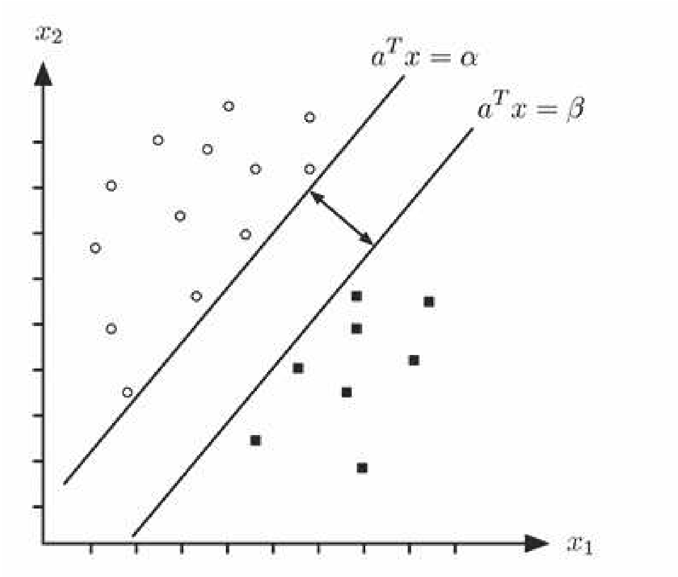


Formulate the problem of computing a, α, and β achieving the above-mentioned goals as an LP.

1.3 Integer programs

An *integer program* is obtained by taking a linear program and adding the condition that a nonempty subset of the variables be required to take integer values. When all variables are required to take integer values, the integer program is called a *pure integer program* otherwise it is called a *mixed integer program*. We will abbreviate the term integer program by IP, throughout this book.

Example 3 The following is a mixed IP, where variables x_1 and x_3 are required to take integer values:

$$
\begin{array}{lrcr}
\max & x_1 + x_2 + 2x_4 & & \\
\text{subject to} & & & \\
& x_1 + x_2 & \leq & 1 \\
& -x_2 - x_3 & \geq & -1 \\
& x_1 + x_3 & = & 1 \\
& \multicolumn{3}{l}{x_1, x_2, x_3 \geq 0 \text{ and } x_1, x_3 \text{ integer.}}
\end{array}
$$

In Section 1.1, we introduced the WaterTech production problem. We gave an LP formulation (1.5) and a solution to that formulation in (1.7). This solution told us to manufacture, 16+ $\frac{2}{3}$ units of product 1. Depending on the nature of product 1, it may not make sense to produce a fractional number of units of this product. Thus, we may want to add the condition that each of x_1, x_2, x_3, x_4 is an integer. The resulting program would be an IP. In this example, we could try to ignore the integer condition, and round down the solution, hoping to get a reasonably good approximation to the optimal solution.

1.3.1 Assignment problem

Our friends at WaterTech are once again looking to us for help. The company faces the following problem: there is a set of J jobs that need to be handled urgently. The company has selected I of its most trusted employees to handle these jobs. Naturally, the skill sets of these employees differ, and not all of the jobs are equally well handled by all of the employees. From past experience, management knows the number of hours c_{ij} each worker $i \in I$ is expected to take in order to complete any of the jobs $j \in J$. The following table gives an example for a case with $|J| = 4$ jobs and $|I| = 4$ employees:

	Jobs			
Employees	1	2	3	4
1	3	5	1	7
2	8	2	2	4
3	2	1	6	8
4	8	3	3	2

For instance, the table says that $c_{3,4} = 8$, i.e. employee 3 would take eight hours to finish job 4. WaterTech wants to assign jobs to employees with the conditions that:

(1) each employee $i \in I$ is assigned exactly one job $j \in J$,

(2) each job $j \in J$ is assigned to exactly one employee $i \in I$.

Both of these conditions can only be satisfied when $|I| = |J|$. Naturally, we want to find such an assignment that minimizes the total expected amount of time needed to process all jobs J. A feasible solution would be to assign job k to employee k, for $k = 1, 2, 3, 4$. The total amount of time required for this assignment is $3 + 2 + 6 + 2 = 13$ hours. This not an optimal solution, however. We wish to find an IP formulation for this problem. Thus, we need to determine the variables, the constraints, and the objective function.

Variables. In this case, we need to decide for each employee $i \in I$ and each job $j \in J$ whether employee i is assigned job j. We will introduce for every such pair i, j a variable x_{ij} that we restrict to take values 0 or 1, where $x_{ij} = 1$ represents the case where employee i is assigned job j, and $x_{ij} = 0$ means that employee i is not assigned job j. Thus, we have $|I||J|$ variables. In the example given by the above table we end up with 16 variables.

Constraints. We need to encode condition (1) as a mathematical constraint. Let $i \in I$ be an employee, then $\sum_{j \in J} x_{ij}$ is the number of jobs employee i is assigned to (we do the sum over all jobs). We want this quantity to be one, thus the following should hold:

$$\sum_{j \in J} x_{ij} = 1. \tag{1.13}$$

In the example given by the table, this says that $x_{i1} + x_{i2} + x_{i3} + x_{i4} = 1$ for all jobs $i \in \{1, 2, 3, 4\}$. We need to encode condition (2) as a mathematical constraint. Let $j \in J$ be a job then $\sum_{i \in I} x_{ij}$ is the number of employees job j is assigned to (we do the sum over all employees). We want this quantity to be one, thus the following should hold:

$$\sum_{i \in I} x_{ij} = 1. \tag{1.14}$$

Objective function. The objective function should calculate the total amount of time spent to complete the jobs. For every employee $i \in I$ and job $j \in J$, if employee i is assigned job j, then we should contribute c_{ij} to the objective function, otherwise we should contribute 0. Thus, we should contribute $c_{ij}x_{ij}$. Therefore, the objective function is given by

$$\sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij}. \tag{1.15}$$

For instance, in our specific example, the objective function is

$$3x_{11} + 5x_{12} + 1x_{13} + 7x_{14} + 8x_{21} + 2x_{22} + 2x_{23} + 4x_{24} + 2x_{31} + 1x_{32} + 6x_{33} + 8x_{34} + 8x_{41} + 3x_{42} + 3x_{43} + 2x_{44}.$$

Thus, the IP formulation is given by objective function (1.15), and constraints (1.13) and (1.14), as well as the condition that each variable x_{ij} can only take values 0, 1

$$\begin{array}{lll} \text{min} & \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} & \\ \text{subject to} & & \\ & \sum_{j \in J} x_{ij} = 1 & (i \in I) \\ & \sum_{i \in I} x_{ij} = 1 & (j \in J) \\ & x_{ij} \in \{0, 1\} & (i \in I, j \in J). \end{array} \tag{1.16}$$

Note, formally speaking, that (1.16) is not an IP, because of the constraints $x_{ij} \in \{0, 1\}$. However, we can clearly replace these constraints by the constraints $x_{ij} \geq 0$, $x_{ij} \leq 1$ and x_{ij} integer. If we do this for all $i \in I$ and $j \in J$, then the resulting optimization formulation is an IP. Hence, we will abuse notation slightly and call the formulation (1.16) an IP.

Solving this IP for the special case given in the table, yields

$$x_{11} = 1, x_{23} = 1, x_{32} = 1, x_{44} = 1$$

and all other values $x_{ij} = 0$. Thus, an optimal solution is to assign job 1 to employee 1, job 3 to employee 2, job 2 to employee 3, and job 4 to employee 4.

Note, in this example we represent a binary choice (whether to assign job j to employee i) by a variable taking values 0 or 1. We call such a variable a *binary variable*. When using a binary variable y, we can express $y \in \{0, 1\}$ by $0 \leq y \leq 1$ and y integer, but the condition that y is integer cannot easily be omitted, as we may otherwise get any value between 0 and 1 (say $\frac{1}{2}$ for instance) and this value gives no information as to what our binary choice should be.

1.3.2 Knapsack problem

The company KitchTech wishes to ship a number of crates from Toronto to Kitchener in a freight container. The crates are of six possible types, say type 1 through type 6. Each type of crate has a given weight in kilograms and has a particular retail value in \$, as indicated in the following table:

Type	1	2	3	4	5	6
Weight (kg)	30	20	30	90	30	70
Value (\$)	60	70	40	70	20	90

In addition you have the following constraints:

(1) you cannot send more than ten crates of the same type in the container;
(2) you can only send crates of type 3, if you send at least one crate of type 4;
(3) at least one of the following two conditions has to be satisfied:
  i. a total of at least four crates of type 1 or type 2 is selected or
  ii. a total of at least four crates of type 5 or type 6 is selected.

Finally, the total weight allowed on the freight container is 1000 kilograms. Your goal is to decide how many crates of each type to place in the freight container so that the value of the crates in the container is maximized. We wish to find an IP formulation for this problem. Thus, we need to determine the variables, the constraints, and the objective function.

Variables. We will have variables x_i for $i = 1, \ldots, 6$ to indicate how many crates of type i we place in the container. We will also require an additional binary variable y to handle condition (3).

Constraints. The total weight of the crates selected in kilograms is given by

$$30x_1 + 20x_2 + 30x_3 + 90x_4 + 30x_5 + 70x_6.$$

This weight should not exceed 1000 kilograms. Thus

$$30x_1 + 20x_2 + 30x_3 + 90x_4 + 30x_5 + 70x_6 \leq 1000. \tag{1.17}$$

Condition (1) simply says that for all $i = 1, \ldots, 6$

$$x_i \leq 10. \tag{1.18}$$

We claim that condition (2) can be stated as

$$x_3 \le 10x_4. \tag{1.19}$$

If no crates of type 4 is sent, then $x_4 = 0$, which implies by (1.19) that $x_3 = 0$ as well, i.e. no crates of type 3 are sent. On the other hand, if at least one crate of type 4 is sent, then $x_4 \ge 1$ and (1.19) says that $x_3 \le 10$, which is the maximum number of crates of type 3 we can send anyway.

It remains to express condition (3). The binary variable y will play the following role. If $y = 1$, then we want (i) to be true, and if $y = 0$, then we want (ii) to be true. This can be achieved by adding the following two constraints:

$$\begin{aligned} x_1 + x_2 &\ge 4y \\ x_5 + x_6 &\ge 4(1 - y). \end{aligned} \tag{1.20}$$

Let us verify that (1.20) behaves as claimed. If $y = 1$, then the conditions become $x_1 + x_2 \ge 4$ and $x_5 + x_6 \ge 0$, which implies that (i) holds. If $y = 0$, then the conditions become $x_1 + x_2 \ge 0$ and $x_5 + x_6 \ge 1$, which implies that (ii) holds.

Objective function. The total value of the crates selected for the container is given by

$$60x_1 + 70x_2 + 40x_3 + 70x_4 + 20x_5 + 90x_6. \tag{1.21}$$

Thus, the IP formulation is given by objective function (1.21), and constraints (1.17), (1.18), (1.19), (1.20) as well as the condition that each variable x_i is integer, and $y \in \{0, 1\}$. We obtain

$$\begin{array}{lrcll}
\max & \multicolumn{4}{l}{60x_1 + 70x_2 + 40x_3 + 70x_4 + 20x_5 + 90x_6} \\
\text{subject to} & & & & \\
& \multicolumn{4}{l}{30x_1 + 20x_2 + 30x_3 + 90x_4 + 30x_5 + 70x_6 \le 10000} \\
& x_i & \le & 10 & (i = 1, \ldots, 6) \\
& x_3 & \le & 10x_4 & \\
& x_1 + x_2 & \ge & 4y & \\
& x_5 + x_6 & \ge & 4(1 - y) & \\
& x_i & \ge & 0 & (i = 1, \ldots, 6) \\
& x_i & & \text{integer} & (i = 1, \ldots, 6) \\
& y & \in & \{0, 1\}. &
\end{array}$$

Exercises

1 You are about to trek across the desert with a vehicle having 3.6 cubic metres (3.6m^3) of cargo space for goods. There are various types of items available for putting in this space, each with a different volume and a different net value for your trip, shown as follows:

Item type i	1	2	3	4	5	6	7
Volume v_i (m^3)	0.55	0.6	0.7	0.75	0.85	0.9	0.95
Net value n_i	250	300	500	700	750	900	950

(a) You need to decide which items to take, not exceeding the volume constraint. You can take at most one of any item. No item can be split into fractions. The total net value must be maximized. Formulate this problem as an LP or IP. (You may use the notation v_i and n_i for volume and net value of item i.)

(b) Your two friends have decided to come as well, each with an identical vehicle. There are exactly two items of each type. The question is, can you fit all 14 items in the vehicles without exceeding the volume constraints? No cutting items into pieces is permitted! Each item taken must be placed entirely in one of the vehicles. Formulate an LP or IP that has a feasible solution if and only if the items can be packed as desired. Describe how to determine from a feasible solution how to pack the items into the vehicles. Note that net value is ignored for part (b).

2 Consider a public swimming pool. In the following table, we give a list of seven potential lifeguards. For each lifeguard, we have the time he/she wants to start and finish work and how much he/she wishes to be paid for the work. The problem is to find a selection of lifeguards so that there is at least one (but possibly more than one) lifeguard present at each time between 1pm. and 9pm. An example of a possible selection would be Joy, Tim, and Beth. This selection has a total cost of 30 + 21 + 20.

Lifeguards	Joy	Dean	Tim	Celicia	Beth	Ivar	Eilene
Hours	1–5	1–3	4–7	4–9	6–9	5–8	8–9
Amount required	30	18	21	38	20	22	9

Formulate this problem as an IP.

3 You have gone to an exotic destination during the summer vacation and decided to do your part to stimulate the economy by going on a shopping spree. Unfortunately, the day before your return you realize that you can only take 20 kilograms of luggage on your flight which is less than the total weight of the items that you purchased. The next table gives the value and the weight in kilograms of each item:

	A	B	C	D	E	F
Weight (kg)	6	7	4	9	3	8
Value ($)	60	70	40	70	16	100

The problem is to decide which subset of the items to put in your luggage so that you maximize the total value of the items selected without exceeding the weight requirement (i.e. that the total weight of the items selected is no more than 20 kilograms). For instance, you could select items *A, C*, and *D* for a total value of \$170 and a total weight of 19 kilograms.

(a) Formulate this problem as an IP.

(b) Suppose that you only want to pack item *D* when item *A* is selected, but it is ok to pack item *A* without item *D*. Add constraints to your formulation that impose this additional condition.

(c) Suppose that the airline allows you exceed the 20 kilogram weight limit at a cost of \$15 per additional kilogram. For instance, you could select items *A, B* and *D* for a total value of \$200 and a total weight of 22 kilogram and pay $2 \times$ \$15 to the airline for exceeding the maximum capacity by 2 kilograms. Modify your formulation so that you are allowed to go over the 20 kilogram capacity and such that you maximize the total value of the items packed minus the cost paid to the airline.

Note, for (b) and (c) the resulting formulation should remain an IP.

4 The Waterloo hotel wants to rent rooms 1, 2, and 3 for New Year's night. Abby is willing to pay \$60 for room 1, \$50 for room 2, but is not interested in room 3. Bob is willing to pay \$40 for room 1, \$70 for room 2, and \$80 for room 3. Clarence is not interested in room 1, but is willing to pay \$55 for room 2 and \$75 for room 3. Donald is willing to pay \$65 for room 1, \$90 for room 2, but is not interested in room 3. The information is summarized in the following table:

Room number	Abby's offer	Bob's offer	Clarence's offer	Donald's offer
1	\$60	\$40	not interested	\$65
2	\$50	\$70	\$55	\$90
3	not interested	\$80	\$75	not interested

The hotel wants to fill up rooms 1,2,3 with some of the potential clients (Abby, Bob, Clarence, and Donald) in a way that maximizes the total revenue. Each room is to be assigned to *exactly one* potential client, and each potential client is to be assigned *at most one* room. As an example, Room 1 could be assigned to Bob, room 2 to Abby, and room 3 to Clarence (while Donald would not get to stay in the hotel). This would yield a revenue of \$40+\$50+\$75=\$165.

(a) Formulate this problem as an IP. Your solution should be easy to modify if we change the values in the table.

(b) Abby and Bob have a history of loud and rude behavior when celebrating together. In the interest of keeping the New Year's eve party orderly, the hotel management decides that it does not wish to rent rooms to *both* Abby and Bob. Add a constraint to the IP in (a) that will enforce this condition (the resulting formulation should still be an IP).

5 You wish to find out how to pack crates on a transport plane in an optimal way. The crates are of five possible types, namely *A, B, C, D, E*. For each crate type, the next table gives its weight (in kg), its volume (in cubic meters), and its value (in dollars):

Type	A	B	C	D	E
Weight	500	1500	2100	600	400
Volume	25	15	13	20	16
Value	50000	60000	90000	40000	30000

The transport plane is divided into three segments: Front, Middle, and Back. Each segment has a limited volume (in cubic meters), and a limit on the weight of the cargo in that segment (in kg):

Segment	Front	Middle	Back
Available volume	200	500	300
Weight capacity	8000	20000	6000

Finally, to keep the plane balanced we need to satisfy the following constraints:

$$\text{weight of Middle cargo} \geq \text{weight of Front cargo} + \text{weight Back cargo},$$

$$\text{weight of Middle cargo} \leq 2 \times (\text{weight of Front cargo} + \text{weight Back cargo}).$$

Suppose that there are 12 crates of type A, eight crates of type B, 22 crates of type C, 15 crates of type D, and 11 crates of type E that are waiting to be transported. Your goal is to maximize the total value of the crates on the plane. You need to decide how many crates of each type are going in what segment of the plane. Formulate your problem as an IP.

6 Consider an LP with variables x_1, x_2, x_3, x_4. Suppose that the LP includes the constraints $x_1, x_2, x_3, x_4 \geq 0$.

(a) Consider the constraint:

$$x_4 \geq |x_3 - 2x_1|. \tag{1.22}$$

Suppose that we want to add to the LP the condition that (1.22) is satisfied. Show how to satisfy this requirement so that the resulting formulation is an LP.

HINT: rewrite (1.22) as a pair of linear inequalities.

(b) Consider the following inequalities:

$$6x_1 + 2x_2 + 3x_3 + 3x_4 \geq 3, \tag{1.23}$$

$$2x_1 + 4x_2 + 2x_3 + 7x_4 \geq 9. \tag{1.24}$$

Suppose that we want to add to an IP the condition that at *least one* of constraints (1.23) or (1.24) is satisfied. Show how to satisfy this requirement so that the resulting formulation is an IP.

HINT: add a binary variable indicating whether (1.23) or (1.24) must be satisfied. Note that the left-hand side of either (1.23) or (1.24) is always nonnegative.

(c) Suppose that for $i = 1, \ldots, k$ we have a non-negative vector a^i with four entries and a number β_i (both a^i and β_i are constants). Let r be any number between 1 and k. Consider the following set of inequalities:

$$(a_i)^\top x \geq \beta_i \qquad (i = 1, \ldots, k). \tag{1.25}$$

We want to add to an IP the condition that at *least* r of the constraints are satisfied. Show how to satisfy this requirement so that the resulting formulation is an IP.

HINT: add a binary variable for each constraint in (1.25).

(d) Consider the following set of values:

$$\mathscr{S} := \{3, 9, 17, 19, 36, 67, 1893\}.$$

Suppose that we want to add to an IP the condition that the variable x takes only one of the values in $\mathscr{S}$. Show how to satisfy this requirement so that the resulting formulation is an IP.

HINT: add a binary variables for each number in the set $\mathscr{S}$.

7 The company C & O operates an oil pipeline pumping oil from Alberta to various states in the Northwestern USA. Figure 1.2 shows the direction of flow, four input lines, and the three output lines. Note for instance that State A can only get its oil from either Input 1 or from the Yukon input line.

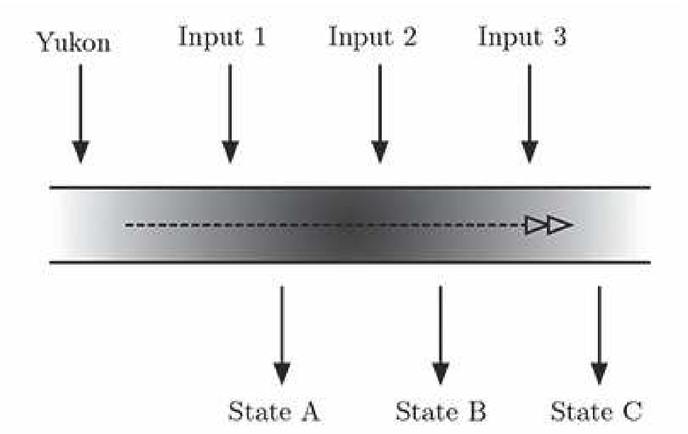


Figure 1.2 The structure of the oil pipeline, inputs and outputs.

Each input line has a capacity (barrels/day) and a cost per barrel:

Input line	1	2	3	Yukon
Capacity	4000	2000	3000	10000
Cost per barrel ($)	70	50	30	60

Each state has a daily demand (barrels/day) that must be met exactly:

State	A	B	C
Demand	3500	3000	4000

The input from the Yukon is not owned by the company and activating that line has a fixed cost of $11 000 per day.

Write an IP that plans the activities of the company C & O for a day (how many barrels of oil to pump from each input line) by minimizing the total daily cost of the company while meeting all the demand.

8 A company won a government bid to meet the yearly demands $d_1, d_2, \ldots, d_n$ in the areas $j \in \{1, 2, \ldots, n\}$. Now the company has to decide where to build its factories and how much of each factory's output will be shipped to which of these n areas.

There are m potential locations for building the factories. If the company decides to build at location $i \in \{1, 2, \ldots, m\}$, then the fixed cost of building the factory (yearly amortized version) is f_i and the yearly capacity of the factory will be s_i. The cost of transporting one unit of the product from location i to area j is given as c_{ij}.

Construct an IP whose solution indicates where to build the factories, how many units of product to ship from each factory to each demand area so that the demand is met and the total yearly cost of the company is minimized.

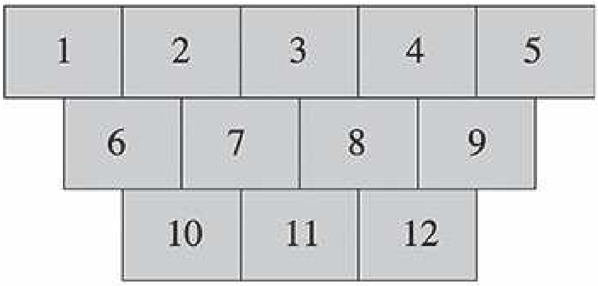
9 Your boss asks you to purchase b_i units of product i for each i in a set P of products. (These products are all divisible, i.e. they can be obtained in fractional amounts.) Of course, your boss wants you to spend as little money as possible. You call up all the stores in a set S of stores, and store j gives you a per-unit price c_{ij} for product i for all i, j.

(a) You decide to just order all b_i units of product i from the store that gives the cheapest per-unit price for each i. Show that this is optimal.

(b) Actually, there is another constraint. Your boss forgot to tell you that he does not want you to buy from too many different stores – he wants you to keep the number of stores from which you buy to at most (integer) k. Modify your formulation in (a), the resulting formulation should be an IP.

(c) It turns out that the stores have special deals. If the total value of your order from store j is at least t_j dollars, it will give you d_j cash back. (All stores j offer such a deal, with perhaps different values of t_j and d_j.) Modify your formulation in (b), the resulting formulation should be an IP.

10 KW mining has an open-pit mine with 12 blocks of ore as shown in the figure below. The *mineability condition* says that no block can be mined without mining all blocks which lie at least partially above it. For example, block 7 cannot be mined unless blocks 2 and 3 are mined, and block 12 requires blocks 8 and 9, and hence 3, 4, and 5 too.



(a) The net value of mining block i is given by m_i (in \$) for $i = 1, \ldots, 12$. Formulate as an IP the problem of deciding which blocks should be mined, in order to maximize the total value of blocks mined, and satisfy the mineability condition if at most seven blocks can be mined.

(b) The volume of mining block i is given by v_i (in m^3) for $i = 1, \ldots, 12$. What extra constraint would you add if, in addition to all constraints needed in (a), it is required that the total volume mined must not exceed 10 000 m^3?

(c) Mines often have a minimum short-term return requirement. For this mine, the board of the company requires that the total value of blocks mined in the first two years must total at least \$1 000 000. Each block takes one year to mine, at most two blocks can be mined at once, and a block cannot be mined in a given year unless all blocks lying at least partially above it were mined by the year before. Besides this new condition, the mineability condition still applies, and at most seven blocks can be mined. Formulate as an IP the problem of deciding which blocks should be mined, subject to these constraints, in order to maximize total value of blocks mined.

11 Suppose that you are given an $N \times N$ chessboard. We wish to place N queens on the board such that no two queens share any row, column, or diagonal. Figure 1.3 shows a solution for $N = 8$.

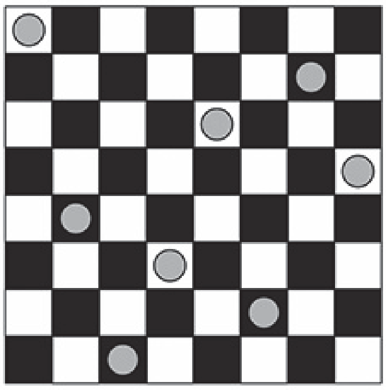


Figure 1.3

Formulate this problem as an integer feasibility problem (i.e. an IP without an objective function).

12 A 9×9 matrix A is partitioned into nine 3×3 submatrices $A_1, \ldots, A_9$ (of consecutive elements). Certain entries of A contain numbers from the set $\{1, \ldots, 9\}$. An example of such a pre-assignment is shown in Figure 1.4. A solution to the Sudoku game is an assignment of integers from 1 to 9 to each (unassigned) entry of the matrix such that:

- each row of A,
- each column of A,
- each 3 by 3 submatrix $A_1, \ldots, A_9$

contains every number from $\{1, \ldots, 9\}$ exactly once.

		5						3
				4	6			
		7						2
	1				3		6	9
	4		6		9		5	
9	8		2				7	
2						9		
			8	1				
6						4		

Figure 1.4

Formulate the problem of checking whether there is a solution to the Sudoku game as an integer feasibility problem (i.e. an IP without an objective function).

HINT: define a binary variable x_{ijk} that takes value 1 when entry i, j is assigned value k.

13 (Advanced) In *Conway's Game of Life*, we are given a chess board of size $n \times n$ for some positive integer n. Each cell (i, j) of this board has up to eight neighboring cells $N_{i,j}$. A cell (i, j) can be either *alive* or *dead* and a configuration of the game consists of a set of cells that are alive: $\mathscr{L} = \{(i_1, j_1), \ldots, (i_k, j_k)\}$. We use a set of simple rules to compute the successor configuration $\texttt{succ}(\mathscr{L})$ to $\mathscr{L}$:

(a) if there is at most one living cell in the neighborhood of (i, j), then (i, j) will be dead in the next iteration;

(b) if there are exactly two living cells in $N_{i,j}$, then we do not change the status of (i, j);

(c) if (i, j) has exactly three living neighbors in $N_{i,j}$, then its status in the next configuration will be alive;

(d) if there are at least four living cells in the neighborhood of (i, j), then (i, j) will be dead in the next iteration.[3]

A *still life* is a configuration $\mathscr{L}$ such that $\texttt{succ}(\mathscr{L}) = \mathscr{L}$ and the density of $\mathscr{L}$ is defined as $|\mathscr{L}|/n_2$. Given an $n \times n$ board, we are interested in finding a still life of maximum density.

Formulate the problem of finding a maximum density still life as an IP.