

Sistema de gestió d'ordres de transport en logística interna basada en AGV

Joan Jimenez Jané

Resum— Els sistemes centralitzats de gestió d'operacions de transport intern en plantes de manufactura i d'emmagatzematge que hi ha hagut fins ara, no són prou flexibles per adaptar-se a les noves formes de producció i, als canvis que es produeixen a la indústria actual a causa dels sistemes de la fabricació sota demanda i al trànsit que es genera derivat de la complexitat de les plantes actuals. És per això que es necessiten noves tècniques de programació perquè els sistemes de gestió es puguin adaptar i actuar sobre el dinamisme que es produeix.

En aquest projecte es proposa un mètode de gestió basat en subhastes d'ordres de transport utilitzant una metodologia multi-agent, l'arquitectura serà distribuïda i, s'utilitzarà la programació basada en multi-agents per implementar la solució.

Es parteix d'un model base que tracta sobre taxis i passatgers, que inclou una subhasta per decidir quin taxi ha de fer el transport. El model s'ha modificat per incloure-hi múltiples destinacions, posar límits al mapa, arreglar errors de comportament dels agents i s'han afegit nous mecanismes de subhasta dinàmics.

El simulador resultant es pot aprofitar per dissenyar i dimensionar correctament els sistemes de transport intern de les plantes i indústries.

Paraules clau— AGV, Netlogo, Agents, Multiagents, Multi-agents, logística, subhasta

Abstract— Centralized management systems for internal transportation operations in manufacturing and warehousing facilities that have been up to now are not flexible enough to accommodate new forms of production and changes due to on-demand manufacturing. That is why new programming techniques are needed so that management systems can adapt and act on the dynamism that occurs.

This project proposes a management method based on auctions of transport orders using a multi-agent methodology, the architecture will be distributed and multi-agent based programming will be used to implement the solution.

A base model is chosen as the basis, that deals with taxis and passengers, which includes an auction to decide which taxi should make the transport.

The model has been modified to include boundaries to the map, multiple destinations, to resolve agent behavior errors and to add new dynamic auction mechanisms.

The remaining simulator can be used to properly design and dimension the internal transport systems of plants and industries.

Index Terms— AGV, Netlogo, Agents, Multiagents, Multi-agents, logistics, auction



1 INTRODUCCIÓ

Els sistemes tradicionals per a la repartició d'ordres de transport que utilitzen una arquitectura centralitzada no s'adapten correctament al dinamisme de la indústria actual, és a dir, a la incertesa que hi ha quant a l'arribada de noves ordres de transport [2,4].

Per a això, es necessiten nous mètodes de programació que permetin actuar davant els canvis que s'efectuen a l'entorn i que no són fàcilment planificables.

En aquest sentit surt la programació basada en agents permet desenvolupar entorns de fabricació intel·ligents, autònoms i utilitzant una arquitectura distribuïda.

Els **agents** són entitats capaces de percebre el seu entorn a partir de diferents sensors, processar aquestes percepcions i actuar envers les situacions que es va trobant [8].

El projecte es focalitza en la part de la cadena de subministrament intern d'una empresa, que hauria d'estar formada

per robots AGV (vehicle de guiat automàtic) i les diferents ordres de transport que van d'un origen cap a una destinació.

Els **AGV** són vehicles que es mouen de forma independent sense conductor i són utilitzats per fer feines repetitives com podria ser el transport de materials.

Per dur a terme el desenvolupament s'ha fet ús del simulador Netlogo el qual ens permet treballar amb la programació basada en agents i, a la vegada, ens permet realitzar simulacions.

Netlogo és un entorn multi-agent programable que permet simular sistemes per tal de veure el comportament d'aquests[15].

S'ha partit d'un model base que tracta del transport de passatgers d'un origen aleatori cap a l'aeroport [10].

Aquest model ja implementa una solució bàsica de subhasta entre els agents per decidir quin AGV fa el transport. El que es farà és adaptar el model perquè s'assembli més a la realitat d'un magatzem d'una organització i aplicarem sobre el model noves solucions per fer-lo més eficient.

1.2 Definició del problema

El problema que es vol resoldre és l'elaboració d'un sistema de gestió d'ordres de transport en logística interna basada en AGV.

Posem-nos en el cas d'un magatzem d'una empresa, on hi ha dos o més AGV que treballen de forma independent, disponibles per efectuar un seguit de transports, referint-nos per a transport, moure un paquet d'una estanteria a una altra. Segons el transport que s'hagi de fer, el més lògic (per temes de rendibilitat i eficiència) és que l'efectuï el AGV que es troba més a prop i no un que està lluny. Així doncs, el problema que sorgeix és el de donar les ordres de transport al AGV que impliqui un menor cost per realitzar l'operació.

D'altra banda, els AGV funcionen de forma autònoma i, per tant, són reactius amb el seu entorn. Així doncs, ens podem trobar que els AGV arribin a un estat en el qual es vegin compromesos i hagin de deixar de banda el transport, doncs també haurem de lidiar amb problemes d'aquest tipus i implementar solucions envers aquestes situacions.

Per resoldre el problema exposat, s'ha partit d'un model base que es diu *taxisCooperative* i, consisteix, en la subhasta de transports entre els taxis, on la destinació sempre serà a l'aeroport.

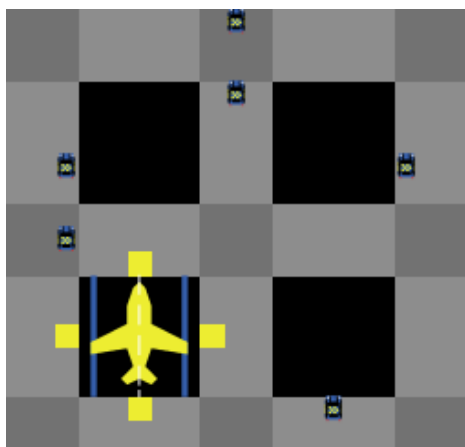


Figura 1 - Model base *taxisCooperative*

1.3 Objectius

L'objectiu principal del projecte és elaborar un model de subhasta d'operacions de transport utilitzant la metodologia de programació multi-agents, que funcioni de forma distribuïda i que sigui més eficient realitzant les operacions de transport vers al model base del què partim.

Per altra banda es vol desenvolupar un nou concepte de subhasta, que funcioni com a suport de la subhasta principal i, que ajudi a triar el taxi més proper per realitzar l'operació de transport.

El motiu de realitzar aquest nou concepte de subhasta, és el de reduir el temps de servei d'una operació de transport.

Els AGV poden endarrerir-se a causa de les col·lisions, durant aquest temps de resolució pot donar-se el cas que un AGV acabi estant més a prop, llavors el més ideal seria que l'operació de transport el fes aquest AGV i no el que l'estava fent.

També pot donar-se el cas, que quan un AGV acaba una operació de transport, aquest estigui més a prop de l'origen, per tant també seria convenient fer el canvi.

Els objectius que deriven del principal són:

Millorar el comportament dels taxis del model base perquè resolguin millor les col·lisions amb l'entorn, detectant els límits del mapa i actuant de forma correcta per no quedar-se en un bucle infinit.

Resoldre els problemes que comporta utilitzar sistemes distribuïts com, per exemple els de comunicació entre els agents, ja sigui perquè no s'ha respost a temps, o perquè el missatge no arriba al destinatari.

Implementar mètodes més precisos per al càlcul del cost de les puges en les subhastes.

1.4 Metodologia

Per a la realització del projecte s'ha seguit un desenvolupament en cascada per anar complint cada bloc en el temps especificat.

Aquesta metodologia bàsicament es basa en què, quan has acabat un bloc, pots començar a realitzar el bloc següent.

El motiu de què no puguis fer el bloc següent abans d'acabar l'actual és perquè hi ha dependència entre les tasques s'han de realitzar en un bloc i el següent.

El projecte ha constatat de 5 blocs:

Aprenentatge: En aquest bloc el que s'ha fet és conciliar les bases sobre la programació d'agents, fer el primer contacte amb el simulador Netlogo, estudiar el comportament BDI dels agents i aprendre com funciona FIPA/ACL.

Anàlisis del model: En el següent bloc es porta a terme l'estudi del model base en el qual partim. És important entendre'l bé per tal de realitzar les modificacions de forma correcta.

Desenvolupament: És el bloc en el qual s'apliquen i es desenvolupen les idees sobre el model.

Extracció de resultats: S'elaboren tot un seguit d'execucions per tal d'extreure dades per després poder analitzar-les i treure diferents conclusions.

Documentació: Es tracta de l'elaboració de l'article i el dossier.

2 ESTAT DE L'ART

2.1 Mètodes de programació d'operacions

A la indústria hi ha dos tipus de gestió d'operacions, aquests es distingeixen per la forma de treballar [17].

PULL: Els articles es preparen segons la demanda. S'utilitza molt quan es competeix per innovació i flexibilitat, la seva implementació requereix informació ràpida des dels punts de venda, i a demès, la cadena de producció ha de ser flexible pels canvis de demanda que es puguin realitzar.

PUSH: Els articles es preparen segons les previsions que es fan.

La producció es basa en pronòstics i s'acostuma a generar una gran quantitat de producte per satisfer un llarg període de temps. El problema d'aquesta gestió radica en què si hi ha un canvi molt gran en la demana, tot el producte generat no servirà per res.

Dels dos sistemes de gestió d'operacions que s'acaben d'explicar, el que més interessa i que està més relacionat amb el projecte és el sistema PULL, ja que, és un entorn on el flux de material pot anar canviant amb el temps, i per tant, la gestió d'operacions de transport serà variable segons la demanda que es faci.

Els sistemes de producció automatitzada es poden classificar en 3 categories[18]:

Automatització fixa: És un sistema en el qual les seqüències de les operacions està determinada per la configuració de l'equip. Aquest tipus d'automatització permet una producció alta, però no té flexibilitat per realitzar canvis en els productes.

Automatització programable: El sistema és capaç de canviar la seqüència de les operacions per adaptar-se a diferents configuracions del producte.

Automatització flexible: El concepte és el mateix que l'automatització programable, la diferència entre un i altre, és que l'automatització flexible no perd temps entre la configuració d'un producte i un altre, per tant no es perd temps de producció.

En l'automatització programable i flexible, la seqüència de les operacions pot canviar en la cadena de producció, i per tant, això afecta el subministrament de material fent que algunes operacions de transport s'hagin de fer abans o més tard.

La cadena de producció està formada per una part de logística, aquesta es pot classificar de la manera següent:

Amb infraestructura fixa: És la logística que es mou per una estructura física i, per tant, té els moviments limitats al que l'estructura li permet.

Sense infraestructura: És la logística que es pot moure de forma independent per tota una àrea de forma autònoma.

Finalment, dins de la logística ens trobem amb vehicles que són més intel·ligents que d'altres, segons la intel·ligència tenim els que tenen les **rutes calculades** i els que realitzen **el càlcul de la ruta de forma independent**.

Així doncs, el projecte se situa en els vehicles(taxis) que es mouen sense infraestructura, en una zona lliure sense limitacions quant a moviment, de forma que qualsevol vehicle podrà realitzar l'operació de transport, és a dir, que ha de ser intel·ligent.

2.2 Explicació del model base taxisCooperative

El model es compon de 2 fases, la fase de configuració i la fase d'execució[10].

El que es fa en la fase de configuració és muntar tot el mapa, és a dir, els carrers i cruïlles per on circularan els taxis, muntar l'aeroport, establir les zones no circulables, fer el càlcul de diverses variables i situar els taxis.

La creació del mapa és molt senzilla perquè és simètric. Es basa en un bucle aniuat on es va recorrent totes les cel·les del mapa, i mitjança 4 càlculs senzills posa un tipus d'element o un altre.

Durant la fase d'execució, els passatgers van apareixent a l'escenari de forma aleatòria i en temps aleatori, és a dir, no apareixen tots els passatgers de cop sinó que van sortint de forma progressiva, amb una probabilitat del 30% a cada pas de simulació.

Mentre els passatgers van apareixent, els taxis van donant voltes pel mapa de forma autònoma aleatòria.

A partir del moment en què el passatger apareix a l'escenari, comença la subhasta per decidir quin taxi durà a terme el transport. Per tant, cada passatger equival a una ordre de transport.

S'aplica una **subhasta a sobre tancat**, que consisteix en què tots els postors presenten la seva oferta una vegada al subhastador de forma secreta, per tant la resta de postors no saben el preu que els altres han presentat. Un cop el subhastador té totes les ofertes, les mira i tria la que té el preu més alt. Per tant el guanyador sempre és el que té el preu més alt.

En el cas del model taxisCooperative no es tria el cost més alt, sinó que es tria el cost més baix perquè, com més alt és

el cost, pitjor li va al taxi per realitzar l'operació de transport.

Quan apareix un passatger, aquest comunica a tots els taxis que hi ha disponibles la necessitat d'un taxi per anar a l'aeroport i, juntament amb el missatge, adjunta les coordenades d'on es troba.

Quan els taxis reben el missatge, aquests calculen la distància en línia recta del punt on es troben al punt on es troba el passatger i ho comuniquen el passatger.

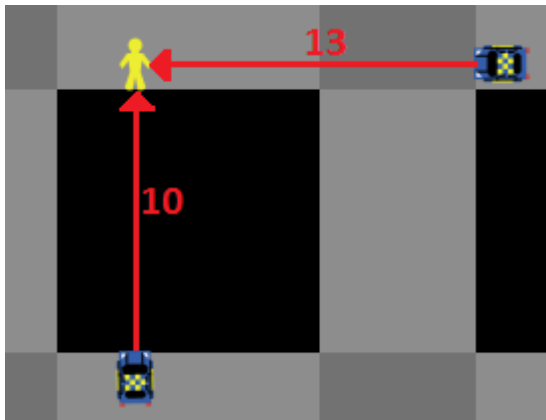


Figura 2 - Càlcul del cost

El passatger recollirà lectures durant 15 cicles de simulació, i un cop acabat aquest període, mirarà totes les lectures i triarà la lectura que tingui el valor més baix. En aquest punt la subhasta ha finalitzat.

A continuació, el passatger comunica al taxi guanyador que el passi a buscar.

El taxi per calcular la ruta cap al passatger, fa ús d'una distància euclidiana per trobar quina és la cruïlla que està més a prop del passatger.

Un cop ha calculat la cruïlla més propera, aquest posa rumb cap a les coordenades de la cruïlla, i un cop el taxi es troba a la cruïlla posa rumb cap a les coordenades del passatger.

Per entendre-ho millor, a la figura 3 hi ha un exemple de com fa el càlcul de la ruta.

Quan ja ha recollit el passatger, el taxi l'ha de portar a l'aeroport.

Per saber el recorregut més curt per anar a l'aeroport, cada taxi mira el valor **distance-airport** dels carrers i cruïlles que l'envolta i se'n va cap al que té el valor més baix, i així fins que arriba a l'aeroport i deixa el passatger.

Un cop arriben a l'aeroport, els taxis deixen els passatgers i tornen a quedar lliures per fer un nou viatge.

A la figura 4 es pot veure en forma d'esquema com està estructurat el temps de servei d'un taxi, i els elements que el componen: el període de subhasta, el temps de recollida i el temps de lliurament.

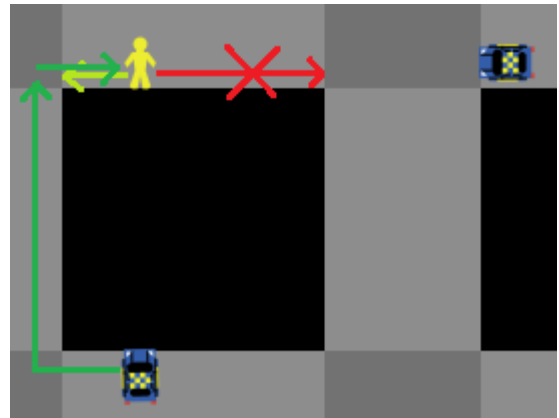


Figura 3 - Càlcul de la ruta del model base

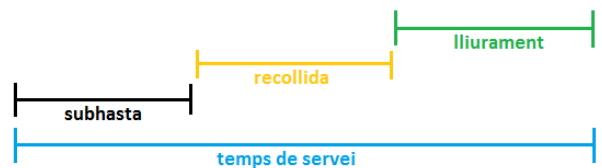


Figura 4 - Esquema del temps de servei d'un taxi

2.3 Problemes específics del model taxisCooperative

Tal com està el model base s'acosta poc al problema que hem plantejat. Els tres principals problemes són:

- Tots els transports finalitzen al mateix lloc, a l'aeroport.
- El model no té límits, és com una bola del món i quan un taxi surti dels límits apareix per l'altre cantó del mapa.
- Un altre problema del model està en el càlcul que es realitza per saber la distància que hi ha entre el taxi i el passatger, aquest càlcul és simple i no s'acosta a la realitat, ja que, no té en compte les zones circulables i les zones no circulables, i no assegura que el taxi que realitza el transport sigui el més a proper. Això passa perquè fa una distància euclidiana, és a dir, fa el càlcul de la distància en línia recta.

A la figura 2 es pot veure com aquest càlcul no és el més correcte.

Per acabar, durant la fase de configuració del model, que és quan es munten els diferents carrers i cruïlles, a aquests se'ls hi assigna una variable anomenada **airport-distance** en el qual es guarda la distància que hi ha del carrer/cruïlla al qual es troba el taxi fins a l'aeroport.

El taxi, a l'hora d'anar a l'aeroport, va mirant els trossos de carrer que té el voltant per veure quin té el valor més petit de la variable **airport-distance**.

3 CANVIS REALITZATS AL MODEL

Ara que es té la problemàtica definida, anem a veure les tasques que s'han realitzat durant el cicle de vida del projecte, on s'han dut a terme diferents tasques d'adaptació del model i el desenvolupament de noves funcionalitats.

3.1 Treure l'aeroport del mapa

S'ha tret l'aeroport del mapa per tal que el model s'aproximi més a la realitat.

Com a conseqüència, s'han hagut de fer diverses modificacions en el model perquè funcioni correctament:

3.1.1 S'ha modificat els carrers i cruïlles perquè ja no tinguin la variable **airport-distance**.

3.1.2 S'ha modificat el comportament dels passatgers. Ara quan apareixen en el model han de generar les coordenades de la destinació on volen anar. Aquestes coordenades es generen de forma aleatòria.

3.1.3 Quan un taxi agafa un passatger, ha de tornar a calcular la ruta per arribar al destinació.

3.2 Canviar el mètode del càlcul de la mínima ruta

Fins ara, el càlcul de la mínima ruta que feien servir els taxis per arribar al passatger, es realitzava mitjançant la distància euclidiana.

Aquest mètode de càlcul s'ha canviat per l'algoritme de Dijkstra, en el qual aquest sabem que sempre ens donarà el camí de mínim cost per arribar al passatger.

3.3 Canvi en el mètode de crear el mapa

Com a conseqüència de l'aplicació de l'algoritme de Dijkstra s'ha hagut de modificar la forma en què es genera el mapa.

S'ha modificat la manera en què es genera el mapa perquè es necessita crear un graf topològic en el qual poder aplicar l'algoritme de Dijkstra

Es segueix fent el bucle cel·la per cel·la tal com s'ha explicat en el punt 2.2, però s'han substituït tots els càlculs que es realitzaven per la interpretació d'una matriu que representa el mapa de l'àrea de treball.

Aquesta matriu conté tots els elements que han d'aparèixer al model, és a dir: carreteres, cruïlles, cases, zones no circulables i els nodes de l'arbre topològic.

És important dir que es creen els nodes però no els arcs que relacionen els nodes, per crear els arcs entre els nodes s'ha desenvolupat l'algoritme següent.

Per traçar els arcs entre els nodes, el que es fa és fer 3 passes pels nodes, i en cada passada s'intenta fer un tipus d'arc diferent:

Primera passada: L'algoritme mira node per node si pot establir un arc horitzontal cap a un altre node, tenint en compte que no traspassa una zona no circulable.

Segona passada: L'algoritme mira node per node si pot establir un arc vertical cap a un altre node, tenint en compte que no traspassa una zona no circulable.

Tercera passada: L'algoritme intenta establir arcs en diagonal cap a altres nodes. La diagonal que intenta traçar seria la d'un quadrat, si està fora d'aquest rang no s'enllaça.

3.4 Canviar la durada de la subhasta

Tal com s'explica en el punt 2.2, la subhasta dura 15 cicles de programa.

Això comporta un problema, ja que, des que el taxi ha calculat la millor ruta per anar a buscar el passatger fins que acaba la subhasta han passat 15 cicles, i per tant el taxi es pot trobar lluny del node inicial.

Per resoldre aquest problema es decideix posar que la duració de la subhasta sigui d'un sol cicle de programa.

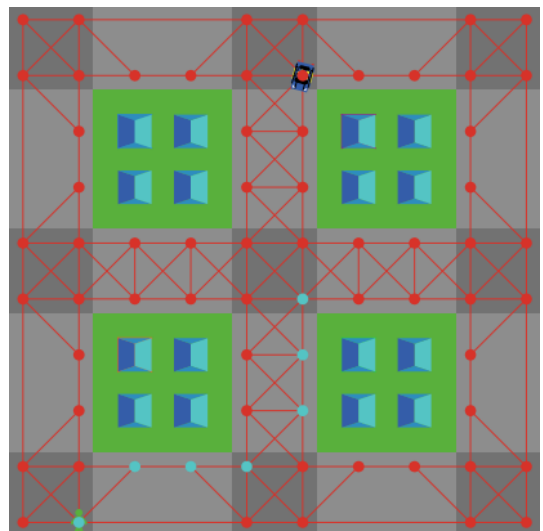


Figura 5 - Arbre topològic i defase per la duració

3.5 Aplicació del concepte de la segona subhasta

El sistema de la segona subhasta consisteix en què el passatger un cop a fet la subhasta per determinar quin taxi té més a prop, aquest segueix fent subhastes mentre el taxi guanyador va cap a buscar-lo, d'aquesta forma pot determinar si hi ha un taxi més a prop i fer el canvi.

Si es dóna el cas de què un taxi està més a prop, fa el canvi i a partir d'ara el taxi que realitzarà el transport serà el nou taxi.

Però el tema no acaba aquí, perquè es segueixen fent segones subhastes fins que en una de les subhastes, el cost del taxi guanyador és inferior al 20% de la distància de la ruta que ha calculat el primer taxi guanyador.

Aquest percentatge s'ha determinat després de fer una bateria d'execucions i analitzar les dades resultants. Aquestes dades es poden veure de forma més detallada a l'apartat de resultats.

El motiu pel qual es para de fer la segona subhasta quan s'arriba a un determinat moment, és perquè s'ha considerat que els taxis ja estan suficientment a prop i perquè a partir del 20% disminueix el nombre de canvis de taxi.

Anem a veure un exemple de segona subhasta per entendre millor el concepte.

Suposarem que tenim 2 taxis ($T1$, $T2$) i 2 passatgers ($P1$, $P2$).

L'escenari inicial en què partirem és el que es pot veure a la part esquerra de la figura 7, com es pot observar $T1$ va a buscar $P1$ i $T2$ està anant cap al destí de $P2$.

Quan $T2$ arribi al destí i deixi $P2$, $T2$ estarà més a prop de $P1$ que $T1$, per tant la segona subhasta que realitzarà $P1$ acabarà fent un canvi de taxi i finalment el que vagi a buscar $P1$ serà $T2$. A la part dreta de la figura 6 podem veure el canvi.

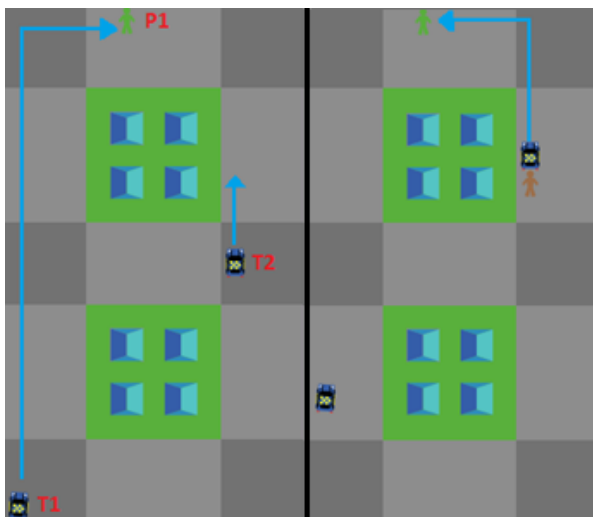


Figura 6 - Exemple de segona subhasta

Finalment, a la figura 7 es pot veure en forma d'esquema com està estructurat el temps de servei dels taxi del model modificat.

Aquest cop l'esquema no es tan senzill, en aquest entra el factor de la segona subhasta, i com es pot veure la recollida i la segona subhasta funcionen de forma concurrent.

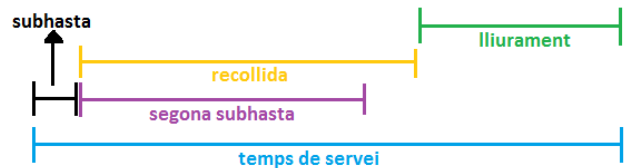


Figura 7 - Esquema del temps de servei d'un taxi després de les modificacions

4. DISCUSSIÓ DE RESULTATS

En aquest apartat es veuran els resultats obtinguts, i una petita discussió quant al seu impacte.

Una equivalència del model desenvolupat a la realitat, podria ser el magatzem que es veu a la imatge de la figura 8, amb illes de materials i passadissos entre mig.



Figura 8 - Exemple d'equivalència del model a la realitat

4.1 Anàlisi de la segona subhasta

En aquest apartat es farà una anàlisi de la segona subhasta per determinar quin és el llindar per deixar d'efectuar-la i, per altra banda, es veurà en quin punt de la ruta és on es produeixen més i menys canvis de taxi.

Per agafar les dades necessàries, s'ha fet tot un seguit d'execucions on, per a cada taxi i per a cada ordre de transport, s'ha capturat a quin percentatge de la ruta es realitzen els canvis de taxi.

Aquestes execucions s'han realitzat amb el valor del llindar a 0, i per tant durant tota la ruta hi podia haver un canvi de taxi.

Les execucions s'han realitzat per diferents paràmetres de configuració, que van des d'1 taxi a 60 taxis, i d'1 passatger a 60 passatgers, incrementant aquests valors de 5 en 5.

Cal aclarir però, que l'aparició dels passatgers es fa de forma progressiva, seguint una distribució aleatòria amb una probabilitat d'aparició del 30% en cada cicle de simulació. Aquest fet provoca que l'aparició dels passatgers no segueixi un patró definit, similar al que passa a una indústria que fabrica sota demanda.

Per tant els resultats que es mostren poden variar per casos on la probabilitat d'aparició sigui més alta o més baixa.

En el gràfic de la figura 10, es quantifica les vegades que hi ha hagut un canvi de taxi en un interval de distància de la ruta establerta, i per tant permet saber en quin punt es produeixen més o menys canvis.

Les dades de les diferents execucions s'han normalitzat per tal d'agrupar-les totes i, per tant a l'hora d'interpretar les dades no es parlarà amb metres o quilòmetres, sinó que es parlarà amb percentatges.

Els intervals [0,5] fa referència al fet que es produeixen canvis quan el taxi es troba entre el 0 i el 5 per cent de distància de la destinació i, quan parlem d'intervals >100 es refereix que s'ha produït un canvi de taxi en una distància més llunyana a l'origen de la ruta.

A l'eix vertical es mostra el nombre de canvis que hi ha hagut, i en l'eix horitzontal es mostra a quin percentatge de la ruta s'han realitzat els canvis.

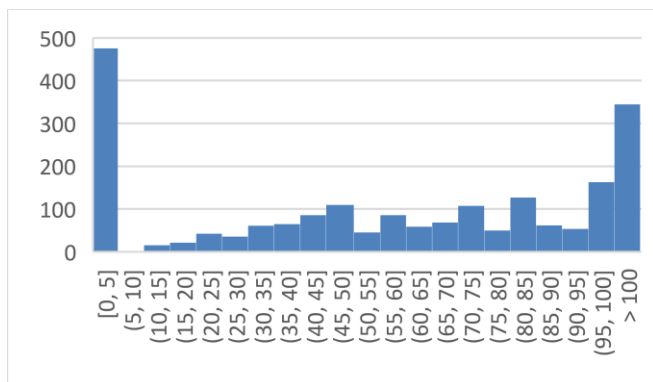


Figura 9 - Gràfic on es mostra el nombre de canvis durant la segona subhasta i el % de ruta on es realitzen els canvis

L'interval que té més canvis de taxi és del 0% al 5% de la ruta amb més de 450 canvis. Aquesta gran quantitat de canvis en intervals tan baixos ve causat per la saturació de taxis en el model.

En l'interval >100% hi ha més de 300 canvis de taxi, aquest fenomen ve degut a les col·lisions que es produeixen entre els taxis quan hi ha molt trànsit.

Els taxis quan detecten que van a col·lisionar, apliquen un seguit de tècniques per evitar xocar i, en aplicar les tècniques poden allunyar-se de l'objectiu, donant peu a què s'allunyi del punt origen de la ruta establerta.

En el cas que el taxi s'allunyi de la destinació, dona la possibilitat que un altre taxi estigui més a prop, i per tant, que es realitzi un canvi.

Primer de tot s'ha de considerar que la segona subhasta s'ha de deixar de fer en un interval raonable, és a dir, que no es deixi de fer quan el taxi està molt a prop del destí, ni quan el taxi està molt lluny del destí. S'ha de buscar un equilibri entre la distància i el volum de canvis que es realitzen a cada interval.

El motiu de què el llinard no pot ser molt proper a la destinació, és perquè pot tindre un efecte contraproductiu, si un taxi ja va cap al destí i quan ja està a punt d'arribar fa el canvi per un altre, pot donar-se el cas de col·lisionar i per tant el temps de servei es vegi afectat.

Tampoc és bo tindre un llinard a una distància molt llunyana, ja que s'estarà limitant molt quan es pot realitzar un canvi de taxi, i per tant, no s'estarà aprofitant bé el concepte de la segona subhasta.

Dit això, es pot veure que a l'interval [5,10] hi ha 0 canvis de taxi, però no és considerat un llinard vàlid perquè es molt proper al destí.

Els intervals que van del [10,15] al [25,30], són intervals que tenen pocs canvis de taxi i, a demès són distàncies raonables tenint en compte l'explicació anterior, sent potencialment candidats per ser utilitzats com a llinard.

En aquest cas el llinard que s'ha posat és el interval [20,25] perquè es considera una distància raonable sobre la destinació i perquè el nombre de canvis que es realitzen en aquest interval són pocs.

Els intervals [10,15] i [15,20] s'han descartat perquè es segueixen considerant pròxim a la destinació.

4.2 Comparació entre el model base i el modificat

En aquest apartat, es comparen el nombre de cicles que tarden de les simulacions realitzades en el model base i el model modificat.

No té molt sentit comparar els temps d'execució dels dos models per la aleatorietat que existeix quan a l'aparició d'ordres de transport, és a dir, no es possible simular els mateixos escenaris en els dos models.

Tampoc es del tot correcte, perquè en un cas real no interessa el tems d'execució, sinó el temps de servei dels taxis.

El que si que te sentit es comparar els missatges intercanviats entre els agents, el recorregut total amb un mètode i l'altre, nombre de col·lisions que es produeixen i temps de servei.

Totes aquestes dades no s'han obtingut, i per tant, no es poden comparar específicament els models en aquests sentits.

No obstant, amb la comparació del nombre de cicles que dura una simulació es pot veure reflectit que els temps de servei del model adaptat es inferior al model base, aquestes dades es poden veure reflectides a la figura 10.

En l'eix Y s'exposa el percentatge de guany del model modificat respecte el model base, en l'eix X el nombre d'ordres de transport que s'utilitza a la simulació i en l'eix Z el nombre de taxis.

Per calcular el percentatge de millora, s'ha realitzat la següent fórmula:

$$100 - \left(\frac{n. \text{cicles modificat} * 100}{n. \text{cicles base}} \right)$$

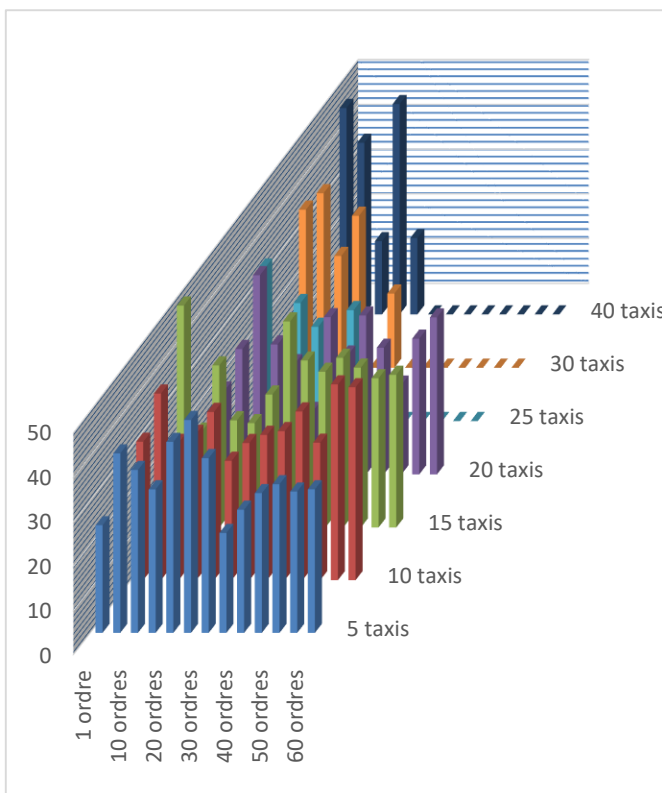


Figura 10 - Millora respecte el model base

Es pot observar que per valors d'ordres de transport compresos entre 30 i 60, i amb una quantitat de taxis alta per la grandària del model, el model adaptat no baixa la quantitat de cicles que tarda a realitzar les ordres de transport.

Això ve degut a que l'execució del model adaptat no sempre acaba bé quan hi ha molts taxis. Quan hi ha molta concurrència de taxis en el model, es produeix un error entre les comunicacions de la segona subhasta, que fa que un taxi es quedi buscant un passatger que no existeix.

En canvi el model base funciona correctament i si finalitza les execucions.

La resta de combinacions es pot observar que els cicles que tarda el model adaptat vers el model base, baixen entre el 20% i el 50%, segons el cas.

4.3 Configuració òptima

A continuació es veurà quin és el nombre de taxis adequat donat un nombre d'ordres de transport en el model modificat.

A la figura 11 es pot observar aquesta comparació, en l'eix X tenim el nombre d'ordres de transport, en l'eix Y els cicles mig que ha durat la simulació i en l'eix Z el nombre de taxis.

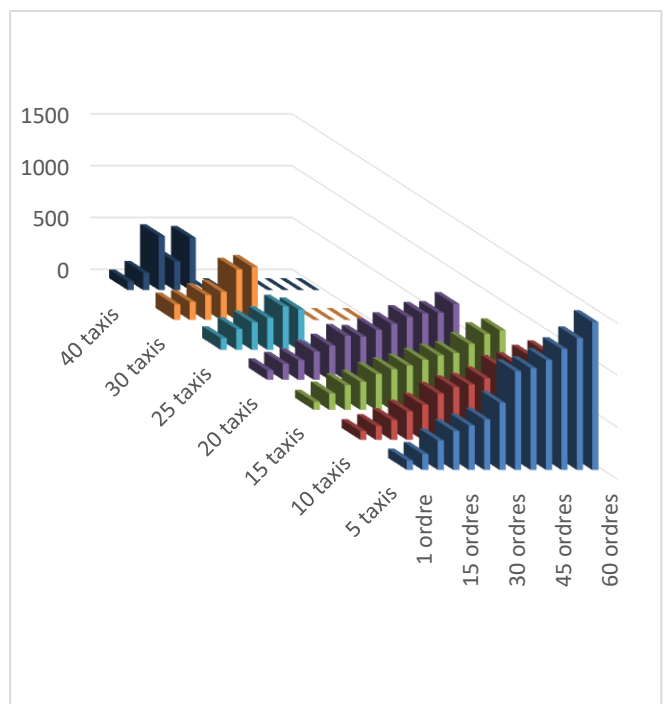


Figura 11 - Gràfic on es veu els cicles mig que han durat les execucions

En el gràfic es pot observar que el nombre de taxis òptim per la grandària del model que s'ha utilitzat, és de 15 taxis, independentment de la quantitat d'ordres de transport que s'hagi configurat la simulació.

Es pot deduir que 15 taxis és l'equilibri entre que hi hagi el model molt saturat i per tant es produeixi un col·lapse, o molt buit i que no s'atenguin les ordres de transport lo suficientment ràpid.

5. CONCLUSIONS

L'objectiu principal del projecte era elaborar un model de subhasta d'ordres de transport utilitzant una arquitectura multi-agent. El resultat ha estat un model el qual es realitzen diferents operacions de transport i aquests s'assignen mitjançant una "subhasta a sobre tancat".

A més a més, s'ha aconseguit reemplaçar la funció de cost per una funció de cost que assegura que el taxi que realitzarà l'operació de transport és el més proper.

Per altra banda, la implementació d'un nou tipus de subhasta ha sigut satisfactori. S'ha aconseguit aplicar el concepte de la "segona subhasta" i fer que el model fos més flexible en la disponibilitat dels taxis.

S'ha de destacar que el més important és que si es compara el model adaptat vers el model base, els cicles que tarda el model adaptat es veu reduït entre el 20 i el 50, complint amb les expectatives que s'havien fet.

Com a punt negatiu de la realització del projecte, ha quedat un error de comunicació en el model quan hi ha molta quantitat de taxis, fent que no acabin l'execució.

AGRAÏMENTS

Agraïments sobretot a la família i amics propers que m'han animat en tot el procés i al tutor Lluís Ribas que m'ha guiat durant tot el projecte.

Bibliografia

- [1] Pablo Noriega, Carles Sierra, "Subastas y sistemas multiagente." IIIA CSIC. [online] (2016)
<http://www.iiia.csic.es/~pablo/PNCSRIIA.pdf>
- [2] Vicene J. Botti Navarro, Adriana Giret Boggino. "Aplicaciones Industriales de los Sistemas Multiagente." Universidad Politécnica de Valencia. [online] (2016)
<http://www.upv.es/sma/teoria/aplicaciones/industria.pdf>
- [3] Pau Fernández Quero. Teoría de subastas: pujar en internet. Universitat Autònoma de Barcelona. [online] (2015)
https://ddd.uab.cat/pub/tfg/2015/136853/TFG_pfernandez-quero.pdf
- [4] José Alberto Araúzo, Ricardo del Olmo, Juan José Laviós. Subasta combinatoria para la programación dinámica en sistemas de fabricación distribuidos. Revistadyo. [online] (2013)
www.revistadyo.com/index.php/dyo/articulo/download/438/458
- [5] Alejandro E., Juliano Moreno, Sebastián F. Modelo de simulación de una subasta de doble punta mediante el paradigma multi-agente. Universidad Nacional de Colombia. [online] (2009)
<http://www.bdigital.unal.edu.co/18622/1/14490-43457-1-PB.pdf>
- [6] Marin Lujak. Agentes Inteligentes y los Sistemas Multiagente. Universidad Rey Juan Carlos. [online] (2016)
http://www.ia.urjc.es/cms/sites/default/files/userfiles/file/MUII-IAD/2014-15/IAD_Sesion2.pdf
- [7] S. Izquierdo, José A. Pascual, Pedro Sanz, Cesáreo Hernández. Creación de programas multi-participante en NetLogo. Aplicaciones en Ingeniería de Organización. Universidad de Valladolid. [online] (2011)
http://adingor.es/congresos/web/uploads/cio/cio2011/medios_cuantitativos/825-834.pdf
- [8] María del Carmen Romero Ternero. Sistemas MultiAgente (MAS). Universidad de Sevilla. [online] (2010)
<http://www.dte.us.es/personal/mcromero/masresdes/docs/SMARTD.0910.mas.pdf>
- [9] Jose Aguilar, Addison Rios Bolivar, Francisco Hidrobo, Mariela Cerrada. Sistemas MultiAgentes y sus Aplicaciones en Automatización Industrial. Universidad de los Andes. [online] (2013)
<http://www.ing.ula.ve/~aguilar/actividad-docente/SMA/documentos/editor.pdf>
- [10] Ilias Sakellariou, Petros Kefalas, Ioanna Stamatopoulou. MAS Coursework Design in NetLogo. Universitat de Macedonia, Universitat de Sheffield. [online] (2009)
http://users.uom.gr/~iliass/projects/NetLogo/Papers/NetLogoCoursework_EDUMAS.pdf
- [11] Alejandro Guerra-Hernández, Amal El Fallah-Seghrouchni, Henry Soldano. Learning in BDI Multi-Agent Systems. Universitat de Paris. [online] (2016)
http://www.wabi.snv.jussieu.fr/people/soldano/clima_CR.pdf
- [12] Juan Pavón-Mestras. Modelos y arquitecturas de Agentes. Universidad computense de Madrid. [online] (2006)
<http://www.fdi.ucm.es/profesor/jpavon/doctorado/arquitecturas.pdf>
- [13] Philippe Caillou, Benoit Gaudou, Arnaud Grignard, Chi Quang Truong, Patrick Taillandier. A Simple-to-use BDI architecture for Agent-based Modeling and Simulation. The Eleventh Conference of the European Social Simulation Association.. [online] (2015)
<https://hal.inria.fr/hal-01216165/document>
- [14] GruSMA: Lenguaje de comunicación entre agentes FIPA. Universitat Rovira i Virgili. [online] (2016)
<http://deim.urv.cat/~itaka/presentacions/fipa-acl.htm>
- [15] The center for Connected learning and computer-based learning. University of Northwestern. [online] (2016)
<https://ccl.northwestern.edu/netlogo>
- [16] Lluís Ribas-Xirgo, Plataforma de simulació de sistemes d'AGV, Comunicació personal, 2016.
- [17] Gestipolis. [online]. (2001)
<http://www.gestipolis.com/que-son-los-sistemas-de-jalar-pull-y-empujar-push/>
- [18] Gestipolis. [online]. (2012)
<http://flobaautomation.blogspot.com.es/2012/02/definicion-de-automatizacion.html>