia Elétrica da PUC-Rio - Abril de 2000.
- SABUNCUOGLU, I.; BAYIZ, M. Job Shop Scheduling with Beam Search, **European Journal of Operational Research**, vol. 118, pp. 390-412, 1999.
- SAHA, D.; MUKHERJEE, A.; BATTARCHARYA, P. S. A simple heuristic for assignment of cells to switches in a PCS network, **Wireless Personal Communications** 12 (3) 209-224, 2000.
- SALOMÃO, N. A. **Métodos de Geração de Colunas para Problemas de Atribuição**, Tese de Doutorado em Computação Aplicada, INPE, São José dos Campos, SP, 2005.
- SCHENDEL, V. **Introduction to Numerical Methods for Parallel Computers**, Ellis Horwood Series, John Wiley and Sons N. Y., 1984.
- VALENTE, J. M. S.; ALVES, R. A. F. S. Filtered and Recovering Beam Search Algorithm for the Early/Tardy Scheduling Problem with No Idle Time, **Computers & Industrial Engineering**, vol. 48, pp. 363-375, 2005.

