The detail listing of each model discussed in Chapter 2 is included in this Appendix. The purpose is to present the GAMS programming environment to the reader who has not had experience with GAMS. Due to the length of the Chapter 3 and Chapter 4 models, the listing of these models will not be included.
Sets
  p         set of time points
             /p1*p18/
  i         set of mines
             /m1*m3/
  j         set of stockpiles
             /sp1*sp3/

Alias
  (i,ii)
  (j,jj):

Parameters
  Cap_s(j)         The capacity per stockpile (kt)
                  /sp1  50
                  sp2  40
                  sp3  45/
  rate_b(i)        The rate at which mine coal is supplied (kt per hr)
                  /m1  2
                  m2  1.8
                  m3  1.5/
  rate_r(j)        The rate at which stockpile coal can be supplied (kt per hr)
                  /sp1  2
                  sp2  2.2
                  sp3  2/
  cost(i)          The cost of transporting coal from a mine (R per kt)
                  /m1  2
                  m2  1.8
                  m3  1.5/;

Table
  ST0_s(i,j)       Starting levels of mine coal on each stockpile (kt)
                  sp1    sp2    sp3
                  m1  0      0      0
                  m2  0      0      0
                  m3  0      0      0;

 Scalars
  H         Time horizon (hr)
            /12/
  Demand    The demand at the end of the time horizon (kt)
            /20/
  Income    Payment for coal delivered to the factory (R per kt)
            /10/
Delta size of the time intervals;

\[ \text{Delta} = \frac{H}{\text{card}(p)}; \]

**Binary variables**

- \( w(i,j,p) \) Indicates coal transported from mine \( i \) to stockpile \( j \)
- \( x(j,p) \) Indicates coal supplied to the factory from stockpile \( j \)

**Positive variables**

- \( q_b(i,j,p) \) Quantity from mine \( i \) to stockpile \( j \) (kt)
- \( q_r(i,j,p) \) Quantity from mine \( i \) on stockpile \( j \) to the factory (kt)
- \( ST_s(i,j,p) \) Amount of coal from mine \( i \) stored in stockpile \( j \) (kt)

**Variables**

- \( z \) Objective function;

**Equations**

- **Objective** Objective function to maximise profit
- **Allocate_1** Only one mine \( i \) supplying a stockpile \( j \) at a time
- **Allocate_2** A mine \( i \) can supply to only one stockpile \( j \) at a time
- **Allocate_5** Stacking and reclaiming cannot happen simultaneously
- **Storage_1** Starting levels of coal on stockpiles
- **Storage_2** The stockpile material balance
- **Storage_3** The maximum capacity limit for the stockpile
- **Storage_4** Ensure an equal portion of each mine's coal is reclaimed
- **Storage_5** Set upper limit for coal supplied to factory
- **Demand_1** Ensure the factory's demand is met
- **Bunker_3** Calculate quantity conveyed from mine \( i \) to stockpile \( j \)
- **Stock_3** Calculate quantity conveyed from stockpile \( j \) to factory

**Objective..**

\[ z = (\sum(i,j,p,q_r(i,j,p))*\text{Income}) - \sum(i,j,p,q_b(i,j,p)*\text{cost}(i)); \]

**Allocate_1(j,p)..

\[ \sum(i,w(i,j,p)) \leq 1; \]

**Allocate_2(i,p)..

\[ \sum(j,w(i,j,p)) \leq 1; \]

**Allocate_5(j,p)..

\[ \sum(i,w(i,j,p)) + x(j,p) \leq 1; \]

**Storage_1(i,j,p)($(ord(p)=1)$).

\[ ST_s(i,j,p) = ST0_s(i,j); \]

**Storage_2(i,j,p)($(ord(p)>1)$).

\[ ST_s(i,j,p) = ST_s(i,j,p-1) + q_b(i,j,p-1) - q_r(i,j,p-1); \]
\textbf{Model} eventbased /all/;

eventbased.optfile =1;

Option limrow = 0;
Option limcol = 0;
Option iterlim = 10000000;
Option reslim = 300000;

\textbf{Solve} eventbased using minlp maximizing z;
Sets
  \( p \) set of slots
  \( /p1*p6/ \)
  \( i \) set of mines
  \( /m1*m3/ \)
  \( j \) set of stockpiles
  \( /sp1*sp3/ \)

Alias
  \((i,ii)\)
  \((j,jj)\):

Parameters
  \( \text{Cap}_s(j) \) The capacity per stockpile (kt)
  \( /sp1 50\)
  \( sp2 40\)
  \( sp3 45/ \)
  \( \text{rate}_b(i) \) The rate at which mine coal is supplied (kt per hr)
  \( /m1 2\)
  \( m2 1.8\)
  \( m3 1.5/ \)
  \( \text{rate}_r(j) \) The rate at which stockpile coal can be supplied (kt per hr)
  \( /sp1 2\)
  \( sp2 2.2\)
  \( sp3 2/ \)
  \( \text{cost}(i) \) The cost of transporting coal from a mine (R per kt)
  \( /m1 2\)
  \( m2 1.8\)
  \( m3 1.5/; \)

Table
  \( \text{ST0}_s(i,j) \) Starting levels of mine coal on each stockpile (kt)
  \( sp1 sp2 sp3 \)
  \( m1 0 0 0\)
  \( m2 0 0 0\)
  \( m3 0 0 0; \)

Scalars
  \( H \) Time horizon (hr)
  \( /12/ \)
  \( \text{Demand} \) The demand at the end of the time horizon (kt)
  \( /20/ \)
  \( \text{Income} \) Payment for coal delivered to the factory (R per kt)
  \( /10/; \)
Binary variables

\[ w(i,j,p) \quad \text{Indicates coal transported from mine i to stockpile j} \]
\[ x(j,p) \quad \text{Indicates coal supplied to the factory from stockpile j} \]

Positive variables

\[ q_{\_b}(i,j,p) \quad \text{Quantity from mine i to stockpile j (kt)} \]
\[ q_{\_r}(i,j,p) \quad \text{Quantity from mine i on stockpile j to the factory (kt)} \]
\[ ST_s(i,j,p) \quad \text{Amount of coal from mine i stored in stockpile j (kt)} \]
\[ Ts_b(i,j,p) \quad \text{Starting time for transporting coal from mine i to stockpile j (hr)} \]
\[ Tf_b(i,j,p) \quad \text{Finish time for transporting coal from mine i to stockpile j (hr)} \]
\[ Dur_b(i,j,p) \quad \text{Duration of transporting coal from mine i to stockpile j (hr)} \]
\[ Ts_r(j,p) \quad \text{Starting time for supplying coal from stockpile j to the factory (hr)} \]
\[ Tf_r(j,p) \quad \text{Finish time for supplying coal from stockpile j to the factory (hr)} \]
\[ Dur_r(j,p) \quad \text{Duration of transporting coal from stockpile j to the factory (hr)} \]

Variables

\[ z \quad \text{Objective function;} \]

Equations

Objective

\[ \text{Objective function to maximise profit} \]

Allocate_1

\[ \text{Only one mine i supplying a stockpile j at a time} \]

Allocate_2

\[ \text{A mine i can supply to only one stockpile j at a time} \]

Allocate_5

\[ \text{Stacking and reclaiming cannot happen simultaneously} \]

Storage_1

\[ \text{Starting levels of coal on stockpiles} \]

Storage_2

\[ \text{The stockpile material balance} \]

Storage_3

\[ \text{The maximum capacity limit for the stockpile} \]

Storage_4

\[ \text{Ensure an equal portion of each mine's coal is reclaimed} \]

Storage_5

\[ \text{Set upper limit for coal supplied to factory} \]

Demand_1

\[ \text{Ensure the factory's demand is met} \]

Bunker_1

\[ \text{Calculate finish time based on starting time and duration} \]

Bunker_2

\[ \text{Set upper limit for duration based on } w(ijp) \]

Bunker_3

\[ \text{Calculate quantity conveyed from mine i to stockpile j} \]

Bunker_4a

\[ \text{Ensure time sequence of events at a mine} \]

Bunker_4b

\[ \text{Ensure time sequence of events at a stockpile} \]

Bunker_5

\[ \text{Set upper limit for finish time} \]

Bunker_6

\[ \text{Sequencing starting times} \]

Bunker_7

\[ \text{Sequencing finishing times} \]

Stock_1

\[ \text{Calculate finish time based on starting time and duration} \]

Stock_2

\[ \text{Set upper limit for duration based on } x(jp) \]

Stock_3

\[ \text{Calculate quantity conveyed from stockpile j to factory} \]

Stock_4

\[ \text{Ensure time sequence of events} \]

Stock_5

\[ \text{Set upper limit for finish time} \]

Stock_6

\[ \text{Sequencing starting times} \]

Stock_7

\[ \text{Sequencing finishing times} \]

Sequence_1

\[ \text{Ensure sequence between stacking and reclaiming} \]
Objective..  
\[ z = (\text{sum}((i,j,p), q_r(i,j,p)) \times \text{Income}) - \text{sum}((i,j,p), q_b(i,j,p) \times \text{cost}(i)); \]

Allocate_1(j,p).  
\[ \text{sum}(i, w(i,j,p)) = 1; \]

Allocate_2(i,p).  
\[ \text{sum}(j, w(i,j,p)) = 1; \]

Allocate_5(j,p).  
\[ \text{sum}(i, w(i,j,p)) + x(j,p) = 1; \]

Storage_1(i,j,p)$(ord(p)=1).  
\[ \text{ST}_s(i,j,p) = \text{ST}_0_s(i,j); \]

Storage_2(i,j,p)$(ord(p)>1).  
\[ \text{ST}_s(i,j,p) = \text{ST}_s(i,j,p-1) + q_b(i,j,p-1) - q_r(i,j,p-1); \]

Storage_3(j,p).  
\[ \text{sum}(i, \text{ST}_s(i,j,p)) = \text{Cap}_s(j); \]

Storage_4(i,ii,j,p)$(ord(i)<=ord(ii)).  
\[ q_r(i,j,p) \times \text{ST}_s(ii,j,p) = q_r(ii,j,p) \times \text{ST}_s(i,j,p); \]

Storage_5(i,j,p).  
\[ q_r(i,j,p) \leq \text{ST}_s(i,j,p); \]

Demand_1..  
\[ \text{sum}((i,j,p), q_r(i,j,p)) \geq \text{Demand}; \]

Bunker_1(i,j,p).  
\[ \text{Tf}_b(i,j,p) = \text{Ts}_b(i,j,p) + \text{Dur}_b(i,j,p); \]

Bunker_2(i,j,p).  
\[ \text{Dur}_b(i,j,p) \leq H \times w(i,j,p); \]

Bunker_3(i,j,p).  
\[ q_b(i,j,p) = \text{rate}_b(i) \times \text{Dur}_b(i,j,p); \]

Bunker_4a(i,j,ii,jj,p)$(ord(p)>1).  
\[ \text{Ts}_b(i,j,p) = \text{Tf}_b(i,jj,p-1) - H \times (1 - w(i,j,p)); \]

Bunker_4b(i,ii,j,jj,p)$(ord(p)>1).  
\[ \text{Ts}_b(ii,j,p) = \text{Tf}_b(ii,j,p-1) - H \times (1 - w(i,j,p)); \]

Bunker_5(i,j,p).  
\[ \text{Tf}_b(i,j,p) = H; \]

Bunker_6(i,j,p)$(ord(p)>1).  
\[ \text{Ts}_b(i,j,p) = \text{Ts}_b(i,j,p-1); \]

Bunker_7(i,j,p)$(ord(p)>1).  
\[ \text{Tf}_b(i,j,p) = \text{Tf}_b(i,j,p-1); \]

Stock_1(j,p).  
\[ \text{Tf}_r(j,p) = \text{Ts}_r(j,p) + \text{Dur}_r(j,p); \]
Stock_2(j,p)..
   Dur_r(j,p) =l= H*x(j,p);

Stock_3(j,p)..
   sum(i,q_r(i,j,p)) =e= rate_r(j)*Dur_r(j,p);

Stock_4(j,p)$\text{(ord}(p)\text{)}$>1)..
   Ts_r(j,p) =g= Tf_r(j,p-1) - H*(1 - x(j,p));

Stock_5(j,p)..
   Tf_r(j,p) =l= H;

Stock_6(j,p)$\text{(ord}(p)\text{)}$>1)..
   Ts_r(j,p) =g= Ts_r(j,p-1);

Stock_7(j,p)$\text{(ord}(p)\text{)}$>1)..
   Tf_r(j,p) =g= Tf_r(j,p-1);

Sequence_1(i,j,p)$\text{(ord}(p)\text{)}$>1)..
   Ts_r(j,p) =g= Tf_b(i,j,p-1) - H*(1 - x(j,p));

**Model** eventbased /all/;

eventbased.optfile =1;

Option limrow = 0;
Option limcol = 0;
Option iterlim = 100000000;
Option reslim = 3000000;

**Solve** eventbased using minlp maximizing z;
Sets

- p set of time points
  /p1*p6/
- i set of mines
  /m1*m3/
- j set of stockpiles
  /sp1*sp3/

Alias

- (i,ii)
- (j,jj):

Parameters

- Cap_s(j): The capacity per stockpile (kt)
  /sp1 50
  sp2 40
  sp3 45/
- rate_b(i): The rate at which mine coal is supplied (kt per hr)
  /m1 2
  m2 1.8
  m3 1.5/
- rate_r(j): The rate at which stockpile coal can be supplied (kt per hr)
  /sp1 2
  sp2 2.2
  sp3 2/
- cost(i): The cost of transporting coal from a mine (R per kt)
  /m1 2
  m2 1.8
  m3 1.5/;

Table

- ST0_s(i,j): Starting levels of mine coal on each stockpile (kt)
  sp1 sp2 sp3
  m1 0 0 0
  m2 0 0 0
  m3 0 0 0;

Scalars

- H Time horizon (hr)
  /12/
- Demand The demand at the end of the time horizon (kt)
  /20/
- Income Payment for coal delivered to the factory (R per kt)
  /10/;
Binary variables

\( w(i,j,p) \) Indicates coal transported from mine \( i \) to stockpile \( j \)
\( x(j,p) \) Indicates coal supplied to the factory from stockpile \( j \)

Positive variables

\( q_{\text{b}}(i,j,p) \) Quantity from mine \( i \) to stockpile \( j \) (kt)
\( q_{\text{r}}(i,j,p) \) Quantity from mine \( i \) on stockpile \( j \) to the factory (kt)
\( ST_s(i,j,p) \) Amount of coal from mine \( i \) stored in stockpile \( j \) (kt)
\( Du_{\text{r}}(i,j,p) \) Duration of transporting coal from mine \( i \) to stockpile \( j \) (hr)
\( Du_{\text{r}}(j,p) \) Duration of transporting coal from stockpile \( j \) to the factory (hr)
\( T(p) \) Indicate the time of an event point (hr)

Variables

\( z \) Objective function;

Equations

Objective function to maximise profit

- **Allocate_1** Only one mine \( i \) supplying a stockpile \( j \) at a time
- **Allocate_2** A mine \( i \) can supply to only one stockpile \( j \) at a time
- **Allocate_3** No conveying on the last time point
- **Allocate_4** No supplying on the last time point
- **Allocate_5** Stacking and reclaiming cannot happen simultaneously

- **Storage_1** Starting levels of coal on stockpiles
- **Storage_2** The stockpile material balance
- **Storage_3** The maximum capacity limit for the stockpile
- **Storage_4** Ensure an equal portion of each mine's coal is reclaimed
- **Storage_5** Set upper limit for coal supplied to factory

- **Demand_1** Ensure the factory's demand is met

- **Bunker_2** Set upper limit for duration based on \( w(i,j,p) \)
- **Bunker_3** Calculate quantity conveyed from mine \( i \) to stockpile \( j \)
- **Stock_2** Set upper limit for duration based on \( x(j,p) \)
- **Stock_3** Calculate quantity conveyed from stockpile \( j \) to factory

- **Time_1a** Calculate duration of coal conveyed from mine \( i \) to stockpile \( j \)
- **Time_1b**
- **Time_2a** Calculate duration of coal conveyed from stockpile \( j \) to factory
- **Time_2b**
- **Time_3** Set upper limit on \( T(p) \)
- **Time_4** Sequencing

\[
\text{Objective..} \quad z = e = (\text{sum}(i,j,p), q_{\text{r}}(i,j,p) ) e^{\text{Income}} - \text{sum}(i,j,p), q_{\text{b}}(i,j,p) e^{\text{cost}(i)};
\]

\[
\text{Allocate_1}(j,p) .. \quad \text{sum}(i,w(i,j,p)) = l = 1;
\]
Allocate_2(i,p)..
\[ \sum_{j} w(i,j,p) \leq 1; \]

Allocate_3(i,j,p)$(ord(p)=\text{card}(p))$..
\[ w(i,j,p) = 0; \]

Allocate_4(j,p)$(ord(p)=\text{card}(p))$..
\[ x(j,p) = 0; \]

Allocate_5(j,p)..
\[ \sum_{i} w(i,j,p) + x(j,p) \leq 1; \]

Storage_1(i,j,p)$(ord(p)=1)$..
\[ \text{ST}_s(i,j,p) = \text{ST0}_s(i,j); \]

Storage_2(i,j,p)$(ord(p)>1)$..
\[ \text{ST}_s(i,j,p) = \text{ST}_s(i,j,p-1) + q_b(i,j,p-1) - q_r(i,j,p-1); \]

Storage_3(j,p)..
\[ \sum_{i} \text{ST}_s(i,j,p) \leq \text{Cap}_s(j); \]

Storage_4(i,ii,j,p)$(ord(i)<=ord(ii))$..
\[ q_r(i,j,p) \cdot \text{ST}_s(ii,j,p) = q_r(ii,j,p) \cdot \text{ST}_s(i,j,p); \]

Storage_5(i,j,p)..
\[ q_r(i,j,p) \leq \text{ST}_s(i,j,p); \]

Demand_1..
\[ \sum_{(i,j,p)} q_r(i,j,p) \geq \text{Demand}; \]

Bunker_2(i,j,p)..
\[ \text{Dur}_b(i,j,p) \leq H \cdot w(i,j,p); \]

Bunker_3(i,j,p)..
\[ q_b(i,j,p) = \text{rate}_b(i) \cdot \text{Dur}_b(i,j,p); \]

Stock_2(j,p)..
\[ \text{Dur}_r(j,p) \leq H \cdot x(j,p); \]

Stock_3(j,p)..
\[ \sum_{i} q_r(i,j,p) = \text{rate}_r(j) \cdot \text{Dur}_r(j,p); \]

Time_1a(i,j,p)$(ord(p)>1)$..
\[ T(p) = T(p-1) + \text{Dur}_b(i,j,p-1) - H \cdot (1 - w(i,j,p-1)); \]

Time_1b(i,j,p)$(ord(p)>1)$..
\[ T(p) = T(p-1) + \text{Dur}_b(i,j,p-1) + H \cdot (1 - w(i,j,p-1)); \]

Time_2a(j,p)$(ord(p)>1)$..
\[ T(p) = T(p-1) + \text{Dur}_r(j,p-1) - H \cdot (1 - x(j,p-1)); \]

Time_2b(j,p)$(ord(p)>1)$..
\[ T(p) = T(p-1) + \text{Dur}_r(j,p-1) + H \cdot (1 - x(j,p-1)); \]

Time_3(p)..
\[ T(p) = H; \]
\[
T_4(p) \text{if } \text{ord}(p) > 1.
\]

\[
T(p) \geq T(p-1);
\]

**Model** eventbased /all/;

eventbased.optfile =1;

Option limrow = 0;
Option limcol = 0;
Option iterlim = 10000000;
Option reslim = 300000;

**Solve** eventbased using minlp maximizing;
Sets
p set of time points
   /p1*p6/
i set of mines
   /m1*m3/
j set of stockpiles
   /sp1*sp3/

Alias
(i,ii)
   (j,jj):

Parameters
Cap_s(j) The capacity per stockpile (kt)
   /sp1  50
   sp2  40
   sp3  45/
rate_b(i) The rate at which mine coal is supplied (kt per hr)
   /m1  2
   m2  1.8
   m3  1.5/
rate_r(j) The rate at which stockpile coal can be supplied (kt per hr)
   /sp1  2
   sp2  2.2
   sp3  2/
cost(i) The cost of transporting coal from a mine (R per kt)
   /m1  2
   m2  1.8
   m3  1.5/;

Table
ST0_s(i,j) Starting levels of mine coal on each stockpile (kt)
   sp1 sp2 sp3
   m1  0  0  0
   m2  0  0  0
   m3  0  0  0;

Scalars
H Time horizon (hr)
   /12/
Demand The demand at the end of the time horizon (kt)
   /20/
Income Payment for coal delivered to the factory (R per kt)
   /10/;
Binary variables
- \( w(i,j,p) \) Indicates coal transported from mine \( i \) to stockpile \( j \)
- \( x(j,p) \) Indicates coal supplied to the factory from stockpile \( j \)

Positive variables
- \( q_{b}(i,j,p) \) Quantity from mine \( i \) to stockpile \( j \) (kt)
- \( q_{r}(i,j,p) \) Quantity from mine \( i \) on stockpile \( j \) to the factory (kt)
- \( ST_{s}(i,j,p) \) Amount of coal from mine \( i \) stored in stockpile \( j \) (kt)
- \( Ts_b(i,j,p) \) Starting time for transporting coal from mine \( i \) to stockpile \( j \) (hr)
- \( Tf_b(i,j,p) \) Finish time for transporting coal from mine \( i \) to stockpile \( j \) (hr)
- \( Dur_b(i,j,p) \) Duration of transporting coal from mine \( i \) to stockpile \( j \) (hr)
- \( Ts_r(j,p) \) Starting time for supplying coal from stockpile \( j \) to the factory (hr)
- \( Tf_r(j,p) \) Finish time for transporting coal from stockpile \( j \) to the factory (hr)
- \( Dur_r(j,p) \) Duration of transporting coal from stockpile \( j \) to the factory (hr)

Variables
- \( z \) Objective function;

Equations
- **Objective**
  - Objective function to maximise profit
- **Allocate_1**
  - Only one mine \( i \) supplying a stockpile \( j \) at a time
- **Allocate_2**
  - A mine \( i \) can supply to only one stockpile \( j \) at a time
- **Allocate_3**
  - No conveying on the last time point
- **Allocate_4**
  - No supplying on the last time point
- **Allocate_5**
  - Stacking and reclaiming cannot happen simultaneously
- **Storage_1**
  - Starting levels of coal on stockpiles
- **Storage_2**
  - The stockpile material balance
- **Storage_3**
  - The maximum capacity limit for the stockpile
- **Storage_4**
  - Ensure an equal portion of each mine's coal is reclaimed
- **Storage_5**
  - Set upper limit for coal supplied to factory
- **Demand_1**
  - Ensure the factory's demand is met
- **Bunker_1**
  - Calculate finish time based on starting time and duration
- **Bunker_2**
  - Set upper limit for duration based on \( w(ijp) \)
- **Bunker_3**
  - Calculate quantity conveyed from mine \( i \) to stockpile \( j \)
- **Bunker_4a**
  - Ensure time sequence of events at a mine
- **Bunker_4b**
  - Ensure time sequence of events at a stockpile
- **Bunker_5**
  - Set upper limit for finish time
- **Bunker_6**
  - Sequencing starting times
- **Bunker_7**
  - Sequencing finishing times
- **Stock_1**
  - Calculate finish time based on starting time and duration
- **Stock_2**
  - Set upper limit for duration based on \( x(jp) \)
- **Stock_3**
  - Calculate quantity conveyed from stockpile \( j \) to factory
- **Stock_4**
  - Ensure time sequence of events
- **Stock_5**
  - Set upper limit for finish time
- **Stock_6**
  - Sequencing starting times
- **Stock_7**
  - Sequencing finishing times
- **Sequence_1**
  - Ensure sequence between stacking and reclaiming;
Objective.. 
\[ z = e = \text{sum}((i,j,p), q_r(i,j,p) \times \text{Income}) - \text{sum}((i,j,p), q_b(i,j,p) \times \text{cost}(i)) \]

Allocate_1(j,p)..
\[ \text{sum}(i, w(i,j,p)) = l = 1 \]

Allocate_2(i,p)..
\[ \text{sum}(j, w(i,j,p)) = l = 1 \]

Allocate_3(i,j,p)$\text{ord}(p)=\text{card}(p)$..
\[ w(i,j,p) = e = 0 \]

Allocate_4(j,p)$\text{ord}(p)=\text{card}(p)$..
\[ x(j,p) = e = 0 \]

Allocate_5(j,p)..
\[ \text{sum}(i, w(i,j,p)) + x(j,p) = l = 1 \]

Storage_1(i,j,p)$\text{ord}(p)=1$..
\[ ST_s(i,j,p) = e = ST_0_s(i,j) \]

Storage_2(i,j,p)$\text{ord}(p)>1$..
\[ ST_s(i,j,p) = e = ST_s(i,j,p-1) + q_b(i,j,p-1) - q_r(i,j,p-1) \]

Storage_3(j,p)..
\[ \text{sum}(i, ST_s(i,j,p)) = l = \text{Cap}_s(j) \]

Storage_4(i,ii,j,p)$\text{ord}(i)<>\text{ord}(ii)$..
\[ q_r(i,j,p) \times ST_s(ii,j,p) = e = q_r(ii,j,p) \times ST_s(i,j,p) \]

Storage_5(i,j,p)..
\[ q_r(i,j,p) = l = ST_s(i,j,p) \]

Demand_1..
\[ \text{sum}((i,j,p), q_r(i,j,p)) = g = \text{Demand} \]

Bunker_1(i,j,p)$\text{ord}(p)>1$..
\[ Tf_b(i,j,p) = e = T_s_b(i,j,p-1) + Dur_b(i,j,p-1) \]

Bunker_2(i,j,p)..
\[ Dur_b(i,j,p) = l = H \times w(i,j,p) \]

Bunker_3(i,j,p)..
\[ q_b(i,j,p) = e = \text{rate}_b(i) \times Dur_b(i,j,p) \]

Bunker_4a(i,j,ii,jj,p)..
\[ T_s_b(i,j,p) = g = Tf_b(i,jj,p) - H \times (1 - w(i,j,p)) \]

Bunker_4b(i,ii,j,jj,p)..
\[ T_s_b(i,j,p) = g = Tf_b(ii,j,p) - H \times (1 - w(i,j,p)) \]

Bunker_5(i,j,p)..
\[ Tf_b(i,j,p) = l = H \]

Bunker_6(i,j,p)$\text{ord}(p)>1$..
\[ T_s_b(i,j,p) = g = T_s_b(i,j,p-1) \]
Bunker_7(i,j,p)$(ord(p)>1).
   Tf_b(i,j,p) =g= Tf_b(i,j,p-1);

Stock_1(j,p)$(ord(p)>1).
   Tf_r(j,p) =e= Ts_r(j,p-1) + Dur_r(j,p-1);

Stock_2(j,p).
   Dur_r(j,p) =l= H*x(j,p);

Stock_3(j,p).
   sum(i,q_r(i,j,p)) =e= rate_r(j)*Dur_r(j,p);

Stock_4(j,p).
   Ts_r(j,p) =g= Tf_r(j,p) - H*(1 - x(j,p));

Stock_5(j,p).
   Tf_r(j,p) =l= H;

Stock_6(j,p)$(ord(p)>1).
   Ts_r(j,p) =g= Ts_r(j,p-1);

Stock_7(j,p)$(ord(p)>1).
   Tf_r(j,p) =g= Tf_r(j,p-1);

Sequence_1(i,j,p)$(ord(p)>1).
   Ts_r(j,p) =g= Tf_b(i,j,p) - H*(1 - x(j,p));

Model eventbased /all/;

eventbased.optfile =1;

Option limrow = 0;
Option limcol = 0;
Option iterlim = 10000000;
Option reslim = 300000;

Solve eventbased using minlp maximizing z;
An example of the user input interface in MS Excel is presented.
SETS
p set of time points
    /p1*p18/

i set of mines
    /m1*m3/

j set of stockpiles
    /sp1*sp3/

Alias
   (i,ii)
   (j,jj):

Parameters
Cap_s(j) The capacity per stockpile (kt)
    /sp1 50
    sp2 40
    sp3 45/

rate_b(i) The rate at which mine coal is supplied (kt per hr)
    /m1 2
    m2 1.8
    m3 1.5/

rate_r(j) The rate at which stockpile coal can be supplied (kt per hr)
    /sp1 2
    sp2 2.2
    sp3 2/

cost(i) The cost of transporting coal from a mine (R per kt)
    /m1 2
    m2 1.8
    m3 1.5/

Table
ST0_s(i,j) Starting levels of mine coal on each stockpile (kt)
    sp1 sp2 sp3
    m1 0 0 0
    m2 0 0 0
    m3 0 0 0;

Scalars
H Time horizon (hr)
    /12/

Demand The demand at the end of the time horizon (kt)
    /20/

Income Payment for coal delivered to the factory (R per kt)
    /10/
Delta size of the time intervals;

\[ \Delta = \frac{H}{\text{card}(p)}; \]

**Binary variables**
- \( w(i,j,p) \) Indicates coal transported from mine \( i \) to stockpile \( j \)
- \( x(j,p) \) Indicates coal supplied to the factory from stockpile \( j \)

**Positive variables**
- \( q_b(i,j,p) \) Quantity from mine \( i \) to stockpile \( j \) (kt)
- \( q_r(i,j,p) \) Quantity from mine \( i \) on stockpile \( j \) to the factory (kt)
- \( ST_s(i,j,p) \) Amount of coal from mine \( i \) stored in stockpile \( j \) (kt)

**Variables**
- \( z \) Objective function;

**Equations**
- **Objective** Objective function to maximise profit
- **Allocate_1** Only one mine \( i \) supplying a stockpile \( j \) at a time
- **Allocate_2** A mine \( i \) can supply to only one stockpile \( j \) at a time
- **Allocate_5** Stacking and reclaiming cannot happen simultaneously
- **Storage_1** Starting levels of coal on stockpiles
- **Storage_2** The stockpile material balance
- **Storage_3** The maximum capacity limit for the stockpile
- **Storage_4** Ensure an equal portion of each mine's coal is reclaimed
- **Storage_5** Set upper limit for coal supplied to factory
- **Demand_1** Ensure the factory's demand is met
- **Bunker_3** Calculate quantity conveyed from mine \( i \) to stockpile \( j \)
- **Stock_3** Calculate quantity conveyed from stockpile \( j \) to factory

**Objective..**
\[ z = e^k (\sum((i,j,p), q_r(i,j,p)) \cdot \text{Income}) - \sum((i,j,p), q_b(i,j,p) \cdot \text{cost}(i)); \]

**Allocate_1(j,p)..**
\[ \sum(i,w(i,j,p)) = l= 1; \]

**Allocate_2(i,p)..**
\[ \sum(j,w(i,j,p)) = l= 1; \]

**Allocate_5(j,p)..**
\[ \sum(i,w(i,j,p)) + x(j,p) = l= 1; \]

**Storage_1(i,j,p)$(ord(p)=1)...**
\[ ST_s(i,j,p) = e^k ST0_s(i,j); \]

**Storage_2(i,j,p)$(ord(p)>1)...**
\[ ST_s(i,j,p) = e^k ST_s(i,j,p-1) + q_b(i,j,p-1) - q_r(i,j,p-1); \]
Storage_3(j,p)..
  sum(i, ST_s(i,j,p)) =i= Cap_s(j);

Storage_4(i,ii,j,p)$(ord(i)<>>ord(ii))..
  q_r(i,j,p)*ST_s(ii,j,p) =e= q_r(ii,j,p)*ST_s(i,j,p);

Storage_5(i,j,p)..
  q_r(i,j,p) =l= ST_s(i,j,p);

Demand_1..
  sum((i,j,p), q_r(i,j,p)) =g= Demand;

Bunker_3(i,j,p)..
  q_b(i,j,p) =e= rate_b(i)*Delta*w(i,j,p);

Stock_3(j,p)..
  sum(i, q_r(i,j,p)) =e= rate_r(j)*Delta*x(j,p);

**Model** eventbased /all/;

eventbased.optfile =1;

Option limrow = 0;
Option limcol = 0;
Option iterlim = 10000000;
Option reslim = 300000;

**Solve** eventbased **using minlp maximizing z**;
Sets
p set of slots
   /p1*p6/
i set of mines
   /m1*m3/
j set of stockpiles
   /sp1*sp3/

Alias
(i,ii)
(j,jj): 

Parameters
Cap_s(j) The capacity per stockpile (kt)
   /sp1  50
         sp2  40
         sp3  45/
rate_b(i) The rate at which mine coal is supplied (kt per hr)
   /m1   2
         m2   1.8
         m3   1.5/
rate_r(j) The rate at which stockpile coal can be supplied (kt per hr)
   /sp1  2
         sp2  2.2
         sp3  2/
cost(i) The cost of transporting coal from a mine (R per kt)
   /m1   2
         m2   1.8
         m3   1.5/

Table
ST0_s(i,j) Starting levels of mine coal on each stockpile (kt)
   sp1  sp2  sp3
   m1   0   0   0
   m2   0   0   0
   m3   0   0   0;

Scalars
H Time horizon (hr)
   /12/
Demand The demand at the end of the time horizon (kt)
   /20/
Income Payment for coal delivered to the factory (R per kt)
   /10/;
<table>
<thead>
<tr>
<th>Binary variables</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( w(i,j,p) )</td>
<td>Indicates coal transported from mine ( i ) to stockpile ( j )</td>
</tr>
<tr>
<td>( x(j,p) )</td>
<td>Indicates coal supplied to the factory from stockpile ( j )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Positive variables</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( q_{b}(i,j,p) )</td>
<td>Quantity from mine ( i ) to stockpile ( j ) (kt)</td>
</tr>
<tr>
<td>( q_{r}(i,j,p) )</td>
<td>Quantity from mine ( i ) on stockpile ( j ) to the factory (kt)</td>
</tr>
<tr>
<td>( ST_{s}(i,j,p) )</td>
<td>Amount of coal from mine ( i ) stored in stockpile ( j ) (kt)</td>
</tr>
<tr>
<td>( Ts_{b}(i,j,p) )</td>
<td>Starting time for transporting coal from mine ( i ) to stockpile ( j ) (hr)</td>
</tr>
<tr>
<td>( Tf_{b}(i,j,p) )</td>
<td>Finish time for transporting coal from mine ( i ) to stockpile ( j ) (hr)</td>
</tr>
<tr>
<td>( Dur_{b}(i,j,p) )</td>
<td>Duration of transporting coal from mine ( i ) to stockpile ( j ) (hr)</td>
</tr>
<tr>
<td>( Ts_{r}(j,p) )</td>
<td>Starting time for supplying coal from stockpile ( j ) to the factory (hr)</td>
</tr>
<tr>
<td>( Tf_{r}(j,p) )</td>
<td>Finish time for transporting coal from stockpile ( j ) to the factory (hr)</td>
</tr>
<tr>
<td>( Dur_{r}(j,p) )</td>
<td>Duration of transporting coal from stockpile ( j ) to the factory (hr)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Variables</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( z )</td>
<td>Objective function;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Equations</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objective</td>
<td>Objective function to maximise profit</td>
</tr>
<tr>
<td>( Allocate_1 )</td>
<td>Only one mine ( i ) supplying a stockpile ( j ) at a time</td>
</tr>
<tr>
<td>( Allocate_2 )</td>
<td>A mine ( i ) can supply to only one stockpile ( j ) at a time</td>
</tr>
<tr>
<td>( Allocate_5 )</td>
<td>Stacking and reclaiming cannot happen simultaneously</td>
</tr>
<tr>
<td>( Storage_1 )</td>
<td>Starting levels of coal on stockpiles</td>
</tr>
<tr>
<td>( Storage_2 )</td>
<td>The stockpile material balance</td>
</tr>
<tr>
<td>( Storage_3 )</td>
<td>The maximum capacity limit for the stockpile</td>
</tr>
<tr>
<td>( Storage_4 )</td>
<td>Ensure an equal portion of each mine's coal is reclaimed</td>
</tr>
<tr>
<td>( Storage_5 )</td>
<td>Set upper limit for coal supplied to factory</td>
</tr>
<tr>
<td>( Demand_1 )</td>
<td>Ensure the factory's demand is met</td>
</tr>
<tr>
<td>( Bunker_1 )</td>
<td>Calculate finish time based on starting time and duration</td>
</tr>
<tr>
<td>( Bunker_2 )</td>
<td>Set upper limit for duration based on ( w(ijp) )</td>
</tr>
<tr>
<td>( Bunker_3 )</td>
<td>Calculate quantity conveyed from mine ( i ) to stockpile ( j )</td>
</tr>
<tr>
<td>( Bunker_4a )</td>
<td>Ensure time sequence of events at a mine</td>
</tr>
<tr>
<td>( Bunker_4b )</td>
<td>Ensure time sequence of events at a stockpile</td>
</tr>
<tr>
<td>( Bunker_5 )</td>
<td>Set upper limit for finish time</td>
</tr>
<tr>
<td>( Bunker_6 )</td>
<td>Sequencing starting times</td>
</tr>
<tr>
<td>( Bunker_7 )</td>
<td>Sequencing finishing times</td>
</tr>
<tr>
<td>( Stock_1 )</td>
<td>Calculate finish time based on starting time and duration</td>
</tr>
<tr>
<td>( Stock_2 )</td>
<td>Set upper limit for duration based on ( x(jp) )</td>
</tr>
<tr>
<td>( Stock_3 )</td>
<td>Calculate quantity conveyed from stockpile ( j ) to factory</td>
</tr>
<tr>
<td>( Stock_4 )</td>
<td>Ensure time sequence of events</td>
</tr>
<tr>
<td>( Stock_5 )</td>
<td>Set upper limit for finish time</td>
</tr>
<tr>
<td>( Stock_6 )</td>
<td>Sequencing starting times</td>
</tr>
<tr>
<td>( Stock_7 )</td>
<td>Sequencing finishing times</td>
</tr>
<tr>
<td>( Sequence_1 )</td>
<td>Ensure sequence between stacking and reclaiming</td>
</tr>
</tbody>
</table>
Objective.. 
\[ z = e = (\text{sum}(i,j,p), q_r(i,j,p)) \text{Income} - \text{sum}(i,j,p), q_b(i,j,p) \text{cost}(i); \]

Allocate_1(j,p).. 
\[ \text{sum}(i, w(i,j,p)) \leq 1; \]

Allocate_2(i,p).. 
\[ \text{sum}(j, w(i,j,p)) \leq 1; \]

Allocate_5(j,p).. 
\[ \text{sum}(i, w(i,j,p)) + x(j,p) \leq 1; \]

Storage_1(i,j,p)$\text{(ord}(p)=1)$.. 
\[ \text{ST}_s(i,j,p) = e = \text{ST}0_s(i,j); \]

Storage_2(i,j,p)$\text{(ord}(p)>1)$.. 
\[ \text{ST}_s(i,j,p) = e = \text{ST}_s(i,j,p-1) + q_b(i,j,p-1) - q_r(i,j,p-1); \]

Storage_3(j,p).. 
\[ \text{sum}(i, \text{ST}_s(i,j,p)) \leq \text{Cap}_s(j); \]

Storage_4(i,ii,j,p)$\text{(ord}(i)<=\text{ord}(ii))$.. 
\[ q_r(i,j,p) \times \text{ST}_s(ii,j,p) = e = q_r(ii,j,p) \times \text{ST}_s(i,j,p); \]

Storage_5(i,j,p).. 
\[ q_r(i,j,p) \leq \text{ST}_s(i,j,p); \]

Demand_1.. 
\[ \text{sum}(i,j,p), q_r(i,j,p) \geq \text{Demand}; \]

Bunker_1(i,j,p).. 
\[ \text{Tf}_b(i,j,p) = e = \text{Ts}_b(i,j,p) + \text{Dur}_b(i,j,p); \]

Bunker_2(i,j,p).. 
\[ \text{Dur}_b(i,j,p) \leq H \times w(i,j,p); \]

Bunker_3(i,j,p).. 
\[ q_b(i,j,p) = e = \text{rate}_b(i) \times \text{Dur}_b(i,j,p); \]

Bunker_4a(i,ij,ii,j,p)$\text{(ord}(p)>1)$.. 
\[ \text{Tf}_b(i,j,p) = g = \text{Tf}_b(i,ij,p-1) - H \times (1 - w(i,j,p)); \]

Bunker_4b(i,ii,j,p)$\text{(ord}(p)>1)$.. 
\[ \text{Tf}_b(i,j,p) = g = \text{Tf}_b(ii,j,p-1) - H \times (1 - w(i,j,p)); \]

Bunker_5(i,j,p).. 
\[ \text{Tf}_b(i,j,p) \leq H; \]

Bunker_6(i,j,p)$\text{(ord}(p)>1)$.. 
\[ \text{Tf}_b(i,j,p) = g = \text{Tf}_b(i,j,p-1); \]

Bunker_7(i,j,p)$\text{(ord}(p)>1)$.. 
\[ \text{Tf}_b(i,j,p) = g = \text{Tf}_b(i,j,p-1); \]

Stock_1(j,p).. 
\[ \text{Tf}_r(j,p) = e = \text{Tf}_r(j,p) + \text{Dur}_r(j,p); \]
\[ \text{Stock}_2(j,p). \quad \text{Dur}_r(j,p) = l = H \times x(j,p); \]
\[ \text{Stock}_3(j,p). \quad \text{sum}(i, q_r(i,j,p)) = \text{rate}_r(j) \times \text{Dur}_r(j,p); \]
\[ \text{Stock}_4(j,p) \text{($)$(ord(p)>1)$.} \quad \text{T}_s_r(j,p) = \text{g} \geq \text{Tf}_r(j,p-1) - H \times (1 - x(j,p)); \]
\[ \text{Stock}_5(j,p). \quad \text{Tf}_r(j,p) = l \leq H; \]
\[ \text{Stock}_6(j,p) \text{($)$(ord(p)>1)$.} \quad \text{T}_s_r(j,p) = \text{g} \geq \text{T}_s_r(j,p-1); \]
\[ \text{Stock}_7(j,p) \text{($)$(ord(p)>1)$.} \quad \text{Tf}_r(j,p) = \text{g} \geq \text{Tf}_r(j,p-1); \]
\[ \text{Sequence}_1(i,j,p) \text{($)$(ord(p)>1)$.} \quad \text{T}_s_r(j,p) = \text{g} \geq \text{Tf}_b(i,j,p-1) - H \times (1 - x(j,p)); \]

**Model** eventbased /all/;

```
eventbased.optfile = 1;
Option limrow = 0;
Option limcol = 0;
Option iterlim = 100000000;
Option reslim = 3000000;
```

**Solve** eventbased using minlp maximizing z;
Sets
   p  set of time points
      /p1*p6/  
   i  set of mines
      /m1*m3/  
   j  set of stockpiles
      /sp1*sp3/  

Alias
   (i,ii)  
   (j,jj):  

Parameters
   Cap_s(j)  The capacity per stockpile (kt)
      /sp1  50  
            sp2  40  
            sp3  45/  

   rate_b(i)  The rate at which mine coal is supplied (kt per hr)
      /m1  2  
            m2  1.8  
            m3  1.5/  

   rate_r(j)  The rate at which stockpile coal can be supplied (kt per hr)
      /sp1  2  
            sp2  2.2  
            sp3  2/  

   cost(i)  The cost of transporting coal from a mine (R per kt)
      /m1  2  
            m2  1.8  
            m3  1.5/;  

Table
   ST0_s(i,j)  Starting levels of mine coal on each stockpile (kt)
      sp1  sp2  sp3  
      m1  0   0   0  
      m2  0   0   0  
      m3  0   0   0;  

Scalars
   H  Time horizon (hr)
      /12/  

   Demand  The demand at the end of the time horizon (kt)
      /20/  

   Income  Payment for coal delivered to the factory (R per kt)
      /10/;
Binary variables
\[ w(i,j,p) \] Indicates coal transported from mine i to stockpile j
\[ x(j,p) \] Indicates coal supplied to the factory from stockpile j

Positive variables
\[ q_{b}(i,j,p) \] Quantity from mine i to stockpile j (kt)
\[ q_{r}(i,j,p) \] Quantity from mine i on stockpile j to the factory (kt)
\[ ST_s(i,j,p) \] Amount of coal from mine i stored in stockpile j (kt)
\[ Dur_b(i,j,p) \] Duration of transporting coal from mine i to stockpile j (hr)
\[ Dur_r(j,p) \] Duration of transporting coal from stockpile j to the factory (hr)
\[ T(p) \] Indicate the time of an event point (hr)

Variables
\[ z \] Objective function;

Equations
Objective
Objective function to maximise profit
Allocate_1
Only one mine i supplying a stockpile j at a time
Allocate_2
A mine i can supply to only one stockpile j at a time
Allocate_3
No conveying on the last time point
Allocate_4
No supplying on the last time point
Allocate_5
Stacking and reclaiming cannot happen simultaneously
Storage_1
Starting levels of coal on stockpiles
Storage_2
The stockpile material balance
Storage_3
The maximum capacity limit for the stockpile
Storage_4
Ensure an equal portion of each mine's coal is reclaimed
Storage_5
Set upper limit for coal supplied to factory
Demand_1
Ensure the factory's demand is met
Bunker_2
Set upper limit for duration based on w(ijp)
Bunker_3
Calculate quantity conveyed from mine i to stockpile j
Stock_2
Set upper limit for duration based on x(jp)
Stock_3
Calculate quantity conveyed from stockpile j to factory
Time_1a
Calculate duration of coal conveyed from mine i to stockpile j
Time_1b
Calculate duration of coal conveyed from stockpile j to factory
Time_2a
Set upper limit on T(p)
Time_2b
Sequencing

Objective..
\[ z = \text{maximize} \sum (i,j,p, q_{r}(i,j,p)) \times \text{Income} - \sum (i,j,p, q_{b}(i,j,p)) \times \text{cost(i)}; \]

Allocate_1(j,p)..
\[ \sum(i,w(i,j,p)) \leq 1; \]
Allocate_2(i,p).
\[ \sum_{j} w(i,j,p) = 1; \]

Allocate_3(i,j,p)$(ord(p) = \text{card}(p))$.
\[ w(i,j,p) = 0; \]

Allocate_4(j,p)$(ord(p) = \text{card}(p))$.
\[ x(j,p) = 0; \]

Allocate_5(j,p).
\[ \sum_{i} w(i,j,p) + x(j,p) = 1; \]

Storage_1(i,j,p)$(ord(p) = 1)$.
\[ ST_s(i,j,p) = ST0_s(i,j); \]

Storage_2(i,j,p)$(ord(p) > 1)$.
\[ ST_s(i,j,p) = ST_s(i,j,p-1) + q_b(i,j,p-1) - q_r(i,j,p-1); \]

Storage_3(j,p).
\[ \sum_{i} ST_s(i,j,p) = \text{Cap}_s(j); \]

Storage_4(ii,i,j,p)$$(ord(i) < ord(ii))$.
\[ q_r(i,j,p) \times ST_s(ii,j,p) = q_r(ii,j,p) \times ST_s(i,j,p); \]

Storage_5(i,j,p).
\[ q_r(i,j,p) \leq ST_s(i,j,p); \]

Demand_1.
\[ \sum((i,j,p), q_r(i,j,p)) \geq \text{Demand}; \]

Bunker_2(i,j,p).
\[ \text{Dur}_b(i,j,p) \leq H \times w(i,j,p); \]

Bunker_3(i,j,p).
\[ q_b(i,j,p) = \text{rate}_b(i) \times \text{Dur}_b(i,j,p); \]

Stock_2(j,p).
\[ \text{Dur}_r(j,p) \leq H \times x(j,p); \]

Stock_3(j,p).
\[ \sum(i, q_r(i,j,p)) = \text{rate}_r(j) \times \text{Dur}_r(j,p); \]

Time_1a(i,j,p)$$(ord(p) > 1)$.
\[ T(p) = T(p-1) + \text{Dur}_b(i,j,p-1) - H \times (1 - w(i,j,p-1)); \]

Time_1b(i,j,p)$$(ord(p) > 1)$.
\[ T(p) = T(p-1) + \text{Dur}_b(i,j,p-1) + H \times (1 - w(i,j,p-1)); \]

Time_2a(j,p)$$(ord(p) > 1)$.
\[ T(p) = T(p-1) + \text{Dur}_r(j,p-1) - H \times (1 - x(j,p-1)); \]

Time_2b(j,p)$$(ord(p) > 1)$.
\[ T(p) = T(p-1) + \text{Dur}_r(j,p-1) + H \times (1 - x(j,p-1)); \]

Time_3(p).
\[ T(p) = \leq H; \]
Time_4(p)$\,(\text{ord}(p)>1)\,$.

T(p) \geq T(p-1);

\textbf{Model} eventbased /all/;

eventbased.optfile =1;

Option limrow = 0;
Option limcol = 0;
Option iterlim = 100000000;
Option reslim = 300000;

\textbf{Solve} eventbased using minlp maximizing;
Sets

p         set of time points
    /p1*p6/

i         set of mines
    /m1*m3/

j         set of stockpiles
    /sp1*sp3/

Alias

(i,ii)
(j,jj);

Parameters

Cap_s(j)  The capacity per stockpile (kt)
    /sp1    50
    sp2    40
    sp3    45/

rate_b(i)        The rate at which mine coal is supplied (kt per hr)
    /m1     2
    m2     1.8
    m3     1.5/

rate_r(j)        The rate at which stockpile coal can be supplied (kt per hr)
    /sp1    2
    sp2    2.2
    sp3    2/

cost(i)          The cost of transporting coal from a mine (R per kt)
    /m1     2
    m2     1.8
    m3     1.5/;

Table

ST0_s(i,j)       Starting levels of mine coal on each stockpile (kt)

sp1     sp2     sp3
m1      0       0       0
m2      0       0       0
m3      0       0       0;

Scalars

H         Time horizon (hr)
    /12/

Demand    The demand at the end of the time horizon (kt)
    /20/

Income    Payment for coal delivered to the factory (R per kt)
    /10/;
Binary variables
\[ w(i,j,p) \] Indicates coal transported from mine i to stockpile j
\[ x(j,p) \] Indicates coal supplied to the factory from stockpile j

Positive variables
\[ q_b(i,j,p) \] Quantity from mine i to stockpile j (kt)
\[ q_r(i,j,p) \] Quantity from mine i on stockpile j to the factory (kt)
\[ ST_s(i,j,p) \] Amount of coal from mine i stored in stockpile j (kt)
\[ Ts_b(i,j,p) \] Starting time for transporting coal from mine i to stockpile j (hr)
\[ Tf_b(i,j,p) \] Finish time for transporting coal from mine i to stockpile j (hr)
\[ Dur_b(i,j,p) \] Duration of transporting coal from mine i to stockpile j (hr)
\[ Ts_r(j,p) \] Starting time for supplying coal from stockpile j to the factory (hr)
\[ Tf_r(j,p) \] Finish time for transporting coal from stockpile j to the factory (hr)
\[ Dur_r(j,p) \] Duration of transporting coal from stockpile j to the factory (hr)

Variables
\[ z \] Objective function;

Equations
Objective function to maximise profit

Allocate_1 Only one mine i supplying a stockpile j at a time
Allocate_2 A mine i can supply to only one stockpile j at a time
Allocate_3 No conveying on the last time point
Allocate_4 No supplying on the last time point
Allocate_5 Stacking and reclaiming cannot happen simultaneously

Storage_1 Starting levels of coal on stockpiles
Storage_2 The stockpile material balance
Storage_3 The maximum capacity limit for the stockpile
Storage_4 Ensure an equal portion of each mine's coal is reclaimed
Storage_5 Set upper limit for coal supplied to factory

Demand_1 Ensure the factory's demand is met

Bunker_1 Calculate finish time based on starting time and duration
Bunker_2 Set upper limit for duration based on w(ijp)
Bunker_3 Calculate quantity conveyed from mine i to stockpile j
Bunker_4a Ensure time sequence of events at a mine
Bunker_4b Ensure time sequence of events at a stockpile
Bunker_5 Set upper limit for finish time
Bunker_6 Sequencing starting times
Bunker_7 Sequencing finishing times

Stock_1 Calculate finish time based on starting time and duration
Stock_2 Set upper limit for duration based on x(jp)
Stock_3 Calculate quantity conveyed from stockpile j to factory
Stock_4 Ensure time sequence of events
Stock_5 Set upper limit for finish time
Stock_6 Sequencing starting times
Stock_7 Sequencing finishing times

Sequence_1 Ensure sequence between stacking and reclaiming;
Objective.. 
\[ z = \sum_{(i,j,p)} (q_r(i,j,p) \cdot \text{Income}) - \sum_{(i,j,p)} (q_b(i,j,p) \cdot \text{cost}(i)); \]

Allocate_1(j,p)..
\[ \sum_{i} w(i,j,p) = 1; \]

Allocate_2(i,p)..
\[ \sum_{j} w(i,j,p) = 1; \]

Allocate_3(i,j,p)$\text{ord}(p)/authentication(card(p))$..
\[ w(i,j,p) = 0; \]

Allocate_4(j,p)$\text{ord}(p)/authentication(card(p))$..
\[ x(j,p) = 0; \]

Allocate_5(j,p)..
\[ \sum_{i} w(i,j,p) + x(j,p) = 1; \]

Storage_1(i,j,p)$\text{ord}(p)/authentication(1)$..
\[ ST_s(i,j,p) = ST0_s(i,j); \]

Storage_2(i,j,p)$\text{ord}(p)>1$..
\[ ST_s(i,j,p) = ST_s(i,j,p-1) + q_b(i,j,p-1) - q_r(i,j,p-1); \]

Storage_3(j,p)..
\[ \sum_{i} ST_s(i,j,p) = \text{Cap}_s(j); \]

Storage_4(i,ii,j,p)$\text{ord}(i)<>\text{ord}((ii))$..
\[ q_r(i,j,p) \cdot ST_s(ii,j,p) = q_r(ii,j,p) \cdot ST_s(i,j,p); \]

Storage_5(i,j,p)..
\[ q_r(i,j,p) = ST_s(i,j,p); \]

Demand_1..
\[ \sum_{(i,j,p)} (q_r(i,j,p)) \geq \text{Demand}; \]

Bunker_1(i,j,p)$\text{ord}(p)>1$..
\[ Tf_b(i,j,p) = Ts_b(i,j,p-1) + Dur_b(i,j,p-1); \]

Bunker_2(i,j,p)..
\[ Dur_b(i,j,p) = H \cdot w(i,j,p); \]

Bunker_3(i,j,p)..
\[ q_b(i,j,p) = \text{rate}_b(i) \cdot Dur_b(i,j,p); \]

Bunker_4a(i,ii,j,jj,p)..
\[ Ts_b(i,j,j) = g = Tf_b(i,ii,j,j) - H^*(1 - w(i,j,j)); \]

Bunker_4b(i,ii,j,jj,p)..
\[ Ts_b(i,j,j) = g = Tf_b(ii,j,j) - H^*(1 - w(i,j,j)); \]

Bunker_5(i,j,p)..
\[ Tf_b(i,j,p) = H; \]

Bunker_6(i,j,p)$\text{ord}(p)>1$..
\[ Ts_b(i,j,p) = g = Ts_b(i,j,p-1); \]
Bunker_7(i,j,p)$(ord(p)>1)..
    Tf_b(i,j,p) =g= Tf_b(i,j,p-1);

Stock_1(j,p)$(ord(p)>1)..
    Tf_r(j,p) =e= Ts_r(j,p-1) + Dur_r(j,p-1);

Stock_2(j,p)..
    Dur_r(j,p) =l= H*x(j,p);

Stock_3(j,p)..
    sum(i,q_r(i,j,p)) =e= rate_r(j)*Dur_r(j,p);

Stock_4(j,p)..
    Ts_r(j,p) =g= Tf_r(j,p) - H*(1 - x(j,p));

Stock_5(j,p)..
    Tf_r(j,p) =l= H;

Stock_6(j,p)$(ord(p)>1)..
    Ts_r(j,p) =g= Ts_r(j,p-1);

Stock_7(j,p)$(ord(p)>1)..
    Tf_r(j,p) =g= Tf_r(j,p-1);

Sequence_1(i,j,p)$(ord(p)>1)..
    Ts_r(j,p) =g= Tf_b(i,j,p) - H*(1 - x(j,p));

**Model** eventbased /all/;

eventbased.optfile =1;

Option limrow = 0;
Option limcol = 0;
Option iterlim = 10000000;
Option reslim = 300000;

**Solve** eventbased using minlp maximizing z;