

5 Resultados numéricos

O resultados apresentados nesta seção devem ser considerados como ilustrativos das idéias, conceitos, resultados e teoremas apresentados nas seções anteriores. Sua finalidade é provar que os conceitos funcionam, pelo menos para simulações básicas. Comparações quantitativas precisas e formais, que provem ser os métodos apresentados neste trabalho melhores ou não do que alguns outros existentes na literatura correntes, não foram realizadas. Isto porque, em primeiro lugar a modelagem matemática foi feita sobre um sistema de características completamente gerais, sem nenhuma das particularidades que os sistemas reais em geral apresentam; em segundo lugar porque não foram sugeridas aplicações específicas para as formulações deste trabalho. Por exemplo, o teorema 2 tanto poderia ser efetivamente utilizado em algum algoritmo de alocação de canais, como também poderia ser aplicado como critério de qualidade na comparação entre métodos aqui apresentados e outras heurísticas. Em terceiro, deve-se levar em conta que os programas efetuados no curso desta tese concentraram-se na implementação dos métodos aqui discutidos sem que houvesse exagerada preocupação com a velocidade de execução. Em sua maioria foram implementados no ambiente MATLAB, que na maioria dos casos é ineficiente sob o ponto de vista de velocidade de execução de códigos.

Assim as metodologias propostas serão ilustradas e alguns resultados relativos aos seus desempenhos computacionais serão apresentados.

5.1. Cenário

O cenário relativo ao ambiente onde os enlaces se localizam, foi escolhido com o objetivo de ilustrar os métodos, algoritmos e teoremas discutidos em seções anteriores. É composto por uma região plana e quadrada de dimensão pré-especificada onde não se considera as presenças de obstáculos ou espalhadores, assim como também se considera que não há interferência proveniente de fora desta região. Este cenário é totalmente atípico, mas a única utilidade dele é gerar uma matriz de \mathbf{G} de ganhos. Poderia ter sido usado um gerador de números aleatórios para atribuir valores aos seus elementos, mas o método utilizado ao menos garante que haverá um comportamento de atenuação de sinal com a distância.

Nela, N_u unidades transmissoras e N_r unidades receptoras são previamente estabelecidas e aleatoriamente posicionadas, sendo este processo caracterizado pela escolha de uma semente para o gerador de números aleatórios, de modo a permitir a reprodutibilidade do cenário entre os vários programas que serão utilizados. Cada unidade transmissora estabelece um enlace unidirecional com a unidade receptora geograficamente mais próxima, sendo este modelo semelhante ao do enlace de subida dos sistemas de telefonia celular. Apesar disso este modelo é facilmente generalizável podendo também simular o comportamento de redes MANET. Em qualquer uma das situações discutidas, o nível de interferência é investigado apenas nas unidades receptoras.

Assume-se que cada unidade transmissora gera um campo eletromagnético de potência isotrópico, que será considerado como sinal para a unidade receptora a que se destina, e será considerada como interferência para as outras unidades receptoras. Assume-se que a intensidade do campo eletromagnético decaia segundo uma lei exponencial com a distância, embora outros modelos de propagação possam ser testados. Esta escolha não influirá no que se deseja mostrar. Para manter a coerência com a realidade se espera um comportamento de atenuação do sinal com a distância, mas este varia de acordo com as

características do local, por exemplo uma região bastante construída têm um comportamento, já uma área rural apresenta outro comportamento e assim por diante. Como, conforme foi dito antes, o único interesse é gerar uma matriz \mathbf{G} que seja semelhante à obtida em um caso real, os detalhes de modelo de propagação e tipo de terreno não são importantes.

Os valores máximos e mínimos de potência transmitida, a potência média de ruído, e o valor mínimo da SINR permitido também são configurados no cenário. Esse conjunto de parâmetros caracteriza o cenário para todos os programas de computador implementados. A tabela abaixo exemplifica um cenário específico:

Parâmetro	Valor	Significado
LL	10	dimensão da região de interesse (km)
Nu	20	número de usuários
Ne	8	número de erbs
Pn	1.00E-05	potência media de ruído (watt)
SIRdb	25	SNIR mínima (dB)
Pmin	0.1	potencia mínima (watt)
Pmax	1	potencia máxima (watt)
Semente	144	semente para o cenário

Tabela 4- Parâmetros de configuração de cenário

A partir da definição de um cenário específico (como o da tabela acima), elaborou-se um programa em Matlab, que posiciona os transmissores e receptores, e calcula a matriz \mathbf{G} de ganhos dos enlaces desejados e interferentes. Um exemplo do cenário formado pelos parâmetros apresentados na tabela acima é ilustrado na figura abaixo.

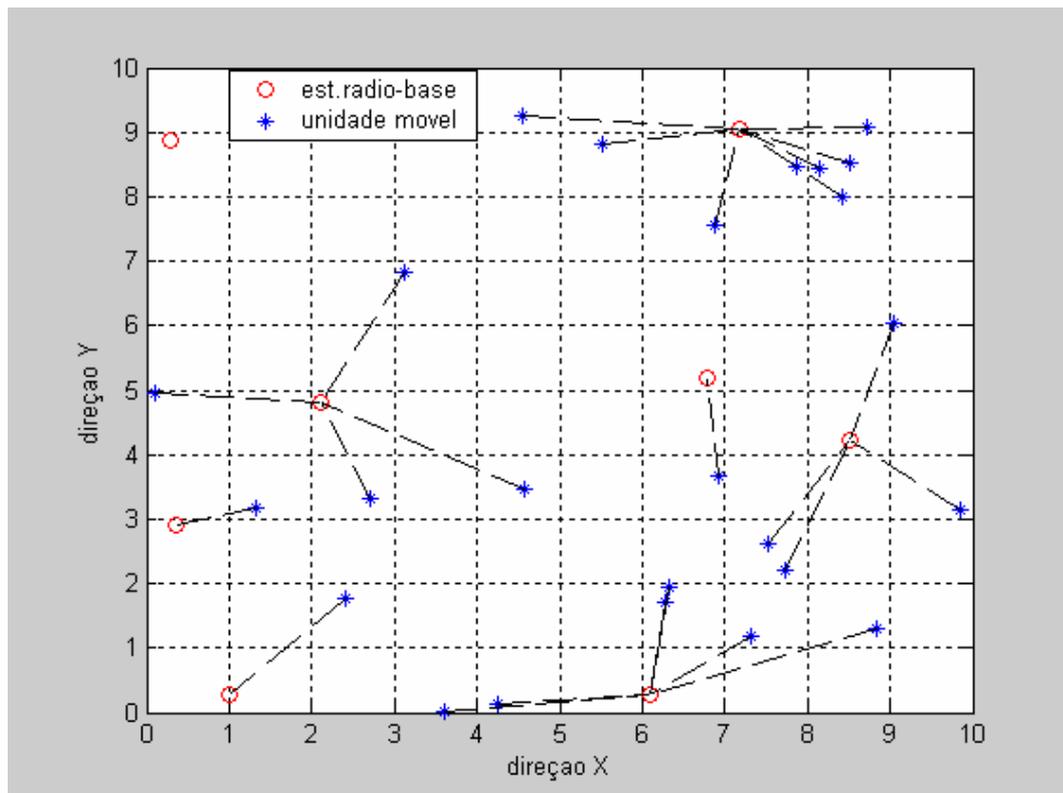


Figura 5-1 Posicionamento de transmissores e receptores no cenário

5.2. Determinação dos conjuntos de enlaces viáveis

Com base na matriz \mathbf{G} de ganho dos enlaces, um outro programa também implementado em Matlab, determina todos os subconjuntos de enlaces e-viáveis do cenário em consideração. Este programa utiliza os procedimentos descritos no algoritmo apresentado na seção 4.2. Primeiro os subconjuntos f-viáveis são calculadas por meio do procedimento de determinantes definidos pelo Teorema 2. A seguir, a e-viabilidade destes subconjuntos é testada por meio de um par de problemas de otimização, um linear e outro de natureza minimax, conforme descrito nas seções 4.3 e 4.4.2. Estes dois problemas foram respectivamente resolvidos pelos procedimentos *linprog* e *fminimax* existentes no pacote de otimização (Optimization Toolbox) do MATLAB. Estes programas funcionaram de forma eficiente em termos de convergência. O quadro abaixo ilustra uma saída

típica deste programa para o cenário já mencionado e apresentado em tabela anterior

```

-----VIABILIDADE DA MATRIZ G -----
  No. de usuarios      = 20
  No. de erbs         = 8
  Pot. de Ruído       = 1.000e-002 mW
  SIR Mínima          = 25.0 dB
  Potencia             = [1.000e-001,1.000e+000] W
--> 2-uplas achadas = 55 (190/89)
--> 3-uplas achadas = 67 (122/93)
--> 4-uplas achadas = 12 (42/37)
--> 5-uplas achadas = 0 (0/0)

--> TOTAL = 134
Duração = 5.6 seg

```

Tabela 5- Resultado do programa de verificação de e-viabilidade

Neste exemplo foram determinados 354 subconjuntos f-viáveis (a soma dos primeiros números entre parênteses), sendo destes, 190 pares, 122 triplos, e 42 quádruplos. Dentre desses casos, 134 (36,8% do total de f-viáveis) foram considerados e-viáveis, implicando que além de terem solução possível com potências positivas, todos possuem potências dentro dos limites pré-estabelecidos. Em 219 dos casos, (a soma dos segundos números entre parênteses), o ponto extremo da região de f-viabilidade não se localizava na região de potências permitidas, requerendo a execução dos procedimentos de otimização discutidos em parágrafo anterior. Podemos concluir que foram executadas 219 vezes o par de subrotinas linprog e fminimax de otimização, há um tempo máximo de 25.57 mseg por par. Este tempo é alto para fins de aplicações em tempo real, mas é possível que com melhorias no código, e processadores mais rápidos, esta limitação seja superada. Este tempo é satisfatório para aplicações que não demandem tempo real, como simulações de sistemas.

Para ilustrar a variação da quantidade de subconjuntos e-viáveis existentes em relação a mudanças dos parâmetros do cenário, usou-se o cenário apresentado na tabela anterior como referência e escolheram-se alguns parâmetros para alterações. Em relação ao cenário-referência, alterou-se:

- caso 1 : a variação da semente
- caso 2 : a SNIR variando de 30db a 20dB em passo de 1 dB
- caso 3 : a potência média de ruído variando entre $10^{-4}W$ a $10^{-8}W$
- caso 4: a potência máxima do transmissor de 0.3W para 10000W

No caso 1 ocorre uma grande variação no número de subconjuntos f-viáveis com a mudança da semente. A tabela 6 apresenta alguns valores de semente.

SEMENTE	NÚMERO DE CONJUNTOS E-VIÁVEIS
144	134
12	282
18	170
45	202
94	86
95	83
112	26

Tabela 6 – Variação do número de conjuntos e-viáveis com variação da semente

O que se vê é uma enorme variação, da ordem de dez vezes. Isto ocorre porque a mudança da semente implica uma nova disposição aleatória das estações transmissoras e receptoras, alterando fortemente as possibilidades de interferências.

O caso 2, com a variação do requisito de SINR, varia-se também intensamente o número possível de conjuntos e-viáveis. Ao se relaxar a exigência

da relação sinal ruído um maior número de usuários podem ocupar o mesmo canal. A tabela 7 mostra essa variação para o caso exemplo.

SINR	NÚMERO DE CONJUNTOS E-VIÁVEIS
30	3
29	27
28	55
27	72
26	72
25	134
24	164
23	313
22	385
21	460
20	575

Tabela 7 – Variação do número de conjuntos e-viáveis com a SINR

Também no caso 3 ocorre uma grande variação do número de conjuntos e-viáveis. Na tabela 8 observa-se uma saturação a partir de 0.05 mW indicando um ruído de fundo muito alto, que na prática não permite que nenhum enlace satisfaça os requisitos de SINR. Já em potências muito baixas de ruído, a saturação é devida a este ficar desprezível comparada à interferência dos outros usuários.

POTÊNCIA MÉDIA DE RUÍDO (MW)	NÚMERO DE CONJUNTOS E-VIÁVEIS
0.1	1
0.05	1
0.03	16
0.02	61

0.01	134
0.008	152
0.006	271
0.004	332
0.001	596
0.0001	625
0.00001	638

Tabela 8 - Variação do número de conjuntos e-viáveis com a potência de ruído

Por fim o quarto caso analisado serve para mostrar o espalhamento dos conjuntos f-viáveis ao longo dos níveis de potência. Isto porque o que difere o conjunto f-viável do e-viável, é o segundo estar dentro dos limites estabelecidos (máximo e mínimo) de potência. Nota-se uma concentração dos conjuntos em níveis inferiores 5 Watts, mas também ocorre um considerável número de enlaces e-viáveis espalhados em faixas de potência de até 10 MW, o que é totalmente inviável para casos reais. A figura 5-2 mostra esse espalhamento.

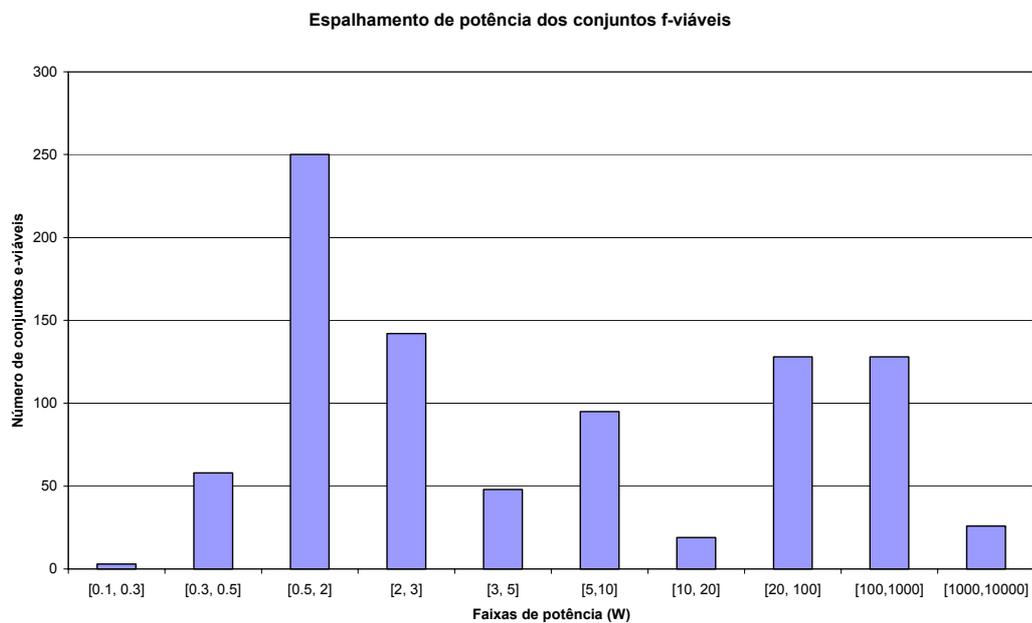


Figura 5-2 – Espalhamento de potência dos conjuntos f-viáveis

Estes exemplos de nenhuma forma exaurem o estudo do impacto destes parâmetros na quantidade de subconjuntos e-viáveis, mas permite ilustrar que variações em valores de alguns parâmetros podem produzir aumento substancial na quantidade de subconjuntos e-viáveis. Isto gera um impacto na dimensão dos problemas de otimização que se seguirão.

Convém salientar que este programa também gera em sua saída, matrizes de restrições e vetores de coeficientes para os problemas denominados de “mínimo número de canais” e “mínima potência máxima” que foram conceitualmente abordados em capítulos anteriores e que serão numericamente ilustrados nas seções seguintes.

5.3. Minimização do número de canais, ou da máxima potência

Os problemas de otimização que determinam respectivamente o número mínimo de canais (seção 4.3) e a mínima potência máxima (4.4.2) são de natureza binária e o MATLAB não dispõe de ferramentas para resolvê-los. Foram utilizadas duas ferramentas de programação para tratar tais problemas. A primeira foi o “solver” denominado de Visual Xpress³ específico para programação linear, que tem capacidade de resolver problemas contínuos, discretos e binários. O segundo foi uma ferramenta de domínio público GLPK desenvolvida dentro do projeto GNU (ferramentas de software livre, com fonte aberto, e sujeito a licenciamento GPL, o Linux faz parte deste projeto).

³ Visual Xpress é um produto da empresa Dash Associates.

Ambas ferramentas resolvem os problemas propostos de maneira eficiente, mas o Visual Xpress apresentou uma limitação de leitura dos arquivos de restrições gerados pelo programa feasible visto na seção anterior. Com isto não se podem tratar problemas com grande número de casos e-viáveis. Outra desvantagem deste pacote é que é um produto comercial, necessitando a aquisição de uma licença de uso. Assim sendo as simulações executadas para ilustrar esta tese foram feitas usando o pacote GLPK.

Desenvolver usando o GLPK é razoavelmente simples. A programação é feita na linguagem C, e pode ser compilada em qualquer compilador. Foram escritos dois programas, um para resolver o problema do número mínimo de canais, cuja formulação se encontra em 4.4.1, e outro que determina a configuração de subconjuntos e-viáveis que produzem a mínima potência máxima, e que foi formulado em 4.4.2. Convém salientar que com relação a este último problema de otimização foi artificialmente introduzida uma restrição adicional que define o número de canais utilizados. O objetivo desta restrição é avaliar o valor desta mínima potência máxima em função da quantidade de canais disponíveis.

O tempo de execução do primeiro programa é mínimo, só como exemplo, num caso com 7832 conjuntos e-viáveis o programa feasible demorou 967 segundos numa cpu Pentium 2.4 GHz, já o programa de minimização de canais executou quase instantaneamente. Assim o limitante para o estudo casos com grande quantidade de conjuntos e-viáveis é a rotina feasible.

Ainda utilizando o exemplo ilustrado na tabela no início deste capítulo onde o número de usuários é 20, a solução ótima encontrada para o número mínimo de canais foi 12, onde 7 canais apresentaram um único usuário, 3 canais apresentaram dois usuários, e 2 canais apresentaram três usuários, gerando um aproveitamento médio de 1.666 usuários por canal. Executando vários casos onde se aumenta o número de transmissores e de receptores, são encontrados valores entre 1.5 e 2.5. Quando se relaxa o parâmetro de SINR para 20 dB o aproveitamento médio no exemplo padrão sobe para 2.5, o mesmo acontece quando se diminui a intensidade do ruído de fundo para 0.005 mW.

Quanto ao programa de solução de mínima potência máxima, os tempos de execução são muito altos, da ordem de vários minutos para casos com muitos conjuntos e-viáveis. Um resultado interessante que pode ser obtido é como o valor de mínima potência máxima se altera quando se relaxa a condição de mínimos de canais. O exemplo abaixo demonstra isso para o caso exemplo escolhido.

No. De Canais	Min. Pot. Max. (W)	Redução Percentual
12	0.856	0.00
13	0.804	6.07
14	0.541	36.80
15	0.467	45.44
16	0.391	54.32
17	0.321	62.5
18	0.221	74.18
19	0.1	88.32

Tabela 9 – Efeito do relaxamento da condição de número de canais

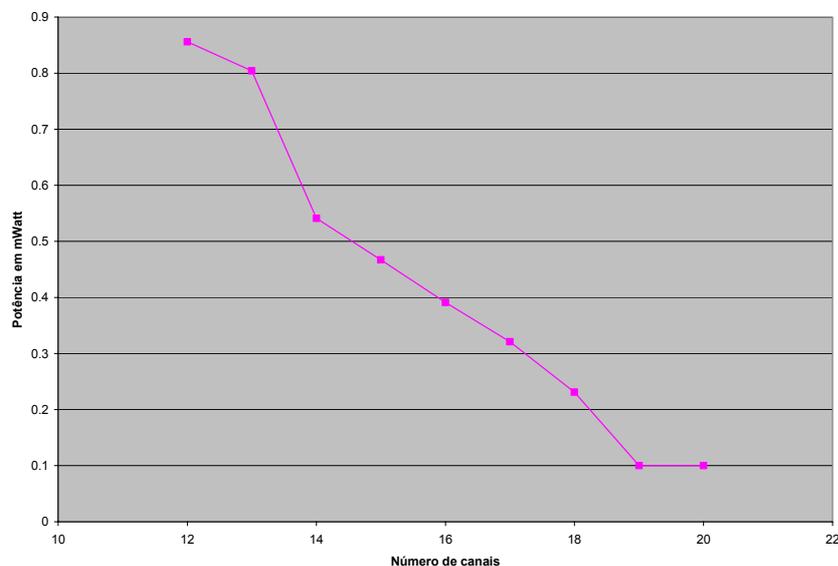


Figura 5-3 – Efeito do relaxamento da condição de número de canais

Note-se houve uma redução acentuada quando passou de 13 para 14 canais, e depois continuou caindo monotonicamente até a situação onde não há interferência, que corresponde a 1 usuário por canal. Ilustra-se assim o fato de que pequenos relaxamentos na condição de número de canais podem gerar diminuição significativa na potência máxima requerida

Se parâmetros são alterados de modo que o número de subconjuntos e-viáveis cresce, tem-se então maior flexibilidade para redução da quantidade de canais ou para redução da mínima potência máxima. A figura abaixo ilustra uma situação idêntica ao caso anterior onde se reduziu a SINR requerida de 25dB para 20dB, com o conseqüente aumento da quantidade de subconjuntos e-viáveis de 134 para 237. A mínima potência máxima nas situações de 11 e 12 canais caiu de 0.829 para 0.216, representando uma queda percentual de 74%.

No. De Canais	Min. Pot. Max. (W)	Redução Percentual
11	0.829	0.00
12	0.216	73.94
13	0.105	87.33
14	0.1	87.93

Tabela 10 - Efeito do relaxamento da condição de número de canais caso 2

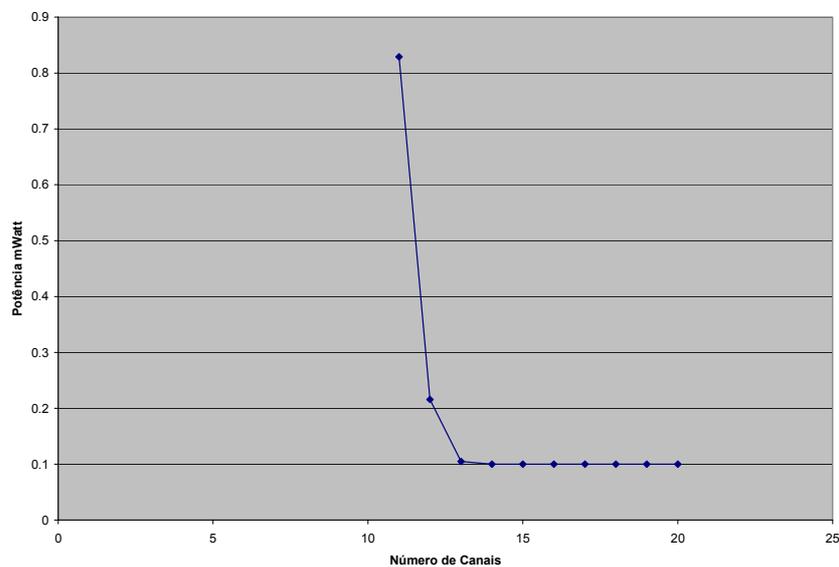


Figura 5-4 – Efeito do relaxamento da condição de número de canais 2

5.4.

Heurística para determinação do número de canais

Os procedimentos utilizados na seção anterior são extremamente caros sob o ponto de vista computacional, principalmente a rotina feasible, e a de mínimo

máximo. Portanto a busca de estratégias sub-ótimas, e de reduzido esforço computacional muitas vezes são necessárias.

A heurística proposta e descrita na seção 4.5 deste trabalho descreve um procedimento onde apenas é necessário a investigação da e-viabilidade de ordem 2. A geração de subconjuntos de ordem superior a 2 é feita somente levando em conta a compatibilidade 2 a 2. Assim, subconjuntos gerados podem não ser e-viáveis, de modo a requerer uma repartição dos mesmos para obter subconjuntos e-viáveis.

Abaixo apresentamos duas saídas típicas em que a primeira investigação par a par foi suficiente para gerar uma solução e-viável, enquanto que no segundo caso vários subconjuntos não e-viáveis foram gerados. Entretanto tais conjuntos podem ser fracionados de forma rápida e fácil, por exemplo retirando um usuário e o colocando num canal livre.

```

Raio espectral de H = 1017.5914
===== PROGRAMA DE VIABILIDADE DE CANAIS =====

No. de ERBs      : 8
No. de USUARIOS  : 20
Usuario 1 incompativel com 2 9 13 14 16 18 20
Usuario 2 incompativel com 1 8 9 13 14 16 18 20
Usuario 3 incompativel com 8 9 10 12 13 14 17 20
Usuario 4 incompativel com 5 6 7 9 11 13 14 15 20
Usuario 5 incompativel com 4 6 7 8 9 10 11 13 14 15 17 19 20
Usuario 6 incompativel com 4 5 7 8 9 11 13 14 15 19 20
Usuario 7 incompativel com 4 5 6 8 9 11 13 14 15 17 19 20
Usuario 8 incompativel com 2 3 5 6 7 9 11 12 13 14 15 17 19 20
Usuario 9 incompativel com 1 2 3 4 5 6 7 8 10 11 12 13 14 15
16 17 18 19 20
Usuario 10 incompativel com 3 5 9 12 13 14 15 17 20
Usuario 11 incompativel com 4 5 6 7 8 9 13 14 15 19 20
Usuario 12 incompativel com 3 8 9 10 13 14 15 17 20
Usuario 13 incompativel com 1 2 3 4 5 6 7 8 9 10 11 12 14 15
16 17 18 19 20
Usuario 14 incompativel com 1 2 3 4 5 6 7 8 9 10 11 12 13 15
16 17 18 19 20
Usuario 15 incompativel com 4 5 6 7 8 9 10 11 12 13 14 17 19 20
Usuario 16 incompativel com 1 2 9 13 14 18 20
Usuario 17 incompativel com 3 5 7 8 9 10 12 13 14 15 20
Usuario 18 incompativel com 1 2 9 13 14 16 20
Usuario 19 incompativel com 5 6 7 8 9 11 13 14 15 20
Usuario 20 incompativel com 1 2 3 4 5 6 7 8 9 10 11 12 13 14
15 16 17 18 19

ALGORIMO DE ALOCACAO DE CANAIS

```

```

Canal 1 -> Usuarios = 4 19    OK
Canal 2 -> Usuarios = 6 10    OK
Canal 3 -> Usuarios = 9      OK
Canal 4 -> Usuarios = 13     OK
Canal 5 -> Usuarios = 14     OK
Canal 6 -> Usuarios = 8 16    OK
Canal 7 -> Usuarios = 5 12 18  OK
Canal 8 -> Usuarios = 20     OK
Canal 9 -> Usuarios = 1 11 17 OK
Canal 10 -> Usuarios = 2 7    OK
Canal 11 -> Usuarios = 3 15   OK

```

Duração = 1.6 seg

```

Raio espectral de H = 1642.9404
===== PROGRAMA DE VIABILIDADE DE CANAIS =====
No. de ERBs      : 8
No. de USUARIOS  : 20

No. de incompatibilidades = 63

Usuario 1 incompativel com 2 3 6 12 17 18 20
Usuario 2 incompativel com 1 3 6 12 17 18 20
Usuario 3 incompativel com 1 2 6 10 12 16 17
Usuario 4 incompativel com 8 9 13 17
Usuario 5 incompativel com 11 14 15 17 19
Usuario 6 incompativel com 1 2 3 10 12 16 17
Usuario 7 incompativel com 8 9 10 13 16 17
Usuario 8 incompativel com 4 7 9 13 17
Usuario 9 incompativel com 4 7 8 13 17
Usuario 10 incompativel com 3 6 7 12 16 17
Usuario 11 incompativel com 5 14 15 17 18 19
Usuario 12 incompativel com 1 2 3 6 10 16 17
Usuario 13 incompativel com 4 7 8 9 17
Usuario 14 incompativel com 5 11 15 17 19
Usuario 15 incompativel com 5 11 14 17 19
Usuario 16 incompativel com 3 6 7 10 12 17
Usuario 17 incompativel com 1 2 3 4 5 6 7 8 9 10 11 12 13
14 15 16 18 19 20
Usuario 18 incompativel com 1 2 11 17 20
Usuario 19 incompativel com 5 11 14 15 17
Usuario 20 incompativel com 1 2 17 18

```

ALGORIMO DE ALOCACAO DE CANAIS

```

Canal 1 -> Usuarios = 4 5 7    NAO E-VIAVEL
Canal 2 -> Usuarios = 2 10 11  NAO E-VIAVEL
Canal 3 -> Usuarios = 3 14 18  NAO E-VIAVEL
Canal 4 -> Usuarios = 6 13 15 20 OK
Canal 5 -> Usuarios = 8 12 19  OK
Canal 6 -> Usuarios = 17      OK
Canal 7 -> Usuarios = 1 9 16   NAO E-VIAVEL

```

Duração = 0.1 seg

Foi desenvolvido um script em MATLAB que implementa o algoritmo sub-ótimo descrito em 4.5. Ao mesmo tempo o pacote GLPK foi utilizado para obter o número mínimo de canais. Realizaram-se 26 simulações diferentes, onde se variou

a semente, o número de usuários, e o número de receptores. O objetivo foi avaliar a qualidade da solução produzida pela heurística definida em 4.5, e estes resultados podem ser visualizados no gráfico abaixo. Nos casos simulados a maior diferença entre os dois métodos foi de dois canais. Este parece ser um pequeno preço a ser pago quando se considera a redução maciça do tempo computacional. Conforme indicado nas figuras acima, o tempo total de geração de alocação sub-ótima de canais é da ordem de 2 seg, um tempo em média bem inferior a execução do feasible, necessário para a obtenção de soluções ótimas.

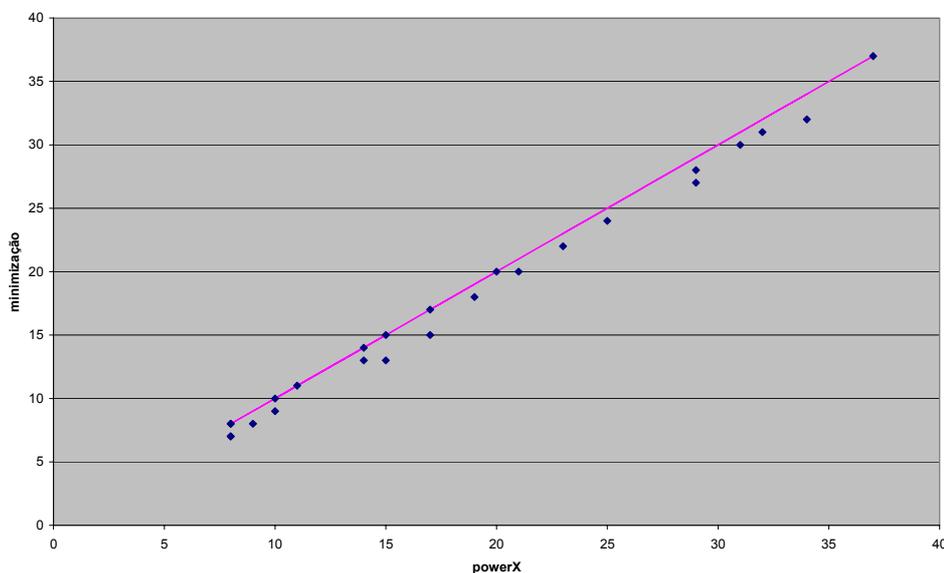


Figura 5-5 – Comparação do algoritmo sub-ótimo com o resultado da otimização

A vantagem desse tipo de abordagem é clara, pois a um custo computacional bem menor, consegue-se resultados qualitativamente tão bons quanto o da otimização anteriormente feita.