Optimisation Models: exercises

Dipartimento di Ingegneria e Architettura
Università degli studi di Trieste

23/4/2021, Trieste

Exercise 1: Manpower planning

Exercise 1: Manpower planning

- 3 projects must be completed in 6 months

Exercise 1: Manpower planning

- 3 projects must be completed in 6 months
- project 1 takes 3 **consecutive** months: in its first month it uses 3 people, in its second, 4 people, and in its third, 2 people. **Once completed** it brings a profit of 10.2K€ per month.

Project	Duration	Profile	Gain/month
1	3	3,4,2	10.2
2	3	4,1,5	12.3
3	4	3,2,1,2	11.2

Exercise 1: Manpower planning

- 3 projects must be completed in 6 months
- project 1 takes 3 **consecutive** months: in its first month it uses 3 people, in its second, 4 people, and in its third, 2 people. **Once completed** it brings a profit of 10.2K€ per month.

Project	Duration	Profile	Gain/month
1	3	3,4,2	10.2
2	3	4,1,5	12.3
3	4	3,2,1,2	11.2

Month	1	2	3	4	5	6
Personnel	5	6	7	7	6	6

- for each month, it is not possible to use more manpower than is available

Exercise 1: Manpower planning

- 3 projects must be completed in 6 months
- project 1 takes 3 **consecutive** months: in its first month it uses 3 people, in its second, 4 people, and in its third, 2 people. **Once completed** it brings a profit of 10.2K€ per month.

Project	Duration	Profile	Gain/month
1	3	3,4,2	10.2
2	3	4,1,5	12.3
3	4	3,2,1,2	11.2

Month	1	2	3	4	5	6
Personnel	5	6	7	7	6	6

- for each month, it is not possible to use more manpower than is available

- maximise the total benefit obtained through the projects **once they are finished**.

Notation

Notation

$$
\begin{aligned}
\mathcal{P} &: \text{set of projects} \\
\mathcal{M} &: \text{set of months } = 1, \ldots, nM \\
D_p &: \text{duration of project } p \\
G_p &: \text{monthly benefit once project } p \text{ is finished} \\
CAP_m &: \text{available manpower for month } m
\end{aligned}
$$

Notation

$\mathcal{P}$: set of projects

$\mathcal{M}$: set of months $= 1, \ldots, nM$

D_p : duration of project p

G_p : monthly benefit once project p is finished

CAP_m : available manpower for month m

$RU_{p,k}$: resource usage of project p in **it's** k-th month

Decision variables

Decision variables

$$x_{p,m} = \begin{cases} 1 & \text{if project } p \text{ starts in month } m \\ 0 & \text{otherwise} \end{cases} \qquad \forall p \in \mathcal{P}, m \in \mathcal{M}$$

Decision variables

$$x_{p,m} = \begin{cases} 1 & \text{if project } p \text{ starts in month } m \\ 0 & \text{otherwise} \end{cases} \qquad \forall p \in \mathcal{P}, m \in \mathcal{M}$$

$start_p$: starting month of project p

Decision variables constraints

Decision variables constraints

$$\sum_{m \in \mathcal{M}} x_{p,m} = 1 \qquad \forall p \in \mathcal{P}$$

Decision variables constraints

$$\sum_{m \in \mathcal{M}} x_{p,m} = 1 \qquad \forall p \in \mathcal{P}$$

$$\sum_{m \in \mathcal{M}} m \cdot x_{p,m} = start_p \qquad \forall p \in \mathcal{P}$$

Decision variables constraints

$$\sum_{m \in \mathcal{M}} x_{p,m} = 1 \qquad \forall p \in \mathcal{P}$$

$$\sum_{m \in \mathcal{M}} m \cdot x_{p,m} = start_p \qquad \forall p \in \mathcal{P}$$

$$start_p \leq nM - D_p + 1 \qquad \forall p \in \mathcal{P}$$

Decision variables constraints

```
! Each project starts once and only once
 forall(p in P) sum(m in M) x(p,m) = 1

! Connect variables x(p,t) and start(p)
 forall(p in P) sum(m in M) m*x(p,m) = start(p)

! Finish everything by the end of the planning period
 forall(p in P) start(p) <= nM-D(p)+1
```

Resource constraints

Resource constraints

What is the resource usage of project p in a given month m ($\mathbf{r_p(m)}$)?

Resource constraints

What is the resource usage of project p in a given month m ($\mathbf{r_p(m)}$)?
It depends on $start_p$:

if $\mathbf{start_p \leq m}$, in month m, p is in its $(\mathbf{m - start_p + 1})$-st month

Resource constraints

What is the resource usage of project p in a given month m ($\mathbf{r_p(m)}$)?
It depends on $start_p$:

if $\mathbf{start_p} \leq \mathbf{m}$, in month m, p is in its $(\mathbf{m} - \mathbf{start_p} + \mathbf{1})$-st month

$$r_p(m) = \sum_{s=1}^{m} RU_{p,m-s+1} \cdot x_{p,s}$$

Resource constraints

What is the resource usage of project p in a given month m ($\mathbf{r_p(m)}$)?
It depends on $start_p$:

if $\mathbf{start_p} \leq \mathbf{m}$, in month m, p is in its $(\mathbf{m} - \mathbf{start_p} + \mathbf{1})$-st month

$$r_p(m) = \sum_{s=1}^{m} RU_{p,m-s+1} \cdot x_{p,s}$$

$$\sum_{p \in \mathcal{P}} r_p(m) \leq CAP_m \qquad \forall m \in \mathcal{M}$$

Resource constraints

What is the resource usage of project p in a given month m ($\mathbf{r_p(m)}$)?
It depends on $start_p$:

if $\mathbf{start_p} \leq \mathbf{m}$, in month m, p is in its $(\mathbf{m} - \mathbf{start_p} + \mathbf{1})$-st month

$$r_p(m) = \sum_{s=1}^{m} RU_{p,m-s+1} \cdot x_{p,s}$$

$$\sum_{p \in \mathcal{P}} r_p(m) \leq CAP_m \qquad \forall m \in \mathcal{M}$$

```
! Resource availability
 forall(m in M)
  sum(p in P,s in 1..m) RU(p,m-s+1)*x(p,s) <= CAP(m)
```

Objective function

Objective function

For how many months project p contributes to the profit? ($\mathbf{n_p}$)

Objective function

For how many months project p contributes to the profit? $(\mathbf{n_p})$
Project p ends in the $(\mathbf{start_p} + \mathbf{D_p} - \mathbf{1})$-th month

Objective function

For how many months project p contributes to the profit? ($\mathbf{n_p}$)
Project p ends in the ($\mathbf{start_p + D_p - 1}$)-th month

$$n_p = nM - (start_p + D_p - 1)$$

Objective function

For how many months project p contributes to the profit? ($\mathbf{n_p}$)
Project p ends in the $(\mathbf{start_p} + \mathbf{D_p} - \mathbf{1})$-th month

$$n_p = nM - (start_p + D_p - 1)$$

$$max \sum_{p \in \mathcal{P}} G_p \cdot n_p$$

Objective function

For how many months project p contributes to the profit? ($\mathbf{n_p}$)
Project p ends in the ($\mathbf{start_p + D_p - 1}$)-th month

$$n_p = nM - (start_p + D_p - 1)$$

$$max \sum_{p \in \mathcal{P}} G_p \cdot n_p$$

```
! Objective: Maximize Benefit
MaxBen:=
 sum(p in P) (G(p)*(nM-start(p)-D(p)+1))

! Solve the problem
maximize(MaxBen)
```

SOS1

Special Ordered Sets of type 1: an ordered set of non-negative variables at most **one** of which can take a non-zero value.

- the **order** is specified by assigning weights (*reference row values*) to each variable
- used for branching on **sets** of variables, rather than individual variables
- the search procedure will generally be noticeably faster

SOS1

Special Ordered Sets of type 1: an ordered set of non-negative variables at most **one** of which can take a non-zero value.

- the **order** is specified by assigning weights (*reference row values*) to each variable
- used for branching on **sets** of variables, rather than individual variables
- the search procedure will generally be noticeably faster

```
if not USESOS then  ! Turn variables x into binaries
 forall(p in P,m in M) x(p,m) is_binary
else                ! Define SOS-1 sets
 forall(p in P) sum(m in M) m*x(p,m) is_sos1
end-if
```

SOS1

Standard branching

- "1-branch" (**strong decision**) eliminates a very large number of possible solutions
- "0-branch" (**weak decision**) eliminates only a few
- **unbalanced** tree

SOS1

Standard branching

- "1-branch" (**strong decision**) eliminates a very large number of possible solutions
- "0-branch" (**weak decision**) eliminates only a few
- **unbalanced** tree

SOS1 branching

- all the variables in one subset are set to 0
- any combination of the variables in the other subset sums to 1
- **strong decision deferred**
- unpromising variables placed in the new "set-to-0" subset
- **better balanced** and **smaller** tree

Exercise 2: Purchasing with price breaks

Exercise 2: Purchasing with price breaks

- 3 suppliers offer decreasing prices for increased lot size (incremental discounts)

Exercise 2: Purchasing with price breaks

- 3 suppliers offer decreasing prices for increased lot size (incremental discounts)
- we wish to buy 600 items in total
- we want to buy at least total cost, yet not buy too much from any one supplier

Exercise 2: Purchasing with price breaks

- 3 suppliers offer decreasing prices for increased lot size (incremental discounts)
- we wish to buy 600 items in total
- we want to buy at least total cost, yet not buy too much from any one supplier

Supplier	Max%	Unit Price	Breakpoint	UP	BP	UP	BP
1	40%	9.2	100	9	200	7	1000
2	35%	9	50	8.5	250	8.3	2000
3	40%	11	100	8.5	300	7.5	4000

- if you buy 220 items from supplier 1, you pay $9.2 \cdot \mathbf{100} + 9 \cdot \mathbf{100} + 7 \cdot \mathbf{20}$. We can buy at most 40% of the requirement from supplier (240 items)

Notation

Notation

$\mathcal{S}$: set of suppliers

$\mathcal{R}$: set of price ranges

$COST_{s,r}$: unit cost of price range r for supplier s

$B_{s,r}$: breakpoint where the unit cost change from $COST_{s,r}$ to $COST_{s,r+1}$ for supplier s

T : total amount we wish to buy

Q_s : maximum percentage that may be bought from supplier s

Decision variables

Decision variables

$x_{s,r}$: number of items bought from suppliers s at the price range r

Decision variables

$x_{s,r}$: number of items bought from suppliers s at the price range r

$$y_{s,r} = \begin{cases} 1 & \text{if we have bought any items from} \\ & \quad \text{supplier } s \text{ at the price range } r \\ 0 & \text{otherwise} \end{cases} \qquad \forall s \in \mathcal{S}, r \in \mathcal{R}$$

Objective function

Objective function

$$min \sum_{s \in \mathcal{S}, r \in \mathcal{R}} x_{s,r} \cdot COST_{s,r}$$

Constraints

Constraints

$$\sum_{s \in \mathcal{S}, r \in \mathcal{R}} x_{s,r} \geq T$$

Constraints

$$\sum_{s \in \mathcal{S}, r \in \mathcal{R}} x_{s,r} \geq T$$

$$\sum_{r \in \mathcal{R}} x_{s,r} \leq Q_s \cdot T \qquad \forall s \in \mathcal{S}$$

Decision variables constraints

Decision variables constraints

$$y(s, r) \geq y(s, r+1) \qquad \forall s \in \mathcal{S}, r : r, r+1 \in \mathcal{R}$$

Decision variables constraints

$$y(s,r) \geq y(s,r+1) \qquad \forall s \in \mathcal{S}, r : r, r+1 \in \mathcal{R}$$

$$x_{s,r} \leq (B_{s,r} - B_{s,r-1}) \cdot y(s,r) \qquad \forall s \in \mathcal{S}, r \in \mathcal{R}$$

Decision variables constraints

$$y(s,r) \geq y(s,r+1) \qquad \forall s \in \mathcal{S}, r : r, r+1 \in \mathcal{R}$$

$$x_{s,r} \leq (B_{s,r} - B_{s,r-1}) \cdot y(s,r) \qquad \forall s \in \mathcal{S}, r \in \mathcal{R}$$

$$(B_{s,r} - B_{s,r-1}) \cdot y(s,r+1) \leq x_{s,r} \qquad \forall s \in \mathcal{S}, r : r, r+1 \in \mathcal{R}$$

Decision variables constraints

$$y(s,r) \geq y(s,r+1) \qquad \forall s \in \mathcal{S}, r : r, r+1 \in \mathcal{R}$$

$$x_{s,r} \leq (B_{s,r} - B_{s,r-1}) \cdot y(s,r) \qquad \forall s \in \mathcal{S}, r \in \mathcal{R}$$

$$(B_{s,r} - B_{s,r-1}) \cdot y(s,r+1) \leq x_{s,r} \qquad \forall s \in \mathcal{S}, r : r, r+1 \in \mathcal{R}$$

$$(B_{s,0} := 0 \quad \forall s \in \mathcal{S})$$

SOS2

Special Ordered Sets of type 2: an ordered set of non-negative variables, of which at most two can be non-zero, and if two are non-zero these must be consecutive in their ordering

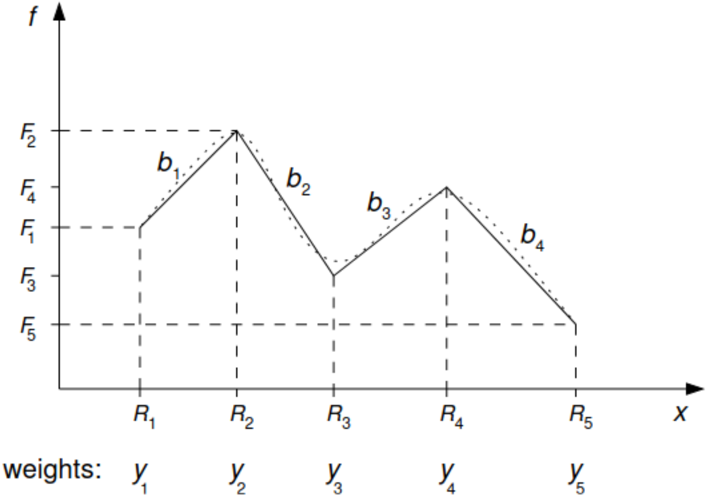
- the **order** is specified by assigning weights (*reference row values*) to each variable
- used for branching on **sets** of variables, rather than individual variables
- the search procedure will generally be noticeably faster

SOS2

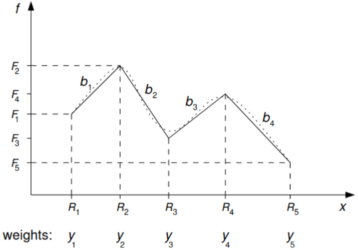
Special Ordered Sets of type 2: an ordered set of non-negative variables, of which at most two can be non-zero, and if two are non-zero these must be consecutive in their ordering

- the **order** is specified by assigning weights (*reference row values*) to each variable
- used for branching on **sets** of variables, rather than individual variables
- the search procedure will generally be noticeably faster
- generally used for modeling **piecewise approximations of functions** of a single variable

SOS2: piecewise functions



SOS2: piecewise functions

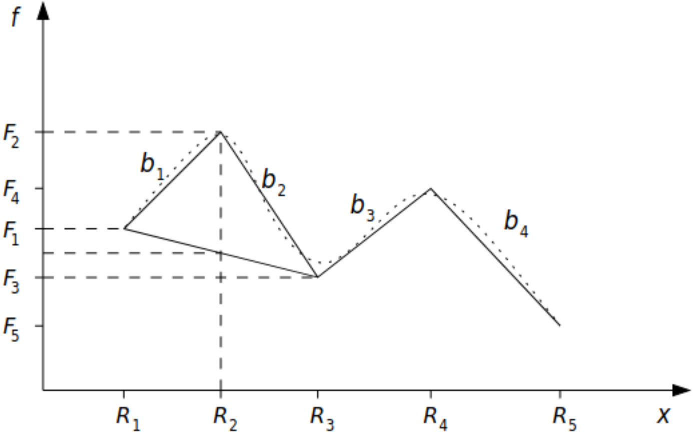


$$x = \sum_{i=1}^{5} R_i \cdot y_i \qquad \textbf{(reference row)}$$

$$f = \sum_{i=1}^{5} F_i \cdot y_i$$

$$\sum_{i=1}^{5} y_i = 1 \qquad \textbf{(convexity row)}$$

SOS2: adjacency condition



$y_1 = 0.5$

$y_2 = 0$

$y_3 = 0.5$

$y_4 = 0$

$y_5 = 0$

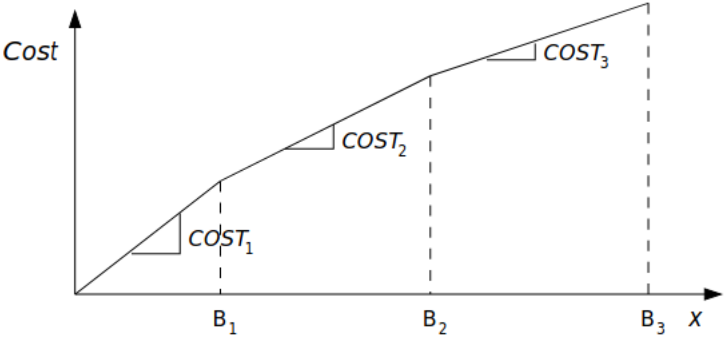
SOS2: alternative formulation

$$b_i = \begin{cases} 1 & \text{if the value of } x \\ & \quad \text{lies between } R_i \text{ and } R_{i+1} \\ 0 & \text{otherwise} \end{cases} \qquad \forall i \in 1, \dots, 4$$

$$\begin{aligned} y_1 &\leq b_1 \\ y_2 &\leq b_1 + b_2 \\ y_3 &\leq b_2 + b_3 \\ y_4 &\leq b_3 + b_4 \\ y_5 &\leq b_4 \end{aligned}$$

$$b_1 + b_2 + b_3 + b_4 = 1$$

SOS2: Purchasing with price breaks



Notation

$\mathcal{S}$: set of suppliers

$\mathcal{R}$: set of price ranges

$COST_{s,r}$: unit cost of price range r for supplier s

$B_{s,r}$: breakpoint where the unit cost change from $COST_{s,r}$ to $COST_{s,r+1}$ for supplier s

T : total amount we wish to buy

Q_s : maximum percentage that may be bought from supplier s

Notation

$\mathcal{S}$: set of suppliers

$\mathcal{R}$: set of price ranges

$COST_{s,r}$: unit cost of price range r for supplier s

$B_{s,r}$: breakpoint where the unit cost change from $COST_{s,r}$ to $COST_{s,r+1}$ for supplier s

T : total amount we wish to buy

Q_s : maximum percentage that may be bought from supplier s

$\mathcal{P}_0$: set of breakpoints including $0 \ = 0, 1, 2, 3$

$CBP_{s,p}$: total cost at breakpoint p for supplier s

Notation

$\mathcal{P}_0$: set of breakpoints including 0 $= 0, 1, 2, 3$

$CBP_{s,p}$: total cost at breakpoint p for supplier s

$$CBP_{s,0} := 0$$

$$CBP_{s,p} := CBP_{s,p-1} + COST_{s,p} \cdot (B_{s,p} - B_{s,p-1}) \quad \forall p \in \mathcal{P}_0, p > 0$$

$$(B_{s,0} := 0 \qquad \forall s \in \mathcal{S})$$

Decision variables

$x_{s,r}$: number of items bought from suppliers s at the price range r

$$y_{s,r} = \begin{cases} 1 & \text{if we have bought any items from} \\ & \quad \text{supplier } s \text{ at the price range } r \\ 0 & \text{otherwise} \end{cases} \qquad \forall s \in \mathcal{S}, r \in \mathcal{R}$$

Decision variables

$x_{s,r}$: number of items bought from suppliers s at the price range r

$$y_{s,r} = \begin{cases} 1 & \text{if we have bought any items from} \\ & \quad \text{supplier } s \text{ at the price range } r \\ 0 & \text{otherwise} \end{cases} \qquad \forall s \in \mathcal{S}, r \in \mathcal{R}$$

n_s : number of items bought from suppliers s

Decision variables

$x_{s,r}$: number of items bought from suppliers s at the price range r

$$y_{s,r} = \begin{cases} 1 & \text{if we have bought any items from} \\ & \quad \text{supplier } s \text{ at the price range } r \\ 0 & \text{otherwise} \end{cases} \qquad \forall s \in \mathcal{S}, r \in \mathcal{R}$$

n_s : number of items bought from suppliers s

$w_{s,p}$: weight variables associated with breakpoint p

Decision variables constraints

$$
\begin{aligned}
y(s,r) &\geq y(s,r+1) && \forall s \in \mathcal{S}, r : r, r+1 \in \mathcal{R} \\
x_{s,r} &\leq (B_{s,r} - B_{s,r-1}) \cdot y(s,r) && \forall s \in \mathcal{S}, r \in \mathcal{R} \\
(B_{s,r} - B_{s,r-1}) \cdot y(s,r+1) &\leq x_{s,r} && \forall s \in \mathcal{S}, r : r, r+1 \in \mathcal{R}
\end{aligned}
$$

Decision variables constraints

$$y(s,r) \geq y(s,r+1) \qquad \forall s \in \mathcal{S}, r : r, r+1 \in \mathcal{R}$$

$$x_{s,r} \leq (B_{s,r} - B_{s,r-1}) \cdot y(s,r) \qquad \forall s \in \mathcal{S}, r \in \mathcal{R}$$

$$(B_{s,r} - B_{s,r-1}) \cdot y(s,r+1) \leq x_{s,r} \qquad \forall s \in \mathcal{S}, r : r, r+1 \in \mathcal{R}$$

(**reference row**) $$n_s = \sum_{p \in \mathcal{P}_0} w_{s,p} \cdot B_{s,p} \quad \forall s \in \mathcal{S}$$

Decision variables constraints

$$y(s,r) \geq y(s,r+1) \qquad \forall s \in \mathcal{S}, r : r, r+1 \in \mathcal{R}$$

$$x_{s,r} \leq (B_{s,r} - B_{s,r-1}) \cdot y(s,r) \qquad \forall s \in \mathcal{S}, r \in \mathcal{R}$$

$$(B_{s,r} - B_{s,r-1}) \cdot y(s,r+1) \leq x_{s,r} \qquad \forall s \in \mathcal{S}, r : r, r+1 \in \mathcal{R}$$

(**reference row**) $$n_s = \sum_{p \in \mathcal{P}_0} w_{s,p} \cdot B_{s,p} \quad \forall s \in \mathcal{S}$$

(**convexity row**) $$\sum_{p \in \mathcal{P}_0} w_{s,p} = 1 \quad \forall s \in \mathcal{S}$$

Decision variables constraints

$$
\begin{aligned}
y(s,r) &\geq y(s,r+1) && \forall s \in \mathcal{S}, r : r, r+1 \in \mathcal{R} \\
x_{s,r} &\leq (B_{s,r} - B_{s,r-1}) \cdot y(s,r) && \forall s \in \mathcal{S}, r \in \mathcal{R} \\
(B_{s,r} - B_{s,r-1}) \cdot y(s,r+1) &\leq x_{s,r} && \forall s \in \mathcal{S}, r : r, r+1 \in \mathcal{R}
\end{aligned}
$$

$$
\textbf{(reference row)} \qquad n_s = \sum_{p \in \mathcal{P}_0} w_{s,p} \cdot B_{s,p} \quad \forall s \in \mathcal{S}
$$

$$
\textbf{(convexity row)} \qquad \sum_{p \in \mathcal{P}_0} w_{s,p} = 1 \quad \forall s \in \mathcal{S}
$$

$$
\forall s \in \mathcal{S} \quad \bigcup_{p \in \mathcal{P}_0} \{w_{s,p}\} \quad \textbf{SOS-2} \text{ with coeff. } B_{s,p}
$$

Decision variables constraints

```
! Define z and also order the weight variables by breakpoint
    quantities
 forall(s in S) sum(p in P0) B(s,p) * w(s,p) = z(s)

! The convexity row (w sum to 1)
 forall(s in S) sum(p in P0) w(s,p) = 1

! Define the w as SOS-2 as we can linearly interpolate
   between the breakpoints
forall(s in S)
makesos2(union(p in P0){w(s,p)}, sum(p in P0) B(s,b)*w(s,p))

! Alternative formulation:
!The weight coefficients BR are all augmented by EPS
!since Mosel does not accept 0-valued weights with ‘is_sos2’
forall(s in S)
  sum(p in P0) (B(s,p)+10E-20) * w(s,p) is_sos2
```

Constraints

$$\sum_{s \in \mathcal{S}, r \in \mathcal{R}} x_{s,r} \geq T$$

$$\sum_{r \in \mathcal{R}} x_{s,r} \leq Q_s \cdot T \qquad \forall s \in \mathcal{S}$$

Constraints

$$\sum_{s \in \mathcal{S}, r \in \mathcal{R}} x_{s,r} \geq T$$

$$\sum_{r \in \mathcal{R}} x_{s,r} \leq Q_s \cdot T \qquad \forall s \in \mathcal{S}$$

$$\sum_{s \in \mathcal{S}} n_s \geq T$$

Constraints

$$\sum_{s\in\mathcal{S},r\in\mathcal{R}} x_{s,r} \geq T$$

$$\sum_{r\in\mathcal{R}} x_{s,r} \leq Q_s \cdot T \qquad \forall s \in \mathcal{S}$$

$$\sum_{s\in\mathcal{S}} n_s \geq T$$

$$n_s \leq Q_s \cdot T \qquad \forall s \in \mathcal{S}$$

Constraints

$$\sum_{s\in\mathcal{S},r\in\mathcal{R}} x_{s,r} \geq T$$

$$\sum_{r\in\mathcal{R}} x_{s,r} \leq Q_s \cdot T \qquad \forall s \in \mathcal{S}$$

$$\sum_{s\in\mathcal{S}} n_s \geq T$$

$$n_s \leq Q_s \cdot T \qquad \forall s \in \mathcal{S}$$

```
! The minimum quantity that must be bought
 sum(s in S) z(s) >= T

! No more than the maximum percentage from each supplier
 forall(s in S) z(s) <= Q(s)*T
```

Objective function

$$min \sum_{s \in \mathcal{S}, r \in \mathcal{R}} x_{s,r} \cdot COST_{s,r}$$

Objective function

$$min \sum_{s \in \mathcal{S}, r \in \mathcal{R}} x_{s,r} \cdot COST_{s,r}$$

$$min \sum_{s \in \mathcal{S}, p \in \mathcal{P}_0} w_{s,p} \cdot CBR_{s,p}$$

Objective function

$$min \sum_{s \in \mathcal{S}, r \in \mathcal{R}} x_{s,r} \cdot COST_{s,r}$$

$$min \sum_{s \in \mathcal{S}, p \in \mathcal{P}_0} w_{s,p} \cdot CBR_{s,p}$$

```
! Objective: sum of costs*weights
 MinCost:= sum(s in S, p in P0) CB(s,p) * w(s,p)

 minimize(MinCost)
```