

**Algoritmos para o problema  
da cobertura por sensores**

Rafael da Ponte Barbosa

DISSERTAÇÃO APRESENTADA  
AO  
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA  
DA  
UNIVERSIDADE DE SÃO PAULO  
PARA  
OBTENÇÃO DO TÍTULO  
DE  
MESTRE EM CIÊNCIAS

Programa: Ciência da Computação  
Orientadora: Profa. Dra. Yoshiko Wakabayashi

Durante o desenvolvimento deste trabalho o autor recebeu auxílio financeiro da FAPESP

São Paulo, fevereiro de 2012

# Algoritmos para o problema da cobertura por sensores

Esta dissertação contém as correções e alterações sugeridas pela Comissão Julgadora durante a defesa realizada por Rafael da Ponte Barbosa em 12/12/2011.

O original encontra-se disponível no Instituto de Matemática e Estatística da Universidade de São Paulo.

Comissão Julgadora:

- Profa. Dra. Yoshiko Wakabayashi (orientadora) - IME-USP
- Prof. Dr. Flávio Keidi Miyazawa - UNICAMP
- Profa. Dra. Gordana Manic - UFABC

# Agradecimentos

Em primeiro lugar, agradeço aos meus pais e a meu irmão, que, perto ou longe, sempre me deram apoio incondicional ao longo dessa trajetória e se preocuparam com meu bem-estar. Boa parte deste trabalho se deve a eles.

Agradeço também profundamente à minha orientadora, professora Yoshiko Wakabayshi, por todos os ensinamentos, toda paciência e toda ajuda durante esses anos de trabalho.

Por fim, agradeço aos amigos pelas discussões e pelos momentos de descontração. Em especial, Guilherme Mota e Roberto Parente (os demais componentes do “trio cearense”), que estiveram comigo desde o início aqui em São Paulo.



# Resumo

Neste trabalho estudamos aspectos algorítmicos do Problema da Cobertura por Sensores. Em linhas gerais, neste problema a entrada consiste em uma região a ser monitorada por um conjunto de sensores previamente posicionados, cada qual dotado de bateria com duração limitada, e o objetivo é atribuir a cada sensor um tempo de início, de modo que toda a região seja coberta o maior tempo possível.

Focamos nosso estudo no caso unidimensional do problema, chamado Problema da Cobertura de Faixa Restrita, no qual a região a ser monitorada é um intervalo (da reta real). Estudamos diversas variantes, de acordo com os subintervalos que os sensores cobrem (se de tamanhos fixos ou variados), e de acordo com a duração das baterias (se uniformes ou não). Estudamos também o caso preemptivo: quando os sensores podem ser ligados mais de uma vez. Para este último caso, projetamos um algoritmo polinomial bem simples.

O Problema da Cobertura de Faixa Restrita é NP-difícil no caso não-preemptivo em que os sensores têm bateria de duração variável. Para este caso, em 2009 Gibson e Varadarajan apresentaram um algoritmo polinomial que provaram ser uma 5-aproximação. Provamos que este algoritmo tem fator de aproximação 4, e mostramos que este fator é justo. Apresentamos também formulações lineares inteiras para este caso, e os resultados computacionais obtidos.

**Palavras-chave:** Problema da Cobertura por Sensores, Problema da Cobertura de Faixa Restrita, Algoritmos de Aproximação, Programação Inteira.



# Abstract

We study the algorithmic aspects of the Sensor Cover Problem. Broadly speaking, in this problem the input consists of a region to be covered by a set of sensors previously positioned, each one powered with a battery of limited duration, and the objective is to assign to each sensor an initial time, so as to cover the given region for as long as possible.

We focus our study on the one-dimensional case of the problem, called Restricted Strip Cover Problem, in which the region to be covered is an interval (of the real line). We study several variants, according to the type of the subintervals the sensors cover (if they have fixed length or not), to the duration of the batteries (if uniform or not). We also study the preemptive case: when the sensors can be turned on and off more than once. For this case, we designed a simple polynomial-time algorithm.

The Restricted Strip Cover Problem is NP-hard in the non-preemptive case in which the sensors have non-uniform duration batteries. For this case, in 2009 Gibson and Varadarajan designed a polynomial-time algorithm which they proved to be a 5-approximation. We prove that this algorithm has approximation ratio 4, and show that this ratio is tight. We also present two integer linear formulations for this case, and report on the computational results obtained with this approach.

**Keywords:** Sensor Cover Problem, Restricted Strip Cover Problem, Approximation Algorithms, Integer Programming.



# Sumário

<b>Lista de Figuras</b>	<b>ix</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Problema da Cobertura por Sensores . . . . .	2
1.1.1 Caso unidimensional . . . . .	3
1.2 Problemas relacionados . . . . .	3
1.3 Organização do trabalho . . . . .	5
<b>2 Caso unidimensional</b>	<b>7</b>
2.1 Caso uniforme . . . . .	7
2.2 Caso variável . . . . .	8
2.2.1 Algoritmo de Gibson e Varadarajan . . . . .	8
2.3 Complexidade . . . . .	14
<b>3 Caso unidimensional: duas formulações lineares inteiras</b>	<b>17</b>
3.1 Formulação 1 . . . . .	17
3.2 Formulação 2 . . . . .	18
3.3 Testes computacionais . . . . .	18
<b>4 Caso unidimensional com preempção</b>	<b>21</b>
4.1 Algoritmo para o caso unidimensional com preempção . . . . .	21
<b>5 Caso geral</b>	<b>25</b>
5.1 Algoritmo de aproximação . . . . .	25
5.2 Complexidade . . . . .	27
<b>6 Considerações finais</b>	<b>29</b>
<b>A NP-Completeness do Problema da Alocação Dinâmica de Memória</b>	<b>31</b>
<b>Referências Bibliográficas</b>	<b>33</b>



# Lista de Figuras

1.1	Exemplo introdutório do problema . . . . .	1
1.2	Exemplo introdutório do problema – um agendamento . . . . .	2
1.3	Exemplo introdutório do problema – agendamento ótimo . . . . .	2
1.4	Agendamentos para instância do CFR . . . . .	4
2.1	Intervalo $[i, j]$ . . . . .	10
2.2	Casos (a), (b1) and (b2), respectivamente . . . . .	11
2.3	(a) O agendamento devolvido pelo algoritmo. (b) Um agendamento ótimo. . . . .	14
2.4	As partes inicial e final de um agendamento ótimo de $(U_I, S_I)$ . . . . .	15
2.5	Instância do CFR com $L = 4$ e $OPT = 3$ . . . . .	16



# Capítulo 1

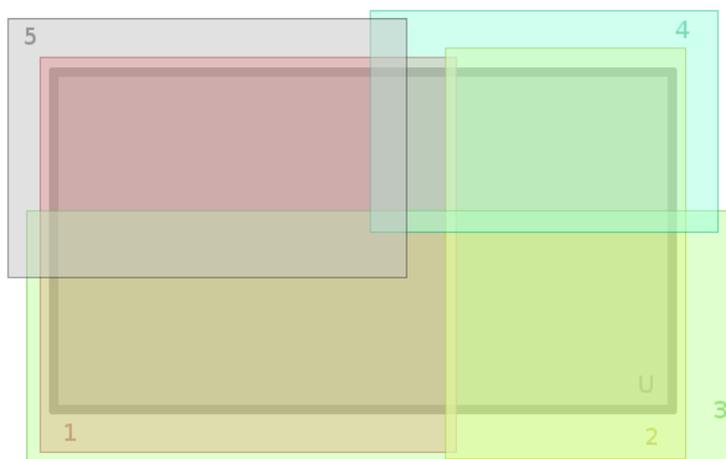
## Introdução

Redes de sensores possuem diversas aplicações práticas. Contudo, o uso de energia é um fator crítico na implementação destas redes, devido ao fato de que muitos sensores são equipados com baterias não recarregáveis. Desta forma, um problema de grande importância na área é encontrar uma estratégia eficiente de uso da energia, de modo a estender o tempo em que a região monitorada pela rede é coberta.

Há diversas variantes desse problema, que dependem dos tipos de sensores utilizados, da estrutura da rede e dos objetivos a serem alcançados. Para um estudo sobre uso eficiente de energia em redes de sensores, indicamos a resenha de Wang e Xiao [WX06].

Nosso interesse aqui é na seguinte variante: considere uma região (uma cerca, uma área ou um espaço) a ser monitorada e um conjunto de sensores já posicionados, cada um com uma bateria limitada e capaz de cobrir parte da região em questão. O objetivo é indicar para cada sensor um tempo de início de funcionamento de modo que toda a região seja coberta pelo maior tempo possível.

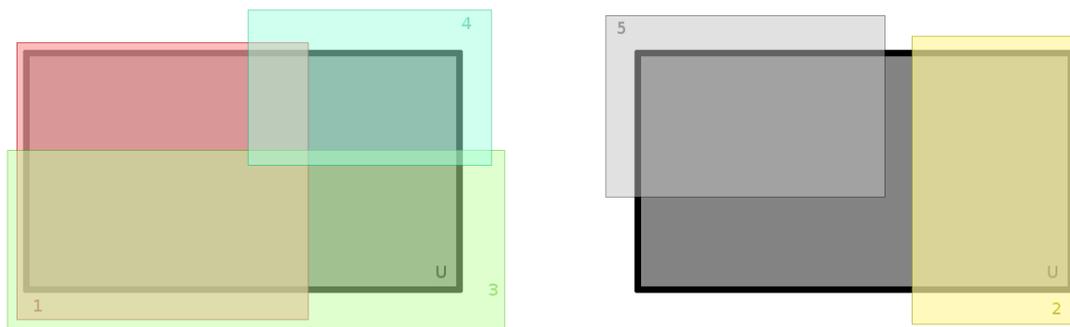
Para exemplificar, considere a Figura 1.1, que ilustra um plano  $U$  como região a ser coberta e cinco sensores. Consideraremos aqui que todos os sensores possuem duração de bateria unitária.



**Figura 1.1:** *Exemplo introdutório do problema*

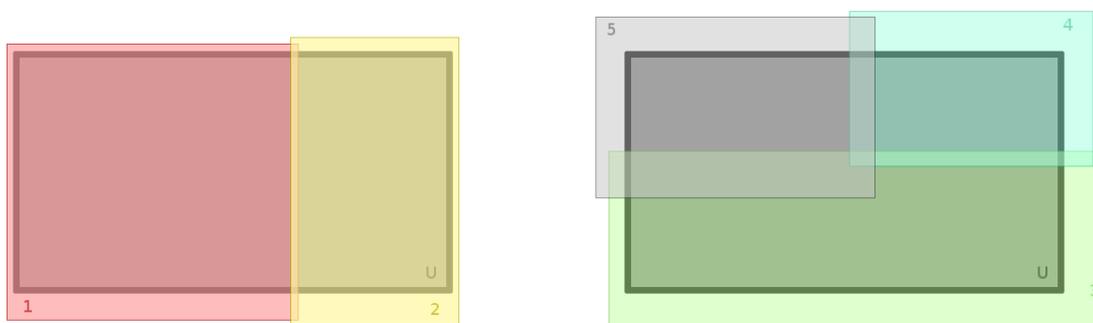
A Figura 1.2 mostra um exemplo de agendamento, isto é, uma possível atribuição de tempos de início aos sensores. Note, porém, que no exemplo de agendamento da Figura 1.2, a região seria coberta somente por uma unidade de tempo, na cobertura mostrada na imagem à esquerda, uma

vez que os sensores restantes, exibidos na imagem à direita, não cobrem a região completamente.



**Figura 1.2:** *Exemplo introdutório do problema – um agendamento*

A Figura 1.3 mostra um agendamento ótimo, de duração dois, onde cada uma das duas imagens forma uma cobertura completa da região.



**Figura 1.3:** *Exemplo introdutório do problema – agendamento ótimo*

Uma definição formal deste problema será apresentada na próxima seção. Conforme veremos, trata-se de um problema NP-difícil.

Temos interesse nos aspectos algorítmicos deste problema. Dentre as abordagens que usaremos em nossas investigações, daremos ênfase especial a algoritmos de aproximação, isto é, algoritmos eficientes que encontram solução cujo valor está dentro de um certo fator do valor de uma solução ótima do problema. É de nosso interesse também resultados relacionados a limites de aproximabilidade, ou seja, um limiar além do qual a solução do problema não pode ser aproximada, a menos que  $P = NP$ .

## 1.1 Problema da Cobertura por Sensores

Seja uma região  $U$  e um conjunto de  $n$  sensores  $S$ . Suponha que cada sensor  $s$  em  $S$  possui uma bateria de duração  $d(s)$  finita e uma região de cobertura  $R(s)$ , que está contida em  $U$ . Dado um sensor  $s$ , para todo  $i$  em  $R(s)$ ,  $s$  é dito *efetivo* em  $i$ .

Além disso, cada sensor  $s$  pode permanecer *ativo* por uma duração finita  $d(s)$ . Na versão *não-preemptiva* do problema, cada sensor só pode ser ligado uma vez, permanecendo ativo até que termine a carga de sua bateria, em contraste com a versão *preemptiva*, onde os sensores podem

ser desligados e ligados novamente conforme for conveniente. Trataremos, ao longo deste texto, da versão não-preemptiva do problema, a menos que explicitamente indicado.

Definimos um *agendamento*  $A$  para um conjunto  $S$  de sensores cobrindo  $U$  como uma atribuição a cada sensor  $s$  em  $S$  de um tempo inicial de funcionamento  $t(s)$ . Dado um agendamento, dizemos que um sensor  $s$  cobre uma posição  $i \in U$  no tempo  $t$  se  $i \in R(s)$  e  $t(s) \leq t < t(s) + d(s)$ .

Definimos ainda a *duração de um agendamento*  $A$  na posição  $i$ , como o maior tempo em que a posição  $i$  é coberta em  $A$ :

$$M(A, i) = \max\{t : \forall t' \leq t, \exists s \in S \text{ tal que } s \text{ cobre } i \text{ no tempo } t'\}.$$

A *duração do agendamento*  $A$  é então definida pela menor das durações em cada uma das posições:

$$M(A) = \min_i \{M(A, i)\}.$$

O problema central de nosso estudo, introduzido por Buchsbaum e outros [BEJ<sup>+</sup>07], é definido a seguir.

**Problema 1.** *O Problema da Cobertura por Sensores (CS), na sua forma geral, consiste em encontrar um agendamento de maior duração dentre todos possíveis para uma dada região  $U$  e um conjunto  $S$  de sensores. Dizemos que tal agendamento é ótimo, e tem duração  $OPT$ .*

Definimos a *carga* em uma posição  $i \in U$  como  $L(i) = \sum_{s \in S, s \text{ efetivo em } i} d(s)$ . A *carga total*  $L$  é dada por  $L = \max_i L(i)$ . Note que esta definição não depende de um agendamento. Por outro lado, é fácil ver que  $OPT \leq L$ .

Concentraremos nosso estudo no caso unidimensional do problema, apresentado a seguir.

### 1.1.1 Caso unidimensional

Denominamos o caso unidimensional como *Problema da Cobertura de Faixa Restrita* (CFR). A região  $U$  a ser coberta é um intervalo (uma cerca, por exemplo) descrita por um conjunto de  $m$  pontos, isto é,  $U = \{1, \dots, m\}$ . A cobertura de um sensor  $s$  é dada pelas duas extremidades do intervalo que  $s$  é capaz de cobrir:  $R(s) = [l(s), r(s)]$ , onde  $1 \leq l(s) \leq r(s) \leq m$ .

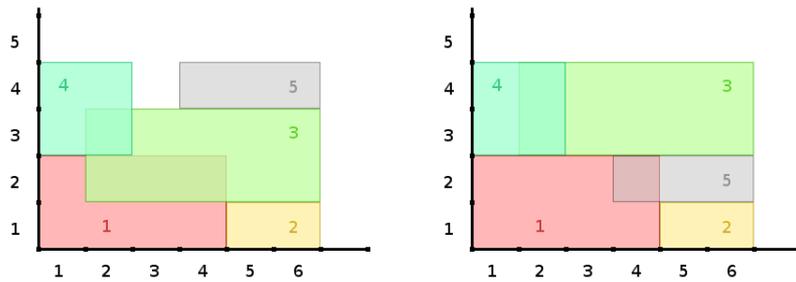
Podemos imaginar, neste caso, cada sensor como um retângulo no plano com base em  $[l(s), r(s)]$  e altura  $d(s)$ , a duração de sua bateria. O objetivo consiste então em deslizar os retângulos verticalmente de modo a maximizar a área do retângulo com base em  $U$  formado pela união deles.

A Figura 1.4 ilustra esta intuição. Nela, consideramos  $U = \{1, 2, 3, 4, 5, 6\}$  e um conjunto de 5 sensores. Note que o agendamento da esquerda tem duração 3, pois a posição 3 não é coberta no tempo 4. À direita, obtemos um agendamento de duração 4 movendo verticalmente os sensores 3 e 5.

## 1.2 Problemas relacionados

No contexto de uso eficiente de energia em redes de sensores, Slijepcevic e Potkonjak [SP01] introduziram o problema a seguir.

**Problema 2.** *O Problema da  $k$ -Cobertura de Conjuntos consiste em, dados conjunto  $U$ , coleção  $S$  de subconjuntos de  $U$  e inteiro  $k$ , decidir se é possível obter pelo menos  $k$  coberturas disjuntas de  $U$ .*



**Figura 1.4:** Agendamentos para instância do CFR

Os autores provaram que o problema é NP-completo. Abrams, Goel e Plotkin [AGP04] estudaram a variação deste problema em que o objetivo é particionar a coleção  $S$  de modo a maximizar o número de conjuntos da partição em que aparece cada elemento de  $U$ . Eles propõem três algoritmos de aproximação para o problema: um aleatorizado, um guloso distribuído e um guloso centralizado, que garantem, respectivamente, fatores de  $1 - 1/e$ ,  $1/2$  e  $1 - 1/e$ . Eles provam ainda que não é possível aproximar o problema com fator melhor que  $15/16$ , a menos que  $P = NP$ .

Considere também o Problema de Empacotamento de Coberturas por Conjuntos, abordado em [FHK00]. Este problema pode ser considerado um caso particular do Problema da Cobertura por Sensores, onde todos os sensores possuem duração unitária.

**Problema 3.** O Problema do Empacotamento de Coberturas por Conjuntos (ECC) consiste em, dados conjunto  $U$  e coleção  $S$  de subconjuntos de  $U$ , encontrar o maior número de coberturas mutuamente disjuntas.

Feige e outros [FHK00] mostraram que este é um problema NP-difícil e, além disso, não é possível aproximar ECC com fator melhor que  $\ln n$ , a menos que  $NP \subseteq DTIME(n^{O(\log \log n)})$ .

Por fim, outro problema relacionado é o Problema da Alocação Dinâmica de Memória.

**Problema 4.** Considere um conjunto  $U = \{1, \dots, m\}$  indicando unidades de tempo e um conjunto  $S$  de tarefas, onde cada tarefa  $j$  possui um tempo de chegada  $l(j)$  e um tempo de saída  $r(j)$ , além de necessitar de uma quantidade de memória  $d(j)$ .

O Problema da Alocação Dinâmica de Memória (ADM) consiste em atribuir a cada tarefa  $j$  uma posição de início na memória  $t(j)$  de modo a minimizar o máximo  $t(j) + d(j) - 1$ , para toda tarefa  $j$ , garantindo que  $[t(j_1), \dots, t(j_1) + d(j_1) - 1] \cap [t(j_2), \dots, t(j_2) + d(j_2) - 1] = \emptyset$  sempre que  $[l(j_1), r(j_1)] \cap [l(j_2), r(j_2)] \neq \emptyset$ , para quaisquer duas tarefas  $j_1$  e  $j_2$  distintas.

Note que uma instância deste problema é como uma instância do Problema da Cobertura de Faixa Restrita, isto é, um conjunto de retângulos (cada um representando uma tarefa) no plano que só podem ser movidos verticalmente. No Problema da Alocação Dinâmica de Memória, contudo, devemos assegurar a restrição de que não há sobreposição de quaisquer dois retângulos, e o objetivo é minimizar o maior  $t(j) + d(j) - 1$ , para toda tarefa  $j$ .

Stockmeyer [GJ79] demonstrou através de redução a partir do Problema da 3-Partição que, dada uma instância de ADM, decidir se  $OPT = L$  é NP-completo, onde  $L = \max_{i \in U} L(i)$ . A prova é exibida em [BEJ<sup>+</sup>07]. Reproduzimo-la no Apêndice A.

Kierstead [Kie88] propôs o primeiro algoritmo polinomial de aproximação de fator constante (igual a 80) em 1988. Posteriormente, o mesmo autor obteve um algoritmo de razão 6 [Kie91].

Gergov, em seguida, propôs um algoritmo que garantia fator igual a 5 em 1996 [Ger96] e depois outro de fator igual a 3 em 1999 [Ger99]. Mais recentemente, Buchsbaum e outros [BKK<sup>+</sup>04] propuseram uma  $(2 + \epsilon)$ -aproximação para o problema, a melhor obtida até o momento.

### 1.3 Organização do trabalho

Apresentamos no Capítulo 2 resultados conhecidos sobre o Problema da Cobertura de Faixa Restrita, o caso unidimensional do Problema da Cobertura por Sensores. Dentre eles, descrevemos um algoritmo de aproximação obtido por Gibson e Varadarajan [GV09], que foi provado ter fator constante 5. Provamos que este algoritmo tem fator de aproximação 4 e mostramos que este fator é justo. Também mencionamos algoritmos polinomiais para casos especiais e resultados sobre complexidade computacional.

No Capítulo 3, introduzimos formulações lineares inteiras para o Problema da Cobertura de Faixa Restrita, uma abordagem ainda não tratada na literatura, e exibimos os resultados computacionais obtidos.

No Capítulo 4, tratamos o caso preemptivo do problema Problema da Cobertura de Faixa Restrita. Apresentamos um algoritmo polinomial que desenvolvemos para este caso, que é bem simples e tem complexidade quadrática.

Um algoritmo de aproximação probabilístico para caso geral do problema, devido a Buchsbaum, Efrat, Shaili, Venkatasubramanian e Yi [BEJ<sup>+</sup>07], é apresentado no Capítulo 5, assim como um limiar de aproximabilidade.



# Capítulo 2

## Caso unidimensional

Conforme vimos no Capítulo 1, o caso unidimensional do Problema da Cobertura por Sensores é chamado Problema da Cobertura de Faixa Restrita. Nele, a região  $U$  a ser coberta é um intervalo, dada por um conjunto de  $m$  pontos, ou seja,  $U = \{1, \dots, m\}$ ; e a cobertura de um sensor  $s$  é um subintervalo  $R(s) = \{l(s), \dots, r(s)\}$ , onde  $l(s)$  representa a extremidade esquerda do intervalo e  $r(s)$  a direita.

Deste modo, conforme mostrado também no Capítulo 1 os sensores podem ser pensados como retângulos no plano, com base em  $\{l(s), \dots, r(s)\}$  e altura  $d(s)$ , a duração de sua bateria. O objetivo é maximizar a altura do retângulo formado pela união deles, com base em  $U$ .

No restante deste capítulo, dividimos a análise em dois casos: *caso uniforme* (Seção 2.1) em que todos os sensores possuem baterias de duração uniforme; e *caso variável* (Seção 2.2), onde os sensores possuem baterias de duração variada.

### 2.1 Caso uniforme

Para o caso de sensores com baterias de duração uniforme, Buchsbaum e outros [BEJ<sup>+</sup>07] mostraram que um algoritmo polinomial guloso é capaz de devolver solução exata de valor  $L$ .

O Algoritmo 1, mostrado abaixo, aloca os sensores de modo a cobrir as posições  $i$  do intervalo de 1 a  $m$ .

---

**Algoritmo 1** CFR-Uniforme ( $U, S$ )

---

- 1:  $i \leftarrow 1$
  - 2: Selecciona  $L$  sensores efetivos em 1
  - 3: Enquanto  $i < m$  faça
  - 4:      $i \leftarrow i + 1$
  - 5:     Se há lacuna na posição  $i$
  - 6:         Aloca sensor (ainda não alocado) para cada lacuna em  $i$
- 

Note que, no Algoritmo 1, para cada posição representada por  $i$ , as seguintes invariantes são mantidas:

1. não há sobreposição em posição  $x$  maior ou igual a  $i$ ; e
2.  $M(A, i) = L$ .

Em  $i = 1$ , o algoritmo seleciona  $L$  sensores quaisquer e os aloca sem sobreposição, estabelecendo inicialmente as invariantes. Assumindo que as invariantes são verdadeiras em  $i$ , alocamos os sensores em  $i + 1$  da seguinte forma. Se não há “buracos” em  $i + 1$ , não há nada a fazer e as invariantes se estendem a esta posição. Se não, assumamos que há  $k > 0$  “buracos” de duração unitária em  $i + 1$ . As invariantes implicam que há pelo menos  $k$  sensores efetivos em  $i + 1$  ainda não agendados, que podem ser utilizados para preencher os espaços, preservando as invariantes.

## 2.2 Caso variável

Para o caso de sensores com baterias de duração variável, Buchsbaum e outros [BEJ<sup>+</sup>07] exibiram também um algoritmo que é uma  $O(\log \log \log n)$ -aproximação.

A abordagem do algoritmo é agrupar sensores de menor duração em um sensor de maior duração, de modo a transformar o caso variável no caso uniforme e aplicar então o algoritmo exato citado na seção anterior. Contudo, a redução implica uma perda não constante na carga total  $L$ , e por isso o fator  $O(\log \log \log n)$ .

Mais recentemente, Gibson e Varadarajan [GV09] publicaram um algoritmo simples com fator de aproximação igual a 5 para o problema, mostrando que o CFR pertence a APX.

### 2.2.1 Algoritmo de Gibson e Varadarajan

Detalhamos a seguir o algoritmo de aproximação proposto por Gibson e Varadarajan, em [GV09].

Antes de apresentarmos o algoritmo, precisamos de algumas definições. Com respeito a um agendamento  $A$ , dizemos que um sensor  $s$  *domina a coordenada  $i$  para a direita* se, dentre todos os sensores ainda não escolhidos e efetivos em  $i$ ,  $s$  é o que possui a maior coordenada de cobertura direita  $r(s)$ . Em caso de empate, é escolhido o sensor que possui a menor coordenada de cobertura esquerda  $l(s)$ . Qualquer outro empate é decidido arbitrariamente. Um sensor que *domina a coordenada  $i$  para a esquerda* é definido de modo análogo. Definimos ainda  $M(A, 0) = M(A, m + 1) = \infty$ .

---

#### Algoritmo 2 CFR-GV ( $U, S$ )

---

```

1:  $t \leftarrow 0$ 
2:  $A \leftarrow \emptyset$ ;  $M(A) = 0$ 
3: Enquanto verdade
4:    $t \leftarrow M(A) + 1$ 
5:    $i \leftarrow$  ponto mais à esquerda não coberto no tempo  $t$ 
6:    $j \leftarrow \max\{j' \in U : [i, j'] \text{ não é totalmente coberto no tempo } t\}$ 
7:    $s \leftarrow$  sensor que domina  $i$  para a direita
8:   Se  $s$  não existe, pare
9:   Senão,
10:     Se  $s$  é efetivo em  $j$  e  $M(A, i - 1) < M(A, j + 1)$ 
11:        $s' \leftarrow$  sensor que domina  $j$  para a esquerda
12:        $A \leftarrow A \cup \{s'\}$ ;  $t(s') \leftarrow t$ 
13:     Senão
14:        $A \leftarrow A \cup \{s\}$ ;  $t(s) \leftarrow t$ 
15: Devolve  $A$ 

```

---

Apresentamos a demonstração de que o algoritmo CFR-GV é, na verdade, uma 4-aproximação polinomial para o problema. A primeira parte da análise é similar à apresentada pelos autores

em [GV09]. Por completude, reproduzimos os lemas 5 e 6, provados no artigo citado.

**Lema 5** (Gibson e Varadarajan, 2009). *Seja  $A$  o agendamento devolvido pelo algoritmo CFR-GV quando aplicado a uma instância  $(U, S)$ . Sejam  $s'$  e  $s''$  dois sensores distintos que foram agendados em  $A$ . Se  $R(s'')$  está estritamente contido em  $R(s')$ , então  $s''$  é agendado em  $A$  após  $s'$  e, ademais,  $t(s'') \geq t(s') + d(s')$ .*

*Demonstração.* Um sensor só é agendado quando domina dada posição, para a esquerda ou para a direita. Suponha que  $s'$  e  $s''$  ainda não foram alocados e queremos achar o sensor que domina posição  $i \in R(s'')$ . Nesse caso, serão considerados  $s'$  e  $s''$ , mas  $s'$  será escolhido pela definição de dominância.

Desta forma,  $s'$  será agendado antes de  $s''$ , e não consideraremos outro sensor para dominar qualquer posição em  $R(s')$  até o instante  $t(s') + d(s')$ .  $\square$

Para qualquer ponto  $i \in U$  e tempo  $t > 0$ , definimos  $\text{cobertura}(i, t)$  como sendo o número de sensores que cobrem  $i$  no tempo  $t$  no agendamento devolvido pelo algoritmo. Definimos  $\text{MaxCobertura}$  de um agendamento  $A$ , denotado  $\text{MaxCobertura}(A)$ , como o valor  $\max\{\text{cobertura}(i, t) : i \in U \text{ e } t > 0\}$ . A duração de um agendamento  $A$  está relacionada com  $\text{MaxCobertura}(A)$  como segue.

**Lema 6** (Gibson e Varadarajan, 2009). *Seja  $A$  o agendamento devolvido pelo algoritmo CFR-GV. Se  $\text{MaxCobertura}(A) \leq c$ , então  $M(A) \geq L/c$ .*

*Demonstração.* Ao final da execução, há um ponto  $i' \in U$  tal que  $M(A, i') = M(A)$  e não há mais nenhum sensor não alocado efetivo em  $i'$ . Então,  $cM(A) \geq L(i') \geq L$  e, portanto,  $M(A) \geq L/c$ .  $\square$

Gibson e Varadarajan provaram que o algoritmo CFR-GV devolve um agendamento  $A$  tal que  $\text{MaxCobertura}(A) \leq 5$ , mostrando o fator 5 obtido pelo algoritmo. Provaremos a seguir que  $\text{MaxCobertura}(A) \leq 4$ , fazendo uma análise mais rigorosa. Para isso, precisaremos de mais algumas definições.

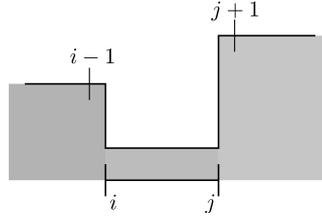
A cada iteração, para um tempo  $t$ , o algoritmo considera um ponto não coberto  $i$  e o maior ponto  $j$  tal que  $[i, j]$  não está coberto. O intervalo  $[i, j]$  define (geometricamente) um “vale”, considerando todos os sensores que já foram alocados até então, como ilustrado na Figura 2.1. Para tal  $[i, j]$ , um sensor  $s$  é escolhido e agendado em  $A$ . Dizemos que  $[i, j]$  é o *intervalo para o qual  $s$  foi alocado*, ou ainda, que  $s$  foi *alocado em  $A$  devido a  $[i, j]$* . Um sensor que é escolhido por dominar  $i$  para a direita é dito *direita-dominante*. De modo análogo, um sensor que é alocado por dominar  $j$  para a esquerda é dito *esquerda-dominante*. Se um ponto  $i$  não estava coberto no tempo  $t(s)$  antes que  $s$  fosse alocado, mas foi coberto por  $s$  no tempo  $t(s)$  (instante em que  $s$  é agendado), dizemos que  $s$  *fecha  $i$*  no tempo  $t(s)$ . Denotamos por  $A'_s$  o agendamento construído pelo algoritmo imediatamente antes da alocação do sensor  $s$ .

A seguir, adotamos a seguinte convenção: se um sensor é denotado  $s_p$ , então  $[i_p, j_p]$  representa o intervalo para o qual  $s_p$  foi agendado.

**Lema 7.** *Para qualquer agendamento  $A$  devolvido pelo algoritmo CFR-GV, temos*

$$\text{MaxCobertura}(A) \leq 4.$$

*Demonstração.* Seja  $(U, S)$  uma instância do CFR e seja  $A$  o agendamento devolvido pelo algoritmo CFR-GV. Fixamos um  $i \in U$  e um tempo  $t$ ,  $0 < t \leq M(A)$ . Mostraremos que  $\text{cobertura}(i, t) \leq 4$ .



**Figura 2.1:** Intervalo  $[i, j]$

Denotamos por  $s_0$  o primeiro sensor a cobrir  $i$  no tempo  $t$ . Por convenção,  $[i_0, j_0]$  é o intervalo para o qual  $s_0$  foi agendado. Agora classificamos qualquer outro sensor  $s_p$  que cobre  $i$  no tempo  $t$  nos quatro seguintes tipos:

- Tipo EE : se  $[i_p, j_p]$  está à esquerda de  $i$  e  $s_p$  é esquerda-dominante;
- Tipo ED : se  $[i_p, j_p]$  está à esquerda de  $i$  e  $s_p$  é direita-dominante;
- Tipo DE : se  $[i_p, j_p]$  está à direita de  $i$  e  $s_p$  é esquerda-dominante;
- Tipo DD : se  $[i_p, j_p]$  está à direita de  $i$  e  $s_p$  é direita-dominante;

Os principais componentes da prova são as três afirmativas a seguir.

**Afirmativa 8.** No máximo dois sensores de tipos EE ou ED são agendados em  $A$ .

*Demonstração da Afirmativa 8.* Sejam  $s_1$  e  $s_2$  os dois primeiros sensores de tipos EE ou ED que são agendados após  $s_0$ . Suponha que  $s_2$  é agendado depois de  $s_1$ . Considere os dois casos seguintes, ilustrados na Figura 2.2.

**Caso (a):** Sensor  $s_1$  é do tipo EE

Lembramos que  $[i_1, j_1]$  denota o intervalo para o qual  $s_1$  foi agendado. Como  $s_1$  é esquerda-dominante (domina  $j_1$  para a esquerda), temos  $l(s_1) \leq l(s_2)$ . Assim,  $i_2 \in [l(s_2), i] \subseteq [l(s_1), i]$ .

Neste caso, note que o intervalo  $[i_2, j_2]$  só pode ser considerado depois do tempo  $t(s_1) + d(s_1) - 1$  (ou seja, quando  $s_1$  não estiver mais ativo). Logo,  $t(s_2) > t$ . Mas então  $s_2$  não cobre  $i$  no tempo  $t$ , uma contradição. Logo, depois de um sensor de tipo EE, nenhum outro sensor de tipos EE ou DE é agendado.

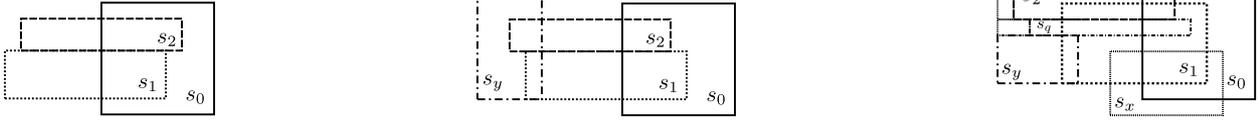
**Caso (b):** Sensor  $s_1$  é do tipo ED

Neste caso, como  $s_1$  é direita-dominante (domina  $i_1$  para a direita), e é efetivo em  $i$ , então é efetivo também em  $j_1$  e, portanto, temos

$$M(A'_{s_1}, i_1 - 1) \geq M(A'_{s_1}, j_1 + 1).$$

Se  $i_1 = 1$ , então claramente  $i_2$  está no intervalo  $[i_1, l(s_0)]$ . Neste caso, depois que  $s_1$  é agendado, o intervalo  $[i_1, i]$  é inteiramente coberto no tempo  $t$ . Assim, o algoritmo somente irá considerar um sensor de tipo EE ou ED depois do tempo  $t$ . Assim,  $t(s_2) > t$ . Mas, então,  $s_2$  não cobre  $i$  no tempo  $t$ , uma contradição.

Vamos assumir que  $i_1 > 1$ . Neste caso, o ponto  $i_1 - 1$  é coberto por algum sensor, digamos  $s_y$ , no tempo  $M(A'_{s_1}, i_1 - 1)$ . Note que  $l(s_2) > l(s_y)$  (respectivamente,  $l(s_1) > l(s_y)$ ), ou teríamos



**Figura 2.2:** Casos (a), (b1) and (b2), respectivamente

um contradição com Lema 5, pois teríamos  $R(s_y) \subset R(s_2)$  (respectivamente  $R(s_y) \subset R(s_1)$ ) e  $s_2$  (respectivamente  $s_1$ ) teria sido agendado em  $A$  antes de  $s_y$ .

Dado um sensor agendado  $s$ , definimos  $h(s) := t(s) + d(s) - 1$ , ou seja, o tempo em que termina a execução de  $s$ . Analisamos os dois subcasos.

**Subcaso (b1):**  $j_1 + 1 = l(s_0)$ .

Neste caso, sabemos que  $h(s_y) \geq h(s_0)$  (pois  $s_1$  é direita-dominante). Depois que  $s_1$  é agendado a  $A$ , todas posições no intervalo  $[l(s_y), i]$  são cobertas no tempo  $t$ . Como  $l(s_2) > l(s_y)$ , o algoritmo somente pode considerar posição  $i_2 \in [l(s_2), l(s_0)]$  depois que o sensor  $s_1$  não está mais ativo, isto é, depois do tempo  $t$ . Assim,  $s_2$  não cobre  $i$  no tempo  $t$ , uma contradição.

**Subcaso (b2):**  $j_1 + 1 < l(s_0)$ .

Seja  $s_x$  o sensor agendado em  $A$  antes de  $s_1$  tal que  $j_1 + 1 = l(s_x)$  e  $t(s_x) < t(s_1) \leq t(s_x) + d(s_x) - 1$  (isto é, no instante em que  $s_1$  foi agendado em  $A$ , o sensor  $s_x$  ainda estava ativo).

Agora considere o agendamento de  $s_2$ . Se  $s_2$  é de tipo EE, usando argumento análogo ao do Caso (a), concluímos que nenhum outro sensor de tipo EE ou ED é agendado em  $A$ . Então, vamos assumir agora que  $s_2$  é de tipo ED.

Note que  $i_2 > l(s_y)$  (pois  $l(s_2) > l(s_y)$ ). Da mesma forma, note que  $l(s_1) > l(s_y)$ .

Se  $h(s_1) \leq h(s_y)$ , então  $t \leq h(s_y)$ . Assim, como sabemos que  $i_2 > l(s_y)$ , segue que  $t(s_2) > t$ . Neste caso,  $s_2$  não cobre  $i$  no tempo  $t$ , uma contradição.

Vamos supor agora que  $h(s_1) > h(s_y)$ . Como  $s_2$  cobre  $i$  no tempo  $t$ , então  $[i_2, j_2]$  está no intervalo  $[l(s_y), l(s_1)]$ .

Argumentamos que  $j_2 + 1 = l(s_1)$ . Seja  $s_q$  o último sensor agendado antes de  $s_2$  tal que  $j_q + 1 = l(s_1)$ . Afirmamos (e provamos em seguida) que  $s_2$  somente é agendado depois do fim da execução de  $s_q$  e, portanto,  $j_2 + 1 = l(s_1)$ . Note que se não há tal  $s_q$ , a afirmação vale por vacuidade.

Se  $s_q$  foi escolhido por ser esquerda-dominante, devemos ter  $t(s_2) > h(s_q)$ , raciocinando como no Caso (a). Se não, se  $s_q$  foi escolhido por ser direita-dominante, temos  $M(A'_{s_q}, i_q - 1) \geq M(A'_{s_q}, j_q + 1) = h(s_1) \geq t$ . Note que após o agendamento de  $s_q$ , o intervalo  $[i_q - 1, i]$  está completamente coberto até o tempo  $h(s_q)$ . Logo,  $t(s_2) > h(s_q)$ .

Segue que  $j_2 + 1 = l(s_1)$ . Neste caso,  $M(A'_{s_2}, i_2 - 1) \geq M(A'_{s_2}, j_2 + 1)$ , pois  $s_2$  é direita-dominante. Como  $s_2$  é efetivo em todas posições em  $[i_1, j_1]$ , pela prioridade dada a  $s_1$ , concluímos que  $r(s_2) < r(s_1)$ .

Suponha agora que existe um terceiro sensor, digamos  $s_3$ , que é de tipo EE ou ED e é o próximo destes tipos agendados após  $s_2$ . Suponha  $i_2 > 1$  e seja  $s_r$  o sensor que cobre a posição  $i_2 - 1$  no tempo  $M(A'_{s_2}, i_2 - 1)$  (se  $i_2 = 1$ , é imediato que  $s_3$  não existe). Se  $h(s_2) \leq h(s_r)$ , usando argumentos similares àqueles usados anteriormente, podemos concluir que nenhum outro sensor agendado após  $s_2$  pode cobrir  $i$  no tempo  $t$ . Então, suponha  $h(s_2) > h(s_r)$ . Como  $h(s_r) > h(s_1)$  (pois  $s_2$  é direita-dominante) e  $t \leq h(s_1)$ , temos que  $h(s_r) > t$ . Uma vez que  $[i_3, j_3]$  deve estar contido no intervalo  $[l(s_r), l(s_2)]$ , é imediato que  $s_3$  não pode cobrir  $i$  no tempo  $t$ .

Resumimos o que provamos no subcaso (b2). Quando há sensores  $s_1$  de tipo ED e  $s_2$  de tipo EE ou ED (cobrindo  $i$  no tempo  $t$ ), provamos que existe um sensor  $s_x$  tal que vale o seguinte:

$$l(s_1) < l(s_x) \text{ e } r(s_1) < r(s_x). \quad (2.1)$$

□

**Afirmativa 9.** *No máximo dois sensores de tipos DE ou DD são agendados em  $A$ .*

*Demonstração da Afirmativa 9.* Sejam  $s_3$  e  $s_4$  os primeiros sensores de tipos DE ou DD que são agendados após  $s_0$ . Suponha que  $s_3$  é agendado antes de  $s_4$ . Analogamente à Afirmativa 8, analisamos **Caso (c)**, onde sensor  $s_3$  é do tipo DD, e **Caso (d)**, onde sensor  $s_3$  é do tipo DE.

No Caso (c), concluímos que nenhum sensor de tipos DE ou DD pode ser agendado após  $s_3$ . O Caso (d) é subdividido em dois subcasos: **Subcaso (d1)**, se  $i_3 - 1 = r(s_0)$ , e **Subcaso (d2)**, se  $i_3 - 1 > r(s_0)$ .

No Subcaso (d1), podemos concluir que  $s_4$  não cobre  $i$  no tempo  $t$ , uma contradição. No Subcaso (d2), podemos provar que ou  $s_4$  é um sensor de tipo DD e nenhum outro sensor de tipo DE ou DD é agendado posteriormente; ou que  $s_4$  é de tipo DE e, novamente, nenhum outro sensor de tipo DE ou DD é agendado depois dele.

Assim como no Caso (b2), neste subcaso temos que existe um sensor  $s_w$  tal que vale o seguinte.

$$l(s_w) < l(s_3) \text{ e } r(s_w) < r(s_3). \quad (2.2)$$

□

**Afirmativa 10.** *Os subcasos (b2) e (d2) não ocorrem simultaneamente.*

*Demonstração da Afirmativa 10.* Suponha, por contradição, que ambos subcasos ocorrem. Considere os sensores  $s_1$ ,  $s_2$  e  $s_x$  (respectivamente,  $s_3$ ,  $s_4$  e  $s_w$ ) que mencionamos na análise do subcaso (b2) (respectivamente, (d2)), que satisfazem as desigualdades (2.1) e (2.2). Destas desigualdades, e da hipótese de que  $s_1, \dots, s_4$  cobrem  $i$  no tempo  $t$ , concluímos que

$$r(s_x) > i > l(s_w). \quad (2.3)$$

Por outro lado, valem as seguintes afirmativas (provadas posteriormente):

**Afirmativa 11.** *Existe sensor  $\hat{s}_E$  tal que*

- (1)  $l(s_x) \leq l(\hat{s}_E) < l(s_0)$  e  $r(s_x) \leq r(\hat{s}_E) < r(s_0)$ ; e
- (2)  $t(\hat{s}_E) \leq t(s_0) \leq t(\hat{s}_E) + d(\hat{s}_E) - 1$ .

**Afirmativa 12.** *Existe sensor  $\hat{s}_D$  tal que*

- (1)  $l(s_0) < l(\hat{s}_D) \leq l(s_w)$  e  $r(s_0) < r(\hat{s}_D) \leq r(s_w)$ ; e
- (2)  $t(\hat{s}_D) < t(s_0) \leq t(\hat{s}_D) + d(\hat{s}_D) - 1$ .

Assim,  $s_0$  deve ser agendado depois do agendamento de  $\hat{s}_E$  e de  $\hat{s}_D$ , mas antes do fim da execução destes. Deste fato e das condições nas afirmativas 11 e 12, segue que

$$r(s_x) \leq r(\hat{s}_E) < i_0 \leq j_0 < l(\hat{s}_D) \leq l(s_w), \quad (2.4)$$

uma vez que o intervalo  $[i_0, j_0]$  não é coberto no tempo  $t(s_0)$  no agendamento  $A'_{s_0}$  (pois é fechado por  $s_0$ ). Temos, portanto, uma contradição com a desigualdade (2.3). Assim, terminamos a prova da Afirmativa 10.  $\square$

Das afirmativas 8, 9 e 10, concluímos que  $\text{MaxCobertura}(i, t) \leq 4$ .  $\square$

Provamos agora a Afirmativa 11. A prova da Afirmativa 12 é análoga.

*Demonstração da Afirmativa 11.* Sabemos que  $t(s_0) \leq t(s_1) \leq t(s_x) + d(s_x) - 1$ . Se  $t(s_x) \leq t(s_0)$ , então  $s_x$  satisfaz as condições exigidas para  $\hat{s}_E$  e portanto a afirmativa é verdade (tomando  $\hat{s}_E = s_x$ ). Vamos assumir que  $t(s_0) < t(s_x)$ . Ao longo desta prova, consideraremos  $i$  a posição e  $t$  o tempo fixados na prova do Lema 7.

Seja  $s'$  o primeiro sensor agendado depois de  $s_0$  tal que  $l(s_x) \leq l(s') < l(s_0)$  e  $t(s') < t(s_0) + d(s_0) - 1$ , e, mais ainda, existe sensor  $s''$  tal que  $j'' + 1 = l(s')$  e  $t(s') < t(s'') \leq t(s') + d(s') - 1$ , onde  $[i'', j'']$  é o intervalo para o qual  $s''$  foi agendado. Sabemos que existe pelo menos um tal  $s'$ , uma vez que pelas hipóteses  $s_x$  e  $s_1$  satisfazem as condições necessárias para  $s'$  and  $s''$ , respectivamente. Vale notar também que  $s'$  é efetivo em  $i$ , pois  $s'$  ou é  $s_x$  ou foi agendado antes deste.

Seja  $[i', j']$  o intervalo para o qual  $s'$  foi agendado. Note que  $j' \leq l(s_0)$ . De  $l(s'') < l(s')$ , e do fato de que  $s'$  foi agendado antes de  $s''$ , concluímos que  $s'$  foi agendado por ser direita-dominante (já que  $s''$  é efetivo em qualquer posição no intervalo  $[i', j']$ ). Note também que  $t > t(s') + d(s') - 1$ , pois  $s'$  é efetivo em  $i$  mas não cobre  $i$  no tempo  $t$ .

Suponha  $j' + 1 = l(s_0)$ . Neste caso,  $M(A'_{s'}, i' - 1) \geq M(A'_{s'}, j' + 1) = M(A'_{s'}, l(s_0)) \geq t > t(s') + d(s') - 1$ . Sob estas condições, teríamos  $t(s'') > t(s') + d(s') - 1$ , uma contradição com nossa escolha de  $s'$  e  $s''$ .

Assim,  $j' + 1 < l(s_0)$ . Neste caso, há algum sensor  $\hat{s}_E$  tal que  $j' + 1 = l(\hat{s}_E)$  e  $t(\hat{s}_E) < t(s') \leq t(\hat{s}_E) + d(\hat{s}_E) - 1$ .

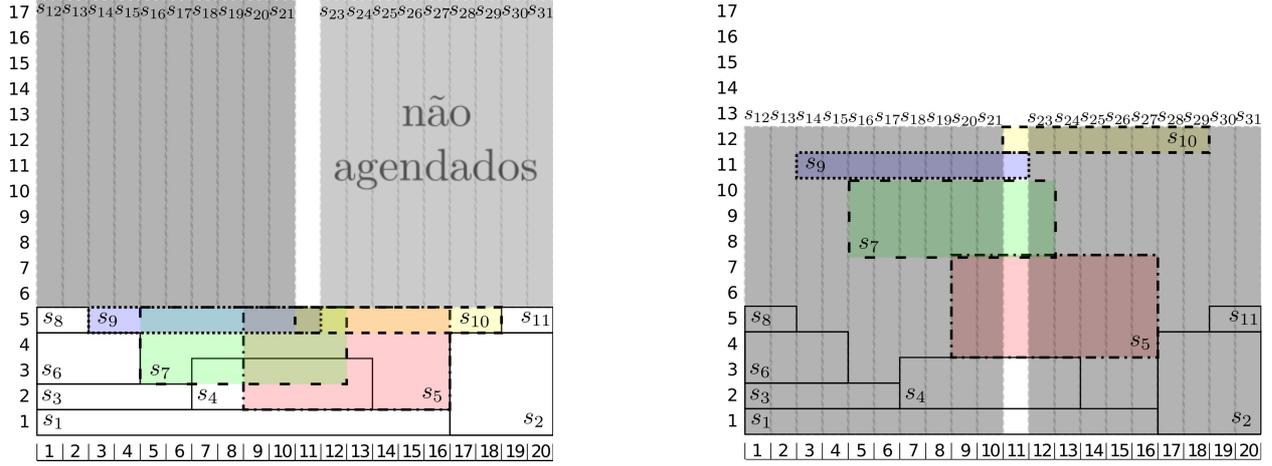
Note, contudo, que se  $\hat{s}_E$  é agendado depois de  $s_0$ , ele contradiz a escolha de  $s'$ . Logo, concluímos que  $\hat{s}_E$  é agendado antes de  $s_0$  e sua execução termina depois do agendamento de  $s_0$ . Ademais, temos que  $l(s_x) \leq l(\hat{s}_E) < l(s_0)$ . Deste fato, concluímos que  $r(s_x) \leq r(\hat{s}_E) < r(s_0)$ , pois, do contrário, teríamos uma contradição com Lema 5.  $\square$

**Teorema 13.** *O algoritmo CFR-GV é uma 4-aproximação polinomial para o CFR. Ademais, o fator 4 é justo.*

*Demonstração.* O fator de aproximação segue do Lema 6 e do Lema 7. O laço na linha 3 é executado no máximo  $n$  vezes, uma vez que um sensor é agendado em cada iteração. Cada iteração pode ser implementada para executar em tempo  $O(n + m) = O(n)$ , pois podemos assumir que  $m \leq 2n$ , sem perda de generalidade, já que uma instância do problema pode ser descrita pelas extremidades dos sensores. Assim, o algoritmo executa em tempo  $O(n^2)$ .

Para mostrar que o fator 4 é justo, considere primeiro a instância do problema CFR mostrada na Figura 2.3, onde temos uma saída  $A$  correspondente, dada pelo algoritmo, e um agendamento ótimo. Como podemos ver,  $M(A) = 5$ , mas  $OPT = L = 12$ .

Note que há um ponto  $i$  tal que  $\text{cobertura}(i, 5) = 4$  no agendamento  $A$  (Fig. 2.3(a)). Baseado nisto, podemos redimensionar as durações dos sensores na instância mostrada e obter uma família de instâncias em que  $OPT = L = 4(M(A) - 4) + 8$ . Deste modo, o fator de aproximação para tais instâncias é  $(4M(A) - 8)/M(A) = 4 - 8/M(A)$ . Para tal, basta escolher a duração dos sensores de



**Figura 2.3:** (a) O agendamento devolvido pelo algoritmo. (b) Um agendamento ótimo.

forma que  $\text{cobertura}(i, \cdot) = 4$  por um período de tempo  $M(A) - 4$ . Para um valor suficientemente grande de  $M(A)$ , este fator pode ser arbitrariamente próximo de 4, mostrando que o fator é justo.  $\square$

## 2.3 Complexidade

Dada a similaridade entre os problemas, a demonstração de NP-completude para o Problema da Alocação Dinâmica de Memória (reproduzida no Apêndice A) pode ser adaptada para o Problema da Cobertura de Faixa Restrita. Exibimo-la a seguir. Antes, introduzimos o Problema da 3-Partição.

**Problema 14.** *Seja  $W$  um conjunto de  $3k$  elementos;  $B$  um limitante; e um tamanho (inteiro positivo)  $z(w)$  para cada  $w$  em  $W$  de forma que  $\sum_{w \in W} z(w) = kB$ , e  $B/4 < z(w) < B/2$  para todo  $w$ .*

O Problema da 3-Partição consiste em determinar se é possível particionar  $W$  em conjuntos disjuntos  $W_1, \dots, W_k$  de modo que, para  $1 \leq i \leq k$ ,  $\sum_{w \in W_i} z(w) = B$ .

**Teorema 15.** *Dada uma instância  $(U, S)$  do Problema da Cobertura de Faixa Restrita, decidir  $OPT = L$  é NP-completo.*

*Demonstração.* Dada uma instância  $I$  do Problema da 3-Partição como acima, construímos uma instância  $(U_I, S_I)$  de CFR tal que a carga em toda posição  $i \in U_I$  é  $L(i) = k(B + 1) + 2$ . Assim,  $L = k(B + 1) + 2$ . Mais ainda, tomamos  $U_I = \{1, 2, \dots, m\}$ , onde  $m = 2(L + 1)$ .

Para cada  $w \in W$ , temos um sensor  $q_w$  com duração  $z(w)$ . Ademais, temos  $L$  sensores  $f_1, f_2, \dots, f_L$ ;  $L - 1$  sensores  $g_2, g_3, \dots, g_L$ ;  $L$  sensores  $h_1, h_2, \dots, h_L$ ; e  $L - 1$  sensores  $t_1, t_2, \dots, t_{L-1}$ . Todos os sensores  $f_i, g_i$  e  $h_i$  possuem duração 1; e todos os sensores  $t_i$  possuem duração 2.

Agora definimos a região de cobertura de cada um desses sensores. Determinamos

- $R(q_w) = \{m\}$  para todo  $w \in W$ ;
- $R(f_1) = R(f_2) = \{1\}$  e  $R(f_i) = [1, 2i - 3]$  para  $i = 3, \dots, L$ ;
- $R(g_i) = \{2i - 1\}$  para  $i = 2, \dots, L - 1$ , e  $R(g_L) = \{m - 3, m - 2\}$ ;
- $R(t_i) = \{2i\}$  para  $i = 1, \dots, L - 1$ ; e

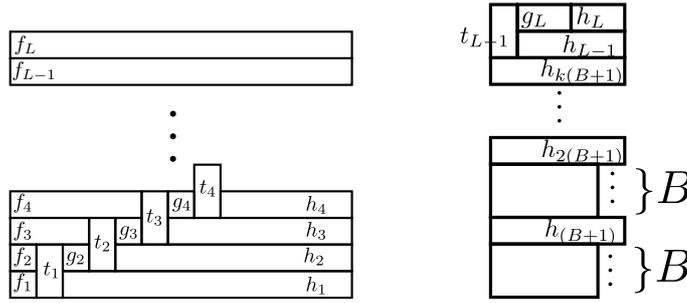


Figura 2.4: As partes inicial e final de um agendamento ótimo de  $(U_I, S_I)$

- $R(h_i) = [2i + 1, m]$  se  $i$  é múltiplo de  $B + 1$  ou  $i > L - 2$ ; e
- $R(h_i) = [2i + 1, m - 1]$  nos casos remanescentes.

Esta redução, claramente polinomial, forma uma instância de CFR como na Figura 2.4. Note que na posição  $m$  há  $k$  blocos livres de tamanho  $B$ . Argumentamos que  $I$  possui resposta “sim” se e somente se  $OPT(U_I, S_I) = L$  para a instância  $(U_I, S_I)$  correspondente do Problema da Cobertura de Faixa Restrita.

De fato, se  $I$  possui resposta “sim”, podemos agrupar os três sensores  $q_w$  referentes a cada  $w \in W_i$ , para  $i \in \{1, \dots, k\}$ , de modo a preencher os  $k$  blocos livres. Assim,  $OPT(U_I, S_I) = L$ .

Por outro lado, lembramos que  $L(m) = L$ . Se  $OPT(U_I, S_I) = L$ , então os sensores  $q_w$  se encaixam nos  $k$  blocos livres de tamanho  $B$ , determinando uma solução para  $I$ .  $\square$

Temos então o seguinte limiar de aproximabilidade para o CFR.

**Teorema 16.** *Não é possível aproximar Problema da Cobertura de Faixa Restrita com fator melhor que  $L/(L - 1)$ , a menos que  $P = NP$ .*

*Demonstração.* Suponha que CFR pode ser aproximado com fator  $\alpha$ , onde  $\alpha < \frac{L}{L-1}$ . Isto é, suponha que temos algoritmo  $A$  tal que  $A(U, S) \geq \frac{1}{\alpha}OPT(U, S) > (\frac{L-1}{L})OPT(U, S)$ , para qualquer instância  $(U, S)$ .

Considere uma instância  $I$  do Problema da 3-Partição e seja  $(U_I, S_I)$  a instância de CFR obtida usando a construção descrita anteriormente.

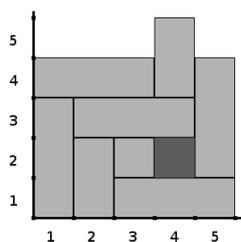
Se  $I$  tem resposta “sim”, então  $OPT(U_I, S_I) = L$ . Assim, a saída do algoritmo  $A$  seria um agendamento com duração  $A(U_I, S_I) > (\frac{L-1}{L})L = L - 1$ , o que implica  $A(U_I, S_I) = L$ .

Se  $I$  tem resposta “não”, então  $OPT(U_I, S_I) < L$ . Neste caso, a saída do algoritmo  $A$  seria um agendamento de duração  $A(U_I, S_I) \leq OPT(U_I, S_I) < L$ .

Assim, usando a redução descrita na demonstração do Teorema 15 e o algoritmo  $A$ , poderíamos decidir Problema da 3-Partição em tempo polinomial, uma contradição, assumindo  $P \neq NP$ .  $\square$

Além disso, todos os algoritmos propostos para o problema usam  $L$  como limitante para  $OPT$ . A Figura 2.5, extraída de [BEJ<sup>+</sup>07], mostra uma instância de CFR onde  $L = 4$ , mas  $OPT = 3$  (o quadrado mais escuro é um buraco). Redimensionando-a, podemos obter instâncias mantendo tal relação entre  $OPT$  e  $L$ . Temos então o seguinte.

**Lema 17.** *Não é possível aproximar Problema da Cobertura de Faixa Restrita com fator melhor que  $4/3$ , se o parâmetro  $L$  for usado como limitante para  $OPT$ .*



**Figura 2.5:** Instância do CFR com  $L = 4$  e  $OPT = 3$

*Demonstração.* Denotaremos por  $L(U, S)$  a carga  $L$  de dada instância  $(U, S)$  de CFR. Suponha que existe algoritmo  $A$  para CFR tal que  $A(U, S) \geq \frac{1}{\alpha}L(U, S)$ , onde  $\alpha < 4/3$ . Isto é,  $A(U, S) > \frac{3}{4}L(U, S)$ , para toda instância  $(U, S)$ .

Contudo, pela Figura 2.5, sabemos que há instâncias onde  $L(U, S) \geq \frac{4}{3}OPT(U, S)$ . Assim,  $A(U, S) > OPT(U, S)$  para tais instâncias, uma contradição.  $\square$

## Capítulo 3

# Caso unidimensional: duas formulações lineares inteiras

Apresentamos neste capítulo duas formulações lineares inteiras para o problema, uma abordagem ainda não tratada na literatura, e exibimos os resultados computacionais obtidos.

Inicialmente, introduzimos os conjuntos de índices e variáveis comuns às duas formulações. Para facilitar a identificação dos índices das variáveis, consideramos que a instância dada é da forma  $(\mathcal{I}, \mathcal{S})$ , onde  $\mathcal{I} = \{1, 2, \dots, m\}$  (corresponde ao intervalo  $U$ ) e  $\mathcal{S} = \{1, 2, \dots, n\}$  (corresponde ao conjunto dos sensores). Lembramos que para essa instância,  $L$  denota a carga total. Para especificar o tempo em que cada sensor é ligado, usamos o conjunto de índices (de tempo)  $\mathcal{J} = \{1, 2, \dots, L\}$ .

Quanto às variáveis, usamos dois conjuntos de variáveis binárias:

- o conjunto das variáveis  $y_{i,j}$ , para todo  $i \in \mathcal{I}$  e  $j \in \mathcal{J}$ , tais que  $y_{i,j} = 1$  se e só se o ponto  $i$  é coberto por algum sensor no tempo  $j$ ;
- o conjunto das variáveis  $z_{s,j}$ , para todo  $s \in \mathcal{S}$  e  $j \in \mathcal{J}$ , tais que  $z_{s,j} = 1$  se e só se o sensor  $s$  é ligado no tempo  $j$ .

### 3.1 Formulação 1

A primeira formulação, mostrada a seguir, busca maximizar a variável inteira  $M$ .

$$\begin{aligned} \max \quad & M \\ \text{s.a.} \quad & \sum_j z_{s,j} \leq 1 && \forall s \in \mathcal{S} && (3.1) \\ & y_{i,j} \leq \sum_{s:i \in R(s)} \sum_{k=j-d(s)+1}^j z_{s,k} && \forall i \in \mathcal{I}, \forall j \in \mathcal{J} && (3.2) \\ & y_{i,j+1} \leq y_{i,j} && \forall i \in \mathcal{I}, \forall j \in \mathcal{J} \setminus \{L\} && (3.3) \\ & M \leq \sum_j y_{i,j} && \forall i \in \mathcal{I} && (3.4) \\ & y_{i,j} \in \{0, 1\} && \forall i \in \mathcal{I}, \forall j \in \mathcal{J} \\ & z_{s,j} \in \{0, 1\} && \forall s \in \mathcal{S}, \forall j \in \mathcal{J} \\ & M \in \mathbb{Z} \end{aligned}$$

As restrições (3.1) expressam que cada sensor pode ser ligado somente uma vez; as restrições (3.2)

declaram que um ponto  $i$  somente é coberto num tempo  $j$  se há algum sensor ligado naquele instante; as restrições (3.3) asseguram que um ponto  $i$  somente é coberto num tempo  $j + 1$  se o mesmo ponto é coberto no tempo  $j$ ; e, finalmente, as restrições (3.4) garantem que  $M$  é o mínimo, se considerados todos os pontos, do tempo total em que cada posição é coberta.

## 3.2 Formulação 2

A segunda formulação, mostrada a seguir, busca minimizar a soma das sobreposições em qualquer posição  $i$  e em qualquer tempo  $j$ . A variável inteira  $x_{i,j}$  conta o número de sensores que cobrem a posição  $i$  no tempo  $j$ .

$$\begin{aligned} \min \quad & \sum_i \sum_j (x_{ij} - y_{ij}) \\ \text{s.a.} \quad & \sum_j z_{s,j} \geq 1 \quad \forall s \in \mathcal{S} \end{aligned} \quad (3.5)$$

$$x_{ij} \geq \sum_{s:i \in R(s)} \sum_{k=j}^{j+d(s)-1} z_{sk} \quad \forall i \in \mathcal{I}, \forall j \in \mathcal{J} \quad (3.6)$$

$$x_{ij} \geq y_{ij} \quad \forall i \in \mathcal{I}, \forall j \in \mathcal{J} \quad (3.7)$$

$$y_{i,j+1} \leq y_{ij} \quad \forall i \in \mathcal{I}, \forall j \in \mathcal{J} \setminus \{L\} \quad (3.8)$$

$$x_{i,j} \in \mathbb{Z} \quad \forall i \in \mathcal{I}, \forall j \in \mathcal{J}$$

$$y_{i,j} \in \{0, 1\} \quad \forall i \in \mathcal{I}, \forall j \in \mathcal{J}$$

$$z_{s,j} \in \{0, 1\} \quad \forall s \in \mathcal{S}, \forall j \in \mathcal{J}$$

As restrições (3.5), novamente, expressam que cada sensor pode ser ligado somente uma vez; as restrições (3.6) asseguram que o número de sobreposições na posição  $i$  e no tempo  $j$  é pelo menos o número de sensores ligados naquela posição e naquele instante; as restrições (3.3) garantem que um ponto  $i$  só é coberto num tempo  $t$  se houver pelo menos um sensor ligado naquele ponto no mesmo instante; e, finalmente, as restrições (3.4) declaram que um ponto  $i$  somente é coberto num tempo  $j + 1$  se o mesmo ponto é coberto no tempo  $j$ ;

## 3.3 Testes computacionais

Mostramos agora alguns resultados computacionais obtidos com a implementação dos modelos propostos.

As instâncias foram geradas de instâncias do Problema de *Strip Packing*, um problema de minimização que consiste em empacotar retângulos em uma caixa de largura fixa e altura ilimitada. Estas foram obtidas de OR-LIBRARY (<http://people.brunel.ac.uk/~mastjjb/jeb/info.html>)

Dada uma instância do Problema de *Strip Packing* com largura  $m'$ , traduzimos cada retângulo como um sensor  $s$  em nossa instância do CFR. De modo a obter uma distribuição uniforme de carga, procedemos da seguinte maneira para estabelecer  $l(s)$ . Escolhemos  $l(s)$  aleatoriamente de maneira uniforme entre 1 e  $m := m'/2$ . Se  $r(s) > m$ , “quebramos”  $s$  em dois sensores  $s_1$  e  $s_2$ , ambos com duração  $d(s)$ , mas um com  $l(s_1) = l(s)$  e  $r(s_1) = m$  e o outro com  $l(s_2) = 1$  e  $r(s_2) = w - (m - l(s) + 1)$ , onde  $w$  é a largura do sensor original  $s$ .

$n$	$m$	$L$	$L_{\max}$	$OPT$	tempo 1	tempo 2	CFR-GV
19	10	27	59	27	0.89	0.94	24
23	10	20	60	20	0.10	0.12	17
24	10	27	52	27	0.24	0.16	25
30	7	64	110	64	0.97	2.48	55
34	7	61	109	61	2.55	3.10	60
35	7	47	118	47	1.37	1.50	46
34	15	56	191	56	15.3	7.26	52
35	15	59	185	59	1.63	11.9	58
35	15	83	180	83	32.2	37.1	76
49	30	50	181	50	13.5	35.8	44
64	30	62	164	62	94.6	69.7	58
65	30	68	179	68	161	134	63
81	45	62	179	62	191	167	48
85	45	73	166	73	498	508	72
96	45	57	309	57	119	137	51

**Tabela 3.1:** *Resultados computacionais*

A implementação foi feita usando IBM ILOG CPLEX Optimizer. A tabela 3.1 mostra o tempo (em segundos) que as formulações levaram para achar uma solução ótima para as instâncias (geradas como acima mencionado) com os valores indicados de  $n$ ,  $m$ ,  $L$  e  $L_{\max}$  e  $OPT$ , onde  $L_{\max} = \max_i L(i)$  e  $OPT$  denota, como usualmente, o valor da solução ótima. Na tabela, *tempo 1* indica o tempo que levou o programa referente à primeira formulação e *tempo 2* o tempo que levou o programa referente à segunda. Na última coluna, exibimos ainda os valores encontrados pela implementação do algoritmo CFR-GV, descrito no capítulo 2, para fins de análise da performance da aproximação. Observamos que o fator para estas instâncias variou entre 1.013 e 1.291.

Considerando que para os tempos indicados o programa achou soluções ótimas para todas as instâncias geradas (aleatoriamente), estes modelos parecem úteis quando  $nL$  não é muito grande. Como podemos ver, em geral, o tempo gasto aumenta à medida que  $nL$  aumenta, o que é natural, visto que o número de restrições é  $O(nL)$ . Além disso, podemos observar que, para a maioria das instâncias de teste, o primeiro modelo se mostrou mais rápido que o segundo. Ainda não é compreendido, contudo, em que casos o segundo modelo consome tempo menor.



## Capítulo 4

# Caso unidimensional com preempção

No contexto do Problema da Cobertura por Sensores, permitir preempção significa permitir que cada sensor seja ligado ou desligado sempre que conveniente, economizando bateria quando possível. Abordamos neste capítulo agendamentos com preempção.

Ainda não é completamente compreendida a influência da preempção no caso geral do problema. Para o caso unidimensional, Buchsbaum e outros [BEJ<sup>+</sup>07] afirmaram que um algoritmo simples baseado em fluxo máximo devolve um agendamento preemptivo ótimo em tempo polinomial para o Problema da Cobertura de Faixa Restrita, mas não exibiram um tal algoritmo <sup>1</sup>. Não conseguimos formular o caso preemptivo como problema de fluxo máximo, mas obtivemos um algoritmo simples (de complexidade melhor do que um possível algoritmo baseado em fluxo máximo). Neste capítulo apresentamos o algoritmo que desenvolvemos, que devolve um agendamento de duração  $L$  em tempo quadrático.

### 4.1 Algoritmo para o caso unidimensional com preempção

Nesta seção, descrevemos um algoritmo exato polinomial, chamado CFR-Preemptivo, para a versão preemptiva do Problema da Cobertura de Faixa Restrita. Provamos em seguida que este algoritmo produz um agendamento de duração  $L$ .

Antes de descrevermos o algoritmo, vejamos algumas definições. Seja  $(U, S)$  uma instância do Problema da Cobertura de Faixa Restrita, e  $L$  a carga total de  $(U, S)$ . Chamamos de *cobertura* de  $U$  qualquer conjunto de sensores  $C \subseteq S$  tal que, para todo  $i$  em  $U$ , temos que  $i$  pertence a  $R(s)$ , para algum sensor  $s$  em  $C$ . Para uma dada cobertura  $C$ , chamamos de *redundante* um sensor  $s$  em  $C$  tal que para todo  $i$  em  $R(s)$  existe um sensor  $s'$  em  $C$  tal que  $i \in R(s')$ . Dizemos que uma cobertura  $C$  é *minimal* se  $C$  não contém sensores redundantes. Dizemos que um subintervalo  $U'$  de  $U$  é uma *região crítica* se  $L(i) = L$ , para toda posição  $i$  em  $U'$ .

A ideia central do algoritmo CFR-Preemptivo consiste em encontrar uma cobertura minimal  $C$  (que satisfaz certas propriedades), identificar o tempo  $h$  correspondente à menor das durações dos sensores em  $C$ , e agendar todos os sensores de  $C$  deixando-os ligados pelo tempo  $h$ . A seguir, atualizar as durações dos sensores em  $C$ , e repetir o processo com a nova instância, até que não haja nenhuma cobertura minimal.

Descrevemos a seguir o algoritmo CFR-Preemptivo. Consideramos que a instância dada  $(U, S)$

---

<sup>1</sup>Indagamos esses autores a respeito, mas não obtivemos resposta.

é tal que  $U = \{1, 2, \dots, m\}$  e o conjunto  $S$  de sensores está ordenado conforme o seguinte critério: em ordem não-decrescente de  $l(\cdot)$  e, em caso de empate, em ordem não-decrescente de  $r(\cdot)$ .

---

**Algoritmo 3** CFR-Preemptivo ( $U, S$ )
 

---

```

1: Se  $L = 0$  então devolva  $\emptyset$ 
2: Senão,
3:    $C \leftarrow \emptyset$ 
4:    $i \leftarrow 1$ 
5:   Enquanto  $i \leq m$ 
6:     Seja  $s$  o último sensor em  $S$  que é efetivo em  $i$  /*  $s$  é tal que  $l(s)$  é máximo */
7:     Seja  $C' := \{s' \in C : l(s) < l(s')\}$  /* conjunto de sensores redundantes */
8:      $C \leftarrow C \setminus C' \cup \{s\}$ 
9:      $i \leftarrow r(s) + 1$ 
10:     $h \leftarrow \min_{s \in C} \{d(s)\}$  /* dizemos que  $h$  ou  $h(C)$  é a altura da cobertura  $C$  */
11:    Para todo sensor  $s$  em  $C$ ,
12:       $d(s) \leftarrow d(s) - h$ 
13:      Se  $d(s) = 0$  então  $S \leftarrow S \setminus \{s\}$ 
14:     $L \leftarrow L - h$ 
15:    Devolva  $(C, h) \cup \text{CFR-Preemptivo}(U, S)$ 

```

---

Provamos a seguir que o algoritmo produz um agendamento ótimo.

**Teorema 18.** *O algoritmo CFR-Preemptivo devolve um agendamento de duração  $L$ .*

*Demonstração.* Seja  $(U, S)$  a entrada recebida pelo algoritmo. Note que o algoritmo é recursivo e que ele pára somente quando recebe como entrada uma instância onde  $L = 0$ .

Denote por  $C_1, \dots, C_k$  as coberturas encontradas pelo algoritmo, na ordem em que foram construídas. Denote ainda por  $h(C_i)$  a altura  $h$  da cobertura  $C_i$  (veja o passo 10).

Note que  $L$  decresce (de  $h(C_i)$ ) cada vez que uma nova cobertura  $C_i$  é encontrada. Vamos provar que  $\sum_{j=1}^k h(C_j) = L$ , por indução em  $k$ .

Se  $k = 1$ , somente uma cobertura,  $C_1$ , é encontrada. Temos que  $h(C_1) = \min_{s \in C_1} d(s)$ . Como nenhuma outra cobertura foi encontrada, então existe uma posição  $i$  tal que somente um sensor  $s$ , de duração  $h(C_1)$ , é efetivo em  $i$ . Neste caso,  $L \leq L(i) = h(C_1)$ . Segue que  $h(C_1) = L$ , pois  $h(C_1) \leq L$ .

Suponha agora que  $k > 1$ . Considere a primeira iteração, que encontra a cobertura  $C_1$ . Seja  $h := h(C_1)$ . Vamos provar que a nova instância  $(U, S')$ , obtida após atualizar  $S$  (linhas 11 a 13 do algoritmo) possui carga  $L'$  igual a  $L - h$ . Para isso, é suficiente mostrarmos as duas propriedades seguintes (da cobertura  $C_1$ ):

- (1) Não há sobreposição de sensores de  $C_1$  em regiões críticas; e
- (2) Se  $i$  é uma posição em que ocorre sobreposição de sensores de  $C_1$ , então vale que  $L(i) \geq L + h$ .

Note que a remoção de sensores feita no passo 8 do algoritmo assegura que as coberturas encontradas são minimais (ou seja, não têm sensores redundantes). Portanto, cada posição é coberta por no máximo dois sensores de cada cobertura.

Para mostrar (1), suponha que há sobreposição numa posição  $i$ , onde  $i$  pertence a uma região crítica. Sejam  $s_1$  e  $s_2$  os dois sensores que se sobrepõem em  $i$ , onde  $l(s_1) < l(s_2)$ . Seja  $i' \in R(s_2)$  a

primeira posição (menor inteiro) onde não há sobreposição de sensores de  $C_1$ , isto é,  $s_2$  é o único em  $C_1$  a cobrir  $i'$  (deve haver pelo menos uma tal posição, pois sensores redundantes são removidos).

Pela escolha de  $s_2$ , todo sensor de  $S$  efetivo em  $i'$  é também efetivo em  $i$ . Assim,  $L(i) \geq L(i') + d(s_1) > L$ , onde a última desigualdade segue pelo fato de  $d(s_1)$  ser não-nulo. Pela definição de região crítica, porém,  $L(i) = L$ . Chegamos assim a uma contradição, donde concluímos a prova da propriedade (1).

Vamos agora provar (2), usando raciocínio similar ao que apresentamos para a propriedade (1). Seja  $i$  uma posição em que há sobreposição. Por (1), sabemos que  $i$  não pertence a uma região crítica. Sejam  $s_1$  e  $s_2$  sensores de  $C_1$  que se sobrepõem em  $i$ , onde  $l(s_1) < l(s_2)$ ; e seja  $i' \in R(s_2)$  a primeira posição em que não há sobreposição de sensores de  $C_1$ .

Pela escolha de  $s_2$ , todo sensor de  $S$  efetivo em  $i'$  é também efetivo em  $i$ . Logo, temos que

$$\begin{aligned} L(i) &\geq L(i') + d(s_1) \\ &\geq L(i') + h \quad (\text{pois } h(C_1) = \min_{s \in C_1} d(s)) \\ &\geq L + h \quad (\text{pois } L(i') \geq L). \end{aligned}$$

Com isso, provamos que  $C_1$  tem as propriedades (1) e (2). Essas propriedades garantem que, após o agendamento dos sensores da cobertura  $C_1$ , a nova instância  $(U, S')$ , obtida após remover de  $S$  os sensores da cobertura  $C_1$ , e atualizar as durações dos sensores em  $C_1$ , tem carga total  $L' = L - h$ . Pela hipótese de indução,  $\sum_{j=2}^k h(C_j) = L - h$ . Assim, temos que  $\sum_{j=1}^k h(C_j) = L$ , e portanto o algoritmo devolve um agendamento cuja carga total é  $L$ .  $\square$

Do teorema anterior e da análise do algoritmo CFR-Preemptivo temos o seguinte resultado.

**Corolário 19.** *O algoritmo CFR-Preemptivo encontra solução exata em tempo polinomial para o caso preemptivo do Problema da Cobertura de Faixa Restrita.*

*Demonstração.* A ordenação prévia dos  $n$  sensores em  $S$  requer tempo  $O(n \log n)$ . A cada iteração (em que uma nova cobertura é encontrada), o algoritmo remove pelo menos um sensor (aquele tal que a duração é mínima dentre todos da cobertura). Então, há no máximo  $n$  chamadas do algoritmo.

Note que a construção de uma cobertura leva tempo  $O(n)$ , pois o conjunto ordenado dos (no máximo  $n$ ) sensores é percorrido no máximo uma vez, e cada sensor só pode ser selecionado uma vez. Logo, cada iteração consome tempo  $O(n)$ . Portanto, o algoritmo encontra um agendamento ótimo em tempo  $O(n^2)$ .

Pelo Teorema 18, o algoritmo devolve um agendamento de duração  $L$ . Como  $L$  é um limitante superior para  $OPT$ , concluímos que o algoritmo obtém uma solução ótima em tempo quadrático.  $\square$



# Capítulo 5

## Caso geral

Abordamos neste capítulo o caso geral do Problema da Cobertura por Sensores, onde cada  $R(\cdot)$  é um subconjunto arbitrário de um conjunto finito  $U$  de pontos tal que  $|U| = O(n)$ .

Para esta versão do problema, Buchsbaum e outros [BEJ<sup>+</sup>07] propuseram um algoritmo probabilístico que garante fator de aproximação  $O(\ln n)$ , apresentado na seção 5.1.

Por outro lado, foi mostrado que o é NP-difícil aproximar o caso geral com fator melhor que  $\ln n$ . Apresentamos este resultado na Seção 5.2.

### 5.1 Algoritmo de aproximação

Exibimos abaixo o algoritmo probabilístico proposto por Buchsbaum e outros [BEJ<sup>+</sup>07]. O algoritmo recebe como entrada um conjunto  $S$  de  $n$  sensores e uma região  $U$ , com  $|U| = O(n)$  a ser coberta, assim como no caso unidimensional, e devolve um agendamento válido de duração  $T = \lfloor cL/\ln n \rfloor$  com alta probabilidade, conforme demonstraremos no Teorema 21.

No algoritmo,  $c$  é uma constante menor que  $1/16$ , fato que será justificado na análise do Teorema 21.

---

**Algoritmo 4** CS-BEJVY ( $U, S$ )

---

- 1:  $T \leftarrow \lfloor cL/\ln n \rfloor$
  - 2: Para cada sensor  $s$  em  $S$
  - 3:     Se  $d(s) \geq T$
  - 4:          $t(s) \leftarrow 0$
  - 5:     Senão
  - 6:          $t(s) \leftarrow \text{rand}(-d(s), T)$ .
- 

O algoritmo basicamente escolhe os tempos de início dos sensores aleatoriamente, de modo uniforme, entre 0 e  $T$ . Para evitar “desvios de borda” (não uniformidade na distribuição da probabilidade entre as posições), para cada sensor  $s$  o seu tempo de início (um número inteiro) é sorteado uniformemente entre  $-d(s)$  e  $T$ . Assim,  $\text{rand}(-d(s), T)$  é uma função que devolve um inteiro no intervalo  $[-d(s), T]$ .

Usaremos na prova do Teorema 21 o seguinte fato.

**Fato 20.** *Se  $p$  é um número real tal que  $0 \leq p \leq 1$ , então  $1 - p \leq e^{-p}$ .*

**Teorema 21** (Buchsbaum e outros, 2007). *O algoritmo CS-BEJVY devolve um agendamento válido de duração  $T = \lfloor cL/\ln n \rfloor$  com alta probabilidade.*

*Demonstração.* Consideraremos aqui que  $d(s) \leq T$ , para todo sensor  $s$ . Podemos supor isso sem perda de generalidade, pois a saída do algoritmo será a mesma se modificarmos a instância de entrada fazendo  $d(s) = T$  para todo  $s$  em  $S$  tal que  $d(s) > T$ . Suporemos também que  $n \geq 2$ .

Considere o intervalo  $[0, T]$  dividido igualmente em  $2n$  subintervalos de tempo:  $[t_0 = 0, t_1]$ ,  $[t_1, t_2]$ ,  $\dots$ ,  $[t_{2n-1}, t_{2n} = T]$ , cada um com duração  $T/2n$ . Sem perda de generalidade, suponha que  $T/2n$  é um inteiro.

Dado um sensor  $s$  tal que  $d(s) \geq T/n$ , temos que, para qualquer  $i$  em  $R(s)$  e em qualquer um dos subintervalos de tempo,  $i$  é coberto por  $s$  nesse subintervalo com probabilidade pelo menos  $(d(s) - T/(2n))/(T + d(s))$ . Isto se deve ao fato de que o tamanho do espaço amostral é  $(T + d(s))$  e no máximo  $T/2n$  da duração da bateria de  $s$  não cobre um subintervalo completo.

Mais ainda,

$$\frac{d(s) - T/(2n)}{T + d(s)} \geq \frac{d(s)}{2(T + d(s))} \geq \frac{d(s)}{4T}.$$

A primeira desigualdade segue usando o fato de que  $T/n \leq d(s)$ . A segunda desigualdade vale pois  $d(s) \leq T$ . Logo, dado um sensor  $s$  tal que  $d(s) \geq T/n$ , temos que para qualquer  $i$  em  $R(s)$  e em qualquer subintervalo de tempo,  $i$  é coberto por  $s$  (neste subintervalo) com probabilidade pelo menos  $d(s)/(4T)$ .

No que segue, para simplificar a notação, se  $\hat{S}$  é um conjunto de sensores, então escrevemos  $d(\hat{S})$  para denotar a soma  $\sum_{s \in \hat{S}} d(s)$ .

Considere agora uma posição  $i$  em  $U$ , e seja  $S_i$  o conjunto dos sensores efetivos em  $i$ . Sabemos que

$$L \leq d(S_i).$$

Sejam  $S_{i1} := \{s \in S_i : d(s) \geq T/n\}$  e  $S_{i2} := S_i \setminus S_{i1}$ . Temos

$$d(S_{i2}) \leq \left(\frac{T}{n}\right)n = T.$$

Logo,

$$\begin{aligned} d(S_{i1}) + d(S_{i2}) &\geq L \\ d(S_{i1}) + T &\geq L \\ d(S_{i1}) &\geq L - T \\ d(S_{i1}) &\geq \frac{L}{2}. \end{aligned}$$

Assim, a soma das durações dos sensores efetivos em  $i$  com carga de bateria maior ou igual a  $T/n$  é pelo menos  $L/2$ .

Seja  $S_{i1} = \{s_1, \dots, s_k\}$ . Vamos determinar a seguir um limitante para a probabilidade de que, para um dado subintervalo de tempo  $[t_x, t_{x+1}]$ , a posição  $i$  não seja coberta por nenhum dos sensores em  $S_{i1}$ .

$$\begin{aligned}
\prod_{j=1}^k \left(1 - \frac{d(s_j)}{4T}\right) &\leq \prod_{j=1}^k \exp\left(-\frac{d(s_j)}{4T}\right) \\
&= \exp\left(-\frac{d(S_i)}{4T}\right) \\
&\leq \exp\left(-\frac{L}{8T}\right) \\
&\leq \exp\left(-\frac{\ln n}{8c}\right) \\
&= n^{-1/8c}.
\end{aligned}$$

Como,  $|U| = O(n)$ , então há somente  $O(n^2)$  pares  $(i, [t_x, t_{x+1}])$  diferentes. Logo, a probabilidade de que alguma posição  $i$  em  $U$  não seja coberta em algum subintervalo é, no máximo,

$$\begin{aligned}
O(n^2) n^{-1/8c} &= O(n^{2-1/8c}) \\
&= O(n^{-\epsilon}),
\end{aligned}$$

para algum  $\epsilon > 0$ , pois  $c < 1/16$ . Logo, com alta probabilidade o algoritmo devolve um agendamento válido de duração  $T = \lfloor cL/\ln n \rfloor$ . □

O algoritmo passa pelo laço uma vez para cada sensor no conjunto de entrada e, portanto, executa em tempo polinomial. Deste fato, e do Teorema 21, temos o resultado a seguir.

**Teorema 22.** *O algoritmo CS-BEJVY é uma  $O(\ln n)$ -aproximação polinomial probabilística para o Problema da Cobertura por Sensores.*

## 5.2 Complexidade

Feige e outros [FHK00] provaram um limiar de inaproximabilidade de  $\ln n$  para o Problema do Empacotamento de Coberturas por Conjuntos, onde  $n = |U|$ , sob a hipótese de que NP não possui algoritmo que execute em tempo  $n^{O(\log \log n)}$ . Contudo, conforme argumentado na Seção 1.2, Empacotamento de Coberturas por Conjuntos é um caso particular do Problema da Cobertura por Sensores. Logo, o mesmo limiar vale para o último.

**Teorema 23.** *Não é possível aproximar o Problema da Cobertura por Sensores com fator melhor que  $\ln n$ , a menos que  $\text{NP} \subseteq \text{DTIME}(n^{O(\log \log n)})$ .*



## Capítulo 6

# Considerações finais

Neste trabalho, nosso objetivo foi apresentar um estudo sobre o estado-da-arte do Problema da Cobertura por Sensores, mostrando o que é conhecido acerca de algoritmos e aproximabilidade, para cada um de seus casos. Concentramos nossos estudos no caso unidimensional, chamado Problema de Cobertura de Faixa Restrita.

Para o caso unidimensional, mostramos que um algoritmo de Gibson e Varadarajan de 2009, apresentado como sendo uma 5-aproximação, tem na verdade fator de aproximação 4, e que esse fator é justo. Apresentamos também dois modelos de programas lineares inteiros para o caso unidimensional, uma abordagem que não encontramos na literatura. Adicionalmente, para este caso, projetamos um algoritmo exato para a versão preemptiva: trata-se de um algoritmo polinomial simples, bem fácil de ser implementado.

Por se tratar de um problema recente, ainda há muitas questões em aberto a serem investigadas. No caso unidimensional em que as baterias são de duração variável, por exemplo, o algoritmo CFR-GV é a única aproximação de fator constante conhecida para o problema. Seria interessante, se possível, obter algoritmo de aproximação com fator menor que 4. Além disso, não é conhecido limiar de aproximabilidade melhor que  $L/(L-1)$ . Também não é descartada a hipótese de que haja um PTAS para o caso unidimensional. Mostrar a existência de um ou eliminar esta possibilidade representaria um resultado significativo.

Sobre as formulações lineares inteiras que propusemos, vimos que ambas possuem número de restrições exponencial no tamanho da entrada, o que inviabiliza até mesmo o uso da relaxação como parte de um algoritmo de aproximação de tempo polinomial. Um modelo com um número polinomial de restrições poderia ser de grande utilidade neste aspecto.

Quanto ao caso preemptivo, embora no caso unidimensional o problema esteja bem resolvido, em outras dimensões não se sabe quão distantes soluções não-preemptivas podem estar da melhor solução preemptiva.



## Apêndice A

# NP-Completeness do Problema da Alocação Dinâmica de Memória

Mostramos uma demonstração da NP-Completeness do Problema da Alocação Dinâmica de Memória. A prova a seguir é devida a Larry Stockmeyer e é citada no livro de Garey e Johnson [GJ79] como “*private communication*”. A prova é exibida em [BEJ<sup>+</sup>07]. Reproduzimos aqui traduzido, para fins de documentação.

A redução para demonstrar o resultado é feita a partir do Problema da 3-Partição, introduzido na seção 2.3 do capítulo 2. Repetimos a descrição do problema a seguir, por facilidade.

**Problema 24.** *Seja  $W$  um conjunto de  $3m$  elementos;  $B$  um limitante; e um tamanho (inteiro positivo)  $z(w)$  para cada  $w$  em  $W$  de forma que  $\sum_{w \in W} z(w) = mB$ , e  $B/4 < z(w) < B/2$  para todo  $w$ .*

O Problema da 3-Partição consiste em determinar se é possível particionar  $W$  em conjuntos disjuntos  $W_1, \dots, W_m$  de modo que, para  $1 \leq i \leq m$ ,  $\sum_{w \in W_i} z(w) = B$ .

O Problema da 3-Partição é fortemente polinomial. A condição  $B/4 < z(w) < B/2$  não é usada na redução a seguir.

Dada uma instância do Problema da 3-Partição, a instância correspondente do Problema da Alocação Dinâmica de Memória possui tamanho de memória  $D = m(B + 1) + 2$ . A instância será descrita por uma sequência cronologicamente ordenada de chegadas e partidas de itens de vários tamanhos. Por conveniência, também será permitido a um item  $t$ , de tamanho 2, chegar e partir várias vezes, dado que ele parte antes de chegar novamente. Logo, se um item  $t$  chega  $k$  vezes em toda a descrição, na realidade há  $k$  itens diferentes  $t_i$ , para  $1 \leq i \leq k$ , todos de tamanho 2, e nenhum par deles podem existir ao mesmo tempo.

A seguir, os itens  $f_i$ ,  $g_i$  e  $h_i$  têm todos tamanho unitário. Comece com a chegada de  $D$  itens  $f_1, \dots, f_D$ . Em seguida, saem  $f_1$  e  $f_2$ , chega  $t$  e sai, e então chegam  $g_1$  e  $g_2$ . Agora, faça em sequência, para  $i = 2, 3, \dots, D - 1$ :

1. itens  $g_i$  e  $f_{i+1}$  partem;
2. item  $t$  chega e parte; e
3. itens  $h_i$  e  $g_{i+1}$  chegam.

Finalmente,  $g_1$  parte e  $h_1$  chega, e então  $g_D$  parte e  $h_D$  chega. Neste ponto, a ordem do itens na memória deve ser  $h_1, h_2, \dots, h_{D-2}, h_{D-1}, h_D$  ou  $h_1, h_2, \dots, h_{D-2}, h_D, h_{D-1}$  ou o inverso de alguma destas ordens.

Parte então todo  $h_i$ , para  $i \leq D - 2$  que não é múltiplo de  $B + 1$ . Agora, a memória consiste de  $m$  blocos de espaço livre, cada um com tamanho  $B$  e há barreiras  $(h_{B+1}, h_{2B+2}, \dots)$  entre cada par de blocos adjacentes.

Considere o caso em que o tamanho dos itens não está restrito a 1 ou 2. Para cada  $w \in W$ , um item  $q_w$  de tamanho  $z(w)$  chega. Se a instância do Problema da 3-Partição tem uma solução  $W_1, \dots, W_m$ , então os itens  $q_w$  com  $w \in W_1$  podem ficar no primeiro bloco de espaço livre, os itens  $q_w$  com  $w \in W_2$  pode ficar no segundo bloco de espaço livre, e assim por diante, para  $W_3, \dots, W_m$ .

Por outro lado, se todos  $q_w$  encaixam nos  $m$  blocos de espaço livre de tamanho  $B$ , então deve haver uma solução para a instância do Problema da 3-Partição.

Esta é uma transformação de tempo polinomial, pois os números  $z(w)$  estão limitados superiormente por um polinômio no tamanho da instância do Problema da 3-Partição. Note ainda que a redução constrói uma instância do Problema da Alocação Dinâmica de Memória de carga uniforme com a questão final sendo se  $OPT = L$ .

# Referências Bibliográficas

- [AGP04] Z. Abrams, A. Goel e S. Plotkin. Set k-cover algorithms for energy efficient monitoring in wireless sensor networks. Em *Proceedings of the 3rd international symposium on Information processing in sensor networks*, página 432. ACM, 2004. 4
- [BEJ<sup>+</sup>07] A. Buchsbaum, A. Efrat, S. Jain, S. Venkatasubramanian e K. Yi. Restricted strip covering and the sensor cover problem. Em *Proceedings of the 18th Annual ACM Symposium on Discrete Algorithms*, páginas 1056–1065, New York, 2007. ACM. 3, 4, 5, 7, 8, 15, 21, 25, 31
- [BKK<sup>+</sup>04] A. Buchsbaum, H. Karloff, C. Kenyon, N. Reingold e M. Thorup. OPT versus LOAD in dynamic storage allocation. *SIAM Journal on Computing*, 33(3):632–646, 2004. 5
- [FHK00] U. Feige, M. M. Halldórsson e G. Kortsarz. Approximating the domatic number. Em *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing*, páginas 134–143 (electronic), New York, 2000. ACM. 4, 27
- [Ger96] J. Gergov. Approximation algorithms for dynamic storage allocations. Em *Proceedings of the 4th Annual European Symposium on Algorithms*, páginas 52–61. Springer-Verlag, 1996. 5
- [Ger99] J. Gergov. Algorithms for compile-time memory optimization. Em *Proceedings of the 10th Annual ACM Symposium on Discrete Algorithms*, página 908. Society for Industrial and Applied Mathematics, 1999. 5
- [GJ79] M.R. Garey e D.S. Johnson. *Computers and intractability. A guide to the theory of NP-completeness. A Series of Books in the Mathematical Sciences*. WH Freeman and Company, San Francisco, Calif, 1979. 4, 31
- [GV09] M. Gibson e K. Varadarajan. Decomposing coverings and the planar sensor cover problem. Em *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science*, páginas 159–168. IEEE, 2009. 5, 8, 9
- [Kie88] H. Kierstead. The linearity of first-fit coloring of interval graphs. *SIAM Journal on Discrete Mathematics*, 1:526, 1988. 4
- [Kie91] H. Kierstead. A polynomial-time approximation algorithm for dynamic storage allocation. *Discrete Mathematics*, 88(2-3):231–237, 1991. 4
- [SP01] S. Slijepcevic e M. Potkonjak. Power efficient organization of wireless sensor networks. Em *Proceedings of the IEEE International Conference on Communications*, páginas 472–476, 2001. 3
- [WX06] L. Wang e Y. Xiao. A survey of energy-efficient scheduling mechanisms in sensor networks. *Mobile Networks and Applications*, 11(5):740, 2006. 1