Θεωρία Βελτιστοποίησης

Βασίλειος Μαχαιράς
Πολιτικός Μηχανικός Ph.D.

Ακέραιος προγραμματισμός – πολύ-κριτηριακές αντικειμενικές συναρτήσεις

Πανεπιστήμιο Θεσσαλίας
Σχολή Θετικών Επιστημών
Τμήμα Πληροφορικής

Διάλεξη 12-13η/2017

Τι παρουσιάστηκε έως σήμερα

- Αναλυτικές μέθοδοι επίλυσης προβλημάτων βελτιστοποίησης.
- Μέθοδοι γραμμικού προγραμματισμού για την επίλυση προβλημάτων με γραμμική αντικειμενική συνάρτηση και γραμμικούς περιορισμούς ισότητας ή ανισότητας.
- Μέθοδοι επίλυσης μονοδιάστατων προβλημάτων βελτιστοποίησης, οι οποίες είναι χρήσιμες στην εύρεση του βέλτιστου μήκους βήματος σε προβλήματα επίλυσης με επαναληπτικές μεθόδους.
- Μέθοδοι επίλυσης μη γραμμικών προβλημάτων με και χωρίς περιορισμούς
- Μοντέρνες μέθοδοι βελτιστοποίησης: γενετικοί αλγόριθμοι, simulated annealing, particle swarm optimization, ant colony optimization

Τι θα παρουσιαστεί

- Μέθοδοι επίλυσης προβλημάτων με διακριτές μεταβλητές: ακέραιος προγραμματισμός
- Βελτιστοποίηση συναρτήσεων με πολλά κριτήρια (Multi-objective optimization)

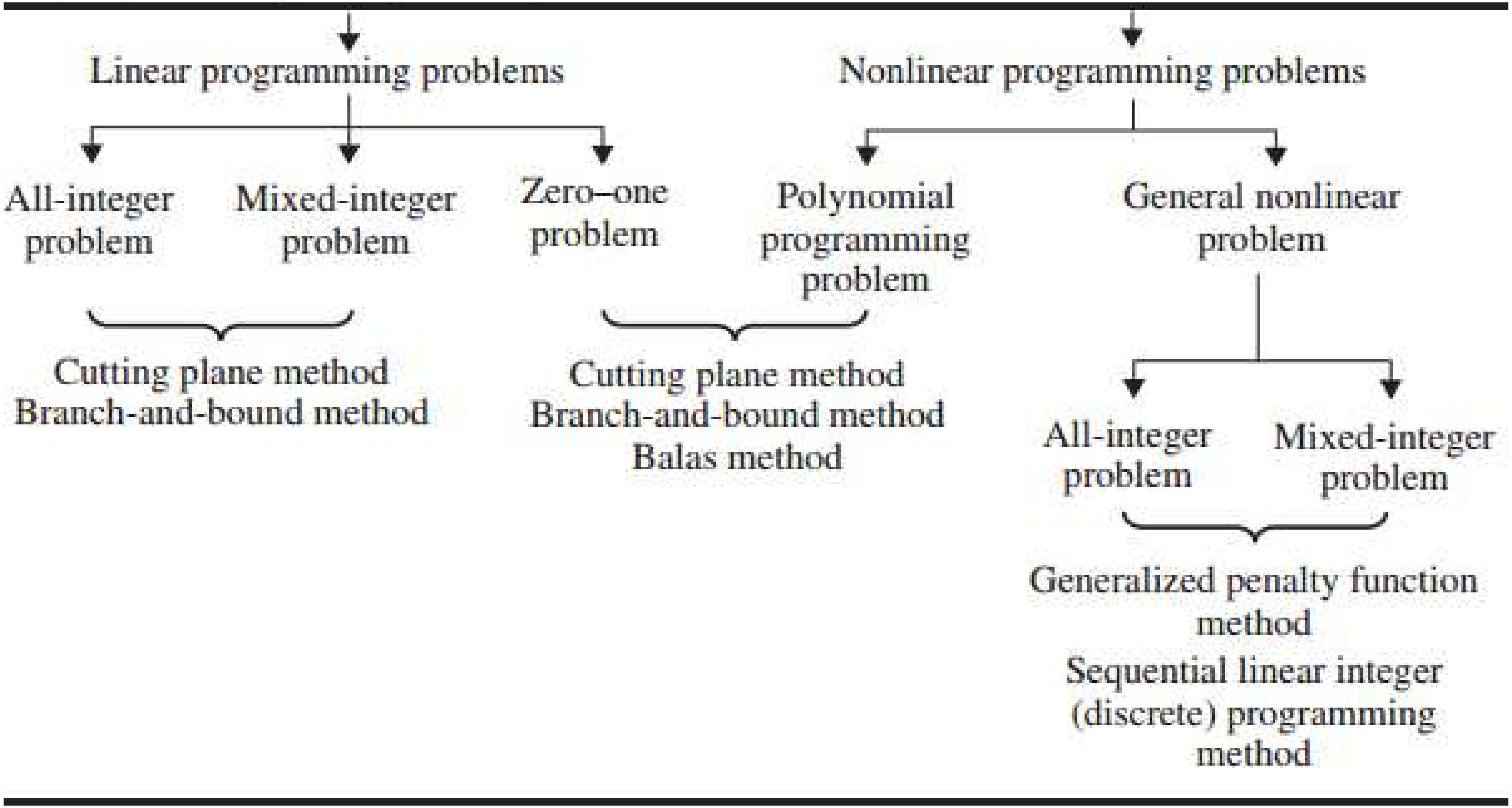
Ακέραιος προγραμματισμός

- Μέχρι τώρα είδαμε προβλήματα με μεταβλητές απόφασης, οι οποίες είναι συνεχείς και πραγματικοί αριθμοί (ανήκουν στο σύνολο πραγματικών αριθμών $\mathbb{R}$).
- Πολλά πρακτικά προβλήματα μπορούν να λάβουν ακέραιες αριθμητικές τιμές (διακριτές τιμές).
- Στις τεχνικές που παρουσιάστηκαν θα μπορούσε το πρόβλημα να επιλυθεί συμβατικά και το αποτέλεσμα να στρογγυλοποιηθεί στον κοντινότερο ακέραιο. Όμως πολλές φορές η στρογγυλοποίηση της λύσης παραβιάζει τους περιορισμούς.
- Αυτό μπορεί να αποφευχθεί αν προσεγγιστεί ως πρόβλημα ακέραιου προγραμματισμού.

Ακέραιος προγραμματισμός

- Όταν όλες οι μεταβλητές λαμβάνουν μόνο ακέραιες τιμές τότε ονομάζεται *all-integer programming problem.*
- Όταν λαμβάνουν μόνο διακριτές τιμές τότε ονομάζεται *discrete programming problem*
- Όταν μερικές μόνο από τις μεταβλητές λαμβάνουν ακέραιες τιμές τότε ονομάζεται *mixed-integer (discrete) programming problem.*
- Όταν οι μεταβλητές λαμβάνουν μόνο τιμές 0 ή 1 τότε ονομάζεται *zero-one programming problem*

Table 10.1 Integer Programming Methods



Γραφική αναπαράσταση

Consider the following integer programming problem:

$$\text{Maximize } f(\mathbf{X}) = 3x_1 + 4x_2$$

subject to

$$\begin{aligned} 3x_1 - x_2 &\leq 12 \\ 3x_1 + 11x_2 &\leq 66 \\ x_1 &\geq 0 \\ x_2 &\geq 0 \\ x_1 \text{ and } x_2 &\text{ are integers} \end{aligned} \tag{10.1}$$

The graphical solution of this problem, by ignoring the integer requirements, is shown in Fig. 10.1. It can be seen that the solution is $x_1 = 5\frac{1}{2}$, $x_2 = 4\frac{1}{2}$ with a value of $f = 34\frac{1}{2}$. Since this is a noninteger solution, we truncate the fractional parts and obtain the new solution as $x_1 = 5$, $x_2 = 4$, and $f = 31$. By comparing this solution with all other integer feasible solutions (shown by dots in Fig. 10.1), we find that this solution is optimum for the integer LP problem stated in Eqs. (10.1).

It is to be noted that truncation of the fractional part of a LP problem will not always give the solution of the corresponding integer LP problem. This can be illustrated

Γραφική αναπαράσταση

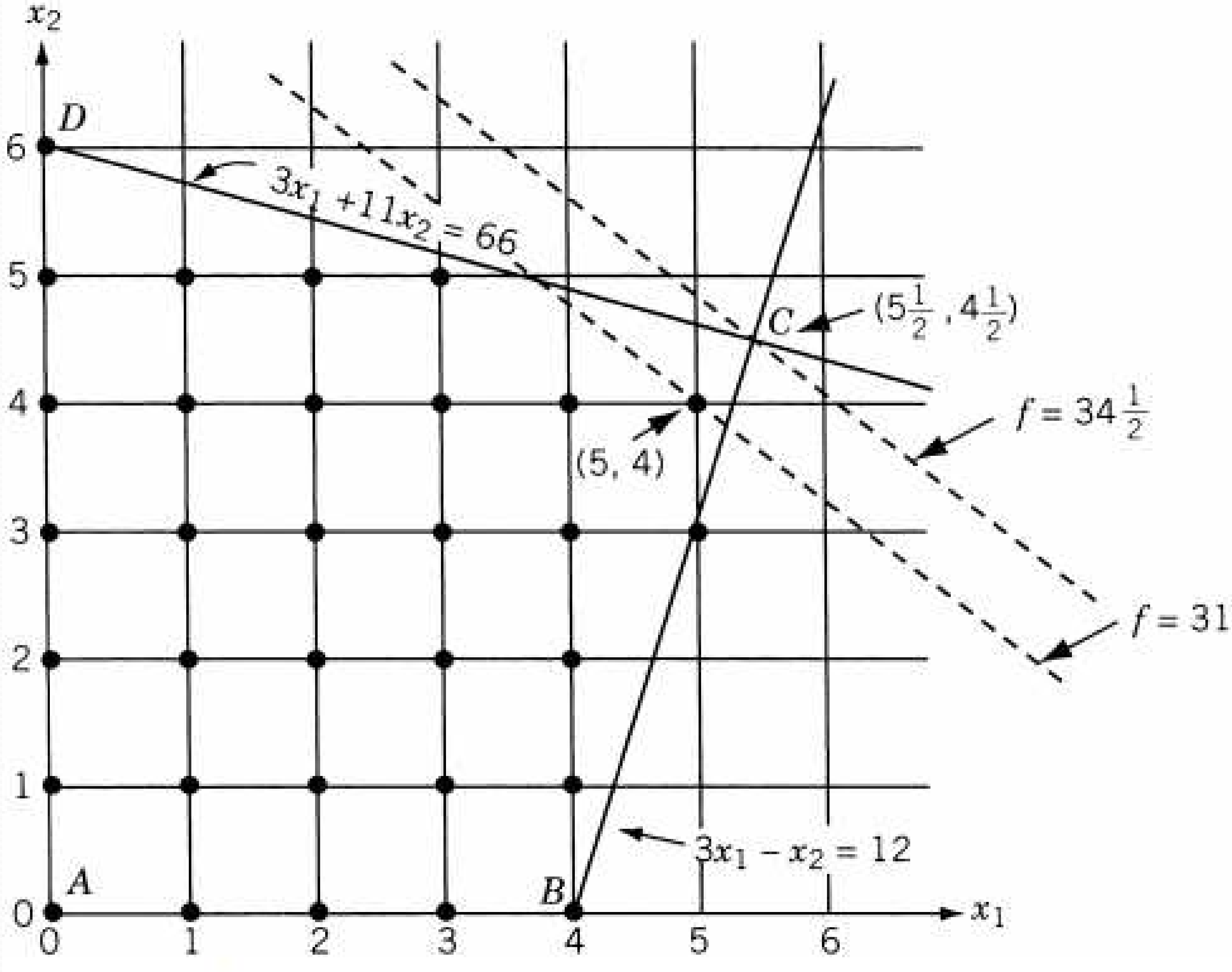


Figure 10.1 Graphical solution of the problem stated in Eqs. (10.1).

by changing the constraint $3x_1 + 11x_2 \leq 66$ to $7x_1 + 11x_2 \leq 88$ in Eqs. (10.1). With this altered constraint, the feasible region and the solution of the LP problem, without considering the integer requirement, are shown in Fig. 10.2. The optimum solution of this problem is identical with that of the preceding problem: namely, $x_1 = 5\frac{1}{2}$,

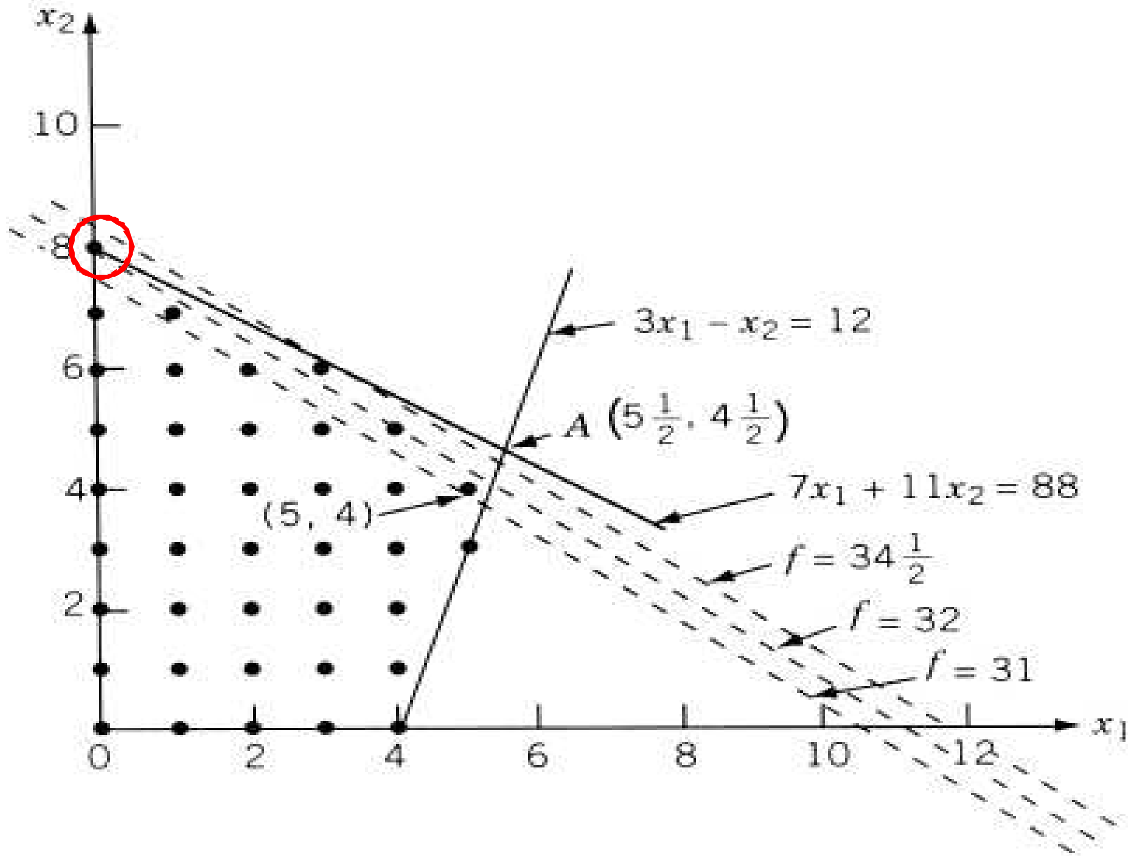


Figure 10.2 Graphical solution with modified constraint.

$x_2 = 4\frac{1}{2}$, and $f = 34\frac{1}{2}$. The truncation of the fractional part of this solution gives $x_1 = 5$, $x_2 = 4$, and $f = 31$. Although this truncated solution happened to be optimum to the corresponding integer problem in the earlier case, it is not so in the present case. In this case the optimum solution of the integer programming problem is given by $x_1^* = 0$, $x_2^* = 8$, and $f^* = 32$.

Gomory's cutting plane method

- Η μέθοδος βασίζεται στη δημιουργία επιπλέον περιορισμών, ώστε να μειωθεί το αρχικό σύνολο εφικτών λύσεων με τέτοια μεθοδολογία, η οποία θα περιέχει στο όριό της τη νέα λύση ως ακέραια.
- Το νέο σύνολο εφικτών λύσεων που προκύπτει από τους νέους περιορισμούς πρέπει να είναι κυρτό (convex) και
- το τμήμα που αφαιρείται να μην περιέχει διακριτές λύσεις του προβλήματος.
- Οι νέοι περιορισμοί εισάγονται με συστηματικό τρόπο.

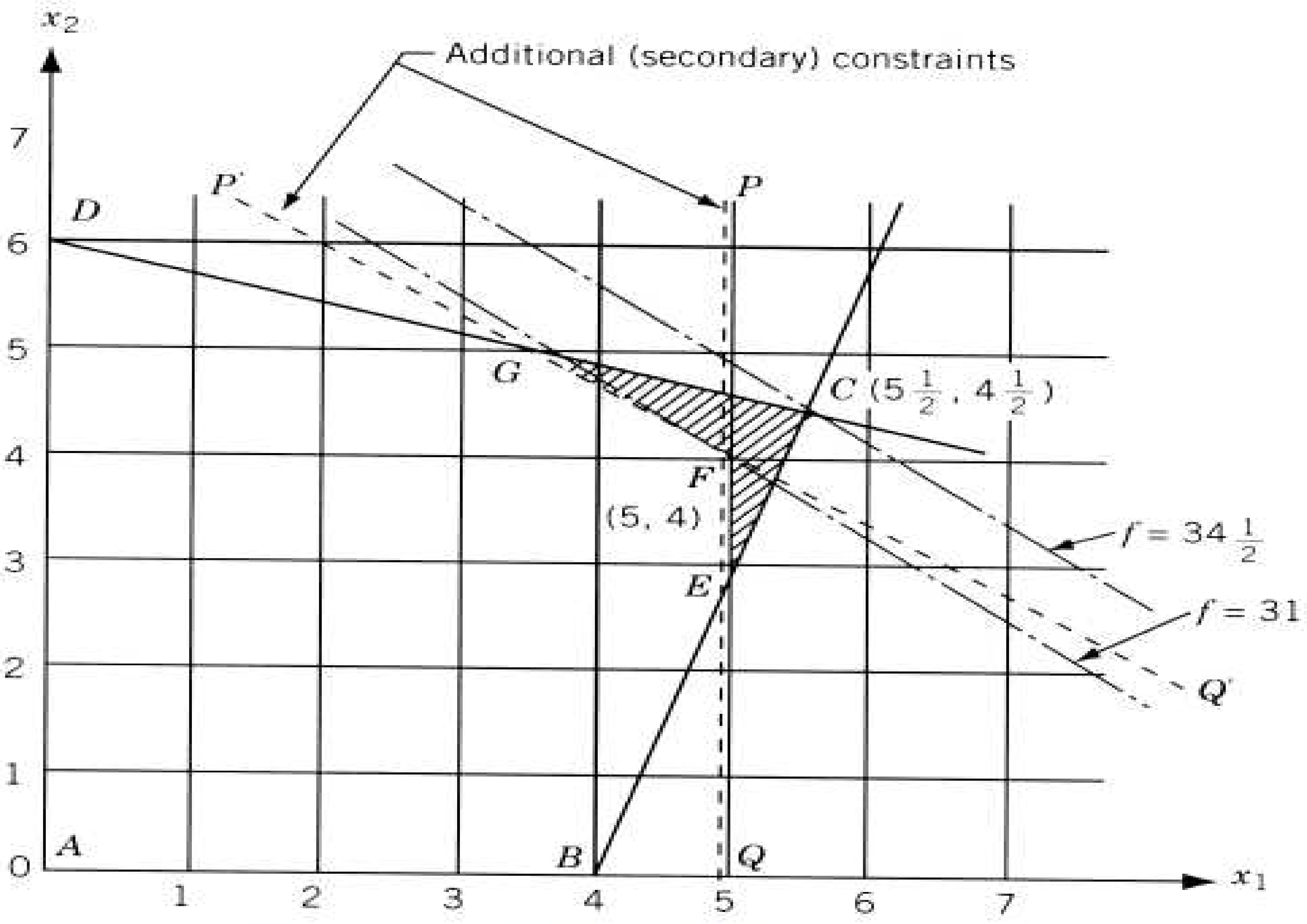


Figure 10.3 Effect of additional constraints.

Example 10.1

$$\text{Minimize } f = -3x_1 - 4x_2$$

subject to

$$3x_1 - x_2 + x_3 = 12$$

$$3x_1 + 11x_2 + x_4 = 66$$

$$x_i \geq 0, \quad i = 1 \text{ to } 4$$

$$\text{all } x_i \text{ are integers}$$

This problem can be seen to be same as the one stated in Eqs. (10.1) with the addition of slack variables x_3 and x_4.

SOLUTION

Step 1: Solve the LP problem by neglecting the integer requirement of the variables x_i, $i = 1$ to 4, using the regular simplex method as shown below:

Basic variables	Coefficients of variables x_1	x_2	x_3	x_4	$-f$	$\overline{b}_i$	$\overline{b}_i/\overline{a}_{is}$ for $\overline{a}_{is} > 0$
x_3	3	−1	1	0	0	12	
x_4	3	11 (Pivot element)	0	1	0	66	6 ←
$-f$	−3	−4	0	0	1	0	
		↑ Most negative $\overline{c}_j$					

Result of pivoting:

x_3	$\frac{36}{11}$ (Pivot element)	0	1	$\frac{1}{11}$	0	18	$\frac{11}{2}$ ← Smaller one
x_2	$\frac{3}{11}$	1	0	$\frac{1}{11}$	0	6	22
$-f$	$-\frac{21}{11}$	0	0	$\frac{4}{11}$	1	24	
	↑ Most negative $\overline{c}_j$						

Result of pivoting:

x_1	1	0	$\frac{11}{36}$	$\frac{1}{36}$	0	$\frac{11}{2}$
x_2	0	1	$-\frac{1}{12}$	$\frac{1}{12}$	0	$\frac{9}{2}$
$-f$	0	0	$\frac{7}{12}$	$\frac{5}{12}$	1	$\frac{69}{2}$

Since all the cost coefficients are nonnegative, the last tableau gives the optimum solution as

$$x_1 = \frac{11}{2}, \quad x_2 = \frac{9}{2}, \quad x_3 = 0, \quad x_4 = 0, \quad f_{\min} = -\frac{69}{2}$$

which can be seen to be identical to the graphical solution obtained in Section 10.2.

Step 2: Generate a Gomory constraint. Since the solution above is noninteger, a Gomory constraint has to be added to the last tableau. Since there is a tie between x_1 and x_2, let us select x_1 as the basic variable having the largest fractional value. From the row corresponding to x_1 in the last tableau, we can write

$$x_1 = \frac{11}{2} - \frac{11}{36} y_1 - \frac{1}{36} y_2 \tag{E$_1$}$$

where y_1 and y_2 are used in place of x_3 and x_4 to denote the nonbasic variables. By comparing Eq. (E$_1$) with Eq. (10.2), we find that

$$i = 1, \quad \overline{b}_1 = \frac{11}{2}, \quad \hat{b}_1 = 5, \quad \beta_1 = \frac{1}{2}, \quad \overline{a}_{11} = \frac{11}{36},$$

$$\hat{a}_{11} = 0, \quad \alpha_{11} = \frac{11}{36}, \quad \overline{a}_{12} = \frac{1}{36}, \quad \hat{a}_{12} = 0, \quad \text{and} \quad \alpha_{12} = \frac{1}{36}$$

From Eq. (10.9), the Gomory constraint can be expressed as

$$s_1 - \alpha_{11} y_1 - \alpha_{12} y_2 = -\beta_1 \tag{E$_2$}$$

where s_1 is a new nonnegative (integer) slack variable. Equation (E$_2$) can be written as

$$s_1 - \frac{11}{36} y_1 - \frac{1}{36} y_2 = -\frac{1}{2} \tag{E$_3$}$$

By introducing this constraint, Eq. (E$_3$), into the previous optimum tableau, we obtain the new tableau shown below:

Basic variables	Coefficients of variables							
	x_1	x_2	y_1	y_2	$-f$	s_1	$\overline{b}_i$	$\overline{b}_i/\overline{a}_{is}$ for $\overline{a}_{is} > 0$
x_1	1	0	$\frac{11}{36}$	$\frac{1}{36}$	0	0	$\frac{11}{2}$	
x_2	0	1	$-\frac{1}{12}$	$\frac{1}{12}$	0	0	$\frac{9}{2}$	
$-f$	0	0	$\frac{7}{12}$	$\frac{5}{12}$	1	0	$\frac{69}{2}$	
s_1	0	0	$-\frac{11}{36}$	$-\frac{1}{36}$	0	1	$-\frac{1}{2}$	

Step 3: Apply the dual simplex method to find a new optimum solution. For this, we select the pivotal row r such that $\bar{b}_r = \min(\bar{b}_i < 0) = -\frac{1}{2}$ corresponding to s_1 in this case. The first column s is selected such that

$$\frac{\bar{c}_s}{-\bar{a}_{rs}} = \min_{\bar{a}_{rj} < 0} \left(\frac{\bar{c}_j}{-\bar{a}_{rj}} \right)$$

Here

$$\frac{\bar{c}_j}{-\bar{a}_{rj}} = \frac{7}{12} \times \frac{36}{11} = \frac{21}{11} \quad \text{for column } y_1$$

$$= \frac{5}{12} \times \frac{36}{1} = 15 \quad \text{for column } y_2.$$

Since $\frac{21}{11}$ is minimum out of $\frac{21}{11}$ and 15, the pivot element will be $-\frac{11}{36}$. The result of pivot operation is given in the following tableau:

Basic variables	Coefficients of variables							$\bar{b}_i/\bar{a}_{is}$ for $\bar{a}_{is} > 0$
	x_1	x_2	y_1	y_2	$-f$	s_1	$\bar{b}_i$	
x_1	1	0	0	0	0	1	5	
x_2	0	1	0	$\frac{1}{11}$	0	$-\frac{3}{11}$	$\frac{51}{11}$	
$-f$	0	0	0	$\frac{4}{11}$	1	$\frac{21}{11}$	$\frac{369}{11}$	
y_1	0	0	1	$\frac{1}{11}$	0	$-\frac{36}{11}$	$\frac{18}{11}$	

The solution given by the present tableau is $x_1 = 5$, $x_2 = 4\frac{7}{11}$, $y_1 = 1\frac{7}{11}$, and $f = -33\frac{6}{11}$, in which some variables are still nonintegers.

Step 4: Generate a new Gomory constraint. To generate the new Gomory constraint, we arbitrarily select x_2 as the variable having the largest fractional value (since there is a tie between x_2 and y_1). The row corresponding to x_2 gives

$$x_2 = \frac{51}{11} - \frac{1}{11}y_2 + \frac{3}{11}s_1$$

From this equation, the Gomory constraint [Eq. (10.9)] can be written as

$$s_2 - \frac{1}{11}y_2 + \frac{3}{11}s_1 = -\frac{7}{11}$$

When this constraint is added to the previous tableau, we obtain the following tableau:

Basic variables	Coefficients of variables							
	x_1	x_2	y_1	y_2	$-f$	s_1	s_2	$\bar{b}_i$
x_1	1	0	0	0	0	1	0	5
x_2	0	1	0	$\frac{1}{11}$	0	$-\frac{3}{11}$	0	$\frac{51}{11}$
y_1	0	0	1	$\frac{1}{11}$	0	$-\frac{36}{11}$	0	$\frac{18}{11}$
$-f$	0	0	0	$\frac{4}{11}$	1	$\frac{21}{11}$	0	$\frac{369}{11}$
s_2	0	0	0	$-\frac{1}{11}$	0	$\frac{3}{11}$	1	$-\frac{7}{11}$

Step 5: Apply the dual simplex method to find a new optimum solution. To carry the pivot operation, the pivot row is selected to correspond to the most negative value of $\overline{b}_i$. This is the s_2 row in this case.

Since only $\overline{a}_{rj}$ corresponding to column y_2 is negative, the pivot element will be $-\frac{1}{11}$ in the s_2 row. The pivot operation on this element leads to the following tableau:

Basic variables	Coefficients of variables							
	x_1	x_2	y_1	y_2	$-f$	s_1	s_2	$\overline{b}_i$
x_1	1	0	0	0	0	1	0	5
x_2	0	1	0	0	0	0	1	4
y_1	0	0	1	0	0	−3	1	1
$-f$	0	0	0	0	1	3	4	31
y_2	0	0	0	1	0	−3	−11	7

The solution given by this tableau is $x_1 = 5$, $x_2 = 4$, $y_1 = 1$, $y_2 = 7$, and $f = -31$, which can be seen to satisfy the integer requirement. Hence this is the desired solution.

BRANCH-AND-BOUND method

The branch-and-bound method is very effective in solving mixed-integer linear and nonlinear programming problems. The method was originally developed by Land and Doig [10.8] to solve integer linear programming problems and was later modified by Dakin [10.23]. Subsequently, the method has been extended to solve nonlinear mixed-integer programming problems. To see the basic solution procedure, consider the following nonlinear mixed-integer programming problem:

$$\text{Minimize } f(\mathbf{X}) \tag{10.45}$$

subject to

$$g_j(\mathbf{X}) \geq 0, \quad j = 1, 2, \ldots, m \tag{10.46}$$

$$h_k(\mathbf{X}) = 0, \quad k = 1, 2, \ldots, p \tag{10.47}$$

$$x_j = \text{integer}, \quad j = 1, 2, \ldots, n_0 \ (n_0 \leq n) \tag{10.48}$$

In the branch-and-bound method, the integer problem is not directly solved. Rather, the method first solves a continuous problem obtained by relaxing the integer restrictions on the variables. If the solution of the continuous problem happens to be an integer solution, it represents the optimum solution of the integer problem. Otherwise, at least one of the integer variables, say x_i, must assume a nonintegral value. If x_i is not an integer, we can always find an integer $[x_i]$ such that

$$[x_i] < x_i < [x_i] + 1 \tag{10.49}$$

Then two subproblems are formulated, one with the additional upper bound constraint

$$x_i \leq [x_i] \tag{10.50}$$

and another with the lower bound constraint

$$x_i \geq [x_i] + 1 \tag{10.51}$$

The process of finding these subproblems is called *branching*.

The branching process eliminates some portion of the continuous space that is not feasible for the integer problem, while ensuring that none of the integer feasible solutions are eliminated. Each of these two subproblems are solved again as a continuous problem. It can be seen that the solution of a continuous problem forms a *node* and from each node two branches may originate.

The process of branching and solving a sequence of continuous problems discussed above is continued until an integer feasible solution is found for one of the two continuous problems. When such a feasible integer solution is found, the corresponding value of the objective function becomes an upper bound on the minimum value of the objective function. At this stage we can eliminate from further consideration all the continuous solutions (nodes) whose objective function values are larger than the upper bound. The nodes that are eliminated are said to have been *fathomed* because it is not possible to find a better integer solution from these nodes (solution spaces) than what we have now. The value of the upper bound on the objective function is updated whenever a better bound is obtained.

It can be seen that a node can be fathomed if any of the following conditions are true:

1. The continuous solution is an integer feasible solution.
2. The problem does not have a continuous feasible solution.
3. The optimal value of the continuous problem is larger than the current upper bound.

The algorithm continues to select a node for further branching until all the nodes have been fathomed. At that stage, the particular fathomed node that has the integer feasible solution with the lowest value of the objective function gives the optimum solution of the original nonlinear integer programming problem.

- Solve linear programming relaxations to bound the objective function
- Create branches by adding constraints that eliminate non-integer values

Considering the following Integer Programming problem:

$$Max \;\; z = 11x_1 + 14x_2$$

Subject to

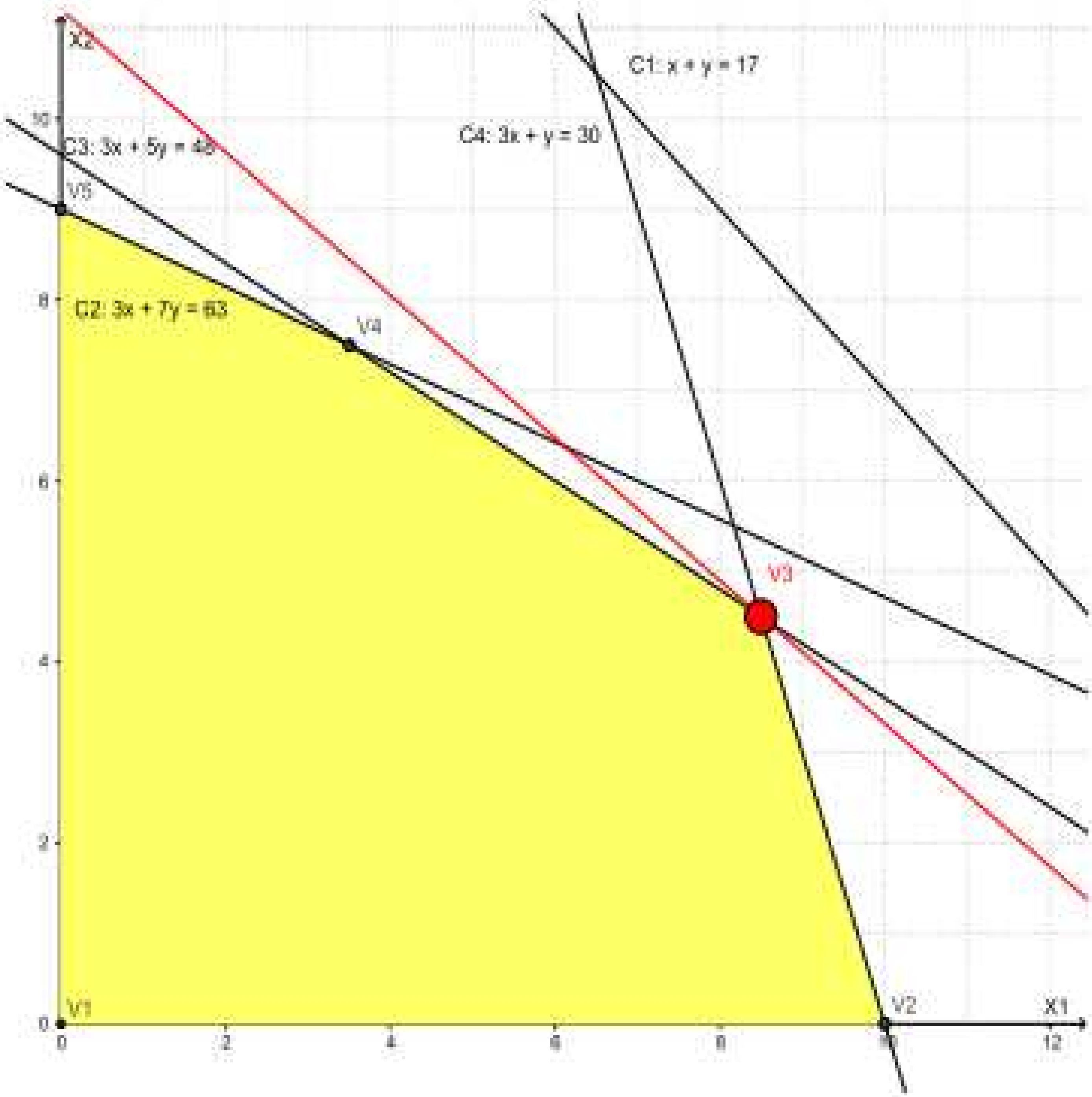
$$\begin{aligned} 1x_1 + 1x_2 &\leq 17 \\ 3x_1 + 7x_2 &\leq 63 \\ 3x_1 + 5x_2 &\leq 48 \\ 3x_1 + 1x_2 &\leq 30 \\ x_1, x_2 &\geq 0 \\ x_1, x_2 &\in \mathbb{Z} \end{aligned}$$

$$Max \;\; z = 11x_1 + 14x_2$$

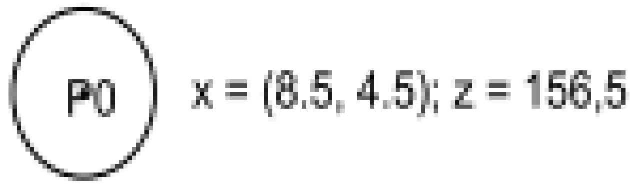
Subject to

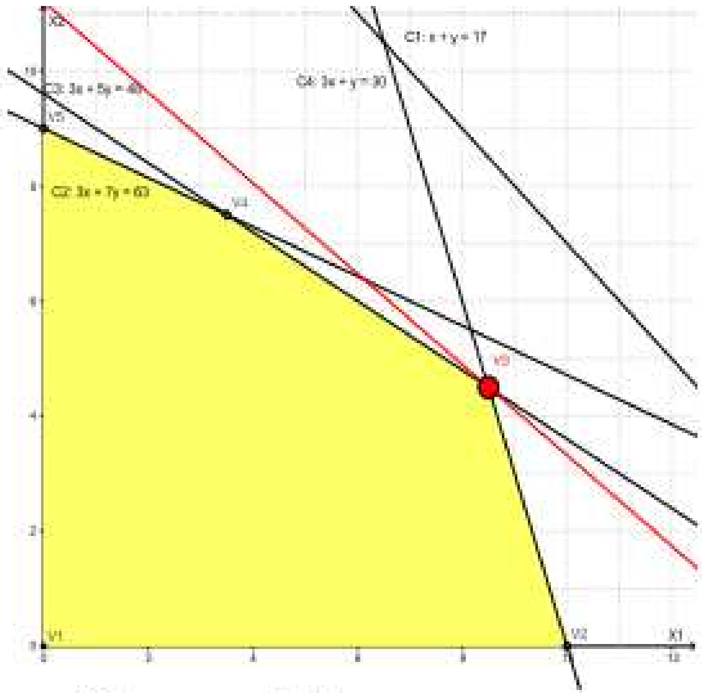
$$\begin{aligned} 1x_1 + 1x_2 &\leq 17 \\ 3x_1 + 7x_2 &\leq 63 \\ 3x_1 + 5x_2 &\leq 48 \\ 3x_1 + 1x_2 &\leq 30 \\ x_1, x_2 &\geq 0 \end{aligned}$$

$x^* = (8.5, 4.5) \;\; z^* = 156.5$

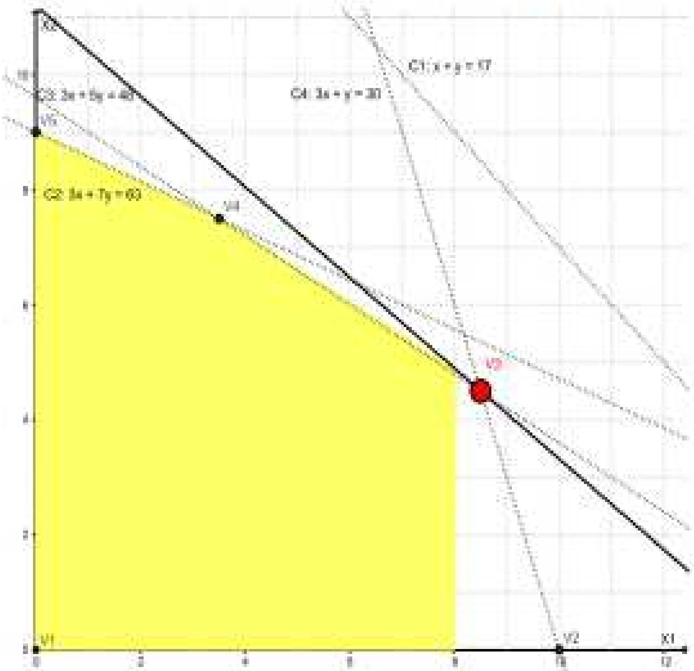


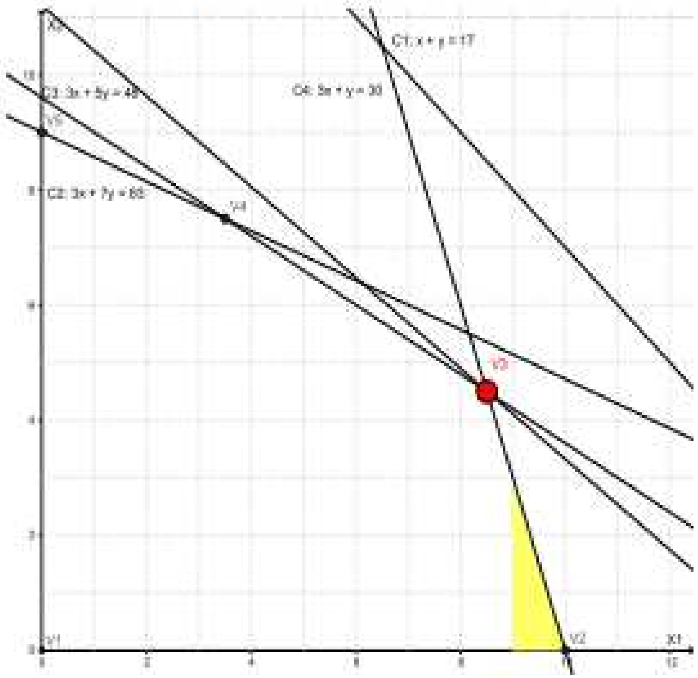
X2
10
C3: 3x + 5y = 48
V5
C1: x + y = 17
C4: 3x + y = 30
8
C2: 3x + 7y = 63
V4
6
V3
4
2
0
V1
V2
X1
0
2
4
6
8
10
12


P0
x = (8.5, 4.5); z = 156,5



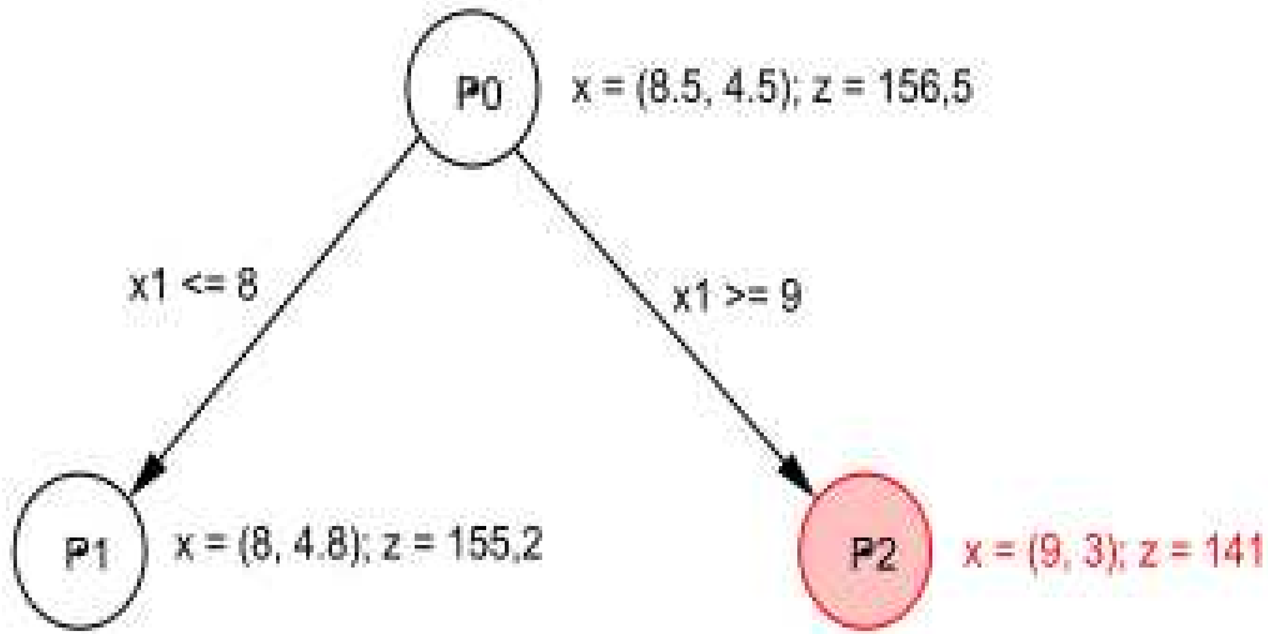
Branching on x_1:



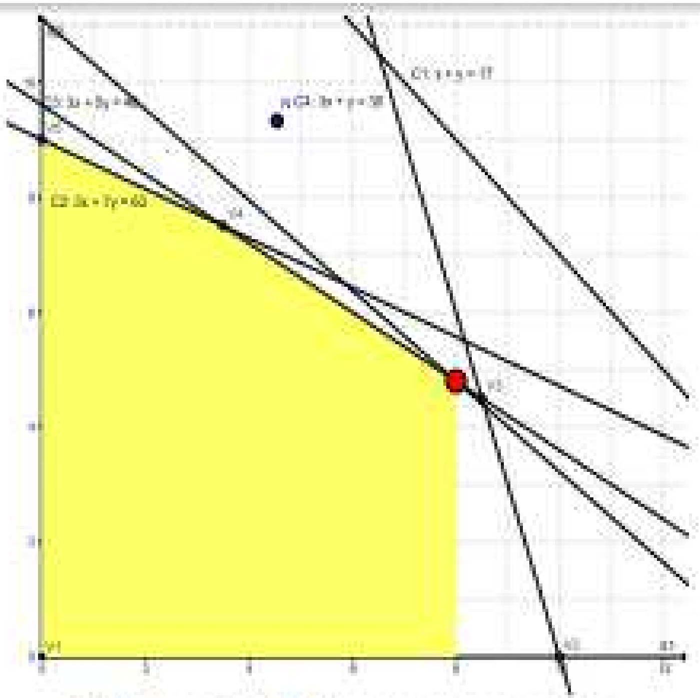


$$
\begin{array}{lrcl}
 & Max \;\; z = 11x_1 + 14x_2 & & \\
\text{Subject to} & 1x_1 + 1x_2 & \leq & 17 \\
 & 3x_1 + 7x_2 & \leq & 63 \\
 & 3x_1 + 5x_2 & \leq & 48 \\
 & 3x_1 + 1x_2 & \leq & 30 \\
 & x_1 & \leq & 8 \\
 & x_1, x_2 & \geq & 0 \quad x^* = (8, 4.8) \;\; z^* = 155.2
\end{array}
$$

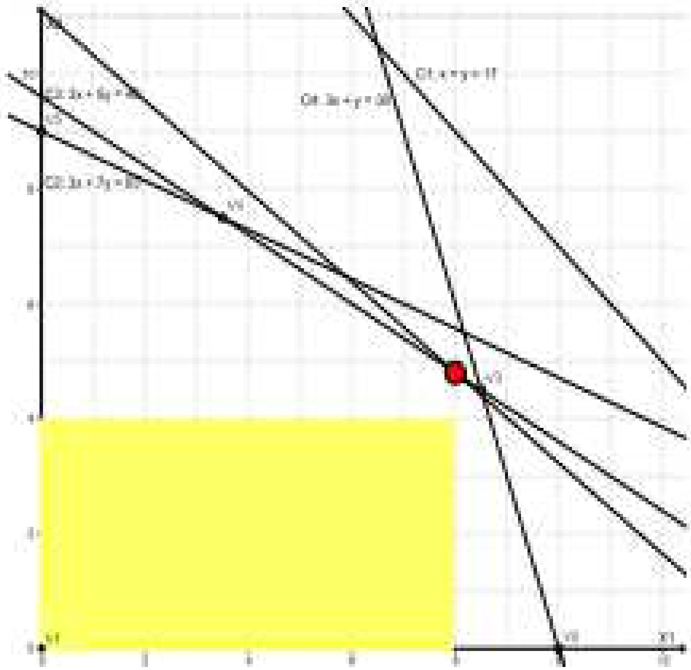
$$
\begin{array}{lrcl}
 & Max \;\; z = 11x_1 + 14x_2 & & \\
\text{Subject to} & 1x_1 + 1x_2 & \leq & 17 \\
 & 3x_1 + 7x_2 & \leq & 63 \\
 & 3x_1 + 5x_2 & \leq & 48 \\
 & 3x_1 + 1x_2 & \leq & 30 \\
 & x_1 & \geq & 9 \\
 & x_1, x_2 & \geq & 0 \quad x^* = (9, 3) \;\; z^* = 141
\end{array}
$$

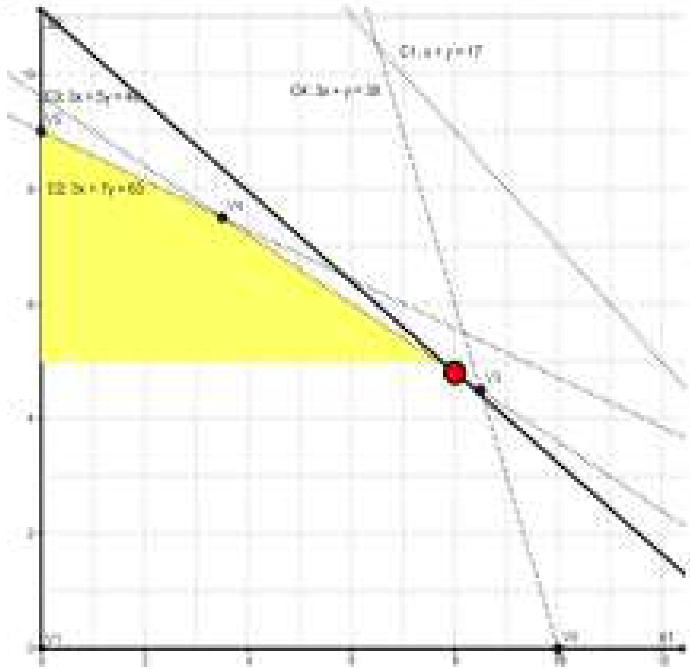


P0
x = (8.5, 4.5); z = 156,5
x1 <= 8
x1 >= 9
P1
x = (8, 4.8); z = 155,2
P2
x = (9, 3); z = 141



Branching on x_2:





$$Max \;\; z = 11x_1 + 14x_2$$

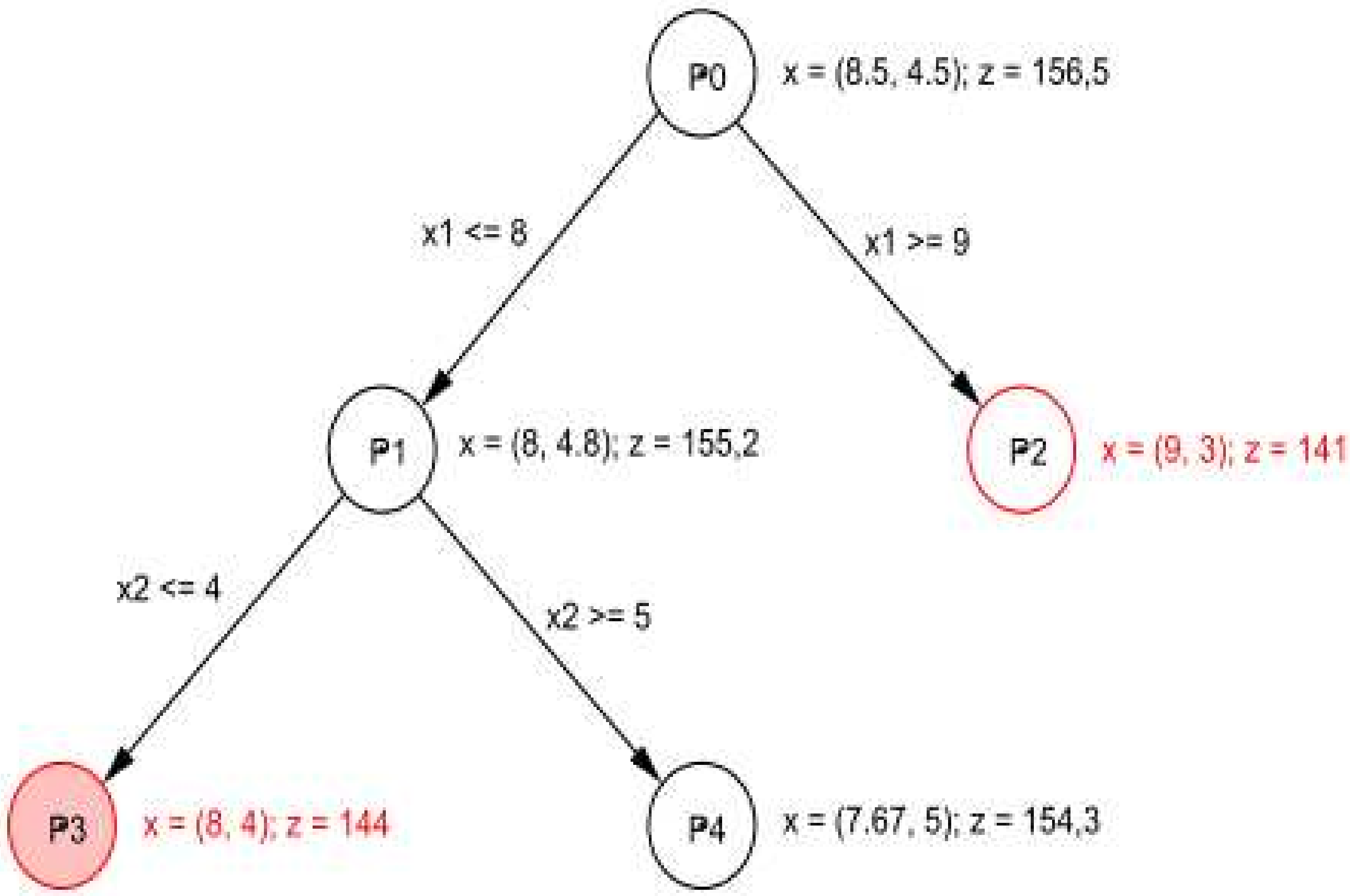
Subject to

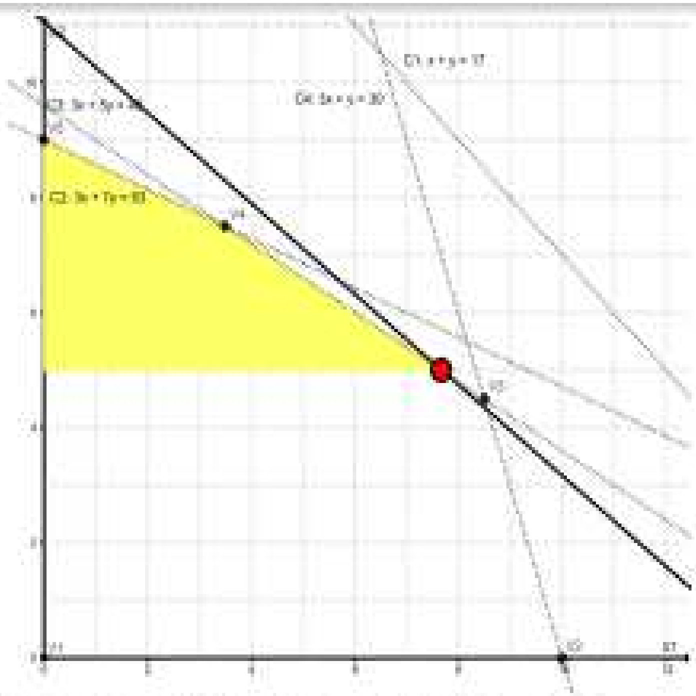
$$\begin{aligned} 1x_1 + 1x_2 &\leq 17 \\ 3x_1 + 7x_2 &\leq 63 \\ 3x_1 + 5x_2 &\leq 48 \\ 3x_1 + 1x_2 &\leq 30 \\ x_1 \leq 8 \;\; &\& \;\; x_2 \leq 4 \\ x_1, x_2 &\geq 0 \quad x^* = (8, 4) \;\; z^* = 144 \end{aligned}$$

$$Max \;\; z = 11x_1 + 14x_2$$

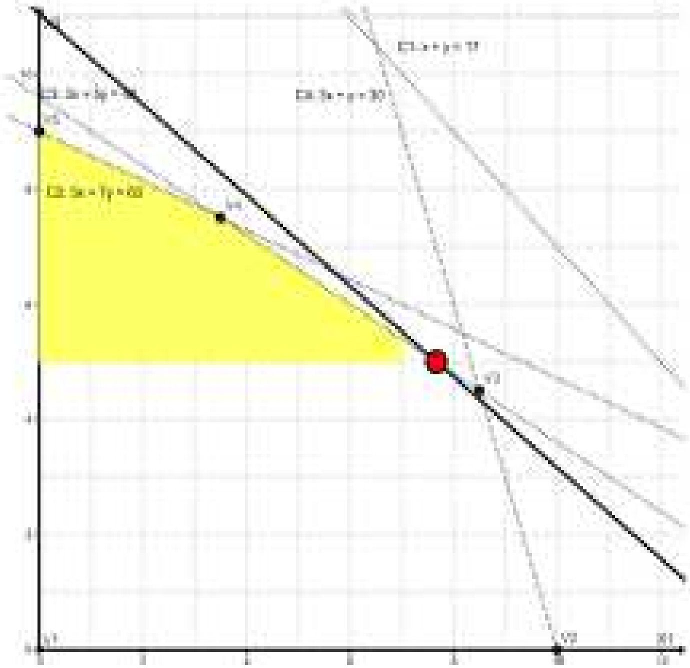
Subject to

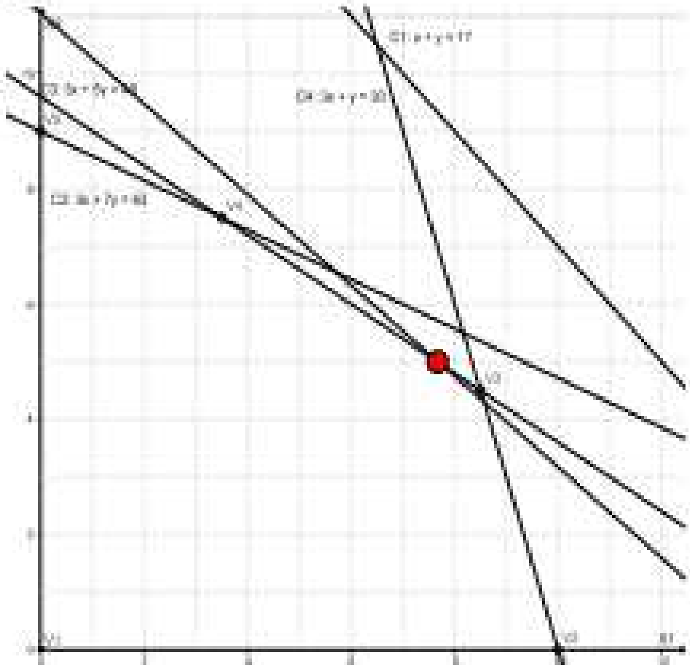
$$\begin{aligned} 1x_1 + 1x_2 &\leq 17 \\ 3x_1 + 7x_2 &\leq 63 \\ 3x_1 + 5x_2 &\leq 48 \\ 3x_1 + 1x_2 &\leq 30 \\ x_1 \leq 8 \;\; &\& \;\; x_2 \geq 5 \\ x_1, x_2 &\geq 0 \quad x^* = (7.67, 5) \;\; z^* = 154.3 \end{aligned}$$


P0
x = (8.5, 4.5); z = 156,5
x1 <= 8
x1 >= 9
P1
x = (8, 4.8); z = 155,2
P2
x = (9, 3); z = 141
x2 <= 4
x2 >= 5
P3
x = (8, 4); z = 144
P4
x = (7.67, 5); z = 154,3



Branching again on x_1:





$$Max \quad z = 11x_1 + 14x_2$$

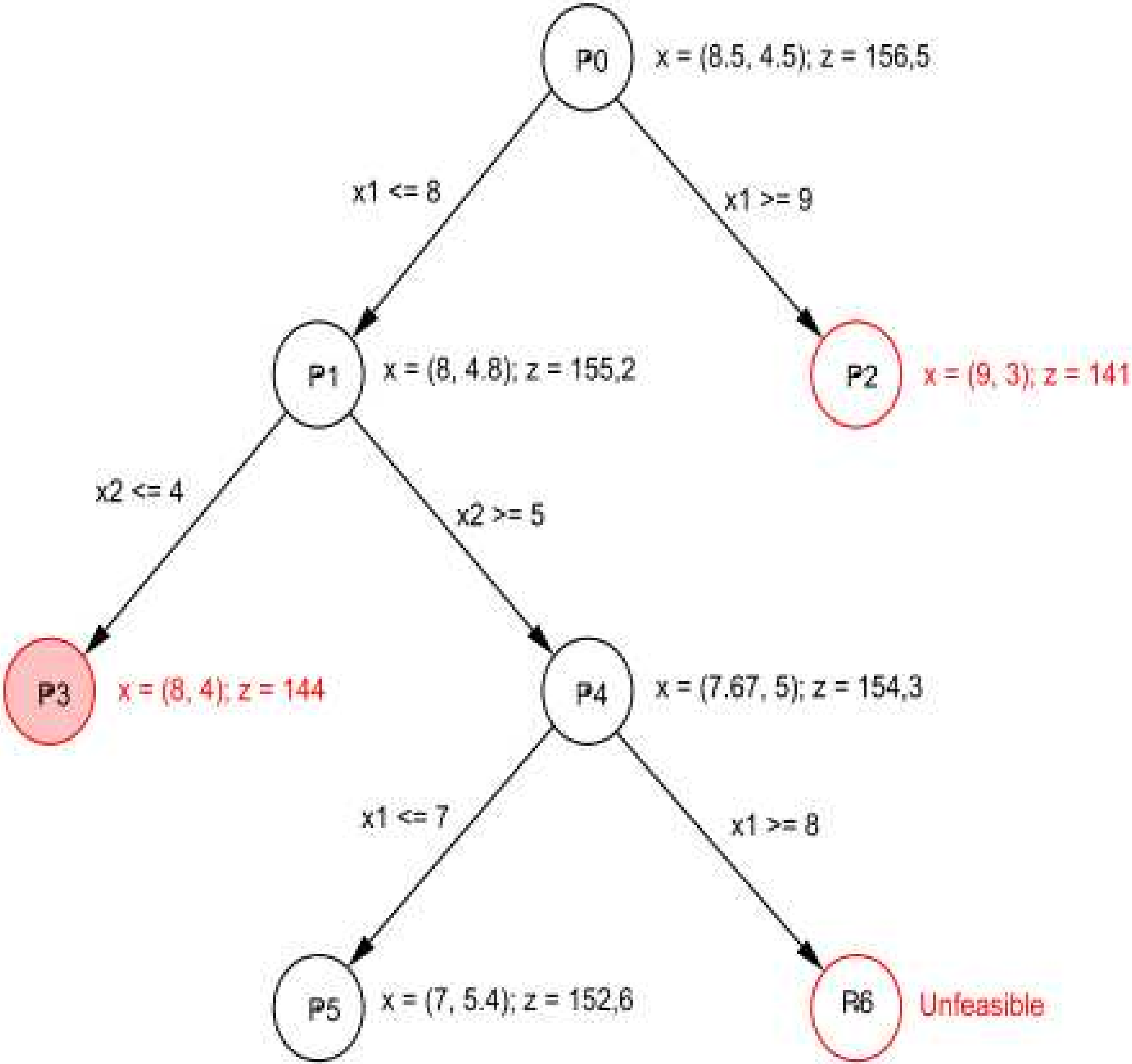
Subject to

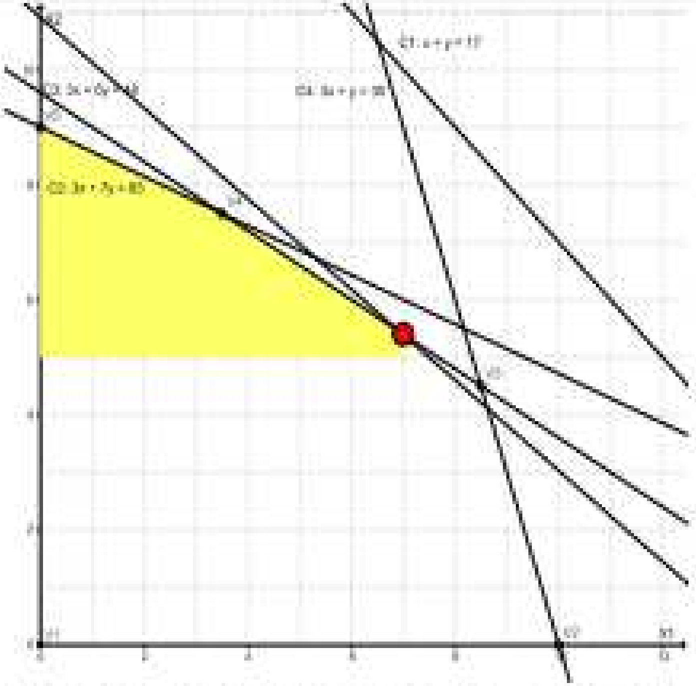
$$\begin{aligned}
1x_1 + 1x_2 &\leq 17 \\
3x_1 + 7x_2 &\leq 63 \\
3x_1 + 5x_2 &\leq 48 \\
3x_1 + 1x_2 &\leq 30 \\
x_1 \leq 8 \ \& \ x_2 &\geq 5 \ \& \ x_1 \leq 7 \\
x_1, x_2 &\geq 0 \quad x^* = (7, 5.4) \quad z^* = 152.6
\end{aligned}$$

$$Max \quad z = 11x_1 + 14x_2$$

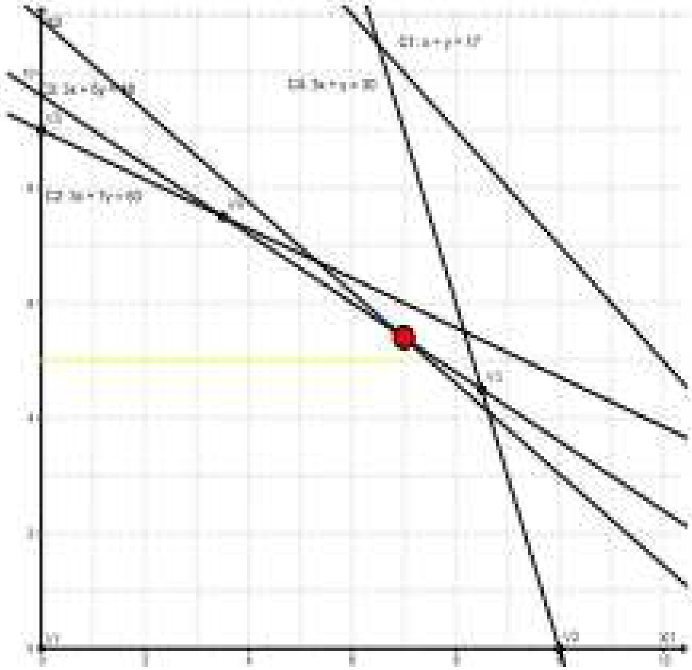
Subject to

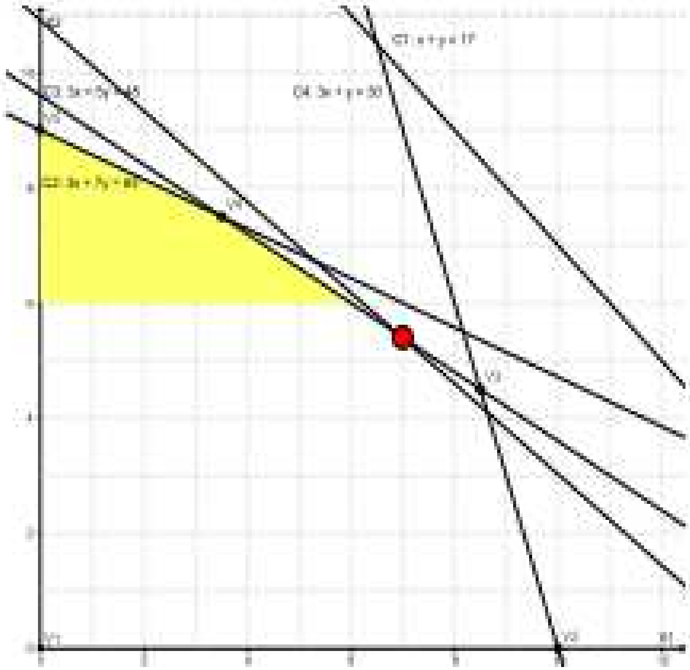
$$\begin{aligned}
1x_1 + 1x_2 &\leq 17 \\
3x_1 + 7x_2 &\leq 63 \\
3x_1 + 5x_2 &\leq 48 \\
3x_1 + 1x_2 &\leq 30 \\
x_1 \leq 8 \ \& \ x_2 &\geq 5 \ \& \ x_1 \geq 8 \\
x_1, x_2 &\geq 0 \quad \textit{Unfeasible}
\end{aligned}$$


P0
x = (8.5, 4.5); z = 156,5
x1 <= 8
x1 >= 9
P1
x = (8, 4.8); z = 155,2
P2
x = (9, 3); z = 141
x2 <= 4
x2 >= 5
P3
x = (8, 4); z = 144
P4
x = (7.67, 5); z = 154,3
x1 <= 7
x1 >= 8
P5
x = (7, 5.4); z = 152,6
R6
Unfeasible



Branching again on x_2:





$$Max \quad z = 11x_1 + 14x_2$$

Subject to

$$\begin{aligned} 1x_1 + 1x_2 &\leq 17 \\ 3x_1 + 7x_2 &\leq 63 \\ 3x_1 + 5x_2 &\leq 48 \\ 3x_1 + 1x_2 &\leq 30 \end{aligned}$$

$x_1 \leq 8$ & $x_2 \geq 5$ & $x_1 \leq 7$ & $x_2 \leq 5$

$x_1, x_2 \geq 0 \quad x^* = (7, 5) \quad z^* = 147$

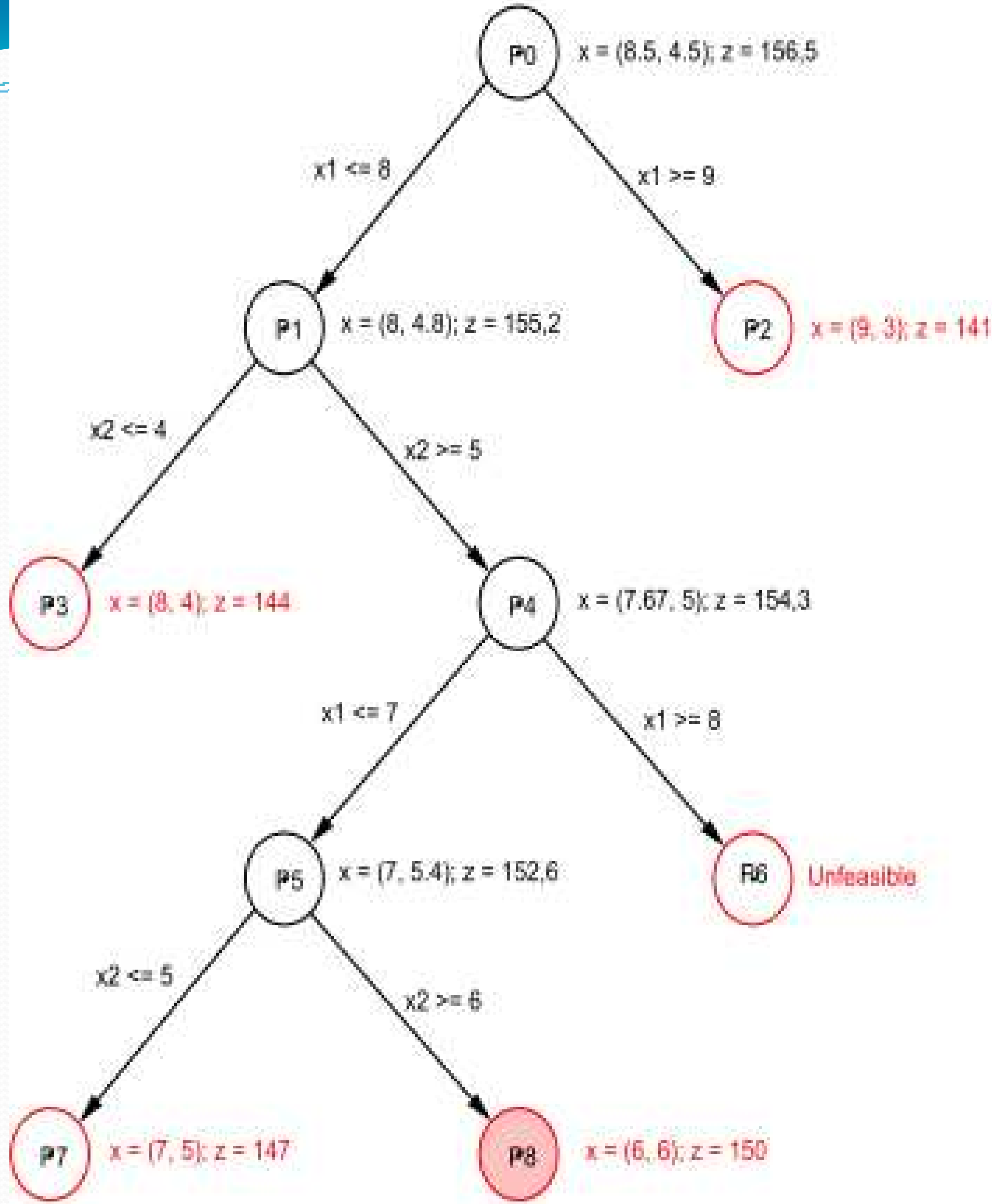
$$Max \quad z = 11x_1 + 14x_2$$

Subject to

$$\begin{aligned} 1x_1 + 1x_2 &\leq 17 \\ 3x_1 + 7x_2 &\leq 63 \\ 3x_1 + 5x_2 &\leq 48 \\ 3x_1 + 1x_2 &\leq 30 \end{aligned}$$

$x_1 \leq 8$ & $x_2 \geq 5$ & $x_1 \leq 7$ & $x_2 \geq 6$

$x_1, x_2 \geq 0 \quad x^* = (6, 6) \quad z^* = 150$


P0
x = (8.5, 4.5); z = 156,5
x1 <= 8
x1 >= 9
P1
x = (8, 4.8); z = 155,2
P2
x = (9, 3); z = 141
x2 <= 4
x2 >= 5
P3
x = (8, 4); z = 144
P4
x = (7.67, 5); z = 154,3
x1 <= 7
x1 >= 8
P5
x = (7, 5.4); z = 152,6
P6
Unfeasible
x2 <= 5
x2 >= 6
P7
x = (7, 5); z = 147
P8
x = (6, 6); z = 150

Branch and Bound for Binary IP

- Solve linear programming relaxations to bound the objective function
- Create branches by adding constraints that fix variables

$$Max \ \ z = 9x_1 + 5x_2 + 6x_3 + 4x_4$$

Subject to

$$\begin{aligned} 6x_1 + 3x_2 + 5x_3 + 2x_4 &\leq 10 \\ x_3 + x_4 &\leq 1 \\ -x_1 + x_3 &\leq 0 \\ -x_2 + x_4 &\leq 0 \\ x_i &\in \{0, 1\} \ \ \forall i = 1, \ldots, 4 \end{aligned}$$

A constraint allowing values for variables:

$$x_i \in \{0, 1\}$$

is equivalent to the following constraints:

$$\begin{aligned} x_i &\geq 0 \\ x_i &\leq 1 \\ x_i &\in \mathbb{Z} \end{aligned}$$

This correspond to an IP problem with upper bounds for all variables.

$$Max \ \ z = 9x_1 + 5x_2 + 6x_3 + 4x_4$$

Subject to

$$\begin{aligned}
6x_1 + 3x_2 + 5x_3 + 2x_4 &\leq 10 \\
x_3 + x_4 &\leq 1 \\
-x_1 + x_3 &\leq 0 \\
-x_2 + x_4 &\leq 0 \\
x_i &\leq 1 \ \forall i = 1, \ldots, 4 \\
x_i &\geq 0 \ \forall i = 1, \ldots, 4 \quad x^* = (\frac{5}{6}, 1, 0, 1); \ z^* = 16\frac{1}{2}
\end{aligned}$$


P0
x = (5/6, 1, 0, 1); z = 16 1/2

Remember that $x_i \in \{0, 1\}$ is equivalent to the following constraints:

$$
\begin{aligned}
x_i &\geq 0 \\
x_i &\leq 1 \\
x_i &\in \mathbb{Z}
\end{aligned}
$$

When branching we will add constraints $x_i \leq 0$ and $x_i \geq 1$:

$$
\begin{array}{cc}
x_i \geq 0 & x_i \geq 0 \\
x_i \leq 1 & x_i \leq 1 \\
x_i \leq 0 & x_i \geq 1 \\
x_i \in \mathbb{Z} & x_i \in \mathbb{Z} \\
\Downarrow & \Downarrow \\
x_i = 0 & x_i = 1
\end{array}
$$

Subproblem 1 (original problem $\wedge$ $x_1 = 0$):

$$
\begin{aligned}
&Max \quad z = 9x_1 + 5x_2 + 6x_3 + 4x_4 \\
\text{Subject to} \quad & 6x_1 + 3x_2 + 5x_3 + 2x_4 \leq 10 \\
& x_3 + x_4 \leq 1 \\
& -x_1 + x_3 \leq 0 \\
& -x_2 + x_4 \leq 0 \\
& x_1 = 0 \\
& x_i \in \{0, 1\} \ \forall i = 1, \ldots, 4
\end{aligned}
$$

Subproblem 2 (original problem $\wedge$ $x_1 = 1$):

$$
\begin{aligned}
&Max \quad z = 9x_1 + 5x_2 + 6x_3 + 4x_4 \\
\text{Subject to} \quad & 6x_1 + 3x_2 + 5x_3 + 2x_4 \leq 10 \\
& x_3 + x_4 \leq 1 \\
& -x_1 + x_3 \leq 0 \\
& -x_2 + x_4 \leq 0 \\
& x_1 = 1 \\
& x_i \in \{0, 1\} \ \forall i = 1, \ldots, 4
\end{aligned}
$$

Subproblem 1: $Max\ \ z = 5x_2 + 6x_3 + 4x_4$

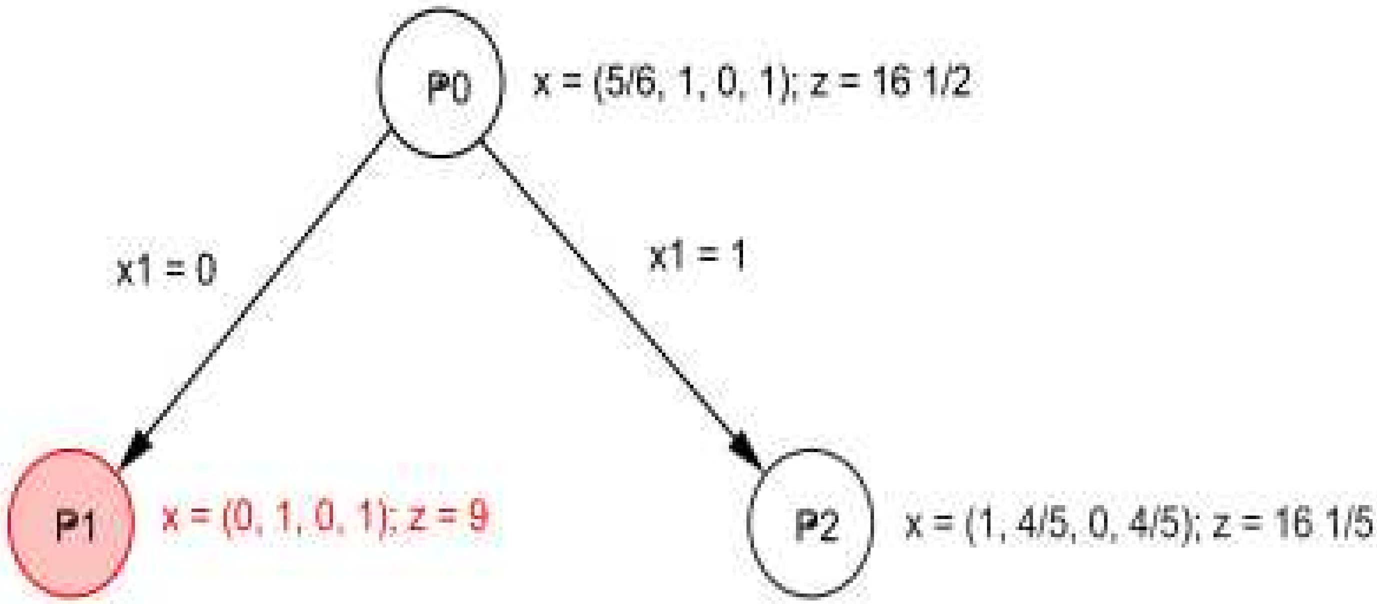
Subject to

$$\begin{aligned} 3x_2 + 5x_3 + 2x_4 &\leq 10 \\ x_3 + x_4 &\leq 1 \\ x_3 &\leq 0 \\ -x_2 + x_4 &\leq 0 \\ x_i &\leq 1\ \forall i = 2, \ldots, 4 \\ x_i &\geq 0\ \forall i = 2, \ldots, 4 \quad x^* = (0, 1, 0, 1);\ \ z^* = 9 \end{aligned}$$

Subproblem 2: $Max\ \ z = 9 + 5x_2 + 6x_3 + 4x_4$

Subject to

$$\begin{aligned} 3x_2 + 5x_3 + 2x_4 &\leq 4 \\ x_3 + x_4 &\leq 1 \\ x_3 &\leq 1 \\ -x_2 + x_4 &\leq 0 \\ x_i &\leq 1\ \forall i = 2, \ldots, 4 \\ x_i &\geq 0\ \forall i = 2, \ldots, 4 \quad x^* = (1, \frac{4}{5}, 0, \frac{4}{5});\ \ z^* = 16\frac{1}{5} \end{aligned}$$


P0
x = (5/6, 1, 0, 1); z = 16 1/2
x1 = 0
x1 = 1
P1
x = (0, 1, 0, 1); z = 9
P2
x = (1, 4/5, 0, 4/5); z = 16 1/5

Subproblem 3 (subproblem 2 $\wedge$ $x_2 = 0$):

Subject to

$$Max \;\; z = 9 + 5x_2 + 6x_3 + 4x_4$$

$$\begin{aligned} 3x_2 + 5x_3 + 2x_4 &\leq 4 \\ x_3 + x_4 &\leq 1 \\ x_3 &\leq 1 \\ -x_2 + x_4 &\leq 0 \\ x_2 &= 0 \\ x_i &\in \{0, 1\} \;\; \forall i = 2, \ldots, 4 \end{aligned}$$

Subproblem 4 (subproblem 2 $\wedge$ $x_2 = 1$):

Subject to

$$Max \;\; z = 9 + 5x_2 + 6x_3 + 4x_4$$

$$\begin{aligned} 3x_2 + 5x_3 + 2x_4 &\leq 4 \\ x_3 + x_4 &\leq 1 \\ x_3 &\leq 1 \\ -x_2 + x_4 &\leq 0 \\ x_2 &= 1 \\ x_i &\in \{0, 1\} \;\; \forall i = 2, \ldots, 4 \end{aligned}$$

Subproblem 3:
Subject to

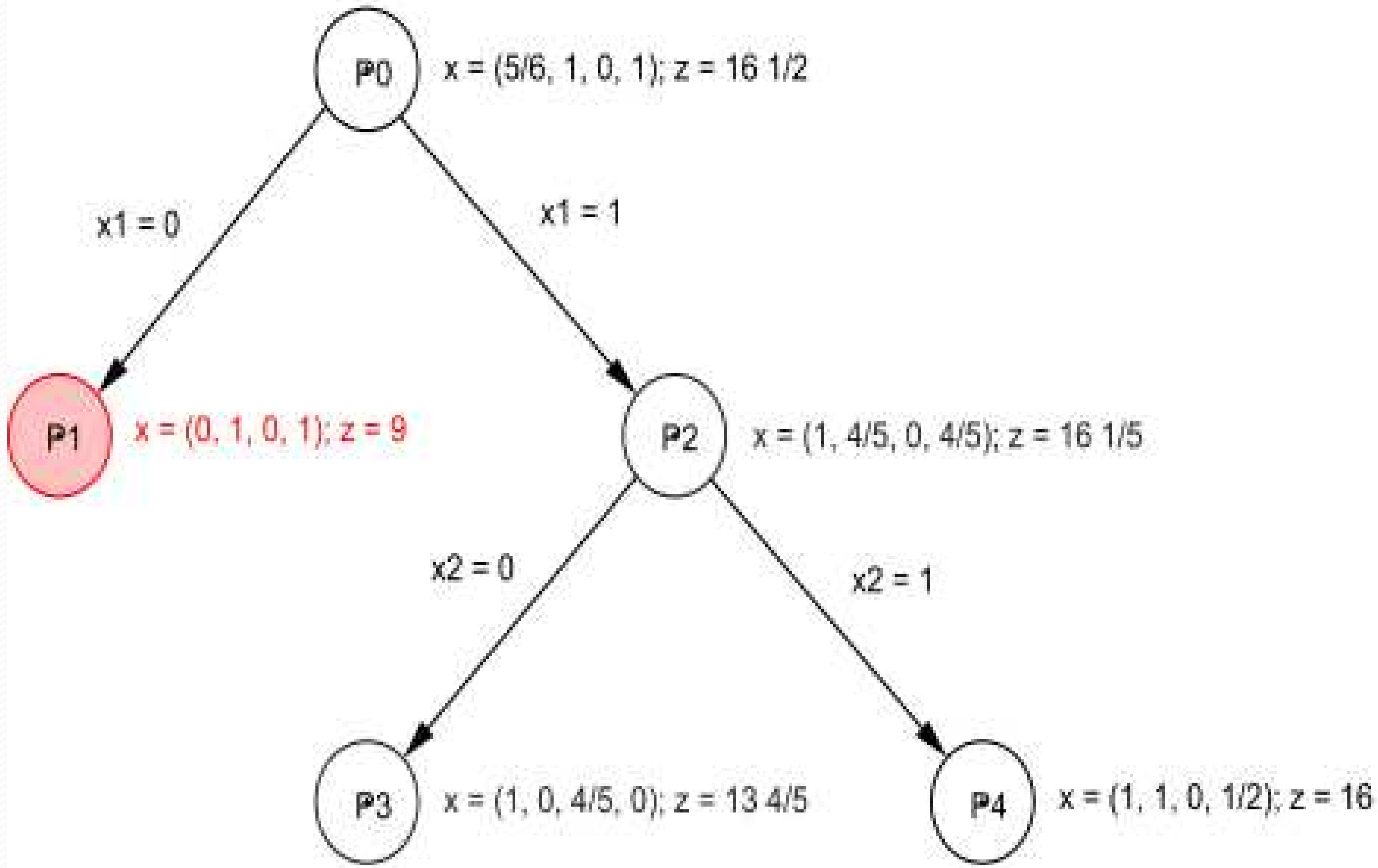
$$Max \quad z = 9 + 6x_3 + 4x_4$$

$$\begin{aligned} 5x_3 + 2x_4 &\leq 4 \\ x_3 + x_4 &\leq 1 \\ x_3 &\leq 1 \\ x_4 &\leq 0 \\ x_i &\leq 1 \ \forall i = 3,4 \\ x_i &\geq 0 \ \forall i = 3,4 \quad x^* = (1, 0, \frac{4}{5}, 0);\ z = 13\frac{4}{5} \end{aligned}$$

Subproblem 4:
Subject to

$$Max \quad z = 14 + 6x_3 + 4x_4$$

$$\begin{aligned} 5x_3 + 2x_4 &\leq 1 \\ x_3 + x_4 &\leq 1 \\ x_3 &\leq 1 \\ x_4 &\leq 1 \\ x_i &\leq 1 \ \forall i = 3,4 \\ x_i &\geq 0 \ \forall i = 3,4 \quad x^* = (1, 1, 0, \frac{1}{2});\ z^* = 16 \end{aligned}$$



P0
x = (5/6, 1, 0, 1); z = 16 1/2
x1 = 0
x1 = 1
P1
x = (0, 1, 0, 1); z = 9
P2
x = (1, 4/5, 0, 4/5); z = 16 1/5
x2 = 0
x2 = 1
P3
x = (1, 0, 4/5, 0); z = 13 4/5
P4
x = (1, 1, 0, 1/2); z = 16

Subproblem 5 (subproblem 4 $\wedge$ $x_4 = 0$):

Subject to

$$Max \quad z = 14 + 6x_3 + 4x_4$$

$$\begin{aligned}
5x_3 + 2x_4 &\leq 1 \\
x_3 + x_4 &\leq 1 \\
x_3 &\leq 1 \\
x_4 &\leq 1 \\
x_4 &= 0 \\
x_i &\in \{0, 1\} \quad \forall i = 3, 4
\end{aligned}$$

Subproblem 6 (subproblem 4 $\wedge$ $x_4 = 1$):

Subject to

$$Max \quad z = 14 + 6x_3 + 4x_4$$

$$\begin{aligned}
5x_3 + 2x_4 &\leq 1 \\
x_3 + x_4 &\leq 1 \\
x_3 &\leq 1 \\
x_4 &\leq 1 \\
x_4 &= 1 \\
x_i &\in \{0, 1\} \quad \forall i = 3, 4
\end{aligned}$$

Subproblem 5:

Subject to

$$Max \;\; z = 14 + 6x_3$$

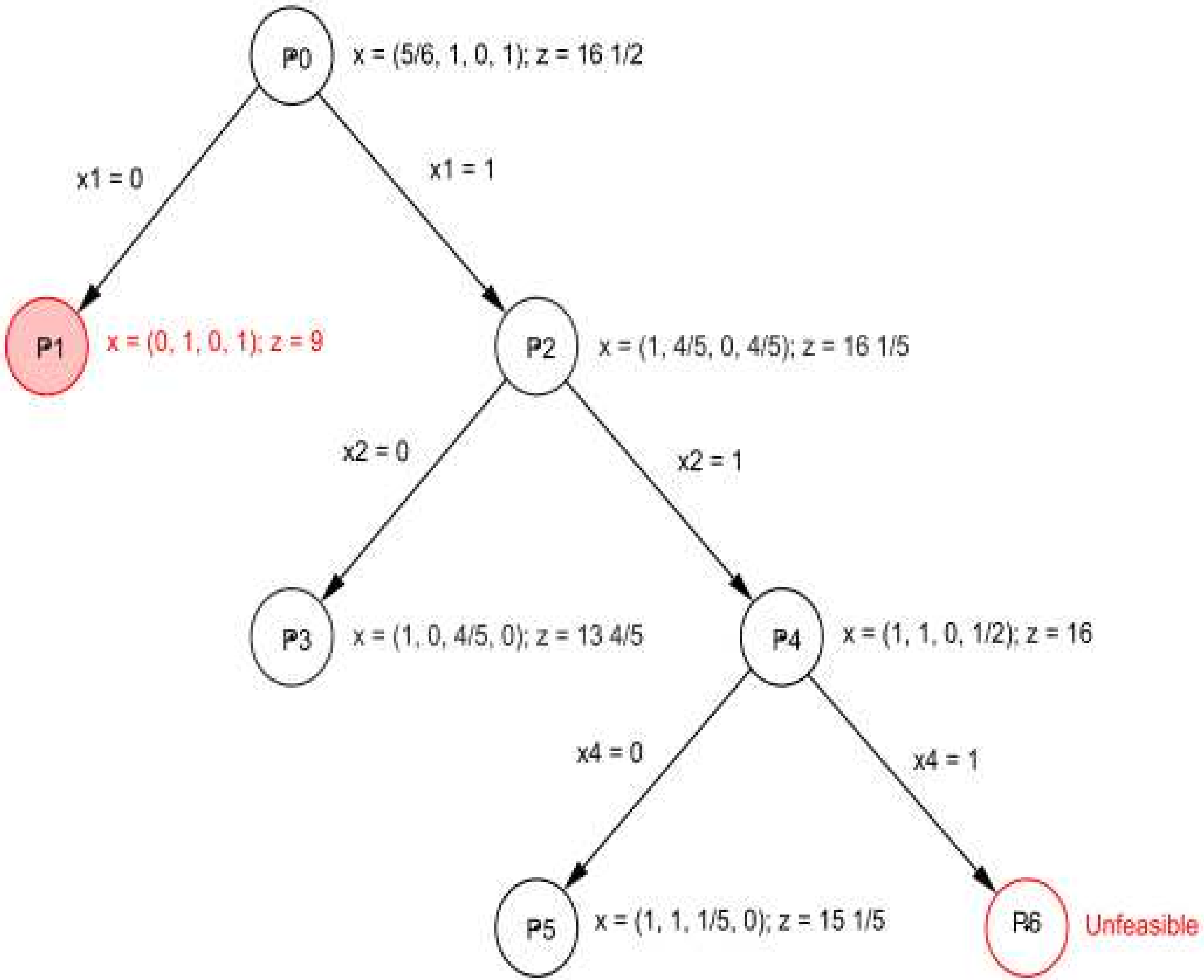
$$\begin{aligned} 5x_3 &\leq 1 \\ x_3 &\leq 1 \\ x_3 &\leq 1 \\ x_3 &\geq 0 \quad x^* = (1, 1, \tfrac{1}{5}, 0);\; z^* = 15\tfrac{1}{5} \end{aligned}$$

Subproblem 6:

Subject to

$$Max \;\; z = 18 + 6x_3$$

$$\begin{aligned} 5x_3 &\leq -1 \\ x_3 &\leq 0 \\ x_3 &\leq 1 \\ x_3 &\leq 1 \\ x_3 &\geq 0 \quad \textit{Unfeasible} \end{aligned}$$


P0
x = (5/6, 1, 0, 1); z = 16 1/2
x1 = 0
x1 = 1
P1
x = (0, 1, 0, 1); z = 9
P2
x = (1, 4/5, 0, 4/5); z = 16 1/5
x2 = 0
x2 = 1
P3
x = (1, 0, 4/5, 0); z = 13 4/5
P4
x = (1, 1, 0, 1/2); z = 16
x4 = 0
x4 = 1
P5
x = (1, 1, 1/5, 0); z = 15 1/5
P6
Unfeasible

Subproblem 7 (subproblem 5 $\wedge$ $x_3 = 0$):

Subject to

$$
\begin{aligned}
Max \;\; z = 14 + 6x_3 & \\
5x_3 &\leq 1 \\
x_3 &\leq 1 \\
x_3 &\leq 1 \\
x_3 &\geq 0 \\
x_3 &= 0 \\
x_3 &\in \{0, 1\}
\end{aligned}
$$

Subproblem 8 (subproblem 5 $\wedge$ $x_3 = 1$):

Subject to

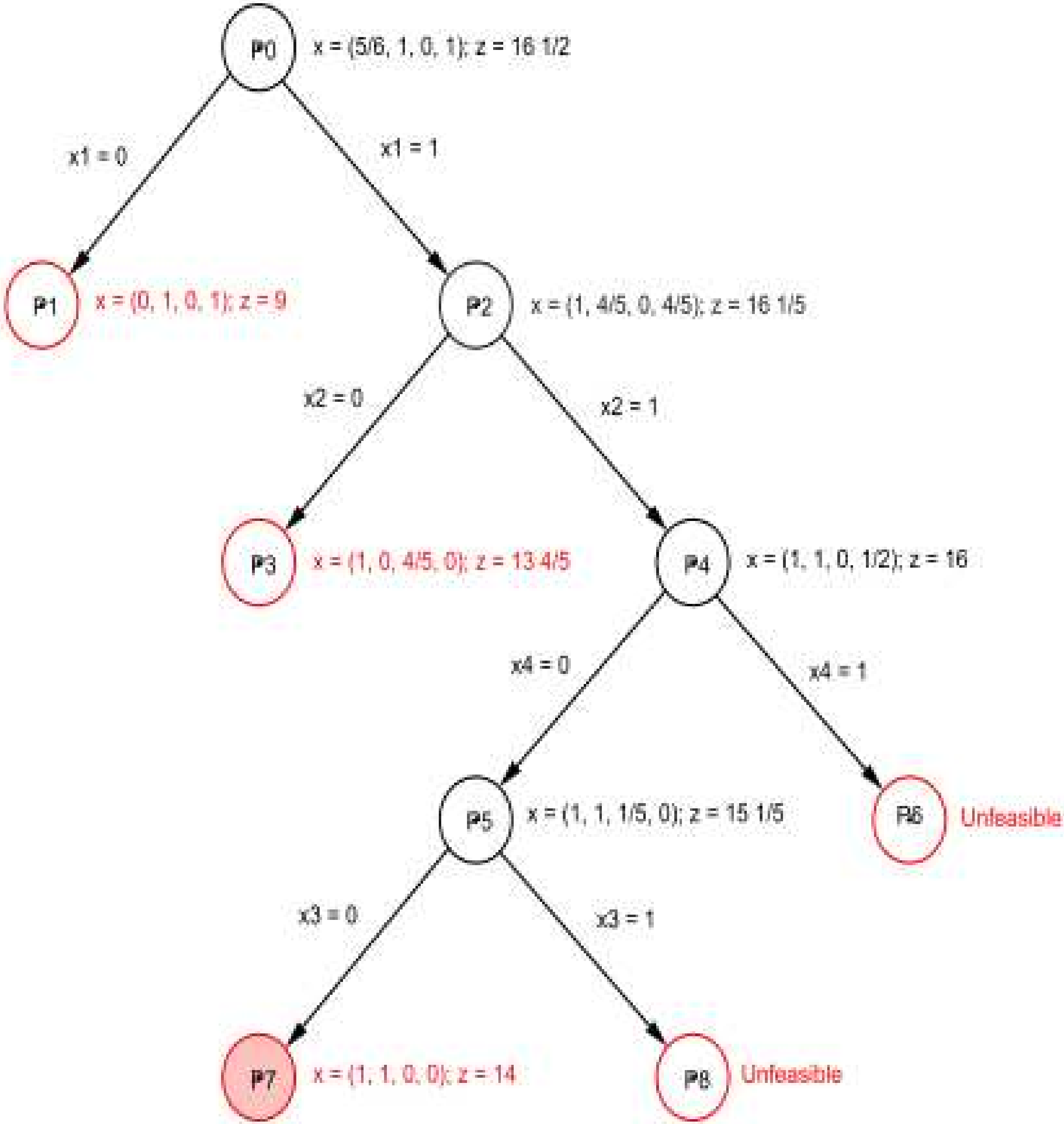
$$
\begin{aligned}
Max \;\; z = 14 + 6x_3 & \\
5x_3 &\leq 1 \\
x_3 &\leq 1 \\
x_3 &\leq 1 \\
x_3 &\geq 0 \\
x_3 &= 1 \\
x_3 &\in \{0, 1\}
\end{aligned}
$$

Subproblem 7:

- Variable enumeration: $x = (1, 1, 0, 0)$
- Feasible solution
- $z = 14$

Subproblem 8:

- Variable enumeration: $x = (1, 1, 1, 0)$
- Unfeasible solution



P0
x = (5/6, 1, 0, 1); z = 16 1/2
x1 = 0
x1 = 1
P1
x = (0, 1, 0, 1); z = 9
P2
x = (1, 4/5, 0, 4/5); z = 16 1/5
x2 = 0
x2 = 1
P3
x = (1, 0, 4/5, 0); z = 13 4/5
P4
x = (1, 1, 0, 1/2); z = 16
x4 = 0
x4 = 1
P5
x = (1, 1, 1/5, 0); z = 15 1/5
P6
Unfeasible
x3 = 0
x3 = 1
P7
x = (1, 1, 0, 0); z = 14
P8
Unfeasible

Βελτιστοποίηση συναρτήσεων με πολλά κριτήρια
Multi-objective optimization

Introduction

- Is an area of multiple criteria decision making, that is concerned with mathmatical optimization problem involving more than one objective function to be optimized simultaneously.
- Most real-world engineering optimization

problems are multi-Objective in nature

- Objectives are often conflicting
  - Performance vs. Silicon area
  - Quality vs. Cost
  - Efficiency vs. Portability
- The notion of ”*optimum*” has to be redefined

- Multiobjective optimization (multicriteria, multiperformance, vector optimization or Pareto optimization)
  - Find a vector of decision variables which satisfies constraints and optimizes a vector function whose elements represent the objective functions
  - Objectives are usually in conflict with each other
  - **Optimize**: finding solutions which would give the values of all the objective functions acceptable to the designer

Mathematical Formulation

Find the vector

$$\bar{x}^* = \left[x_1^*, x_2^*, \ldots, x_n^*\right]^T$$

Which will satisfy the *m* inequality constraints

$$g_i(\bar{x}) \geq 0 \quad i = 1,2,\ldots,m$$

The *p* equality constraints

$$h_i(\bar{x}) = 0 \quad i = 1,2,\ldots,p$$

And optimizes the vector function

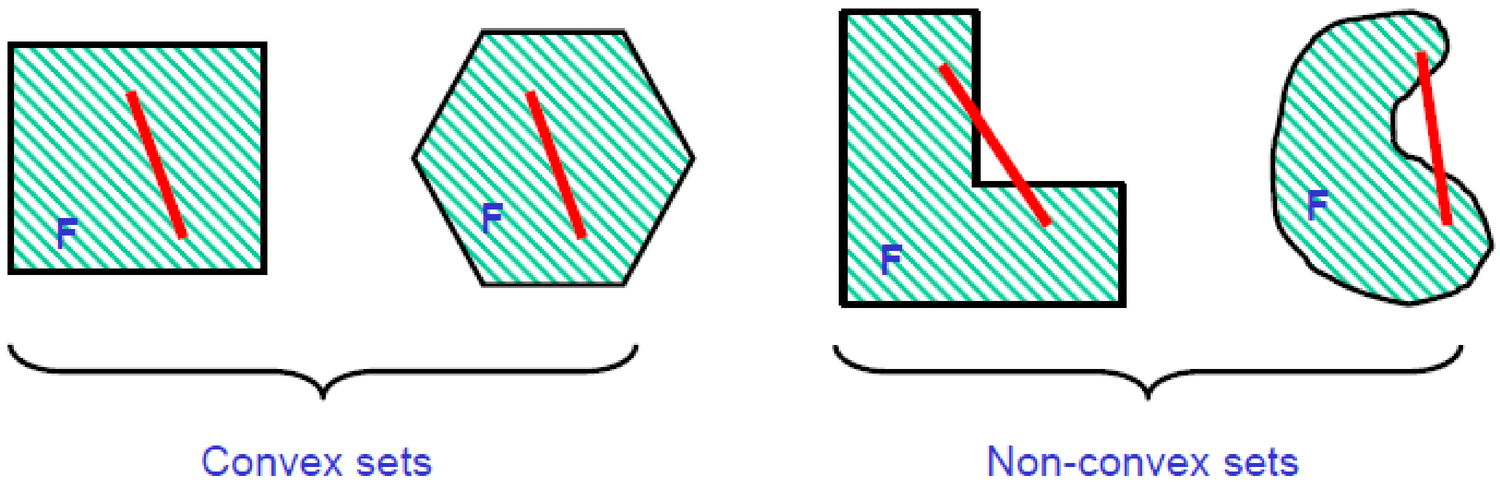
$$\bar{f}(\bar{x}) = [f_1(\bar{x}), f_2(\bar{x}), \ldots, f_k(\bar{x})]^T$$

Feasible Region

$$g_i(\bar{x}) \geq 0 \quad i = 1,2,\ldots,m$$

$$h_i(\bar{x}) = 0 \quad i = 1,2,\ldots,p$$

Define the *feasible region F*



Pareto Optimum

Formulated by Vilfredo Pareto

Vilfredo Pareto 1848-1923

A point $\bar{x}^* \in F$ is *Pareto optimal* if for every $\bar{x} \in F$ either

$$f_i(\bar{x}) = f_i(\bar{x}^*), \quad i = 1,2,\ldots,k$$

or, there is at least one $i \in \{1,2,\ldots,k\}$ such that

$$f_i(\bar{x}) > f_i(\bar{x}^*)$$

Pareto Optimum

In words, this definition says that $\bar{x}^*$ is *Pareto optimal* if there exists no feasible vector of decision variables $\bar{x}^* \in F$ which would decrease some criterion without causing a simultaneous increase in at least one other criterion

Vilfredo Pareto 1848-1923

Pareto Optimum

A solution $x \in F$ is said to dominate $y \in F$ if

- ➔ **x** is better or equal to **y** in all attributes
- ➔ **x** is strictly better than **y** in at least one attribute

Vilfredo Pareto 1848-1923

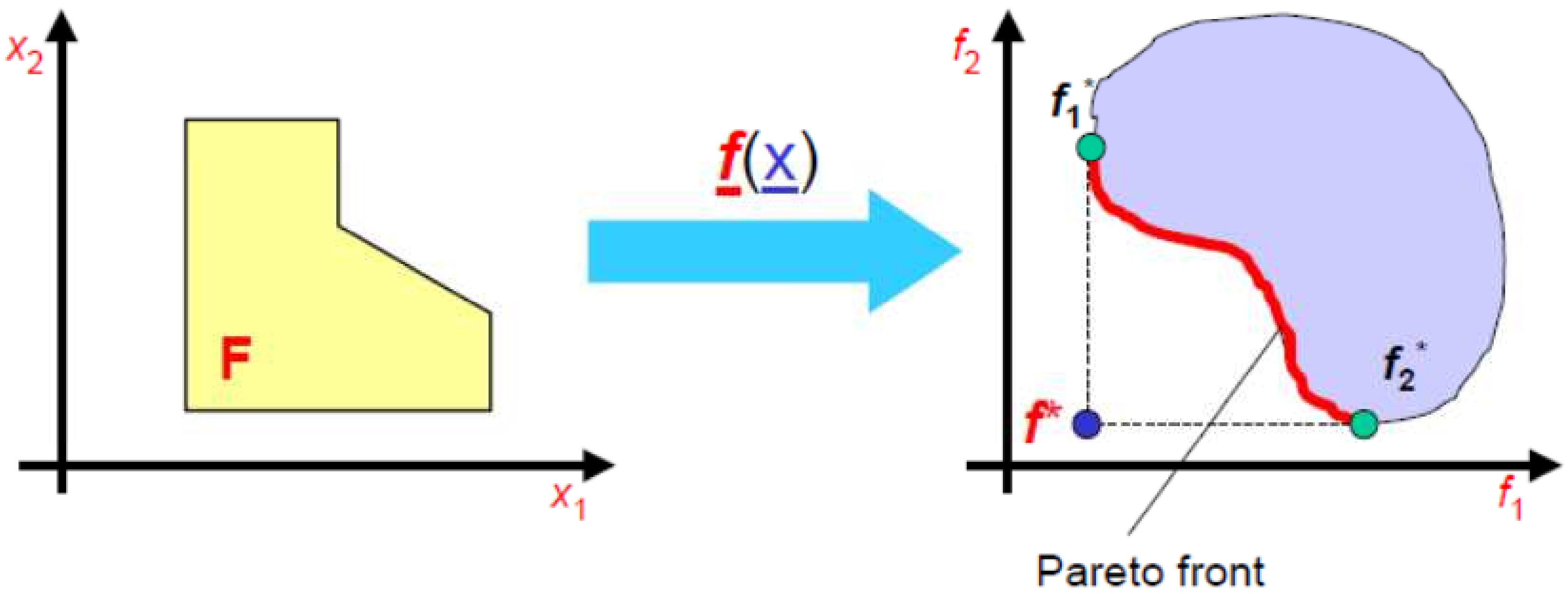
Formally, **x** dominate **y** ($x \succ y$)

$$f_i(\bar{x}) \le f_i(\bar{y}), \quad i = 1,2,\ldots,k$$

$$\exists\, j \in \{1,2,\ldots,k\} : f_j(\bar{x}) < f_j(\bar{y})$$

The *Pareto set* consists of solutions that are not dominated by any other solutions

Pareto Front



Currently, there are over 30 mathematical programming techniques for multiobjective optimization

However, these techniques tend to generate elements of the Pareto optimal set one at a time

Additionally, most of them are very sensitive to the shape of the Pareto front (e.g., they do not work when the Pareto front is concave or when the front is disconnected)

Multi-Objective Optimization

Classic Methods :

1- Weighted Sum Method

2- Constraint method

3- Weighted Metric Methods

4- Rotated Weighted Metric Method

5- Benson's Method

5- Value Function Method

- Currently an Evolutionary Algorithm Methods are Used For MOOP

Multi-Objective Optimization

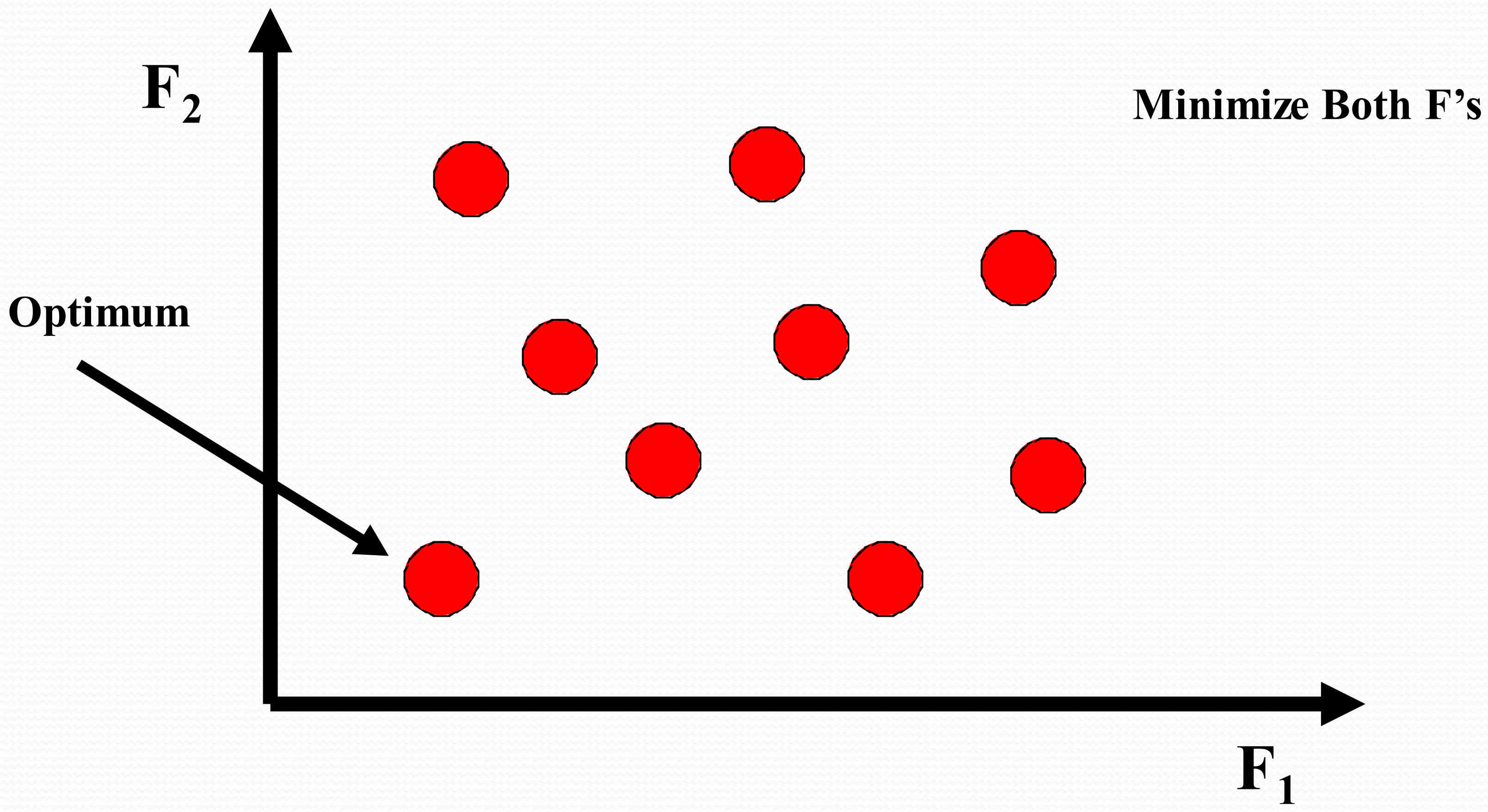
- All the problems that we have considered in this class have been comprised of a single objective function with perhaps multiple constraints and design variables.

Minimize $$F(\bar{x})$$

Subject To: $$\bar{g}(\bar{x}) \leq 0 \;\ldots.$$

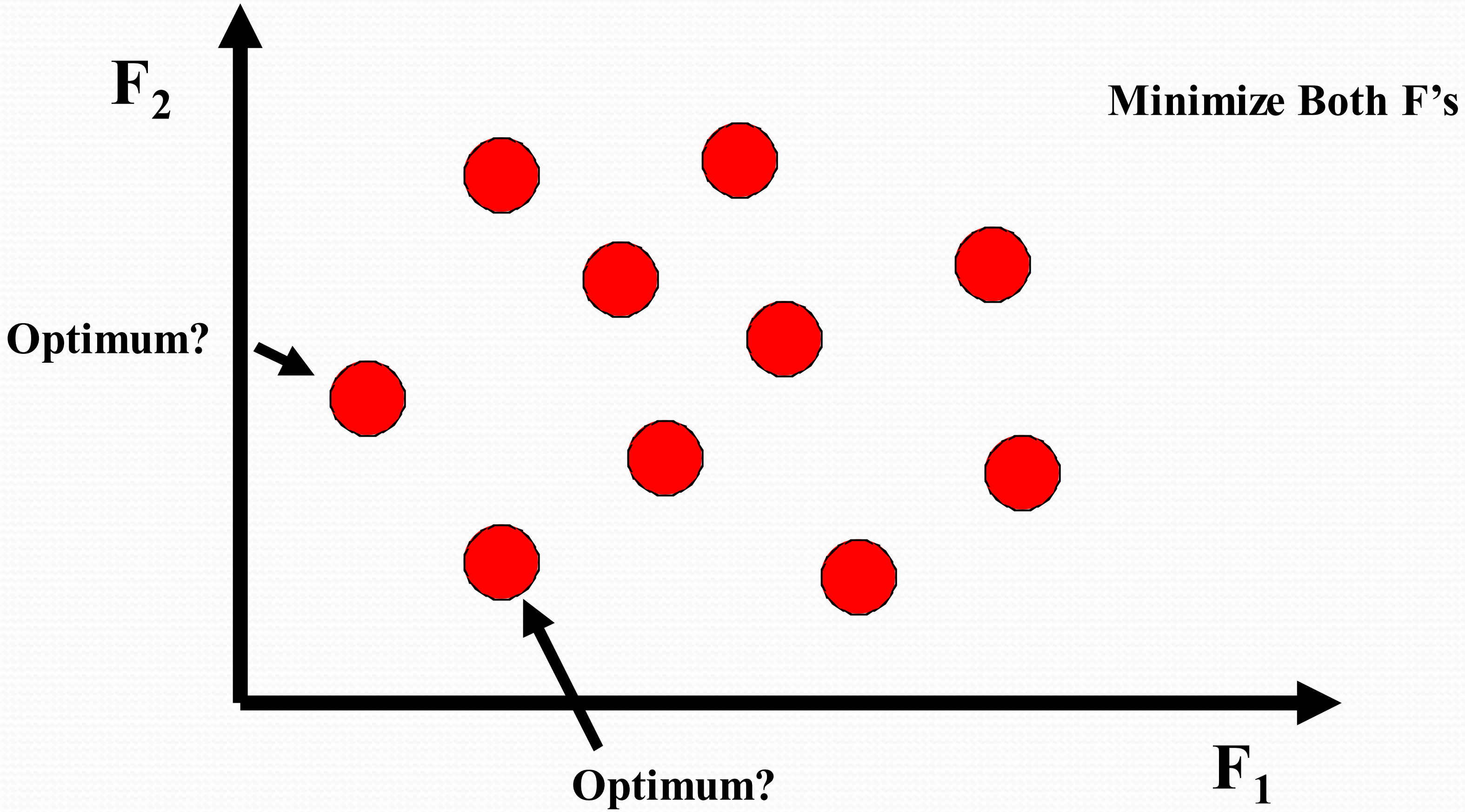
Multi-Objective Optimization

Consider the following 2D performance space:



Multi-Objective Optimization

But what happens in a case like this:



Multi-Objective Optimization

Suppose I wish to minimize both of the following functions simultaneously:

$F_1 = 750x_1+60(25-x_1)\,x_2+45(25-x_1)(25-x_2)$

$F_2 = (25-x_1)\,x_2$

For the typical weighted sum approach, I would assign a weight to each function such that:

$$w_1 + w_2 = 1 \qquad and \qquad w_1, w_2 \geq 0$$

Multi-Objective Optimization

I would then combine the two functions into a single function as follows and solve:

$$F_T = \sum_i w_i F_i$$

$$= w_1 F_1 + w_2 F_2$$

Multi-Objective Optimization

The net effect of our weighted sum approach is to convert a multiple objective problem into a single objective problem.

But this will only provide us with a single Pareto point. How will be go about finding other Pareto points?

By altering the weights and solving again.

Multi-Objective Optimization

Ok, so I march up and down my weights generating Pareto points and then I've got a good representation of my set.

Unfortunately not. As it turns out it is seldom this easy. There are a number of pitfalls associated with using weighted sums to generate Pareto points.

Multi-Objective Optimization

Some of those pitfalls are:

- Inability to generate points in non-convex portions of the frontier
- Inability to generate a uniform sampling of the frontier
- A non-intuitive relationship between combinatorial parameters (weights, etc.) and performances
- Poor efficiency (can require an excessive number of function evaluations).

Methods with a priori articulation of preferences

Allow user to specify preferences for, or relative importance of, objective functions

Weighted Sum Method

$$U = \sum_{i=1}^{k} w_i F_i(\mathbf{x})$$

Sufficient for Pareto optimality

no guarantee of final result acceptable
impossible to find points in non-convex sections
not even distribution

Lexicographic Method

objective functions arranged in order of importance

solve following optimization problems one at a time

$$\underset{\mathbf{x}\in\mathbf{X}}{\text{Minimize}} \; F_i(\mathbf{x})$$

subject to $F_j(\mathbf{x}) \le F_j(\mathbf{x}_j^*)$, $j = 1, 2, \ldots, i-1$, $i > 1$,

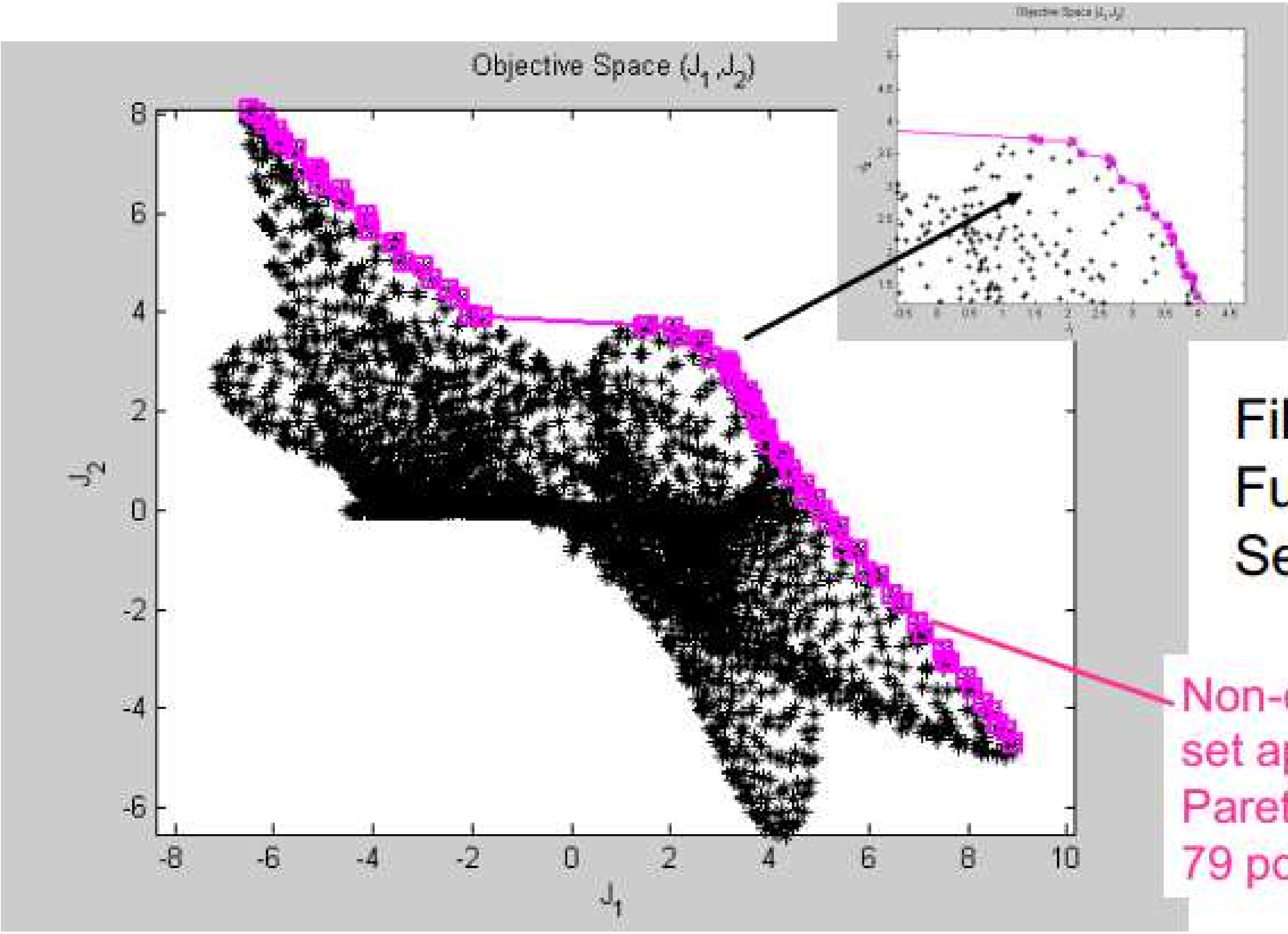
$i = 1, 2, \ldots, k$.

Pareto optimal solution

if there does not exist another feasible design objective vector such that all objective functions are better than or equal to and at least one objective function is better

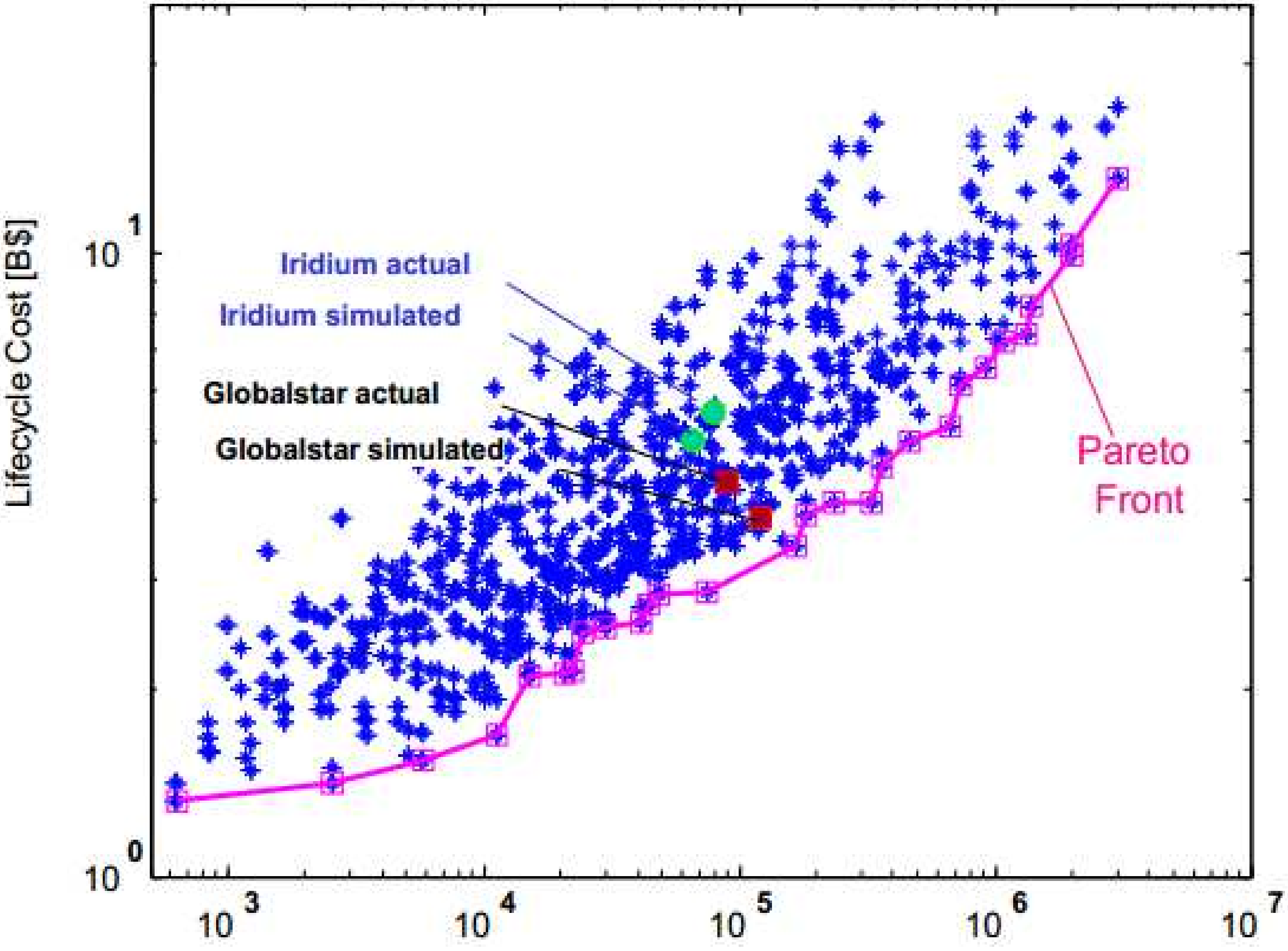
i.e., there is no x' such that x' > x

i.e., it is not dominated by any other point

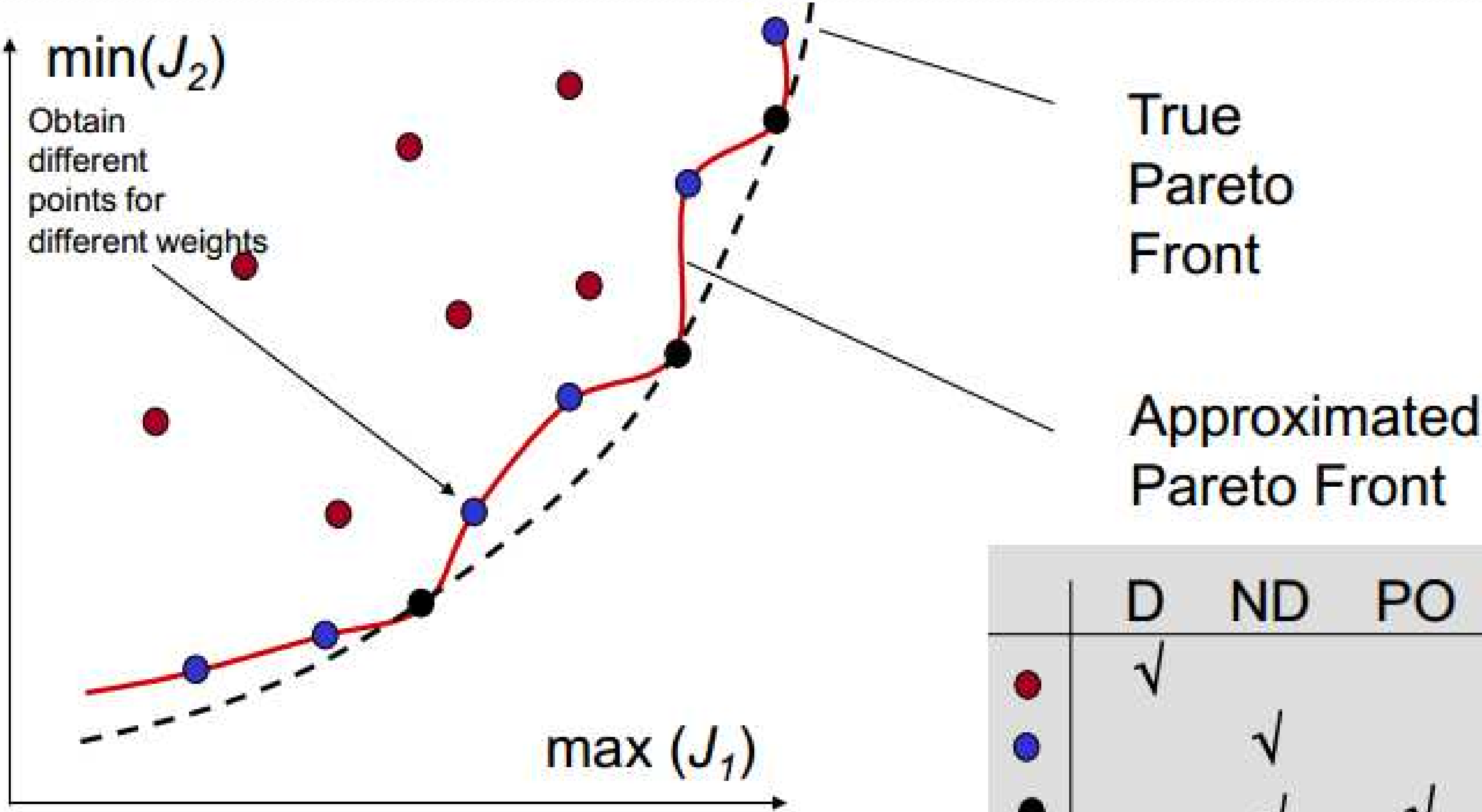


Filtered the Full Factorial Set: 3721

Non-dominated set approximates Pareto frontier: 79 points (2.1%)



Lifecycle Cost [B$]
Iridium actual
Iridium simulated
Globalstar actual
Globalstar simulated
Pareto Front
10^1
10^0
10^3
10^4
10^5
10^6
10^7



	D	ND	PO
● (red)	√		
● (blue)		√	
● (black)		√	√

All pareto optimal points are non-dominated
Not all non-dominated points are pareto-optimal

It's easier to show dominatedness than non-dominatedness !!!

- A multiobjective problem has more than one optimal solution
- All points on Pareto Front are non-dominated
- Methods:
  - Weighted Sum Approach (Caution: Scaling !)
  - Pareto-Filter Approach