APPENDIX A: DETAIL GAMS LISTING OF CHAPTER 2 EXAMPLES

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.

```
$Title  Chapter 2 example model: DISCRETE
$Offupper
$Offlisting


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 = H/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= (sum((i,j,p),q_r(i,j,p))*Income) - sum((i,j,p),q_b(i,j,p)*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= ST0_s(i,j);

Storage_2(i,j,p)$(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)..
                    sum(i,ST_s(i,j,p)) =l= 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;
```

```
$Title  Chapter 2 example model: TIME-SLOTS
$Offupper
$Offlisting


Sets
        p                   set of slots
            /p1*p6/

        i                   set of mines
            /m1*m3/

        j                   et 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_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= (sum((i,j,p),q_r(i,j,p))*Income) - sum((i,j,p),q_b(i,j,p)*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= ST0_s(i,j);

Storage_2(i,j,p)$(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)..
                    sum(i,ST_s(i,j,p)) =l= 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_1(i,j,p)..
                    Tf_b(i,j,p) =e= Ts_b(i,j,p) + Dur_b(i,j,p);

Bunker_2(i,j,p)..
                    Dur_b(i,j,p) =l= H*w(i,j,p);

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

Bunker_4a(i,j,jj,p)$(ord(p)>1)..
                    Ts_b(i,j,p) =g= Tf_b(i,jj,p-1) - H*(1 - w(i,j,p));

Bunker_4b(i,ii,j,p)$(ord(p)>1)..
                    Ts_b(i,j,p) =g= Tf_b(ii,j,p-1) - H*(1 - w(i,j,p));

Bunker_5(i,j,p)..
                    Tf_b(i,j,p) =l= H;

Bunker_6(i,j,p)$(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)..
                    Tf_r(j,p) =e= Ts_r(j,p) + 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)$(ord(p)>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)$(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-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;
```

```
$Title  Chapter 2 example model: GLOBAL EVENT BASED
$Offupper
$Offlisting


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	
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

;

```
Objective..
                    z =e= (sum((i,j,p),q_r(i,j,p))*Income) - sum((i,j,p),q_b(i,j,p)*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_3(i,j,p)$(ord(p)=card(p))..
                    w(i,j,p) =e= 0;

Allocate_4(j,p)$(ord(p)=card(p))..
                    x(j,p) =e= 0;

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= ST0_s(i,j);

Storage_2(i,j,p)$(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)..
                    sum(i,ST_s(i,j,p)) =l= 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_2(i,j,p)..
                    Dur_b(i,j,p) =l= H*w(i,j,p);

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

Time_1a(i,j,p)$(ord(p)>1)..
                    T(p) =g= T(p-1) + Dur_b(i,j,p-1) - H*(1 - w(i,j,p-1));

Time_1b(i,j,p)$(ord(p)>1)..
                    T(p) =l= T(p-1) + Dur_b(i,j,p-1) + H*(1 - w(i,j,p-1));

Time_2a(j,p)$(ord(p)>1)..
                    T(p) =g= T(p-1) + Dur_r(j,p-1) - H*(1 - x(j,p-1));

Time_2b(j,p)$(ord(p)>1)..
                    T(p) =l= T(p-1) + Dur_r(j,p-1) + H*(1 - x(j,p-1));

Time_3(p)..
                    T(p) =l= H;
```

```
Time_4(p)$(ord(p)>1)..
                         T(p) =g= T(p-1);

Model eventbased /all/;

eventbased.optfile =1;

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

Solve eventbased using minlp maximizing;
```

```
$Title  Chapter 2 example model: UNIT-SPECIFIC EVENT BASED
$Offupper
$Offlisting


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= (sum((i,j,p),q_r(i,j,p))*Income) - sum((i,j,p),q_b(i,j,p)*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_3(i,j,p)$(ord(p)=card(p))..
                    w(i,j,p) =e= 0;

Allocate_4(j,p)$(ord(p)=card(p))..
                    x(j,p) =e= 0;

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= ST0_s(i,j);

Storage_2(i,j,p)$(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)..
                    sum(i,ST_s(i,j,p)) =l= 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_1(i,j,p)$(ord(p)>1)..
                    Tf_b(i,j,p) =e= Ts_b(i,j,p-1) + Dur_b(i,j,p-1);

Bunker_2(i,j,p)..
                    Dur_b(i,j,p) =l= H*w(i,j,p);

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

Bunker_4a(i,j,jj,p)..
                    Ts_b(i,j,p) =g= Tf_b(i,jj,p) - H*(1 - w(i,j,p));

Bunker_4b(i,ii,j,p)..
                    Ts_b(i,j,p) =g= Tf_b(ii,j,p) - H*(1 - w(i,j,p));

Bunker_5(i,j,p)..
                    Tf_b(i,j,p) =l= H;

Bunker_6(i,j,p)$(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;
```

APPENDIX B: EXAMPLE OF THE USER INPUT INTERFACE

An example of the user input interface in MS Excel is presented.

```
$Title  Chapter 2 example model: DISCRETE
$Offupper
$Offlisting


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 = H/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= (sum((i,j,p),q_r(i,j,p))*Income) - sum((i,j,p),q_b(i,j,p)*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= ST0_s(i,j);

Storage_2(i,j,p)$(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)..
                    sum(i,ST_s(i,j,p)) =l= 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;
```

```
$Title  Chapter 2 example model: TIME-SLOTS
$Offupper
$Offlisting


Sets
        p                    set of slots
             /p1*p6/

        i                    set of mines
             /m1*m3/

        j                    et 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_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= (sum((i,j,p),q_r(i,j,p))*Income) - sum((i,j,p),q_b(i,j,p)*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= ST0_s(i,j);

Storage_2(i,j,p)$(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)..
                              sum(i,ST_s(i,j,p)) =l= 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_1(i,j,p)..
                              Tf_b(i,j,p) =e= Ts_b(i,j,p) + Dur_b(i,j,p);

Bunker_2(i,j,p)..
                              Dur_b(i,j,p) =l= H*w(i,j,p);

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

Bunker_4a(i,j,jj,p)$(ord(p)>1)..
                              Ts_b(i,j,p) =g= Tf_b(i,jj,p-1) - H*(1 - w(i,j,p));

Bunker_4b(i,ii,j,p)$(ord(p)>1)..
                              Ts_b(i,j,p) =g= Tf_b(ii,j,p-1) - H*(1 - w(i,j,p));

Bunker_5(i,j,p)..
                              Tf_b(i,j,p) =l= H;

Bunker_6(i,j,p)$(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)..
                              Tf_r(j,p) =e= Ts_r(j,p) + 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)$(ord(p)>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)$(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-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*;

```
$Title  Chapter 2 example model: GLOBAL EVENT BASED
$Offupper
$Offlisting


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	
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

;

```
Objective..
                    z =e= (sum((i,j,p),q_r(i,j,p))*Income) - sum((i,j,p),q_b(i,j,p)*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_3(i,j,p)$(ord(p)=card(p))..
                        w(i,j,p) =e= 0;

Allocate_4(j,p)$(ord(p)=card(p))..
                        x(j,p) =e= 0;

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= ST0_s(i,j);

Storage_2(i,j,p)$(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)..
                        sum(i,ST_s(i,j,p)) =l= 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_2(i,j,p)..
                        Dur_b(i,j,p) =l= H*w(i,j,p);

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

Time_1a(i,j,p)$(ord(p)>1)..
                        T(p) =g= T(p-1) + Dur_b(i,j,p-1) - H*(1 - w(i,j,p-1));

Time_1b(i,j,p)$(ord(p)>1)..
                        T(p) =l= T(p-1) + Dur_b(i,j,p-1) + H*(1 - w(i,j,p-1));

Time_2a(j,p)$(ord(p)>1)..
                        T(p) =g= T(p-1) + Dur_r(j,p-1) - H*(1 - x(j,p-1));

Time_2b(j,p)$(ord(p)>1)..
                        T(p) =l= T(p-1) + Dur_r(j,p-1) + H*(1 - x(j,p-1));

Time_3(p)..
                        T(p) =l= H;
```

```
Time_4(p)$(ord(p)>1)..
                          T(p) =g= T(p-1);
```

Model eventbased /all/;

eventbased.optfile =1;

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

Solve *eventbased using minlp maximizing;*

```
$Title  Chapter 2 example model: UNIT-SPECIFIC EVENT BASED
$Offupper
$Offlisting


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= (sum((i,j,p),q_r(i,j,p))*Income) - sum((i,j,p),q_b(i,j,p)*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_3(i,j,p)$(ord(p)=card(p))..
                    w(i,j,p) =e= 0;

Allocate_4(j,p)$(ord(p)=card(p))..
                    x(j,p) =e= 0;

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= ST0_s(i,j);

Storage_2(i,j,p)$(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)..
                    sum(i,ST_s(i,j,p)) =l= 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_1(i,j,p)$(ord(p)>1)..
                    Tf_b(i,j,p) =e= Ts_b(i,j,p-1) + Dur_b(i,j,p-1);

Bunker_2(i,j,p)..
                    Dur_b(i,j,p) =l= H*w(i,j,p);

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

Bunker_4a(i,j,jj,p)..
                    Ts_b(i,j,p) =g= Tf_b(i,jj,p) - H*(1 - w(i,j,p));

Bunker_4b(i,ii,j,p)..
                    Ts_b(i,j,p) =g= Tf_b(ii,j,p) - H*(1 - w(i,j,p));

Bunker_5(i,j,p)..
                    Tf_b(i,j,p) =l= H;

Bunker_6(i,j,p)$(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;
```