

ÍNDEX MEMÒRIA

Índex memòria.....	1
Resum.....	3
Resumen	3
Abstract.....	3
Agraïments	5
Capítol 1: Introducció.....	7
1.1. Problemàtica actual.....	7
1.2. Solució	8
1.2.1. Supply Chain	8
1.2.2. Closed Loop Supply Chain	9
Capítol 2: Objectius del projecte.....	11
Capítol 3: Selecció de l'eina	13
3.1. MATLAB.....	13
3.2. GAMS.....	14
3.3. OPL.....	15
3.4. AIMMS.....	16
Capítol 4: Descripció del procés.....	19
Capítol 5: Model.....	23
5.1. Model Base	23
5.2. Model desenvolupat	28
5.3. Resultats del model desenvolupat.....	33
Capítol 6: Validació del model.....	37
6.1. Factor de penalització del retorn de clients.....	37
6.2. Factor de penalització de la demanda	40
6.3. Límits de subministrament.....	43

6.4.	Demanda dels clients	46
6.5.	Factor mínim de reciclatge	49
Capítol 7: Conclusions i idees de futur.....		53
Capítol 8: Avaluació econòmica del projecte.....		55
Capítol 9: Bibliografia		59
9.1.	Referències bibliogràfiques.....	59
9.2.	Bibliografia de Consulta	60
Annex : Codi AIMMS		61

RESUM

La societat actual està molt conscienciada sobre aspectes mediambientals, però l'aspecte econòmic té un gran paper en el món industrial. Es pretén elaborar una eina de càlcul que permeti establir la planificació òptima d'una cadena de subministrament, tenint en compte objectius econòmics i , al mateix temps, respectant el medi ambient mitjançant un retorn de producte a la cadena.

RESUMEN

La sociedad actual está muy concienciada sobre aspectos medioambientales, pero el aspecto económico tiene un gran papel en el mundo industrial. Se pretende elaborar una herramienta de cálculo que permita establecer la planificación óptima de una cadena de suministro, teniendo en cuenta objetivos económicos y, al mismo tiempo, respetando el medio ambiente mediante un retorno de producto a la cadena.

ABSTRACT

Current society is very aware about environmental issues, but the economic aspect has a great paper in the industrial world. It is intended to develop a calculation tool that allows establishing an optimum supply chain planning, taking into account economic targets and, at the same time, respecting the environment through a product return to the chain.

AGRAÏMENTS

Es vol agrair al professor Moisès Graells el seu temps i interès dedicat a la realització d'aquest projecte. Els seus consells i suggeriments han servit per encaminar i accelerar el projecte quan ha estat necessari, i resoldre els dubtes que han aparegut al llarg de l'elaboració del projecte.

CAPÍTOL 1:

INTRODUCCIÓ

1.1. Problemàtica actual

A arrel de la revolució industrial amb la producció en cadena, l'aparició de l'automòbil, l'ús excessiu del petroli i el consumisme desmesurat, s'han generat unes condicions delicades pel medi ambient. Entre algunes de les conseqüències d'aquest fenomen es pot trobar el canvi climàtic, l'efecte hivernacle, l'esgotament de l'aigua dolça i la destrucció de la capa d'ozó.



Figura 1. Representació simbòlica dels efectes negatius de la indústria (Font:Ecured)

La creixent preocupació de la societat per aspectes relacionats amb el medi ambient ha estat el principal motor de la creació de nombroses mesures legislatives. Aquestes mesures tenen com a objectiu la minimització de l'impacte ambiental regulant l'abocament de residus, el reciclatge de matèria i la recuperació d'energia entre d'altres.

Fent referència al reciclatge de matèria, cada cop més, les empreses es preocupen per allargar el cicle de vida dels seus productes incorporant un reciclatge d'aquests. D'aquesta manera el que s'aconsegueix és reduir la quantitat de residus a llarg termini.



Figura 2. Representació de l'evolució del reciclatge d'envasos (Font:ECOEMBES)

A la figura anterior queda reflectida la importància que té la reducció de l'impacte ambiental al llarg del últims anys, on, cada vegada més, es decideix augmentar la quantitat de matèria reciclada mitjançant legislacions amb la incorporació de penalitzacions.

En definitiva, s'ha decidit fer una aposta clara de futur adreçada a la protecció i conservació del medi ambient.

Per una altra banda, l'aspecte econòmic també ha estat un factor clau a l'hora de prendre decisions en el món de la indústria, igual que en tots els sectors. A aquest àmbit se li associa la gestió i eficiència dels processos, és a dir, trobar la combinació que permeti realitzar un procés o conjunt de processos amb el menor cost possible. Es tracta d'un aspecte més complex que requereix d'un estudi a fons per a cada situació.

1.2. Solució

1.2.1. Supply Chain

El terme cadena de subministrament (de l'anglès *Supply Chain*) va ser introduït a finals del segle XX, per tal de solucionar els problemes de gestió i eficiència de processos. La cadena de subministrament està formada per totes aquelles parts involucrades de manera directa o indirecta en la satisfacció de la sol·licitud d'un client. No només està formada pel fabricant i el proveïdor, sinó que també s'inclouen els transportistes, magatzems i els mateixos clients. En altres paraules, és la seqüència d'esdeveniments que cobreixen el cicle de vida sencer d'un producte o servei des de que és produït fins que és consumit.

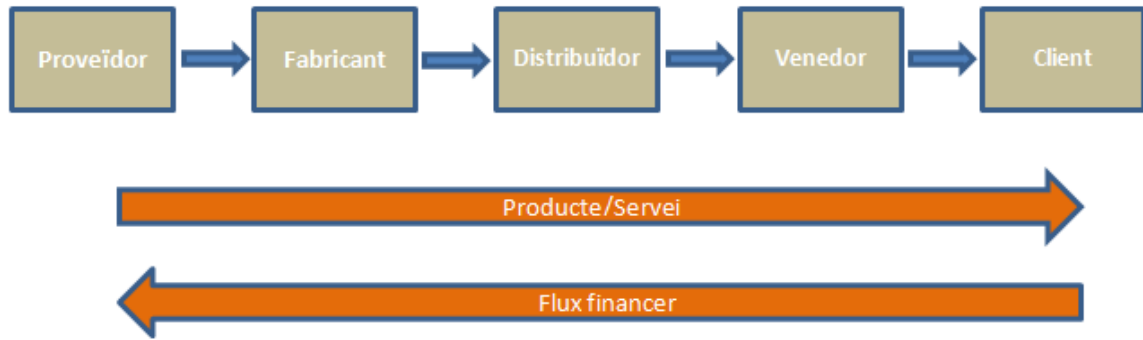


Figura 3a. Representació d'una cadena de subministrament general

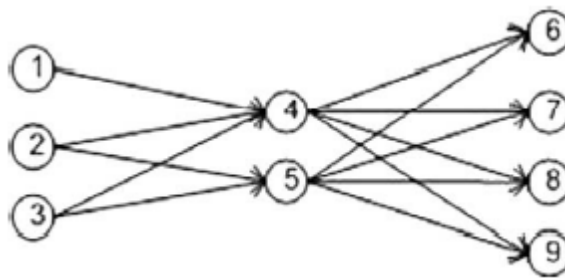


Figura 3b. Representació general de les connexions d'una cadena de subministrament
(Font: Article Gomes et al., 2009)

Des del punt de vista de cadena de subministrament, es tracta el pas del producte per una entitat com a un sol procés. Si en una entitat el producte ha de ser sotmès a diferents subprocessos, aquests s'hauran d'estudiar des d'un altre enfocament més específic, però no seran tinguts en compte des del punt de vista de cadena de subministrament.

1.2.2. Closed Loop Supply Chain

Es tracta d'una variant de cadena de subministrament, en la que el cicle de vida del producte no s'acaba quan arriba al client, sinó que després d'aquesta etapa, el producte és retornat a la cadena per la seva posterior reutilització. Al mateix temps, la quantitat de matèria primera requerida es veu disminuïda al llarg del temps, ja que en el seu lloc es disposa de producte reciclat.

Amb aquesta variant, s'aconsegueix allargar el cicle de vida del producte i, per tant, es tracta d'una cadena de subministrament més sostenible.

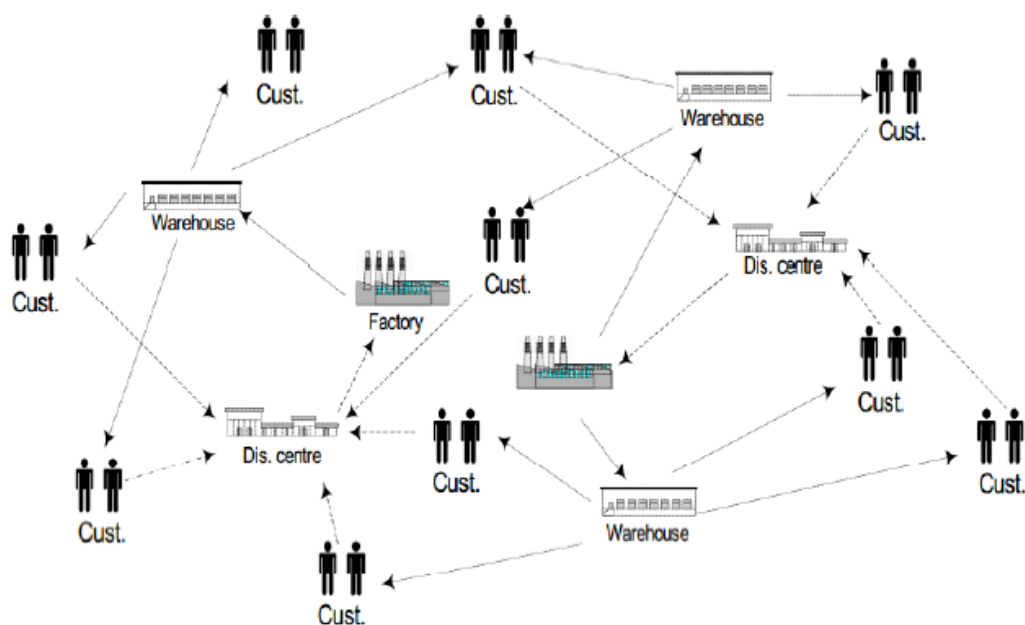


Figura 4. Representació d'una cadena tancada en forma de xarxa (Font: Gomes et al., 2009)

CAPÍTOL 2:

OBJECTIUS DEL

PROJECTE

S'han establert com a objectius d'aquest projecte els següents:

- Desenvolupament d'un model de cadena de subministrament (*Supply Chain*) amb *Closed-Loop* per a que contempli la possibilitat de tenir reciclatge i abocament de residus i, així, respectar aspectes mediambientals.
- Obtenció de la planificació diària de manera que es pugui dur a terme la posta en marxa de la cadena de subministrament.
- Elecció del software més adient per la implantació del model i representació dels resultats.
- Crear una interfície d'usuari per facilitar la visualització de la representació dels resultats.
- Elaboració d'una sèrie de casos a estudiar per validar el model implementat.

CAPÍTOL 3:

SELECCIÓ DE L'EINA

Actualment, existeix un ampli ventall de softwares com a eina per modelar, resoldre i representar problemes matemàtics com el que es té per objecte en aquest projecte. Cal recordar que es tracta d'un problema matemàtic de programació lineal (LP), i que aquest tipus de problemes no ofereixen molta complexitat a l'hora de ser resolts. Per tant, serà un problema que estarà a l'abast de ser resolt per diferents softwares. Pel que fa a la representació de la solució, no es pot dir el mateix. Així que l'aspecte representatiu a l'hora d'escollir el software tindrà un gran pes en la seva elecció.

A continuació, es troben detallats una sèrie de softwares i, posteriorment, l'elecció del software més adient per aquest projecte.

3.1. MATLAB

El Matlab (abreviatura de MATrix LABoratory, "Laboratori de Matrius") és una eina de software matemàtic que ofereix un entorn de desenvolupament integrat (IDE) amb llenguatge de programació propi.

Matlab és un programa de càlcul numèric dissenyat per treballar amb matrius. Per tant, serà més eficient si es dissenyen els algoritmes en termes de matrius i vectors. Es tracta d'una eina molt útil per treballar amb sistemes lineals d'equacions, anàlisi de Fourier, equacions diferencials ordinàries (EDOS), interpolacions i regressions i, sobretot, amb matrius i vectors.

A l'hora de fer representacions, destaca per graficar gran varietat de funcions en dos i tres dimensions.

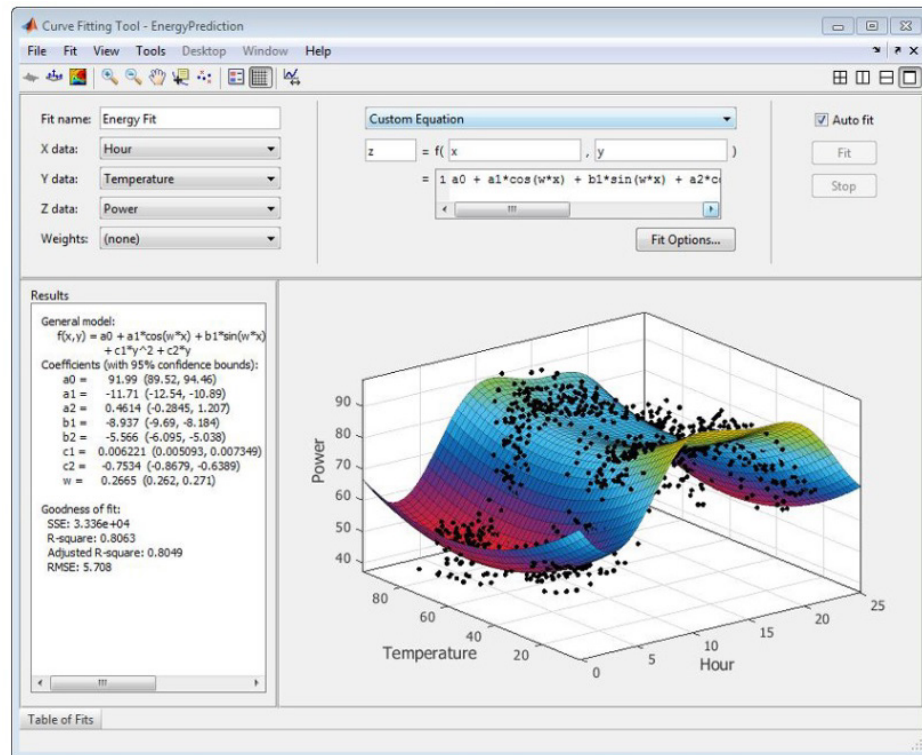


Figura 5. Representació en Matlab d'una funció en tres dimensions

Per concloure, s'ha decidit no utilitzar aquest software perquè s'ha d'introduir el model en forma de matrius i vectors.

3.2. GAMS

El General Algebraic Modeling System (GAMS) es un software d'alt nivell pel modelatge de sistema per la optimització matemàtica. Està dissenyat per modelar i resoldre problemes lineals, no lineals i optimització entera i mixta. GAMS va ser el primer llenguatge de modelat algebraic (AML) i és similar al que es fa servir comunament en els llenguatges de programació de quarta generació.

Es tracta però, d'un software que no presenta la suficient actualització en comparació a altres softwares més recents.

El seu punt fort és, sens dubte, la universalitat del llenguatge que fa servir, ja que també l'utilitzen molts altres softwares amb aquesta finalitat.

D'altra banda, els seus aspectes negatius són la manca d'actualització respecte altres softwares, les limitacions en quant a possibilitats a l'hora de fer representacions i les dificultats per importar dades des de softwares com Excel.

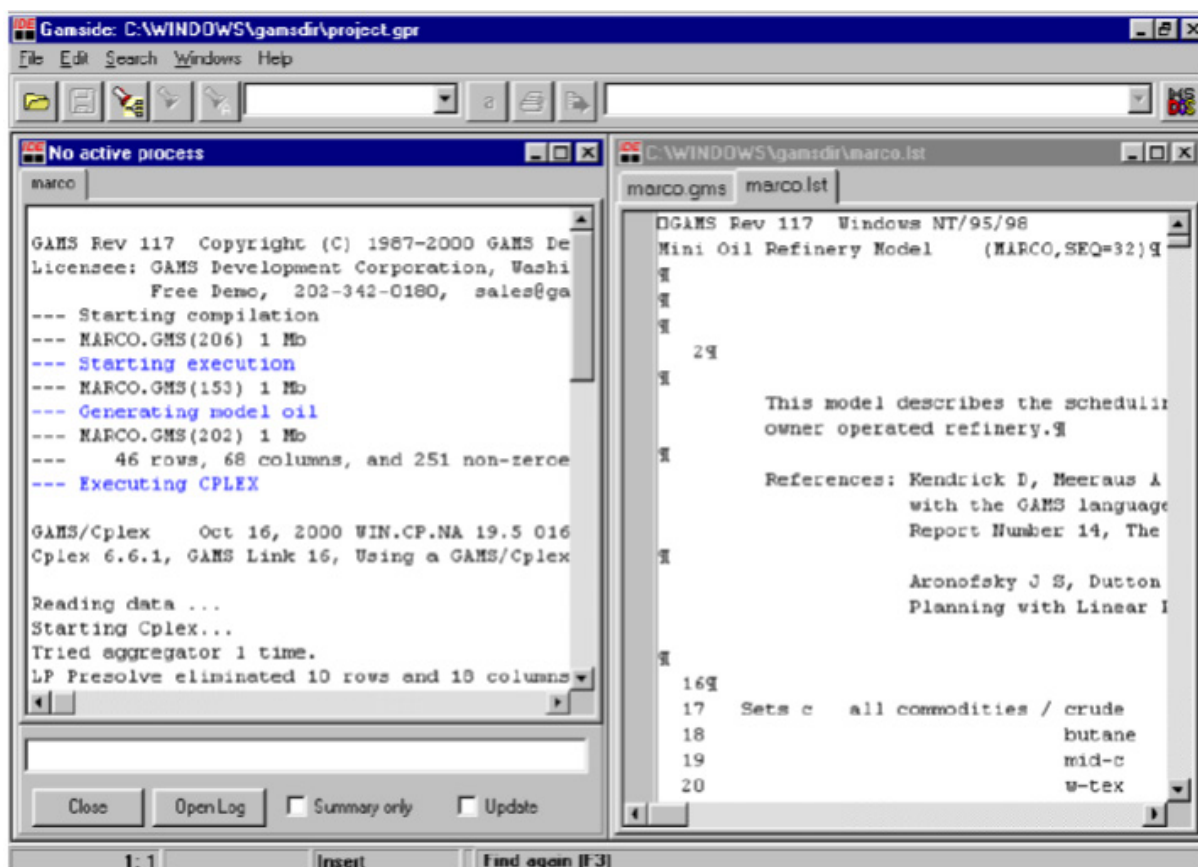


Figura 6. Interfície d'usuari de GAMS

Es decideix prescindir d'aquest software pels aspectes esmentats anteriorment.

3.3. OPL

Optimization Programming Language (OPL) es tracta d'un software que pertany a l'empresa IBM. Utilitza un únic optimitzador, el CPLEX, que permet resoldre problemes exclusivament lineals.

Utilitza un llenguatge exclusiu per l'optimitzador de l'empresa IBM, cosa que dificulta l'aprenentatge si mai s'ha fet servir i limita el model només a l'optimitzador d'aquesta empresa en concret. Permet una bona comunicació entre el software i programes com l'Excel o XML.

Pertany una interfície d'usuari actualitzada amb la que es pot treballar de forma dinàmica.

En quant a representacions gràfiques, no ofereix tant potencial com, per exemple, la interfície d'usuari.

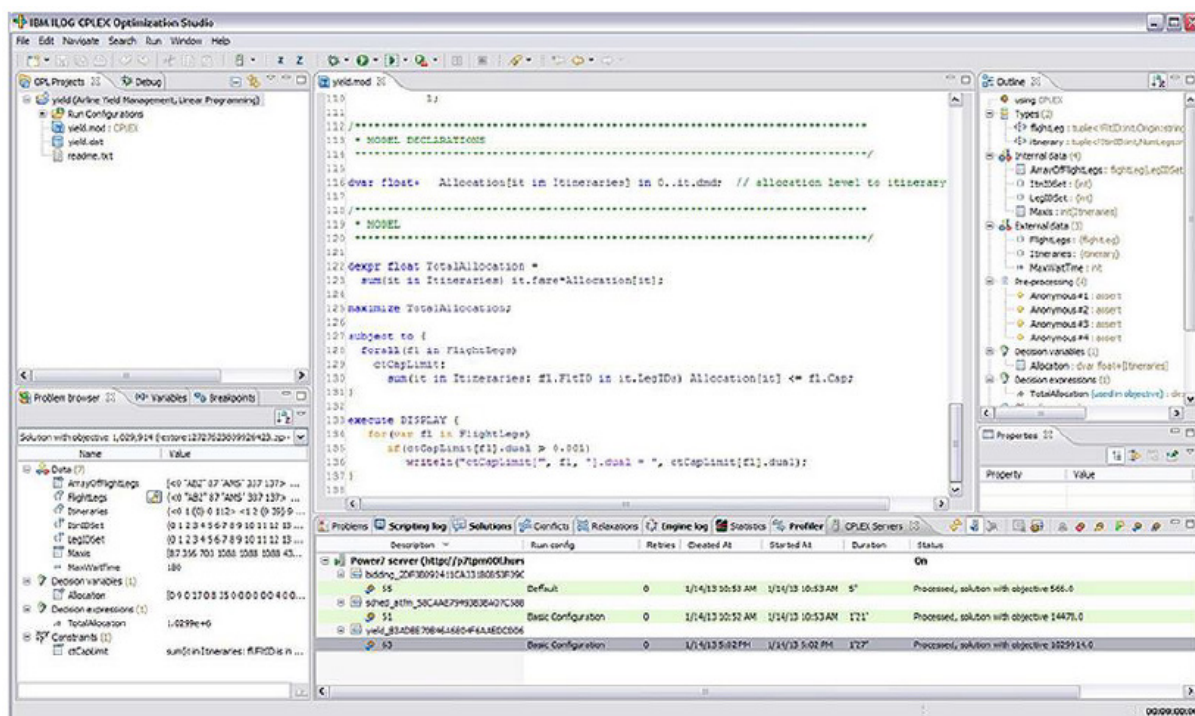


Figura 7. Interfície d'usuari de OPL

Es decideix descartar aquest software per l'exclusivitat del llenguatge i el baix potencial a l'hora de fer representacions.

3.4. AIMMS

Advanced Interactive Multidimensional Modeling Systemb (AIMMS) és un software dissenyat per la resolució d'optimitzacions de gran envergadura i problemes de planificació. Consisteix d'un llenguatge algebraic de modelització i un entorn de desenvolupament integrat (IDE) per editar models i crear interfícies gràfiques d'usuari en base a aquests models. Posseeix una gran varietat d'optimitzadors, com ara el CPLEX, Gurobi i MOSEK.

Compta amb una interfície d'usuari actualitzada que permet treballar de forma dinàmica i organitzada. No requereix el coneixement de cap llenguatge de programació per introduir el model.

Ofereix la possibilitat de representar diferents casos amb l'eina MultipleCases.

Té una bona comunicació amb programes com Excel i XML.

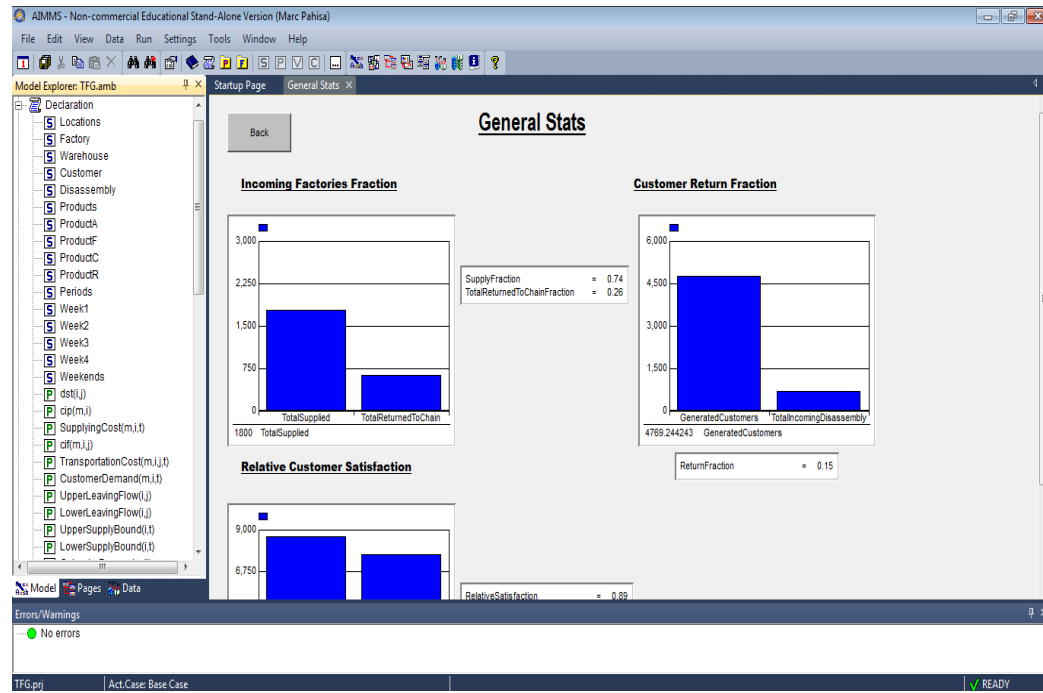


Figura 8. Implementació del projecte amb AIMMS

Per la senzillesa a l'hora d'utilitzar el software, potencial de representacions gràfiques i oferta de diferents optimitzadors, s'ha decidit escollir aquest programa per dur a terme la implementació del projecte.

CAPÍTOL 4:

DESCRIPCIÓ DEL PROCÉS

Tal i com descriu l'article utilitzat com a punt de partida (Gomes et al., 2009), es tracta d'una cadena de subministrament situada a Portugal on hi ha plantes de processat, magatzems, clients i centres de desmuntatge com a entitats. Té com a objecte la fabricació i distribució d'envasos de vidre.

L'envàs és produït a la planta de processat, després s'envia a centres d'emmagatzematge per ser empaquetat. A continuació es distribueixen els envasos als clients, a qui més tard se'ls recollirà una part dels envasos utilitzats per traslladar-los a centres de desmuntatge. Allà seran tractats per re introduir-los de nou a la planta de processat.

Cal remarcar que els clients d'aquest procés no són consumidors finals del producte, sinó que utilitzen l'envàs de vidre com a mitjà per dur a terme els seus propis productes (omplen els envasos de vidre amb líquids, per exemple).

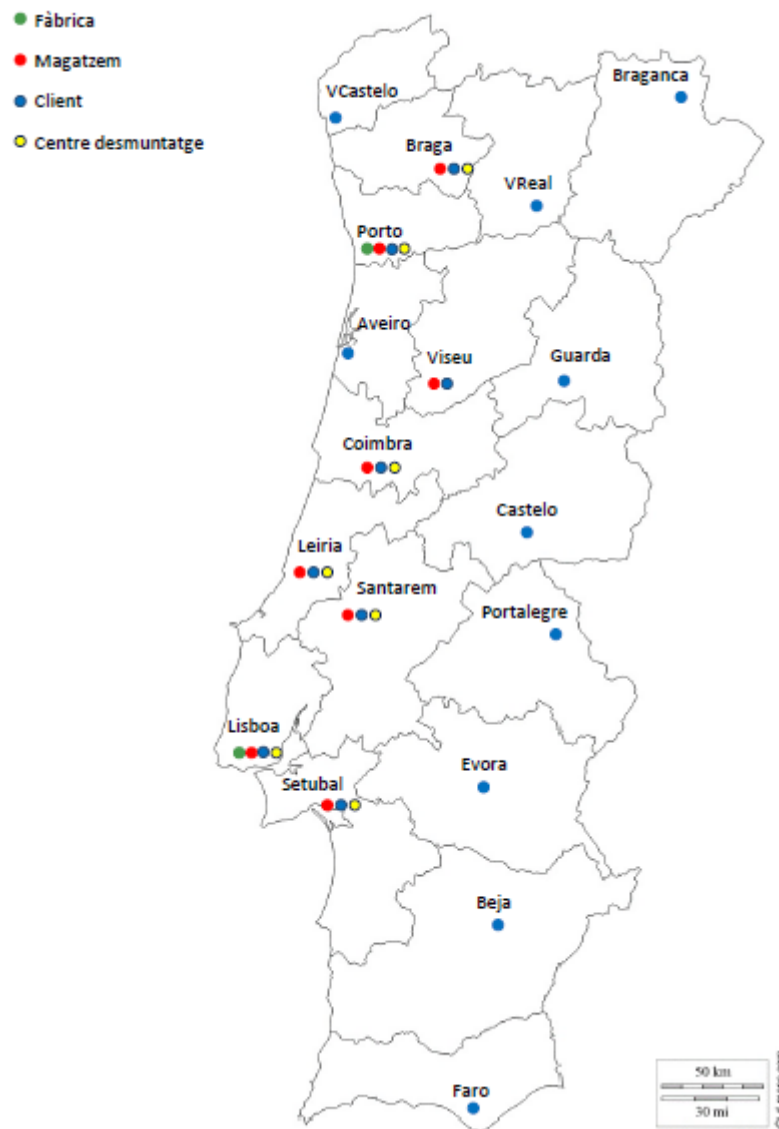


Figura 9. Representació de les entitats de la cadena de subministrament estudiada

La cadena inicia amb el subministrament a la planta de processat de dos productes (vidre blanc i vidre no blanc). A la planta de processat es produeixen tres tipus de famílies de producte, diferenciades pel color (blanc, verd i marró). Dins de cada família hi ha diferents subproductes en funció de la seva forma i capacitat.

A continuació es distribueixen els envasos a magatzems per ser empaquetats en sis tipus de lots diferents, segons color i capacitat.

Seguidament, es venen els lots d'envasos a clients, on conformaran el producte final de venda posterior al consumidor.

Fins aquest punt arribaria el cicle d'una cadena de subministrament general, però tenint en compte els aspectes mediambientals exposats a l'apartat introductori del projecte, obliga a implementar una recuperació de producte convertint la cadena en un sistema de producció tancat (*Closed Loop*). Així, la següent etapa és recollir parcialment els envasos utilitzats pels clients i traslladar-los al centre de desmuntatge, on es torna a fer la divisió entre vidre blanc i no blanc per re

introduir-los a la planta de processat juntament amb el subministrament inicial. També hi ha la possibilitat de descartar com a residus una part dels envasos retornats pels clients al centre de desmuntatge, sempre i quant es respectin els límits legals.

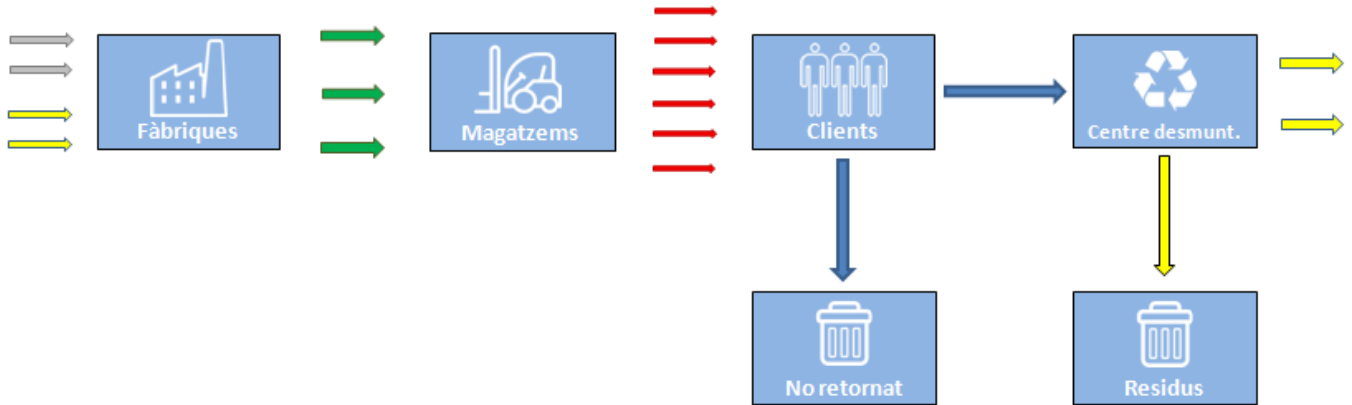


Figura 10. Esquema principal de la cadena de subministrament estudiada

Els balanços de productes estan regits per una taula de conversions (*Product bill of materials*). És a dir, no té perquè complir-se el balanç de matèria abans i després d'una entitat si, per exemple, una unitat d'A produeix 0,45 unitats de B.

Per aquest motiu, el balanç ha de contemplar els paràmetres de la relació de productes en les equacions.

CAPÍTOL 5:

MODEL

5.1. Model Base

S'agafa com a punt de partida el model de l'article Gomes et al., 2009. L'objectiu d'aquest model és obtenir el disseny i la planificació d'una cadena de subministrament amb *closed-loop* amb un horitzó de 5 anys. La funció objectiu són els costos i es pretén optimitzar la cadena minimitzant-los.

El model de l'article base (Gomes et al., 2009) és el següent:

$$\begin{aligned} \text{Min } F = & \sum_{i \in I} f_i Y_i + \sum_{i \in I_c} c_i (1 - Y_i) + \sum_{mij:(m,i,j) \in F} \sum_{t' \in T_m} c_{mijt'} X_{mijt'} + \sum_{mi:(m,i) \in N_c} \sum_{t \in T} c_{mit}^u U_{mit} \\ & + \sum_{mi:(m,i) \in N \setminus N_c} \sum_{t' \in T'} c_{mit'}^s S_{mit'} \end{aligned} \quad (1)$$

s.t.

$$S_{mij(t,t'-1)} + \sum_{m'j:(m',j,i) \in F} \beta_{mm'} X_{m'ji\gamma(t,t'-\tau_{ji}-\phi_m)} = \sum_{m'j:(m',i,j) \in F \setminus F_s} \beta_{mm'} X_{m'itjt'} + S_{mit'}, \quad (m,i) \in V' \wedge (t,t') \in T', \quad (2)$$

$$\sum_{j:(m,j,i) \in F} \sum_{t' \in T'} X_{mji\gamma(t,t'-\tau_{ji})} + U_{mit} = d_{mit} Y_i, \quad (m,i) \in V_c^{out} \wedge t \in T, \quad (3)$$

$$\sum_{j:(m,i,j) \in F} \sum_{t' \in T'} X_{mijt'} + U_{mit} = \sum_{m'j:(m',j,i) \in F} \sum_{t' \in T'} \beta_{mm'} X_{m'ji\gamma(t,t'-\tau_{ji}-\phi_m)}, \quad (m,i) \in V_c^{in} \wedge t \in T, \quad (4)$$

$$\sum_{j \in I_0} \sum_{t' \in T'} X_{mijt'} \leq \sum_{m:(m,i,j) \in F_s} X_{mijt'} \leq g_j^p Y_j, \quad (i,j) \in A_s \wedge (t,t') \in T', \quad (6)$$

$$\sum_{m:(m,i,j) \in F_s} X_{mijt'} \geq h_j^p Y_j, \quad (i,j) \in A_s \wedge (t,t') \in T', \quad (7)$$

$$\sum_{m:(m,i) \in N \setminus N_c} S_{mit'} \leq g_i^s Y_i, \quad i \in I \wedge (t, t') \in T', \quad (8)$$

$$\sum_{m:(m,i,j) \in F} X_{mijt'} \leq g_{ij} E_{ijt'}, \quad (i, j) \in A \wedge (t, t') \in T', \quad (9)$$

$$\sum_{m:(m,i,j) \in F} X_{mijt'} \geq h_{ij} E_{ijt'}, \quad (i, j) \in A_{f_1} \cup A_{f_2} \wedge (t, t') \in T', \quad (10)$$

$$2E_{ijt'} \leq Y_i + Y_j, \quad (i, j) \in A \wedge (t, t') \in T', \quad (11)$$

$$\sum_{j:(m,i,j) \in F} X_{mijt'} \geq h_{mi}^c Y_i, \quad (m, i) \in V_c^{in} \cup V_c^{out} \wedge ((t, t') \in T': t' \geq \xi_{mi} \vee t \geq t_2), \quad (12)$$

$$X_{mijt'}, S_{mit'}, U_{mit'} \in \mathfrak{R}_0^+, \quad Y_i, E_{ijt'} \in \{0,1\}. \quad (13)$$

L'equació (1) és la funció objectiu a minimitzar i és una suma de costos associats a obrir entitats, penalització per deixar fora de la cadena de subministrament a un client, transport (que inclou subministrament de matèria prima i abocament de residus), penalització per no satisfer la demanda dels clients i per no retornar a la cadena la matèria utilitzada pels clients, i emmagatzematge (excepte als clients). L'equació (2) és la restricció corresponent als balanços de matèria. L'equació (3) és la restricció de la demanda. L'equació (4) és la restricció del retorn de matèria del client a la cadena. L'equació (5) és la restricció que assegura que els centres de desmuntatge no poden abocar més fracció de residu del permès per la legislació europea. Les equacions (6) i (7) són les restriccions referents als límits màxim i mínim de flux de matèria prima subministrat a les plantes de processat. L'equació (8) és la restricció que determina el límit d'emmagatzematge a cada entitat. Les equacions (9), (10) i (12) són les restriccions que determinen els límits del flux de matèria entre les diferents entitats. L'equació (11) és la restricció que relaciona el flux existent entre dues entitats que estan operatives. L'equació (13) defineix el domini de les variables de decisió.

- On els paràmetres són els següents:

τ_{ji} : temps de viatge entre entitats i i j ,

ϕ_m : temps d'utilització/processat del producte m ,

$\xi_{mi} = f(\tau_{ji}, \phi_m)$ funció dels temps de viatge i processat, assignant a la primera unitat de microtemps un corrent de producte $m \in M$, amb origen a l'entitat $i \in I$, quan pot succeir.

α_m : retorn mínim de producte m segons la legislació vigent, $\alpha_m \in [0,1]$,

$\beta_{mm'}$: relació entre el producte m i m' ,

ρ_m : fracció de retorn del producte $m \in M_a$,

s_{mi0} : stock inicial de producte m a l'entitat $i \in I$,

f_i : cost d'inversió de l'entitat $i \in I$,

c_i : cost de deixar fora de la cadena de subministrament al client i ,

g_i^s : capacitat màxima d'emmagatzematge de l'entitat $i \in I$,

g_i^p i h_i^p : màxim i mínim límit de subministrament de matèria prima de l'entitat $i \in I_f$,

g_i : valor del límit superior dels fluxos sortints de l'entitat i ,

h_i^f : valor del límit inferior dels fluxos sortints de l'entitat $i \in I_f$,

h_{mi}^c : valor del límit inferior pel producte m dels fluxos sortints de l'entitat $i \in I_c$,

d_{mit} : demanda del producte m per l'entitat $i \in I_c$ en el macroperíode t ,

c_{mit}^u : cost unitari variable de la demanda no satisfeta/retorn no satisfet de producte m a l'entitat $i \in I_c$, pel macroperíode t ,

$c_{mijt'}$: cost unitari de transport del producte m des de l'entitat i cap a l'entitat j , en el temps t' ,

c_{mit}^s : cost unitari d'emmagatzematge a l'entitat i , en el temps t' .

- On les variables són les següents:

$x_{mijt'}$: quantitat de producte m transportat des de l'entitat i cap a l'entitat j , en el temps t' ,

s_{mit} : quantitat de producte m emmagatzemat a l'entitat i , durant el període t' ,

u_{mit} : quantitat no satisfeta de producte m a l'entitat $i \in I_c$, durant el macroperíode t ,

$Y_i = 1$ si l'entitat i es troba oberta/servida; 0 en cas contrari,

$E_{ijt'}$: variable auxiliar que permet el modelatge de límits mínims imposats en fluxos del model; $E_{ijt'} = 1$ si els fluxos entre entitat i i entitat j succeeixen en el temps t' .

- On les referències al temps són les següents:

t : índex pel macrotemps,

t' : índex pel microtemps,

$T = \{t_1, t_2, \dots, t_h\}$ i $T_p = \{t'_1, t'_2, \dots, t'_n\}$

$T' = \{(t, t') : t \in T \wedge t' \in T_p\}$.

- On les entitats són les següents:

I_f : fàbriques $i \in I_f \in V$,

I_a : magatzems $i \in I_a \in V$,

I_c : clients $i \in I_c \in V$,

I_r : centres de desmuntatge $i \in I_r \in V$,

I_o : opció d'eliminació $i \in I_o \in V$,

$I = I_f \cup I_a \cup I_r$.

- On els tipus de producte són els següents:

M_f : producte sortint de les fàbriques $m \in M_f \in M$,

M_a : producte sortint dels magatzems $m \in M_a \in M$,

M_c : producte sortint dels clients $m \in M_c \in M$,

M_r : producte sortint dels centres de desmuntatge $m \in M_r \in M$.

- On les relacions entre els productes i les entitats són les següents:

$V_f^{out} = \{(m,i) : m \in M_f \wedge i \in I_f\}$,

$V_a^{out} = \{(m,i) : m \in M_a \wedge i \in I_a\}$,

$V_c^{out} = \{(m,i) : m \in M_c \wedge i \in I_c\}$,

$V_r^{out} = \{(m,i) : m \in M_r \wedge i \in I_r\}$,

$V_c^{in} = \{(m,i) : m \in M_a \wedge i \in I_c\}$,

$V_f^{in} = \{(m,i) : m \in M_r \wedge i \in I_f\}$,

$V' = V_f^{in} \cup V_r^{out} \cup V_c^{out} \cup V_a^{out}$.

- On els fluxos són els següents:

$A_{f1} = \{(i,j) : i \in I_f \wedge j \in I_a\}$,

$A_{f2} = \{(i,j) : i \in I_a \wedge j \in I_c\}$,

$A_{f3} = \{(i,j) : i \in I_c \wedge j \in I_r\}$,

$A_{f4} = \{(i,j) : i \in I_r \wedge j \in I_f\}$,

$A_s = \{(i,i) : i \in I_f\}$,

$A_d = \{(i,j) : i \in I_r \wedge j \in I_o\}$,

$A = \bigcup_{k \in K} A_k, K = \{f_1, f_2, r_1, r_2\}$.

- On les relacions entre els fluxos i els tipus de producte són les següents:

$$F_{f1} = \{(m, i, j) : m \in M_f \wedge (i, j) \in A_{f1}\},$$

$$F_{f2} = \{(m, i, j) : m \in M_a \wedge (i, j) \in A_{f2}\},$$

$$F_{r1} = \{(m, i, j) : m \in M_r \wedge (i, j) \in A_{r1}\},$$

$$F_{r2} = \{(m, i, j) : m \in M_c \wedge (i, j) \in A_{r2}\},$$

$$F_s = \{(m, i, i) : m \in M_c \wedge (i, i) \in A_s\},$$

$$F_d = \{(m, i, j) : m \in M_c \wedge (i, j) \in A_d\},$$

$$F = \cup_{k \in K} F_k, K = \{f_1, f_2, r_1, r_2, s, d\}.$$

5.2. Model desenvolupat

Després d'estudiar detingudament el model base, s'ha arribat a la conclusió de que presenta una limitació clara: el model és insuficient per la posta en marxa de la cadena de subministrament. Mentre que el disseny és obtingut satisfactòriament, la planificació establerta resulta massa general. Es tracta d'una planificació orientativa, ja que les dades d'aquesta són donades al llarg de períodes bimestrals o anuals. D'aquesta manera, no es disposa de la informació necessària per generar una planificació que pugui ser executada per les entitats que formen la cadena.

El model desenvolupat presenta una sèrie de variacions respecte l'original:

- Adaptació de l'horitzó a 1 mes (28 dies) enlloc dels 5 anys de l'article base. S'ha pres aquesta decisió per centrar el projecte en l'optimització de la planificació i operació de la cadena estudiada. A l'article base es pretenia prendre decisions sobre el disseny de la cadena i, per això, era necessari tenir un horitzó relativament a llarg termini per veure les conseqüències del disseny escollit.
Aquest nou projecte té com a punt de partida el disseny escollit en el projecte anterior, i la seva funció és optimitzar la planificació del dia a dia de cada entitat pel període d'un mes. El disseny consta de dos plantes de processat, vuit magatzems, divuit clients i vuit centres de desmuntatge. Per tal de fer aquesta adaptació, són requerides algunes modificacions en el model:
 - Eliminació de les variables binàries que indiquen si una entitat està oberta o tancada. El disseny serà inflexible, ja que ha estat determinat anteriorment. Totes les variables són contínues, per tant, passa a ser un problema LP (*linear programming*).
 - Canvi dels sets referents a macro períodes i micro períodes. Els macro períodes passen a ser setmanes enlloc d'anys, i els micro períodes són dies enlloc de bimestres. També es veuen afectades les dades i paràmetres, ja que han de ser redimensionades a quantitats diàries.
 - Ajust dels índexs d'algunes restriccions, com ara el compliment de la demanda, que anteriorment s'havien de complir al llarg de cada any. Actualment han de ser satisfetes a diari.
- Introducció d'un calendari amb la simulació de la planificació de treball en caps de setmana i dies festius. També és útil pels dies de manteniment en els que s'interromp la producció. Per implementar-lo es declara un vector com a paràmetre format pels micro períodes (28) amb valors binaris (1 pels dies laborals i 0 pels dies no laborals) i es multiplica a les restriccions relacionades amb els límits superiors de flux entre entitats i de subministrament. D'aquesta manera, l'única operació possible en els micro períodes amb valor 0 és tenir stock.
- Aproximació del temps de transport entre entitats a zero.

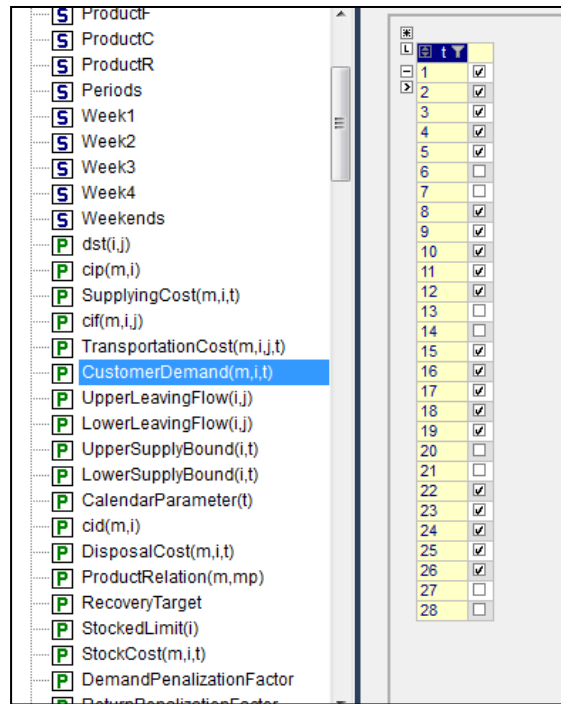


Figura 11. Paràmetre Calendari implementat en AIMMS

A continuació, s'adjunten els plans de producció d'una planta de processat per diferents mesos. S'ha escollit el mes de febrer, mes en el que no hi ha cap dia festiu, i el mes de desembre, on és festa els dies 6, 8 i 25.

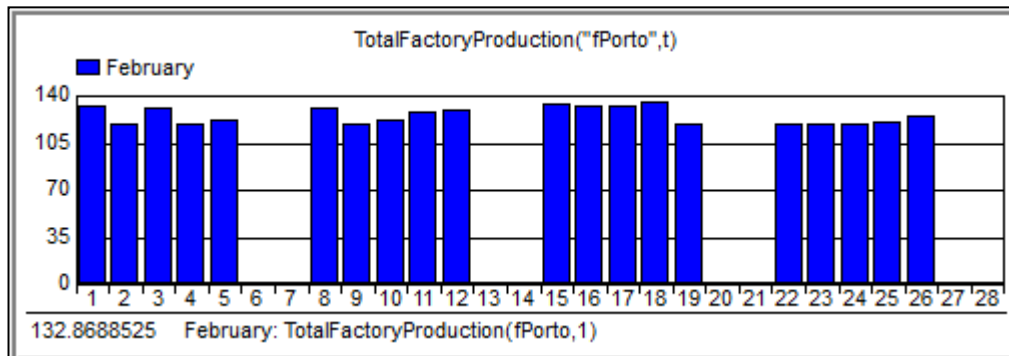


Figura 12. Pla de producció de la planta Porto pel mes de febrer

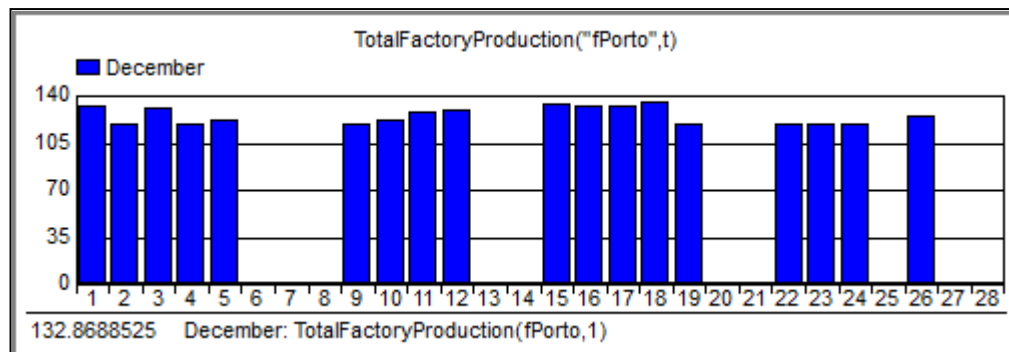


Figura 13. Pla de producció de la planta Porto pel mes de desembre

El software AIMMS permet organitzar de forma ordenada totes les dades que es tenen en compte, classificant-les en sets, paràmetres, variables i restriccions.

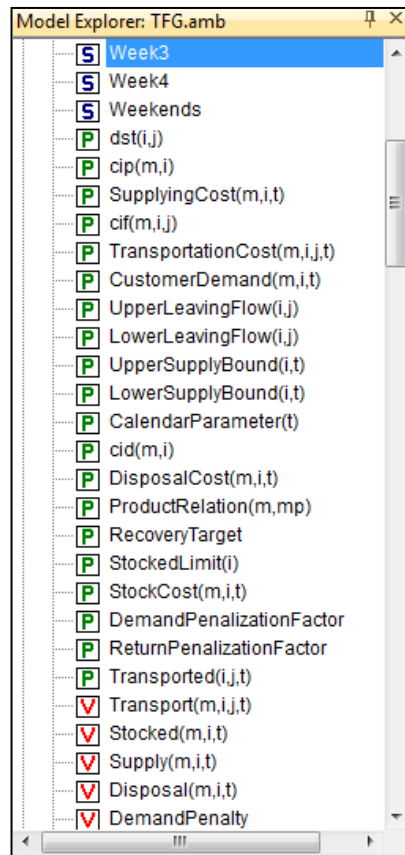


Figura 14. Classificació de les dades en AIMMS segons el seu tipus

Per una altra banda, també ofereix la possibilitat de representar diferents escenaris per comparar els resultats mitjançant l'eina "Multiple Cases".

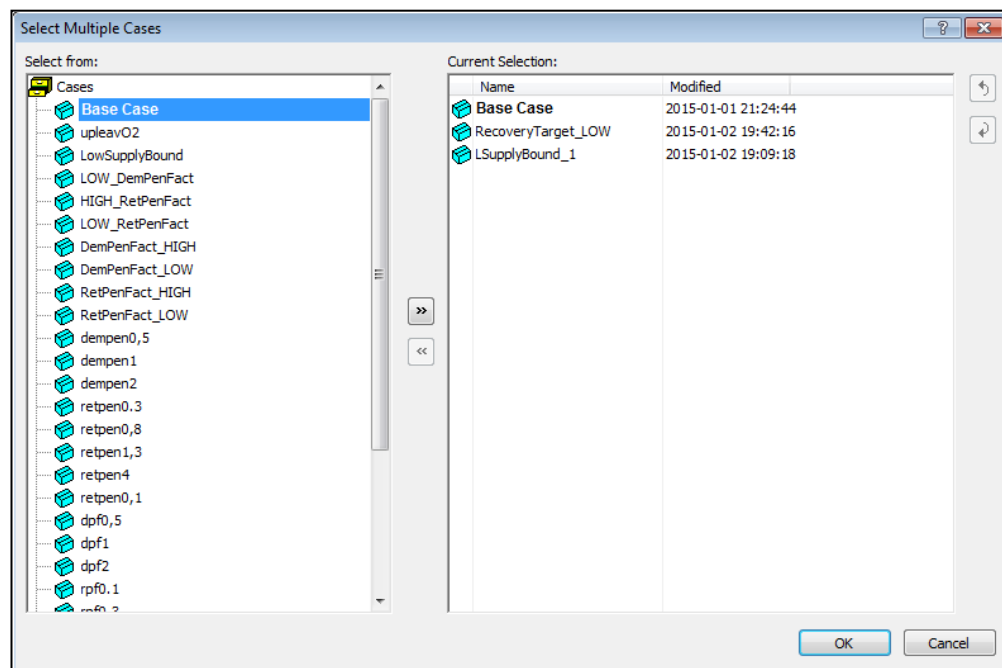


Figura 15. Eina "Multiple Cases" per representar diferents escenaris

Per últim, s'ha incorporat una interfície d'usuari al iniciar el programa per tal de fer més fàcil la navegació pels resultats del projecte.

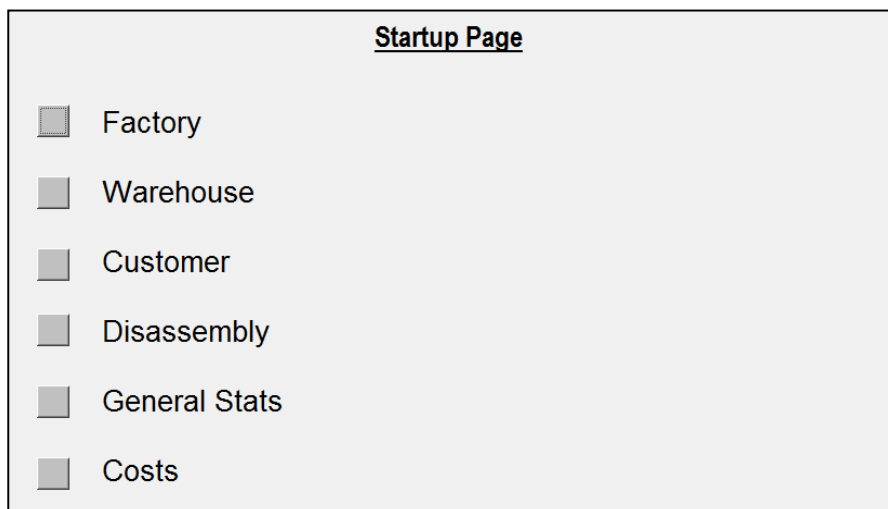


Figura 16. Interfície d'usuari al obrir el programa

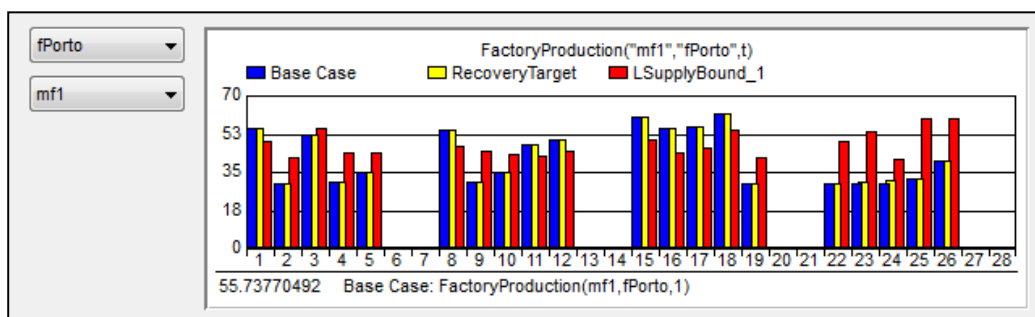


Figura 17. Interfície d'usuari que permet navegar pels resultats

Ha estat necessari crear una sèrie de paràmetres addicionals per tal de representar les dades que s'ha cregut convenient amb sumatoris i percentatges. D'aquesta manera es poden veure "dades resum" i dades totals al llarg dels dies.

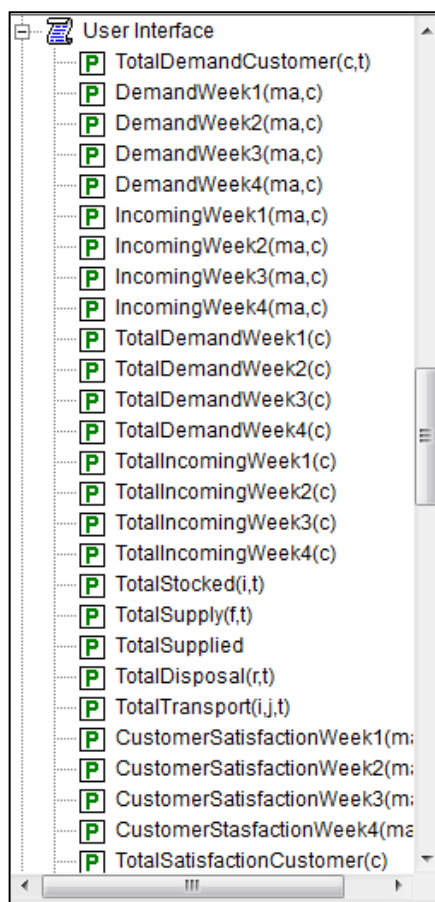


Figura 18. Llistat d'alguns paràmetres addicionals per la representació dels resultats

A l'hora de fer els balanços de les entitats, s'han de tenir en compte uns coeficients que fan referència a la relació entre productes (*Bill of materials*):

mp	mf1	mf2	mf3	ma1	ma2	ma3	ma4	ma5	ma6	mc1	mc2	mr1
m												
mf1	1			0.61		0.39						
mf2		1			0.5		0.5					
mf3			1					0.61	0.39			
ma1				1								0.45
ma2					1							0.7
ma3						1						0.5
ma4							1					0.8
ma5								1				0.4
ma6									1			0.9
mc1	2	1	1							1		
mc2		1	2								1	
mr1										1.333	1	1

Taula 1. Relació de productes

On mf1, mf2 i mf3 són els productes processats a les plantes, ma1, ma2, ma3, ma4, ma5 i ma6 són els productes empaquetats als magatzems, mc1 i mc2 són els productes de subministrament a les plantes i aquells que han estat tractats

als centres de desmuntatge, i mr1 és el producte retornat pels clients als centres de desmuntatge.

5.3. Resultats del model desenvolupat

En aquest apartat es mostra la planificació a la que s'ha arribat amb el model desenvolupat. Es tracta de la planificació diària de les entitats per a un mes sense dies festius ni de manteniment (febrer, per exemple). Es representen dades referents a fluxos totals diaris de cada entitat, és a dir, la xifra que apareix és el sumatori dels diferents productes que passen per cada entitat. S'ha decidit fer-ho d'aquesta manera per evitar posar un nombre excessiu de gràfics. Si es desitja veure la planificació amb cada producte distingit, és possible des del software AIMMS i el fitxer que s'adjunta amb el CD.

Es recorda que la cadena està formada per 2 plantes de processat (fLisboa i fPorto), 8 magatzems (aBraga, aCoimbra, aLeiria, aLisboa, aPorto, aSantarem, aSetubal i aViseu), 18 clients i 8 centres de desmuntatge (rBraga, rCoimbra, rLeiria, rLisboa, rPorto, rSantarem, rSetubal i rViseu).

A continuació es representa la planificació obtinguda per al cas base. Tan sols es mostra una entitat de cada tipus com a exemple. La resta d'entitats es troben representades a l'arxiu d'AIMMS del CD adjuntat.

- Planta de processat



Figura 19. Planificació de la planta Lisboa

Està representada la quantitat de matèria prima subministrada (*TotalSupply*), la quantitat provinent del reciclatge (*TotalRecycleIncoming*), la producció de la planta (*TotalFactoryProduction*), la quantitat sortint de la planta (*TotalLeavingFactory*) i la quantitat estocada (*TotalStocked*), que en aquest cas és inexistent.

- Magatzem

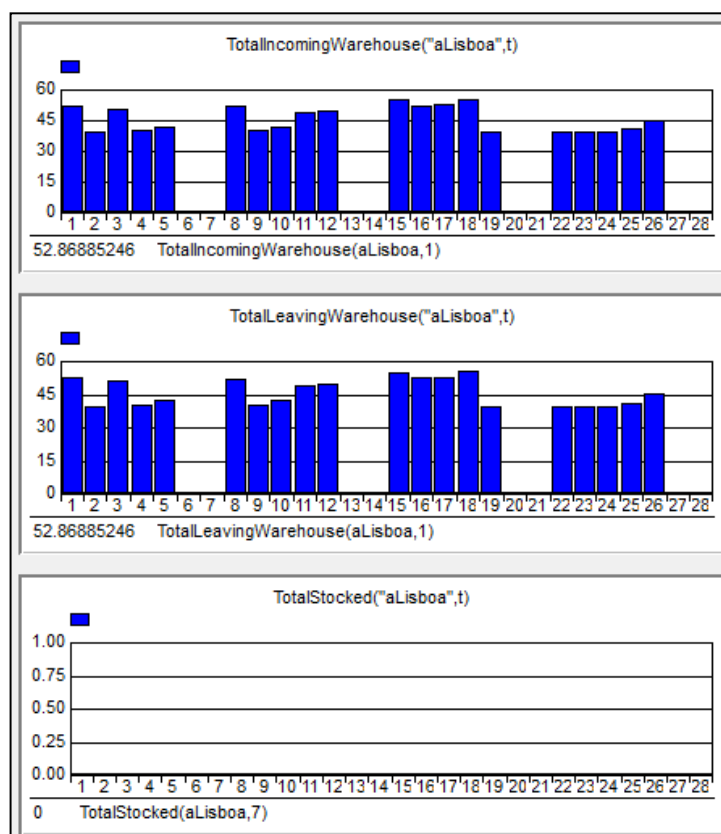


Figura 20. Planificació del magatzem Lisboa

En la planificació del magatzem es representa la quantitat entrant a la entitat (*TotalIncomingWarehouse*), la quantitat sortint del magatzem (*TotalLeavingWarehouse*) i la quantitat estocada (*TotalStocked*).

- Client

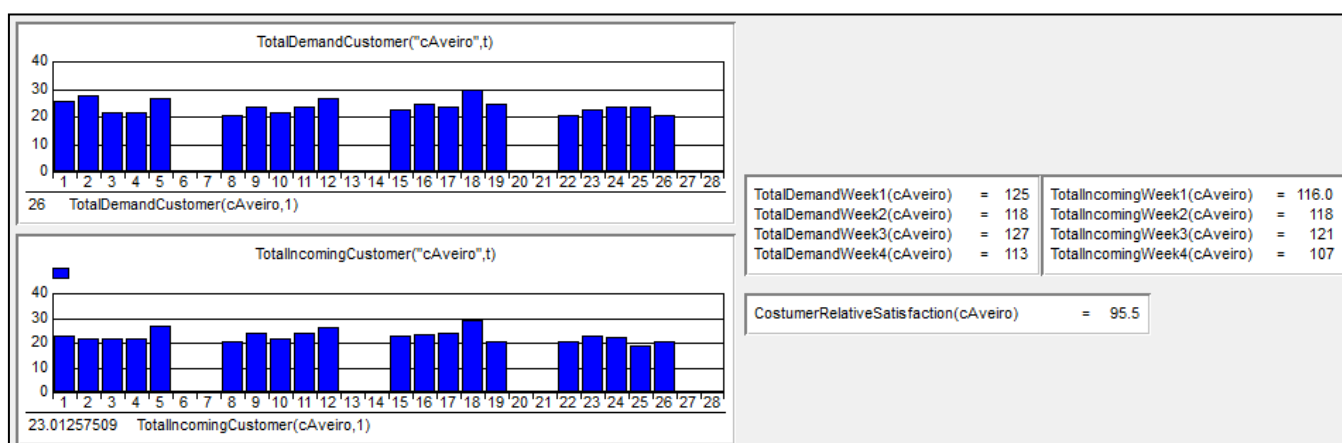


Figura 21. Planificació del client Aveiro

Es representa la demanda del client (*TotalDemandCustomer*) i la quantitat que se li ha enviat (*TotalIncomingCustomer*). S'hi acompanya una sèrie de paràmetres al costat que representen les dades setmanals i la fracció de demanda satisfeta del client.

- Centre de desmuntatge

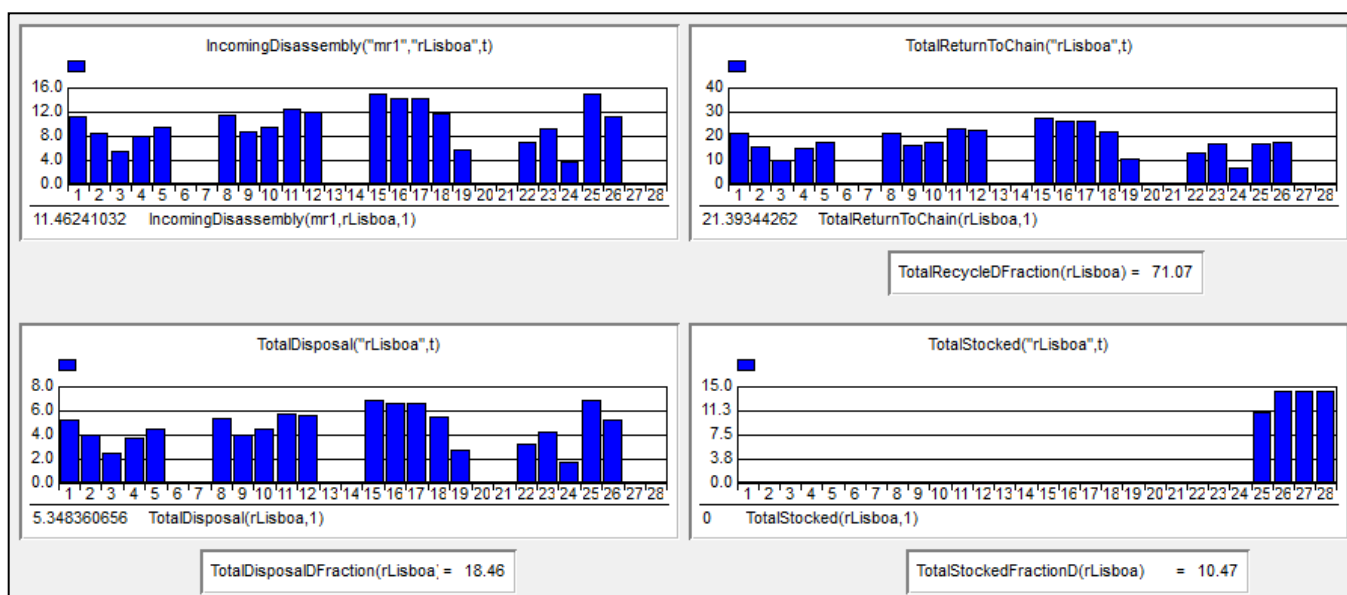


Figura 22. Planificació del centre de desmuntatge Lisboa

Es representa la quantitat entrant al centre (*IncomingDisassembly*), la quantitat que retorna a les plantes de processat (*TotalReturnToChain*), la quantitat que s'aboca com a residus (*TotalDisposal*) i la quantitat estocada al centre (*TotalStocked*). A més, s'incorporen uns paràmetres indicant la fracció que representa cada quantitat.

Per últim, es mostra el cost total i de forma desglossada de la planificació:

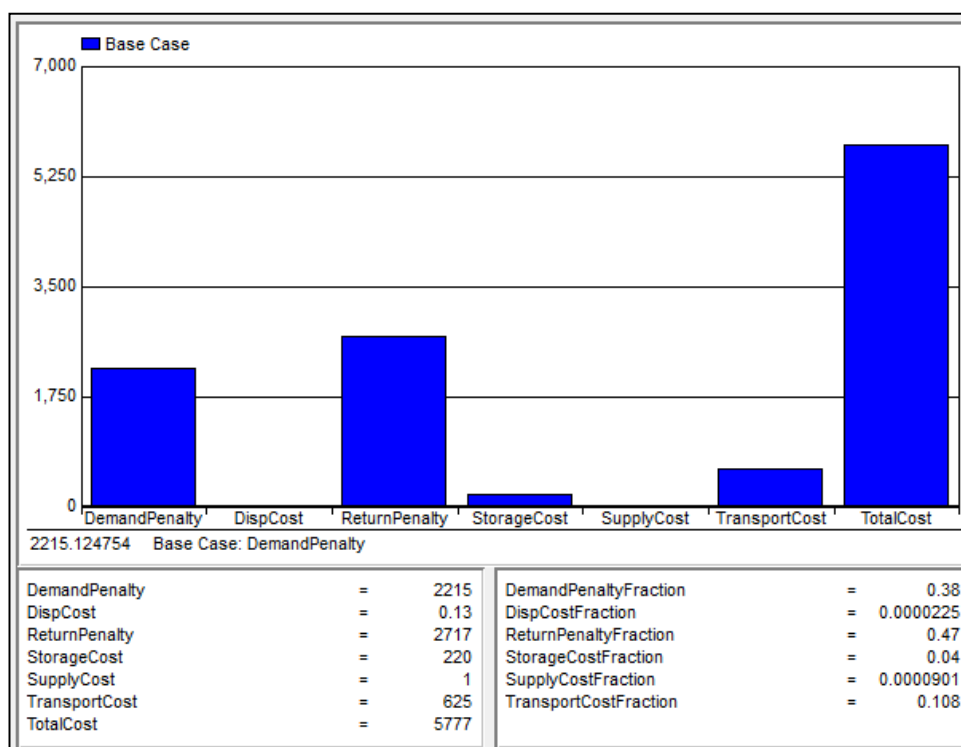


Figura 23. Costos del cas base

Els costos d'abocament de residus (*Disposal*) i de subministrament (*Supply*) són tant baixos degut al preu unitari tant petit que tenen (entre 0,0003 i 0,0005). A més, com que hi ha reciclatge de producte, justament aquestes dues quantitats haurien de ser més petites que la resta. Es recorda que només es pot abocar com a residus una fracció del producte que passa pels centres de desmuntatge establerta per la legislació, i que el subministrament de matèria prima va acompanyat del reciclatge de producte, cosa que fa que es requereixi un menor subministrament del proveïdor.

CAPÍTOL 6:

VALIDACIÓ DEL MODEL

Un cop implementat el model amb el software AIMMS, es procedeix a la validació d'aquest, per la qual cosa es fa l'estudi de diferents casos en els que es varia el valor d'alguns paràmetres per veure com respon el model. Els paràmetres variats no faran referència al disseny de la cadena de subministrament, ja que es decideix que sigui inflexible. Per tant, seran paràmetres relacionats amb la política d'empresa, operació i legislació.

Per a cada cas es representen els resultats obtinguts al software AIMMS i, addicionalment, amb un diagrama de Sankey. El diagrama de Sankey es tracta d'un diagrama de flux en el que l'amplada de les fletxes entre entitats és proporcional a la quantitat de flux existent. Estan representades les quantitats totals mensuals de producte.

Els casos estudiats es compararan amb el cas base.

Els paràmetres que s'ha decidit variar són:

1. Factor de penalització del retorn de clients
2. Factor de penalització de la demanda
3. Límits de subministrament
4. Demanda dels clients
5. Factor mínim de reciclatge

6.1. Factor de penalització del retorn de clients

Els clients es veuen obligats a retornar el producte fet servir. L'empresa d'envasos decideix la quantitat de producte fet servir que recullen els centres de desmuntatge. La part de producte que no ha estat recollida té un cost com a penalització.

Es disposa de tres casos: BaseCase (RetPenFact=0,8), RPF_HIGH (RetPenFact=1,3) i RPF_LOW (RetPenFact=0,3).

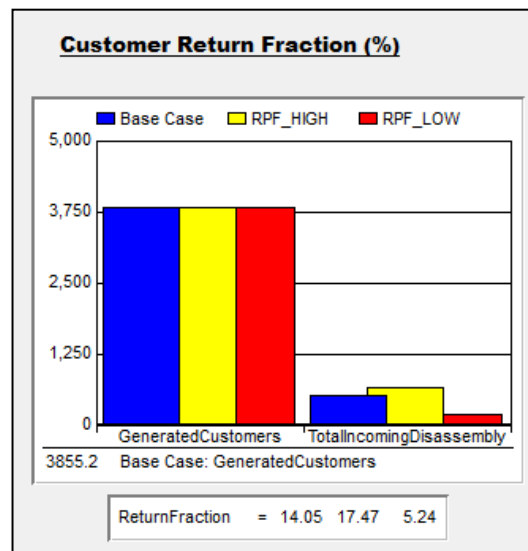


Figura 24. Representació de la fracció de retorn per a cada cas

La figura 24 indica que els clients segueixen rebent la mateixa quantitat de producte al llarg del mes, però la quantitat que retorna als centres de desmuntatge es veu afectada per la variació del paràmetre. Quant més gran és el valor del factor de penalització, més gran és el flux de retorn per evitar augmentar el cost total de penalització.

ReturnPenalizationFactor=1,3

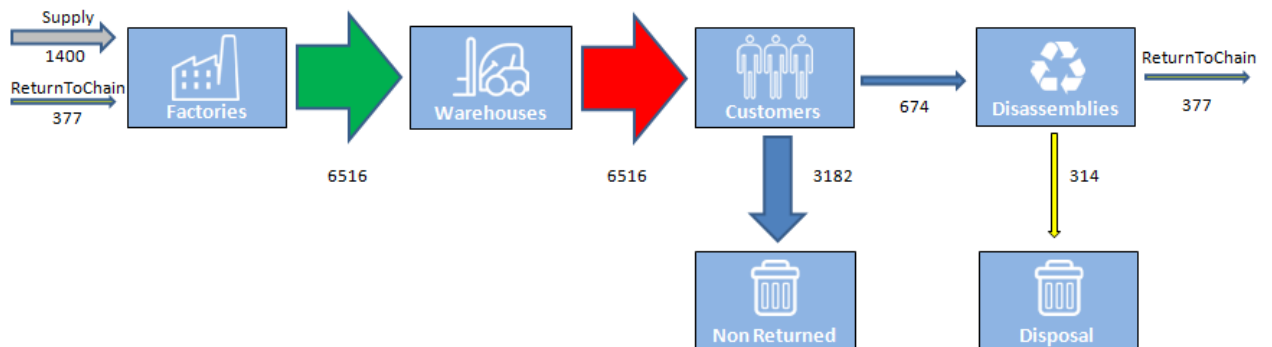


Figura 25a. Diagrama de Sankey del cas RPF_HIGH

ReturnPenalizationFactor=0,3

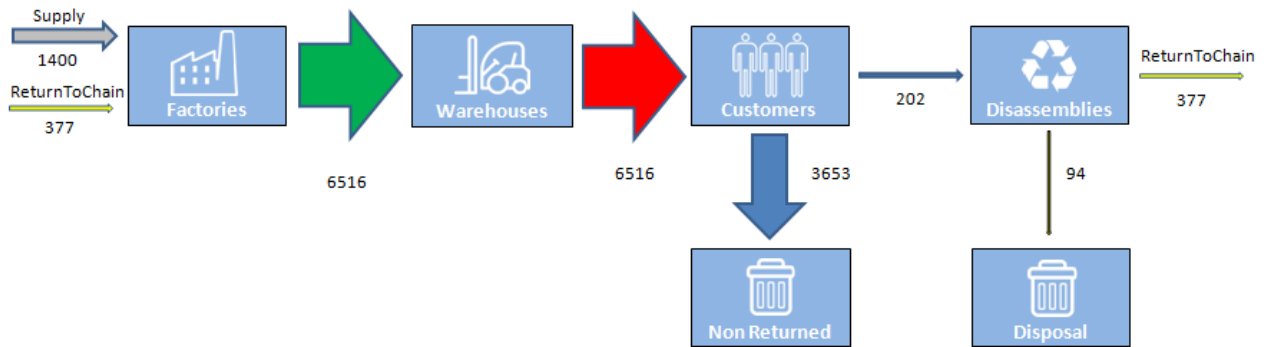


Figura 25b. Diagrama de Sankey del cas RPF_LOW

Comparant els dos casos, les úniques xifres que canvien són les quantitats retornades, no retornades i l'abocament als centres de desmuntatge. Per tant, es pot dir que aquest paràmetre tan sols afecta als fluxos de clients i centres de desmuntatge.

Es torna a recordar que els balanços de matèria de les entitats estan regits pels coeficients de relació de productes. Per aquest motiu la quantitat entrant no sempre és igual a la quantitat sortint d'una entitat, a més a més de la possible existència de stock.

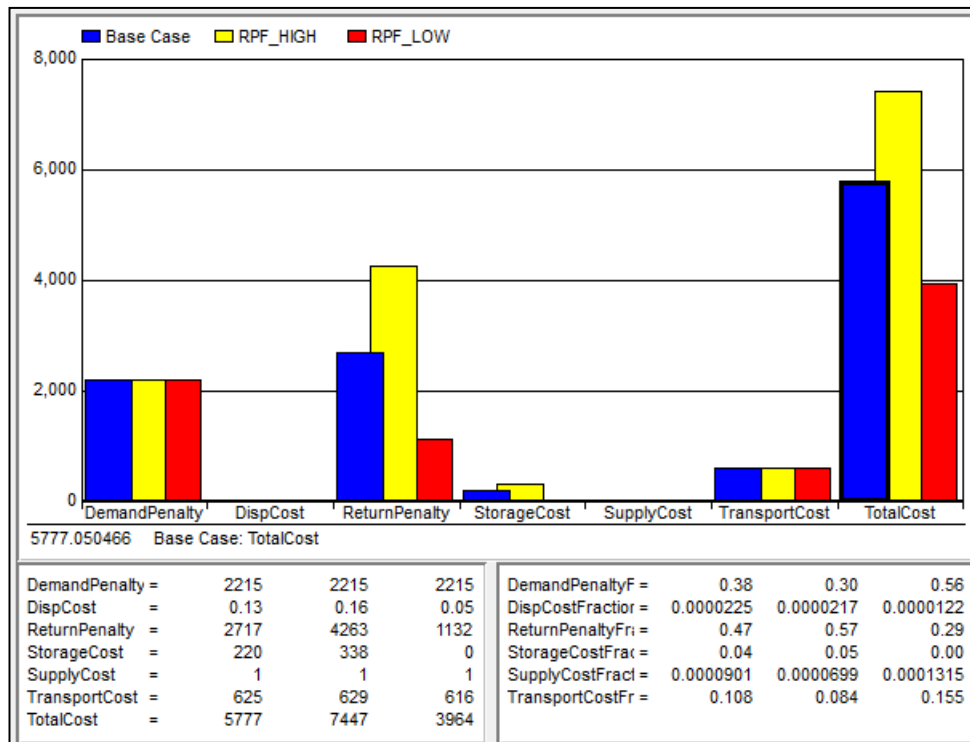


Figura 26. Costos dels casos variant la penalització de retorn de clients

En quant a costos, a mida que augmenta el factor de penalització també ho fa el cost de penalització, el cost d'estocar i, com a conseqüència, el total. Encara que

es decideixi recollir més quantitat de producte als clients, la quantitat no recollida penalitza encara més. Per una altra banda, al augmentar la quantitat que entra al centre de desmuntatge fa augmentar el cost d'estoc. El retorn a les plantes no es pot augmentar perquè aleshores es tindria més subministrament del necessari, i l'abocament de residus té un límit que no es pot superar. L'única opció es mantenir la quantitat sobrant com a estoc.

6.2. Factor de penalització de la demanda

Quan no es satisfà tota la demanda dels clients, es té en compte una penalització.

En el BaseCase aquest factor té un valor d'1, en el cas DPF_HIGH té un valor de 2 i en el cas DPF_LOW té un valor de 0,5.

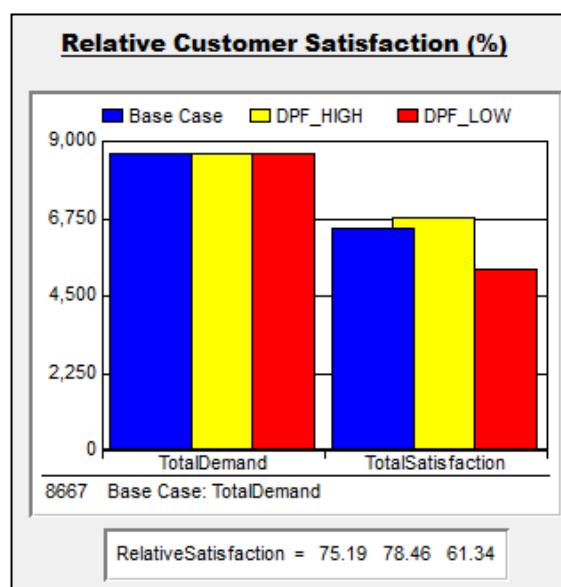


Figura 27. Satisfacció relativa de la demanda de cada cas

Augmentant el factor de penalització de la demanda s'aconsegueix que la satisfacció dels clients sigui més alta. Aquest fet comporta les següents conseqüències:

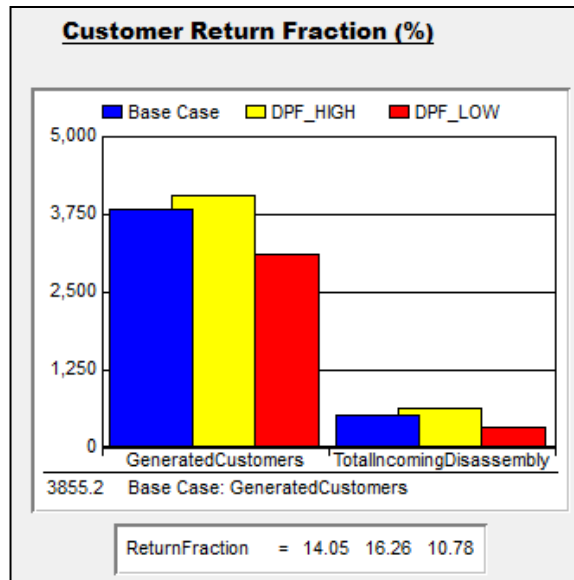


Figura 28. Representació del retorn de clients

Com que els clients reben més quantitat de producte, generen més matèria per ser retornada. Per tant, es decideix augmentar el retorn als centres de desmuntatge per evitar un sobre cost.

Al mateix temps, al necessitar més producte per subministrar als clients, s'augmenta la quantitat reciclada que retorna a les plantes de processat per no haver d'augmentar el subministrament de matèria prima.

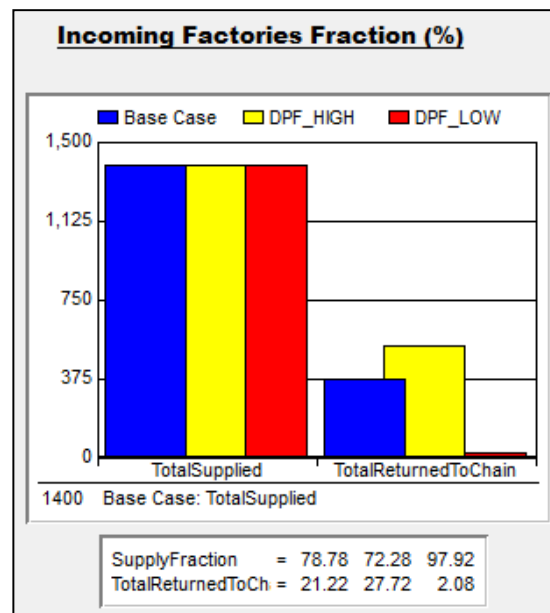


Figura 29. Representació de les fraccions de subministrament a les plantes de processat

DemandPenalizationFactor=2

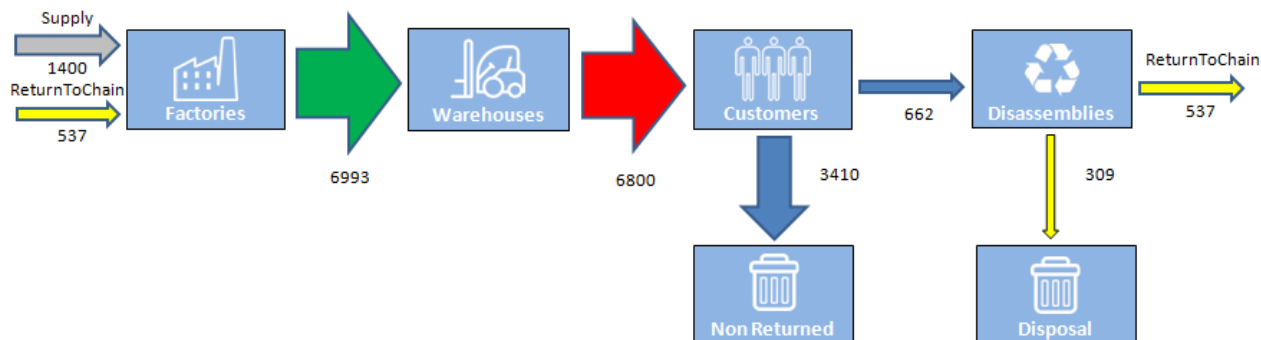


Figura 30a. Diagrama de Sankey del cas DPF_HIGH

DemandPenalizationFactor=0,5

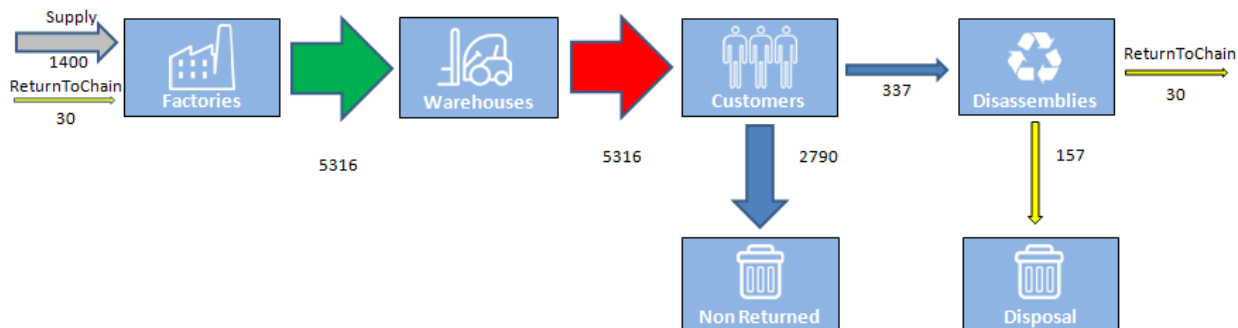


Figura 30b. Diagrama de Sankey del cas DPF_LOW

Una altra diferència entre el casos de les figures 30a i 30b és que el valor dels fluxos totals varia d'acord amb el factor de penalització de la demanda.

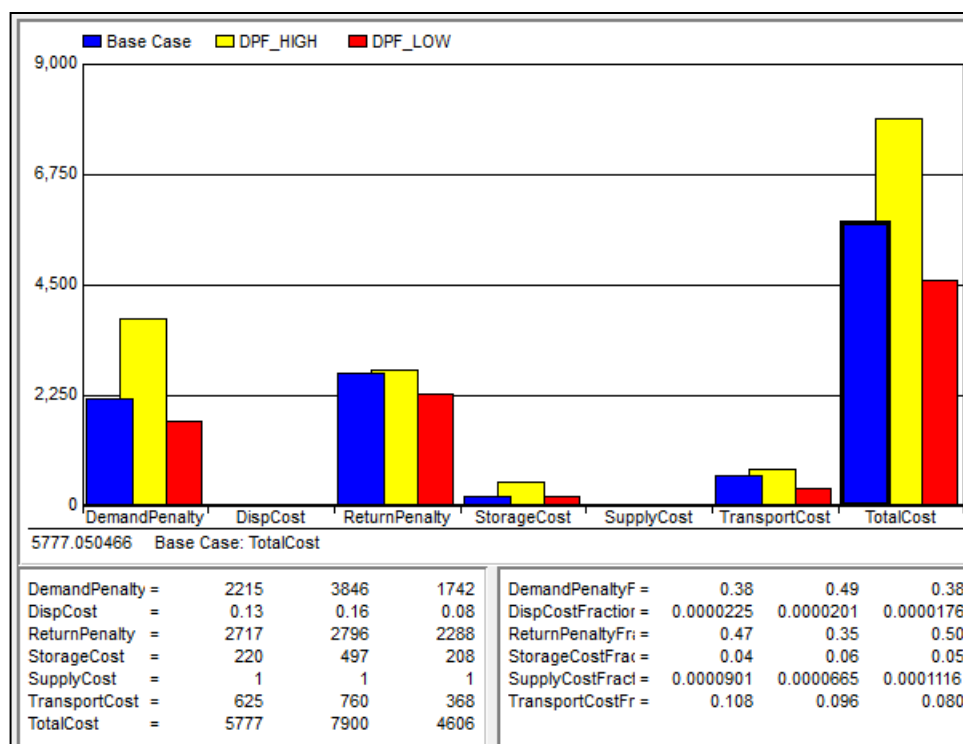


Figura 31. Costos dels casos variant la penalització de la demanda

Al augmentar aquest factor de penalització també augmenten la resta de costos excepte el de subministrament. Per tenir menor quantitat de demanda que penalitzi, es decideix augmentar el flux entre entitats. S'aconsegueix augmentant la quantitat de producte reciclat de la cadena, així el subministrament del proveïdor roman constant. Tot i així, per cada unitat de demanda no complerta es penalitza més i, és per això, que s'eleva el cost d'aquesta penalització.

6.3. Límits de subministrament

Segons el contracte que es tingui amb el proveïdor, els límits superiors i inferiors de subministrament de matèria prima poden variar.

Primerament, l'estudi es centra en el límit inferior de subministrament. Es tracta de la quantitat mínima de matèria prima (en tones) que la planta de processat està obligada a acceptar cada cert temps. En els tres casos estudiats, s'ha establert que la quantitat sigui per a cada dia laboral.

En el BaseCase aquest paràmetre té un valor de 35, en el cas LSupB_LOW(20) té un valor de 20 i en el cas LSupB_LOW(1) té un valor d'1.

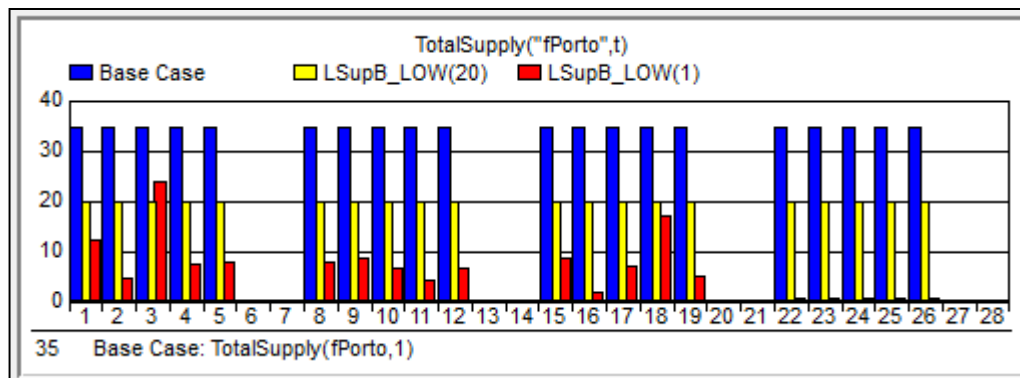


Figura 32. Quantitats de matèria prima subministrada a la planta de processat de Porto

Es pot veure que quan més baix és el límit inferior, més flexibilitat ofereix a les plantes de processat.

En el cas LSupB_LOW(1), la planta de processat té com a subministrament la quantitat que necessita per a cada dia, de manera que només produeix segons les necessitats.

En canvi, en el BaseCase, la planta de processat es veu obligada a acceptar cada dia laboral la mateixa quantitat de matèria prima i, justament, coincideix amb el límit inferior de subministrament. D'aquesta manera, la planta de processat es troba amb un excés de matèria prima i la cadena de subministrament es veu obligada a prendre una sèrie de decisions. Cal remarcar que els clients segueixen tenint la mateixa quantitat de demanda satisfeta en els tres casos. Per tal d'evitar un excés de producció, la solució és reciclar menys quantitat de matèria:

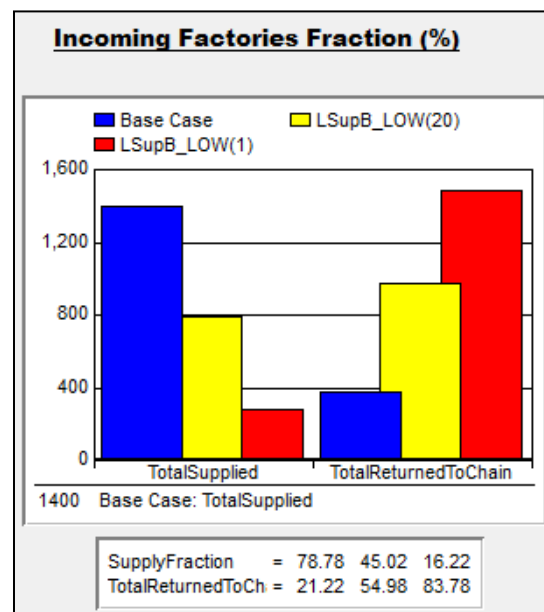


Figura 33. Fraccions de subministrament a les plantes de processat

Així, sempre es disposa de la mateixa quantitat de matèria a l'entrada de la planta de processat. La diferència és senzillament el lloc del que prové

Per aconseguir aquesta quantitat de matèria reciclada a la planta de processat cal modificar la quantitat recollida al client segons convingui:

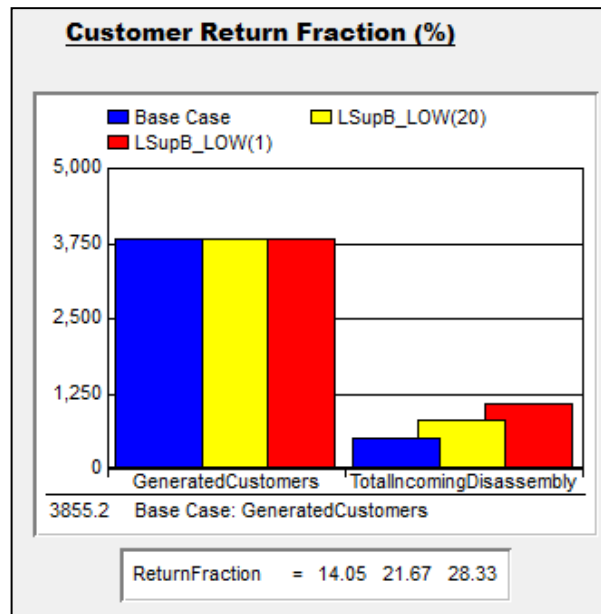


Figura 34. Fracció de retorn dels clients als centres de desmuntatge

LowerSupplyBound(f,t)=20

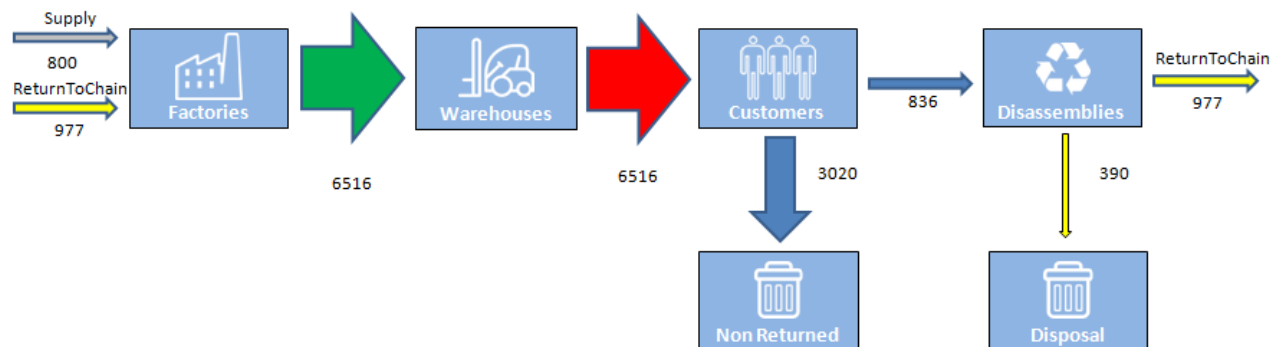


Figura 35a. Diagrama de Sankey del cas LSupB_LOW(20)

LowerSupplyBound(f,t)=1

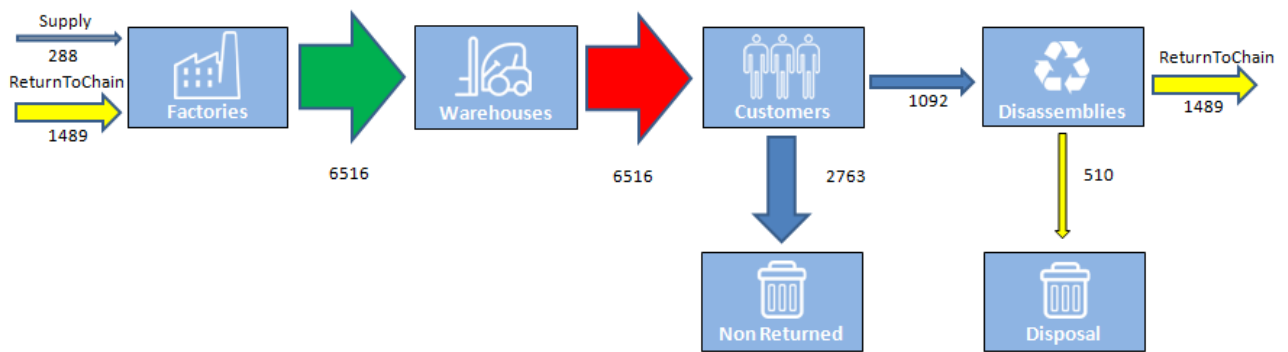


Figura 35b. Diagrama de Sankey del cas LSupB_LOW(1)

A les figures 35a i 35b es poden veure els canvis exposats anteriorment.

S'ha decidit no fer un estudi del límit superior de subministrament degut a que la solució òptima sempre seria escollir un subministrament quant més baix millor (sempre i quan no augmentés la demanda). Per tant, el límit amb més pes seria l'inferior.

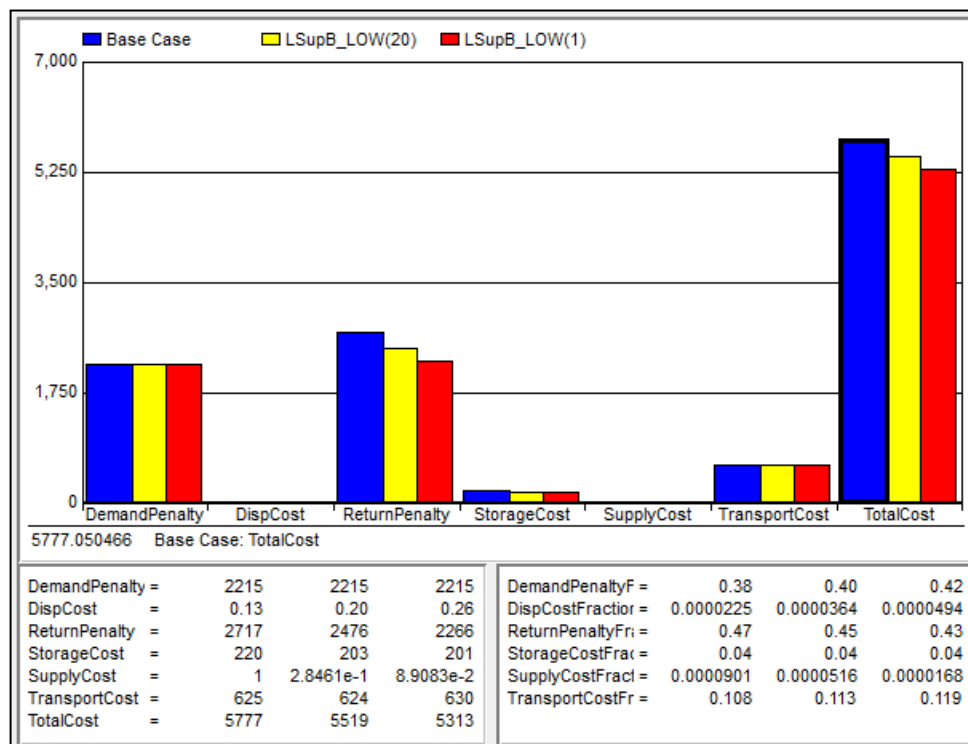


Figura 36. Costos dels casos variant els límits de subministrament

Com que la quantitat reciclada de producte serà major, es decideix augmentar el retorn dels clients. Així, la penalització del retorn es veu disminuïda.

6.4. Demanda dels clients

S'ha pres la decisió d'estudiar com variarien els resultats si canviessin les demandes dels clients. Per a tots els casos, la demanda és introduïda amb un valor (en tones) per a cada dia. La demanda ha de ser complerta el mateix dia.

El BaseCase té com a demanda una funció de valors aleatoris entre 2 i 6, el cas CustomerDemand_HIGH té una funció de valors aleatoris entre 3 i 8, i el cas CustomerDemand_LOW entre 1 i 4.

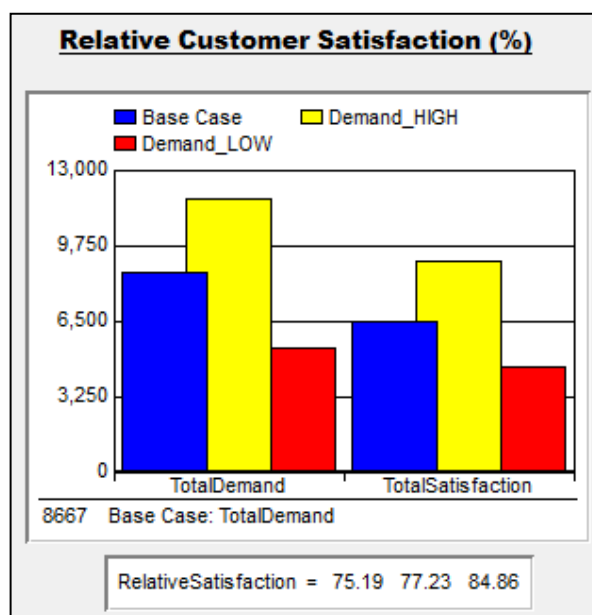


Figura 37. Satisfacció relativa de la demanda dels clients

No hi ha cap comportament general en quant a la satisfacció relativa de la demanda. Tot dependrà de les decisions que es prenguin en cada cas en particular.

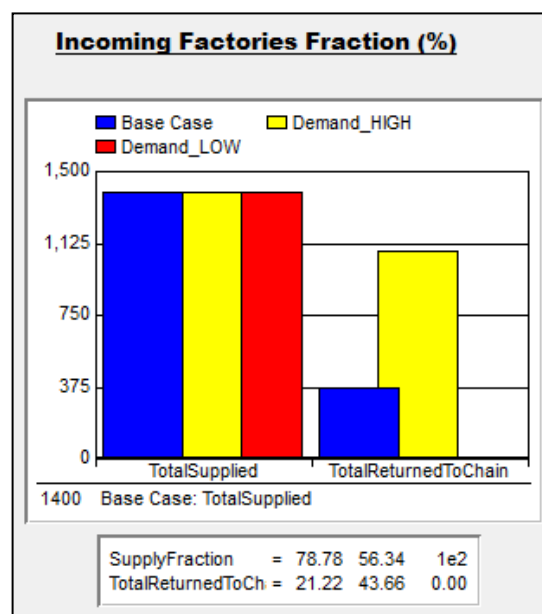


Figura 38. Fraccions de subministrament de les plantes de processat

En el cas Demand_LOW no hi ha retorn de matèria dels centres de desmuntatge a plantes de processat. La explicació d'aquest fet és que com que s'està obligat a acceptar una quantitat de matèria prima mínima (relativament alta) i la demanda és baixa, es té un excés de producte. Aquest excés roman com a stock a les

entitats, ja que la demanda és insuficient. Com a conseqüència d'aquest stock, no és necessari acumular-ne més, és a dir, s'ha de produir el mínim possible per evitar que aquest stock vagi creixent. La única solució és no recircular matèria a les plantes de processat i alimentar-les solament amb el subministrament mínim del proveïdor. D'altra banda, el retorn de clients vindrà marcat per la restricció legal d'abocament de residus i per la capacitat i cost de stock de cada centre de desmuntatge, ja que seran les úniques sortides que tindrà el producte dels centres de desmuntatge.

CustomerDemand(ma,c,t)=aleatorio.entre(3;8)

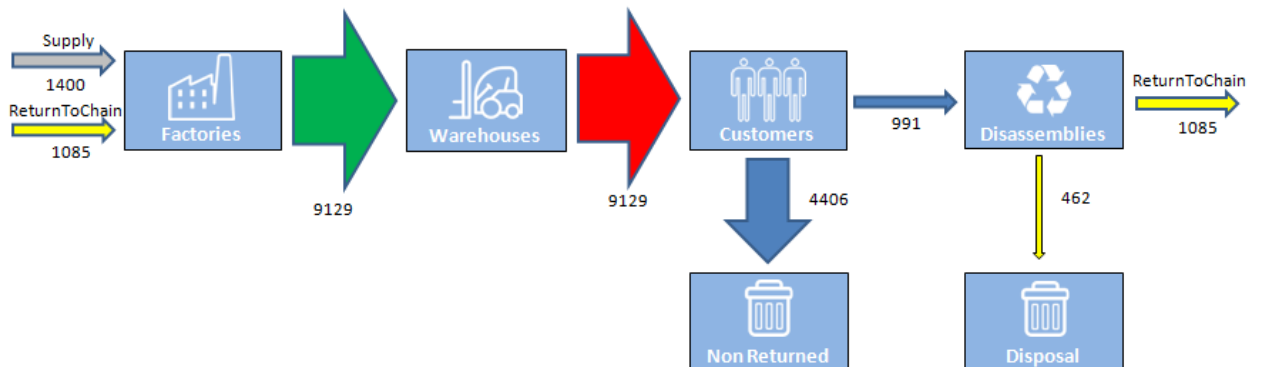


Figura 39a. Diagrama de Sankey del cas Demand_HIGH

CustomerDemand(ma,c,t)=aleatorio.entre(1;4)

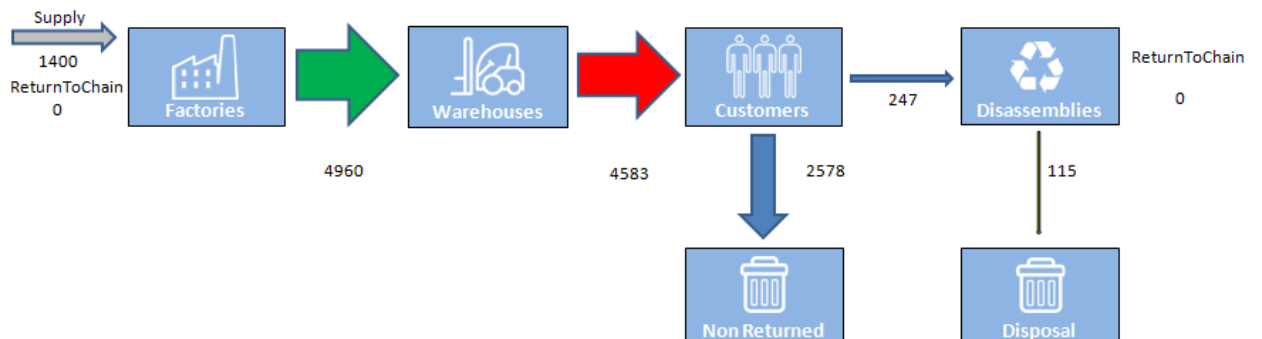


Figura 39b. Diagrama de Sankey del cas Demand_LOW

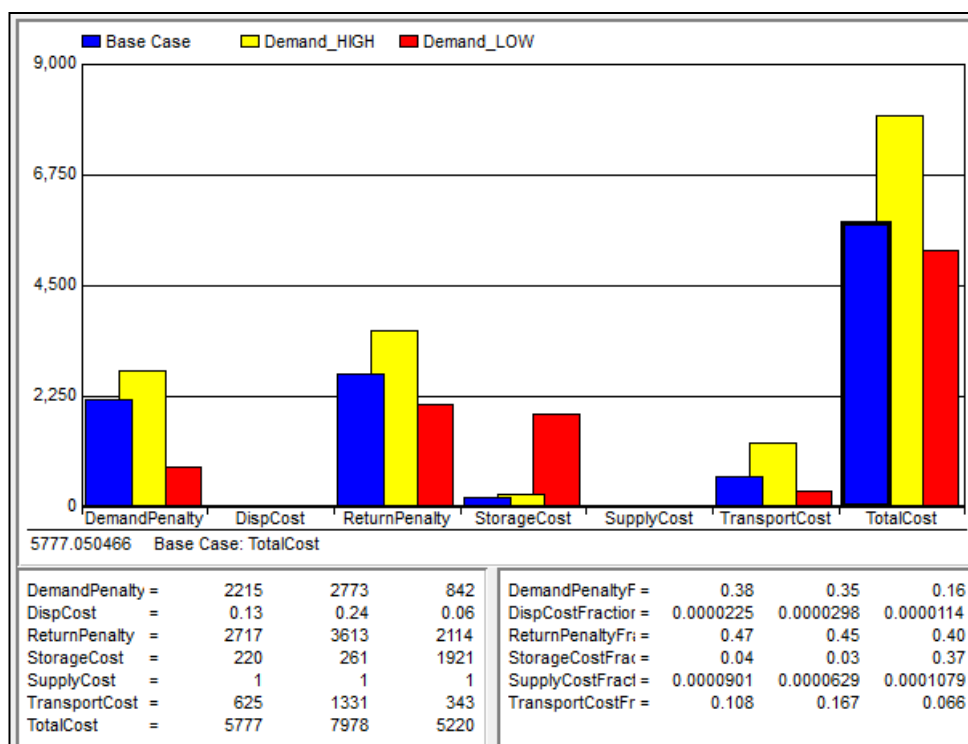


Figura 40. Costos dels casos variant la demanda dels clients

Al tenir una demanda més alta es requereix de més quantitat de producte en la majoria dels fluxos, per tant, augmenta la majoria dels costos. No passa això amb el cost d'estocar. Com que la demanda és relativament baixa i es té un excés de producte com a conseqüència del límit inferior de subministrament, augmenta tant el cost d'estocar, degut a que el producte sobrant queda estocat.

6.5. Factor mínim de reciclatge

Es tracta d'un paràmetre imposat per la legislació que indica la fracció mínima que s'ha de reciclar. Si es resta a la unitat s'obté la fracció màxima que es pot abocar com a residus.

En aquest model s'ha implementat de manera que els centres de desmuntatge no poden superar la fracció màxima d'abocament de residus. La quantitat emmagatzemada en aquests centres és vista com a fracció reciclada.

El BaseCase té un valor de 0,8 i el RecoveryTarget_LOW té un valor de 0,4. En el primer cas es pot abocar com a residus un 20% i en el segon cas un 60%.

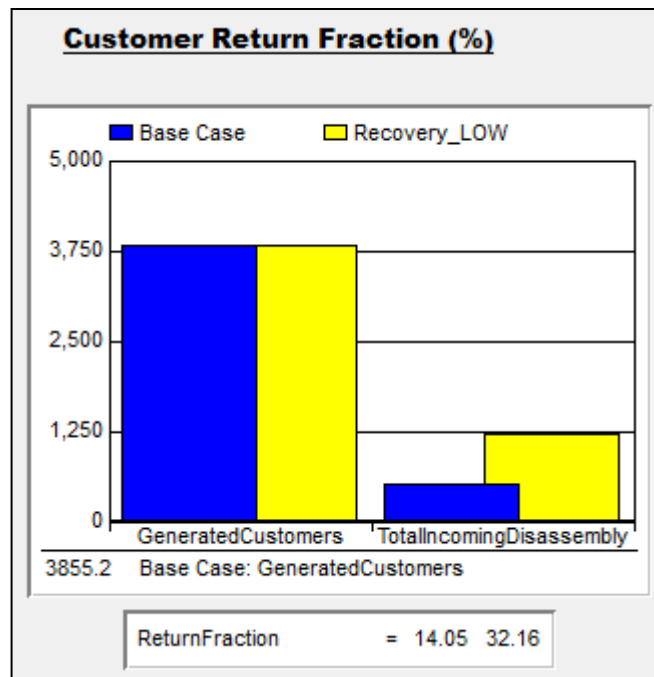


Figura 41. Fracció de retorn dels clients a centres de desmuntatge

Com que en el segon cas es pot abocar una quantitat més alta de residus als centres de desmuntatge, es decideix augmentar el retorn dels clients per, així, estalviar-se penalitzacions i llençar-lo com a residus.

En la següent figura es representa els percentatges d'abocament de residus:

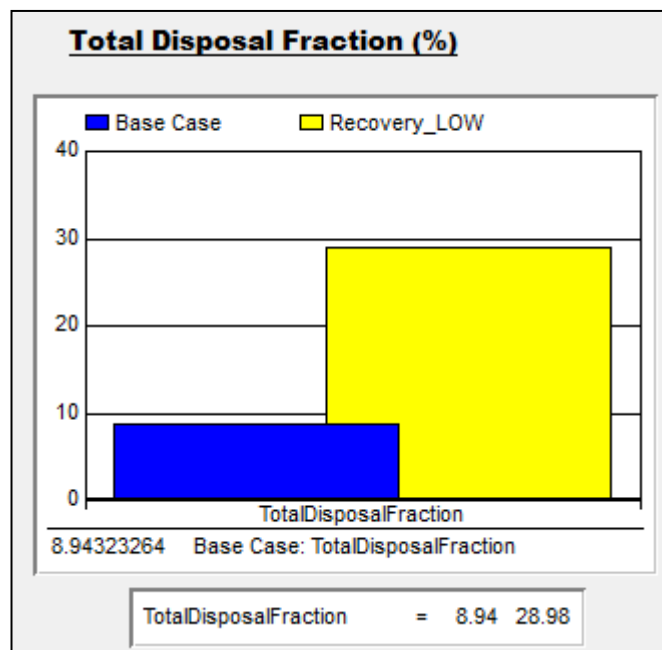


Figura 42. Fracció de l'abocament de residus als centres de desmuntatge

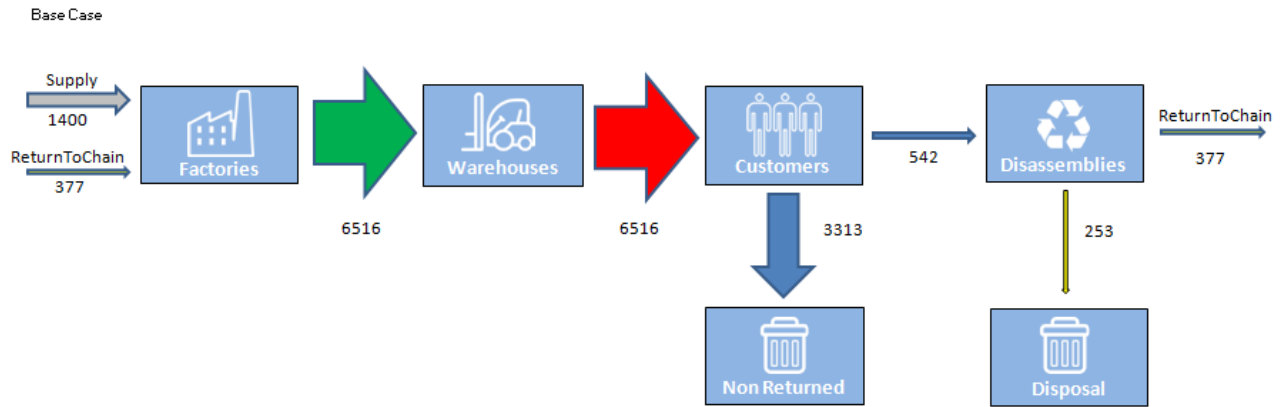


Figura 43a. Diagrama de Sankey del cas BaseCase

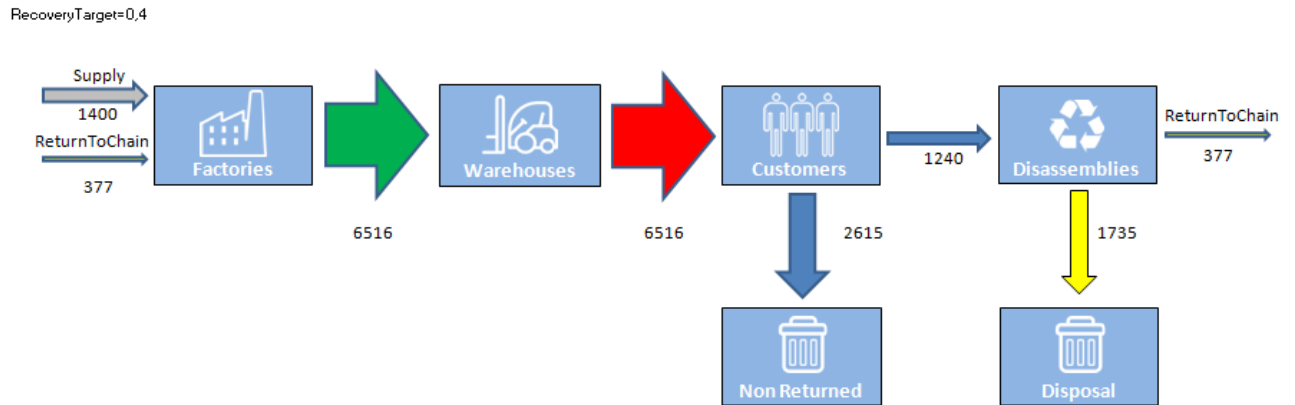


Figura 43b. Diagrama de Sankey del cas Recovery_LOW

Es pot veure com augmenten considerablement els fluxos de retorn dels clients i d'abocament de residus als centres de desmuntatge en el segon cas (Figura 43b).

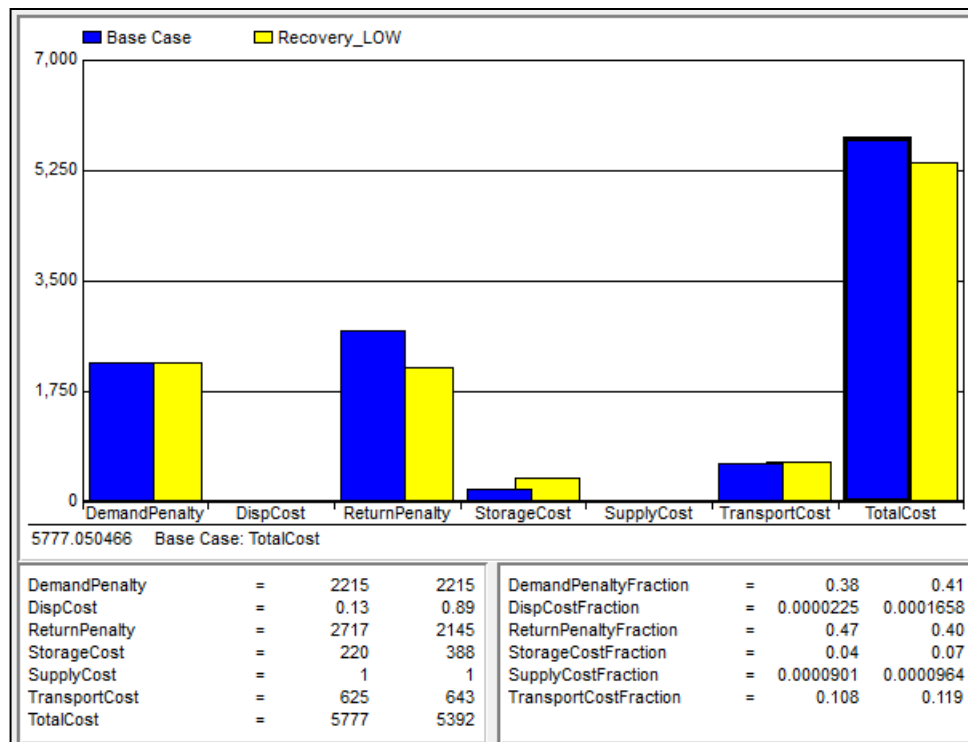


Figura 44. Costos dels casos variant el factor mínim de reciclatge

Al poder abocar més quantitat de residus, s'augmenta la el retorn dels clients per a que no penalitzi i s'aboca com a residu posteriorment. Per això, la penalització del retorn és menor en el segon cas. Com a conseqüència augmenta el cost de transportar aquest retorn de clients. Per una altra banda, al tenir més retorn als centres de desmuntatge augmenta l'estoc per respectar els límits d'abocament.

Després de realitzar els casos d'estudi que s'han cregut convenients, es pot determinar la validesa del model implementat, ja que els resultats obtinguts són lògics i raonables..

CAPÍTOL 7:

CONCLUSIONS I IDEES

DE FUTUR

Després de l'anàlisi dels resultats obtinguts es pot afirmar que:

- S'ha desenvolupat el model base de *Closed-Loop Supply Chain* per tal d'obtenir una planificació més detallada i disposar de les dades necessàries per la posta en marxa de la cadena de subministrament.
- S'ha escollit el software AIMMS per implementar el model després d'estudiar detingudament les diferents alternatives de software.
- S'ha implementat el model desenvolupat amb les adaptacions corresponents en quant a horitzó i s'hi han afegit utilitats, com per exemple, la funció calendari.
- S'ha creat una interfície d'usuari per representar els resultats obtinguts i facilitar la navegació de l'usuari.
- S'han realitzat una sèrie de variacions sobre els paràmetres que s'ha cregut convenient per validar el model implementat. Posteriorment s'ha procedit a explicar els efectes d'aquestes variacions i es determina la validesa del model.

Tot i que el resultat del projecte ha estat satisfactori, s'ha estudiat la possibilitat de millorar-lo en un futur. La millora estaria basada en introduir un *rolling horizon* al model.

En aquest projecte, s'ha establert un horitzó d'un mes i abans del primer dia ja es coneix què s'haurà de fer exactament cada dia, és a dir, es planifica el mes i es preveu el que s'ha de complir cada dia. En la teoria resulta molt efectiu aquest mètode, però en la pràctica no sempre s'arriba a complir el que s'ha previst. Un exemple de que no es compleixi el que està previst seria que a mitjans de mes, per exemple, canviés la normativa de residus i s'hagués d'adaptar la planificació a les restriccions. Un altre cas seria que, per qualsevol motiu, una planta de processat no pogués estar operativa durant un temps. en aquests casos s'hauria de refer la planificació amb les noves dades a partir d'un dia determinat.

En resum, amb la implantació del *rolling horizon* es podria refer la planificació des d'un dia en endavant en el cas de que sorgís qualsevol imprevist.

El model funcionaria amb dos conjunts de dades. El primer seria per les dades teòriques conegudes abans del primer dia, mentre que el segon conjunt seria per les dades reals, que s'haurien d'anar introduint al finalitzar cada dia. D'aquesta manera, s'introduirien les dades obtingudes a diari i, en el cas de ser diferents a les teòriques, el model actualitzaria la planificació restant.

S'ha decidit no implementar aquesta funció, degut a que les dades que s'haurien d'introduir diàriament (de forma manual) seria una quantitat molt gran i requeriria molt de temps. Per una altra banda, la gestió de dades amb aquesta funció seria més complexa degut al gran tamany de les dades. Es creu que aquesta feina estaria més al camp d'un informàtic per tal de fer la introducció d'aquestes més eficient. Tot i així, s'ha deixat el model preparat per tenir una continuïtat amb la funció esmentada.

CAPÍTOL 8: AVALUACIÓ ECONÒMICA DEL PROJECTE

En aquest capítol es calcula el cost que té dur a terme aquest projecte per un enginyer professional.

Primerament es disposa a calcular les despeses. Es divideixen en despeses directes (afecten exclusivament a la realització del projecte) i despeses indirectes (repercuteixen a la realització de diferents projectes).

Com a despeses directes únicament són els impresos, còpies i enquadernació de la memòria del projecte:

Despeses Directes	
Impresos, còpies i enquadernació	120,00 €

Taula 2. Despeses directes del projecte

Com a despeses indirectes hi ha el lloguer del despatx, l'assegurança d'enginyer, les factures d'aigua, electricitat, gas i telefonia del despatx, l'equipament informàtic (ordinador, monitors, etc) i el software (Windows, Microsoft Office, AIMMS i el Solver CPLEX). A més, s'afegeix un 10% d'aquestes despeses com a marge per assegurar-se de que no es perden diners. Aquestes despeses són anuals, de manera que s'han de convertir posteriorment a €/h dividint la xifra total per les hores hàbils que té l'any:

Despeses Indirectes		
Lloguer del despatx		9.600,00 €/any
Assegurança enginyer		3.500,00 €/any
Aigua		300,00 €/any
Electricitat		1.080,00 €/any
Gas		700,00 €/any
Telefonia		720,00 €/any
Equipament informàtic		356,50 €/any
Software		2.988,00 €/any
	SubTotal	19.244,50 €/any
Marge sobre despeses		1.924,45 €/any
	TOTAL	21.168,95 €/any
	TOTAL/h	11,46 €/h

Taula 3. Despeses indirectes del projecte

Paràmetres		
1 any	Hàbil	Treballat
Mesos	11	4
Dies/mes	21	21
Hores/dia	8	4

Marge despeses	10%
----------------	-----

Taula 4. Paràmetres de les despeses indirectes

Software		
Windows 8.1	119,00 €/any	119,00 €/any
Microsoft Office	269,00 €/any	269,00 €/any
Llicència AIMMS	6.500,00 €/5anys	1.300,00 €/any
Solver CPLEX	6.500,00 €/5anys	1.300,00 €/any

Taula 5. Despeses de software

Equipament informàtic		
PC	949,00 €/4anys	237,25 €/any
Ratolí	35,00 €/4anys	8,75 €/any
Teclat	70,00 €/4anys	17,50 €/any
Monitor	372,00 €/4anys	93,00 €/any

Taula 6. Despeses d'equipament informàtic

Un cop calculades les despeses, cal establir els preu/hora de l'enginyer.

Al la xifra dels honoraris base de l'enginyer, cal sumar-li el preu/hora dels costos indirectes calculats anteriorment. Seguidament es multiplica aquest valor pels dies dedicats a aquest projecte (4 mesos) i es té una xifra en € que només cal sumar-li els costos directes:

Honoraris	
Honoraris enginyer	30,00 €/h
Suma honoraris/h	41,46 €/h
Honoraris Totals	14.048,90 €

Taula 7. Honoraris del projecte

Com a resultat surten uns honoraris de 14.048,90€ per la realització d'aquest projecte.

CAPÍTOL 9:

BIBLIOGRAFIA

9.1. Referències bibliogràfiques

- Articles:

Gomes, Maria Isabel, Ana Paula Barbosa, i Augusto Q. Novais. 2009. Simultaneous design and planning of supply chains with reverse flows: A generic modelling framework.

V. Daniel R. Guide, Jr and Luck N. Van Wassenhove. 2010. The Evolution of Closed-Loop Supply Chain Research.

Zukui Li, Marianthi G. Ierapetritou. 2010. Rolling horizon based planning and scheduling integration with production capacity consideration.

Kannan Govindan, Hamed Soleimani, Devika Kannan. 2013. Reverse logistics and closed-loop supply chain: A comprehensive review to explore the future.

Guang-fen Yang, Zhi-ping Wang, Xiao-qiang Li. 2011. The optimization of the closed-loop supply chain network.

- Pàgines web:

www.aimms.com (Consultada 21/10/14)

www.ibm.com/es/es (Consultada 04/10/14)

www.gams.com (Consultada 24/10/14)

www.es.mathworks.com (Consultada 24/10/14)

www.aecoc.es (Consultada 15/11/14)

9.2. Bibliografía de Consulta

- Pàgines web:

www.lifeinformatica.com (consultada 04/01/15)

www.microsoftstore.com (consultada 04/01/15)

www.facturaluz.cnmc.es (consultada 04/01/15)

www.gasnaturalfenosa.es (consultada 04/01/15)

www.aiguesdebarcelona.cat (consultada 04/01/15)

www.soloingenieria.net (consultada 04/01/15)

www.telefonica.es (consultada 04/01/15)

www.habitaclia.com (consultada 04/01/15)

www.copisteriaonline.es (consultada 04/01/15)

ANNEX : CODI AIMMS

MAIN MODEL Main_TFG

DECLARATION SECTION

```
SET:
  identifier   : Locations
  indices      : i, j
  definition    : data { aBraga, aCoimbra, aLeiria, aLisboa, aPorto,
aSantarem, aSetubal, aViseu, fLisboa, fPorto, cAveiro, cBeja, cBraga, cBraganca,
cCBranco, cCoimbra, cEvora, cFaro, cGuarda, cLeiria, cLisboa, cPortalegre, cPorto,
cSantarem, cSetubal, cVCastelo, cViseu, cVReal, rBraga, rCoimbra, rLeiria, rLisboa,
rPorto, rSantarem, rSetubal, rViseu } ;

SET:
  identifier   : Factory
  subset of    : Locations
  index        : f
  definition    : data { fPorto, fLisboa } ;

SET:
  identifier   : Warehouse
  subset of    : Locations
  index        : a
  definition    : data { aBraga, aCoimbra, aLisboa, aPorto, aSetubal, aLeiria,
aSantarem, aViseu } ;

SET:
  identifier   : Customer
  subset of    : Locations
  index        : c
  definition    : data
                  { cAveiro      , cBeja      , cBraga      , cBraganca   ,
cCBranco      , cCoimbra      , cEvora      , cFaro      , cGuarda     ,
                  cLeiria      , cLisboa      , cPortalegre, cPorto      ,
cSantarem     , cSetubal     , cVCastelo   , cViseu     , cVReal      } ;

SET:
  identifier   : Disassembly
  subset of    : Locations
  index        : r
  definition    : data { rBraga, rCoimbra, rLeiria, rLisboa, rPorto,
rSantarem, rSetubal, rViseu } ;

SET:
  identifier   : Products
  indices      : m, mp
  definition    : data { mf1, mf2, mf3, ma1, ma2, ma3, ma4, ma5, ma6, mc1,
mc2, mrl } ;
```

```
SET:
  identifier : ProductA
  subset of  : Products
  index      : ma
  definition : data { ma1, ma2, ma3, ma4, ma5, ma6 } ;

SET:
  identifier : ProductF
  subset of  : Products
  index      : mf
  definition : data { mf1, mf2, mf3 } ;

SET:
  identifier : ProductC
  subset of  : Products
  index      : mc
  definition : data { mc1, mc2 } ;

SET:
  identifier : ProductR
  subset of  : Products
  index      : mr
  definition : data { mr1 } ;

SET:
  identifier : Periods
  index      : t
  definition : data{1..28} ;

SET:
  identifier : Week1
  subset of  : Periods
  index      : t1
  definition : data { 1, 2, 3, 4, 5, 6, 7 } ;

SET:
  identifier : Week2
  subset of  : Periods
  index      : t2
  definition : data { 8, 9, 10, 11, 12, 13, 14 } ;

SET:
  identifier : Week3
  subset of  : Periods
  index      : t3
  definition : data { 15, 16, 17, 18, 19, 20, 21 } ;

SET:
  identifier : Week4
  subset of  : Periods
  index      : t4
  definition : data { 22, 23, 24, 25, 26, 27, 28 } ;

SET:
  identifier : Weekends
  subset of  : Periods
  index      : tw
  definition : data { 6, 7, 13, 14, 20, 21, 27, 28 } ;

PARAMETER:
  identifier : dst
  index domain : (i,j)
  definition : data

PARAMETER:
  identifier : cip
  index domain : (m,i)
  definition : data
```

```

PARAMETER:
  identifier   : SupplyingCost
  index domain : (m,i,t)
  definition   : cip(m,i)*(1+0.03) ;

PARAMETER:
  identifier   : cif
  index domain : (m,i,j)
  definition   : data

PARAMETER:
  identifier   : TransportationCost
  index domain : (m,i,j,t)
  definition   : cif(m,i,j)*dst(i,j)*(1+0.03) ;

PARAMETER:
  identifier   : CustomerDemand
  index domain : (m,i,t)
  definition   : data

PARAMETER:
  identifier   : UpperLeavingFlow
  index domain : (i,j)
  definition   : data

PARAMETER:
  identifier   : LowerLeavingFlow
  index domain : (i,j)
  definition   : data { } ;

PARAMETER:
  identifier   : UpperSupplyBound
  index domain : (i,t)
  definition   : data

PARAMETER:
  identifier   : LowerSupplyBound
  index domain : (i,t)
  definition   : data

PARAMETER:
  identifier   : CalendarParameter
  index domain : (t)
  range       : binary
  definition   : data
1, 10 : 1, 11 : 1, 12 : 1, 15 : 1, 16 : 1, 17 : 1, 18 : 1,
    19 : 1, 22 : 1, 23 : 1, 24 : 1, 25 : 1, 26 : 1 } ;

PARAMETER:
  identifier   : cid
  index domain : (m,i)
  definition   : data

PARAMETER:
  identifier   : DisposalCost
  index domain : (m,i,t)
  definition   : cid(m,i)*(1+0.03) ;

PARAMETER:
  identifier   : ProductRelation
  index domain : (m,mp)
  definition   : data
: 0.390, ( mf2, mf2 ) : 1.000, ( mf2, ma2 ) : 0.500,
    ( mf2, ma4 ) : 0.500, ( mf3, mf3 ) : 1.000, ( mf3, ma5 )
: 0.610, ( mf3, ma6 ) : 0.390, ( ma1, ma1 ) : 1.000,
    ( ma1, mr1 ) : 0.450, ( ma2, ma2 ) : 1.000, ( ma2, mr1 )
: 0.700, ( ma3, ma3 ) : 1.000, ( ma3, mr1 ) : 0.500,

```

```

( ma4, ma4 ) : 1.000, ( ma4, mr1 ) : 0.800, ( ma5, ma5 )
: 1.000, ( ma5, mr1 ) : 0.400, ( ma6, ma6 ) : 1.000,
( ma6, mr1 ) : 0.900, ( mc1, mf1 ) : 2.000, ( mc1, mf2 )
: 1.000, ( mc1, mf3 ) : 1.000, ( mc1, mc1 ) : 1.000,
( mc2, mf2 ) : 1.000, ( mc2, mf3 ) : 2.000, ( mc2, mc2 )
: 1.000, ( mr1, mc1 ) : 1.333, ( mr1, mc2 ) : 1.000,
( mr1, mr1 ) : 1.000 } ;

```

```

PARAMETER:
  identifier : RecoveryTarget
  initial data : 0.8 ;

```

```

PARAMETER:
  identifier : StockedLimit
  index domain : (i)
  definition : data

```

```

PARAMETER:
  identifier : StockCost
  index domain : (m,i,t)
  definition : data

```

```

PARAMETER:
  identifier : DemandPenalizationFactor
  initial data : 1 ;

```

```

PARAMETER:
  identifier : ReturnPenalizationFactor
  initial data : 0.8 ;

```

```

PARAMETER:
  identifier : Transported
  index domain : (i,j,t)
  definition : sum[m, Transport(m,i,j,t)] ;

```

```

VARIABLE:
  identifier : Transport
  index domain : (m,i,j,t)
  range : nonnegative ;

```

```

VARIABLE:
  identifier : Stocked
  index domain : (m,i,t)
  range : nonnegative ;

```

```

VARIABLE:
  identifier : Supply
  index domain : (m,i,t)
  range : nonnegative ;

```

```

VARIABLE:
  identifier : Disposal
  index domain : (m,i,t)
  range : nonnegative ;

```

```

VARIABLE:
  identifier : DemandPenalty
  range : nonnegative
  definition : (sum[ (ma,c,t), CustomerDemand(ma,c,t) ]-sum[ (ma,a,c,t),
Transport(ma,a,c,t) ])*round(DemandPenalizationFactor*(1+0.03),+2) ;

```

```

VARIABLE:
  identifier : ReturnPenalty
  range : nonnegative
  definition : (sum[ (ma,mr,a,c,t),
ProductRelation(ma,mr)*transport(ma,a,c,t) ]-sum[ (mr,c,r,t),
transport(mr,c,r,t) ])*round(ReturnPenalizationFactor*(1+0.03),+2) ;

```

```

VARIABLE:
  identifier : TransportCost

```



```

        range      : nonnegative
        definition  : sum[ (m,i,j,t) ,
TransportationCost(m,i,j,t)*Transport(m,i,j,t)] ;

VARIABLE:
    identifier    : SupplyCost
    range         : nonnegative
    definition     : sum[ (m,i,t) , SupplyingCost(m,i,t)*Supply(m,i,t)] ;

VARIABLE:
    identifier    : DispCost
    range         : nonnegative
    definition     : sum[ (m,i,t) , DisposalCost(m,i,t)*Disposal(m,i,t)] ;

VARIABLE:
    identifier    : StorageCost
    range         : nonnegative
    definition     : sum[ (m,i,t) , Stocked(m,i,t)*StockCost(m,i,t)] ;

VARIABLE:
    identifier    : TotalCost
    range         : free
    definition     :
ReturnPenalty+DemandPenalty+TransportCost+SupplyCost+DispCost+StorageCost ;

CONSTRAINT:
    identifier    : DailyDemand
    index domain  : (ma,c,t)
    definition     : sum[a, Transport(ma,a,c,t)]<=CustomerDemand(ma,c,t) ;

CONSTRAINT:
    identifier    : FactoryBalance
    index domain  : (mf,f,t)
    definition     : if ord(t)=1 then

                        sum[mc, ProductRelation(mc,mf)*(sum[r,
transport(mc,r,f,t)]+Supply(mc,f,t))]=sum[a, transport(mf,f,a,t)]+Stocked(mf,f,t)

                        else

                        Stocked(mf,f,t-1)+sum[mc, ProductRelation(mc,mf)*(sum[r,
transport(mc,r,f,t)]+Supply(mc,f,t))]=sum[a, transport(mf,f,a,t)]+Stocked(mf,f,t)

                        endif ;

CONSTRAINT:
    identifier    : WarehouseBalance
    index domain  : (ma,a,t)
    definition     : if ord(t)=1 then

                        sum[ (mf,f) ,
ProductRelation(mf,ma)*transport(mf,f,a,t)]=sum[c,
transport(ma,a,c,t)]+Stocked(ma,a,t)

                        else

                        Stocked(ma,a,t-1)+sum[ (mf,f) ,
ProductRelation(mf,ma)*transport(mf,f,a,t)]=sum[c,
transport(ma,a,c,t)]+Stocked(ma,a,t)

                        endif ;

CONSTRAINT:
    identifier    : CustomerDailyBalance
    index domain  : (mr,c,t)
    definition     : sum[ (ma,a) ,
ProductRelation(ma,mr)*transport(ma,a,c,t)]>=sum[r, transport(mr,c,r,t)] ;

CONSTRAINT:
    identifier    : DisassemblyBalance

```

```

        index domain : (mc,r,t)
        definition   : if ord(t)=1 then

                                sum[ (mr,c) ,
ProductRelation(mr,mc)*transport(mr,c,r,t)]=sum[f,transport(mc,r,f,t)]+Disposal(mc,
r,t)+Stocked(mc,r,t)

                                else

                                Stocked(mc,r,t-1)+sum[ (mr,c) ,
ProductRelation(mr,mc)*transport(mr,c,r,t)]=sum[f,transport(mc,r,f,t)]+Disposal(mc,
r,t)+Stocked(mc,r,t)

                                endif ;

CONSTRAINT:
    identifier      : DailyRecovery
    index domain    : (mc,r,t)
    definition       : sum[ (mr,c) , ProductRelation(mr,mc)*Transport(mr,c,r,t)]*(1-
RecoveryTarget)>=Disposal(mc,r,t) ;

CONSTRAINT:
    identifier      : StockLimit
    index domain    : (i,t)
    definition       : sum[m, Stocked(m,i,t)]<=StockedLimit(i) ;

CONSTRAINT:
    identifier      : UpperFlow
    index domain    : (i,j,t)
    definition       : sum[m,
Transport(m,i,j,t)]<=UpperLeavingFlow(i,j)*CalendarParameter(t) ;

CONSTRAINT:
    identifier      : LowerFlow
    index domain    : (i,j,t)
    definition       : sum[m,
Transport(m,i,j,t)]>=LowerLeavingFlow(i,j)*CalendarParameter(t) ;

CONSTRAINT:
    identifier      : UpperSupply
    index domain    : (i,t)
    definition       : sum[m, Supply(m,i,t)]<=UpperSupplyBound(i,t) ;

CONSTRAINT:
    identifier      : LowerSupply
    index domain    : (i,t)
    definition       : sum[m, Supply(m,i,t)]>=LowerSupplyBound(i,t) ;

MATHEMATICAL PROGRAM:
    identifier      : Optimize
    objective       : TotalCost
    direction       : minimize
    constraints      : AllConstraints
    variables       : AllVariables
    type            : Automatic ;

ENDSECTION ;

DECLARATION SECTION User_Interface

PARAMETER:
    identifier      : TotalDemandCustomer
    index domain    : (c,t)
    definition       : sum[m, CustomerDemand(m,c,t)] ;

PARAMETER:
    identifier      : DemandWeek1
    index domain    : (ma,c)
    definition       : sum[t1, CustomerDemand(ma,c,t1)] ;

```

PARAMETER:
 identifier : DemandWeek2
 index domain : (ma,c)
 definition : sum[t2, CustomerDemand(ma,c,t2)] ;

PARAMETER:
 identifier : DemandWeek3
 index domain : (ma,c)
 definition : sum[t3, CustomerDemand(ma,c,t3)] ;

PARAMETER:
 identifier : DemandWeek4
 index domain : (ma,c)
 definition : sum[t4, CustomerDemand(ma,c,t4)] ;

PARAMETER:
 identifier : IncomingWeek1
 index domain : (ma,c)
 definition : sum[t1, IncomingCustomer(ma,c,t1)] ;

PARAMETER:
 identifier : IncomingWeek2
 index domain : (ma,c)
 definition : sum[t2, IncomingCustomer(ma,c,t2)] ;

PARAMETER:
 identifier : IncomingWeek3
 index domain : (ma,c)
 definition : sum[t3, IncomingCustomer(ma,c,t3)] ;

PARAMETER:
 identifier : IncomingWeek4
 index domain : (ma,c)
 definition : sum[t4, IncomingCustomer(ma,c,t4)] ;

PARAMETER:
 identifier : TotalDemandWeek1
 index domain : (c)
 definition : sum[(ma,t1), CustomerDemand(ma,c,t1)] ;

PARAMETER:
 identifier : TotalDemandWeek2
 index domain : (c)
 definition : sum[(ma,t2), CustomerDemand(ma,c,t2)] ;

PARAMETER:
 identifier : TotalDemandWeek3
 index domain : (c)
 definition : sum[(ma,t3), CustomerDemand(ma,c,t3)] ;

PARAMETER:
 identifier : TotalDemandWeek4
 index domain : (c)
 definition : sum[(ma,t4), CustomerDemand(ma,c,t4)] ;

PARAMETER:
 identifier : TotalIncomingWeek1
 index domain : (c)
 definition : sum[(ma,t1), IncomingCustomer(ma,c,t1)] ;

PARAMETER:
 identifier : TotalIncomingWeek2
 index domain : (c)
 definition : sum[(ma,t2), IncomingCustomer(ma,c,t2)] ;

PARAMETER:
 identifier : TotalIncomingWeek3
 index domain : (c)
 definition : sum[(ma,t3), IncomingCustomer(ma,c,t3)] ;

```

PARAMETER:
  identifier   : TotalIncomingWeek4
  index domain : (c)
  definition   : sum[ (ma,t4), IncomingCustomer(ma,c,t4)] ;

PARAMETER:
  identifier   : TotalStocked
  index domain : (i,t)
  definition   : sum[ m, Stocked(m,i,t)] ;

PARAMETER:
  identifier   : TotalSupply
  index domain : (f,t)
  definition   : sum[ mc, Supply(mc,f,t)] ;

PARAMETER:
  identifier   : TotalSupplied
  definition   : sum[ (mc,f,t), Supply(mc,f,t)] ;

PARAMETER:
  identifier   : TotalDisposal
  index domain : (r,t)
  definition   : sum[ mc, Disposal(mc,r,t)] ;

PARAMETER:
  identifier   : TotalTransport
  index domain : (i,j,t)
  definition   : sum[ m, Transport(m,i,j,t)] ;

PARAMETER:
  identifier   : CustomerSatisfactionWeek1
  index domain : (ma,c)
  definition   : sum[ (a,t1), Transport(ma,a,c,t1)] ;

PARAMETER:
  identifier   : CustomerSatisfactionWeek2
  index domain : (ma,c)
  definition   : sum[ (a,t2), Transport(ma,a,c,t2)] ;

PARAMETER:
  identifier   : CustomerSatisfactionWeek3
  index domain : (ma,c)
  definition   : sum[ (a,t3), Transport(ma,a,c,t3)] ;

PARAMETER:
  identifier   : CustomerStasfactionWeek4
  index domain : (ma,c)
  definition   : sum[ (a,t4), Transport(ma,a,c,t4)] ;

PARAMETER:
  identifier   : TotalSatisfactionCustomer
  index domain : c
  definition   : sum[ t, TotalIncomingCustomer(c,t)] ;

PARAMETER:
  identifier   : TotalSatisfaction
  definition   : sum[ (c,t), TotalIncomingCustomer(c,t)] ;

PARAMETER:
  identifier   : TotalDemand
  definition   : sum[ (ma,c,t), CustomerDemand(ma,c,t)] ;

PARAMETER:
  identifier   : RecycleIncominFactory
  index domain : (mc,f,t)
  definition   : sum[ r, transport(mc,r,f,t)] ;

PARAMETER:
  identifier   : TotalRecycleIncomingFactory
  index domain : (f,t)

```

```
definition      : sum[(mc,r), transport(mc,r,f,t)] ;

PARAMETER:
  identifier     : FactoryProduction
  index domain   : (mf,f,t)
  definition     : sum[(mc, ProductRelation(mc,mf) * (sum[r,
transport(mc,r,f,t)]+Supply(mc,f,t)))] ;

PARAMETER:
  identifier     : TotalFactoryProduction
  index domain   : (f,t)
  definition     : sum[(mc,mf), ProductRelation(mc,mf) * (sum[r,
transport(mc,r,f,t)]+Supply(mc,f,t)))] ;

PARAMETER:
  identifier     : LeavingFactory
  index domain   : (mf,f,t)
  definition     : sum[a, Transport(mf,f,a,t)] ;

PARAMETER:
  identifier     : TotalLeavingFactory
  index domain   : (f,t)
  definition     : sum[(mf,a), Transport(mf,f,a,t)] ;

PARAMETER:
  identifier     : FactoryToWarehouseSankey
  definition     : sum[(mf,f,a,t), Transport(mf,f,a,t)] ;

PARAMETER:
  identifier     : WarehouseToCustomerSankey
  definition     : sum[(ma,a,c,t), Transport(ma,a,c,t)] ;

PARAMETER:
  identifier     : CustomerToDisassemblySankey
  definition     : sum[(mr,c,r,t), Transport(mr,c,r,t)] ;

PARAMETER:
  identifier     : NonReturnedCustomerSankey
  definition     : GeneratedCustomers-CustomerToDisassemblySankey ;

PARAMETER:
  identifier     : TotalDisposalSankey
  definition     : sum[(r,t), TotalDisposal(r,t)] ;

PARAMETER:
  identifier     : IncomingWarehouse
  index domain   : (mf,a,t)
  definition     : sum[f, Transport(mf,f,a,t)] ;

PARAMETER:
  identifier     : TotalIncomingWarehouse
  index domain   : (a,t)
  definition     : sum[(mf,f), Transport(mf,f,a,t)] ;

PARAMETER:
  identifier     : LeavingWarehouse
  index domain   : (ma,a,t)
  definition     : sum[c, Transport(ma,a,c,t)] ;

PARAMETER:
  identifier     : TotalLeavingWarehouse
  index domain   : (a,t)
  definition     : sum[(ma,c), Transport(ma,a,c,t)] ;

PARAMETER:
  identifier     : IncomingCustomer
  index domain   : (ma,c,t)
  definition     : sum[a, Transport(ma,a,c,t)] ;

PARAMETER:
```

```

        identifier      : TotalIncomingCustomer
        index domain    : (c,t)
        definition      : sum[ (ma,a), Transport (ma,a,c,t)] ;

PARAMETER:
        identifier      : GeneratedCustomers
        definition      : sum[ (ma,mr,a,c,t),
ProductRelation(ma,mr)*transport (ma,a,c,t)] ;

PARAMETER:
        identifier      : IncomingDisassembly
        index domain    : (mr,r,t)
        definition      : sum[ c, Transport (mr,c,r,t)] ;

PARAMETER:
        identifier      : TotalIncomingDisassembly
        definition      : sum[ (mr,c,r,t), Transport (mr,c,r,t)] ;

PARAMETER:
        identifier      : ReturnToChain
        index domain    : (mc,r,t)
        definition      : sum[ f, Transport (mc,r,f,t)] ;

PARAMETER:
        identifier      : TotalReturnedToChain
        definition      : sum[ (mc,r,f,t), Transport (mc,r,f,t)] ;

PARAMETER:
        identifier      : TotalReturnedToChainFraction
        definition      : if TotalReturnedToChain+TotalSupplied=0 then
                        0
                        else
TotalReturnedToChain/(TotalSupplied+TotalReturnedToChain)*100
                        endif ;

PARAMETER:
        identifier      : TotalReturnToChain
        index domain    : (r,t)
        definition      : sum[ (mc,f), Transport (mc,r,f,t)] ;

PARAMETER:
        identifier      : CostumerRelativeSatisfaction
        index domain    : c
        definition      : if TotalDemand=0 then
                        0
                        else
TotalSatisfactionCustomer(c)/(sum[t,
TotalDemandCustomer(c,t)]*100
                        endif ;

PARAMETER:
        identifier      : RelativeSatisfaction
        definition      : TotalSatisfaction/TotalDemand*100 ;

PARAMETER:
        identifier      : TotalDisposalFraction
        definition      : if (sum[ (mc,r,t), Disposal (mc,r,t)]+sum[ (mc,r,f,t),
Transport (mc,r,f,t)]+sum[ (mc,r,t), Stocked (mc,r,t)])=0 then
                        0
                        else
sum[ (mc,r,t), Disposal (mc,r,t)]/(sum[ (mc,r,t),
Disposal (mc,r,t)]+sum[ (mc,r,f,t), Transport (mc,r,f,t)]+sum[ (mc,r,t),
Stocked (mc,r,t)])*100
                        endif ;

```

```
PARAMETER:
  identifier   : DisposalFraction
  index domain : (mc,r)
  definition   : if (sum[t, Disposal(mc,r,t)]+sum[(f,t),
Transport(mc,r,f,t)]+sum[t, Stocked(mc,r,t)])=0 then

      0

    else

      sum[t, Disposal(mc,r,t)]/(sum[t,
Disposal(mc,r,t)]+sum[(f,t), Transport(mc,r,f,t)]+sum[t, Stocked(mc,r,t)])*100

    endif ;

PARAMETER:
  identifier   : TotalDisposalDFraction
  index domain : (r)
  definition   : if (sum[(mc,t), Disposal(mc,r,t)]+sum[(mc,f,t),
Transport(mc,r,f,t)]+sum[(mc,t), Stocked(mc,r,t)])=0 then

      0

    else

      sum[(mc,t), Disposal(mc,r,t)]/(sum[(mc,t),
Disposal(mc,r,t)]+sum[(mc,f,t), Transport(mc,r,f,t)]+sum[(mc,t),
Stocked(mc,r,t)])*100

    endif ;

PARAMETER:
  identifier   : TotalRecycleFraction
  definition   : if (sum[(mc,r,t), Disposal(mc,r,t)]+sum[(mc,r,f,t),
Transport(mc,r,f,t)]+sum[(mc,r,t), Stocked(mc,r,t)])=0 then

      0

    else

      sum[(mc,r,f,t), Transport(mc,r,f,t)]/(sum[(mc,r,t),
Disposal(mc,r,t)]+sum[(mc,r,f,t), Transport(mc,r,f,t)]+sum[(mc,r,t),
Stocked(mc,r,t)])*100

    endif ;

PARAMETER:
  identifier   : TotalRecycledDFraction
  index domain : r
  definition   : if (sum[(mc,t), Disposal(mc,r,t)]+sum[(mc,f,t),
Transport(mc,r,f,t)]+sum[(mc,t), Stocked(mc,r,t)])=0 then

      0

    else

      sum[(mc,f,t), Transport(mc,r,f,t)]/(sum[(mc,t),
Disposal(mc,r,t)]+sum[(mc,f,t), Transport(mc,r,f,t)]+sum[(mc,t),
Stocked(mc,r,t)])*100

    endif ;

PARAMETER:
  identifier   : RecycleFraction
  index domain : (mc,r)
  definition   : if (sum[t, Disposal(mc,r,t)]+sum[(f,t),
Transport(mc,r,f,t)]+sum[t, Stocked(mc,r,t)])=0 then

      0
```

```

else

    sum[(f,t), Transport(mc,r,f,t)]/(sum[t,
Disposal(mc,r,t)]+sum[(f,t), Transport(mc,r,f,t)]+sum[t, Stocked(mc,r,t)])*100

endif ;

PARAMETER:
    identifier    : TotalStockedFractionDisassembly
    definition    : if (sum[(mc,r,t), Disposal(mc,r,t)]+sum[(mc,r,f,t),
Transport(mc,r,f,t)]+sum[(mc,r,t), Stocked(mc,r,t)])=0 then

        0

    else

        sum[(mc,r,t), Stocked(mc,r,t)]/(sum[(mc,r,t),
Disposal(mc,r,t)]+sum[(mc,r,f,t), Transport(mc,r,f,t)]+sum[(mc,r,t),
Stocked(mc,r,t)])*100

    endif ;

PARAMETER:
    identifier    : StockedFractionDisassembly
    index domain : (mc,r)
    definition    : if (sum[t, Disposal(mc,r,t)]+sum[(f,t),
Transport(mc,r,f,t)]+sum[t, Stocked(mc,r,t)])=0 then

        0

    else

        sum[t, Stocked(mc,r,t)]/(sum[t, Disposal(mc,r,t)]+sum[(f,t),
Transport(mc,r,f,t)]+sum[t, Stocked(mc,r,t)])*100

    endif ;

PARAMETER:
    identifier    : TotalStockedFractionD
    index domain : (r)
    definition    : if (sum[(mc,t), Disposal(mc,r,t)]+sum[(mc,f,t),
Transport(mc,r,f,t)]+sum[(mc,t), Stocked(mc,r,t)])=0 then

        0

    else

        sum[(mc,t), Stocked(mc,r,t)]/(sum[(mc,t),
Disposal(mc,r,t)]+sum[(mc,f,t), Transport(mc,r,f,t)]+sum[(mc,t),
Stocked(mc,r,t)])*100

    endif ;

PARAMETER:
    identifier    : SupplyFraction
    definition    : if (sum[(mc,f,t), Supply(mc,f,t)]+sum[(mc,r,f,t),
Transport(mc,r,f,t)])=0 then

        0

    else

        sum[(mc,f,t), Supply(mc,f,t)]/(sum[(mc,f,t),
Supply(mc,f,t)]+sum[(mc,r,f,t), Transport(mc,r,f,t)])*100

    endif ;

PARAMETER:
    identifier    : ReturnFraction

```



```
        definition      :  if sum[(ma,mr,a,c,t),
ProductRelation(ma,mr)*transport(ma,a,c,t)]=0 then

                                0

                                else

                                sum[(mr,c,r,t), transport(mr,c,r,t)]/sum[(ma,mr,a,c,t),
ProductRelation(ma,mr)*transport(ma,a,c,t)]*100

                                endif ;

ELEMENT PARAMETER:
    identifier      :  DisplayedProduct
    range           :  Products ;

ELEMENT PARAMETER:
    identifier      :  DisplayedFactory
    range           :  Factory ;

ELEMENT PARAMETER:
    identifier      :  DisplayedWarehouse
    range           :  Warehouse ;

ELEMENT PARAMETER:
    identifier      :  DisplayedCustomer
    range           :  Customer ;

ELEMENT PARAMETER:
    identifier      :  DsiplayedDisassembly
    range           :  Disassembly ;

ELEMENT PARAMETER:
    identifier      :  DisplayedProductMc
    range           :  ProductC ;

ELEMENT PARAMETER:
    identifier      :  DisplayedProductMf
    range           :  ProductF ;

ELEMENT PARAMETER:
    identifier      :  DisplayedProductMa
    range           :  ProductA ;

PARAMETER:
    identifier      :  DemandPenaltyFraction
    definition      :  if DemandPenalty=0 then
                        0
                        else
                        DemandPenalty/TotalCost
                        endif ;

PARAMETER:
    identifier      :  DispCostFraction
    definition      :  if DispCost=0 then
                        0
                        else
                        DispCost/TotalCost
                        endif ;

PARAMETER:
    identifier      :  ReturnPenaltyFraction
    definition      :  if ReturnPenalty=0 then
                        0
                        else
                        ReturnPenalty/TotalCost
                        endif ;

PARAMETER:
    identifier      :  StorageCostFraction
```

```

        definition      :  if StorageCost=0 then
                           0
                           else
                           StorageCost/TotalCost
                           endif ;

PARAMETER:
  identifier           :  SupplyCostFraction
  definition           :  if SupplyCost=0 then
                           0
                           else
                           SupplyCost/TotalCost
                           endif ;

PARAMETER:
  identifier           :  TransportCostFraction
  definition           :  if TransportCost=0 then
                           0
                           else
                           TransportCost/TotalCost
                           endif ;

PARAMETER:
  identifier           :  OccupationFactory
  index domain        :  (f)
  definition           :  (sum[t, TotalStocked(f,t)]+sum[(m,a,t),
Transport(m,f,a,t)])/(StockedLimit(f)*28+sum[a, UpperLeavingFlow(f,a)]*28)*100
  comment             :  "Stocked+LeavingFlow / MaxStocked + MaxLeavingFlow" ;

PARAMETER:
  identifier           :  OccupationWarehouse
  index domain        :  (a)
  definition           :  (sum[t, TotalStocked(a,t)]+sum[(m,c,t),
Transport(m,a,c,t)])/(StockedLimit(a)*28+sum[c, UpperLeavingFlow(a,c)]*28)*100 ;

PARAMETER:
  identifier           :  OccupationDisassembly
  index domain        :  (r)
  definition           :  (sum[t, TotalStocked(r,t)]+sum[(m,c,t),
Transport(m,c,r,t)])/(StockedLimit(r)*28+sum[c, UpperLeavingFlow(c,r)]*28)*100 ;

PARAMETER:
  identifier           :  AverageOccFactories
  definition           :  sum[f, OccupationFactory(f)]/2 ;

PARAMETER:
  identifier           :  AverageOccWarehouses
  definition           :  sum[a, OccupationWarehouse(a)]/8 ;

PARAMETER:
  identifier           :  AverageOccDisassemblies
  definition           :  sum[r, OccupationDisassembly(r)]/8 ;

ENDSECTION ;

PROCEDURE
  identifier :  MainInitialization

ENDPROCEDURE ;

PROCEDURE
  identifier :  MainExecution
  body      :
    solve Optimize;

ENDPROCEDURE ;

PROCEDURE
  identifier :  MainTermination
  body      :

```

```
        return DataManagementExit();  
  
    ENDPROCEDURE ;  
  
ENDMODEL Main_TFG ;
```