SOLUTION PROBLEM La

```
set    DIST         ordered                                          ; # Districts: North, South, East, West, Central
set    SCHO         ordered                                          ; # Schools: Addison, Beeks, Canfield, Dalcy

param  Ttm          {i in DIST, j in SCHO}                           ; # Travel Time from each District to each School
param  Stp          {i in DIST}                                      ; # Student Population per District
param  Cap                                                           ; # Maximum Capacity per School

var    QS           {i in DIST, j in SCHO} integer >= 0              ; # Quantity of Students per District per School
var    TT                                                            ; # Total Travel Time for all Students
var    QA                                                            ; # Average Travel Time per Student


minimize TotTtm     : TT                                             ; # Minimize Total Travel Time

s.t.   TtmCon       : TT = sum {i in DIST, j in SCHO} QS[i,j] * Ttm[i,j]; # Total Travel Time
s.t.   StpCon       {i in DIST} : sum {j in SCHO} QS[i,j]  = Stp[i]   ; # Students per District = Sum of Students per School
s.t.   CapCon       {j in SCHO} : sum {i in DIST} QS[i,j] <= Cap      ; # Quantity of Students <= Capacity per School
s.t.   AvgCon       : QA = TT / sum {i in DIST} Stp[i]                ; # Avg. Travel Time = Total Time / Student population


data                                                                                                    ;

set    DIST         :=          North       South       East        West        Central     ;
set    SCHO         :=          Addison     Beeks       Canfield    Dalcy                   ;

param  Ttm          :           Addison     Beeks       Canfield    Dalcy                   :=
                    North       12          23          35          17
                    South       26          15          21          27
                    East        18          20          22          31
                    West        29          24          35          10
                    Central     15          10          23          16                      ;

param  Stp          :=          North 250   South 340   East 310    West 210    Central 290 ;
param  Cap          :=          400                                                         ;


option solver gurobi                        ;
solve                                       ;
option display_width 200, display_1col 0    ;

display QS,QA                               ;
```

SOLUTION PROBLEM Lb

```
set    DIST         ordered                                                  ; # Districts: North, South, East, West, Central
set    SCHO         ordered                                                  ; # Schools: Addison, Beeks, Canfield, Dalcy

param  Ttm          {i in DIST, j in SCHO}                                   ; # Travel Time from each District to each School
param  Stp          {i in DIST}                                              ; # Student Population per District
param  Cap                                                                   ; # Maximum Capacity per School

var    QS           {i in DIST, j in SCHO} integer >= 0                      ; # Quantity of Students per District per School
var    TT                                                                    ; # Total Travel Time for all Students
var    QA                                                                    ; # Average Travel Time per Student


minimize TotTtm     : TT                                                     ; # Minimize Total Travel Time

s.t.   TtmCon       : TT = sum {i in DIST, j in SCHO} QS[i,j] * Ttm[i,j]; # Total Travel Time
s.t.   StpCon       {i in DIST} : sum {j in SCHO} QS[i,j] >= Stp[i]      ; # Students per District = Sum of Students per School
s.t.   CapCon       {j in SCHO} : sum {i in DIST} QS[i,j] <= Cap         ; # Quantity of Students <= Capacity per School
s.t.   AvgCon       : QA = TT / sum {i in DIST} Stp[i]                   ; # Avg. Travel Time = Total Time / Student population
s.t.   BalCon       {j in SCHO} : sum {i in DIST} QS[i,j] >= sum {i in DIST} Stp[i] / 4 ; # Even Quantity of Students among Schools


data                                                                                                    ;

set    DIST         :=          North       South       East        West        Central     ;
set    SCHO         :=          Addison     Beeks       Canfield    Dalcy                   ;

param  Ttm          :           Addison     Beeks       Canfield    Dalcy                   :=
                    North       12          23          35          17
                    South       26          15          21          27
                    East        18          20          22          31
                    West        29          24          35          10
                    Central     15          10          23          16                      ;

param  Stp          :=          North 250   South 340   East 310    West 210    Central 290 ;
param  Cap          :=          400                                                         ;


option solver gurobi                        ;
solve                                       ;
option display_width 200, display_1col 0    ;

display QS,QA                               ;
```

SOLUTION PROBLEM M

```
set    EUPT         ordered                                    ; # EU Ports: Hamburg, Marseilles, Liverpool
set    CITY         ordered                                    ; # US Cities: Norfolk, New_York, Savannah
set    DIST         ordered                                    ; # Distribution Centers: Dallas, St_Louis, Chicago

param  Cos          {i in EUPT, j in CITY}                     ; # Cost from each EU Port to each US City
param  Cod          {j in CITY, k in DIST}                     ; # Cost from each US City to each Distribution Center
param  Sup          {i in EUPT}                                ; # Maximum Capacity per EU Port
param  Dmd          {k in DIST}                                ; # Demand per Distribution Center

var    QC           {i in EUPT, j in CITY} integer >= 0        ; # Quantity Shipped from each Port to each City
var    QD           {j in CITY, k in DIST} integer >= 0        ; # Quantity Shipped from each City to each Distribution Center

minimize TotCst     : sum {i in EUPT, j in CITY} QC[i,j] * Cos[i,j]
                          + sum {j in CITY, k in DIST} QD[j,k] * Cod[j,k]          ; # Minimize Total Cost

s.t.   SupCon       {i in EUPT} : sum {j in CITY} QC[i,j] <= Sup[i]     ; # Quantity Shipped from each EU Port <= Supply Capacity
s.t.   DmdCon       {k in DIST} : sum {j in CITY} QD[j,k] >= Dmd[k]     ; # Quantity Shipped from each City >= Demand
s.t.   CitCon       {j in CITY} : sum {i in EUPT} QC[i,j] = sum {k in DIST} QD[j,k]   ; # QC = QD for each City

data                                                                                 ;

set    EUPT         :=          Hamburg      Marseilles         Liverpool           ;
set    CITY         :=          Norfolk      New_York           Savannah            ;
set    DIST         :=          Dallas       St_Louis           Chicago             ;

param  Cos          :           Norfolk      New_York           Savannah            :=
                    Hamburg     420          390                610
                    Marseilles  510          590                470
                    Liverpool   450          360                480                 ;

param  Cod          :           Dallas       St_Louis           Chicago             :=
                    Norfolk     75           63                 81
                    New_York    125          110                95
                    Savannah    68           82                 95                  ;

param  Sup          :=          Hamburg 55   Marseilles   78    Liverpool    37     ;
param  Dmd          :=          Dallas 60    St_Louis     45    Chicago      50     ;

option solver gurobi                         ;
solve                                        ;
option display_width 200, display_1col 0     ;
display QC                                   ;
display QD                                   ;
```

SOLUTION PROBLEM M 3dim

```
set    EUPT         ordered                                                  ; # EU Ports: Hamburg, Marseilles, Liverpool
set    CITY         ordered                                                  ; # US Cities: Norfolk, New_York, Savannah
set    DIST         ordered                                                  ; # Distribution Centers: Dallas, St_Louis, Chicago

param  Cos          {i in EUPT, j in CITY}                                   ; # Cost from each EU Port to each US City
param  Cod          {j in CITY, k in DIST}                                   ; # Cost from each US City to each Distribution Center
param  Sup          {i in EUPT}                                              ; # Maximum Capacity per EU Port
param  Dmd          {k in DIST}                                              ; # Demand per Distribution Center

var    QS           {i in EUPT, j in CITY, k in DIST} integer >= 0           ; # Q Shipped from each Port to each City and Dist. Center


minimize TotCst     : sum {i in EUPT, j in CITY, k in DIST} QS[i,j,k] * Cos[i,j]
                    + sum {i in EUPT, j in CITY, k in DIST} QS[i,j,k] * Cod[j,k]     ; # Minimize Total Cost

s.t.   SupCon       {i in EUPT} : sum {j in CITY, k in DIST} QS[i,j,k] <= Sup[i]     ; # Total Quantity Shipped <= Supply Capacity
s.t.   DmdCon       {k in DIST} : sum {i in EUPT, j in CITY} QS[i,j,k] >= Dmd[k]     ; # Total Quantity Shipped >= Demand


data                                                                             ;

set    EUPT         :=          Hamburg      Marseilles         Liverpool        ;
set    CITY         :=          Norfolk      New_York           Savannah         ;
set    DIST         :=          Dallas       St_Louis           Chicago          ;

param  Cos          :           Norfolk      New_York           Savannah         :=
                    Hamburg     420          390                610
                    Marseilles  510          590                470
                    Liverpool   450          360                480              ;

param  Cod          :           Dallas       St_Louis           Chicago          :=
                    Norfolk     75           63                 81
                    New_York    125          110                95
                    Savannah    68           82                 95               ;

param  Sup          :=          Hamburg 55   Marseilles   78    Liverpool   37   ;
param  Dmd          :=          Dallas 60    St_Louis     45    Chicago     50   ;

option solver gurobi                        ;
solve                                       ;
option display_width 200, display_1col 0    ;
display QS                                  ;
```

SOLUTION PROBLEM M 3dim SumQ

```
set    EUPT         ordered                                          ; # EU Ports: Hamburg, Marseilles, Liverpool
set    CITY         ordered                                          ; # US Cities: Norfolk, New_York, Savannah
set    DIST         ordered                                          ; # Distribution Centers: Dallas, St_Louis, Chicago

param  Cos          {i in EUPT, j in CITY}                           ; # Cost from each EU Port to each US City
param  Cod          {j in CITY, k in DIST}                           ; # Cost from each US City to each Distribution Center
param  Sup          {i in EUPT}                                      ; # Maximum Capacity per EU Port
param  Dmd          {k in DIST}                                      ; # Demand per Distribution Center

var    QS           {i in EUPT, j in CITY, k in DIST} integer >= 0; # Q Shipped from each Port to each City and Dist. Center
var    QC           {i in EUPT, j in CITY} integer >= 0              ; # Quantity Shipped from each Port to each City
var    QD           {j in CITY, k in DIST} integer >= 0              ; # Quantity Shipped from each City to each Distribution Center

minimize TotCst     : sum {i in EUPT, j in CITY, k in DIST} QS[i,j,k] * ( Cos[i,j] + Cod[j,k] )    ; # Minimize Total Cost

s.t.   SupCon       {i in EUPT} : sum {j in CITY, k in DIST} QS[i,j,k] <= Sup[i]      ; # Total Quantity Shipped <= Supply Capacity
s.t.   DmdCon       {k in DIST} : sum {i in EUPT, j in CITY} QS[i,j,k] >= Dmd[k]      ; # Total Quantity Shipped >= Demand
s.t.   SumQC        {i in EUPT, j in CITY} : QC[i,j] = sum {k in DIST} QS[i,j,k]      ; # Total Quantity Shipped >= Demand
s.t.   SumQD        {j in CITY, k in DIST} : QD[j,k] = sum {i in EUPT} QS[i,j,k]      ; # Total Quantity Shipped >= Demand

data                                                                                 ;

set    EUPT         :=          Hamburg      Marseilles         Liverpool           ;
set    CITY         :=          Norfolk      New_York           Savannah            ;
set    DIST         :=          Dallas       St_Louis           Chicago             ;
param  Cos          :           Norfolk      New_York           Savannah            :=
                    Hamburg     420          390                610
                    Marseilles  510          590                470
                    Liverpool   450          360                480                 ;
param  Cod          :           Dallas       St_Louis           Chicago             :=
                    Norfolk     75           63                 81
                    New_York    125          110                95
                    Savannah    68           82                 95                  ;
param  Sup          :=          Hamburg 55   Marseilles   78    Liverpool    37     ;
param  Dmd          :=          Dallas 60    St_Louis     45    Chicago      50     ;

option solver gurobi                        ;
solve                                       ;
option display_width 200, display_1col 0    ;
display QS                                  ;
display QC                                  ;
display QD                                  ;
```

SOLUTION PROBLEM Na

```
set    RESR          ordered                                ; # Resources: Man.Mch.Hrs, Man.Manpow, Dist.Manpow
set    PROD          ordered                                ; # Products: A, B, C, D, E

param  Hrs           {i in RESR, j in PROD}                 ; # Hours required per Resource per Product
param  Cap           {i in RESR}                            ; # Available Hours per Resource (Capacity)
param  Spr           {j in PROD}                            ; # Sale Price per unit per Product
param  Vco           {j in PROD}                            ; # Variable Cost per unit per Product

var    QP            {j in PROD} >= 0                       ; # Quantity to be produced per Product


maximize TotPrf     : sum {j in PROD} QP[j] * ( Spr[j] - Vco[j] )          ; # Maximize Total Profit

s.t.   CapCon        {i in RESR} : sum {j in PROD} QP[j] * Hrs[i,j] <= Cap[i]   ; # Capacity Constraint for each resource


data                                                                                                  ;

set    RESR          :=                 Man.Mch.Hrs  Man.Manpow   Dist.Manpow                        ;
set    PROD          :=                 A            B            C            D            E        ;

param  Hrs           :                  A            B            C            D            E        :=
                     Man.Mch.Hrs        4            3            12           6            3
                     Man.Manpow         6            10           7            8            12
                     Dist.Manpow        4            2            5            5            5        ;

param                :                  Cap                                                          :=
                     Man.Mch.Hrs        1000
                     Man.Manpow         1100
                     Dist.Manpow         600                                                         ;

param  Spr           :=                 A  97        B 110        C 142        D 112        E  97    ;
param  Vco           :=                 A  58        B  75        C  72        D  55        E  40    ;


option solver gurobi                          ;
solve                                         ;
option display_width 200, display_1col 0      ;

display QP                                    ;
```

SOLUTION PROBLEM Nb

```
set    RESR          ordered                                ; # Resources: Man.Mch.Hrs, Man.Manpow, Dist.Manpow
set    PROD          ordered                                ; # Products: A, B, C, D, E

param  Hrs           {i in RESR, j in PROD}                 ; # Hours required per Resource per Product
param  Cap           {i in RESR}                            ; # Available Hours per Resource (Capacity)
param  Spr           {j in PROD}                            ; # Sale Price per unit per Product
param  Vco           {j in PROD}                            ; # Variable Cost per unit per Product

var    QP            {j in PROD} integer >= 0               ; # Quantity to be produced per Product


maximize TotPrf     : sum {j in PROD} QP[j] * ( Spr[j] - Vco[j] )              ; # Maximize Total Profit

s.t.   CapCon        {i in RESR} : sum {j in PROD} QP[j] * Hrs[i,j] <= Cap[i]   ; # Capacity Constraint for each resource


data                                                                                                    ;

set    RESR          :=                 Man.Mch.Hrs  Man.Manpow   Dist.Manpow                          ;
set    PROD          :=                 A            B            C            D            E          ;

param  Hrs           :                  A            B            C            D            E          :=
                     Man.Mch.Hrs        4            3            12           6            3
                     Man.Manpow         6            10           7            8            12
                     Dist.Manpow        4            2            5            5            5          ;

param                :                  Cap                                                           :=
                     Man.Mch.Hrs        1000
                     Man.Manpow         1100
                     Dist.Manpow         600                                                          ;

param  Spr           :=                 A  97        B 110        C 142        D 112        E  97      ;
param  Vco           :=                 A  58        B  75        C  72        D  55        E  40      ;


option solver gurobi                          ;
solve                                         ;
option display_width 200, display_1col 0      ;

display QP                                    ;
```

SOLUTION PROBLEM Nc

```
set    RESR          ordered                                ; # Resources: Man.Mch.Hrs, Man.Manpow, Dist.Manpow
set    PROD          ordered                                ; # Products: A, B, C, D, E

param  Hrs           {i in RESR, j in PROD}                 ; # Hours required per Resource per Product
param  Cap           {i in RESR}                            ; # Available Hours per Resource (Capacity)
param  Spr           {j in PROD}                            ; # Sale Price per unit per Product
param  Vco           {j in PROD}                            ; # Variable Cost per unit per Product
param  Fco           {j in PROD}                            ; # Fixed Cost per Product
param  M                                                    ; # Represents a very large number

var    QP            {j in PROD} integer >= 0               ; # Quantity to be produced per Product
var    PP            {j in PROD} binary                     ; # 1 if the Product is produced, 0 otherwise

maximize TotPrf     : sum {j in PROD} ( QP[j] * ( Spr[j] - Vco[j] ) - PP[j] * Fco[j] )      ; # Maximize Total Profit

s.t.   CapCon {i in RESR} : sum {j in PROD} QP[j] * Hrs[i,j] <= Cap[i]   ; # Capacity Constraint for each resource
s.t.   FcoCon {j in PROD} : QP[j] <= PP[j] * M                           ; # QP equal zero for Products not produced

data                                                                                                    ;

set    RESR          :=                 Man.Mch.Hrs  Man.Manpow   Dist.Manpow                          ;
set    PROD          :=                 A            B            C            D            E          ;

param  Hrs           :                  A            B            C            D            E          :=
                     Man.Mch.Hrs        4            3            12           6            3
                     Man.Manpow         6            10           7            8            12
                     Dist.Manpow        4            2            5            5            5          ;

param                :                  Cap                                                          :=
                     Man.Mch.Hrs        1000
                     Man.Manpow         1100
                     Dist.Manpow         600                                                         ;

param  Spr           :=                 A  97        B 110        C 142        D 112        E  97      ;
param  Vco           :=                 A  58        B  75        C  72        D  55        E  40      ;
param  Fco           :=                 A 400        B 300        C 700        D 500        E 600      ;
param  M             :=                 1000000                                                      ;

option solver gurobi                          ;
solve                                         ;
option display_width 200, display_1col 0      ;

display QP,PP                                 ;
```