AIMMS Modeling Guide - Bandwidth Allocation Problem

This file contains only one chapter of the book. For a free download of the complete book in pdf format, please visit www.aimms.com or order your hard-copy at www.lulu.com/aimms.

Paragon Decision Technology B.V.
Schipholweg 1
2034 LS Haarlem
The Netherlands
Tel.: +31 23 5511512
Fax: +31 23 5511517

Paragon Decision Technology Inc.
500 108th Avenue NE
Ste. # 1085
Bellevue, WA 98004
USA
Tel.: +1 425 458 4024
Fax: +1 425 458 4025

Paragon Decision Technology Pte. Ltd.
80 Raffles Place
UOB Plaza 1, Level 36-01
Singapore 048624
Tel.: +65 9640 4182

Email: info@aimms.com
WWW: www.aimms.com

This documentation was typeset by Paragon Decision Technology B.V. using LATEX and the LUCIDA font family.

Chapter 15

A Bandwidth Allocation Problem

This chapter

This chapter introduces a bandwidth allocation problem and presents two different ways to formulate a binary programming model of it. The incentive to develop the second model formulation arose when the implementation of the first model became unwieldy. For both models, techniques for reducing the complexity of the constraint matrix are demonstrated. Both representations can easily be implemented using the AIMMS modeling language.

Keywords

Integer Program, Mathematical Reformulation, Worked Example.

15.1 Problem description

Bandwidth planning problems

As a result of the growing number of mobile communication systems, there is an increasing need to allocate and re-allocate bandwidth for point-to-point communications. Bandwidth allocations typically remain operational for seconds/minutes (in cellular communications), days/weeks (in military communication systems) or months/years (in television and radio communication systems). During these operational periods the volume of traffic usually changes significantly, which causes point-to-point capacity and interference problems. Consequently, bandwidth allocation is a recurring process in practice. In this chapter a specific bandwidth allocation problem is examined.

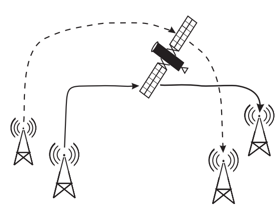

Figure 15.1: A satellite communication system

Basic terminology

Consider a satellite communication system, as shown in Figure 15.1, where the ground stations either transmit or receive messages via the satellite. A *link* in such a communication system is any pair of communicating ground stations. The *bandwidth domain* is the specific range of channels available for allocation. Such a range can be divided up into fixed-width portions, referred to as *channels*. Any specific link requires a pre-specified number of adjacent channels, which is referred to as a *bandwidth interval*. The concepts of "channel" and "bandwidth interval" are illustrated in Figure 15.2. *Link interference* represents a combined measure of the transmitter and receiver interference as caused by other existing communications systems.

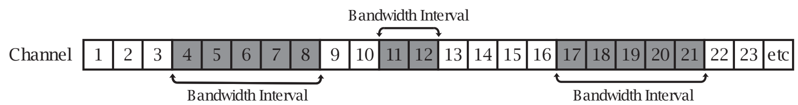

Figure 15.2: Channels and intervals in a bandwidth domain

The problem summarized

A *bandwidth allocation* is the assignment of a bandwidth interval to at most one link in the communication system. An *optimal bandwidth allocation* is one in which some measure of total interference is minimized.

Constructing link interference

For a given link, the overall level of transmitter and receiver interference is dependent on the interference over its entire bandwidth interval. The model formulation in this chapter assumes that interference data is available on a per channel basis for each link. Furthermore, for each interval-link combination it is assumed that the overall interference experienced by the link is equal to the value of the maximum channel interference that is found in the interval.

Running example

This chapter illustrates the bandwidth allocation problem using a small example data set consisting of three communication links with seven adjacent channels available for transmission. The first link requires one channel for transmission, while both the remaining two links must be allocated a bandwidth interval containing three channels. Table 15.1 presents the interference level for each link on a per channel basis. Using this data, the overall interference of each interval-link is found by identifying the maximum channel interference in the corresponding bandwidth interval. These values are presented in Table 15.1.

Model formulation

Based on the simple verbal description of the problem, there is the initial expectation that it should be straightforward to formulate a model of the problem . However, for large instances of the problem this is not the case. This chapter presents two model formulations. The first formulation relies on enumerating all possible intervals and can have implementation difficulties. The second formulation presents a better symbolic representation which improves implementation.

	Channel-link interference			**Interval-link interference**		
	Link 1	Link 2	Link 3	Link 1	Link 2	Link 3
Channel 1	4	5	6	4	8	8
Channel 2	8	8	1	8	8	8
Channel 3	7	1	8	7	2	8
Channel 4	9	2	7	9	8	7
Channel 5	1	1	1	1	8	3
Channel 6	5	8	2	5	-	-
Channel 7	4	5	3	4	-	-

Table 15.1: Channel and interval interference data

15.2 Formulation I: enumerating bandwidth intervals

All bandwidth intervals

This formulation relies on first enumerating all possible intervals of a fixed size. Next, binary variables (yes/no choices) are defined to assign them to communication links of the same size. With seven channels, three links, and two different interval widths, there are twelve possible positioned intervals (seven of width one, and five of width three) numbered from 1-12 as shown in Figure 15.3. These positioned interval numbers are used throughout this section. It should be noted that these numbers have the disadvantage that they do not show any relationship to either the size or the channel numbers contained in the interval.

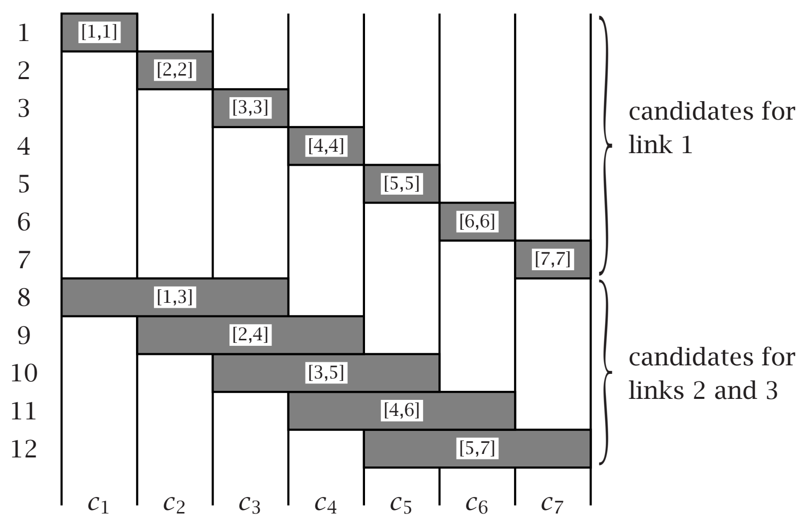

Figure 15.3: Twelve possible positioned intervals

Notation

Together with the positioned interval numbers, the following notation is used.

Indices:

p	*positioned intervals*
l	*links*

Parameter:

$$d_{pl} \quad \begin{cases} 1 & \text{if positioned interval } p \text{ has width required by link } l \\ 0 & \text{otherwise} \end{cases}$$

Variable:

$$x_{pl} \quad \begin{cases} 1 & \text{if positioned interval } p \text{ is assigned to link } l \\ 0 & \text{otherwise} \end{cases}$$

Two apparent constraints

Using the above notation and a previous introduction to the assignment model (see Chapter 5), the following two symbolic constraints can be formulated.

$$\begin{aligned} \sum_p d_{pl} x_{pl} &= 1 \qquad \forall l \\ \sum_l d_{pl} x_{pl} &\leq 1 \qquad \forall p \end{aligned}$$

The first constraint makes sure that a link l uses exactly one of the positioned intervals, and the second constraint makes sure that each positioned interval p is allocated at most once.

The constraints in tabular form

In case you are not (yet) comfortable with the above symbolic notation, Table 15.2 presents the constraints for the example problem in tabular format. The variable names are on top, and the coefficients are written beneath them. At this point there are 15 individual constraints (3+12) and 17 individual variables (7+5+5).

What is missing?

The formulation developed so far is incomplete. It is missing a mechanism that will ensure that selected positioned intervals do not overlap. Without such a mechanism the variables $x_{5,1}$ and $x_{11,2}$ can both be equal to 1, while their corresponding positioned intervals [5,5] and [4,6] overlap.

Two approaches to avoid overlap

To handle this situation, there are at least two approaches. One approach is to build constraints that restrict allocations so there is no overlap of their corresponding bandwidth intervals. The other (less obvious) approach is to define for each variable which channels are covered, and to add a constraint for each channel to limit its use to at most once. Even though these approaches sound distinct, they eventually lead to the same model formulations. This equivalence is shown in the next two subsections.

x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		
p	1	2	3	4	5	6	7	8	9	10	11	12	8	9	10	11	12		
l	1	1	1	1	1	1	1	2	2	2	2	2	3	3	3	3	3		
1	1	1	1	1	1	1	1											=	1
2								1	1	1	1	1						=	1
3													1	1	1	1	1	=	1
1	1																	≤	1
2		1																≤	1
3			1															≤	1
4				1														≤	1
5					1													≤	1
6						1												≤	1
7							1											≤	1
8								1					1					≤	1
9									1					1				≤	1
10										1					1			≤	1
11											1					1		≤	1
12												1					1	≤	1

Table 15.2: The individual assignment constraints

15.2.1 Preventing overlap using pairs of allocations

Consider pairs of allocations

This approach relies on building constraints that prevent allocations with overlap. One way to accomplish this is to first identify all pairs of positioned intervals that overlap, and then to write a constraint for each pair to prevent the overlap. This approach is a form of enumeration. For our example, there are at most $\binom{17}{2} = (17^2 - 17)/2 = 136$ pairs of associated decision variables must be considered to form constraints that restrict overlap. An analysis concludes there are only 63 pairs of overlapping intervals.

Constraint reduction

Fortunately, the 63 constraints identified to avoid overlap can be combined to form a much smaller set of equivalent constraints. To illustrate how, consider the following three restrictions.

$$\begin{array}{lllllll} x_{3,1} & + & x_{9,2} & & & \leq & 1 \\ x_{3,1} & & & + & x_{10,3} & \leq & 1 \\ & & x_{9,2} & + & x_{10,3} & \leq & 1 \end{array}$$

The three positioned intervals represented by numbers 3, 9 and 10 correspond to bandwidth intervals [3,3], [2,4] and [3,5] respectively. Since all three intervals include channel 3, it is possible to add the three constraints together to obtain the following single constraint.

$$2x_{3,1} \quad + \quad 2x_{9,2} \quad + \quad 2x_{10,3} \quad \leq \quad 3$$

Since all of the variables are either zero or one, this constraint can be rewritten after dividing by 2 and then rounding the right-hand side downwards.

$$x_{3,1} \quad + \quad x_{9,2} \quad + \quad x_{10,3} \quad \leq \quad 1$$

This constraint replaces the previous three constraints, and allows for exactly the same *integer solutions*. When viewed in terms of *continuous variables*, this single constraint represents a much tighter formulation. Consider for instance the point (0.5, 0.5, 0.5). This is a feasible point for the first three constraints, but not for the combined constraint. In general, tighter constraints are preferred, because they help the solution algorithm to converge faster.

New constraint is tighter

The reduction scheme can be applied to groups of allocations where their bandwidth intervals have pairwise overlap. The process becomes somewhat more involved when the size of a group of allocations with pairwise overlap increases. The step of extracting the 63 overlapping intervals from all possible intervals takes time, but the subsequent step to reduce the constraint set to just 7 new constraints takes even longer. The new constraints are numbered 1-7, and listed in tabular format in Table 15.3.

Reduced constraint set

x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		
p	1	2	3	4	5	6	7	8	9	10	11	12	8	9	10	11	12		
l	1	1	1	1	1	1	1	2	2	2	2	2	3	3	3	3	3		
1	1							1					1					$\leq$	1
2		1						1	1				1	1				$\leq$	1
3			1					1	1	1			1	1	1			$\leq$	1
4				1					1	1	1			1	1	1		$\leq$	1
5					1					1	1	1			1	1	1	$\leq$	1
6						1					1	1				1	1	$\leq$	1
7							1					1					1	$\leq$	1

Table 15.3: The individual constraints on overlap

15.2.2 Preventing overlap using channel constraints

This approach generates the same seven constraints that resulted from the first approach but they are generated directly. The key observation is that there are seven channels, and that the seven constraints from the first approach actually represent a constraint for each of the seven channels.

Inspect results of first approach

Let constraint c be associated with the set of channels C ($C = \{1, 2, \ldots, 7\}$). Then the coefficients in each constraint c correspond exactly to those variables with positioned intervals that overlap with channel c. Using this new viewpoint, it is possible to formulate one constraint for each channel directly.

Overlap with channels

The 'cover' matrix

Let c refer to channels and define the three-dimensional parameter a_{cpl} as the 'cover' matrix (c refers to rows and pl refers to columns). Specifically,

$$a_{cpl} \quad \begin{cases} 1 & \text{if the (positioned interval, link) pair } pl \text{ contains channel } c \\ 0 & \text{otherwise} \end{cases}$$

The new constraint

With this notation you can write the following symbolic constraint to ensure that intervals do not overlap. The individual constraints that make up this symbolic constraint are exactly the same as those in Table 15.3 from the first approach.

$$\sum_{(pl)} a_{cpl} x_{pl} \leq 1 \qquad \forall c$$

15.3 Formulation II: avoiding bandwidth interval construction

This section

In this section the model is reformulated using different notation. The new formulation avoids the process of positioned interval construction used in Section 15.2.1.

Revise variable definition

In the previous formulation all possible bandwidth intervals were enumerated, which required both a construction process and a matching with links. This entire process can be avoided by letting the variables refer indirectly to the set of enumerated intervals. To do this, variables are defined with both channel and link indices. The channel index represents the channel number where the link bandwidth interval *begins*. The end of the interval is calculated from the known length of the link.

Notation

Parameter:

$$a_{\hat{c}cl} \quad \begin{cases} 1 & \text{if the interval for link } l \text{ starting at channel } c \text{ also covers channel } \hat{c} \\ 0 & \text{otherwise} \end{cases}$$

Variable:

$$x_{cl} \quad \begin{cases} 1 & \text{if the interval for link } l \text{ starts at channel } c \\ 0 & \text{otherwise} \end{cases}$$

Model constraints

The following model constraints can now be written.

$$\begin{aligned} \sum_{c} x_{cl} &= 1 \quad \forall l \in L \\ \sum_{cl} a_{\hat{c}cl} x_{cl} &\leq 1 \quad \forall \hat{c} \in C \end{aligned}$$

The first constraint makes sure that for each link there is exactly one channel at which its corresponding bandwidth interval can start. The second constraint makes sure that for each channel at most one of all overlapping intervals can be allocated.

Be aware of domain restrictions

Note that the variable references in the above symbolic constraints do not reflect their domain of definition. For instance, the variable $x_{6,2}$ is not defined, because a bandwidth interval of length 3 for link 2 cannot start at channel 6. These domain restrictions, while not explicit in the above symbolic model, must be clearly specified in the actual model formulation. This can be easily implemented using the AIMMS modeling language.

The objective function

As described at the beginning of this chapter, the objective of the bandwidth allocation problem is to minimize a specific measure of total communication interference. To this end, the following notation is defined.

g_{cl}	*interference for link l whenever channel c is part of the interval allocated to this link*
w_{cl}	*maximum interference for link l whenever its interval starts at channel c*
J_{cl}	*collection of channels that comprise the particular interval for link l beginning at channel c*

The interference assigned to a link is the maximum channel interference w_{cl} that occurs in the bandwidth interval.

$$w_{cl} = \max_{\hat{c} \in J_{cl}} g_{\hat{c}l} \qquad \forall (c, l)$$

The entire formulation

The model formulation with objective function and constraints is summarized below.

$$\begin{array}{lll} \min & \sum_{cl} w_{cl} x_{cl} & \\ \text{s.t.} & \sum_c x_{cl} = 1 & \forall l \\ & \sum_{cl} a_{\hat{c}cl} x_{cl} \leq 1 & \forall \hat{c} \in C \\ & x_{cl} \text{ binary} & \end{array}$$

Domain checking

In the algebraic description of the model, no attention has been paid to restrictions on the index domain of any of the identifiers. The domain conditions on the decision variables, for instance, should make sure that no bandwidth interval extends beyond the last available channel of the bandwidth domain.

In the optimal solution for the problem instance described in Section 15.1, link 1 is assigned to channel 1, link 2 is assigned to channels 2 through 4, and link 3 is assigned to channels 5 through 7. The corresponding total interference is 15.

Solution

15.3.1 Improving sparsity in overlap constraints

After analyzing the coefficient matrix associated with the individual overlap constraints, some simple mathematical manipulations can be performed to obtain a potentially significant reduction in the number of nonzero coefficients.

Analyzing the coefficient matrix

Before the constraint matrix is manipulated, it is necessary to add a zero-one slack variable to each overlap constraint. For the worked example case, this is illustrated in Table 15.4.

Add slacks to overlap constraints

	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	s	s	s	s	s	s	s		
c	1	2	3	4	5	6	7	8	9	10	11	12	8	9	10	11	12	1	2	3	4	5	6	7		
l	1	1	1	1	1	1	1	2	2	2	2	2	3	3	3	3	3									
1	1							1					1					1							=	1
2		1						1	1				1	1					1						=	1
3			1					1	1	1			1	1	1					1					=	1
4				1					1	1	1			1	1	1					1				=	1
5					1					1	1	1			1	1	1					1			=	1
6						1					1	1				1	1						1		=	1
7							1					1					1							1	=	1

Table 15.4: Constraints on overlap as *equalities*

Consider rows i, $i \geq 2$ and subtract from each row its previous row $i - 1$. This results is a special matrix, in which each column has at most two coefficients. This matrix is presented in Table 15.5. Note that a column has either a 1 and -1 or just a 1.

Subtract previous rows

	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	s	s	s	s	s	s	s		
c	1	2	3	4	5	6	7	8	9	10	11	12	8	9	10	11	12	1	2	3	4	5	6	7		
l	1	1	1	1	1	1	1	2	2	2	2	2	3	3	3	3	3									
1	1							1					1					1							=	1
2	-1	1							1					1				-1	1						=	0
3		-1	1							1					1				-1	1					=	0
4			-1	1				-1			1		-1			1				-1	1				=	0
5				-1	1				-1			1		-1			1				-1	1			=	0
6					-1	1				-1					-1							-1	1		=	0
7						-1	1				-1					-1							-1	1	=	0

Table 15.5: Overlap constraints with a most 2 nonzeros per column

Reduction in nonzero elements

Whenever the length of a link is greater than or equal to three, there will be a reduction in the number of coefficients for that link. The overall savings are particularly significant in those applications where the average link width is much larger than three.

15.4 Summary

In this chapter the process of formulating and then reformulating a binary programming model was demonstrated. The need to reformulate the model arose because of difficulties encountered when implementing the first. In both formulations, techniques to reduce the complexity of the constraint matrix were illustrated. The process of model evolution, as demonstrated in this chapter, is quite common in practical modeling situations. This is especially true when the underlying model is either a binary or a mixed-integer programming model. For this class of models it is often worthwhile to consider alternative formulations which are easier to work with and which may result in strongly reduced solution times.

Exercises

15.1 Implement the bandwidth allocation model according to Formulation I as presented in Section 15.2 using the data provided in Section 15.1.

15.2 Implement the bandwidth allocation model according to Formulation II as presented in Section 15.3, and verify whether the two formulations in Aimms produce the same optimal solution.

15.3 Implement the variant of Formulation II described in Section 15.3.1, in which there are at most two nonzero elements for each column in the overlap constraints. Check whether the optimal solution remains unchanged.