

Treball de Fi de Màster

Màster Universitari en Enginyeria Industrial (MUEI)

Programació bicriteri de treballs en grups de màquines en paral·lel amb elegibilitat

MEMÒRIA

Autor: Oriol Sentmartí
Director: Manel Mateo
Convocatòria: Maig 2018



Escola Tècnica Superior
d'Enginyeria Industrial de Barcelona



Resum

El problema que tracta el present treball es basa en un sistema productiu format per diferents grups de màquines idèntiques situades en paral·lel. Cada grup de màquines té un nivell associat. Hi ha un conjunt de peces dividides en el mateix nombre de nivells i definides pel seu temps de pre-procés, procés i post-procés. Cada màquina podrà processar les peces de nivell igual o inferior al seu. El fet de programar operacions en nivells cada vegada inferiors fa que hi hagi unes penalitzacions superiors.

L'objectiu del projecte és trobar un seguit de programacions de les peces a les màquines que permetin minimitzar l'instant de finalització del procés (F_{MAX}) i alhora minimitzar, també, la penalització (c) associada al cost d'utilitzar màquines menys eficients.

El problema plantejat serà resolt a partir d'un seguit d'algorismes que permetran obtenir un conjunt de solucions (programacions de les peces en les màquines) no dominades. Tots els procediments de resolució estan formats per una Fase 0, que permet obtenir una seqüència inicial i una Fase 1, on es busquen noves solucions mitjançant un procediment arborescent. A més, en una Fase 2, es realitza una cerca més extensa de solucions.

Com a resultat d'aquests procediments proposats, s'obtenen diferents solucions que presenten seqüències per a cada una de les màquines amb els seus respectius valors de c i F_{MAX} . D'aquesta manera es permet, al responsable de planificació, decidir quina d'aquestes programacions és la més indicada per als interessos de l'empresa, en temps o penalització.

Per comprovar el correcte funcionament dels procediments proposats i comparar-los entre ells, en el treball es presenta una experiència computacional on es resolen exemplars de fins a 200 peces amb una sèrie de combinacions de nombre de nivells i nombre de màquines per nivell. S'analitza el nombre mitjà de solucions no dominades trobades, el temps de resolució de cada exemplar i es comparen els algorismes dos a dos.

Sumari

Resum	3
Sumari.....	5
1. Notació	10
1.1. Dades	10
1.2. Resolució	11
2. Prefaci.....	13
2.1. Origen del projecte.....	13
2.2. Motivació	13
2.3. Punt de partida	13
3. Introducció	14
3.1. Objectius del projecte	14
3.2. Abast del projecte	15
3.3. Estructura del projecte	15
4. Problemàtica en una empresa real.....	17
4.1. Descripció de l'empresa.....	17
4.2. Descripció del procés.....	17
4.3. Cas estudiat	18
5. Direcció i programació d'operacions	19
5.1. Direcció d'operacions	19
5.2. Programació d'operacions	20
5.2.1. Característiques generals	20
5.2.2. Entorn físic	21
6. Problema de programació.....	23
6.1. Definició del problema del taller mecànic.....	23
6.2. Classificació d'un problema de programació	23

6 | Programació bicriteri de treballs en grups de màquines en paral·lel amb elegibilitat

6.2.1.	Problema estàtic	24
6.2.2.	Problema semidinàmic	24
6.2.3.	Problema dinàmic	24
6.3.	Hipòtesis del problema	25
6.4.	Nomenclatura	25
7.	Formulació del problema	27
7.1.	Hipòtesis del problema	27
7.2.	Definició dels elements	27
7.2.1.	Peces.....	27
7.2.2.	Exemple de definició de peces (exemple 1).....	29
7.2.3.	Màquines en paral·lel.....	29
7.2.4.	Exemple de definició de màquines (exemple 1)	30
7.3.	Compatibilitat entre les peces i les màquines.....	30
7.3.1.	Programació	31
7.3.2.	Exemple de programació de les peces a les màquines (exemple 1).....	32
7.4.	Índexs d'eficiència	32
7.4.1.	Exemple d'índexs d'eficiència (exemple 1).....	34
7.5.	Formulació del problema.....	34
7.6.	Solució del problema.....	35
8.	Optimització bicriteri.....	36
8.1.	Formulació multicriteri.....	36
8.2.	Espai de solucions i no dominància de solucions.....	37
8.3.	Presa de decisions bicriteri	38
8.3.1.	Càlcul de la no dominància de solucions	39
8.3.2.	Comparació entre dos procediments	39
8.3.3.	Exemple de comparació entre dues heurístiques	40

9.	Procediment de resolució	42
9.1.	Objectiu.....	42
9.2.	Esquema general	42
9.3.	Fase 0: seqüència inicial de l'estructura arborescent ($c = 0$)	46
9.4.	Fase 1: procediment d'exploració arborescent	52
9.4.1.	Estructura del procediment arborescent	52
9.4.2.	Elecció de la màquina origen per obtenir una nova solució.....	54
9.4.3.	Elecció de la peça candidata a canviar de nivell	55
9.4.4.	Procediment de ramificació per obtenir una nova solució	56
9.4.5.	Exemple de la Fase 1	57
9.4.6.	Procediment heurístic per resoldre el cas de $JC \equiv 0$ a la màquina origen (mo) .	59
9.5.	Fase 2: millora de solucions.....	60
9.6.	Algorismes utilitzats	61
9.6.1.	Resum algorismes.....	62
9.6.2.	Algorisme de partida 1A i 1B.....	62
9.6.3.	Algorismes 2A i 2B	64
10.	Experiència computacional.....	68
10.1.	Generació dels exemplars de peces i màquines	68
10.1.1.	Combinacions de màquines	68
10.1.2.	Generació dels exemplars de peces.....	69
10.2.	Estudi dels exemplars.....	70
10.2.1.	Estudi dels exemplars de 20 peces.....	71
10.2.2.	Estudi dels exemplars de 50 peces.....	72
10.2.3.	Estudi dels exemplars de 100 peces.....	74
10.2.4.	Estudi dels exemplars de 200 peces.....	78
11.	Comparativa dels algorismes.....	83

8 | Programació bicriteri de treballs en grups de màquines en paral·lel amb elegibilitat

12.	Pressupost.....	85
13.	Impacte ambiental.....	87
14.	Conclusions.....	88
15.	Agraïments.....	91
16.	Bibliografia.....	92
17.	Bibliografia complementària.....	93

1. Notació

1.1.Dades

g	Nombre total de grups o nivells
m	Nombre total de màquines
n	Nombre total de peces a programar
<hr/>	
h	Índex de les màquines dins de cada nivell
i	Índex de referència de les màquines, $i = \{1, \dots, m\}$
j	Índex de referència de les peces, $j = \{1, \dots, n\}$
k	Índex de cada nivell, $k = \{1, \dots, g\}$
<hr/>	
c_k	Penalització per passar d'assignar una peça del nivell k al nivell $k + 1$ ($c_1 = c_2 = c_3 = \dots = c$)
l_j	Nivell de la peça j
l_{m_i}	Nivell de la màquina
MG_k	Vector dels índexs de les màquines del nivell k
m_k	Nombre de màquines al nivell k
n_k	Nombre de peces al nivell k , amb $k = \{1, \dots, g\}$
$n_{k,h}$	Índex de la peça h del nivell k
p_j	Temps de procés de la peça j , amb $j = \{1, \dots, n\}$
q_j	Temps de postprocés de la peça j , amb $j = \{1, \dots, n\}$
r_j	Temps de preprocés de la peça j , amb $j = \{1, \dots, n\}$

1.2. Resolució

$\overline{F_{MAX}}$	Vector que conté els instants globals de finalització de cada màquina $F_{k,h}$
J_c^{ms}	Conté una de les millors solucions J^s trobades per una p determinada
AJ_k :	Conjunt de peces assignables al nivell k
B	Pila de solucions a partir de les quals s'iniciarà el procés de <i>backtracking</i> de la Fase 2
c^s	Penalització d'una solució s
$f_{[j]}$	Instant de finalització de la peça que ocupa la posició $[j]$ a la seqüència $J_{k,h}$ de la màquina on està assignada
$F_{k,h}$	Vector instant de finalització (incloent temps de postprocés) de les peces programades a màquina h del nivell k : $F_{k,h} = \max_{j \in J_{k,h}} [f[j]]$; $k = 1, \dots, g$; $h = 1, \dots, m_k$
F_{MAX}	Instant global de finalització, que indica l'instant de finalització de la última peça present en el sistema: $F_{MAX} = \max_k [F_{MAX k}]$, $k = 1, \dots, g$
$F_{MAX k}$	Instant de finalització del nivell. Instant en que l'última peça programada al nivell k acaba el procés: $F_{MAX k} = \max_{m_k, h \in MG_k} [F_{k,h}]$, $k = 1, \dots, g$
J	Matriu de seqüència de les n peces distribuïdes per les m màquines
J^s	Matriu de seqüència de les n peces distribuïdes per les m màquines de la solució s
jc	Peça j candidata a canviar d'un nivell superior a un nivell immediatament inferior, sempre que el nivell de la peça sigui igual o inferior al de la màquina destí i que la peça no tingui el moviment restringit
JC	Conjunt de peces candidates a canviar de nivell, $jc \in JC$
$J_{k,h}$	Seqüència de peces assignades a la màquina k del nivell h

jm	Millor peça candidata a canviar de nivell superior a nivell immediatament inferior ja que produeix una major disminució de F_{MAX} , $jm \in JC$
JM_{k,h}:	Índexs de peces assignades a la màquina h del nivell k
kr	Nivell de referència. Nivell que conté la màquina origen
mo	Màquina origen. Màquina que conté la peça jm
ms	Índex de referència de les millors solucions per cada penalització c
MS	Llista de les solucions J_c^{ms} no dominades, $MS \subset S$
ns	Número total de solucions
s	Solució que conté $\overline{F_{MAX}}$, c , J_{sind}
S	Llista de solucions $J_s \rightarrow S = \{s_0, \dots, s_{ns}\}$
sind	Índex de referència de les solucions, $sind = \{1, \dots, ns\}$
SJ_k:	Índexs de les peces assignades al nivell k
t_[j]	Instant de sortida de la peça que ocupa la posició $[j]$ a la seqüència $J_{k,h}$ de la màquina on està assignada
T_{k,h}	Vector instant de sortida de la màquina de les peces per la màquina h del nivell k

2. Prefaci

2.1. Origen del projecte

La idea de realitzar aquest projecte va sorgir de la valoració de les diferents propostes que el professor i director del projecte Manel Mateo em va formular durant una trobada per escollir una temàtica per al meu TFM.

Després d'estudiar diferents propostes em vaig decantar per aquesta temàtica perquè vaig considerar que era interessant, innovadora i útil.

És per aquest motiu que, finalment, vam acordar tirar endavant amb aquest projecte.

2.2. Motivació

La principal motivació per la realització d'aquest projecte ha estat que es tractava d'un estudi o situació que no havia estat resolta, pel que nosaltres sabem, encara per ningú. És a dir, es tracta d'un projecte que pot servir de punt de partida per altres projectes i investigacions.

D'altra banda, el fet de treballar amb el llenguatge de programació C# i treballar amb la plataforma Visual Studio representava un repte per mi, ja que eren entorns totalment nous. Això comportava un plus de motivació ja que en cas d'èxit representaria un gran aprenentatge i una ampliació dels meus horitzons en el camp de la programació.

Finalment, aquest projecte està emmarcat dins del camp de l'organització industrial, la meua branca dins l'enginyeria industrial. Fet que fa que sigui un camp que m'agrada i em motiva de per si.

2.3. Punt de partida

El punt de partida són els projectes de final de carrera Programació bicriteri de peces en màquines en paral·lel considerant la seva especialització de Jordi Camps (Camps, 2013) i Programació de peces a múltiples nivells en un taller de línies de premses de Xavier Garriga (Garriga, 2009). Concretament, el treball és una evolució del projecte de Jordi Camps, ja que ell considerava una sola màquina per nivell (un cas particular) i en aquest cas es considera més d'una màquina per nivell (un cas més genèric).

3. Introducció

Actualment en el sector de la indústria cal produir una gran varietat de productes amb un ampli rang de requisits concrets de la forma més eficient possible per satisfer les necessitats dels clients i oferir un millor producte que els competidors. Per aconseguir la màxima eficiència no n'hi ha prou amb tenir tots els recursos, també s'han de saber com gestionar de manera eficient per tal de treure el millor rendiment de cada un d'ells. Un factor clau és la programació d'operacions, que permet definir en quin instant de temps s'ha d'utilitzar cada recurs disponible per dur a terme cada una de les operacions i utilitzant els algorismes adequats es pot fer de manera més eficient.

En moltes empreses la programació es resol de forma manual a partir de l'experiència dels treballadors. Malgrat això, aquest mètode no garanteix la qualitat de la solució escollida. D'aquí sorgeix la necessitat de trobar eines que permetin resoldre aquesta problemàtica proposant solucions que siguin de qualitat.

3.1. Objectius del projecte

L'objectiu del present projecte és definir, desenvolupar i analitzar un procediment que sigui capaç de programar un conjunt de n peces classificades en g nivells, en un conjunt de m màquines idèntiques en paral·lel distribuïdes en g grups en paral·lel i amb cada grup associat a un nivell.

Cada peça i cada màquina estarà definida pel seu nivell, que determinarà a quines màquines es pot processar la peça corresponent. Es considera que el coll d'ampolla del sistema productiu, que cal programar de la millor manera està a la part central d'aquest procés. Això vol dir que s'hauran de realitzar algunes operacions abans i unes altres després. Per aquest motiu, cada peça j tindrà associat un temps de pre-procés r_j corresponent als tractaments previs que pugui rebre, un temps de procés p_j corresponent al temps necessari per al tractament a la màquina (coll d'ampolla) i un temps de post-procés q_j corresponent al temps necessari per els tractaments finals.

L'objectiu és minimitzar dos índexs d'eficiència proposats: un relacionat amb el temps de finalització de tot el conjunt de peces i l'altre amb l'assignació de les peces a les màquines. En concret es buscarà minimitzar l'instant global de finalització de tot el conjunt de peces (F_{MAX}) i la penalització global per canvi de nivell (c) associada al moviment de peces de nivells

superiors a nivells inferiors. S'entén com a màquines de nivell superior aquelles màquines preferibles per fer una operació. Per tant, es tracta d'un problema bicriteri on, a partir d'aquests dos índexs, s'avaluaran les solucions trobades per tal de determinar les més adequades

Es busca obtenir un conjunt de solucions, és a dir, un conjunt de seqüències per cada màquina amb un valor de F_{MAX} i c associat, per tal de que l'encarregat de decidir la programació de les peces tingui una eina que li faciliti la decisió d'escollir, segons les necessitats del moment, la programació de les peces en les màquines que li sembli més adequada.

Un cop dissenyades les heurístiques s'analitzaran les solucions obtingudes a partir de la resolució dels exemplars de peces creats per simular una situació real de programació. En l'anàlisi comparativa s'estudiarà, per una banda, el nombre mitjà de solucions no dominades i el temps mitjà de resolució. Per l'altra, es compararan les solucions no dominades obtingudes per determinar quin algorisme permet obtenir un major nombre de solucions de qualitat.

3.2. Abast del projecte

Es busca resoldre el problema plantejat en l'apartat anterior amb un nombre total de m màquines, cada una associada a un nivell determinat dels g nivells totals i un nombre màxim i concret de n peces, dividides en els mateixos nivells.

En aquest cas es consideraran totes les màquines idèntiques, totes tenen el mateix temps de procés. D'altra banda, el nombre de màquines per nivell del cas estudiat pot ser una o més.

Només s'abordarà la situació en què la penalització per fabricar una peça en un nivell inferior és constant per a qualsevol canvi de nivell. Es deixa per a un treball posterior introduir la variabilitat en aquesta dada.

3.3. Estructura del projecte

El cos del projecte està estructurat en 4 parts. La primera part tracta de l'aplicació del problema en una situació real d'una empresa, la introducció als conceptes previs i la contextualització dels projecte en una empresa real i dins de l'àmbit de la direcció d'operacions. Aquesta primera part està formada pels **capítol 4**, **capítol 5** i el **capítol 6**.

La segona part conté la descripció i formulació del problema plantejat, explica els conceptes teòrics que engloben el desenvolupament de l'eina i es defineix la metodologia de resolució genèrica utilitzada. Aquesta part està formada pels **capítol 7**, el **capítol 8** i el **capítol 9**.

A la tercera part, es presenten el grup d'heurístiques proposades per a la resolució del cas, així com l'experiència computacional i anàlisi dels resultats corresponents. Aquesta part engloba el **capítol 10** i el **capítol 11**.

Finalment, a la quarta i darrera part, apareixen el pressupost del projecte, l'estudi d'impacte ambiental i les conclusions obtingudes a partir de la realització d'aquest projecte. Aquesta part engloba el **capítol 12**, el **capítol 13** i el **capítol 14**.

D'altra banda també, s'han realitzat un conjunt d'annexos on es pot trobar informació que ajuda a la comprensió del lector. Aquests annexos contenen informació de les aplicacions i de les dades amb les quals s'ha treballat.

4. Problemàtica en una empresa real

Per tal d'il·lustrar la problemàtica que es vol resoldre en el present projecte s'ha agafat un exemple d'una empresa real. Malgrat això la impossibilitat d'obtenir dades no permetrà realitzar un estudi aplicat.

4.1.Descripció de l'empresa

L'empresa fabrica fregones de cordons, de microfibras i de cotó. A partir de bobines de fils, es teixeixen els cordons que posteriorment s'utilitzaran per fabricar les esmentades fregones. Aquests cordons són introduïts en una màquina que els estira, talla i ajunta amb el capçal (coll d'ampolla). Finalment les fregones són embossades, encaixades i paletitzades mitjançant un procés automatitzat.

4.2.Descripció del procés

Es dividirà la cadena productiva que s'ha explicat en l'apartat 4.1 en 5 etapes tal com es representa a la Figura 4.1.

- **Etapa 1:** Consisteix en la preparació de la matèria prima i engloba el desplaçament d'aquesta matèria des del magatzem a la màquina teixidora dels cordons i la col·locació dels fils a la mateixa màquina.
- **Etapa 2:** Consisteix en el procés de teixit dels cordons que realitza la màquina teixidora.
- **Etapa 3:** Es tracta del procés de col·locació de les bobines de cordons que s'extreuen de la màquina teixidora a la màquina de fabricació de les fregones.
- **Etapa 4:** Es tracta del procés de tallat dels cordons i col·locació del capçal de les fregones. És a dir el procés d'assemblatge de les fregones. Aquesta àrea és el coll d'ampolla del procés de fabricació.
- **Etapa 5:** Procés d'embossat, encaixat i paletitzat.

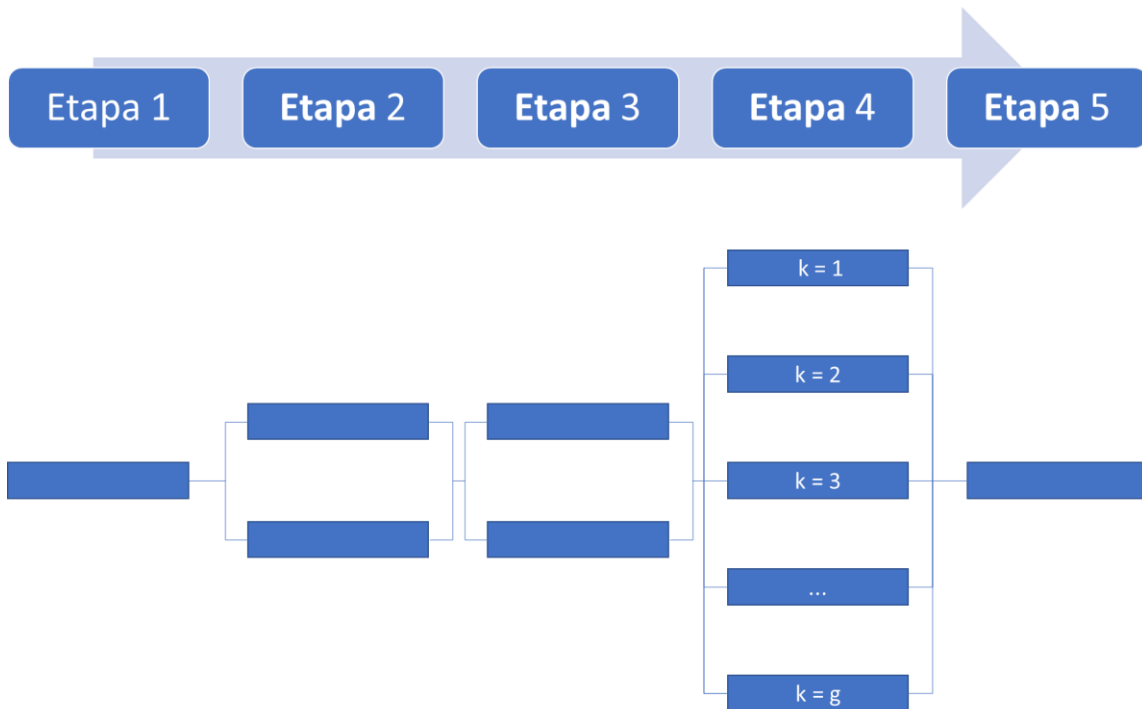


Figura 4.1 Representació de les etapes de la cadena productiva

Els quadres de l'Etapa 4 representats a la Figura 4.1 representen grups de màquines en paral·lel. Cada grup de màquines té unes característiques diferents que fan que siguin més o menys eficients energèticament segons el nombre del grup, essent 1 la més eficient i per tant la desitjada des del punt de vista de consum. Malgrat això els temps de pas de les peces (fregones) per cadascuna de les màquines, independentment del grup, són els mateixos.

4.3.Cas estudiat

El coll d'ampolla del procés és a l'Etapa 4. Per aquest motiu, es farà l'estudi del treball d'un cas extrapolable a aquest. Tot i que la falta de dades no permet realitzar un estudi aplicat, s'utilitzarà el concepte de diferents grups de màquines de diferents nivells per poder trobar un algorisme que pugui resoldre situacions com la presentada. El pre-procés estarà format per les primeres etapes i el post-procés serà la darrera de les operacions a realitzar.

5. Direcció i programació d'operacions

5.1. Direcció d'operacions

Dins l'àmbit empresarial existeixen múltiples processos productius que engloben una gran complexitat i diversitat de tasques. Cada tasca pot tenir unes característiques independents i pot estar dirigida per un equip de treball concret. A més, pot tenir unes restriccions tècniques específiques. En un entorn productiu, la direcció d'operacions és un camp que permet gestionar i coordinar totes aquestes tasques per tal de millorar-ne l'eficiència i fer-les més competitives respecte els competidors. (Departament d'Organització d'Empreses ETS d'Enginyeria Industrial de Barcelona, Universitat Politècnica de Catalunya, 2016)

Per això, la direcció d'operacions suposa la coordinació, la gestió i la millora del dia a dia del sistema productiu per tal de convertir els factors de producció en productes de la manera més eficient possible i maximitzant el benefici de l'organització.

Tota organització té unes estratègies que fixen els objectius i que marquen el seu funcionament. Per tal de complir els objectius, s'han d'anar prenent en seguit de decisions que en el cas de la direcció d'operacions es poden dividir en tres nivells:

- **Decisions estratègiques o planificació d'operacions**

Aquestes decisions són decisions que afecten a llarg termini l'eficàcia de l'empresa o de la cadena de subministrament i fan referència a com s'han d'atendre les necessitats dels clients. Aquestes decisions han d'estar alineades amb l'estratègia corporativa de l'empresa.

Les decisions preses en el nivell estratègic fixen les condicions i/o restringeixen les operacions sota les quals l'empresa haurà de funcionar tant a mitjà com a curt termini.

- **Decisions tàctiques o càlcul de necessitats**

Aquestes decisions es prenen a mitjà termini. Consisteixen, principalment, en preocupar-se de disposar de manera eficient dels recursos del treball i els materials dins l'àmbit restringit per les decisions estratègiques preses anteriorment.

A més, les decisions tàctiques es converteixen en restriccions de les operacions que es desenvoluparan en la planificació operativa i el control. Aquestes decisions tàctiques es tradueixen en el càlcul de necessitats, el qual estableix els ordres d'aprovisionament a realitzar

per complir amb els objectius marcats en la planificació d'operacions i disposar dels recursos suficients en el moment en què s'hagin programat les operacions.

- **Decisions de programació i control d'operacions o programació d'operacions**

Aquestes decisions són preses a curt termini i estan relacionades amb les activitats del dia a dia de l'empresa. Són decisions que busquen definir de forma concreta en quins recursos s'han d'executar les operacions fixades i en quin instant s'han d'executar.

5.2. Programació d'operacions

5.2.1. Característiques generals

La programació d'operacions té per objectiu definir de forma concreta amb quin dels recursos disponibles i en quin instant de temps s'executarà cadascuna de les operacions necessàries per a la realització de les peces i l'instant en què s'ha d'executar cada una d'aquestes operacions. Queda determinada per tres sub-funcions (Companys, 2003):

- La **càrrega**: Assigna a cada operació el recurs on s'aplicarà. Té els següents elements:
 - n peces → com a peça es pot entendre també un lot, una ordre de treball, etc.
 - m màquines → les encarregades de fer les operacions sobre les peces
 - La realització de cada peça requereix l'execució d'una sèrie d'operacions prefixades en un ordre establert
- La **seqüenciació**: Ordre de les operacions assignades a un recurs
- La **temporització**: Instants d'execució de cada operació. S'ha de tenir en compte:
 - La duració de cada operació és coneguda
 - S'ha de buscar un programa que busqui optimitzar un índex d'eficiència

La programació del conjunt d'operacions que el sistema productiu haurà de realitzar es coneix com a programa. Per tant, un programa es pot definir com la temporització d'una seqüència carregada i es considera que és eficaç quan aconsegueix una bona interrelació entre les anteriors funcions.

Per tal d'avaluar els programes existeixen diferents índexs d'eficiència que permeten escollir-ne un en funció dels objectius plantejats. Sovint, és necessari satisfer més d'un objectiu al mateix moment, essent necessari avaluar més d'un criteri alhora. Uns exemples d'índexs d'eficiència podrien ser:

- Minimitzar el temps màxim de permanència dels productes a les instal·lacions
- Minimitzar els costos de producció dels productes
- Minimitzar l'instant global de finalització del procés
- Minimitzar els temps morts en el procés productiu
- Maximitzar la qualitat de les peces fabricades

5.2.2. Entorn físic

La programació d'operacions estarà afectada per l'entorn físic real on s'hagi de realitzar, és a dir, per la classe i la distribució de les màquines que hi hagi al sistema. Per una banda, tal i com es pot observar en la Figura 5.1 les màquines, en el cas de fluxos regulars on la producció avança en un sentit, es poden distribuir de tres maneres diferents:

- En **sèrie**: Les màquines se situen una després de l'altre perquè s'utilitzen per a diferents operacions consecutives, seguint l'ordre d'aquestes. En aquest cas per les 3 operacions hi ha les màquines *M1*, *M2* i *M3*.
- En **paral·lel**: Són màquines que poden realitzar una mateixa operació i per tant, cal escollir una entre les possibles. En aquesta ocasió, les màquines *M1*, *M2* i *M3* poden fer l'operació necessària.
- **Híbrida**: és la combinació de les dues distribucions anteriors. En aquest exemple primer cal escollir entre les màquines *M1*, *M2* i *M3* per fer la primera operació i la segona operació la realitza la màquina *M4*.

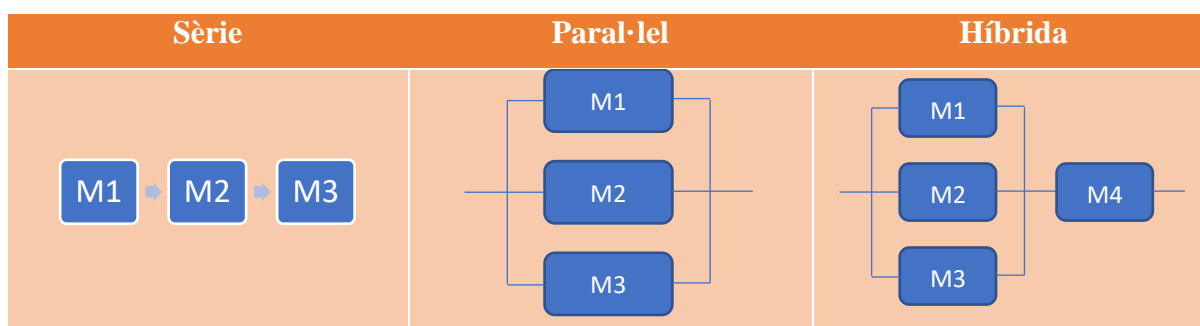


Figura 5.1 Representació de les possibles distribucions de les màquines

D'altra banda, segons la tipologia de les màquines en paral·lel es poden agrupar segons:

- **Màquines idèntiques**: el temps de procés d'una peça és el mateix en qualsevol màquina on sigui tractada.
- **Màquines uniformes**: són màquines que poden produir les peces a diferents velocitats i que, per tant, tindran un temps de procés proporcional a la seva velocitat d'operació.

22 | Programació bicriteri de treballs en grups de màquines en paral·lel amb elegibilitat

- **Màquines independents:** en aquestes màquines el temps de procés d'una peça pot ser completament arbitrari.

En el cas tractat, totes les màquines són idèntiques.

6. Problema de programació

El present treball es basa en resoldre una variant del problema del taller mecànic, que consisteix en un prototip de problema de programació. En aquest capítol s'expliquen les bases i característiques d'aquest problema per tal de poder definir i concretar, en el capítol següent, el problema estudiat.

6.1. Definició del problema del taller mecànic.

L'enunciat genèric del problema del taller mecànic és (Companys, 2003):

Es tenen **n peces** que s'han de realitzar en **m màquines**. El processament de cada peça comporta l'execució, en un ordre preestablert, d'una sèrie d'operacions prefixades. Cada operació està assignada a una de les **m màquines** i està definida per una duració determinada i coneguda. Per resoldre el problema s'ha d'establir un programa, és a dir, trobar una seqüència d'operacions per cada màquina i trobar l'interval temporal d'execució de les operacions. L'objectiu serà optimitzar un índex determinat que mesuri l'eficiència del programa.

És interessant remarcar que dins la programació d'operacions s'utilitza habitualment els termes peces i màquines per definir el procés productiu. Malgrat aquest fet, els problemes que es poden estudiar poden ser molt diversos i englobar molts camps. Principalment només s'ha d'extrapol·lar aquests dos conceptes al que es vol estudiar, per exemple:

- Peces: lots de producció, persones, ordres de treball, etc.
- Màquines: recursos, seccions, treballadors, llocs de treball, etc.

6.2. Classificació d'un problema de programació

El problema del taller mecànic es pot classificar entres categories segons les característiques de la programació que s'utilitza per resoldre'l:

- Estàtic
- Semidinàmic
- Dinàmic

6.2.1. Problema estàtic

Un problema estàtic es caracteritza per:

- Un nombre finit i determinat de peces s'han de realitzar amb un nombre finit de màquines.
- Al realitzar la programació es coneix la ruta de cada peça, les operacions que la componen, la màquina on es realitza cada operació i la seva duració.
- Totes les peces i màquines estan disponibles en el mateix instant de temps (generalment $t = 0$).
- L'objectiu del problema és buscar un programa que optimitzi un índex d'eficiència establert.

6.2.2. Problema semidinàmic

Les característiques d'un problema semidinàmic són gairebé calcades a les d'un problema estàtic. L'única diferència correspon a l'instant de disponibilitat de les peces i les màquines. En el cas d'un problema semidinàmic, a diferència del problema estàtic, els instants de disponibilitat de les peces i les màquines poden no ser idèntics. No obstant, aquests sí que són coneguts al moment de realitzar la programació.

6.2.3. Problema dinàmic

Les característiques d'un problema dinàmic són:

- L'horitzó de funcionament és il·limitat.
- El nombre de peces és il·limitat. Totes les peces a tractar no estan definides en un moment determinat, la definició de les peces es coneix progressivament a mesura que transcorre el temps.
- El nombre de màquines no és il·limitat.
- Una peça queda completament definida quan l'ordre arriba al taller
- Les peces són processades i abandonen el taller progressivament, mentre són substituïdes per altres peces que arriben al taller per ser processades.
- Un únic programa no és suficient, s'han d'establir cicles de reprogramació que vagin actualitzant l'entrada i la sortida de peces.
- La finalitat és establir un procediment de programació i jutjar la seva qualitat. En aquest cas, els índexs d'eficiència van associats a característiques dels programes de valors mitjans al llarg d'un interval de temps suficientment gran.

6.3. Hipòtesis del problema

Conway, Maxwell i Miller (Conway, William, & Miller, 1967) van establir unes hipòtesis per estudiar el problema del taller mecànic que generalment es prenen com a punt de partida. Les que s'han considerat en el present treball són:

- Cada màquina està disponible contínuament a partir d'un instant $t \geq 0$ fins a un instant arbitràriament gran T . No existeixen intervals de no disponibilitat per averies o per manteniment programat.
- Cada operació pot executar-se en un sol tipus de màquina (en aquest cas totes les màquines es consideren d'un sol tipus)
- Quan una operació ha començat no s'admeten interrupcions.
- No poden solapar-se dues operacions de la mateixa peça (en una mateixa màquina o en màquines diferents)
- Cada màquina només pot executar una operació a la vegada.
- L'única restricció activa en el taller és relativa a les màquines.

6.4. Nomenclatura

Els tipus de problemes són molt diversos i cada un té unes característiques diferents. Per aquest motiu s'han definit diferents nomenclatures per tal de facilitar la seva entesa i identificació. La nomenclatura més usada és la de Lawler (Lawler, Lenstra, Rinnooy, & Shmoys, 1993), que defineix un problema en tres símbols $\alpha|\beta|\gamma$, el significat dels quals és:

- α : especifica l'entorn de les màquines i només pot contenir una única inscripció.

Algunes de les opcions són:

- **1** una sola màquina
- **P_m** m màquines idèntiques en paral·lel
- **Q_m** m màquines en paral·lel amb diferent velocitats
- **R_m** m màquines en paral·lel amb durades no relacionades o independents
- ...
- β : defineix les característiques i restriccions dels processos, i en pot contenir varies, una o cap inscripció de les següents:
 - **d_j** dates de lliurament de les peces diferents a zero
 - **p** tots els temps de procés són iguals
 - **r_j** dates de disponibilitat de les peces diferents a zero

- q_j temps de cua o post-procés de les peces diferent a zero
 - M_j subconjunt de les m màquines on es podrà produir una peça
 - ...
- γ : indica l'índex d'eficiència que es vol avaluar, minimitzar o maximitzar, i pot contenir una o varies inscripcions de les següents:
- C_{\max} instant màxim d'ocupació de tota la instal·lació
 - F_{med} temps mitjà d'ocupació de tota la instal·lació
 - T_{\max} retard màxim en el lliurament de les peces
 - ...

7. Formulació del problema

El present projecte pretén resoldre un problema semidinàmic realitzant una programació pel cas d'un sistema productiu format per a un nombre g de grups ($k = 1, \dots, g$) entre les m (màquines en paral·lel $i = 1, \dots, m$), per on s'espera que passi un nombre finit de n peces o elements a processar.

7.1. Hipòtesis del problema

Per tal de resoldre el problema plantejat s'han considerat diferents hipòtesis, prenent algunes de les de Conway, Maxwell i Miller (Conway, William, & Miller, 1967) com a punt de partida i afegint-ne algunes més. Les hipòtesis considerades són:

1. Cada màquina està disponible contínuament des d'un instant $t \geq 0$, fins a un instant arbitràriament gran, T .
2. El nombre de nivells (g) és inicialment conegut.
3. Cada màquina està classificada en un dels g nivells.
4. Cadascuna de les peces està classificada en un dels g nivells.
5. El nombre de peces, màquines, nivells i màquines en cada nivell és inicialment conegut.
6. Cada peça només rep una operació. És a dir, només ha de passar per una de les màquines d'un dels nivells per quedar completada l'operació.
7. Totes les màquines tenen la mateixa velocitat. Per tant, tots els nivells tenen el mateix temps d'execució en cada operació.
8. Els temps de pre-procés, procés i post-procés són valors enters i coneguts en l'instant de realitzar la programació.
9. Cada peça es comença a processar en un instant de temps igual o superior al seu temps de pre-procés.
10. Quan una operació ha començat no s'admeten interrupcions.
11. Cada màquina només pot realitzar una operació alhora.

7.2. Definició dels elements

7.2.1. Peces

El concepte anomenat peça es fa servir per englobar qualsevol part d'un procés que ha de ser processat per una màquina, estació o un conjunt de màquines o estacions. Cada una de les peces j ($j = 1, \dots, n$) queda definida dins del procés productiu per 3 temps: un temps per al pre-procés,

un temps per al propi procés de la màquina i un altre temps per al post-procés. Aquests temps es defineixen com:

- **Temps de pre-procés (r_j):** És el temps de qualsevol activitat prèvia al procés però necessària per la realització d'aquest. Les activitats que conformen el pre-procés d'una peça es poden realitzar simultàniament en totes les peces.
- **Temps de procés (p_j):** És el temps propi d'operació. El temps que triga la peça j a passar per la màquina $m_{k,h}$. Els temps de procés d'una peça són iguals per a qualsevol de les màquines.
- **Temps de post-procés (q_j):** És el temps posterior a l'operació. El temps necessari per treure les peces de la línia de producció, controls de qualitat, embalatge, neteja...
- **Nivell de la peça (l_j):** nivell de la peça j , $l_j \in \{1, \dots, g\}$

Les peces es classifiquen per nivells. Això és degut a les seves característiques físiques i del procés de fabricació que requereixin. Per tant, les peces es podran processar en les màquines segons la relació entre els seus nivells. Les peces es classifiquen en g nivells ($l_i = 1 \dots g$). Aquestes només poden ser processades per màquines del seu nivell o nivells superiors. El nombre total de peces n és la suma de les peces dels diferents grups:

$$\sum_{k=1}^g n_k = n \quad (\text{eq. 7.1})$$

El nombre total de peces n_k contingudes en cada grup és la suma de les peces, el grup de les quals és k .

Dins de cada grup una peça, l' h -èssima peça del grup k es definirà com $n_{k,h}$.

Tal i com es pot veure a la Taula 7.1 les peces queden distribuïdes de la següent forma:

l_j	Peces	Característiques
1	n_1	Només les poden processar les màquines de nivell 1
$\forall k$	n_k	Les poden processar totes les màquines de nivell igual o superior a k
g	n_g	Les poden processar totes les màquines

Taula 7.1 Nivells dels grups de peces

7.2.2. Exemple de definició de peces (exemple 1)

A la Taula 7.2 es mostra un exemple de dades (exemple 1) que defineixen un entorn de $n = 5$ peces, amb els seus nivells, temps de pre-procés, procés i post procés.

	1	2	3	4	5
Nivell peces (l_j)	1	1	2	2	3
Temps de pre-procés (r_j)	1	2	3	2	3
Temps de procés (p_j)	5	3	5	6	4
Temps de post-procés (q_j)	3	1	2	4	1

Taula 7.2 Definició de l'entorn de les peces de l'exemple 1

De les 5 peces presentades, dues són del nivell 1, dues del nivell 2 i una del nivell 3. És a dir:

- $n = 5$
- $n_1 = 2; n_2 = 2; n_3 = 1$
- $n_{1,1} = 1; n_{1,2} = 2; n_{2,1} = 3; n_{2,2} = 4; n_{3,1} = 5$

7.2.3. Màquines en paral·lel

Les màquines del problema plantejat s'agrupen en grups de g diferents nivells. Els grups de màquines estan situats en paral·lel entre ells. Cada màquina pot treballar les peces del seu nivell i les peces dels nivells inferiors. És a dir, en el cas que $k = 3$ les màquines situades al nivell alt poden processar totes les peces; les màquines classificades al nivell mitjà poden processar les peces del nivell mitjà i baix, i les màquines del nivell baix, només poden processar peces d'aquest nivell.

Cada grup de màquines d'un nivell $l m_i$, tindrà un nombre de màquines situades en paral·lel. Per tant, el nombre total de màquines m es definirà com:

$$\sum_{k=1}^g m_k = m \quad (\text{eq 7.2})$$

On g és el nombre total de nivells i m_k és el nombre de màquines de cada nivell k .

Per anomenar les màquines dins de cada nivell s'utilitzarà el subíndex h . La màquina h -èssima del grup k es defineix com $m_{k,h}$.

Els nivells queden definits tal i com es pot observar a la Taula 7.3:

$l m_i$	Màquines	Característiques
1	$m_{11} \dots m_{1m_1}$	Poden processar totes les peces
$\forall k$	$m_{k1} \dots m_{km_k}$	Poden processar les peces de nivell k i inferiors
g	$m_{g1} \dots m_{gm_g}$	Només poden processar les peces de nivell g

Taula 7.3 Nivells dels grups de màquines

Concretant, les màquines estan agrupades en grups en paral·lel, segons el seu nivell, i dins els grups també estan situades en paral·lel. Totes les màquines tenen els mateixos temps de procés per cada peça, ja que s'ha considerat que totes les màquines processen les peces a la mateixa velocitat. La única diferència que tenen les màquines entre dos nivells és el tipus de peces que poden processar.

7.2.4. Exemple de definició de màquines (exemple 1)

Per tal de programar les peces en les diferents màquines, es defineix el vector J_{kh} com el vector que contindrà la seqüència del conjunt de peces assignades i seqüenciades a la màquina h del nivell k , amb $k = 1 \dots g$. En els propers capítols es definirà un criteri per determinar la distribució de les peces en les màquines de cada nivell. Continuant amb l'exemple, si es considera que:

- Hi ha 3 nivells, $g = \{1, 2, 3\}$
- Cada nivell té un número diferent de màquines: $m_1 = 3$; $m_2 = 2$; $m_3 = 1$
- Totes les peces s'assignen al nivell 1

7.3. Compatibilitat entre les peces i les màquines

A la Taula 7.4 es pot visualitzar de forma simple la relació entre les peces i les màquines de diferents nivells. Cada màquina d'un nivell pot processar les peces dels nivells que estan marcats en verd.

		Màquines					
		Nivell 1	Nivell 2	...	Nivell k	...	Nivell g
Peces	Nivell 1						
	Nivell 2						
	...						
	Nivell k						
	...						
	Nivell g						

Taula 7.4 Compatibilitat entre peces i màquines

Idealment totes les peces serien tractades per les màquines de nivell alt, ja que per motius tècnics o per ser de compra més recent aquestes màquines tenen la capacitat de tractar totes les peces. Tot i així, per reduir l’instant de finalització de la producció, les peces es programaran en diferents nivells. D’aquesta manera es garantirà que els recursos s’aprofitaran el màxim que la distribució de peces permeti.

Canviar la producció d’una peça d’una màquina superior a una màquina d’inferior categoria comporta un increment de cost, ja que la tecnologia d’aquesta és de no tanta qualitat i per tant ofereix menys prestacions. Per aquest motiu s’associa una penalització c_j per cada canvi de nivell que realitza la peça. Segons les condicions del problema els únics canvis permesos són entre nivells consecutius, entre el nivells $1 \rightarrow 2$, els nivells $2 \rightarrow 3$, els nivells $3 \rightarrow 4$, etc.

La penalització associada al canvi de nivell d’una peça del nivell k al nivell $k+1$ és de c . En aquest treball es considera $c_1 = \dots = c_g = 1$. Per tant si una peça està seqüenciada a J_1 , la solució tindrà una penalització associada de 0; si està seqüenciada a J_2 tindrà una penalització associada de 1 i així successivament en tots els nivells.

7.3.1. Programació

Per tal de descriure la programació del conjunt de n peces a les m màquines s’utilitzen diferents vectors:

- **A_{J_k}**: Conjunt de peces assignables al nivell k
- **S_{J_k}**: Índexs de les peces assignades al nivell $k \rightarrow \sum_{k=1}^g S_{J_k} = n$
- **J_{M_{k,h}}**: Índexs de peces assignades a la màquina h del nivell $k \rightarrow \sum_{k=1}^g \sum_{h=1}^{m_k} J_{M_{k,h}} = n$

- $J_{k,h}$: Seqüència de les peces assignades a la màquina h del nivell k

Quan $t_0=0$, l'instant de sortida ($t_{[j]}$) de la màquina per una peça que ocupa la posició $[j]$ de la seqüència $J_{h,k}$, es calcula com:

$$t_{[j]} = \max(r_{[j]}, t_{[j-1]}) + p_{[j]} \quad j = 1, \dots, a_{h,k} \quad (\text{eq 7.3})$$

D'altra banda, una peça es considera acabada un cop s'han completat els tres temps: pre-procés, procés i post-procés. Llavors l'instant de finalització de la peça que ocupa la posició $[j]$ de la seqüència $J_{k,h}$ dependrà de la seqüenciació i dels temps de sortida de les peces que ocupen les posicions anteriors de la seqüència. Aquest instant de finalització es calcula com:

$$f_{[j]} = t_{[j]} + q_{[j]} \quad j = 1, \dots, a_{h,k} \quad (\text{eq 7.4})$$

7.3.2. Exemple de programació de les peces a les màquines (exemple 1)

Una possible solució de l'exemple 1 on es troben totes les peces assignades al nivell 1 és:

$J_{h,k}$	k=1	k=2	k=3
h=1	[3]	[∅]	[∅]
h=2	[1, 5]	[∅]	
h=3	[4, 2]		

Taula 7.5 Solució 1 de l'exemple de l'entorn de peces de l'exemple 1

En aquest exemple de solució veiem que els vectors AJ_k , SJ_k , $JM_{k,h}$ i $J_{k,h}$ que defineixen la solució prenen els següents valors:

- $AJ_1 = \{1,2,3,4,5\}$, $AJ_2 = \{3,4,5\}$, $AJ_3 = \{5\}$
- $SJ_1 = \{1,2,3,4,5\}$, $SJ_2 = \{\emptyset\}$, $SJ_3 = \{\emptyset\}$
- $JM_{1,1} = \{3\}$, $JM_{1,2} = \{1,5\}$, $JM_{1,3} = \{2,4\}$, $JM_{2,1} = \{\emptyset\}$, $JM_{2,2} = \{\emptyset\}$, $JM_{3,1} = \{\emptyset\}$
- $J_{1,1} = \{3\}$, $J_{1,2} = \{1,5\}$, $J_{1,3} = \{4,2\}$, $JM_{2,1} = \{\emptyset\}$, $JM_{2,2} = \{\emptyset\}$, $JM_{3,1} = \{\emptyset\}$

7.4. Índexs d'eficiència

Els índexs d'eficiència són els indicadors que permeten discernir i escollir el millor programa en funció dels objectius que es vulguin complir. En el cas del present projecte, per tal d'avaluar les solucions obtingudes i escollir el millor procediment, es tindran en compte dos índexs d'eficiència:

- **Instant global de finalització (F_{\max}):** és el darrer instant en què totes les peces s'han processat i han sortit del sistema, és a dir, moment en què l'etapa de post-procés de totes les peces ha finalitzat.

Existeix un instant global de finalització de cada màquina. Considerant l'instant de finalització d'una peça $f_{[j]}$, l'instant de finalització de la màquina $m_{k,h}$ serà l'instant de finalització més gran d'una de les seves peces:

$$F_{MAX k} = \max(f_{[j]}) \text{ per les peces } j \in J_k \text{ amb } k = [1, \dots, g] \quad (\text{eq. 7.5})$$

O bé:

$$F_{MAX k} = \max(F_{k,h}) \text{ per } \forall k \text{ amb } h = [1, \dots, n_h] \quad (\text{eq. 7.6})$$

Els instants finals de finalització de cada màquina quedaran recollits en el vector $\overline{F_{MAX}}$, a partir del qual es calcularà l'instant global de finalització:

$$F_{MAX} = \max(\overline{F_{MAX}}) = \max(F_{MAX kh}) \text{ amb } k = [1, \dots, g] \quad (\text{eq. 7.7})$$

- **Penalització per canvis de nivells (c):** indica la penalització total després de programar les peces d'un nivell superior a un nivell immediatament inferior per tal de reduir el valor de F_{MAX} .

La penalització global per canvis de nivells (c) serà la suma de les penalitzacions acumulades pels diferents canvis de nivells realitzats en cadascuna de les peces:

$$c = \sum_{j=1}^n c_j \text{ on } c_j = k - l \text{ amb } k = [1, \dots, g] \quad (\text{eq. 7.8})$$

D'altra banda, en el capítol 8 s'explica la manera de prendre decisions en situacions com la tractada, és a dir, situacions bicriteri on s'han d'optimitzar dos índexs d'eficiència alhora.

Mitjançant els dos índexs d'eficiència es podran representar les diferents solucions trobades. Per tant es tindrà un conjunt de solucions formades per als (F_{MAX}, c) corresponents.

7.4.1. Exemple d'índexs d'eficiència (exemple 1)

De la solució 2 de l'exemple 1 mostrada a Taula 7.6 en podem extreure els següents paràmetres:

- $AJ_1 = \{1,2,3,4,5\}$, $AJ_2 = \{3,4,5\}$, $AJ_3 = \{5\}$
- $SJ_1 = \{1,2,3\}$, $SJ_2 = \{4\}$, $SJ_3 = \{5\}$
- $JM_{1,1} = \{3\}$, $JM_{1,2} = \{1\}$, $JM_{1,3} = \{2\}$, $JM_{2,1} = \{4\}$, $JM_{2,2} = \{\emptyset\}$, $JM_{3,1} = \{5\}$
- $J_{1,1} = \{3\}$, $J_{1,2} = \{1\}$, $J_{1,3} = \{2\}$, $J_{2,1} = \{4\}$, $J_{2,2} = \{\emptyset\}$, $J_{3,1} = \{5\}$

$J_{h,k}$	k=1	k=2	k=3
h=1	[3]	[4]	[5]
h=2	[1]	[\emptyset]	
h=3	[2]		

Taula 7.6 Solució 2 de l'exemple de l'entorn de peces de l'exemple 1

A partir de la solució es poden calcular els índexs d'eficiència:

Instant de finalització

- $F_{1,1} = \{10\}$, $F_{1,2} = \{9\}$, $F_{1,3} = \{6\}$, $F_{2,1} = \{12\}$, $F_{2,2} = \{0\}$, $F_{3,1} = \{8\}$
- $F_{MAX,1} = \{10\}$, $F_{MAX,2} = \{12\}$, $F_{MAX,3} = \{8\}$
- $F_{MAX} = \{12\}$

Penalització

- $c = 3$

7.5. Formulació del problema

Un cop descrites les hipòtesis dels problema, concretats els seus elements i definits els seus índexs d'eficiència, el problema estudiat es defineix segons la nomenclatura de Lawler (Lawler, Lenstra, Rinnooy, & Shmoys, 1993) com:

$$P_m \mid r_j, q_j, M_j \mid (F_{max}, c)$$

L'enunciat del problema estudiat és:

Es té un nombre finit de peces classificades en g nivells (denotat per M_j) que s'han de processar en m màquines classificades en els mateixos g nivells. Aquestes peces estan definides pel nivell al qual pertanyen i pels temps, coneguts i enters, de pre-procés (r_j), procés (p) i post-procés (q_j).

Les màquines són idèntiques entre elles i estan situades en paral·lel. Cada nivell té un nombre independent i finit de màquines.

Per resoldre aquest problema s'ha de trobar una seqüència d'operacions per a cada màquina de cada nivell i trobar l'interval temporal d'execució de cadascuna de les operacions sobre les peces. L'objectiu és minimitzar, a la vegada, l'instant global de finalització i la penalització global per canvis de nivells.

7.6. Solució del problema

La solució del problema són les seqüències trobades que donen el valor baix de F_{MAX} per cada valor de penalització.

És a dir, es presentaran un seguit de solucions amb el seu corresponent valor de penalització c i el seu instant de finalització global F_{MAX} per tal de que l'usuari pugui escollir, segons els seus criteris, la seqüència desitjada.

La solució es presentarà de la següent forma:

$$\mathbf{J} = [\{ (\dots), (\dots), (\dots), (\dots) \}, \{ (\dots), (\dots), (\dots), (\dots) \}]$$

- [] → programa global
- { } → programació de cada nivell de penalització (c)
- () → programació de cada màquina

8. Optimització bicriteri

Ja que prendre decisions correctes és la base d'una bona planificació estratègica, en els àmbits industrials moltes vegades no és suficient centrar-se en un criteri a l'hora de buscar millores o optimitzar processos, sinó que és necessari centrar-se en diferents criteris per escollir l'opció millor a seguir.

La presa de decisions multicriteri permet treballar en les situacions on és necessari satisfer més d'un objectiu alhora. En aquest tipus de decisions les conseqüències de cada alternativa estan ben definides. El que no està definit tant clarament és la decisió final quan existeix més d'un criteri en conflicte. La suma de totes les alternatives forma el conjunt d'elecció. El concepte de solució utilitzat normalment és el concepte d'òptim de Pareto. Aquest concepte està basat en el *criteri d'optimitat paretiana*, enunciat per Pareto el 1896: “Una alternativa és *eficient* (o *Pareto òptima*) si tota alternativa que proporcioni una millora en un atribut produeix un empitjorament en, almenys, un dels altres atributs”.

8.1. Formulació multicriteri

La decisió multicriteri tan sols difereix de l'optimització amb un criteri simple en el fet que té varies funcions objectius simultànies. L'objectiu d'aquest tipus d'optimització és assolir el millor entre el que és possible.

En general un problema de decisió multicriteri ve formulat de la següent forma:

$$\begin{aligned} \text{opt } z(x) &= (z_1(x), \dots, z_p(x)) \\ x &\in F \end{aligned}$$

On x és el vector de variables de decisió i F és l'espai de decisions o solucions. Aquest espai engloba totes les possibles solucions del sistema. Cada solució és diferent, exclouent i exhaustiva: no es pot escollir una opció mixta entre dues alternatives ni es pot escollir una solució de fora del conjunt.

A continuació s'exposen alguns conceptes bàsics de la decisió multicriteri (Romero, 1996):

- **Atribut:** valor observat mitjà d'una decisió independentment del decisor; és necessari que es puguin mesurar independentment i que alhora es puguin expressar com una funció de les corresponents variables de decisió.

- **Objectiu:** direcció de millora d'un atribut; aquesta direcció pot ser de maximització o minimització.
- **Nivell d'aspiració:** nivell acceptable d'assoliment per un atribut.
- **Meta:** combinació d'un atribut amb el seu nivell d'aspiració.
- **Criteri:** és cadascun dels atributs, objectius o metes rellevants en un problema de decisió.

Per tant, la metodologia per prendre decisions multicriteri consisteix primer en trobar el conjunt d'elecció F , és a dir, un conjunt de solucions admissibles que compleixin les restriccions plantejades en el problema. Després, cal avaluar cada solució admissible associant-li els valors corresponents dels atributs, per finalment triar el conjunt de solucions que compleixin millor amb les funcions objectius ($z(x)$).

8.2. Espai de solucions i no dominància de solucions

Cal recordar el concepte d'òptim de Pareto: “Una alternativa és *eficient* (o *Pareto òptima*) si tota alternativa que proporcioni una millora en un atribut produeix un empitjorament en, almenys, un dels altres atributs”. De la definició d'alternativa eficient o Pareto òptima, se'n deriva la d'*alternativa dominada* o *alternativa no eficient*, com una alternativa per a la qual existeix una altra alternativa amb un dels atributs millor.

En el cas d'una minimització dels dos criteris es defineix:

- **Dominància de Pareto:** donat un vector $u = (u_1, \dots, u_k)$, es diu que domina a un altre vector $v = (v_1, \dots, v_k)$ si i només si:

$$\forall i \in \{1, \dots, k\}, u_i \leq v_i \quad \exists i_0 \in \{1, \dots, k\} \mid u_{i_0} < v_{i_0}$$

- **Òptim de Pareto:** una solució x es diu que és òptima de Pareto si i només si **no existeix** un altre vector y tal que $f(y) = (v_1, \dots, v_k)$ domini a $f(x) = (u_1, \dots, u_k)$.

En el cas de buscar la maximització dels criteris, únicament s'hauria de canviar el sentit de les inequacions de comparació de les alternatives.

En altres paraules, la definició anterior diu que el punt x és un òptim de Pareto, una solució **no dominada**, si no existeix un vector y que faci millor un dels objectius obtinguts amb x sense que empitjori de forma simultània algun dels altres. Per tant, una solució es considerarà òptima de Pareto si una millora en un dels seus criteris, respecte una altra solució, sempre suposa un empitjorament en algun dels altres.

8.3. Presa de decisions bicriteri

En aquest projecte, mitjançant un procediment que s'estipularà en els propers punts, es determinarà un conjunt de solucions no dominades (frontera de Pareto) a partir de dues funcions objectiu, és a dir, es tractarà de decisions bicriteri.

La presa de decisions per la situació tractada queda definida per:

- El conjunt d'elecció F és el conjunt de solucions factibles que compleixen totes les restriccions de peces i màquines definides anteriorment.
- Els dos criteris escollits per al problema estudiat són:
 - Instant global de finalització (F_{MAX}).
 - Penalització per canvis de nivells (c).
- Les funcions objectiu per cada criteri són:
 - $f_1 \rightarrow [MIN] F_{MAX}$
 - $f_2 \rightarrow [MIN] c$

En el cas bicriteri, al simplificar al nivell bidimensional, el *conjunt eficient* o *de Pareto* o també anomenat *frontera de Pareto*, es pot visualitzar de forma senzilla en forma de gràfic (F_{MAX} , c).

Un exemple d'aquest gràfic seria el mostrat a continuació (Figura 8.1):

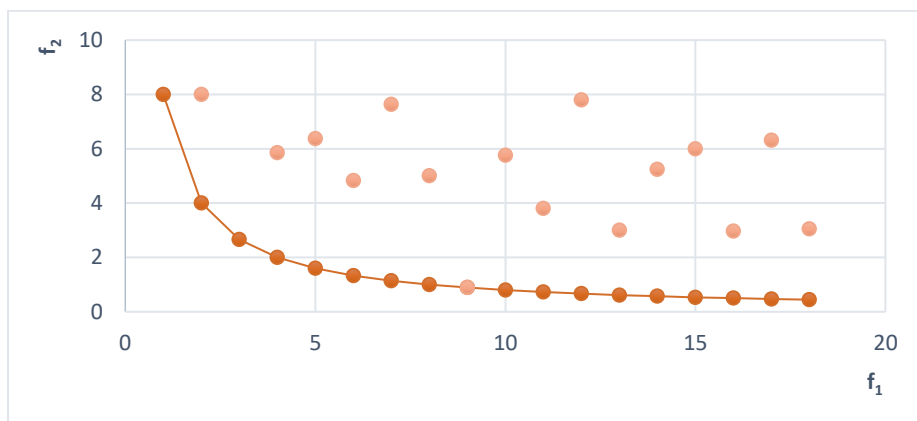


Figura 8.1 Conjunt de solucions amb dos criteris en el cas d'una minimització d'ambdós criteris

Es pot observar que existeixen dos tipus de solucions, les solucions factibles no dominades (marcades més fort i unides amb una línia) que formen l'anomenada frontera de Pareto i les solucions factibles dominades (les marcades amb color suau) que són aquelles que estan dins l'espai F , compleixen totes les restriccions, però que són pitjors, pels criteris escollits, que les que es troben sobre la frontera de Pareto.

En el cas del present projecte a la Figura 8.1 f_1 i f_2 equivaldrien al temps màxim de finalització i a la penalització respectivament: $f_1 \equiv F_{MAX}$ i $f_2 \equiv c$.

8.3.1. Càlcul de la no dominància de solucions

En el present projecte, per tal d'avaluar l'aplicació dels diversos procediments s'estudiaran les quantitats de solucions dominades i no dominades que s'obtidran en cada una d'elles. D'aquesta forma, un procediment amb més solucions no dominades d'entre les obtingudes amb tots els procediments indica que és el procediment que obté les millors solucions.

Essent \mathbf{X} un conjunt de solucions obtingut per a un determinat procediment es calcula el percentatge de solucions no dominades com:

$$\%SND = \frac{\text{Nombre de solucions de } X \text{ no dominades}}{\text{Total de solucions de } X} \cdot 100 \quad (\text{eq. 8.1})$$

El *nombre de solucions de X no dominades* es calcula a partir de la definició de frontera de Pareto donat el conjunt de solucions trobades per tots els procediments. El nombre *total de solucions de X* és el nombre total de solucions que aquest mateix procediment ha trobat.

Essent \mathbf{X} i \mathbf{Y} dos conjunts de solucions diferents obtinguts per procediments diferents, una solució $\mathbf{a} \in \mathbf{X}$ amb F_{MAX}^a i c^a serà no dominada quan $\forall \mathbf{b} \in \mathbf{Y}$ amb F_{MAX}^b i c^b , amb $c^b \leq c^a$, tingui un valor $F_{MAX}^a < F_{MAX}^b$. Alternativament, \mathbf{a} serà una solució no dominada si $\forall \mathbf{b} \in \mathbf{Y}$ amb $F_{MAX}^a \leq F_{MAX}^b$ es té que $c^b < c^a$.

8.3.2. Comparació entre dos procediments

Per comparar el funcionament i la qualitat de les solucions de dos procediments es calcularan les solucions dominades que ambdues obtenen. D'aquesta manera es podrà saber quin dels dos procediments comparats aporta més solucions no dominades. Com major sigui el nombre de solucions no dominades d'un procediment respecte l'altre, significarà que aquest procediment és de major qualitat.

Per a dos conjunts de solucions \mathbf{X} i \mathbf{Y} obtinguts amb procediments diferents (procediment X i procediment Y), es calcularia el percentatge de solucions no dominades del procediment X respecte el total dels procediment X i Y com:

$$\%ND = \frac{\text{Solucions de } X \text{ no dominades per } Y}{\text{Total de solucions no dominades } X \cup Y} \cdot 100 \quad (\text{eq. 8.2})$$

8.3.3. Exemple de comparació entre dues heurístiques

Es parteix dels conjunts de solucions **X** i **Y**, els valors dels quals estan a la Taula 8.1 i la Taula 8.2:

Solucions del conjunt X							
c	0	1	3	5	6	7	8
F _{MAX}	55	50	36	30	28	26	22

Taula 8.1 Exemple de solucions del conjunt X

Solucions del conjunt Y					
c	0	1	2	3	4
F _{MAX}	55	48	36	32	29

Taula 8.2 Exemple de solucions del conjunt Y

Partint de la definició de dominància, a la Taula 8.3 s’especificuen les solucions dominades i les no dominades:

X	P	0	1	2	3	4	5	6	7	8
	F _{MAX}	55	50	-	36	-	30	28	26	22
Y	P	0	1	2	3	4	5	6	7	8
	F _{MAX}	55	48	36	32	29	-	-	-	-
X ∪ Y	P	0	1	2	3	4	5	6	7	8
	F _{MAX}	55	48	36	32	29	-	28	26	22

Taula 8.3 Solucions no dominades de X i Y

En verd podem veure les solucions no dominades de **X** i **Y**, en vermell les solucions dominades d’aquests dos conjunts i en groc aquelles solucions que són no dominades en el conjunt **X ∪ Y** però que pel fet de ser iguals en **X** i en **Y** es considera que **X** domina **Y** i que **Y** domina a **X** en el moment de fer la comparació.

Els percentatges de solucions no dominades de les dues heurístiques, calculat a partir de la eq. 8.2 són:

- Percentatge de solucions de X no dominades per $X \cup Y$: $(3/8) \cdot 100 = 37,5\%$
- Percentatge de solucions de Y no dominades per $X \cup Y$: $(4/8) \cdot 100 = 50\%$

Per tant, es podria afirmar que l'heurística Y és de major qualitat que l'heurística X ja que té un percentatge de solucions no dominades de 50%.

9. Procediment de resolució

9.1. Objectiu

L'objectiu del present projecte és trobar en l'espai de solucions les que són no dominades. És a dir, es pretén trobar un seguit de programacions on les peces estiguin assignades a les màquines, amb un determinat ordre, cadascuna d'elles amb un instant global de finalització (F_{MAX}) determinat i un valor associat de penalització (c).

Es busca obtenir els elements de la frontera de Pareto. Una frontera es compondrà d'un seguit de solucions no dominades amb els seus respectius instants globals de finalització (F_{MAX}) i els diferents valors de penalització (c). El conjunt d'elements de la frontera serà similar al de la Taula 8.1 o al de la Taula 8.2. Es buscarà obtenir un conjunt de solucions el més complet possible que permeti fàcilment escollir quina programació, de les proposades, és millor utilitzar segons la penalització i l'instant global de finalització que desitgi el decisor.

Per trobar aquests programes es partirà d'una solució amb el valor de c mínim ($c = 0$) i es canviaran les peces de nivells superiors a nivells immediatament inferiors, reduint l'instant global de finalització (F_{MAX}) a canvi d'incrementar la penalització (c). Aquest procediment per obtenir un conjunt de solucions ja ha estat utilitzat en (Mateo, Teghem, & Tuytens, 2017) així com en altres treballs com (Camps, 2013)

9.2. Esquema general

El problema plantejat és del tipus combinatori en el qual existeix un conjunt finit de solucions, però molt elevat per a una certa dimensió del problema. Per resoldre'l es poden aplicar mètodes exactes i mètodes heurístics (Pastor, 2016):

- Els **mètodes exactes** són procediments que garanteixen trobar la solució buscada (òptima), si aquesta existeix, en un temps finit. Malgrat això, el temps invertit per aquests pot ser d'un ordre de magnitud molt superior al del necessari per a un procediment heurístic.
- Els **mètodes heurístics** són mètodes per resoldre, mitjançant una aproximació intuïtiva, problemes d'optimització ben definits. Les solucions que troben poden no ser les òptimes del problema. Aquests mètodes són molt útils en sistemes on és igual d'important la rapidesa que la solució obtinguda.

En un sistema com l'estudiat, a la vida real, cal donar solucions de manera ràpida en determinats casos on la seqüència necessita ser reprogramada. Per tant, en aquest projecte, es buscaran les solucions al problema plantejat mitjançant mètodes heurístics. Dins la gran varietat de sistemes heurístics s'utilitzaran les següents opcions:

- **Procediment *greedy* directe:** consisteix en escollir l'opció millor en cada decisió local, amb l'esperança d'arribar a una solució òptima local en general, i en el nostre cas no dominada. Cada solució es construeix de forma progressiva al seleccionar, segons un índex avaluador, el millor candidat d'una llista ordenada.
- **Procediment de *backtracking*:** consisteix en construir les solucions amb la possibilitat de desfer alguna de les decisions preses i reprendre el procés en una altra direcció.

El mètode de resolució per al problema plantejat en el present projecte es basarà en una combinació de diferents variants per a cada fase dels procediments. Els algorismes generaran unes solucions que es crearan en una estructura genèrica d'arbre. Aquest procediment queda definit pels nodes, l'arrel de l'arbre, el procediment de ramificació i l'estratègia d'exploració:

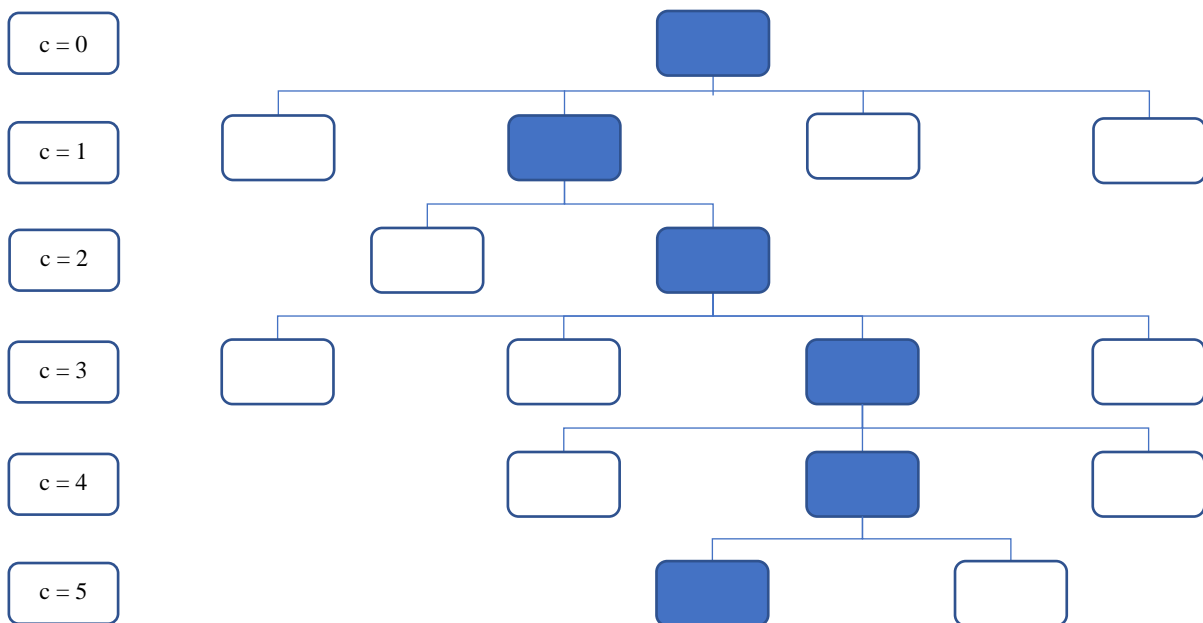


Figura 9.1 Esquema d'una exploració arborescent

- Un **node** queda definit per la solució s i hi inclou la solució trobada (J^s) definida pels instants globals de finalització (F_{MAX}^s), la penalització (c^s) i la programació de les peces de cada màquina.

- L'**arrel de l'arbre** és el primer node a partir del qual es realitza l'exploració i conté la solució inicial ($s=0$) de penalització $c = 0$ mostrat a la Figura 9.1.
- L'**estratègia d'exploració** consisteix en triar la peça que, un cop canviada de nivell doni un nou instant de finalització (F'_{MAX}) menor. Es considera com a nivell superior el nivell de la màquina $m_{k,h}$ amb un instant de finalització major.
- El **procediment de ramificació** consisteix en canviar una peça d'un nivell a un nivell immediatament inferior. El nombre de nodes que són resultat de la ramificació depèn del nombre de peces que podrien estar assignades al nivell immediatament inferior.

A la Figura 9.2 es representa la informació que conté cada node de l'arbre:

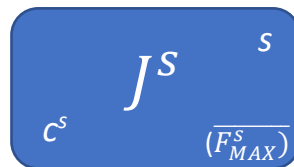


Figura 9.2 Representació d'un node de l'arbre

Inicialment es calcularà la seqüència inicial. Aquesta seqüència és una solució on totes les peces estaran situades a les màquines de nivell superior. Aquesta part s'anomenarà *Fase 0*.

Seguidament, a partir de la seqüència inicial, es realitzarà un procediment d'exploració arborescent de tipus *greedy* directe que permetrà realitzar una primera cerca de nodes amb una solució per nivell per tots els nivells explorats. Aquesta part s'anomenarà *Fase 1*. El procés de la *Fase 1* és similar al de *branch and bound*, amb la principal diferència que no s'utilitzen fites. Un *branch and bound* és un procediment on es realitza una exploració a l'espai de decisions separant la regió factible en conjunts més petits, un indicador de qualitat (fita) i sense eliminar cap solució factible, i resoldre el problema en cada conjunt (Pastor, 2016).

A la Figura 9.1 es pot veure un exemple d'aquesta exploració arborescent on cada requadre és un node i indica una solució factible. A l'esquerra, s'hi indica la penalització que té cada nivell d'aquest arbre (c). D'altra banda, cada quadre ombrejat representa la solució potencialment no dominada d'aquell nivell de penalització. És a dir, la solució on, per un mateix valor de penalització (c), s'ha trobat un millor valor de F_{MAX} . El primer node de l'arbre és el node obtingut a partir de l'heurística de la *Fase 0*. A partir d'aquest node es comença a realitzar el procés arborescent amb el que es trobarà el conjunt de solucions. I dins aquest conjunt, es

trobarà el conjunt de solucions no dominades (frontera de Pareto). A partir d'aquest conjunt de solucions es procedirà a millorar les solucions mitjançant un procediment de *backtracking*. Aquest procediment consistirà en marcar un seguit de nodes destacats per començar una posterior exploració arborescent a partir de cada un d'ells mitjançant una variació del procediment utilitzat a la *Fase 1*. Això permetrà obtenir un nombre major de solucions i, probablement, un nombre major de solucions no dominades.

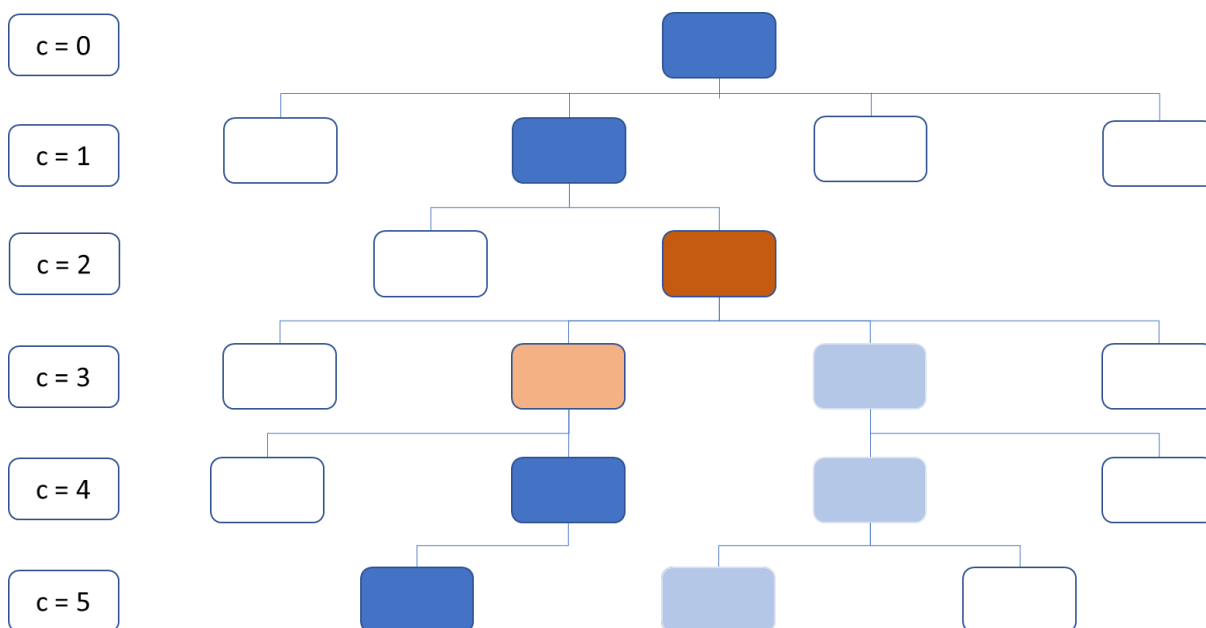


Figura 9.3 Esquema d'una exploració arborescent on s'aplica la Fase 2

A la Figura 9.3 es pot observar un exemple d'exploració arborescent on s'hi pot veure un node marcat (taronja fosc) i el node que se n'obté a partir de l'aplicació de la Fase 2 (taronja clar). A partir d'aquest nou node es torna a aplicar el procediment de la Fase 1. Els nodes marcats amb blau clar, són els nodes obtinguts en l'anterior exploració arborescent de la Fase 1.

En resum, el procés es divideix en tres fases:

- **Fase 0: Vèrtex arrel** → obtenció de la solució de partida on $c = 0$
- **Fase 1: Exploració arborescent mitjançant un procediment *greedy*** → obtenció de les solucions de menys a més penalització mitjançant un procés arborescent
- **Fase 2: Exploració diferents branques segons un procediment de *backtracking*** → cerca de solucions amb menor F_{MAX} a igual penalització passant per una solució d'un nivell que no és la millor

Finalment després d'aquestes tres fases es disposarà de les solucions finals.

9.3.Fase 0: seqüència inicial de l'estructura arborescent ($c = 0$)

Es parteix dels cas ja definit anteriorment, de m màquines idèntiques i en paral·lel classificades en diferents nivells. Hi ha un total de k nivells. Aquestes màquines estan totes disponibles des de l'instant $t=0$. S'ha de programar n peces classificades en els mateixos nivells. Cada una de les peces es comença a processar en un instant de temps igual o superior al temps de pre-procés (r_j) durant un temps de procés (p_j). Finalment, un cop acabat el procés, s'haurà de realitzar un post-procés (q_j) durant un cert temps per acabar el tractament de l'esmentada peça. L'instant de finalització es considera passat aquest temps de post-procés.

Es proposarà una seqüència solució inicial on totes les peces estaran programades a les màquines de nivell alt. Aquesta seqüència es trobarà a l'arrel de l'arbre que contindrà la solució inicial amb penalització $c^0 = 0$.

Quan es vagin explorant nous nodes, es proposarà una alternativa nova de seqüència en cadascun dels nivells on s'incrementi el nombre de peces. S'utilitzarà el mateix mètode que en la seqüència inicial. En el cas de les màquines dels nivells que perdin peces perquè alguna d'elles passa a un nivell inferior, es respectarà les seves seqüències inicials, exceptuant evidentment la que canvia de nivell.

Per tal d'obtenir l'arrel d'on començarà el procés arborescent, es proposa l'heurística de la Fase 0. Aquesta heurística està basada en la proposada per Anis Gharbi i Mohamed Haouari (Gharbi & Haouari, 2002). L'heurística en qüestió consisteix en trobar la programació d'un seguit de peces en un nombre definit de màquines en paral·lel, considerant temps de pre-procés, procés i post-procés.

L'objectiu de l'heurística és determinar quines peces poden ser col·locades més aviat, és a dir, deixen les màquines lliures després de les operacions prèvies el més aviat possible ($\min[r_j+p_j]$) i quines peces haurien de ser col·locades les últimes, és a dir, tarden el mínim a acabar el procés un cop entrades a les màquines ($\min[p_j+q_j]$).

*Procediment heurístic de la solució del vèrtex***FASE 0 – Vèrtex arrel**

Sigui **J** la matriu formada pels vectors **J_{kh}**, que contenen el conjunt de peces assignades a la màquina $h=\{1, \dots, m_k\}$ del nivell $k=\{1, \dots, g\}$.

Per trobar la seqüència del vèrtex arrel, es programarà totes les peces a les màquines de nivell 1, obtenint, per cada màquina el corresponent vector **J_{1h}** de la matriu **J**.

Sigui **AJ** el conjunt de peces disponible per ser programades a la màquines de nivell 1 i siguin **J_R ∈ AJ**, **J_Q ∈ AJ** i **J_M ∈ AJ** subconjunts de peces $j \in \mathbf{AJ}$ seleccionades.

Sigui **T_{kh}** el vector d'**instant de sortida de les peces** per a cada màquina $h=\{1, \dots, m_k\}$ del nivell $k=\{1, \dots, g\}$. Cada component $t_{[j]}$ del vector **T_{kh}** representa l' instant en que la peça que ocupa la posició [j] surt de la màquina h de nivell k .

Sigui **F_{kh}** el vector d'**instant de finalització de les peces** per a cada màquina $h=\{1, \dots, m_k\}$ del nivell $i=\{1, \dots, k\}$. Cada component $t_{[j]}$ del vector **F_{kh}** representa l' instant en que la peça que ocupa la posició [j] a la màquina h de nivell k acaba el seu tractament.

Seqüència inicial (σ) (Gharbi & Haouari, 2002):

Fase A – Algoritme de separació per classificar les peces en tres grups segons els temps r_j i q_j :

1- Sigui **J_M = AJ = {1, 2, 3, 4, ..., n}**, **J_R = 0** i **J_Q = 0**, **m = nombre de màquines**.

2- S'ordena **J_M** en ordre creixent de $r_j + p_j$

Si $r_{j_0} + p_{j_0} \leq r_{j_{(m+1)}}$:

Si $|J_M| \leq m \rightarrow \text{STOP}$

$J_M = J_M \setminus \{j_0\}$

$J_R = J_R \cup \{j_0\}$

Si no hi ha peces \rightarrow Pas 3

3- S'ordena J_M en ordre creixent de $p_j + q_j$

Si $p_{j_0} + q_{j_0} \leq q_{j_{(m+1)}}$:

Si $|J_M| \leq m \rightarrow \text{STOP}$

$J_M = J_M \setminus \{j_0\}$

$J_Q = J_Q \cup \{j_0\}$

Si no hi ha peces \rightarrow Pas 4

4- Si no hi ha hagut canvis al pas 3 \rightarrow **Fase B**

Si hi ha hagut canvis al pas 3 \rightarrow Pas 2

Exemple de la Fase A per el cas de dues màquines ($m_1=2$):

Si es tenen les següents peces amb els corresponents temps r_j , p_j i q_j :

Peça	$j1$	$j2$	$j3$	$j4$	$j5$
r_j	3	9	7	3	6
p_j	5	4	2	1	6
q_j	6	9	3	7	6

Pas 1:

$J_M = AJ = \{1,2,3,4,5\}$

$J_R=0$

$J_Q=0$

Pas 2: $J_M = \{4,1,3,5,2\}$

Si $r_{j_4} + p_{j_4} \leq r_3$ ($3 + 1 < 7$) $\rightarrow J_R = \{4\}$, $J_M = \{1,3,5,2\}$, $|J_M| > m$

Si $r_{j_1} + p_{j_1} \leq r_5$ ($3 + 5 > 6$) \rightarrow no compleix, $|J_M| > m$

Pas 3: $J_R = \{4\}$, $J_Q = \{\}$, $J_M = \{3,1,5,2\}$

Si $p_{j_3} + q_{j_3} \leq q_{j_5}$ ($2 + 3 < 6$) $\rightarrow J_R = \{4\}$, $J_Q = \{3\}$, $J_M = \{1,5,2\}$, $|J_M| > m$

Si $p_{j_1} + q_{j_1} \leq q_{j_2}$ ($5 + 6 > 9$) \rightarrow no compleix, $|J_M| > m$

Pas 4: $J_R = \{4\}$, $J_Q = \{3\}$, $J_M = \{1,5,2\}$

Hi ha hagut canvis al Pas 3 \rightarrow Pas 2

Pas 2: $J_R = \{4\}$, $J_Q = \{3\}$, $J_M = \{1,5,2\}$

Si $r_{j_1} + p_{j_1} \leq r_2$ ($3 + 5 < 9$) $\rightarrow J_R = \{4,1\}$, $J_Q = \{3\}$, $J_M = \{5,2\}$, $|J_M| \leq m \rightarrow$ STOP

$J_R = \{4,1\}$, $J_Q = \{3\}$, $J_M = \{5,2\}$

Fase B – Programació de les peces a les màquines (Global Schedule Construction Algorithm):

- 1- Repartir les peces de J_M a les màquines. S'assigna la peça que primer arriba a la màquina que primer estigui disponible. En cas d'empat s'assignarà la peça amb $r_{j_0} + p_{j_0}$ major.

Un cop assignades les peces de J_M s'ha de definir:

$u_{kh}(\sigma)$: instant en què la màquina h del nivell k està disponible a la programació σ .

σ^{-1} : programació simètrica de σ .

- 2- Assignació de les peces dels subconjunts J_Q i J_R , prioritzant sempre la màquina que es trobi disponible abans.

Assignació de les peces tal que $j \in \bar{J}_Q$

Invertir programa $\rightarrow \sigma = \sigma^{-1}$

Assignació de les peces tal que $j \in \bar{J}_R$

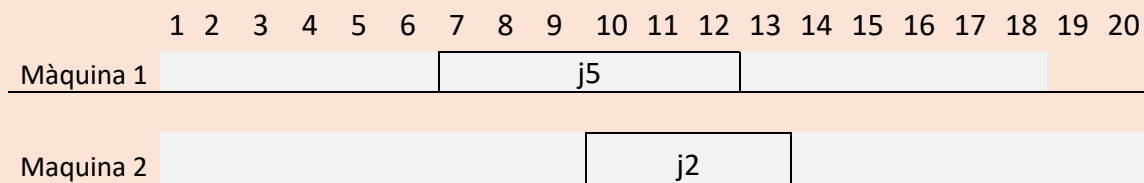
Invertir programa $\rightarrow \sigma = \sigma^{-1}$

- 3- Calcular els vectors T_{kh} segons l'ordre de les seqüències σ_{iz} on cada posició dels vectors es calcula a partir de l'equació: $t_{[j]} = \max(r_{[j]}, t_{[j-1]}) + p_{[j]}$
- 4- Calcular els vectors F_{kh} segons l'ordre de les seqüències σ_{iz} on per a cada posició dels vectors es calcula a partir de l'equació: $f_{[j]} = t_{[j]} + q_{[j]}$
- 5- Calcular l'instant global de finalització a partir de la següent equació: $F_{MAX} = \max(F_{MAXkh})$ on $F_{MAXkh} = \max(F_{kh})$
- 6- Afegir la solució ($s = 0$) a la llista de solucions \mathbf{S} amb penalització $c = 0$ com a: $(F_{MAX}, 0)$

Exemple de la Fase B per al cas de dues màquines ($m_1=2$), seguint l'exemple anterior:

Es té que: $J_R = \{4,1\}$, $J_Q = \{3\}$, $J_M = \{5,2\}$

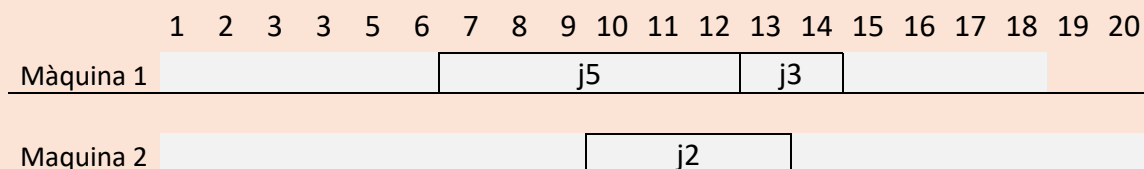
Pas 1:



$u_1(\sigma) = 12, \quad u_2(\sigma) = 13$

Pas 2:


Assignació de les peces tal que $j \in \bar{J}_Q$:




Invertim la programació: $\sigma = \sigma^{-1}$



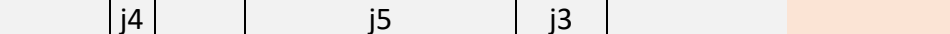
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

Màquina 1  M1: j3 [4, 6], j5 [7, 12]Maquina 2  M2: j2 [10, 13]Assignació de les peces tal que $j \in \bar{J}_R$:

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21

Màquina 1  M1: j3 [4, 6], j5 [7, 12], j4 [13, 15]Maquina 2  M2: j2 [10, 13], j1 [14, 18]Invertim la programació: $\sigma = \sigma^{-1}$:

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22

Màquina 1  M1: j4 [4, 6], j5 [7, 12], j3 [13, 15]Maquina 2  M2: j1 [4, 8], j2 [10, 13]

*Els temps marcats en gris representen els temps de procés compostos per r, p i q . Els temps emmarcats representen els temps d'ocupació de la màquina p .

Programació final:

Peça	$j1$	$j2$	$j3$	$j4$	$j5$
r_j	3	9	7	3	6
p_j	5	4	2	1	6
q_j	6	9	3	7	6

$$J_{11} = [j4, j5, j3]$$

$$T_{11} = [4, 12, 14]$$

$$F_{11} = [11, 18, 17]$$

$$J_{12} = [j_1, j_2]$$

$$T_{12} = [8, 13]$$

$$F_{12} = [14, 22]$$

$$F_{MAX} = 22$$

Les peces de la màquina 1 acaben de ser processades a l' instant $t=18$ i les de la màquina 2 a l' instant $t=22$.

9.4. Fase 1: procediment d'exploració arborescent

El procediment arborescent parteix d'un node arrel. A partir d'aquest node s'explora l'espai de solucions factibles (F) incrementant progressivament la penalització. De tots els nodes d'un nivell de l'arbre, les solucions no dominades presentaran un F_{MAX} (instant màxim de finalització) menor a igual penalització.

En aquesta Fase 1, a partir de la programació del node arrel, es selecciona una màquina d'un nivell, la que comporta la F_{MAX} de la solució. El node es ramifica segons el subconjunt de peces que poden canviar de nivell. Posteriorment s'escull un dels nodes resultant de la ramificació, segons el valor de F_{MAX} que s'obtindrà. Aquest node escollit serà el nou node origen.

Seguidament, un cop escollit un dels nodes, es torna a realitzar una exploració d'un nou nivell de l'arbre seguint els mateixos passos.

Com ja s'ha comentat, es parteix d'una solució inicial de penalització 0 ($c^0=0$) (totes les peces processades per màquines de nivell alt) amb un instant global de finalització elevat (F_{MAX}^0) i es van buscant noves solucions en nivells consecutius reduint aquest instant global de finalització. D'aquesta manera s'obté un espai de solucions amb solucions factibles per a valors de penalització creixents.

9.4.1. Estructura del procediment arborescent

En el cas del projecte present, l'arrel de l'arbre conté la seqüència obtinguda a l'heurística de la Fase 0 esmentada a l'apartat 9.3. Aquesta heurística programa totes les peces a les màquines de nivell 1. A partir d'aquesta seqüència es procedirà a fer l'exploració de l'arbre anant cap a

les fulles, és a dir: avaluar i escollir a partir de cada node un seguit de branques, segons la peça a canviar de nivell, i escollir la peça més indicada per canviar de nivell a un altre immediatament inferior per tal de reduir l'instant global de finalització (F_{MAX}). Tot aquest procediment es farà tenint en compte les restriccions presentades anteriorment. A més, per cada moviment d'una peça a un nivell inferior, la penalització de la solució augmentarà, en aquest cas, en una unitat respecte la solució anterior.

L'estructura genèrica de l'exploració arborescent de la Fase 1 permet generar programacions on, per cada nou nivell, s'incrementa la penalització en una unitat i s'espera que es redueixi (millora) l'instant global de finalització. El procediment de millora de les solucions consistirà, partint d'una seqüència inicial amb F_{MAX}^S i c^s situada en una màquina origen h del nivell k , en moure una de les peces de la màquina origen a una màquina del nivell destí h' del nivell $k+1$ (nivell immediatament inferior) fins que canviï el nivell origen, el nivell origen estigui a l'últim nivell o no es puguin moure més peces des del nivell origen. D'aquesta forma es pretén reduir, en cada increment de la penalització, l'instant global de finalització a partir dels canvis de nivell de peces i , per tant, de la reducció dels instants de finalització de cada màquina.

Cal indicar que durant l'exploració de nous nodes en la Fase 1 es permetrà que el valor trobat en un node sigui pitjor que el valor del node immediatament anterior ($F_{MAX}^{s+1} \geq F_{MAX}^s$). Un cop s'hagi finalitzat la cerca de nous nodes, es farà la cerca de solucions no dominades.

Durant la primera exploració es marcaran un seguit de nodes que s'utilitzaran posteriorment en el procediment de *backtracking* de la Fase 2. Es marcaran els nodes on es produeixi un canvi de nivell origen. És a dir: quan es deixen de poder canviar peces d'una determinada màquina d'un nivell superior per assignar-les a una altra màquina d'un nivell inferior. Aquest fet és un bon punt de partida per trobar noves solucions no dominades durant la Fase 2 que es presentarà en l'apartat 9.5.

FASE 1 – Exploració arborescent

MENTRE NO s'arribi a **FI** (mentre es puguin canviar peces de nivell o no s'indiqui anar a la Fase 2)

S'anomena kr (nivell de referència) el nivell tal que compleix $F_{MAX} = F_{MAXkr}$, i en particular la seva màquina h ($F_{MAXkr,h}$).

S'anomena F'_{MAXkr} el valor de F_{MAXkr} que el nivell kr presenta després de la iteració.

Iteració:

Sigui $F_{MAX} = F_{MAXkr,h} \rightarrow$ moure la peça j de la màquina origen kr,h al nivell $kr+1$ (nivell $kr \rightarrow kr+1$), reprogramant la seqüència $J'_{kr,h}$. La seqüència de les màquines del nivell $k+1$ es reprograma mitjançant l'heurística de la *Fase 0* i es tornen a calcular els nous valors de F_{MAXi} associats a cada màquina i .

- A. Si $F'_{MAXkr} \geq \max(F_{MAX1}, \dots, F_{MAXkr-1}, F'_{MAXkr+1}, \dots, F_{MAXg}) \rightarrow kr$ continua essent el mateix nivell que presenta la F_{MAX} . Repetir la iteració.
- B. Si $F'_{MAXkr+1} > \max(F_{MAX1}, \dots, F'_{MAXkr}, F_{MAXkr+2}, \dots, F_{MAXg})$ i $kr + 1 < g \rightarrow$ repetir la iteració considerant $kr \equiv k+1$ i marcar el node precedent.
- C. Si $F'_{MAXkr+1} \geq \max(F_{MAX1}, \dots, F'_{MAXk}, F_{MAXkr+2}, \dots, F_{MAXg})$ i $kr + 1 = g \rightarrow$ s'acaba la iteració i no es busquen més solucions \rightarrow **FI**

9.4.2. Elecció de la màquina origen per obtenir una nova solució

Per tal de garantir que el temps global de finalització (F_{MAX}) de la solució es redueix en el moment de canviar una peça de nivell a un nivell immediatament inferior, l'elecció de la màquina origen és molt simple: s'escull com a màquina origen aquella màquina que té un instant de finalització major entre totes les màquines de tots els nivells.

En cas que dues màquines tinguin el mateix valor de F_{MAX} serà necessari disposar d'un criteri de desempat, ja que la màquina origen no quedarà definida. En aquest cas se n'han proposat diferents mètodes:

- **Les màquines que presenten el mateix valor de F_{MAX} tenen el mateix nivell:** la màquina origen serà aquella que tingui el menor nombre de peces.
- **Les màquines que presenten el mateix valor de F_{MAX} tenen nivells diferents:** la màquina origen serà la de nivell inferior.

En el cas que després d'aplicar aquests criteris l'empat es segueixi mantenint, la màquina que tindrà prioritat serà la primera cronològicament.

9.4.3. Elecció de la peça candidata a canviar de nivell

L'aspecte més important en la cerca de noves solucions és escollir la peça jm , la millor peça candidata per canviar de nivell. És a dir, triar la peça que en passar d'un nivell superior a un nivell immediatament inferior doni un instant global de finalització (F_{MAX}) menor.

Per trobar aquesta peça, un cop escollida la màquina origen (mo), es proposa el següent mètode: Sigui JC el conjunt de peces candidates a canviar de nivell. essent $jc \in JC$, una peça candidata, j , a canviar de nivell, és a dir, una peça assignada a una màquina de nivell superior (màquina origen) i que pot ser assignada a una de les màquines nivell immediatament inferior. Es tracta d'escollir la peça que té més incidència a l'hora de reduir el valor de F_{MAX} d'aquest conjunt de peces que conformen les peces de la màquina origen i veure en el nivell immediatament inferior quina màquina escollir les peces de la màquina destí. Aquesta peça serà la millor candidata a canviar de nivell (jm).

La màquina origen serà aquella que tingui un $F_{MAX,k,h}$ major entre les màquines del nivell k i la màquina destí serà una de les màquines del nivell $k+1$.

Per tal d'obtenir millors solucions es definiran dos procediments independents per escollir la peça que canviarà de nivell:

- **Procediment A:** La peça escollida per canviar de nivell serà aquella que faci que la $F_{MAX,kr+1}$ sigui menor.
- **Procediment B:** La peça escollida per canviar de nivell serà aquella que faci que la $F_{MAX,kr+1}$ sigui major.

Com ja s'ha comentat, la seqüència a totes les màquines d'un nivell quedarà determinada per l'heurística utilitzada a la *Fase 0* d'aquest projecte. Per tant, la peça escollida per canviar de nivell serà aquella que, juntament amb la resta de peces existents al nivell $kr+1$ (nivell destí) permeti obtenir un valor F_{MAX} global menor o major, segons si s'utilitza el *Procediment A* o el

Procediment B, després d'haver aplicat a les màquines del nivell destí l'heurística de la *Fase 0*, obtenint la nova programació.

9.4.4. Procediment de ramificació per obtenir una nova solució

Tal i com s'ha anat indicant en aquest apartat de definició de la Fase 1, la metodologia per trobar noves solucions partirà d'un node amb una solució definida. És a dir, una solució on a cada màquina hi haurà una programació amb el valor F_{MAX} i la c corresponents.

A partir de la solució definida, es buscarà la màquina candidata a ser la màquina origen que correspon a un cert nivell k . Dins la seva seqüència de peces es buscarà aquella peça que al canviar al nivell immediatament inferior ofereixi un menor o un major temps de finalització a aquest nivell, segons si s'opta pel *Procediment A* o el *Procediment B*. Per cada peça candidata hi haurà un nou node descendent. Per als nodes descendents es reorganitzaran les seqüències de cada màquina del nivell destí utilitzant l'heurística de la *Fase 0*, i trobant una nova solució amb c^{s+1} .

A la Figura 9.4 es pot observar la representació de l'exploració d'un node.

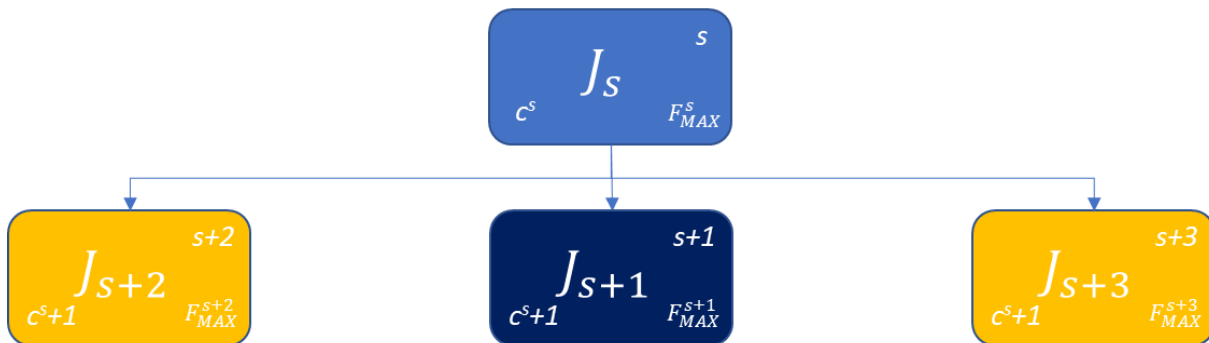


Figura 9.4 Estructura de l'exploració d'un nou node

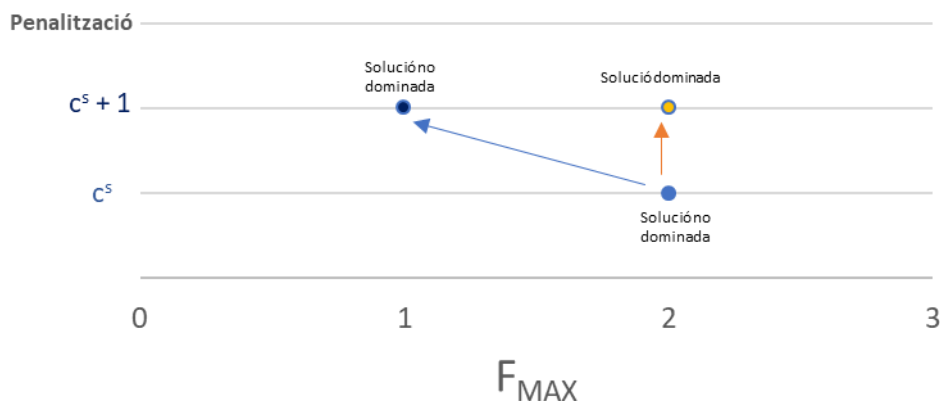


Figura 9.5 Gràfic per explicar l'exploració a partir d'un node

A la Figura 9.5 s'observa la mateixa exploració però en aquest cas es poden veure els valors de F_{MAX} i c . Aquesta figura mostra que la solució $s+1$ (blau fosc) correspon a una solució no dominada. Ja que té un valor de F_{MAX} menor que la solució s i per una penalització c^s+1 . També es pot veure com la solució $s+2$ (taronja clar) és dominada ja que presenta un F_{MAX} igual que la solució s i una penalització amb un valor de c^s+1 .

9.4.5. Exemple de la Fase 1

A continuació es planteja un exemple que il·lustra el procés de canvi de nivell d'una peça on s'obté una nova solució no dominada:

Exemple de la Fase 1 per el cas de 4 màquines i dos nivells: ($m_1=2, m_2=2$)

Si es tenen les següents peces amb el nivell i els corresponents temps r_j, p_j i q_j :

Peça	$j1$	$j2$	$j3$	$j4$	$j5$	$j6$	$j7$	$j8$
l	1	2	2	1	1	2	1	2
r_j	3	1	7	4	6	3	3	1
p_j	5	4	9	5	6	5	8	7
q_j	5	3	3	2	6	4	7	5

Es parteix de la solució 2 (J^2):

$$J^2 = [(j2, j1, j7), (j8, j5, j4), (j3), (j6)]$$

$$c = 2$$

$$F_{MAX1,1} = 25$$

$$F_{MAX1,2} = 21$$

$$F_{MAX2,1} = 19$$

$$F_{MAX2,2} = 12$$

Nivell de referència (kr) = 1 \rightarrow $F_{MAX} = F_{MAX1,1} = 25$

Màquina origen = 1,1

En aquest cas només queda una peça candidata a la màquina origen, ja que només hi ha una peça de nivell 2 a aquesta màquina ($j2$) \rightarrow peça candidata a canviar de nivell = $j2$

Tant el *Procediment A* com el *Procediment B* presentaran la mateixa peça candidata a canviar de nivell: $j2$.

Peces al nivell 2: $SJ_k = \{j2, j3, j6\}$

S'aplica l'heurística de la Fase 0.

Peça	$j2$	$j3$	$j6$
r_j	1	7	3
p_j	4	9	5
q_j	3	3	4

Fase A:

$$J_R = \{j2\}, \quad J_Q = \{\}, \quad J_M = \{j3, j6\}$$

Fase B:

$$J_{2,h} = [(j2, j3), (j6)]$$

$$c = 3$$

$$F_{MAX\ 2} = 19$$

Després de la iteració:

Solució 3

$$J^3 = [(j1, j7), (j8, j5, j4), (j2), (j3, j6)]$$

$$c = 3$$

$$F_{MAX1,1} = 23$$

$$F_{MAX1,2} = 21$$

$$F_{MAX2,1} = 19$$

$$F_{MAX2,2} = 12$$

Resultat final:

$$F_{MAX}^3 = 23$$

$$J^3 = [(j1, j7), (j8, j5, j4), (j2), (j3, j6)]$$

$$c = 3$$

9.4.6. Procediment heurístic per resoldre el cas de $JC \equiv 0$ a la màquina origen (*mo*)

La naturalesa de l'exploració arborescent comporta que pugui haver certs punts on no es troba cap peça candidata a canviar de nivell a la màquina origen, nodes on a la màquina origen (*mo*) no s'hi troba cap peça de nivell superior a *kr*. És a dir, la màquina origen presenta un conjunt buit de peces candidates a canviar de nivell.

Per aquestes situacions s'ha dissenyat una procediment per continuar amb l'exploració arborescent fins arribar al seu FI.

Quan es troba una màquina origen amb el conjunt $JC \equiv 0$, es marca la peça j del nivell kr sigui la màquina que sigui amb p_j major. Seguidament la peça marcada (peça amb major temps de procés, p_j) es canvia de nivell, augmentant la penalització del sistema. Un cop redistribuïda la peça amb major p_j al nivell $kr+1$, s'aplica l'heurística de la Fase 0 amb les màquines i les peces que encara es troben a kr .

Un cop realitzat el canvi de node utilitzant el procediment de la Fase 2, es segueix l'exploració arborescent amb la Fase 1.

El procediment es pot descriure de la següent forma:

Iteració:

- **Si hi ha peces a les màquines de nivell kr candidates a canviar de nivell** (peces de nivell major que kr):
 - A. Es marca la peça j de kr amb p_j major.
 - B. Es canvia de nivell la peça marcada al pas **A** augmentant la penalització en un punt.
 - C. S'aplica a les peces de kr l'heurística de la *Fase 0*
 - D. El procés arborescent continua amb la Fase 1 a partir d'aquest nou node.
- **Si no hi ha peces a kr candidates a canviar de nivell** (peces de nivell igual a kr):
El procés arborescent ha arribat al seu final

Amb aquesta iteració s'aconsegueix que el procés arborescent s'acabi quan no queden peces candidates a canviar de nivell al grup de màquines que té un instant de finalització F_{MAX} més gran. És a dir quan F_{MAX} ja no es pot reduir independentment dels canvis de peces entre els nivells. O bé quan la màquina origen és una màquina del nivell més alt g i per tant les seves peces no poden canviar de nivell ja que no existeix un nivell destí.

9.5.Fase 2: millora de solucions

A la Fase 2 es busca explorar altres branques que durant la Fase 1 no s'han explorat. Aquestes branques podrien permetre trobar noves solucions no dominades. Per aquest motiu durant la Fase 1 es marquen els nodes on el nivell origen (kr) del canvi de peça és diferent del seu node ascendent.

En aquesta Fase 2, per cada node marcat es permet realitzar dues noves iteracions, és a dir, es pot empitjorar en la primera per millorar en la segona i veure si la solució trobada és millor que

les ja obtingudes. En cas afirmatiu, es torna a aplicar la Fase 1 per analitzar aquesta nova branca d'exploració; si no, s'acaba l'exploració del node analitzat.

La Fase 2 segueix el mateix procediment a l'hora d'explorar noves branques que a la Fase 1, amb l'excepció del moment d'escollir la peça que definitivament passarà de la màquina origen (m_o) al nivell destí ($kr + 1$). En aquesta fase, per escollir la peça que canviarà de nivell i el seu node, en comptes d'utilitzar el *Procediment A* o el *Procediment B*, es pren el criteri d'agafar la peça que es troba dins el conjunt AJ i que presenta un temps de procés (p_j) major.

L'estructura genèrica de la Fase 2 per ampliar l'exploració respecte la Fase 1 és:

FASE 2 – Exploració de branques alternatives millors

PER CADA node marcat a la Fase 1:

Aplicar en aquest node dues iteracions, incrementant la penalització dues unitats.

1. La primera iteració amb el criteri d'escollir la peça j que presenta un temps p_j major
 2. La segona mitjançant el *Procediment A* o el *Procediment B* de la Fase 1 aplicat al node obtingut després de la primera iteració
- **Si** la solució està dominada per la solució amb millor solució trobada per la mateixa penalització, aquest node està explorat.
 - **En cas contrari** → Es continua l'exploració arborescent explorant des d'aquest nou node aplicant la Fase 1

A l'Annex C, es detalla el procediment que s'ha utilitzat per decidir el criteri d'elecció de la peça que realitza el canvi de nivell en aquesta Fase 2.

9.6. Algorismes utilitzats

Tots els algorismes presentats en aquest capítol es basaran en l'estructura dels procediments de la Fase 1 (*Procediment A* i *Procediment B*). És a dir, per triar la millor peça candidata (jm) a canviar de nivell per generar una solució descendent a partir d'una solució d'un node es

realitzarà el *Procediment A* o el *Procediment B* explicat a l'apartat 9.4. Primer es defineixen les heurístiques de la Fase 1 per, posteriorment, aplicar-hi la Fase 2 i provar d'obtenir un millor conjunt de solucions no dominades. Per tant, en tots els procediments s'aplicarà la Fase 0 i la Fase 1, però no en tots s'aplicarà la Fase 2. Hi haurà heurístiques on aquesta no s'aplicarà.

Les heurístiques es poden classificar en els següents grups:

- **Heurístiques de la Fase 1:** es caracteritzen per fer un únic pas cada vegada que es vol explorar un nou node. Es fa una exploració tipus *greedy*, tractant de reduir el valor de F_{MAX} augmentant en una unitat la penalització.
- **Heurístiques de la Fase 2:** es caracteritzen per incorporar la Fase 2 als procediments de la Fase 1 realitzant dos passos, en certes ocasions, per explorar un nou node. D'aquesta manera es pretén tenir una visió a un major termini tenint en compte si el moviment realitzat serà profitós per al moviment de la següent iteració o no.

9.6.1. Resum algorismes

A la Taula 9.1 queden resumides la nomenclatura de les heurístiques i procediments que permetran trobar solucions no dominades i les seves principals diferències:

		Variants Fase 1	Fase 2
Algorismes	1A	Fase 1 tipus A	
	1B	Fase 1 tipus B	
	2A	Fase 1 tipus A	Fase 2
	2B	Fase 1 tipus B	Fase 2

Taula 9.1 Resum de les heurístiques

9.6.2. Algorisme de partida 1A i 1B

L'algorisme de partida és el procediment de la Fase 1 plantejat a l'apartat 9.4. Es tracta de l'algorisme més senzill i més restrictiu, ja que és la que té menor grau d'exploració.

Es distingiran dos tipus d'algorismes de partida, l'algorisme **1A** i l'algorisme **1B** corresponents al *Procediment A* i al *Procediment B* respectivament, de la Fase 1.

Representació de l'exploració de la Fase 1

A continuació a la Figura 9.6 es representa l'estructura d'exploració arborescent de la Fase 1, corresponent a la totalitat dels algorismes *1A* i *1B* i comuna en la resta d'algorismes. En aquesta fase es calculen els valors de F_{MAX} per cada peça candidata a canviar de nivell (línies

discontinues) i s'escull la millor candidata jm que provoqui un $F_{MAX,kr+1}$ menor o major (segons si es tracta de 1A o 1B) per cada nivell.

Per tal de simplificar la il·lustració s'ha reduït el contingut del node, tal i com s'havia definit a la Figura 9.2, i s'ha indicat amb el color blau els nodes escollits.

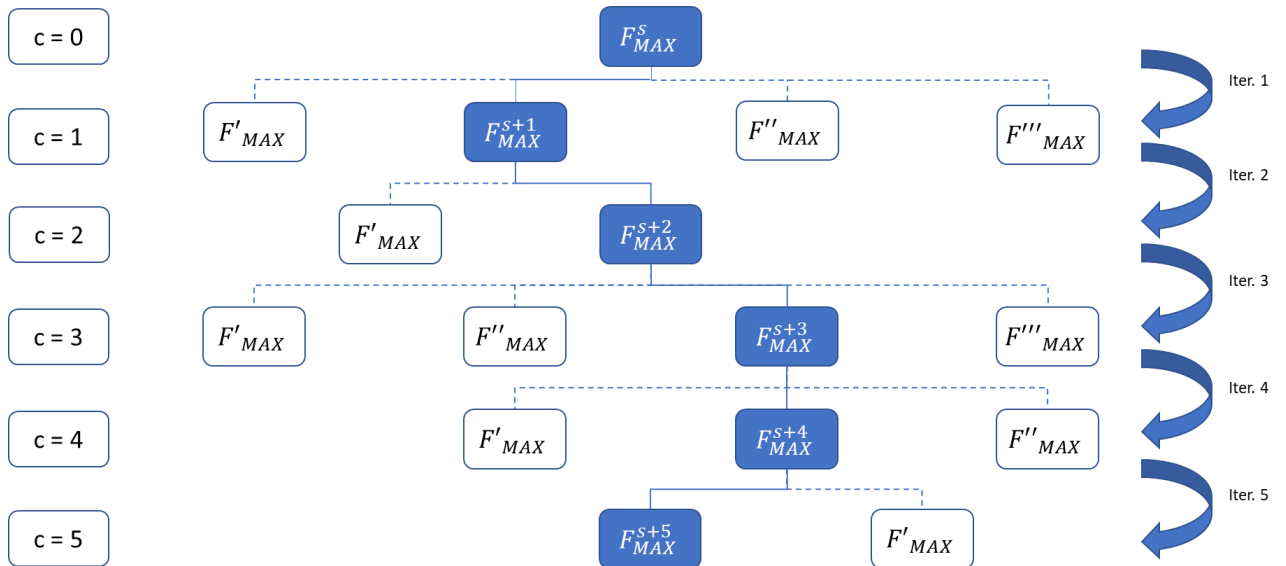


Figura 9.6 Representació de l'arbre de solucions parcial durant la Fase 1

Formalització dels algorismes 1A i 1B

A continuació es detalla el procediment dels algorismes 1A i 1B:

Fase 0 – Trobar la seqüència inicial

Entrada: n peces amb temps de pre-procés r_j , temps de procés p_j , temps de post procés q_j , i nivell l_j , ($j = 1 \dots n$) i m màquines amb nivell lm_i , ($l_i, lm_i = 1 \dots g$)

Sortida: Una llista amb una solució potencialment eficient

1. $S \leftarrow \emptyset$
2. $AJ_1 = \{1, \dots, n\}, AJ_2 = 0, \dots, AJ_g = 0$
3. $s_0 = (F_{MAX}^0, c^0, J^0) \leftarrow$ Procediment Fase 0
4. $c = \emptyset$
5. $S \leftarrow S \cup (F_{MAX}^0, 0, J^0)$
6. $mo \in MG_1$
7. retorna S

La Fase 0 troba una solució amb penalització $c=0$ amb totes les peces situades a les màquines de nivell alt.

Fase 1 – Generació de solucions amb c creixent

Entrada: S

Sortida: Una llista de solucions no dominades

1. *Pila B* (pila de nodes marcats)
2. **while** $mo \in MG_g = False$ **do**
3. **if** $\exists jc$
4. **if** $\exists l_{kr+1}$ a les màquines MG_{kr}
5. return S
6. **else**
7. Aplicar el procediment de la Fase 0 al nivell kr amb JM_{kr} i MG_{kr}
8. $s_{it} = (F_{MAX}^{it}, c^{it}, J^{it}) \leftarrow$ Heurística Fase 1. Seleccionar la millor peça $jm \in A_{J_{kr}} \mid F_{MAX_{kr+1}}$ es minimitzi o maximitzi (segons Procediment A o Procediment B).
Transferir la peça jm a les màquines $\in MG_{kr+1}$
9. $S \leftarrow S \cup (F_{MAX}^{it}, c^{it}, J^{it})$
10. new mo
11. $MS \leftarrow$ Borar les solucions dominades de S
12. retorna MS

9.6.3. Algorismes 2A i 2B

A partir dels algorismes de partida *IA* i *IB* s'han dissenyat dos nous algorismes que permeten tenir un major grau d'exploració i, per tant trobar noves solucions no dominades. Per aquest motiu, mantenint les mateixes Fase 0 i Fase 1 s'ha incorporat la Fase 2 descrita a l'apartat 9.5.

Igual que en els algorismes *IA* i *IB* també hi haurà dos tipus d'algorismes derivats de la Fase 2. Aquests dos algorismes seran el **2A** i el **2B** i consisteixen en buscar més nodes aplicant la Fase 2 als arbres obtinguts per l'algorisme *IA* i l'algorisme *IB* respectivament. L'exploració d'aquests algorismes consisteix en partir dels nodes marcats i aplicar-hi dues noves iteracions tenint en compte el criteri descrit a l'apartat 9.5. Un cop realitzades aquestes noves iteracions i

si s'han trobat noves solucions no dominades es continua el procés arborescent amb el procediment de la Fase 1.

Representació de l'exploració de la Fase 2

A la Figura 9.7 es representa l'exploració que es duu a terme en els algorismes 2A i 2B. De color gris es mostren les solucions no dominades trobades mitjançant la Fase 1 i de color taronja fosc el node marcat. La Fase 2 parteix d'aquest node i hi aplica dues noves iteracions, la primera tenint en compte el criteri descrit a l'apartat 9.5 (el node que s'obté és el marcat en taronja clar) i la segona utilitzant el procediment de la Fase 1 (node blau).

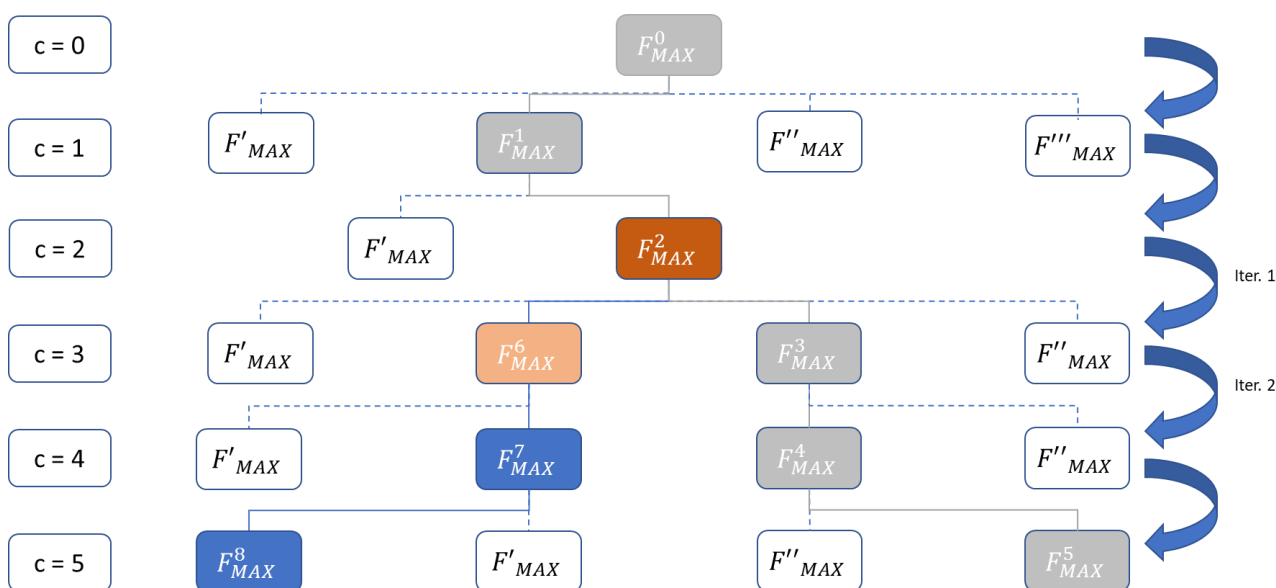


Figura 9.7 Representació de l'arbre de solucions parcial pels algorismes 2A i 2B

Considerant la solució 7 no dominada es segueix realitzant l'exploració arborescent a partir d'aquest node utilitzant els procediments de la Fase 1 tenint en compte que durant aquest nou procediment arborescent es poden marcar més nodes i tornar a realitzar el procediment de la Fase 2 en ells.

D'altra banda, si aquesta solució està dominada per una altra amb el mateix grau de penalització es deixa d'explorar la branca.

Formalització dels algorismes 2A i 2B

A continuació es detalla el procediment dels algorismes 2A i 2B:

Fase 0 – Trobar la seqüència inicial

Entrada: n peces amb temps de pre-procés r_j , temps de procés p_j , temps de post procés q_j , i nivell l_j , ($j = 1 \dots n$) i m màquines amb nivell lm_i , ($l_j, lm_i = 1 \dots g$)

Sortida: Una llista amb una solució potencialment eficient

1. $S \leftarrow \emptyset$
2. $AJ_1 = \{1, \dots, n\}, AJ_2 = 0, \dots, AJ_g = 0$
3. $s_0 = (F_{MAX}^0, c^0, J^0) \leftarrow \text{Procediment Fase 0}$
4. $c = \emptyset$
5. $S \leftarrow S \cup (F_{MAX}^0, 0, J^0)$
6. $mo \in MG_1$
7. *retorna* S

El procediment de la Fase 0 és idèntic a l'utilitzat en els algorismes 1A i 1B.

La part marcada en color més clar es correspon en la part de la Fase 1 que es diferencia de la Fase 1 dels algorismes 1A i 1B. En els algorismes 2A i 2B es marquen els nodes on i ha un canvi de nivell de la màquina origen (mo) i es procedeix a cridar la Fase 2.

Fase 1 – Generació de solucions amb c creixent

Entrada: S

Sortida: Una llista de solucions potencialment eficients

1. *Pila* B (pila de nodes marcats)
2. **while** $mo \in MG_g = \text{False}$ **do**
3. **if** $\nexists jc$
4. **if** $\nexists l_{kr+1}$ a les màquines MG_{kr}
5. *return* S
6. **else**
7. Aplicar el procediment de la Fase 0 al nivell kr amb JM_{kr} i MG_{kr}

8. $s_{it} = (F_{MAX}^{it}, c^{it}, J^{it}) \leftarrow$ *Heurística Fase 1. Seleccionar la millor peça $jm \in A_{J_{kr}} \mid F_{MAX_{kr+1}}$ es minimitzi o maximitzi (segons Procediment A o Procediment B). Transferir la peça jm a les màquines $\in MG_{kr+1}$*
9. $S \leftarrow S \cup (F_{MAX}^{it}, c^{it}, J^{it})$
10. *new mo*
11. **if** $mo \notin MG_{kr}^{it}$
12. *Push* ($B \mid s_{it-1}$)
13. **Call Fase 2**
14. *Borar les solucions dominades de S*
15. *retorna S*

Fase 2

Entrada: S i B

Sortida: Una llista de solucions potencialment eficients

1. **while** $IsEmpty(B) = False$ **do**
2. *Pop* ($B \mid s_{it-1}$)
3. $s'_{it} = (F_{MAX}^{it'}, c^{it'}, J^{it'}) \leftarrow$
Heurística Fase 2. Seleccionar la millor peça $jm \in A_{J_{kr}} \mid p$ sigui màxima
4. $s'_{it+1} = (F_{MAX}^{it'+1}, c^{it'} + 1, J^{it'+1}) \leftarrow$ *Heurística Fase 1 a partir del node s'_{it}*
5. $s_{it+1} = (F_{MAX}^{it+1}, c^{it} + 1, J^{it+1}) \leftarrow$ *Heurística Fase 1 a partir del node s_{it}*
6. **if** $F_{MAX}^{it'+1} < F_{MAX}^{it+1}$
7. $S \leftarrow S \cup s'_{it}$
8. $S \leftarrow S \cup s'_{it+1}$
9. *Exit* \leftarrow *Continua amb la Fase 1*
10. **else**
11. *Exit* \leftarrow *Continua amb la Fase 1*

Finalment la Fase 2 fa una exploració a més profunditat.

10. Experiència computacional

Un cop definida l'estructura genèrica a partir de la qual es dissenyaran les heurístiques, es planteja l'experiència computacional per resoldre el problema definit i així poder comparar els resultats obtinguts per cada procediment (*Procediment A* i *Procediment B*). Per cada un dels casos es programarà les heurístiques dissenyades mitjançant el programari Microsoft Visual Studio per poder facilitar i automatitzar els càlculs. Per realitzar l'anàlisi comparativa s'han utilitzat els mateixos exemplars que Jordi Camps (Camps, 2013) per a l'obtenció dels resultats. A l'esmentat treball només s'estudiava només el cas de una màquina per nivell, per tant només s'han aprofitat les dades de peces dels exemplars i s'han generat noves combinacions de màquines per nivell.

D'altra banda s'ha creat un programa que permetrà comparar les solucions obtingudes determinant el percentatge de solucions no dominades per cada algorisme.

10.1. Generació dels exemplars de peces i màquines

Els exemplars que s'han generat s'han construït a partir dels exemplars utilitzats per Jordi Camps (Camps, 2013) en el seu projecte. Per tal d'estudiar diverses quantitats de peces s'han escollit exemplars de 20, 50, 100 i 200 peces. En els dos primers casos, els de 20 i 50 peces s'estudiaran 1.000 exemplars, mentre que en els altres dos, els de 100 i 200 peces, se n'estudiaran 200 ja que en aquests casos el temps de càlcul és elevat i es considera que amb 200 exemplars es poden obtenir valors significatius.

10.1.1. Combinacions de màquines

Cada un dels exemplars s'estudiarà amb combinacions de màquines diferents, per així poder fer un estudi més complet. En el cas de tres nivells s'estudiarà les diferents composicions de màquines segons l'exemplar estudiat i el nombre total de màquines m del sistema.

		Nombre de màquines m					
		4	5	6	7	9	11
Nombre de peces (n)	20	(2, 1, 1)	(2, 2, 1)	(3, 2, 1)			
	50	(2, 1, 1)	(2, 2, 1)	(3, 2, 1)			
	100	(2, 1, 1)	(2, 2, 1) (3, 1, 1)	(4, 1, 1)	(5, 1, 1) (3, 3, 1)	(4, 4, 1)	(5, 5, 1)
	200	(2, 1, 1)	(2, 2, 1) (3, 1, 1)	(4, 1, 1)	(5, 1, 1) (3, 3, 1)	(4, 4, 1)	(5, 5, 1)

Taula 10.1 Combinacions de màquines en el cas de 3 nivells ($g = 3$), (m_1, m_2, m_3)

Per a més de 3 nivells, només s’estudiaran els casos de 4 i 5 nivells ($g=4$ i $g=5$ respectivament) i es limitarà als exemplars de 100 i 200 peces.

A la Taula 10.2 i a la Taula 10.3 es poden veure les combinacions de màquina per nivell que, en cada cas, s’estudiaran.

		Nombre de màquines m				
		6	8	9	12	16
Nombre de peces (n)	100	(2, 2, 1, 1)	(2, 2, 2, 2)	(3, 3, 2, 1)	(3, 3, 3, 3) (4, 4, 2, 2)	(4, 4, 4, 4)
	200	(2, 2, 1, 1)	(2, 2, 2, 2)	(3, 3, 2, 1)	(3, 3, 3, 3) (4, 4, 2, 2)	(4, 4, 4, 4)

Taula 10.2 Combinacions de màquines en el cas de 4 nivells ($g = 4$), (m_1, m_2, m_3, m_4)

		Nombre de màquines m					
		8	10	12	15	16	20
Nombre de peces (n)	100	(2,2,2,1,1)	(2,2,2,2,2)	(3,3,2,2,2)	(3,3,3,3,3)	(4,4,4,2,2)	(4,4,4,4,4)
	200	(2,2,2,1,1)	(2,2,2,2,2)	(3,3,2,2,2)	(3,3,3,3,3)	(4,4,4,2,2)	(4,4,4,4,4)

Taula 10.3 Combinacions de màquines en el cas de 5 nivells ($g = 5$), (m_1, m_2, m_3, m_4, m_5)

10.1.2. Generació dels exemplars de peces

Els exemplars de peces van ser generats de manera similar a Gharbi i Haouari (Gharbi & Haouari, 2002). En cada cas, els temps de les peces estan generats a partir de distribucions discretes uniformes com es mostra a la Taula 10.4:

	Temps de preprocés (r_j)	Temps de procés (p_j)	Temps de postprocés (q_j)
Distribució discreta uniforme	[1, $K(n/m)$]	[1, 10]	[1, $K(n/m)$]

Taula 10.4 Distribució discreta uniforme dels temps r_j , p_j i q_j

El valor de K és un enter positiu que afecta a les distribucions de temps de pre-procés i post-procés. Per als exemplars de 20 i 50 peces K val 3 ($K=3$) i per a la resta d'exemplars val 5 ($K=5$) (Camps, 2013). Per altre banda, n és el nombre de peces i m el nombre de màquines.

A la Taula 10.5 apareixen els percentatges de peces establerts per a cada nivell, generats també, a partir de distribucions uniformes discretes:

nivell núm. Nivells (g)	nivell				
	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$
3	[20, 30]	[20, 50]	[20, 60]		
4	[20, 25]	[20, 25]	[20, 30]	[20, 40]	
5	[15, 20]	[15, 20]	[15, 20]	[15, 20]	[20, 40]

Taula 10.5 Percentatge de peces segons nivell

A l'Annex A es pot trobar un exemplar de 20 peces per mostrar la forma en que es presenta.

10.2. Estudi dels exemplars

S'han provat els quatre algorismes proposats sobre els diferents exemplars i sobre els diferents nombres de màquines.

Per a cada algorisme per a cada grup d'exemplars i per a cada combinació de màquines s'ha analitzat el nombre de solucions no dominades obtingudes en mitjana i la mitjana dels temps d'execució.

D'altra banda, donats els conjunts de solucions de dues heurístiques X i Y s'ha calculat el percentatge de solucions no dominades corresponents, a partir de l'equació 8.2. Cal recordar que un alt percentatge de solucions no dominades d'una heurística respecte totes les altres indica la major qualitat del procediment de les solucions obtingudes.

10.2.1. Estudi dels exemplars de 20 peces

Un cop executades els quatre algorismes amb els exemplars de 20 i les 4 combinacions de màquines els resultats que s'han obtingut són:

Sol. no dominades	(1,1,1)	(2,1,1)	(2,2,1)	(3,2,1)
1A	11,44	6,25	6,82	4,26
1B	10,52	5,00	5,65	3,79
2A	11,14	6,20	6,83	4,25
2B	11,22	5,38	5,85	3,97

Taula 10.6 Nombre mitjà de solucions no dominades de les heurístiques per als exemplars de 20 peces en el cas de 3 nivells ($g=3$)

A la Taula 10.6 es pot observar que en el cas d'exemplars de 20 peces les dades són força semblants. La principal diferència s'observa en les heurístiques de tipus A. Aquestes presenten un major nombre de solucions no dominades que les de tipus B.

Temps (ms)	(1,1,1)	(2,1,1)	(2,2,1)	(3,2,1)
1A	101,58	24,63	103,25	85,23
1B	111,49	33,16	115,84	102,20
2A	110,24	28,85	108,96	94,32
2B	52,89	26,25	115,92	103,85

Taula 10.7 Temps (ms) mitjà de resolució de les heurístiques per als exemplars de 20 peces en el cas de 3 nivells ($g=3$)

En quan a temps de resolució d'aquests exemplars, a la Taula 10.7 es pot observar com els procediments que obtenen un major nombre de solucions no dominades tenen menor temps d'exploració.

Per altra banda, comparant els algorismes s'obtenen les següents mitjanes de percentatges de solucions no dominades respecte el total entre els diferents algorismes (Taula 10.8):

%	(1,1,1)	(2,1,1)	(2,2,1)	(3,2,1)
1A	40,95%	40,98%	40,63%	35,25%
1B	5,66%	4,83%	4,86%	3,50%
2A	39,05%	40,61%	40,69%	34,94%
2B	5,91%	9,32%	7,99%	6,77%

Taula 10.8 Percentatge de solucions no dominades sobre el total de solucions per als exemplars de 20 peces en el cas de 3 nivells ($g=3$)

A l'Annex B es poden trobar aquests percentatges desglossats per exemplar i algorisme.

A la Taula 10.8 es pot observar com les heurístiques que presenten més solucions no dominades respecte les altres heurístiques són les heurístiques de tipus A. Mirant la representació gràfica, Figura 10.1, es pot observar com, efectivament, els exemplars que segueixen el procediment A

per a la Fase 1 obtenen millors resultats. D'altra banda s'observa que en els exemplars de 20 peces la Fase 2 no aporta una millora significativa en els resultats.

En els exemplars de 20 peces es pot observar que hi ha una correlació entre la quantitat de solucions no dominades que obté cada algorisme i la qualitat d'aquesta respecte les altres.

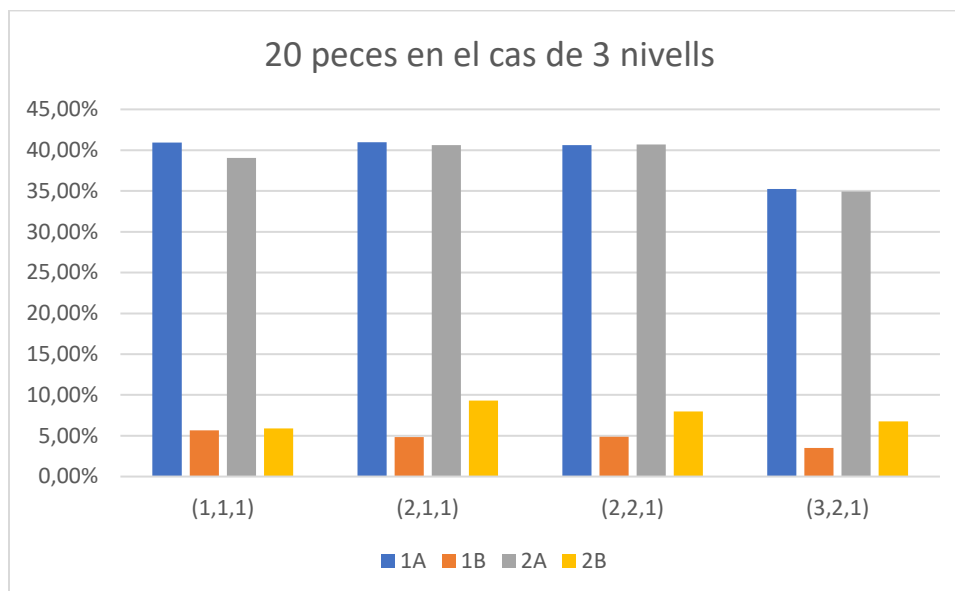


Figura 10.1 Comparació de la no dominància entre heurístiques per als exemplars de 20 peces en el cas de 3 nivells ($g=3$)

10.2.2. Estudi dels exemplars de 50 peces

Un cop executades els quatre algorismes amb els exemplars de 50 i les 4 combinacions de màquines els resultats que s'han obtingut són els de la Taula 10.9:

Sol. no dominades	(1,1,1)	(2,1,1)	(2,2,1)	(3,2,1)
1A	27,10	12,78	13,92	6,98
1B	25,03	9,15	9,33	5,13
2A	26,87	12,61	13,86	7,12
2B	26,80	9,97	10,18	5,75

Taula 10.9 Nombre mitjà de solucions no dominades de les heurístiques per als exemplars de 50 peces en el cas de 3 nivells ($g=3$)

A la Taula 10.9 es pot observar com per als casos on hi ha un menor nombre de màquines el nombre de solucions no dominades obtingudes per cada algorisme és similar, exceptuant l'heurística 1B. En canvi, a l'augmentar el nombre de màquines es torna a observar (igual que en els casos de 20 peces) que els algorismes que han seguit el *procediment A* durant la seva Fase 1 obtenen un millor nombre de solucions no dominades.

Temps (ms)	(1,1,1)	(2,1,1)	(2,2,1)	(3,2,1)
1A	352,09	205,93	217,53	159,65
1B	333,15	251,11	263,66	209,29
2A	322,59	219,12	232,20	171,54
2B	341,95	264,57	282,75	215,09

Taula 10.10 Temps (ms) mitjà de resolució de les heurístiques per als exemplars de 50 peces en el cas de 3 nivells ($g=3$)

En quan als temps de resolució de cada exemplar en els casos de 50 peces, en els algorismes que s'obté un major nombre de solucions no dominades el temps de resolució és lleugerament menor (Taula 10.10).

%	(1,1,1)	(2,1,1)	(2,2,1)	(3,2,1)
1A	54,43%	52,88%	52,47%	45,39%
1B	9,09%	7,31%	5,66%	3,94%
2A	50,45%	53,61%	53,90%	47,24%
2B	5,71%	12,04%	11,50%	10,96%

Taula 10.11 Percentatge de solucions no dominades sobre el total de solucions per als exemplars de 50 peces en el cas de 3 nivells ($g=3$)

A la Taula 10.11 i a la Figura 10.2 es pot observar com la qualitat dels algorismes segueix el mateix patró que en els casos de 20 peces. Els algorismes que utilitzen el *procediment A* presenten millors solucions que els que utilitzen el *procediment B*. En el cas de 50 peces es pot veure que l'algorisme 2A té més facilitat per millorar els resultats de la 1A en els exemplars on hi ha més màquines al sistema.

Igual que en els exemplars de 20 peces, es pot observar que hi ha una correlació entre la quantitat de solucions no dominades que obté cada algorisme i la qualitat d'aquesta respecte les altres.

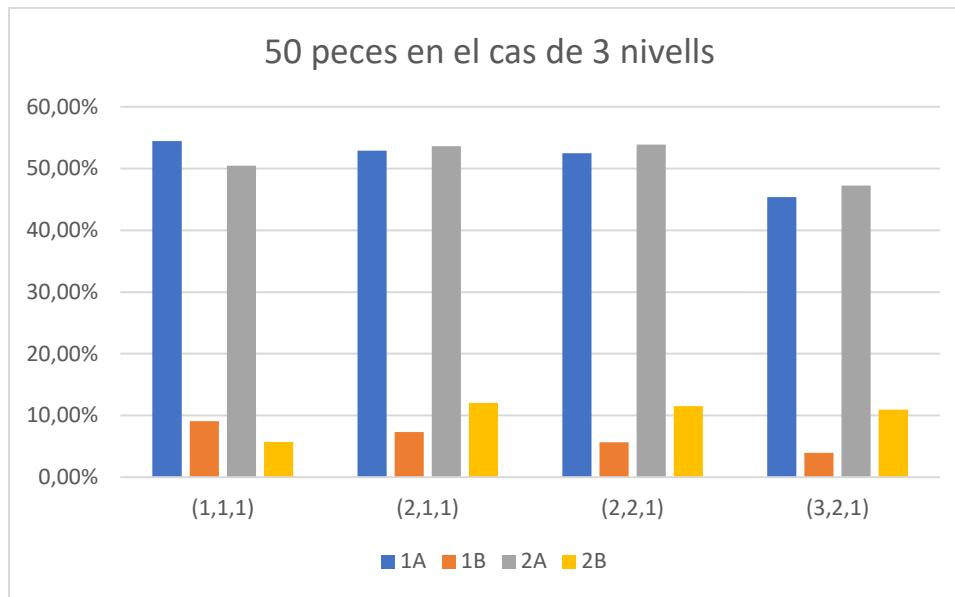


Figura 10.2 Comparació de la no dominància entre heurístiques per als exemplars de 50 peces en el cas de 3 nivells

10.2.3. Estudi dels exemplars de 100 peces

Un cop executades els quatre algorismes amb els exemplars de 100 i diferents combinacions de màquines per a 3,4 i 5 nivells. Els resultats que s’han obtingut estan a la Taula 10.12, Taula 10.13 i Taula 10.14:

Sol. no dominades	(2,1,1)	(2,2,1)	(3,1,1)	(4,1,1)	(3,3,1)	(5,1,1)	(4,4,1)	(5,5,1)
1A	24,28	25,09	11,20	5,66	10,80	3,86	5,82	4,07
1B	15,66	15,89	6,80	4,26	5,93	3,75	3,93	3,54
2A	24,10	25,41	11,33	5,85	11,14	3,92	6,17	4,08
2B	17,32	16,98	7,91	4,90	7,11	3,42	4,56	3,99

Taula 10.12 Nombre mitjà de solucions no dominades de les heurístiques per als exemplars de 100 peces en el cas de 3 nivells (g=3)

Sol. no dominades	(2,2,1,1)	(2,2,2,2)	(3,3,2,1)	(3,3,3,3)	(4,4,2,2)	(4,4,4,4)
1A	28,81	29,62	12,78	13,48	7,00	7,23
1B	18,21	17,79	7,30	7,45	4,58	4,79
2A	28,62	29,26	13,26	13,61	7,19	7,30
2B	20,21	20,35	9,07	9,37	5,57	6,00

Taula 10.13 Nombre mitjà de solucions no dominades de les heurístiques per als exemplars de 100 peces en el cas de 4 nivells (g=4)

Sol. no dominades	(2,2,2,1,1)	(2,2,2,2,2)	(3,3,3,2,2)	(3,3,3,3,3)	(4,4,4,2,2)	(4,4,4,4,4)
1A	30,83	31,17	14,20	14,55	7,42	7,44
1B	19,10	18,89	8,07	8,26	5,05	5,17
2A	29,97	30,26	14,32	14,52	7,48	7,50
2B	22,33	22,33	10,13	10,40	6,17	6,21

Taula 10.14 Nombre mitjà de solucions no dominades de les heurístiques per als exemplars de 100 peces en el cas de 5 nivells (g=5)

A diferència dels exemplars de 20 i 50 peces, als exemplars de 100 peces es pot observar com, per $g=3$ (Taula 10.14), en els casos on hi ha un menor nombre de màquines, les heurístiques de tipus A obtenen un nombre major de solucions no dominades. En canvi, en els casos on en nombre de màquines és major, el nombre de solucions no dominades que obtenen cadascun dels algorismes és més semblant.

En els casos de $g=4$ (Taula 10.12) i $g=5$ (Taula 10.14) s'observa com les heurístiques de tipus A troben en totes les diferents combinacions de màquines un nombre major de solucions no dominades que les heurístiques de tipus B.

Temps (ms)	(2,1,1)	(2,2,1)	(3,1,1)	(4,1,1)	(3,3,1)	(5,1,1)	(4,4,1)	(5,5,1)
1A	839,73	860,67	623,88	565,47	220,09	94,75	555,34	554,78
1B	991,22	1037,19	810,16	759,90	509,48	293,74	725,17	754,51
2A	894,75	930,16	683,90	616,21	441,88	99,51	597,70	583,26
2B	974,09	1047,08	793,31	718,06	499,03	224,73	657,83	688,83

Taula 10.15 Temps (ms) mitjà de resolució de les heurístiques per als exemplars de 100 peces en el cas de 3 nivells ($g=3$)

Temps (ms)	(2,2,1,1)	(2,2,2,2)	(3,3,2,1)	(3,3,3,3)	(4,4,2,2)	(4,4,4,4)
1A	604,11	948,36	689,39	924,57	774,77	720,31
1B	954,53	1158,97	899,79	1114,21	989,19	904,90
2A	788,07	1047,65	803,94	980,94	840,53	776,19
2B	999,48	1181,36	895,79	1112,60	948,31	865,49

Taula 10.16 Temps (ms) mitjà de resolució de les heurístiques per als exemplars de 100 peces en el cas de 4 nivells ($g=4$)

Temps (ms)	(2,2,2,1,1)	(2,2,2,2,2)	(3,3,3,2,2)	(3,3,3,3,3)	(4,4,4,2,2)	(4,4,4,4,4)
1A	1234,48	1771,77	1238,29	1336,92	1113,51	1174,32
1B	1731,52	2110,39	1597,18	1715,38	1533,68	1649,61
2A	1492,42	1912,66	1341,71	1419,88	1160,24	1241,52
2B	1782,40	2167,76	1519,86	1632,82	1363,22	1421,84

Taula 10.17 Temps (ms) mitjà de resolució de les heurístiques per als exemplars de 100 peces en el cas de 5 nivells ($g=5$)

En quan als temps de resolució de cada exemplar, en els casos de 100 peces, en els algorismes que s'obté un major nombre de solucions no dominades els temps de resolució són lleugerament menors. Igual que en els casos de 20 i 50 peces.

Comparant els temps de resolució dels exemplars segons el nombre de nivells que tenen, siguin 3 (Taula 10.15), 4 (Taula 10.16) i 5 (Taula 10.17), es pot veure que com òbviament a major nombre de nivells g , més llargs són els temps de resolució de cada exemplar.

%	(2,1,1)	(2,2,1)	(3,1,1)	(4,1,1)	(3,3,1)	(5,1,1)	(4,4,1)	(5,5,1)
1A	60,02%	58,62%	52,34%	39,73%	52,60%	35,59%	39,84%	37,26%
1B	7,73%	6,84%	5,22%	3,56%	3,49%	2,16%	1,97%	1,26%
2A	61,54%	59,77%	53,74%	40,14%	54,89%	36,58%	41,69%	37,25%
2B	11,81%	11,33%	12,86%	10,79%	12,23%	9,40%	9,02%	6,14%

Taula 10.18 Percentatge de solucions no dominades sobre el total de solucions per als exemplars de 100 peces en el cas de 3 nivells (g=3)

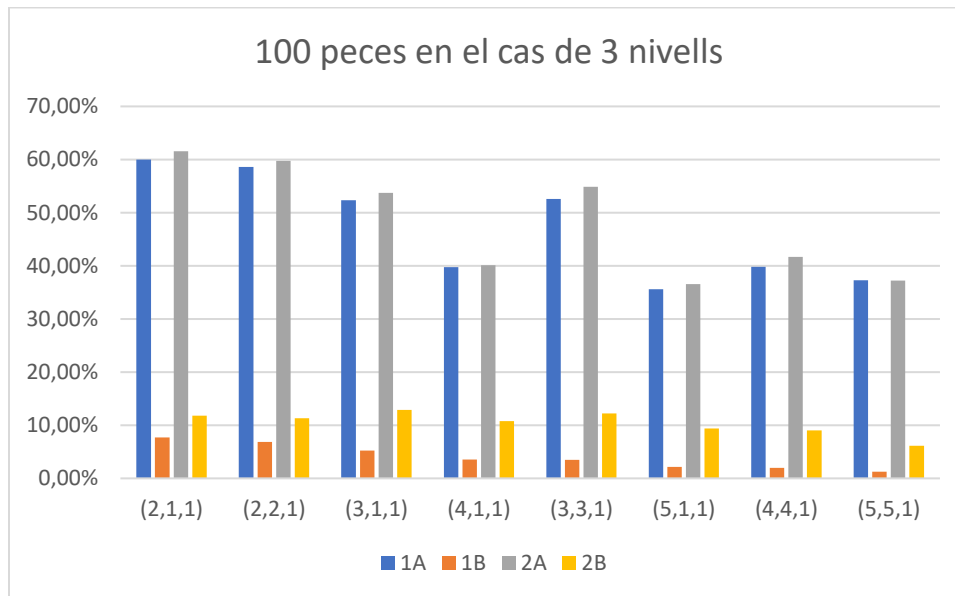


Figura 10.3 Comparació de la no dominància entre heurístiques per als exemplars de 100 peces en el cas de 3 nivells (g=3)

%	(2,2,1,1)	(2,2,2,2)	(3,3,2,1)	(3,3,3,3)	(4,4,2,2)	(4,4,4,4)
1A	62,02%	63,96%	55,72%	56,96%	42,76%	43,18%
1B	6,32%	5,05%	3,40%	3,02%	1,33%	1,56%
2A	63,45%	63,84%	57,66%	57,82%	44,17%	43,86%
2B	13,81%	13,28%	15,93%	15,99%	11,38%	11,76%

Taula 10.19 Percentatge de solucions no dominades sobre el total de solucions per als exemplars de 100 peces en el cas de 4 nivells (g=4)

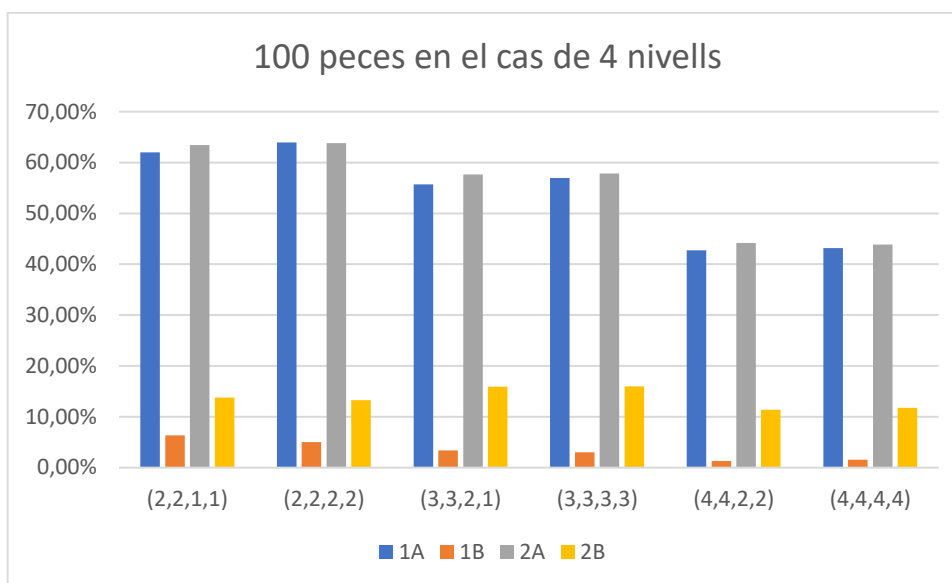


Figura 10.4 Comparació de la no dominància entre heurístiques per als exemplars de 100 peces en el cas de 4 nivells (g=4)

%	(2,2,2,1,1)	(2,2,2,2,2)	(3,3,3,2,2)	(3,3,3,3,3)	(4,4,4,2,2)	(4,4,4,4,4)
1A	64,79%	65,31%	57,76%	58,10%	43,79%	43,77%
1B	5,22%	4,91%	3,44%	3,27%	1,26%	1,37%
2A	64,33%	64,61%	58,43%	58,45%	44,90%	44,99%
2B	15,09%	14,91%	16,63%	16,68%	12,14%	12,07%

Taula 10.20 Percentatge de solucions no dominades sobre el total de solucions per als exemplars de 100 peces en el cas de 5 nivells (g=5)

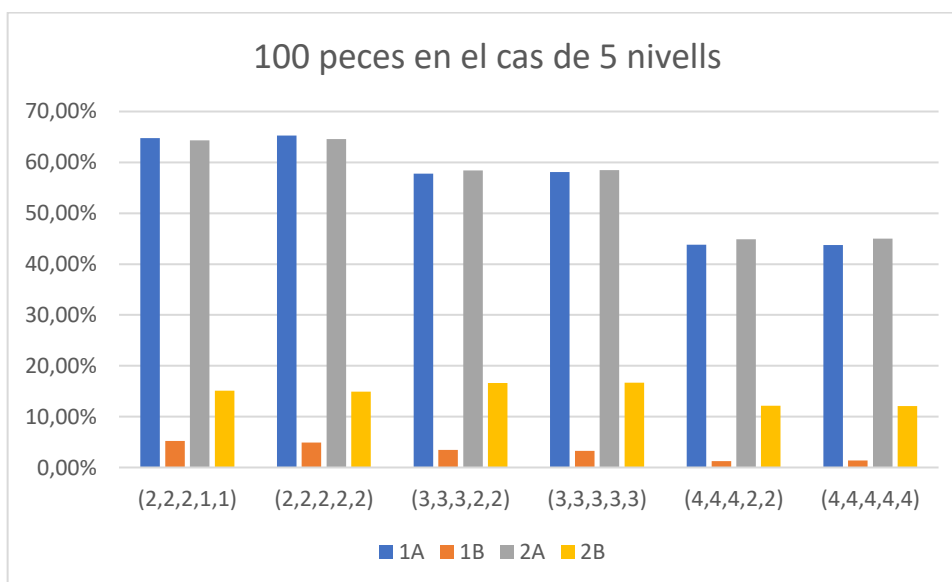


Figura 10.5 Comparació de la no dominància entre heurístiques per als exemplars de 100 peces en el cas de 5 nivells (g=5)

En la comparació dels algorismes per al cas de 100 peces, siguin 3 nivells (Figura 10.3, Taula 10.18), 4 nivells (Figura 10.4, Taula 10.19) i 5 nivells (Figura 10.5, Taula 10.20), s'observen diferents tendències a remarcar:

- La major qualitat de les heurístiques de tipus A respecte a les de tipus B es dona quan el nombre màxim de màquines en un dels nivells és menor.
- El percentatge de solucions no dominades d'un algorisme respecte les altres disminueix a mesura que augmenta el nombre màxim de màquines. És a dir, com més alt és el nombre màxim de màquines els diferents algorismes comparteixen més solucions entre elles. Per tant presenten menys solucions diferents. Aquest fet s'aprecia sobretot als gràfics de barres.
- Els algorismes que incorporen la Fase 2 milloren, en la majoria dels casos, les que no la incorporen. Això és degut a què amb un nombre gran de peces, com ara 100, el nombre de nodes creix significativament i una major exploració de nodes permet obtenir més solucions no dominades.
- A causa de la baixa qualitat de l'heurística 1B, la incorporació de la Fase 2, algorisme 2B, millora molt significativament les solucions que troba.
- La qualitat dels algorismes no varia segons el nombre de nivells. Els algorismes de més qualitat ho són, majoritàriament, en tots els casos.

10.2.4. Estudi dels exemplars de 200 peces

Un cop executades els quatre algorismes amb els exemplars de 100 i les 4 combinacions de màquines els resultats que s'han obtingut estan a la Taula 10.21 (3 nivells), Taula 10.22 (4 nivells) i Taula 10.23 (5 nivells):

Sol. no dominades	(2,1,1)	(2,2,1)	(3,1,1)	(4,1,1)	(3,3,1)	(5,1,1)	(4,4,1)	(5,5,1)
1A	45,10	46,33	17,75	8,43	17,75	4,13	8,41	4,62
1B	26,85	25,94	10,22	5,55	8,44	4,63	4,66	3,77
2A	44,52	46,60	20,35	9,04	18,40	4,29	8,61	4,67
2B	29,91	29,48	12,31	6,53	10,05	5,39	5,70	4,55

Taula 10.21 Nombre mitjà de solucions no dominades de les heurístiques per als exemplars de 200 peces en el cas de 3 nivells ($g=3$)

Sol. no dominades	(2,2,1,1)	(2,2,2,2)	(3,3,2,1)	(3,3,3,3)	(4,4,2,2)	(4,4,4,4)
1A	46,37	46,19	18,34	18,74	8,88	9,61
1B	25,78	25,32	8,39	8,33	4,62	4,83
2A	46,36	46,51	18,86	19,00	9,40	9,98
2B	28,59	28,86	10,57	10,69	5,87	6,26

Taula 10.22 Nombre mitjà de solucions no dominades de les heurístiques per als exemplars de 200 peces en el cas de 4 nivells ($g=4$)

Sol. no dominades	(2,2,2,1,1)	(2,2,2,2,2)	(3,3,3,2,2)	(3,3,3,3,3)	(4,4,4,2,2)	(4,4,4,4,4)
1A	49,22	49,49	19,98	20,06	10,08	10,12
1B	28,23	28,39	9,31	9,34	5,26	5,32
2A	50,26	50,37	20,47	20,49	10,50	10,52
2B	31,89	32,00	12,18	12,21	6,93	6,95

Taula 10.23 Nombre mitjà de solucions no dominades de les heurístiques per als exemplars de 200 peces en el cas de 5 nivells ($g=5$)

Tal com passava en els exemplars de 100 peces, a la Taula 10.21 es pot observar com en els casos de tres nivells ($g=3$), on hi ha un menor nombre total de màquines, les heurístiques de tipus A obtenen un nombre molt major de solucions no dominades que les heurístiques de tipus B. En canvi, en els casos on el nombre de màquines total és major, especialment els casos que tenen 5 màquines en algun dels seus nivells, el nombre de solucions no dominades que obtenen cadascun dels algorismes és similar.

En els casos de $g=4$ (Taula 10.22) i $g=5$ (Taula 10.23) es pot observar com els algorismes que s'ha vist que presentaven solucions de major qualitat (1A i 2A) troben gairebé el doble de solucions no dominades que les heurístiques de tipus B.

En quan als temps de resolució de cada exemplar, en els casos de 200 peces (Taula 10.24, Taula 10.25 i Taula 10.26), en els algorismes que s'obté un major nombre de solucions no dominades els temps de resolució són lleugerament menors, igual que en els anteriors casos.

Temps (ms)	(2,1,1)	(2,2,1)	(3,1,1)	(4,1,1)	(3,3,1)	(5,1,1)	(4,4,1)	(5,5,1)
1A	2152,70	2905,13	1466,20	954,90	1159,61	337,15	920,21	930,33
1B	2640,98	3482,91	2503,89	1617,78	1530,05	1068,06	1151,87	1380,49
2A	2271,48	3090,84	1665,89	981,67	1262,73	368,59	948,27	958,39
2B	2751,37	3587,68	2527,79	1508,52	1509,83	802,85	1228,03	1218,07

Taula 10.24 Temps mitjà de resolució de les heurístiques per als exemplars de 200 peces en el cas de 3 nivells ($g=3$)

Temps (ms)	(2,2,1,1)	(2,2,2,2)	(3,3,2,1)	(3,3,3,3)	(4,4,2,2)	(4,4,4,4)
1A	1887,51	2357,86	801,24	1446,64	1038,53	1124,53
1B	2497,57	2891,71	1385,31	1879,25	1442,06	1576,08
2A	2071,33	2561,57	1007,66	1561,45	1112,71	1247,09
2B	2509,19	2965,35	1389,43	1887,13	1343,38	1510,06

Taula 10.25 Temps(ms) mitjà de resolució de les heurístiques per als exemplars de 200 peces en el cas de 4 nivells ($g=4$)

Temps (ms)	(2,2,2,1,1)	(2,2,2,2,2)	(3,3,3,2,2)	(3,3,3,3,3)	(4,4,4,2,2)	(4,4,4,4,4)
1A	2055,10	2693,85	1455,74	1512,85	1392,23	1568,82
1B	2910,85	3429,79	2070,44	2111,32	1808,74	2299,72
2A	2457,71	2982,00	1680,94	1653,45	1526,47	1691,48
2B	3034,07	3544,93	2038,66	2050,45	1983,53	2093,66

Taula 10.26 Temps (ms) mitjà de resolució de les heurístiques per als exemplars de 200 peces en el cas de 5 nivells ($g=5$)

Com en els exemplars de 100 peces, l'increment del nombre de nivells comporta un increment del temps de resolució de cada exemplar.

%	(2,1,1)	(2,2,1)	(3,1,1)	(4,1,1)	(3,3,1)	(5,1,1)	(4,4,1)	(5,5,1)
1A	63,62%	62,34%	57,20%	41,85%	57,39%	34,83%	44,82%	38,53%
1B	9,22%	6,33%	6,71%	5,11%	5,22%	2,47%	2,08%	0,90%
2A	65,70%	64,31%	61,74%	45,15%	59,36%	37,31%	47,40%	38,92%
2B	11,91%	13,95%	16,19%	12,84%	14,92%	11,60%	9,82%	7,95%

Taula 10.27 Percentatge de solucions no dominades sobre el total de solucions per als exemplars de 200 peces en el cas de 3 nivells (g=3)

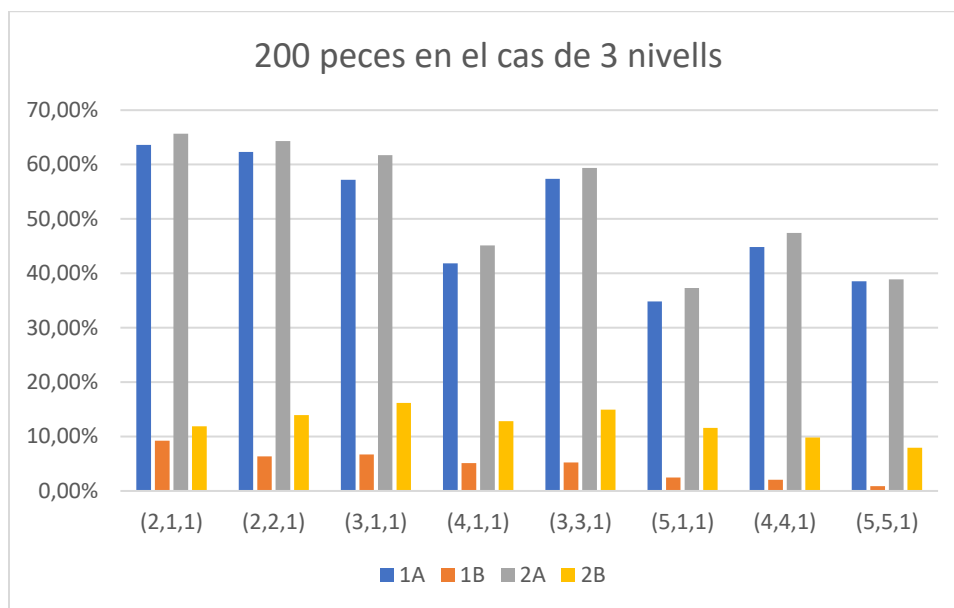


Figura 10.6 Comparació de la no dominància entre heurístiques per als exemplars de 200 peces en el cas de 3 nivells (g=3)

%	(2,2,1,1)	(2,2,2,2)	(3,3,2,1)	(3,3,3,3)	(4,4,2,2)	(4,4,4,4)
1A	63,20%	64,07%	59,11%	60,39%	45,79%	46,91%
1B	6,16%	4,89%	4,16%	3,71%	1,40%	1,48%
2A	64,55%	65,04%	60,62%	60,45%	47,99%	48,75%
2B	13,43%	12,89%	15,31%	15,47%	10,63%	11,38%

Taula 10.28 Percentatge de solucions no dominades sobre el total de solucions per als exemplars de 200 peces en el cas de 4 nivells (g=4)

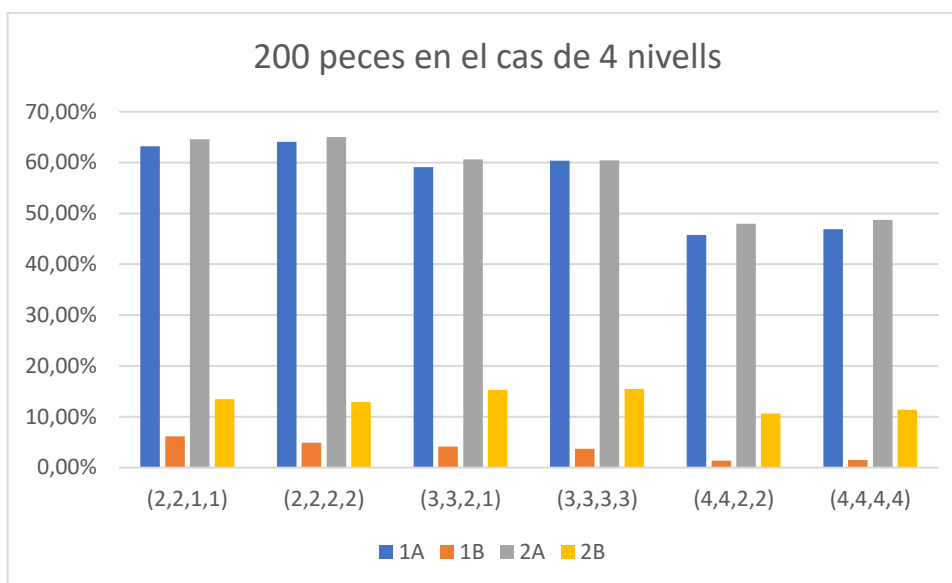


Figura 10.7 Comparació de la no dominància entre heurístiques per als exemplars de 200 peces en el cas de 4 nivells (g=4)

%	(2,2,2,1,1)	(2,2,2,2,2)	(3,3,3,2,2)	(3,3,3,3,3)	(4,4,4,2,2)	(4,4,4,4,4)
1A	65,26%	65,38%	60,52%	60,57%	47,28%	47,28%
1B	6,49%	6,70%	3,77%	3,83%	1,59%	1,65%
2A	66,07%	65,94%	61,74%	61,72%	48,82%	48,80%
2B	14,22%	14,29%	17,89%	17,89%	13,10%	13,16%

Taula 10.29 Percentatge de solucions no dominades sobre el total de solucions per als exemplars de 200 peces en el cas de 5 nivells (g=5)

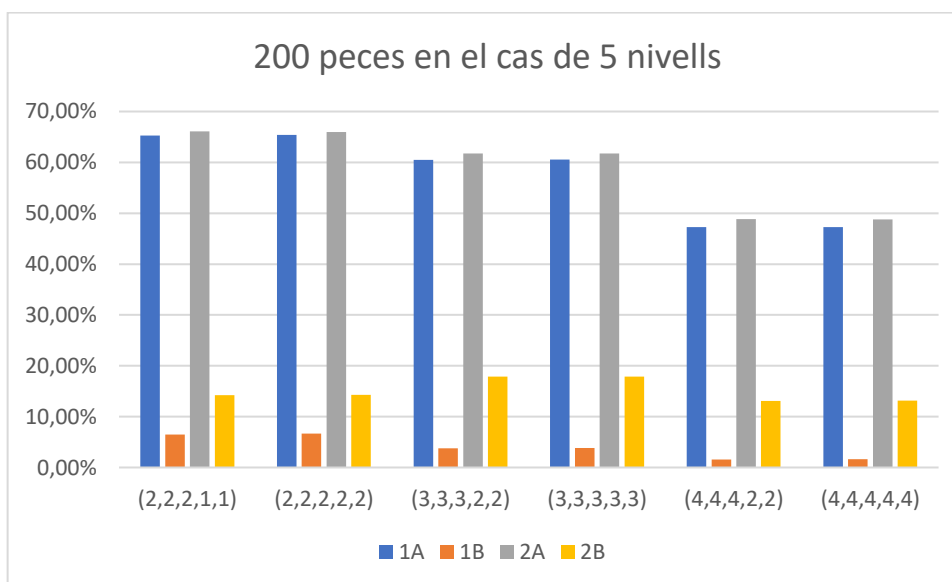


Figura 10.8 Comparació de la no dominància entre heurístiques per als exemplars de 200 peces en el cas de 5 nivells (g=5)

En la comparació dels algorismes per al cas de 200 peces, a la Taula 10.27 i la Figura 10.6 (3 nivells), Taula 10.28 i Figura 10.7 (4 nivells) i Taula 10.29 i Figura 10.8 (5 nivells), s’observen les mateixes tendències que s’observaven en el cas dels exemplars de 100 peces:

- La major qualitat de les heurístiques de tipus A respecte a les de tipus B es dona quan el número màxim de màquines en un dels nivells és menor.
- Els percentatge de solucions no dominades d'un algorisme respecte els altres disminueix a mesura que augmenta el nombre màxim de màquines. És a dir, com més alt és el nombre màxim de màquines, els algorismes comparteixen més solucions entre ells. Per tant presenten menys solucions diferents.
- Els algorismes que incorporen la Fase 2 milloren els que no la incorporen. Això és degut a que amb una mostra gran de peces, 200, el nombre de nodes creix significativament i una major exploració d'aquests permet obtenir més solucions no dominades.
- Degut a la baixa qualitat de l'algorisme 1B, la incorporació de la Fase 2, algorisme 2B, millora molt significativament les solucions que troba.
- La qualitat dels algorismes no varia segons el nombre de nivells. Els algorismes de major qualitat ho són en tots els casos.

11. Comparativa dels algorismes

Un cop analitzats els quatre algorismes per als diferents exemplars de peces (20, 50, 100 i 200 peces) es poden extreure certes relacions:

- Els procediments que incorporen la Fase 2 obtenen un major percentatge de solucions no dominades respecte les de tipus 1. El fet d'explorar més nodes diferents que les de tipus A permet trobar més noves solucions no dominades.
- Les heurístiques de tipus A obtenen un major nombre de solucions no dominades que les de tipus B. A més els temps d'execució d'aquestes dels procediments de tipus A majoritàriament tenen un temps d'execució menor.
- En els casos de 100 i 200 peces s'observa una tendència descendent en els percentatges de solucions no dominades d'un algorisme respecte les altres a mesura que augmenta el nombre màxim de màquines en un nivell. És a dir, com més alt és el nombre màxim de màquines d'un mateix nivell o grup, els algorismes comparteixen més solucions entre elles. Per tant presenten menys solucions diferents.
- Per als exemplars analitzats l'algorisme amb un temps d'execució més petit ha estat l'algorisme 1A ja que té unes condicions més restrictives pel que fa a la limitació de l'exploració i troba un nombre de solucions no dominades major al de l'altre algorisme que no incorpora la Fase 2, l'algorisme 1B.
- Els temps d'execució dels 4 algorismes presenta una relació amb el nombre de nivells de cada exemplar. Com més gran és el nombre de nivells major és el temps de resolució de l'exemplar.
- La baixa qualitat de l'algorisme 1B fa que l'aplicació de la Fase 2 en els nodes marcats en molts dels casos dupliqui el nombre de solucions no dominades respecte els altres algorismes.
- En el cas d'una màquina per nivell, els quatre algorismes presenten un nombre similar de solucions no dominades. En canvi, en el cas de més d'una màquina en algun dels nivells, el nombre de solucions no dominades que presenta cada algorisme varia segons la qualitat del mateix.
- A nivell general, es pot observar que com més nodes s'exploren, millor és la qualitat de l'algorisme (solucions no dominades respecte el total).

A nivell global es pot afirmar que per a tots els grups d'exemplars, l'algorisme amb un major percentatge de solucions no dominades és la 2A. L'explicació és que incorpora una Fase 1, amb un procediment (*procediment A*) que li permet obtenir solucions de qualitat durant l'exploració arborescent, i la Fase 2, que permet ampliar l'exploració en els nodes marcats i així obtenir noves solucions no dominades de major qualitat.

Es pot observar una correlació entre la qualitat dels algorismes i el nombre de solucions no dominades que aquestes troben durant la seva exploració arborescent. A major qualitat d'un algorisme més gran és el nombre de solucions no dominades que aquesta troba.

Majoritàriament els quatre algorismes presenten un comportament similar entre ells independentment del nombre de peces de l'exemplar, el nombre de màquines de cada nivell o el nombre de nivells de cada exemplar.

Comentar que respecte el temps d'exploració de cada algorisme, s'observa una correlació inversament proporcional entre el temps d'execució de cada algorisme i el nombre de solucions no dominades que aquest obté. És a dir, a major qualitat de l'algorisme menys temps d'execució d'aquest.

Finalment, cal remarcar que els resultats comentats s'han obtingut a partir de les mostres creades prèviament tal i com s'indica a l'apartat 10.1.2. No és descartable que en el cas d'utilitzar un altre tipus de mostres, com podria ser amb temps de pre-procés i post-procés diferents per exemple, poguessin sorgir altres conclusions que en aquest cas no s'han pogut veure.

12. Pressupost

A continuació es presenta el pressupost associat a la definició, el disseny i la programació dels procediments plantejats i a l'anàlisi dels resultats obtinguts.

A part del cost associat a les hores de mà d'obra, també s'ha comptabilitzat el cost del software utilitzat. En aquest cas la llicència del programa Microsoft Visual Studio 2010. Val a dir que el temps de redacció de l'informe s'ha considerat en el pressupost en la partida de definició del problema plantejat.

Si els procediments plantejats es volguessin implementar a una empresa s'haurien de preveure els beneficis que aquesta pogués obtenir a l'implementar el projecte i d'aquesta manera valorar el període de recuperació de la inversió.

A la Taula 12.1 Es desglossen els diferents conceptes del pressupost.

Concepte	Quantitat [h]	Cost unitari [€/h]	Total [€]
Estudi situació de partida	20	30	600,00 €
Recerca d'informació	20	30	600,00 €
Definició del problema plantejat	50	30	1.500,00 €
Definició i abast del problema	90	90	8.100,00 €
Procediment Fase 1	60	30	1.800,00 €
Procediment Fase 2	20	30	600,00 €
Disseny dels procediments realitzats	80	60	2.400,00 €
Definició de l'estructura del programa	10	30	300,00 €
Implementació dels procediments de la Fase 1 en llenguatge C#	80	30	2.400,00 €
Implementació dels procediments de la Fase 2 en llenguatge C#	20	30	600,00 €
Proves i millores de programació	30	30	900,00 €
Programació de les heurístiques	140	120	1.500,00 €
Anàlisi comparació dels diferents resultats	20	30	600,00 €
Anàlisi dels resultats	20	30	600,00 €
Microsoft Visual Studio 2010			641,00 €
Subtotal			13.241,00 €
IVA		21%	2.780,61 €
TOTAL			16.021,61 €

Taula 12.1 Pressupost del projecte

Tal i com s'ha comentat anteriorment, no es realitzarà cap estudi dels beneficis que una empresa real podria obtenir en el cas d'aplicar els processos plantejats en el treball en el seu procés industrial. Tanmateix és interessant mencionar les àrees on la implementació de les heurístiques plantejades en una empresa real aportaria més beneficis:

- **Rapidesa en la programació:** Permet trobar de forma ràpida el conjunt de programacions. En el cas de realitzar-se utilitzant altres mètodes es podria trigar molt més a arribar a solucions similars o inclús a les mateixes solucions. Aquesta millora és molt interessant sobretot pels casos de reprogramacions inesperades en què el temps de programació és crucial per no tenir les línies parades (és a dir, el que són aquí màquines en paral·lel).
- **Eina de decisió:** Permet a la persona que dirigeix la planificació de la producció escollir entre un conjunt de programes eficients segons les seves preferències, segons cada iteració en diferents moments del temps.

A més, en el cas de no tenir altres eines de programació i realitzar la planificació de les línies de muntatge manualment, l'aplicació de les heurístiques aportaria els següents beneficis extra:

- **Reducció de temps de finalització:** Aquesta eina proporciona programacions amb temps de finalització i penalitzacions, generalment, menors que les obtingudes a partir de l'experiència de la persona encarregada de la programació.
- **Reducció de la càrrega de treball:** Aquesta eina reduiria els temps que l'encarregat destina a la planificació de les línies de producció i, per tant, permetria que destinés aquest temps a altres tasques o a buscar altres vies d'estalvi.
- **Millor gestió:** Una programació clara, eficient i ben definida comporta una millora en la gestió de tota l'empresa: començant per la mà d'obra directa del procés productiu, les compres de matèria prima, la logística, la producció, etc.

13. Impacte ambiental

Un estudi d'impacte ambiental té com objectiu identificar, descriure i valorar els efectes previsibles que la realització d'un projecte produirà sobre els factors ambientals, per tal de poder-los prevenir i, si és possible, corregir. En el cas del present projecte s'ha realitzat una estimació qualitativa dels efectes que es poden produir feta la implementació d'una eina informàtica que permet programar l'assignació i la càrrega de treball en un taller de producció.

Les programacions obtingudes mitjançant l'aplicació dels procediments proposat s'han basat en la cerca del mínim instant global de finalització (F_{MAX}) i la de mínima penalització (c).

D'una banda l'objectiu de minimitzar l'instant global de finalització proporciona solucions on les peces acaben de ser produïdes abans. Aquesta reducció de temps permet obtenir una major aprofitament de les instal·lacions i, per tant, un estalvi energètic per peça produïda a causa de la reducció de temps ociosos.

D'altra banda, la cerca de solucions amb mínima penalització permet una utilització més eficient de les màquines o recursos ja que tenen una tecnologia més moderna, per tant uns menors consums energètics.

La programació de les peces a realitzar a cada màquina permet conèixer el moment exacte en què es començaran a produir les peces i en quin moment s'acabarà el seu tractament. Aquest fet permet una millor planificació i gestió de l'emmagatzematge i transport de matèria prima i producte acabat produint estalvis en el transport intern i el transport extern.

La utilització d'una eina informàtica permet reduir els suports físics necessaris per realitzar càlculs prendre decisions i per tant reduir el consum de material d'oficina.

Finalment, la velocitat de resolució que el programa ofereix permet gran capacitat de reacció en el cas de la necessitat d'una reprogramació. Això fa que els temps d'inactivitat derivats d'aquests casos es redueixin dràsticament i consegüentment el consum elèctric, el cost d'emmagatzematge, el cost de transport, etc. que hi havia en aquests temps d'inactivitat.

14. Conclusions

El present projecte planteja un problema que sorgeix en una empresa real amb la necessitat de programar diferents peces en diferents conjunts de màquines en paral·lel. El problema tracta sobre la programació d'un sistema productiu format per un nombre a determinar de màquines idèntiques situades en paral·lel en diferents grups, també situats en paral·lel entre ells. Cada grup de màquines està definit per un nivell diferent i per ell s'espera que hi passi un conjunt finit de peces. Cada peça d'aquest conjunt està definida pel temps de pre-procés, temps de procés, temps de procés i nivell de la peça (el nombre de nivells de peces i màquines és el mateix). Cada màquina pot tractar peces del seu nivell o de nivell inferior.

Els objectius de la programació han estat minimitzar l'instant de finalització de la última peça present en el sistema (F_{MAX}) i, alhora, minimitzar, també, la penalització associada a l'intercanvi de peces d'un nivell superior a un nivell inferior (c) relacionada amb els costos d'utilitzar màquines menys eficients.

En el projecte s'han proposat dues fases, cadascuna amb la seves heurístiques per tal d'obtenir un conjunt de programacions (solucions) no dominades per al problema plantejat. En totes es parteix d'un node arrel (Fase 0) on les peces estan programades a les màquines de nivell més alt, a partir de les quals es realitza una exploració arborescent mitjançant un procediment *greedy* directe (Fase 1). Posteriorment, per tal de millorar les solucions obtingudes fins el moment, es realitza un procediment de *backtracking* (Fase 2) en dos dels procediments.

Els dos tipus d'heurístiques queden diferenciats pel fet d'incorporar o no l'esmentada Fase 2 en el seu procediment arborescent.

Primer s'han definit, a partir de diferents procediments, les tres fases que componen els algorismes (Fase 0, Fase 1 i Fase 2) i posteriorment s'han definit els quatre algorismes (1A, 1B, 2A i 2B).

S'ha pogut observar que:

- Els algorismes que incorporen el *procediment A* obtenen aproximadament un nombre 5 cops més gran de solucions no dominades de major qualitat que els algorismes que incorporen el *procediment B*.
- Pel que fa a la qualitat de les solucions obtingudes, es veu clarament que el *procediment A* de la Fase 1 aporta millors solucions que el *procediment B*.

- L'algorisme de major qualitat és l'algorisme **2A** presentant un 2% més de solucions no dominades respecte l'heurística 1A, un 77% més de solucions no dominades que l'heurística 2B i un 92% més de solucions no dominades respecte l'heurística 1B.
- Com més diferència de nombre de màquines per nivell, més destaquen les solucions que presenta l'heurística 2A respecte les altres algorismes.
- Els algorismes que incorporen el *procediment A* troben un major nombre de solucions no dominades durant la seva exploració arborescent. Entre un 30 i un 40% que les que incorporen el *procediment B*. Aquest fet s'accentua en els exemplars on el nombre de peces i màquines és major
- En els casos on hi ha el mateix nombre de màquines per nivell o gairebé el mateix, els quatre algorismes obtenen gairebé el mateix nombre de solucions no dominades, independentment de la qualitat d'aquestes respecte els altres algorismes.
- Els algorismes que incorporen el *procediment B* en general triguen un 20% més en obtenir totes les solucions no dominades en comparació amb els algorismes que incorporen el *procediment A*.
- Els algorismes que utilitzen la Fase 2 tenen uns temps d'exploració majors que els que no l'utilitzen a causa de l'exploració d'un major nombre de nodes.

Per tant, l'algorisme que genera unes millors solucions en el cas del problema plantejat és l'**algorisme 2A** generant més solucions no dominades i de més qualitat que la resta d'algorismes.

Possibles vies d'estudi

Un cop finalitzats els estudis del present projecte, hi ha diverses vies en les quals es podria seguir desenvolupant el projecte. Entre totes aquestes vies es destaquen les més importants:

- Durant el projecte s'ha considerat que la penalització (c) per cada cop que una peça canviava de nivell era d'una unitat. Seria interessant fer el mateix estudi introduint variabilitat en aquest paràmetre, ja sigui relacionant-lo amb la màquina o bé considerant-lo un paràmetre de la definició de cada peça.
- En el projecte s'han considerat màquines idèntiques, és a dir, el temps d'execució d'una peça era el mateix per totes les màquines. Generalment, en una situació real, les màquines presenten diferents temps d'execució. Per tant, es proposa fer el mateix estudi

considerant que els temps d'execució depenguin de la màquina on estigui la peça assignada.

- Ampliar l'exploració arborescent a més nodes dels estudiats al projecte per tal d'obtenir un major nombre de solucions i per tant, fàcilment, més solucions no dominades.
- Incorporació de temps de preparació a les màquines per cada tipus de peça.
- Incorporació d'interval de no disponibilitat de les màquines per avaries o manteniment programat.

15. Agraïments

Voldria agrair especialment al professor Manel Mateo la direcció d'aquest projecte, la seva dedicació i la seva implicació durant tot el procés de realització.

D'altra banda voldria donar les gràcies a totes les persones que en diferent mesura m'han donat suport i m'han ajudat a avançar en aquest projecte.

A tots ells, moltes gràcies!

16. Bibliografia

- Camps, J. (2013). *Programació bicriteri de peces amb temps de pre-procés i post-procés en màquines en paral·lel considerant la seva especialització*. Barcelona: Universitat Politècnica de Catalunya.
- Companys, R. (2003). *Secuenciación, programación de proyectos y de taller. Equilibrado y secuenciación de líneas*. Barcelona: ETSEIB (Publicacions d'abast S.L.L.).
- Conway, R., William, M., & Miller, L. (1967). *Theory of schedulling*. New York: Addison-Wesley.
- Departament d'Organització d'Empreses ETS d'Enginyeria Industrial de Barcelona, Universitat Politècnica de Catalunya. (2016). *Direcció d'Operacions, Programació d'Opearcions*. Barcelona: ATENEA.
- Garriga, X. (2009). *Programació de peces a múltiples nivells en un taller de línies de premses*. Barcelona.
- Gharbi, A., & Haouari, M. (2002). Minimizing Makespan on Parallel Machines Subject to Release Dates and Delivery Times. *Journal of Scheduling*, 1-15.
- Lawler, E., Lenstra, J., Rinnooy, K. A., & Shmoys, D. (1993). *Sequencing and scheduling: algorithms and complexity*. Handbooks in Operations Research and Management Science.
- Mateo, M., Teghem, J., & Tuyttens, D. (2017). A bi-objective parallel machine problem with eligibility, release dates and delivery times of the jobs. *International Journal of Production Research*.
- Pastor, R. (2016). *Mètodes Quantitatius d'Organització Industrial II*. Barcelona: ATENEA.
- Romero, C. (1996). *Análisis de las decisiones multicriterio*. Madrid: Isdefe.

17. Bibliografia complementària

Lopez, P., & Roubellat, F. (2010). *Production Scheduling*. ISTE.