Iron Ore Production and Logistics via Integer Programming

Marcus Poggi
Departamento de Informática, PUC-Rio, Brazil
poggi@inf.puc-rio.br

Marcelo Reis, Cristiane Ferreira, Mario Lobato, André Lima
GAPSO Inc, Brazil
marcelo, cristiane, mlobato, andlima@gapso.com.br

July 15, 2011

Long Term Planning

- Horizon is 10 years
- Years can be considered independently
- Demands can be aggregated to around 90
- Integral solution is not required
- A Linear Program is solved for each year
- Takes less than 2 min
- CPLEX 12 is used for all modules

Medium Term Planning

- Horizon is from 1 to 3 years
- Granularity is month
- Stocks from one month to the next must be considered
- Local Mode: Runs for a subset of Productive Complexes
- Global Mode: Runs for all Supply Network
- Demands can be aggregated
- For one year plans: Local Mode averages 450 demands while Global Mode has 850
- For each demand there may be a minimum percentage of the demand that can be fulfilled
- No minimum percentage: Linear Programming
- Global mode with minimum percentage to fulfill: Runs in less than 5 min
- This is the most popular mode!

Parameters

- $Cp(p)$ - Class of product p.
- $Ctp(p)$ - Category of product p.
- $Pre(i)$ - Instant preceding instant i.
- $Pos(i)$ - Instant succeeding instant i.
- $PP(cp)$ - Products from the class of products cp.
- $Loc(f)$ - Location of provider f.
- $Loc(pn)$ - Location of screening (peneiramento) pn.
- $Loc(c)$ - Location of client c.
- $Forn(l)$ - Provider of location l, should there be one.
- $Peneir(l)$ - Screening (Peneiramento) of location l, should there be one.
- $Cli(l)$ - Client of location l, stbo.
- $ClPr(d)$ - Demand Product d.
- $Cli(d)$ - Demand Client d.
- $MaxDem(d)$ - Maximum Fulfillment of demand d in the year.
- $MinDem(d)$ - Minimum Fulfillment of demand d in the year.
- $LoteMin(d)$ - Lot: Minimum Quantity in the Fulfillment of demand d at once
- $DemMensal(d, i)$ - Quantity of demand d fixed on instant i.
- $Pd(s, lo, ld, i, p)$ - Final product at destination of product p transportation using orig-dest lo to location ld using submodal s on instant i
- $CapCS(l, s, i)$ - Load capacidty (in tons) of submodal s at location l on instant i
- $CapDS(l, s, i)$ - Unload capacity(in tons) of submodal s at location l on instant i

Parameters

- $CapEstoqueCp(l, cp, i)$ - Stock Capacity (in tons) on location l for product class cp on instant i
- $CapEstoque(l, i)$ - Stock Capacity (in tons) on location l on instant i
- $CapS(cp, s, i)$ - Capacidade do submodal s carregando produtos da classe cp no instante i.
- $CapVazao(lo, ld, s, i)$ - Throughput capacity from lo to ld for submodal s on instant i.
- $CapVazaoSr(sr, i)$ - Throughput capacity of sub-network sr on instant i.
- $CustoEstoque(l, i, cp, d)$ - Stock cost, during instant i, for one ton of product class cp in local l to fulfill demand d.
- $CustoForn(f, l, d, i, cp)$ - Custo de comprar do fornecedor f uma tonelada da classe de produto cp para atender a demanda d na localidade l no instante i.
- $CustoProd(l, i, cp)$ - Custo de produzir uma tonelada da classe de produto cp na localidade l no instante i.
- $CustoTransp(s, lo, ld, i, cp)$ - Custo de transportar uma tonelada da classe de produto cp no trecho de lo para ld utilizando o submodal s no instante i.
- $MinVazaoTr(lo, ld, s, i)$ - Mínimo de vazão (em número de submodais) no trecho de lo para ld para o submodal s no instante i.
- $MinVazaoTrCp(cp, lo, ld, s, i)$ - Mínimo de vazão (em número de submodais) para a classe de produto cp no trecho de lo para ld para o submodal s no instante i.
- $MinVazaoSr(sr, i)$ - Mínimo de vazão (em número de submodais) na subrede sr para o submodal s no instante i.
- $MinVazaoSrCp(cp, sr, i)$ - Mínimo de vazão (em número de submodais) para a classe de produto cp na subrede sr no instante i.
- $PercMaxGP(d, i, gp)$ - Percentual máximo de participação do grupo gp no atendimento da demanda d no instante i.
- $PercMinGP(d, i, gp)$ - Percentual mínimo de participação do grupo gp no atendimento da demanda d no instante i.

Parameters

- $PercMaxCP(d, i, cp)$ - Percentual máximo de participação da classe cp no atendimento da demanda d no instante i.
- $PercMinCP(d, i, cp)$ - Percentual mínimo de participação da classe cp no atendimento da demanda d no instante i.
- $QtdMaxForn(f, i)$ - Quantidade máxima que pode ser comprada do fornecedor f no instante i.
- $QtdMaxForn(f, i, cp)$ - Quantidade máxima que pode ser comprada do fornecedor f da classe de produto cp no instante i.
- $QtdMinForn(f, i)$ - Quantidade mínima que pode ser comprada do fornecedor f no instante i.
- $QtdMinForn(f, i, cp)$ - Quantidade mínima que pode ser comprada do fornecedor f da classe de produto cp no instante i.
- $QtdProd(l, i)$ - Quantidade máxima produzida de qualquer produto na localidade l no instante i.
- $QtdProd(l, cp, i)$ - Quantidade máxima produzida do produto da classe cp na localidade l no instante i.
- $QtdMinProd(l, i)$ - Quantidade mínima que pode ser produzida na localidade l no instante i.
- $QtdMinProd(l, i, cp)$ - Quantidade mínima que pode ser produzida na localidade l da classe de produto cp no instante i.
- $QMin(d, pr, i)$ - Qualidade mínima da propriedade pr para a demanda d no instante i.
- $QMax(d, pr, i)$ - Qualidade máxima da propriedade pr para a demanda d no instante i.
- $QMeta(d, pr, i)$ - Qualidade meta da propriedade pr para a demanda d no instante i.
- $Qual(p, pr)$ - Qualidade da propriedade pr para o produto p.
- $Rec(d, i)$ - Receita por tonelada atendida da demanda d no instante i.
- $MinEstoqueCp(l, cp, i)$ - Estoque estratégico da classe de produto cp na localidade l no instante i.
- $MinEstoque(l, i)$ - Estoque estratégico total na localidade l no instante i.
- $EstIni(l, cp)$ - Estoque inicial da classe de produto cp na localidade l.

Variables

- $xp_{l,p,i}$ - quantity (in tons) produced of product p by location l on instant i
- $xf_{f,l,p,i}$ - quantity (in tons) of product p bought from provider f to be delivered at location l on instant i
- $xt_{s,lo,ld,i,p,pd}$ - quantity (in tons) of product p that will converted into pd transported by submodal s from location lo to ld on instant i. Only orig-dest with handling factor will have $p \neq pd$.
- $xb_{l,d,i,po,pd}$ - quantity (in tons) of product po blended to form product pd at location l on instant i to fulfill demand d.
- $el_{l,io,id,p}$ - quantity (in tons) of stock of product p at location l with closure on instant io and opening on instant id
- $xc_{c,d,l,p,i}$ - quantity received of product p by client c to fulfill demand d at location l on instant i.
- $xpen_{l,p,i,pn,cf,pd}$ - quantity screened of product p at location l on instant i using screening configution cf of screening pn resulting in product pd
- $fminf_{f,i}$ - slack variable for constraint of minimum to be bought from provider f on instant i
- $fminfcp_{cp,f,i}$ - slack variable for constraint of minimum to be bought of products of class cp from provider f on instant i
- $fminp_{l,i}$ - slack variable for constraint of minimum to be produced at location l on instant i
- $fminpcp_{cp,l,i}$ - slack variable for constraint of minimum to be produced of products of class cp at location l on instant i
- $fminel_{l,i}$ - slack variable for constraint of minimum to be stocked at location l on instant i.
- $fminelcp_{cp,l,i}$ - slack variable for constraint of minimum to be stocked of products of class cp at location l on instant i

Variables

- $fvazmintr_{lo,ld,s,cp,i}$ - slack variable for constraint of minimum throughput of products of class *cp* for location *lo* to *ld* using submodal *s* on instant *i*
- $fvazminsr_{sr,cp,i}$ - slack variable for constraint of minimum throughput of products of class *cp* for in the sub-network *sr* on instant *i*
- $fmindem_{c,d,l}$ - slack variable for constraint of minimum to be fulfilled of demand *d* of client *c* at location *l*.
- $fqmin_{d,pr,i}$ - slack variable for constraint of minimum quality for property *pr* for demand *d* on instant *i*
- $fqmax_{d,pr,i}$ - slack variable for constraint of maximum quality for property *pr* for demand *d* on instant *i*
- $fqmetaU_{d,pr,i}$ - slack variable for constraint of target (upper) quality for property *pr* for demand *d* on instant *i*
- $fqmetaL_{d,pr,i}$ - slack variable for constraint of target (lower) quality for property *pr* for demand *d* on instant *i*
- $xblm_{d,i}$ - [MT] binary variable identifying whether demand *d*, with minimum lot, will have a (partial) fulfillment or not on instant *i*.
- $fmfixdem_{d,i}$ - [MT] slack variable for partial fulfillment of demand *d* fixed on instant *i*
- $fmdistunifano_{d,i}$ - [MT] slack variable (lower) for annual distribution of demand *d* on instant *i*
- $fpdistunifano_{d,i}$ - [MT] slack variable (upper) for annual distribution of demand *d* on instant *i*
- $fmdistunif_{d,i}$ - [MT] slack variable (lower) for the distribution of demand *d* on instant *i*
- $fpdistunif_{d,i}$ - [MT] slack variable (upper) for the distribution of demand *d* on instant *i*

Objective Function

$$
\begin{aligned}
\text{MAX} \quad & \sum_{p \in P} \sum_{c \in C} \sum_{d \in D} \sum_{i \in I} \sum_{l \in L} Rec(d, i) \times xc_{c,d,l,p,i} \\
& - \sum_{p \in P} \sum_{i \in I} \sum_{l \in L} CustoProd(l, i, Cp(p)) \times xp_{l,p,i} \\
& - \sum_{p \in P} \sum_{f \in F} \sum_{i \in I} \sum_{l \in L} CustoForn(f, l, i, Cp(p)) \times xf_{f,l,p,i} \\
& - \sum_{p \in P} \sum_{s \in S} \sum_{d \in D} \sum_{i \in I} \sum_{lo \in L} \sum_{ld \in L} CustoTransp(s, lo, ld, i, Cp(p)) \times xt_{s,lo,ld,d,i,p,Pd(s,lo,ld,i,p)} \\
& - \sum_{p \in P} \sum_{io \in I} \sum_{id \in I} \sum_{l \in L} CustoEstoque(l, io, Cp(p), d) \times el_{l,io,id,p}
\end{aligned}
$$

Product Flow Conservation

$$
\begin{aligned}
& el_{l,Pre(i),i,p} + xp_{l,p,i} + xf_{f,l,p,i} + \sum_{lo \in L} \sum_{s \in S} \sum_{po \in P} xt_{s,lo,l,i,po,p} \\
& + \sum_{d \in D} \sum_{pd \in P} xb_{l,d,i,p,pd} + \sum_{cf \in CF(pn)} \sum_{po \in P} xpen_{l,po,i,pn,cf,p} \\
& = \\
& \sum_{d \in D} \sum_{po \in P} xb_{l,d,i,po,p} + \sum_{ld \in L} \sum_{s \in S} xt_{s,l,ld,i,p,Pd(s,l,ld,i,p)} \\
& + \sum_{cf \in CF(pn)} \sum_{pd \in P} xpen_{l,p,i,pn,cf,pd} + el_{l,i,Pos(i),p} + \sum_{d \in D} xc_{c,d,l,p,i} \\
& \forall p \in P \quad \forall i \in I \quad \forall l \in L \quad [f = Forn(l),\, pn = Peneir(l),\, c = Cli(l)]
\end{aligned}
$$

Load

$$\sum_{ld \in L} \sum_{p \in P} \frac{xt_{s,l,ld,i,p,Pd(s,l,ld,i,p)}}{CapS(Cp(p), s, i)} \leq CapCS(l, s, i) \qquad \forall l \in L \quad \forall s \in S \quad \forall i \in I$$

Unload

$$\sum_{lo \in L} \sum_{p \in P} \frac{xt_{s,lo,l,i,p,Pd(s,lo,l,i,p)}}{CapS(Cp(p), s, i)} \leq CapDS(l, s, i) \qquad \forall l \in L \quad \forall s \in S \quad \forall i \in I$$

Class of Product

$$\sum_{p \in P(cp)} el_{l,i,Pos(i),p} \leq CapEstoqueCp(l, cp, i) \qquad \forall cp \in CP \quad \forall i \in I \quad \forall l \in L$$

Whole Stock

$$\sum_{p \in P} el_{l,i,Pos(i),p} \leq CapEstoque(l, i) \qquad \forall i \in I \quad \forall l \in L$$

Maximum Fulfillment

$$\sum_{p \in PD(d)} \sum_{i \in I} xc_{c,d,l,p,i} \leq MaxDem(d) \qquad \forall d \in D \quad [c = Cli(d), l = Loc(c)]$$

Minimum Fulfillment

$$\sum_{p \in PD(d)} \sum_{i \in I} xc_{c,d,l,p,i} + fmindem_{c,d,l} \geq MinDem(d) \qquad \forall d \in D \quad [c = Cli(d), l = Loc(c)]$$

Production by Class of Product and Instant

$$\sum_{p \in PP(cp)} xp_{l,p,i} \leq QtdProd(l, cp, i) \qquad \forall i \in I \quad \forall cp \in CP \quad \forall l \in L$$

Production by Instant

$$\sum_{p \in P} xp_{l,p,i} \leq QtdProd(l, i) \qquad \forall i \in I \quad \forall l \in L$$

Transport Capacity Orig-Dest and Instant

$$\sum_{p \in P} \frac{xt_{s,lo,ld,i,p,Pd(s,lo,ld,i,p)}}{CapS(Cp(p),s,i)} \leq CapVazao(lo,ld,s,i) \qquad \forall i \in I \quad \forall s \in S \quad \forall lo \in L \quad \forall ld \in L$$

Minimum Throughput Orig-Dest, Class of Product and Instant

$$\sum_{p \in PP(cp)} \frac{xt_{s,lo,ld,i,p,Pd(s,lo,ld,i,p)}}{CapS(cp,s,i)} + fvazmintr_{lo,ld,s,cp,i} \geq MinVazaoTrCp(cp,lo,ld,s,i)$$

$$\forall i \in I \quad \forall cp \in CP \quad \forall s \in S \quad \forall lo \in L \quad \forall ld \in L$$

Minimum Throughput Orig-Dest and Instant

$$\sum_{p \in P} \frac{xt_{s,lo,ld,i,p,Pd(s,lo,ld,i,p)}}{CapS(Cp(p),s,i)} \geq MinVazaoTr(lo,ld,s,i) \qquad \forall i \in I \quad \forall s \in S \quad \forall lo \in L \quad \forall ld \in L$$

Transport Capacity by Sub-Network and Instant

$$\sum_{s \in S} \sum_{lo \in L} \sum_{ld \in L} \sum_{p \in P} \frac{xt_{s,lo,ld,i,p,Pd(s,lo,ld,i,p)}}{CapS(Cp(p), s, i)} \leq CapVazaoSr(sr, i) \qquad \forall sr \in SR \quad \forall i \in I$$

Minimum Throughput by Sub-Network, Class of Product and Instant

$$\sum_{s \in S} \sum_{lo \in L} \sum_{ld \in L} \sum_{p \in PP(cp)} \frac{xt_{s,lo,ld,i,p,Pd(s,lo,ld,i,p)}}{CapS(cp, s, i)} + fvazminsr_{sr,cp,i} \geq MinVazaoSrCp(cp, sr, i)$$

$$\forall i \in I \quad \forall cp \in CP \quad \forall sr \in SR$$

Minimum Throughput by Sub-Network and Instant

$$\sum_{s \in S} \sum_{lo \in L} \sum_{ld \in L} \sum_{p \in P} \frac{xt_{s,lo,ld,i,p,Pd(s,lo,ld,i,p)}}{CapS(Cp(p), s, i)} \geq MinVazaoSr(sr, i) \qquad \forall i \in I \quad \forall sr \in SR$$

Demand Minimum Quality

$$\sum_{lb \in LB} \sum_{p \in P} \sum_{po \in P \setminus PC(p)} Qual(po, pr) \times xb_{lb,d,i,po,p} + fqmin_{d,pr,i} \geq$$

$$\text{QMin(d,pr,i)} \times \sum_{lb \in LB} \sum_{p \in P} \sum_{po \in P \setminus PC(p)} xb_{lb,d,i,po,p}$$

Demand Maximum Quality

$$\sum_{lb \in LB} \sum_{p \in P} \sum_{po \in P \setminus PC(p)} Qual(po, pr) \times xb_{lb,d,i,po,p} - fqmax_{d,pr,i} \leq$$

$$\text{QMax(d,pr,i)} \times \sum_{lb \in LB} \sum_{p \in P} \sum_{po \in P \setminus PC(p)} xb_{lb,d,i,po,p}$$

Demand Target Quality

$$\sum_{lb \in LB} \sum_{p \in P} \sum_{po \in P \setminus PC(p)} Qual(po, pr) \times xb_{lb,d,i,po,p} + fqmetaU_{d,pr,i} - fqmetaL_{d,pr,i} =$$

$$\text{QMeta(d,pr,i)} \times \sum_{lb \in LB} \sum_{p \in P} \sum_{po \in P \setminus PC(p)} xb_{lb,d,i,po,p}$$

$$\forall i \in I \quad \forall pr \in PR \quad \forall d \in D$$

Demand - Maximum Percentage of a Group of Products in the Blending

$$\sum_{lb \in LB} \sum_{p \in PD(d)} \sum_{po \in PG(gp) \setminus PC(p)} xb_{lb,d,i,po,p} \leq PercMaxGP(d,i,gp) \times \sum_{lb \in LB} \sum_{p \in PD(d)} \sum_{po \in P \setminus PC(p)} xb_{lb,d,i,po,p}$$

$$\forall i \in I \quad \forall gp \in GP \quad \forall d \in D$$

Demand - Minimum Percentage of a Group of Products in the Blending

$$\sum_{lb \in LB} \sum_{p \in PD(d)} \sum_{po \in PG(gp) \setminus PC(p)} xb_{lb,d,i,po,p} \geq PercMinGP(d,i,gp) \times \sum_{lb \in LB} \sum_{p \in PD(d)} \sum_{po \in P \setminus PC(p)} xb_{lb,d,i,po,p}$$

$$\forall i \in I \quad \forall gp \in GP \quad \forall d \in D$$

Demand - Maximum Percentage of a Product in the Blending

$$\sum_{lb \in LB} \sum_{p \in PD(d)} \sum_{po \in PP(cp) \setminus PC(p)} xb_{lb,d,i,po,p} \leq PercMaxCP(d,i,cp) \times \sum_{lb \in LB} \sum_{p \in PD(d)} \sum_{po \in P \setminus PC(p)} xb_{lb,d,i,po,p}$$

$$\forall i \in I \quad \forall cp \in CP \quad \forall d \in D$$

Demand - Minimum Percentage of a Product in the Blending

$$\sum_{lb \in LB} \sum_{p \in PD(d)} \sum_{po \in PP(cp) \setminus PC(p)} xb_{lb,d,i,po,p} \geq PercMinCP(d,i,cp) \times \sum_{lb \in LB} \sum_{p \in PD(d)} \sum_{po \in P \setminus PC(p)} xb_{lb,d,i,po,p}$$

$$\forall i \in I \quad \forall cp \in CP \quad \forall d \in D$$

Maximum buy of iron ore from a provider and class of product

$$\sum_{p \in PP(cp)} xf_{f,l,p,i} \leq QtdMaxForn(f,i,cp) \qquad \forall f \in F \quad \forall i \in I \quad \forall cp \in CP \quad [l = Loc(f)]$$

Maximum buy of iron ore from a provider

$$\sum_{p \in P} xf_{f,Loc(f),p,i} \leq QtdMaxForn(f,i) \qquad \forall f \in F \quad \forall i \in I$$

Minimum buy of iron ore from a provider and class of product

$$\sum_{p \in PP(cp)} xf_{f,Loc(f),p,i} + fminfcp_{cp,f,i} \geq QtdMinForn(f,i,cp) \qquad \forall f \in F \quad \forall i \in I \quad \forall cp \in CP$$

Minimum buy of iron ore from a provider

$$\sum_{p \in P} xf_{f,Loc(f),p,i} + fminf_{f,i} \geq QtdMinForn(f,i) \qquad \forall f \in F \quad \forall i \in I$$

Minimum Production Per location and class of product

$$\sum_{p \in PP(cp)} xp_{l,p,i} + fminpcp_{cp,l,i} \geq QtdMinProd(l, i, cp) \quad \forall l \in L \quad \forall i \in I \quad \forall cp \in CP$$

Minimum Production by Location

$$\sum_{p \in P} xp_{l,p,i} + fminp_{l,i} \geq QtdMinProd(l, i) \quad \forall l \in L \quad \forall i \in I$$

Minimum Stock by Location and class of product

$$\sum_{p \in P(cp)} el_{l,i,Pos(i),p} + fminelcp_{cp,l,i} \geq MinEstoqueCp(l, cp, i) \quad \forall cp \in CP \quad \forall i \in I \quad \forall l \in L$$

Minimum Stock by Location

$$\sum_{p \in P} el_{l,i,Pos(i),p} + fminel_{l,i} \geq MinEstoque(l, i) \quad \forall i \in I \quad \forall l \in L \qquad (1)$$

Uniform Distribution of the Slacks

$$fpdistunifano_{d,i} + fmdistunifano_{d,i} = \frac{1}{12}\sum_{i' \in I}(fpdistunifano_{d,i'} + fmdistunifano_{d,i'}) + fpdistunif_{d,i} - fmdistunif_{d,i}$$

$$\forall i \in I \quad \forall d \in D$$

Percentage of Products in the screening

$$xpen_{l,p,i,pn,cf,pd} = PercPen(pn, cf, pd) \sum_{pd' \in PP(cf)} xpen_{l,p,i,pn,cf,pd'}$$

$$\forall pn \in PN \quad \forall cf \in CF(pn) \quad \forall pd \in PP(cf) \quad \forall i \in I \quad [p = Prod(cf), l = Loc(pn)]$$

Minimum Quantity (Lot) for each demand on instant i

$$xc_{c,d,l,p,i} \geq LoteMin(d) \times xblm_{d,i} \qquad \forall i \in I \quad \forall d \in D \quad \forall p \in P \quad [c = Cli(d), l = Loc(c)]$$

Maximum delivery for each demand on instant i

$$xc_{c,d,l,p,i} \leq MaxDem(d) \times xblm_{d,i} \qquad \forall i \in I \quad \forall d \in D \quad \forall p \in P \quad [c = Cli(d), l = Loc(c)]$$

Fixed Demand Fulfillment on instant i

$$xc_{c,d,l,p,i} = DemMensal(d, i) - fmfixdem_{d,i} \qquad \forall i \in I \quad \forall d \in D \quad \forall p \in P \quad [c = Cli(d), l = Loc(c)]$$

Uniformly Distributed Demand on instant i

$$xc_{c,d,l,p,i} = DemUnif(d, i) + fpdistunifano_{d,i} - fmdistunifano_{d,i}$$

$$\forall i \in I \quad \forall d \in D \quad \forall p \in P \quad [c = Cli(d), l = Loc(c)]$$

Short Term Planning

- Horizon is from 3 to 5 months
- Granularity is day
- Runs for each productive system
- Shall run for a higher hierarchy (complexes seen as mines)
- Shipments CANNOT be aggregated
- Number of iron ore "sale" demands considered is around 30 (aggregated)
- Number "feed" demands (for pellets) considered is around 50 (aggregated)
- Number of shipments is about 150
- Number of stock piles go above 200
- The MIP model has about 200.000 binary or integer variables
- Continuous variables count is over 300.000

Short Term Planning - Difficulties

- Transportation is in "Lots" (by train 80 to 100 cars)

Short Term Planning - Difficulties

- Transportation is in "Lots" (by train 80 to 100 cars)
- Lots are broken into 4 parts (25% each) (our solver is still only good for 2 parts)

Short Term Planning - Difficulties

- Transportation is in “Lots” (by train 80 to 100 cars)
- Lots are broken into 4 parts (25% each) (our solver is still only good for 2 parts)
- Quality assurance for each stockpile instead of the shipment

  This implies to treat stockpiles as demands

Short Term Planning - Difficulties

- Transportation is in "Lots" (by train 80 to 100 cars)
- Lots are broken into 4 parts (25% each) (our solver is still only good for 2 parts)
- Quality assurance for each stockpile instead of the shipment

  This implies to treat stockpiles as demands
- Shipments "must" follow (or convince everything was tried) the FIFO rule.

  Smallest ETA first, if it has arrived then the smallest arrival time first.

  This order is by product. Hard to impose this order in the MIP model, since some shipments may not be attended at all.

Short Term Planning - Difficulties

- Transportation is in “Lots” (by train 80 to 100 cars)
- Lots are broken into 4 parts (25% each) (our solver is still only good for 2 parts)
- Quality assurance for each stockpile instead of the shipment

  This implies to treat stockpiles as demands
- Shipments “must” follow (or convince everything was tried) the FIFO rule.

  Smallest ETA first, if it has arrived then the smallest arrival time first.

  This order is by product. Hard to impose this order in the MIP model, since some shipments may not be attended at all.
- Shipments must be totally completed before the next one starts.

Squeezing Algorithm

1. **Solve a relaxed model**

   Remove Transportation from model. All associated constraints: throughput, load and unload capacities.

   The Blending suggested by the relaxed model determines which blending variables will appear in the "*complete*" model.

Squeezing Algorithm

1. **Solve a relaxed model**

   Remove Transportation from model. All associated constraints: throughput, load and unload capacities.

   The Blending suggested by the relaxed model determines which blending variables will appear in the *"complete"* model.

2. **Solve the Linear Relaxation of the complete model**

Squeezing Algorithm

1. **Solve a relaxed model**

   Remove Transportation from model. All associated constraints: throughput, load and unload capacities.

   The Blending suggested by the relaxed model determines which blending variables will appear in the *"complete"* model.

2. **Solve the Linear Relaxation of the complete model**
3. Fix to zero all blending variables at zero in the Linear Relaxation

Squeezing Algorithm

1. **Solve a relaxed model**
   Remove Transportation from model. All associated constraints: throughput, load and unload capacities.

   The Blending suggested by the relaxed model determines which blending variables will appear in the *"complete"* model.
2. **Solve the Linear Relaxation of the complete model**
3. Fix to zero all blending variables at zero in the Linear Relaxation
4. **Back to MIP run for a fixed amount of time or until a integer feasible solution is found**

Squeezing Algorithm

1. **Solve a relaxed model**
   Remove Transportation from model. All associated constraints: throughput, load and unload capacities.

   The Blending suggested by the relaxed model determines which blending variables will appear in the *"complete"* model.
2. **Solve the Linear Relaxation of the complete model**
3. Fix to zero all blending variables at zero in the Linear Relaxation
4. **Back to MIP run for a fixed amount of time or until a integer feasible solution is found**
5. For each type of product, order the shipments by ETA

Squeezing Algorithm

1. **Solve a relaxed model**
   Remove Transportation from model. All associated constraints: throughput, load and unload capacities.
   The Blending suggested by the relaxed model determines which blending variables will appear in the *"complete"* model.
2. **Solve the Linear Relaxation of the complete model**
3. Fix to zero all blending variables at zero in the Linear Relaxation
4. **Back to MIP run for a fixed amount of time or until a integer feasible solution is found**
5. For each type of product, order the shipments by ETA
6. **Set lower priority shipment (higher ETA's) to zero (fulfillments are not allowed)**
   The higher priority ones (smaller ETA's) will be allowed to be fulfilled

Squeezing Algorithm

1. **Solve a relaxed model**

   Remove Transportation from model. All associated constraints: throughput, load and unload capacities.

   The Blending suggested by the relaxed model determines which blending variables will appear in the *"complete"* model.
2. **Solve the Linear Relaxation of the complete model**
3. Fix to zero all blending variables at zero in the Linear Relaxation
4. **Back to MIP run for a fixed amount of time or until a integer feasible solution is found**
5. For each type of product, order the shipments by ETA
6. **Set lower priority shipment (higher ETA's) to zero (fulfillments are not allowed)**

   The higher priority ones (smaller ETA's) will be allowed to be fulfilled
7. Let D be the last day for the allowed shipments to be fulfilled (we use a different D for each type of product). This is done several times. The D's are slided from the first day to the last day on the planning horizon.

Squeezing Algorithm

1. **Solve a relaxed model**

   Remove Transportation from model. All associated constraints: throughput, load and unload capacities.

   The Blending suggested by the relaxed model determines which blending variables will appear in the *"complete"* model.

2. **Solve the Linear Relaxation of the complete model**
3. Fix to zero all blending variables at zero in the Linear Relaxation
4. **Back to MIP run for a fixed amount of time or until a integer feasible solution is found**
5. For each type of product, order the shipments by ETA
6. **Set lower priority shipment (higher ETA's) to zero (fulfillments are not allowed)**

   The higher priority ones (smaller ETA's) will be allowed to be fulfilled

7. Let D be the last day for the allowed shipments to be fulfilled (we use a different D for each type of product). This is done several times. The D's are slided from the first day to the last day on the planning horizon.
8. Mutation

Squeezing Algorithm

1. **Solve a relaxed model**

   Remove Transportation from model. All associated constraints: throughput, load and unload capacities.

   The Blending suggested by the relaxed model determines which blending variables will appear in the *"complete"* model.
2. **Solve the Linear Relaxation of the complete model**
3. Fix to zero all blending variables at zero in the Linear Relaxation
4. **Back to MIP run for a fixed amount of time or until a integer feasible solution is found**
5. For each type of product, order the shipments by ETA
6. **Set lower priority shipment (higher ETA's) to zero (fulfillments are not allowed)**

   The higher priority ones (smaller ETA's) will be allowed to be fulfilled
7. Let D be the last day for the allowed shipments to be fulfilled (we use a different D for each type of product). This is done several times. The D's are slided from the first day to the last day on the planning horizon.
8. Mutation
9. Try to fulfill shipments

Squeezing Algorithm

1. **Solve a relaxed model**
   Remove Transportation from model. All associated constraints: throughput, load and unload capacities.
   The Blending suggested by the relaxed model determines which blending variables will appear in the *"complete"* model.
2. **Solve the Linear Relaxation of the complete model**
3. Fix to zero all blending variables at zero in the Linear Relaxation
4. **Back to MIP run for a fixed amount of time or until a integer feasible solution is found**
5. For each type of product, order the shipments by ETA
6. **Set lower priority shipment (higher ETA's) to zero (fulfillments are not allowed)**
   The higher priority ones (smaller ETA's) will be allowed to be fulfilled
7. Let D be the last day for the allowed shipments to be fulfilled (we use a different D for each type of product). This is done several times. The D's are slided from the first day to the last day on the planning horizon.
8. Mutation
9. Try to fulfill shipments
10. Allow the next set of priority shipments to be fulfilled

Squeezing Algorithm

1. **Solve a relaxed model**

   Remove Transportation from model. All associated constraints: throughput, load and unload capacities.

   The Blending suggested by the relaxed model determines which blending variables will appear in the *"complete"* model.
2. **Solve the Linear Relaxation of the complete model**
3. Fix to zero all blending variables at zero in the Linear Relaxation
4. **Back to MIP run for a fixed amount of time or until a integer feasible solution is found**
5. For each type of product, order the shipments by ETA
6. **Set lower priority shipment (higher ETA's) to zero (fulfillments are not allowed)**

   The higher priority ones (smaller ETA's) will be allowed to be fulfilled
7. Let D be the last day for the allowed shipments to be fulfilled (we use a different D for each type of product). This is done several times. The D's are slided from the first day to the last day on the planning horizon.
8. Mutation
9. Try to fulfill shipments
10. Allow the next set of priority shipments to be fulfilled
11. Repeat steps 6 to 10 until all the shipments were tried out

Mutation

1. Selects variable types to be fixed
2. Removing all CPLEX heuristics, run as MIP a very short time and return the Best Node.
3. Randomly select 40% of the variables of the type chosen to the value in the current solution.
4. Run the complete model
5. Repeat steps 2 to 4 until the gap between the current objective function value and the Best Node on step 2 becomes smaller enough (1% is used)

Try to fulfill shipments

- Fix all completely fulfilled shipments
- For all other shipments (including the partially fulfilled)

1. Set all as not allowed to be fulfilled
2. One by one, allow all possible blending and and all possible transportation. If any of those variables were fixed, unfix it.
3. Run the complete model.
4. Go sliding one day at a time the D value until the last day in the planning horizon is reached. Stop if the shipment considered is completely fulfilled (we try for 10 days, then feasibility is checked on the last day of the horizon).
5. If the shipment could be fulfilled, fix it as fulfilled. Otherwise, fix it as not to be fulfilled and do not try to fulfill any other shipment for the same product.
6. Fix back to zero all the blending and transportation variables that were fixed on step 33 of the main algorithm and were not used to fulfill this shipment.
7. Repeat steps 2 to 10 until until all unfulfilled shipments were tried out.

Allow the next set of priority shipments to be fulfilled

1. Add a constraint that forbids completing the fulfillment of the current set of priority shipments before the prior shipments are completely fulfilled
2. For all the shipments now considered, set D to the first day after the last shipment of the same product was fulfilled.

Considerations

- Even if we had a method capable of optimally solving the Short Term problem for some time the solution generate would not be considered acceptable by users
- This is due to factors such as the incomplete description of the operational constraints, the distance of the decision maker from the stake holder or historical reasons
- Nevertheless, this Short Term Planning problem is just a less studied combination of well known Optimization Problems: A Multiple Discretized Diet Problem over Space and Time with Time Windows
- The above presented approach is the result of an already 6 month old tunning process with a daily contact with the decision maker. This solver development is more than one year old
- No computational experience beyond decision maker analysis yet
- The time allowed to provide a plan should not exceed 30 min
- This is a global problem with an enormous economic impact! Also social, ecological...

Conclusion

- The MDDSTTWP have a number of characteristics we like on problems.
- It is difficult, it is not that studied (or well formalized)
- Why not work on it?

Conclusion

- The MDDSTTWP have a number of characteristics we like on problems.
- It is difficult, it is not that studied (or well formalized)
- Why not work on it?
- And it is green!

Thank You!