

Pràctica de PDA, Constraints i SAT (2020-2021)

Data màxima de lliurament Moodle: 15 de Gener, lliurament únic.

Entrevista de lliurament obligatòria: a concretar amb cada grup. Dies 18 i 19 de Gener.

Aquesta pràctica constarà de dues parts. La primera part, la de Minizinc, tindrà el major pes (65%) i la segona part, de SAT, contarà un 35%. La pràctica es realitzarà per parelles.

Part Minizinc (Festival de música de cambra), fins a 6.5 punts

El problema a resoldre

Els organitzadors d'un festival de música de cambra ens han demanat que els ajudem a muntar els horaris del festival, degut a la gran combinatòria de possibilitats que tenen, i les restriccions amb les que es troben. En aquest festival es volen interpretar un conjunt de peces musicals amb una orquestra de cambra, és a dir, formada per pocs músics. De cada peça musical sabem el següent:

- Quins instruments i en quina quantitat són necessaris per interpretar-la. Per exemple, en una peça hi intervenen dos violins, tres flautes, un contrabaix i un timbal.
- Quina és la durada de la peça musical. Per facilitar la lectura del programa als assistents es considera que totes les durades són múltiples de 5 minuts, és a dir, sabem que no hi ha cap peça que duri, per exemple 4 minuts.

Els organitzadors del festival ja han contactat amb un conjunt de músics professionals per a interpretar les peces musicals. Cada músic domina un conjunt d'instruments. Hem de decidir quins músics tocaran cadascuna de les peces musicals, tenint en compte quins instruments la composen. Per exemple, per tocar una peça amb dos violins i una flauta, fan falta dos músics que sàpiguen tocar el violí, i un que sàpiga tocar la flauta.

El festival es celebrarà en un espai amb múltiples sales, que sabem que seran suficients per programar tantes peces musicals a la vegada com vulguem. El que sí que hem de tenir en compte és que tenim un nombre de músics limitat, així que hi ha certes peces que no es podran tocar alhora.

No tots els músics cobren el mateix, ja que uns tenen una major formació que altres. Segurament els músics que saben tocar més instruments seran els que cobraran més. Els organitzadors ens han fet saber quants diners poden gastar en la contractació de músics, i sabem el sou que cobra cada músic per cada 5 minuts tocats.

A més a més, per temes estrictament artístics, es demana que hi hagi una relació de precedència entre algunes parelles de peces, és a dir, ens poden demanar que la peça B no s'interpreti abans que la peça A hagi acabat. Això permetrà per exemple, que si tenim 4 peces A, B, C, D que pertanyen a una mateixa col·lecció, els organitzadors ens imposin les precedències $A \prec B$, $B \prec C$ i $C \prec D$, i així el públic podrà escoltar totes les peces de la col·lecció.

Finalment, l'organització ens ha demanat que la jornada comenci **a les 10 del matí**, i acabi com més d'hora millor.

Així doncs, la nostra tasca és fer un model en Minizinc que ens munti l'horari satisfent les restriccions indicades. La sortida del programa ha de deixar clar quina és l'hora d'inici i d'acabament de cada peça musical, quins músics la interpretaran, l'hora d'inici i final del festival, i el preu que costarà contractar els músics.

Dades d'entrada

Es proporciona un conjunt de 10 instàncies del problema plantejat de duresa creixent, seguint la instància `festival1.dzn` la més fàcil i la instància `festival10.dzn` la més difícil. També tenim la instància trivial `festival0.dzn` que es mostra a continuació a tall d'exemple:

```
nPeces = 4;
durada = [25,5,20,15];

nInstruments = 3;
requereix = [|1,1,1,
| 2,0,1,
| 0,2,0,
| 0,1,1,
|];

nMusics = 10;
saptocar = [| true,true,false, | false,false,true,
| false,true,false, | false,true,true,
| true,true,false, | false,true,false,
| true,false,true, | false,true,false,
| true,false,true, | true,true,true,|];

pressupost = 80;
salari = [3,3,2,2,3,2,3,2,3,4];

nPrecs = 2;
```

```
pred = [2,2];  
succ = [3,4];
```

Una solució òptima (no té perquè ser única):

Hora d'inici del festival: 10h 0min
Hora d'acabada del festival: 10h 25min
Cost de contractacio: 79 euros

Peca 1:

Inici: 10h 0min
Final: 10h 25min
Music 7 toca instrument 1
Music 3 toca instrument 2
Music 4 toca instrument 3

Peca 2:

Inici: 10h 0min
Final: 10h 5min
Music 1 toca instrument 1
Music 5 toca instrument 1
Music 10 toca instrument 3

Peca 3:

Inici: 10h 5min
Final: 10h 25min
Music 6 toca instrument 2
Music 8 toca instrument 2

Peca 4:

Inici: 10h 5min
Final: 10h 20min
Music 5 toca instrument 2
Music 2 toca instrument 3

Què es demana

Es demana que entregueu un model en Minizinc **en un arxiu anomenat horaris.mzn** que resolgui **correctament** el problema plantejat. Es valorarà que l'output del Minizinc estigui mínimament ben presentat, quelcom semblant a l'exemple mostrat anteriorment és un bàsic, si trobeu maneres més “maques” millor. Per optar a una bona nota, és necessari que **intenteu**

sol·lucionar quantes més instàncies millor, i en el menor temps possible. Per això haureu de definir un bon model i provar diferents configuracions (solvers diferents, restriccions globals, implicades, trencament de simetries, estratègies de cerca, restarts, ...).

Un mínim per aprobar seria que el vostre sistema fos capaç de resoldre 3/4 instàncies en un temps “raonable” (<5 min.) i que la sortida fos bona i el document explicatiu ben treballat.

D’aquesta pràctica s’espera que experimenteu amb diferents configuracions i expliqueu en un informe quines proves heu fet i quins resultats heu obtingut. De cares a la nota, **l’informe és tan o més important que el model**. Concretament heu d’entregar un fitxer PDF fet en L^AT_EX on es responguin les següents qüestions:

- Explicació del **model** triat: variables, dominis, i restriccions. Expliqueu clàrament el **viewpoint** i si n’heu necessitat més d’un en el model, expliqueu-los i indiqueu quin **channeling** heu fet servir. Si heu fet més d’un model es valorarà positivament, entregueu els que tingueu i indiqueu quin és el millor amb comparacions experimentals.

Indiqueu el nombre de variables i de restriccions del model (en funció del nombre de peces, de músics, ...).

- Sobre les vostres **restriccions**, indiqueu quines **globals** heu fet servir, quines **reificacions**, etc. Heu afegit **variables auxiliars**?
- Heu calculat algun **upper bound** o **lower bound** de la funció objectiu? Quins i com?
- Heu trobat **restriccions implicades**, quines? i **simetries**, com les heu trencat? Quin guany de rendiment heu obtingut?
- Per a les estratègies del **cerca** del Minizinc, quines proves heu fet i quins resultats heu obtingut? Quin solver us ha anat millor? Per què? Doneu-ne les taules amb temps de cada instància per les diferents opcions.

Extres

Es **donarà 0,5 punts extra** al grup que resolgui més instàncies (i/o obtingui millors solucions encara que subòptimes) amb menor temps d’execució.

També es pot aconseguir **0,5 punts extra** implementant la següent restricció addicional:

- Degut a certes propietats de l’instrument número 2 no es vol programar peces amb aquests intruments més de 50 minuts seguits, haurien de passar 5 minuts com a mínim abans que tornés a sonar una peça amb instrument 2.

Canvien els resultats? com? i el rendiment?

Part SAT (Crowded Chessboard), fins a 3.5 punts

En aquesta part es demana que amplieu l'objecte ScalAT amb els mètodes per a codificar cardinality constraints que es descriuran a continuació per a poder donar un model i resoldre el problema *crowded chessboard* (descriu a més envall).

Cardinality Constraints (fins a 1 punt)

Implementeu dins de la llibreria ScalAT les següents codificacions de constraints:

- el mètode `addAMOLog(x)` que codifiqui una restricció de tipus *At-Most-One* a una llista de variables Booleanes `x`, és a dir $(x[0] + x[1] + x[2] + \dots + x[size(x) - 1] \leq 1)$, amb l'encoding logarítmic.
- El mètode `addAMK(x,K)` que codifiqui una restricció de tipus *At-Most-K* a una llista de variables Booleanes `x` amb un enter positiu `K`, és a dir $(x[0] + x[1] + x[2] + \dots + x[size(x) - 1] \leq K)$, fent servir *sorting networks*. La implementació de la *sorting network* ja ve donada a ScalAT.
- El mètode `addALK(x,K)` que codifiqui una restricció de tipus *At-Least-K* a una llista de variables Booleanes `x` amb un enter positiu `K`, és a dir $(x[0] + x[1] + x[2] + \dots + x[size(x) - 1] \geq K)$, fent servir *sorting networks*.
- El mètode `addEK(x,K)` que codifiqui una restricció de tipus *Exactly-K* a una llista de variables Booleanes `x` amb un enter positiu `K`, és a dir $(x[0] + x[1] + x[2] + \dots + x[size(x) - 1] = K)$, fent servir *sorting networks*.

Crowded Chessboard (fins a 2.5 punts)

Codifiqueu i resoleu amb ScalAT el problema del *Crowded Chessboard* (o *tauler atapeït*). Aquest problema consisteix en posar totes les reines, torres i àlfils que es puguin en un tauler i posar-hi també tots els cavalls que es puguin, de manera que les peces del mateix tipus no s'amenacin entre si (obviant les peces que estiguin al mig de la trajectòria d'amenança).

Per exemple, a la Figura 1 hi ha una solució per a un tauler de 5×5 , on s'han de posar 5 reines, 5 torres, 8 àlfils i 5 cavalls.

Aquest problema fou presentat com a “*crowded chessboard*” al 360è problema del llibre de jocs i trencaclosques lògico-matemàtics: *H. E. Dudeney. Amusements in Mathematics. 1917*. que podeu trobar a: <https://www.gutenberg.org/ebooks/16713>. Al llibre hi apareix pel cas d'un tauler de 8×8 ple de peces a reordenar (vegeu Figura 2).

The puzzle is to rearrange the fifty-one pieces on the chessboard so that no queen shall attack another queen, no rook attack another rook, no bishop attack another bishop,

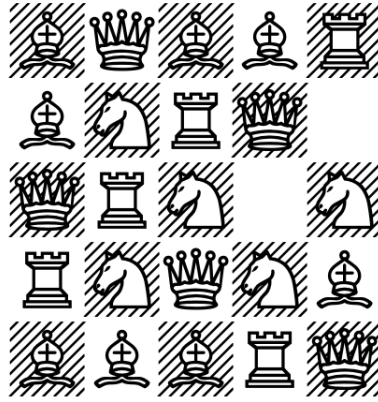


Figura 1: Solució del crowded chessboard per un tauler 5×5 .

and no knight attack another knight. No notice is to be taken of the intervention of pieces of another type from that under consideration—that is, two queens will be considered to attack one another although there may be, say, a rook, a bishop, and a knight between them. And so with the rooks and bishops. It is not difficult to dispose of each type of piece separately; the difficulty comes in when you have to find room for all the arrangements on the board simultaneously.

Aquesta taula mostra les instàncies amb solució que podeu mirar de resoldre amb el vostre encoding amb un temps “raonable”:

mida	reines	torres	àlfils	cavalls
5	5	5	8	5
6	6	6	10	9
7	7	7	12	11
8	8	8	14	21
9	9	9	16	29
10	10	10	18	37
11	11	11	20	47
12	12	12	22	57
13	13	13	24	69
14	14	14	26	81
15	15	15	28	94
16	16	16	30	109

Afegir un cavall més a cada instància les fa **insatisfactibles**.

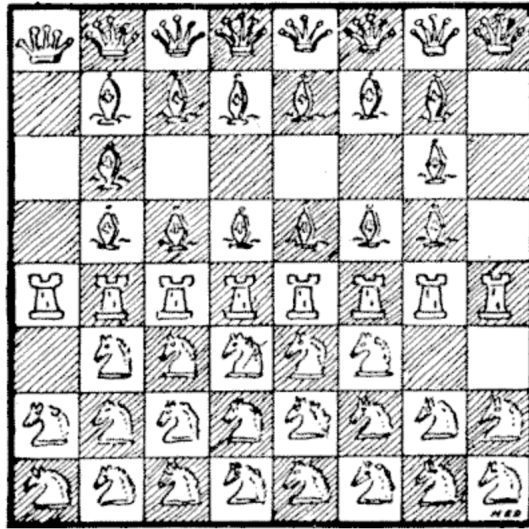


Figura 2: Peces a reordenar per obtenir una solució del crowded chessboard per un tauler de 8×8 .

Què es demana

Caldrà que lliureu un enllaç al vostre Github privat, i m'hi dongueu accés, on hi haurà d'haver l'ScalAT ampliat i un objecte **Crowded** (podeu fer servir el **Queens** com a llavor) que codifiqui i resolgui el *crowded chessboard* usant ScalAT.

Caldrà també lliurar (el podeu posar al Github) un informe PDF (fet amb $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$):

1. Doneu i expliqueu el vostre model: viewpoint, restriccions bàsiques, restriccions implicades que hagueu trobat, trencament de simetries, etc.
2. Òbviament fareu servir cardinality constraints. Cal que proveu les dues configuracions següents del vostre model:

Conf1 feu servir només codificacions quadràtiques pels AMOs i EOs que us apareguin al model.

Conf2 Feu servir només codificacions logarítmiques pels AMOs i EOs que us apareguin al model.

Considerant la instància del tauler de 5×5 ,

- Quines diferències observeu entre les dues configuracions pel que fa a la mida?

- Quina de les dues configuracions és més ràpida? Per què creieu que passa?
3. Mireu de resoldre tantes instàncies com pugueu, tant satisfactibles com insatisfactibles (per a obtenir instàncies insatisfactibles, agafeu les instàncies donades i incrementeu en un el nombre de cavalls a posar), podeu fer servir un *timeout* d'una hora). Doneu una taula de temps per instància. A la taula s'hi ha de poder veure què us ha suposat de millora de temps les millores que hagueu proposat al model (és a dir, feu columnes amb els temps per la codificació bàsica, per la codificació amb restriccions implicades, etc)

Extres

A més a més, es **donarà 0,5 punts extra** al grup que sigui capaç de resoldre més instàncies amb menor temps d'execució.

També es pot aconseguir **0,5 punts extra** implementant la següent restricció adicional al problema:

- Trobar quin és el nombre màxim de reis que no s'amenacin que es poden afegir. Per exemple, en el cas de 5×5 només queden dues caselles lliures però només s'hi podria posar un rei. Existeix alguna altra solució que acceptés posar-hi dos reis? i a la resta d'instàncies, fins quans reis s'hi poden posar?