

Universitat de Girona  
**Escola Politècnica Superior**

Grau en Enginyeria Informàtica

PROJECTE FINAL DE GRAU

---

## **Rush Hour**

---

*Autor:*  
Carla Davesa Sureda

*Tutor:*  
Dr. Mateu Villaret Ausellé

MEMÒRIA

Convocatòria:  
Setembre 2023

Departament :  
Informàtica, matemàtica aplicada i estadística

El joc *Rush Hour* és un trencaclosques de desplaçament de blocs creat per l'enginyer i dissenyador japonès Nob Yoshigahara l'any 1978. A la Figura 1 en podem veure una imatge de com es comercialitza. La seva idea original era proporcionar un enigma de lògica que fos accessible per a tothom i alhora fos un repte intrigant.

Inspirat en el trànsit urbà congestionat, el joc va ser concebut com una representació en miniatura d'una situació de trànsit on caldria moure vehicles per tal d'obrir pas a un cotxe vermell que ha de sortir del tauler.

A primera vista pot semblar una tasca senzilla però el *Rush Hour* és un problema difícil ja que no només requereix de moviments físics, sinó que també cal una capacitat d'anticipar moviments i calcular estratègies.

Inicialment, el *Rush Hour* va ser introduït al mercat japonès com a joc de tauler. No obstant això, va guanyar popularitat de manera ràpida i es va estendre a nivell mundial a través de diferents edicions i adaptacions.



FIGURA 1: Imatge d'una comercialització del joc.

El seu èxit va anar més enllà dels límits del món dels jocs i cridà l'atenció d'entusiastes de la lògica i experts en resolució de problemes. Com veurem, hi ha diversos treballs que es centren en estudiar-ne la complexitat computacional i a cercar mètodes eficients de resolució, ja sigui mitjançant la planificació d'IA o bé la resolució de problemes combinatoris.

La visió del joc com a problema de planificació és molt natural ja que clarament podem veure una noció d'estat del joc (la disposició dels cotxes), un estat inicial (la disposició inicial del nivell del joc que volem resoldre), un estat final (aquell on el cotxe vermell pot sortir) i unes accions que ens permeten evolucionar dins l'espai d'estats (els moviments dels cotxes). La resolució de restriccions es pot fer servir per a resoldre problemes de planificació, i, de fet, serà on ens centrarem especialment.

El nostre treball pretén explorar com la utilització d'eines informàtiques pot oferir noves perspectives als desafiaments que aquest joc planteja. Amb això, buscarem generar nous avenços dins d'àmbits relacionats amb la resolució de problemes combinatoris i la planificació en IA.

Des de sempre, he experimentat una gran atracció pels problemes complexos. Em motiva afrontar-me a qüestions difícils per intentar trobar mètodes eficients que les resolguin de la millor manera possible.

Aquest interès fa que els jocs d'estratègia siguin una part essencial de la meua vida. Concretament, el joc Rush Hour ocupa un lloc molt especial des de la meua infància. Recordo les hores que passava movent els vehicles en el tauler i intentant trobar la manera de fer sortir el cotxe vermell.

Afrontar aquest repte amb una visió més madura i una formació acadèmica més sòlida, m'ha permès canviar la perspectiva del que pensava que era un simple joc entretingut, per veure'l com el que realment és, un autèntic repte computacional.

Els objectius específics que volíem assolir en aquest projecte eren:

1. Aprofundir en l'estudi del joc Rush Hour, i comprendre els detalls que el fan tan fascinant.
2. Explorar i aprendre diferents tècniques útils per resoldre el joc Rush Hour. Es preveu implementar models de programació en diferents llenguatges, com ara MiniZinc, Essence Prime, SAT, SMT i PDDL.
3. Augmentar la mida del tauler original del joc (6x6) fins a tamanys més grans com ara 7x7, 8x8, 9x9, etc. amb l'esperança de trobar configuracions més complexes que posin a prova els models creats, i així comparar la seva eficàcia en un nivell superior de dificultat.
4. Construir un recull de configuracions complexes per analitzar els diferents models i comparar l'eficàcia d'aquests.
5. Analitzar el concepte de dificultat en el joc Rush Hour i identificar els factors que influeixen en aquesta.
6. Explorar l'expressió del problema com una fórmula QBF i avaluar la seva utilitat per a la resolució de problemes més grans i complexos.

Com veurem, no hem assolit tots els objectius (possiblement massa ambiciosos de bon principi per un TFG) però sí el principal, poder analitzar el comportament de diferents aproximacions a la resolució de problemes combinatoris, per a la resolució del Rush Hour, així com explorar la noció de dificultat dels escenaris.

Pel que fa al desenvolupament del projecte s'han seguit les següents fases:

- Investigació preliminar i definició d'objectius
- Funcionament del clúster i execucions
- Generació de configuracions
- Desenvolupament del model MiniZinc
- Desenvolupament dels traductors en Scala
- Desenvolupament del model Essence Prime
- Desenvolupament del model SAT amb Optilog
- Desenvolupament del model PDDL

- Anàlisi de complexitat i dificultat
- Avaluació i comparació de models
- Documentació de la memòria

Per concloure, podem dir que el joc de Rush Hour és un problema complex i presenta reptes de dificultat quan es tracta de buscar solucions òptimes, en concret, i sobretot, a les aproximacions basades en programació de restriccions. Personalment ens ha ofert l'oportunitat d'estudiar nombroses tècniques i marcs de treball per a la resolució de problemes complexos i en particular de planificació: PDDL-planning, MiniZinc-Programació de restriccions, i Essence Prime i Optilog - SAT.

Les conclusions més tècniques les obtenim en el marc de la dificultat de les instàncies i de l'eficiència dels solucionadors:

- L'estudi i classificació de la dificultat de les instàncies és un aspecte complex i que requereix encara de més experiments a poder ser amb instàncies que suposin un repte més significatiu a aproximacions de resolució com la del PDDL o la de SAT. No obstant això hem pogut relacionar un augment de temps de resolució amb un increment en el número de passos. Aquest fenomen s'observa sobretot quan fem l'anàlisi en instàncies que tenen un gran nombre d'estats accessibles i per tant, on l'espai de cerca serà més gran.
- Pel que fa a les aproximacions de solucionadors considerades, clarament el conjunt d'instàncies disponibles no són cap mena de repte per a PDDL. L'aproximació amb Optilog i SAT podríem dir que és la segona més robusta, no obstant, quan estem fent optimització de desplaçaments, comencen a ser instàncies més difícils per al SAT solver. L'aproximació amb SAT solving d'Essence Prime també presenta dificultats en resoldre algunes instàncies fent fins i tot fa algun timeout. Finalment l'aproximació que mostra un comportament menys robust i un tant erràtic és la de resolució de restriccions amb MiniZinc i el LazyFD solver Chuffed.

Per dificultat de problema, on més han patit les aproximacions Essence Prime i MiniZinc ha sigut en la optimització de passos i desplaçaments on han fet nombrosos timeouts mentre que PDDL resolvia en segons (la versió SAT no s'ha implementat per aquest tipus d'optimització).

Pel que fa a les conclusions personals, com he dit, l'aprenentatge de diferents tècniques per a resoldre problemes combinatoris ha sigut un enriquiment molt important. Així com la gestió de clúster per llençar experiments i recuperar resultats, la confecció de gràfiques per a mostrar els resultats, o l'escriptura de programes per a transformació de formats. Per no parlar dels problemes tècnics en instal·lació de softwares diversos.

Espero, i de fet estic convençuda, que aquests aprenentatges em seran molt útils i han sentat la base per a poder dur a terme uns estudis de doctorat en el grup d'IA de la Universitat de St. Andrews.

Per últim, he tingut un primer tast en el desengany d'algunes hipòtesis inicials de treball però també la satisfacció de fer troballes de vegades inesperades. Entenc això és quelcom habitual en el món de la recerca.