



Relatório ALGAV

- G026 - Sprint D
-



Instituto Superior de
Engenharia do Porto

o



o

DEPARTAMENTO DE ENGENHARIA
INFORMÁTICA



Identificações.

- Turma: 3DE.
- Membros:
 - André Filipe Lages Novo, 1181628.
 - Diogo Filipe Freitas Ribeiro, 1180782.
 - Hugo Filipe Gonçalves Frias, 1170500.

- João Pedro Pinho da Cruz, 1161255.
- Vera Lúcia Oliveira Pinto, 1180730.



1. Introdução.

- O objetivo é no final da realização deste **Sprint D**, o grupo ser capaz de escalonar de forma automática os **Drivers** a **Workblocks**.
- Ficando assim com os **WorkBlocks** associados a um **Driver** e a um **Vehicle**, através da criação de **DriverDuties** e **VehicleDuties**, respetivamente.



2. Representação do conhecimento do Domínio.

- Relativamente aos motoristas.
 - Estáticos.
 - `horariomotorista/5`.

```
% horariomotorista(motorista, horarioInicial, horarioFinal, duracaoTotalConducao,
listaDuracaoBlocosMotorista)

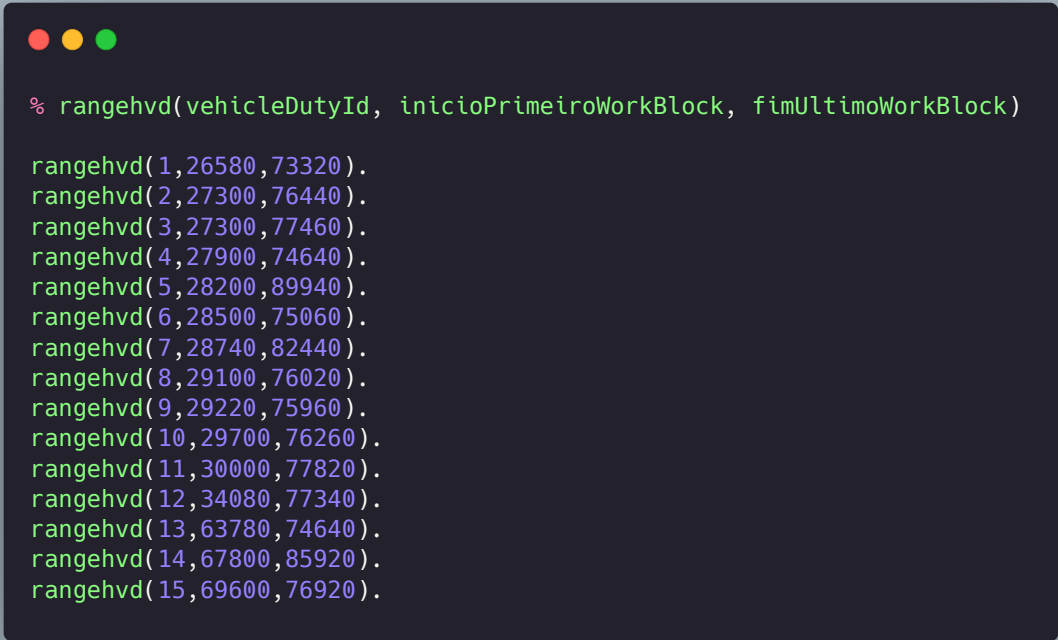
horariomotorista(276,25200,61200,28800, [21600,7200]).
horariomotorista(527, 25200, 61200, 28800, [21600,7200]).
horariomotorista(889, 25200, 61200, 28800, [21600,7200]).
horariomotorista(1055, 25200, 61200, 28800, [14400,14400]).
horariomotorista(1461, 25200, 61200, 28800, [14400,14400]).
horariomotorista(1640, 25200, 61200, 28800, [21600,7200]).
horariomotorista(2049, 25200, 61200, 28800, [21600,7200]).
horariomotorista(5188,33300,69300, 28800, [7200,21600]).
horariomotorista(6616,33300, 69300, 28800, [14400,14400]).
horariomotorista(6697,33300, 69300, 28800, [21600,7200]).
horariomotorista(11018,41400,77400, 28800, [21600,7200]).
horariomotorista(11692,41400, 77400, 28800, [21600,7200]).
horariomotorista(14893,45000,81000, 28800, [10800,18000]).
horariomotorista(16458,50400,86400, 28800, [14400,14400]).
horariomotorista(16690, 50400, 86400, 28800, [7200,21600]).
horariomotorista(16763, 50400, 86400, 28800, [14400,14400]).
horariomotorista(17015, 50400, 86400, 28800, [10800,18000]).
horariomotorista(17552,54000,90000,28800, [10800,18000]).
horariomotorista(17623,25200, 61200, 28800, [21600,7200]).
horariomotorista(17630,25200, 61200, 28800, [21600,7200]).
horariomotorista(17639,27000,48600,21600,[21600]).
horariomotorista(17673,25200, 61200, 28800, [21600,7200]).
horariomotorista(18009,50400, 86400, 28800, [7200,21600]).
horariomotorista(18050,54000,90000,28800, [21600,7200]).
horariomotorista(18097,57600,79200,21600,[21600]).
horariomotorista(18105,57600,79200,21600,[21600]).
horariomotorista(18107,57600,79200,21600,[21600]).
horariomotorista(18119,59400,81000,21600,[21600]).
horariomotorista(18131,66600,88200,21600,[21600]).
```

- Dinâmicos.
 - `t/4`.
 - `t(inicioTrabalhoBlocoMotorista, fimTrabalhoBlocoMotorista, duracaoMaximaBlocoMotorista, motorista)`.
 - `driverduty/2`.
 - `driverduty(driverDutyID,listaDeWorkBlocksAssociados)`.

- Relativamente aos **VehicleDuties**.

- Estáticos.

- rangehvd/3 .



```
% rangehvd(vehicleDutyId, inicioPrimeiroWorkBlock, fimUltimoWorkBlock)

rangehvd(1,26580,73320).
rangehvd(2,27300,76440).
rangehvd(3,27300,77460).
rangehvd(4,27900,74640).
rangehvd(5,28200,89940).
rangehvd(6,28500,75060).
rangehvd(7,28740,82440).
rangehvd(8,29100,76020).
rangehvd(9,29220,75960).
rangehvd(10,29700,76260).
rangehvd(11,30000,77820).
rangehvd(12,34080,77340).
rangehvd(13,63780,74640).
rangehvd(14,67800,85920).
rangehvd(15,69600,76920).
```

- vehicleduty/2 .

- Partilhados por ambos.

- Estáticos.

- workblock/4 .

```
% workbook(WorkBlock, List_of_Trips, StartTime, EndTime)

workblock(12,[459],34080,37620).
workblock(211,[31,63],37620,41220).
workblock(212,[33,65],41220,44820).
workblock(213,[35,67],44820,48420).
workblock(214,[37,69],48420,52020).
workblock(215,[39,71],52020,55620).
workblock(216,[41,73],55620,59220).
workblock(217,[43,75],59220,62820).
workblock(218,[45,77],62820,66420).
workblock(219,[48,82],66420,70020).
workblock(220,[52,86],70020,73620).
workblock(221,[56,432],73620,77220).
workblock(222,[460],77220,77340).
workblock(10,[351,390,455],29700,32520).
workblock(180,[354,390,393],32520,36120).
workblock(181,[357,393,396],36120,39720).
workblock(182,[360,396,399],39720,43320).
workblock(183,[363,399,402],43320,46920).
workblock(184,[366,402,405],46920,50520).
workblock(185,[369,405,408],50520,54120).
workblock(186,[372,408,411],54120,57720).
workblock(187,[375,411,414],57720,61320).
workblock(188,[378,414,417],61320,64920).
workblock(189,[381,417,420],64920,68520).
workblock(190,[384,420,423],68520,72120).
workblock(191,[387,423,426],72120,75720).
workblock(192,[426,456],75720,76260).
```

- Dinâmicos.

- lista_motoristas_nworkblocks/2 .
- **Facto dinâmico auxiliar:** lista_motoristas_nworkblocks_aux/2 .



3. Associação dos Drivers aos VehicleDuties.

Código

```
associarMotoristas:-
    findall((HorarioInicial,VehicleDuty),rangehvd(VehicleDuty,HorarioInicial,_),ListaHorariosVehicleDuty),
    sort(ListaHorariosVehicleDuty,ListaOrdenadaHorariosVehicleDuty),

    findall((InicioTrabalhoBlocoMotorista,FimTrabalhoBlocoMotorista,DuracaoMaximaBlocoMotorista,Motorista),
    t(InicioTrabalhoBlocoMotorista,FimTrabalhoBlocoMotorista,DuracaoMaximaBlocoMotorista,Motorista),ListaTuples),
```

```

sort(ListaTuples,ListaTuplesOrdenada),
(retractall(lista_motoristas_nworkblocks_aux(_,_,_)); true),
(retractall(lista_motoristas_nworkblocks(_,_,_)); true),
obterMaiorWorkBlockVehicleDuty(ListaOrdenadaHorariosVehicleDuty),
gerarListaMotoristasNWorkBlocks(ListaOrdenadaHorariosVehicleDuty),
associarMotoristas1(ListaOrdenadaHorariosVehicleDuty, ListaTuplesOrdenada),
condensadorMotoristas.

% Predicado auxiliar do associarMotoristas.
% Função: Associar Tuples (Drivers) a VehicleDuties, atribuindo-lhes WorkBlocks dos mesmos.
% Observação: Se um Tuple (Driver) não se mostrar capaz de preencher nenhuma lacuna nos WorkBlocks de
todos os VehicleDuties, o mesmo é retirado da lista de Tuples.

associarMotoristas1([],_):-!.
associarMotoristas1(_,[]):-!.
associarMotoristas1(ListaOrdenadaHorariosVehicleDuty, ListaTuplesOrdenada):-
    associarMotoristas2(ListaOrdenadaHorariosVehicleDuty, ListaTuplesOrdenada, ListaTuplesAtribuidos),
    ((ListaTuplesAtribuidos = [], nth1(1,ListaTuplesOrdenada,PrimeiroElemento),
delete(ListaTuplesOrdenada, PrimeiroElemento, NovaListaTuples));
    subtract(ListaTuplesOrdenada, ListaTuplesAtribuidos, NovaListaTuples)),
    verificarAssociacaoMotoristas(ListaOrdenadaHorariosVehicleDuty,ListaResultado),
    associarMotoristas1(ListaResultado,NovaListaTuples).

% Predicado auxiliar do associarMotoristas1.
% Função: Tentar associar um Tuple (Driver) a um VehicleDuties, atribuindo-lhe WorkBlocks do mesmo.

associarMotoristas2([],_,[]):-!.
associarMotoristas2([(_,VehicleDuty)|ListaOrdenadaHorariosVehicleDuty], [(_,_,TempoTrabalho,Motorista)|
ListaTuplesOrdenada], [(_,_,TempoTrabalho,Motorista)| ListaTuplesAtribuidos]):-
    maiorWorkBlockVehicleDuty(VehicleDuty,DuracaoMaior),
    NumeroWorkBlocks is (TempoTrabalho / DuracaoMaior),
    verificar(VehicleDuty,NumeroWorkBlocks),
    retract(lista_motoristas_nworkblocks_aux(VehicleDuty,ListaMotoristas)),
    asserta(lista_motoristas_nworkblocks_aux(VehicleDuty,
[(Motorista,NumeroWorkBlocks)|ListaMotoristas])),
    associarMotoristas2(ListaOrdenadaHorariosVehicleDuty,ListaTuplesOrdenada,ListaTuplesAtribuidos).
associarMotoristas2([(_,_)|ListaOrdenadaHorariosVehicleDuty],ListaTuplesOrdenada,ListaTuplesAtribuidos)
:-
    associarMotoristas2(ListaOrdenadaHorariosVehicleDuty,ListaTuplesOrdenada,ListaTuplesAtribuidos).

% Predicado auxiliar do associarMotoristas.
% Função: Agregar todos os Workblocks de um determinado Driver numa só posição da lista (2º argumento
do facto dinâmico lista_motoristas_nworkblocks/2).

condensadorMotoristas:-
    findall(_,
        (retract(lista_motoristas_nworkblocks_aux(VehicleDuty,ListaWorkBlocks)),
        condensadorMotoristas1(VehicleDuty,ListaWorkBlocks)
        ),_).

% Predicado auxiliar do condensadorMotoristas.
% Função: Fazer a agregação individual de um Driver.

condensadorMotoristas1(_,[]):-!.
condensadorMotoristas1(VehicleDuty,[(Motorista,Numero)|ListaWorkBlocks]):-
    findall((Motorista,Valor),(member((Motorista,Valor),ListaWorkBlocks)),Lista),
    condensadorMotoristas2([(Motorista,Numero)|Lista],Total),
    subtract(ListaWorkBlocks,Lista,NovaLista),
    retract(lista_motoristas_nworkblocks(VehicleDuty,ListaAux)),
    asserta(lista_motoristas_nworkblocks(VehicleDuty,[(Motorista,Total)|ListaAux])),
    condensadorMotoristas1(VehicleDuty,NovaLista).

% Predicado auxiliar do condensadorMotoristas1 que irá calcular o Total de Workblocks de um
determinado Driver, naquele VehicleDuty específico.

condensadorMotoristas2([],0):-!.
condensadorMotoristas2([(_,Valor)|Lista],Total):-
    condensadorMotoristas2(Lista,Total1),
    Total is Total1+Valor.

% Predicado auxiliar do associarMotoristas.
% Função: Obter o maior WorkBlock da lista de WorkBlocks de cada VehicleDuty.

obterMaiorWorkBlockVehicleDuty([]):-!.
obterMaiorWorkBlockVehicleDuty([(_,VehicleDuty)| ListaHorariosVehicleDuty]):-
    vehicleDuty(VehicleDuty,ListaWorkBlocks),
    obterMaiorWorkBlockVehicleDuty1(ListaWorkBlocks,VehicleDuty,DuracaoMaior),
    asserta(maiorWorkBlockVehicleDuty(VehicleDuty,DuracaoMaior)),
    obterMaiorWorkBlockVehicleDuty(ListaHorariosVehicleDuty).

% Predicado auxiliar do obterMaiorWorkBlockVehicleDuty.
% Função: Obter o maior WorkBlock da lista de WorkBlocks de um VehicleDuty.

obterMaiorWorkBlockVehicleDuty1([],_,0):-!.
obterMaiorWorkBlockVehicleDuty1([WorkBlockId|ListaWorkBlocks],VehicleDuty,DuracaoMaior):-
    workblock(WorkBlockId,_,Inicio,Fim),

```

```
Diferenca is Fim - Inicio,
obterMaiorWorkBlockVehicleDuty1(ListaWorkBlocks,VehicleDuty,DuracaoMaior1),
((DuracaoMaior1 < Diferenca,! ,DuracaoMaior is Diferenca);DuracaoMaior is DuracaoMaior1).
```

-

Raciocínio por trás do código

- Seguindo as recomendações do professor regente de **ALGAV**, o grupo criou um facto dinâmico de Tuples ($\tau/4$) para facilitar o escalonamento dos **Drivers**.
 - O facto dinâmico $\tau/4$ vai ter uma entrada para cada **Driver** existente na base de conhecimento.
- Posteriormente, ordenámos os **VehicleDuties** por ordem crescente de tempo de início, bem como os recém-criados Tuples .
- Finalmente, o grupo tenta associar um determinado Tuple a um **VehicleDuty**, dando prioridade claro aos primeiros de ambas as listas, de modo a que ao **VehicleDuty** que mais cedo começa ficará associado o **Driver** que mais cedo entra ao serviço também.
 - Caso isto não se verifique ser possível, avançamos uma posição na lista de **VehicleDuties** e retomamos o processo.

Output Exemplo.

- **Input:** Neste caso em concreto, o algoritmo rodou apenas tendo em consideração os **VehicleDuties** 10 e 12.

```
12, [(1461,4),(527,6),(14893,3)]
```

-

```
10, [(1055,4),(5188,2),(276,8)]
```



4. Chamada adequada do escalonador de Drivers para cada VehicleDuty.

Código

```

% + ----- +

% Predicado auxiliar do escalonamentoMotoristas.
% Função: Percorrer as entradas do facto dinâmico lista_motoristas_nworkblocks/2 e preencher o facto
dinâmico driverduty/2.

gerarDriverDuties:-
    NumeroGeracoes is 5,
    TamanhoPopulacao is 10,
    ProbabilidadeCruzamento is 50,
    ProbabilidadeMutacao is 10,
    (retractall(driverduty(_,_)),!;true),
    findall((VehicleDuty),
        (lista_motoristas_nworkblocks(VehicleDuty,_
        ),ListaVehicleDuties),
    nl, write(ListaVehicleDuties), nl,

gerarDriverDuties1(NumeroGeracoes,TamanhoPopulacao,ProbabilidadeCruzamento,ProbabilidadeMutacao,ListaVe
hicleDuties).

% Predicado auxiliar do gerarDriverDuties.
% Função: Chamar o Algoritmo Genético recursivamente para cada entrada do facto vehicleduty e
posteriormente ir atualizando as entradas do facto dinâmico driverduty/2.

gerarDriverDuties1(_,_,_,[ ]):-!.
gerarDriverDuties1(NumeroGeracoes,TamanhoPopulacao,ProbabilidadeCruzamento,ProbabilidadeMutacao,
[VehicleDuty|ListaVehicleDuties]):-

geraComParametros(VehicleDuty,NumeroGeracoes,TamanhoPopulacao,ProbabilidadeCruzamento,ProbabilidadeMuta
cao),
    melhor((Individuo*_)),
    vehicleduty(VehicleDuty,ListaWorkBlocks),
    gerarDriverDuties2(ListaWorkBlocks,Individuo),

gerarDriverDuties1(NumeroGeracoes,TamanhoPopulacao,ProbabilidadeCruzamento,ProbabilidadeMutacao,ListaVe
hicleDuties).

% Predicado auxiliar do gerarDriverDuties2.
% Função: Preenche o facto dinâmico driverduty/2 levando em consideração os melhores indivíduos gerados
pelo Algoritmo Genético.

gerarDriverDuties2([ ],_):-!.
gerarDriverDuties2([WorkBlock|ListaWorkBlocks], [Motorista|Individuo]):-
    retract(driverduty(Motorista,Lista)),
    asserta(driverduty(Motorista,([WorkBlock|Lista])),
    gerarDriverDuties2(ListaWorkBlocks,Individuo),!.
gerarDriverDuties2([WorkBlock|ListaWorkBlocks], [Motorista|Individuo]):-
    Lista = [ ],
    asserta(driverduty(Motorista,([WorkBlock|Lista])),
    gerarDriverDuties2(ListaWorkBlocks,Individuo).

% + ----- +

```

Raciocínio por trás do código

- O predicado `geraComParametros` é o Algoritmo Genético desenvolvido no **Sprint C**.
- Depois de obtermos o melhor escalonamento para os **WorkBlocks** de um determinado **VehicleDuty**, procedemos à atribuição dos **Drivers** a esses **WorkBlocks**.
 - **Exemplo:** Associamos o **WorkBlock** na posição **X** da lista inicial do **VehicleDuty** ao Driver nessa mesma posição **X** no output do Algoritmo Genético.
- Como um **Driver** pode aparecer em posições diferentes no Output do Algoritmo Genético, bem como pode aparecer em **VehicleDuties** diferentes (Outputs diferentes), o grupo criou uma predicado para concatenar essas atribuições: `gerarDriverDuties2`.

Output Exemplo.

- **Input:** Neste caso em concreto, o algoritmo rodou apenas tendo em consideração os **VehicleDuties** 10 e 12.

```
3 ?- driverduty(X,Y).  
X = 276,  
Y = [192, 191, 188, 185, 182, 181, 180, 10] ;  
X = 5188,  
Y = [190, 183] ;  
X = 1055,  
Y = [189, 187, 186, 184] ;  
X = 1461,  
Y = [222, 216, 215, 214] ;  
X = 527,  
Y = [221, 220, 213, 212, 211, 12] ;  
X = 14893,  
Y = [219, 218, 217].
```

o

- **Observações:**
 - o Considerámos o maior **Workblock** do **VehicleDuty** durante os cálculos, isto é, para saber quantos **WorkBlocks** um **tuple** consegue preencher.
- O grupo começou por ir buscar à base de conhecimento todos os **VehicleDuties** que conhecia.
 - o Considerando o facto dinâmico `lista_motoristas_nworkblocks` já com as entradas preenchidas de forma automática anteriormente, usando os factos `rangehvd`, `horariomotorista` e `t`.
- Posteriormente procedeu à chamada do **Algoritmo Genético** desenvolvido no **Sprint C** para cada um desses **VehicleDuties**.



5. Chamada adequada do algoritmo de cálculo do tempo de mudança de motoristas entre um ponto de rendição/recolha e outro.

- O grupo não realizou este ponto.
 - o Apesar do grupo se sentir confortável o suficiente para afirmar que o algoritmo desenvolvido durante o **Sprint B** se encontra correto, o mesmo não conseguiu arranjar uma forma de conseguir saber em qual **Node** é que um **Driver** fica após a realização de um determinado **WorkBlock**, de modo que não foi possível integrar ambos os algoritmos, infelizmente.



6. Detecção automática de Hard Constraints nos DriverDuties gerados depois do escalonamento.

- O grupo não realizou este ponto.



7. Correção automática de Hard Constraints nos DriverDuties gerados depois do escalonamento.

- O grupo não realizou este ponto.



8. Estado da Arte.

De acordo com o nosso estudo, e de forma a introduzir o leitor, denotamos que o conceito **Machine Learning** (aprendizagem de máquina) define um ramo de inteligência artificial que estuda a forma como os sistemas computacionais aprendem e desenvolvem capacidades, de acordo com a experiência e reconhecimento de padrões.

- Machine learning is the study of computer algorithms that allow computer programs to automatically improve through experience (Mitchell,1997) .
- De forma a compreender melhor a aplicação de **Machine Learning** no conceito real e relacionando com o trabalho que desenvolvemos, pesquisamos 3 áreas diferentes onde este conceito tem usabilidade ao nível dos transportes públicos.



8.1 Primeiro Artigo

Artigo/Paper : [A Machine Learning Approach for Detecting Long-Term Changes in the Weekly Trip Patterns of Public Transport Passengers.](#)



8.1.1 Introdução ao Estudo.

- De acordo com a pesquisa efetuada, o primeiro dos artigos que achamos importante realçar prendesse com o estudo das diferenças de padrões de viagem dos cidadãos a longo prazo. Porquê?! Consideramos importante esta abordagem do tema, na medida em que nos pode dar informações relevantes, como, saber as paragens mais utilizadas pelos passageiros, horários mais afluentes de viagem, linhas mais utilizadas, entre outros. Após a leitura do artigo em questão, percebemos que através do uso de um cartão inteligente o passageiro forneceria dados sobre as suas viagens e variações semanais das mesmas. Relacionando esta informação de um longo prazo de tempo, reunindo dados de inúmeros cidadãos, e visto que os algoritmos de **Machine Learning** através de uma maior quantidade de dados possuem uma melhor eficiência de aprendizagem, achamos claramente eficaz o estudo destes padrões de forma a beneficiar e apoiar uma empresa de gestão de transportes na tomada de decisão.

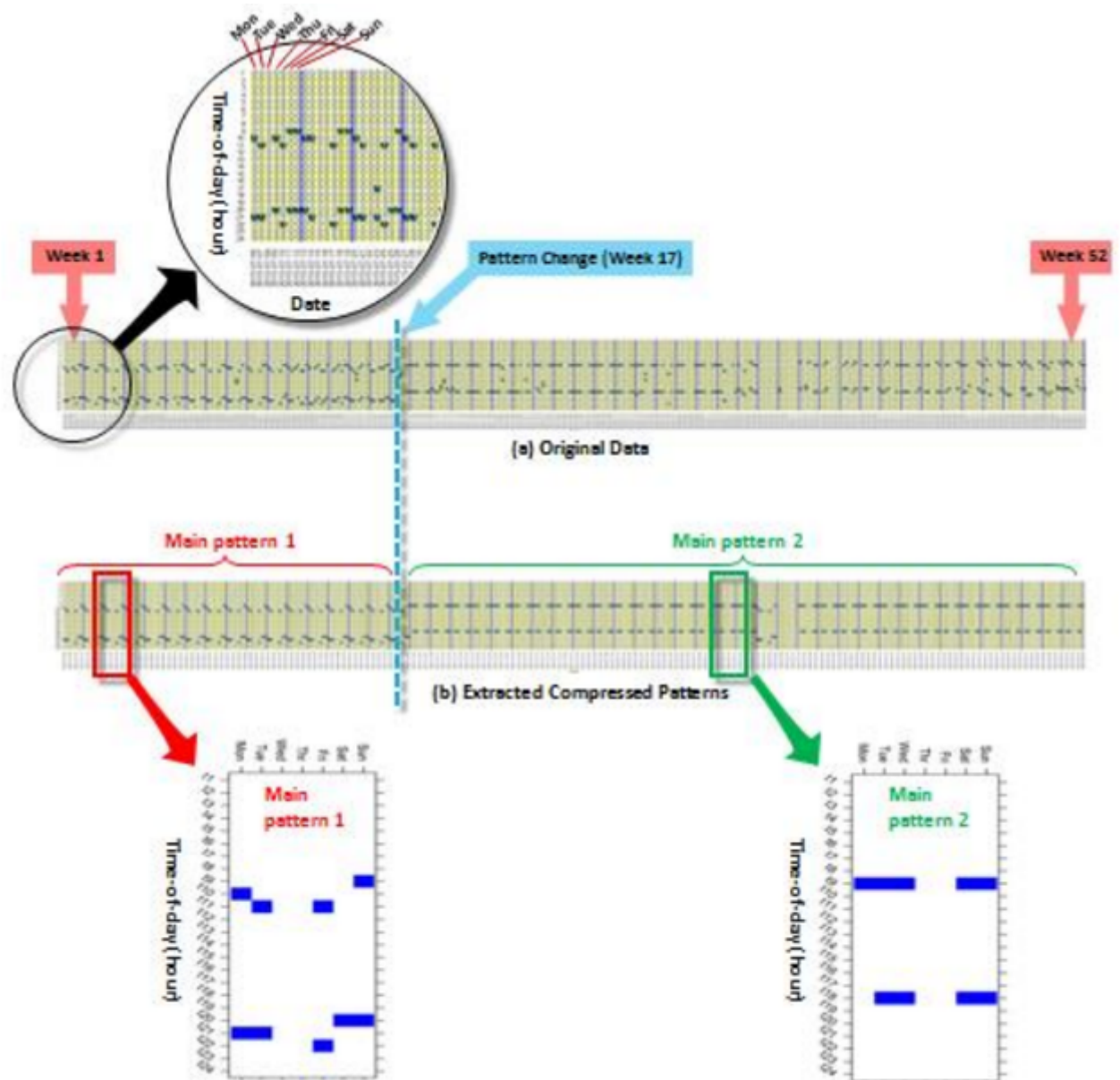
8.1.2 Descrição ao Estudo.

- Então, de acordo com o artigo, é feita numa primeira fase a conversão dos dados do cartão para um formato de dados a ser analisado posteriormente, analisando assim a mudança de comportamento de cada individuo. Após a recolha de dados, é possível tecer conclusões quanto à utilização da rede de transporte em questão, usando **Decision Trees**.

Operator	Operations_Date	Run	Route	Service	Direction	Ticket_Nui	Smartcard_ID	Boarding_Time	Alighting_Time	Passengers	Boarding_Stop
Brisbane Transport	25-Feb-13	3312	P206	7944	Inbound	go card	0090818B44AAAEACEB55COA95B4E8317	25/02/2013 06:41:30	25/02/2013 07:11:02	1	illa Street - 51 (BTOC)
Brisbane Transport	25-Feb-13	3509	P208	7947	Outbound	go card	0090818B44AAAEACEB55COA95B4E8317	25/02/2013 17:09:39	25/02/2013 18:02:09	1	S George St (Stop 8)
Brisbane Transport	26-Feb-13	3312	P206	7944	Inbound	go card	0090818B44AAAEACEB55COA95B4E8317	26/02/2013 06:45:18	26/02/2013 07:09:35	1	illa Street - 51 (BTOC)
Brisbane Transport	26-Feb-13	3509	P208	7947	Outbound	go card	0090818B44AAAEACEB55COA95B4E8317	26/02/2013 17:20:14	26/02/2013 17:58:28	1	S George St (Stop 8)
Brisbane Transport	27-Feb-13	3312	P206	7944	Inbound	go card	0090818B44AAAEACEB55COA95B4E8317	27/02/2013 06:40:02	27/02/2013 07:04:38	1	illa Street - 51 (BTOC)
Brisbane Transport	27-Feb-13	3113	P205	7175	Outbound	go card	0090818B44AAAEACEB55COA95B4E8317	27/02/2013 17:01:05	27/02/2013 17:27:58	1	p Nth Quay (Stop 4)
Brisbane Transport	28-Feb-13	3324	P206	7944	Inbound	go card	0090818B44AAAEACEB55COA95B4E8317	28/02/2013 07:01:52	28/02/2013 07:31:18	1	illa Street - 51 (BTOC)
Brisbane Transport	28-Feb-13	3509	P208	7947	Outbound	go card	0090818B44AAAEACEB55COA95B4E8317	28/02/2013 17:07:02	28/02/2013 17:54:19	1	S George St (Stop 8)
Brisbane Transport	01-Mar-13	3324	P206	7944	Inbound	go card	0090818B44AAAEACEB55COA95B4E8317	01/03/2013 07:03:38	01/03/2013 07:32:00	1	illa Street - 51 (BTOC)

date	Mon																								Tue																							
	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
25-Feb	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
04-Mar	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11-Mar	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18-Mar	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25-Mar	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- Este estudo adota uma abordagem de aprendizado de máquina para detectar mudanças a longo prazo em padrões de viagem repetidos de maneira não supervisionada. Os investigadores extraíram recursos dos padrões de viagem semanal de cada passageiro do PT e aplicaram um algoritmo de agrupamento (**clustering algorithm**) para identificar os principais padrões no uso semanal do PT por um passageiro por um longo período de tempo. Depois de aplicar um algoritmo de agrupamento, aplicaram um algoritmo de classificação (**classification algorithm**) a cada passageiro para determinar se esse passageiro exhibe uma mudança de padrão comportamental a longo prazo.
- Este estudo analisa os principais padrões de passageiros e as mudanças a longo prazo no movimento dos mesmos. O termo "padrão principal" é usado para se referir a um padrão de embarque semanal representativo de um determinado passageiro, que acaba por se repetir ao longo de várias semanas.



An illustrative example of a passenger's long-term behaviour change and the extracted patterns that were identified by the proposed framework.

o

8.1.2 Conclusão do Estudo.

- As vantagens que encontramos num estudo de dados como este prendem-se com variados fatores como a afetação de mais horários e autocarros a paragens mais utilizadas pelos passageiros, determinar motivos pelos quais certos horários são ou não vantajosos, as mudanças de padrões de viagem consoante atrasos em certas paragens, entre outros.

8.2 Segundo Artigo

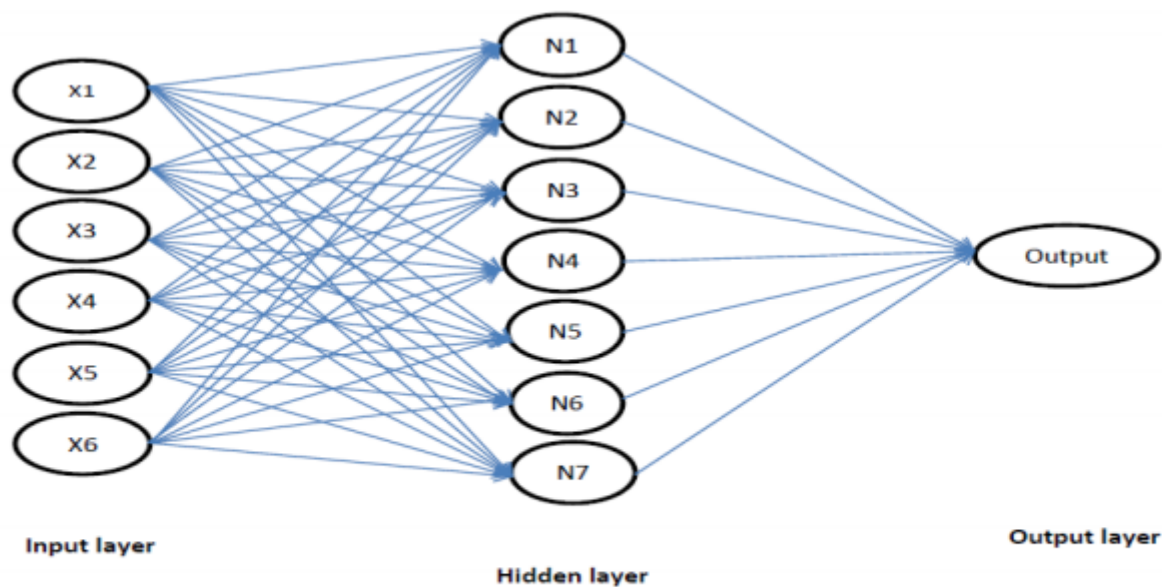
Artigo/Paper : [Comparison of Model Based and Machine Learning Approaches for Bus Arrival Time Prediction.](#)

8.2.1 Introdução ao Estudo.

- Outro dos artigos que analisamos, debruça-se sobre o facto de se conseguir prever o tempo de viagem que um autocarro pode demorar, analisando um conjunto enorme de dados de forma à máquina poder analisar padrões e automatizar-se no cálculo de durações de viagens padronizadas consoante intensidade de trânsito, hora da viagem, etc. Visto que o principal objetivo de qualquer rede de oferta de transporte é a eficácia e eficiência do serviço, é importante dar a conhecer ao cliente a melhor informação acerca desse mesmo serviço prestado. Esta é uma forma de atrair mais população para o transporte público, garantindo-lhes segurança e confiabilidade.

🦉 8.2.2 Descrição do Estudo.

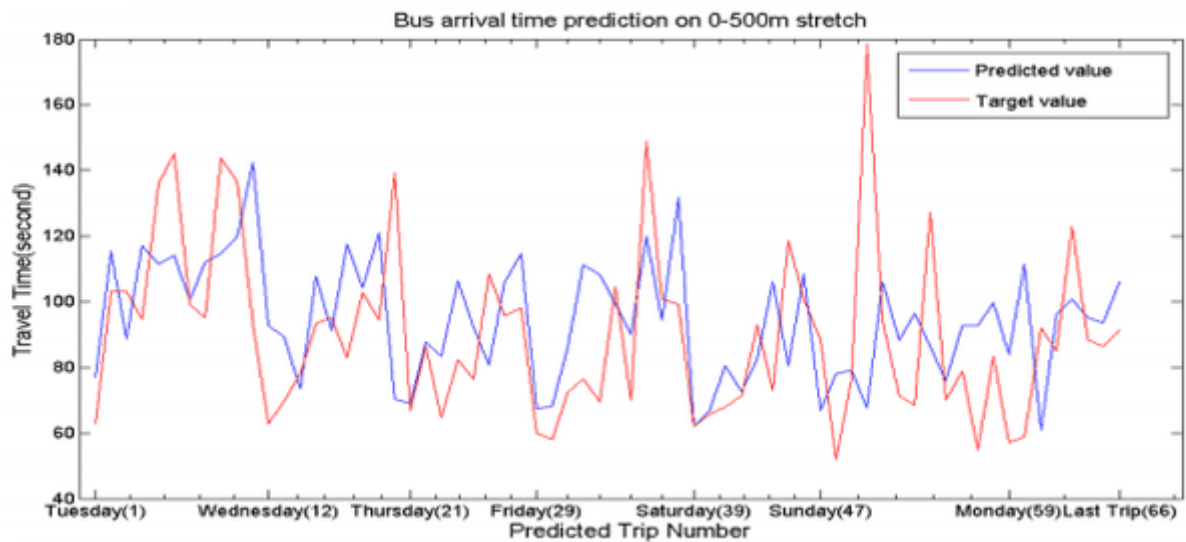
- Como tal, e de acordo com a leitura do artigo, uma das variantes estudadas relaciona-se com **Artificial Neural Networks (ANN)**. De acordo com o artigo, **ANN** está provado como uma das melhores ferramentas para o reconhecimento de padrões mediante um conjunto de problemas. Neste estudo, o autor preocupou-se com a análise de várias viagens numa rede indiana, diária e semanalmente. Desvantagens que o mesmo apresenta para este tipo de estudo, devem-se ao facto de ser estritamente necessária a recolha de imensa informação de forma a realizar uma boa previsão do facto em estudo.



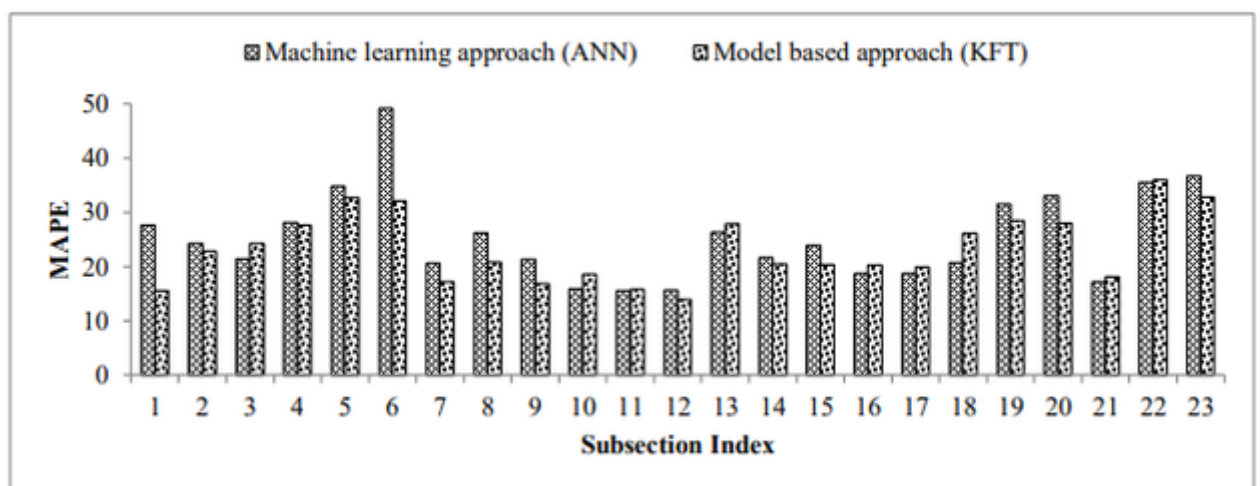
BASIC DIAGRAM OF NEURAL NETWORK OF THE MODEL USED IN THE PRESENT STUDY

o

- Para a recolha de dados, foram usados equipamentos **GPS** dos autocarros da **MTC (Metropolitan Transport Corporation, Chennai, India)**. Foram tidos em conta fatores como a distância entre paragens, tempo, latitude-longitude das paragens. Sumariamente, e de acordo com o que concluímos, a metodologia **ANN** resume-se a um estudo dum conjunto de comportamentos, que vão criar processos e rotinas no cérebro da máquina de forma à mesma adaptar-se e responder à chegada de um novo evento. Embora com o problema associado à necessidade de uma grande variedade de dados, as redes neurais artificiais (**ANNs**) revelaram-se capazes de obter previsões semelhantes ao valor real demorado pelo autocarro no percorrer do mesmo percurso.



MEASURED TRAVEL TIME AND PREDICTED TRAVEL TIME USING ANN.



COMPARISON BETWEEN MACHINE LEARNING AND MODEL BASED APPROACHES.

8.2.3 Conclusão do Estudo.

- Este estudo comparou o desempenho de duas técnicas de previsão comumente usadas, uma orientada a dados (requer uma quantidade elevada de informação) e outra que tem requisitos mínimos de dados. A técnica orientada a dados selecionada foi a **ANN**, enquanto que a outra abordagem foi efetuada usando a Técnica de Filtragem de *Kalman*. As redes neurais artificiais, como "aplicação" de **Machine Learning** que são, tornam-se mais eficientes à medida que mais vezes vão sendo utilizadas, contudo, a comparação dos dois métodos mostrou um melhor desempenho da **ANN** em relação à **KFT** (*Kalman Filtering Technique*).

8.3 Terceiro Artigo

Artigo/Paper : [Route Choice Stickiness of Public Transport Passengers: Measuring Habitual Bus Ridership Behaviour using Smart Card Data.](#)

8.3.1 Introdução ao Estudo.

- Este estudo tem como objetivo medir e analisar o comportamento de escolha de rota dos passageiros de autocarros, com ênfase particular na captura da "aderência" no processo da escolha de rota dos mesmos (**SI** - *Stickiness Index*). A consistência da escolha da rota é definida como a tendência do passageiro de fazer persistentemente as mesmas escolhas de rotas ao viajar entre um determinado par Nó Origem - Nó Destino (**OD**). Um passageiro com alta aderência tenderá a usar sempre a mesma rota, apesar da disponibilidade de várias rotas. Uma vez que a aderência de escolha de rota é definida para um passageiro específico para um par **OD** específico, podemos também definir a aderência (consistência da adesão) ao nível de um **OD** agregando as tendências de aderência para os passageiros, quando este OD se encontra disponível.

8.3.2 Descrição do Estudo.

- Neste último artigo que analisámos, observámos que este estudo assenta em 3 questões principais para o seu desenvolvimento: saber as características de escolha e decisão dos passageiros, quais os fatores que influenciam essas escolhas e qual a dinâmica geográfica dos percursos escolhidos pelos utilizadores.
- De forma a tecer conclusões sobre tudo isto, o estudo assenta na definição de um índice de aderência (**SI**) que mede quantitativamente a escolha/tendência para determinado par Nó Destino-Nó Chegada. Após ser bem definido, este índice foi adaptado ao **Índice de Diversidade de Simpson** (uma métrica comumente usada que mede a biodiversidade na ecologia, aqui adaptada a outro contexto - *Regression Analysis using Simpson's Diversity Index*) que tem como base e princípio a medição da diversidade de escolha.

Ordinary least square regression and quantile regression at the 25th, 50th, 75th and 95th percentile stickiness index values (number of observations = 100,373).

Category	Independent variable	Coefficient β (Standard Error)				
		OLS regression	Quantile regression at 0.25 quantile	Quantile regression at 0.5 quantile	Quantile regression at 0.75 quantile	Quantile regression at 0.95 quantile
User characteristics	<i>#Journeys per day</i>	0.0335*** (0.001)	0.0080*** (0.000)	0.0256*** (0.000)	0.0549*** (0.001)	0.1010*** (0.001)
	<i>OD Usage Fraction</i>	0.0420*** (0.001)	0.0103*** (0.000)	0.0317*** (0.000)	0.0659*** (0.001)	0.1100*** (0.001)
	<i>#Alternatives</i>	-0.0010 (0.002)	-0.0054*** (0.001)	-0.0084*** (0.001)	0.0079*** (0.002)	0.0207*** (0.003)
OD characteristics	<i>#OD Users</i>	0.0014*** (0.001)	0.0007*** (0.000)	0.0011*** (0.000)	0.0019*** (0.001)	-0.0025** (0.001)
	<i>Distance between Origin and CBD</i>	-0.0144*** (0.001)	-0.0032*** (0.000)	-0.0088*** (0.001)	-0.0207*** (0.001)	-0.0237*** (0.002)
	<i>Distance between Destination and CBD</i>	-0.0025*** (0.001)	-0.0008*** (0.000)	-0.0022*** (0.001)	-0.0008 (0.001)	0.0024 (0.002)
	<i>Is Outbound</i>	0.0123*** (0.001)	0.0039*** (0.000)	0.0094*** (0.001)	0.0135*** (0.001)	0.0108*** (0.002)

o

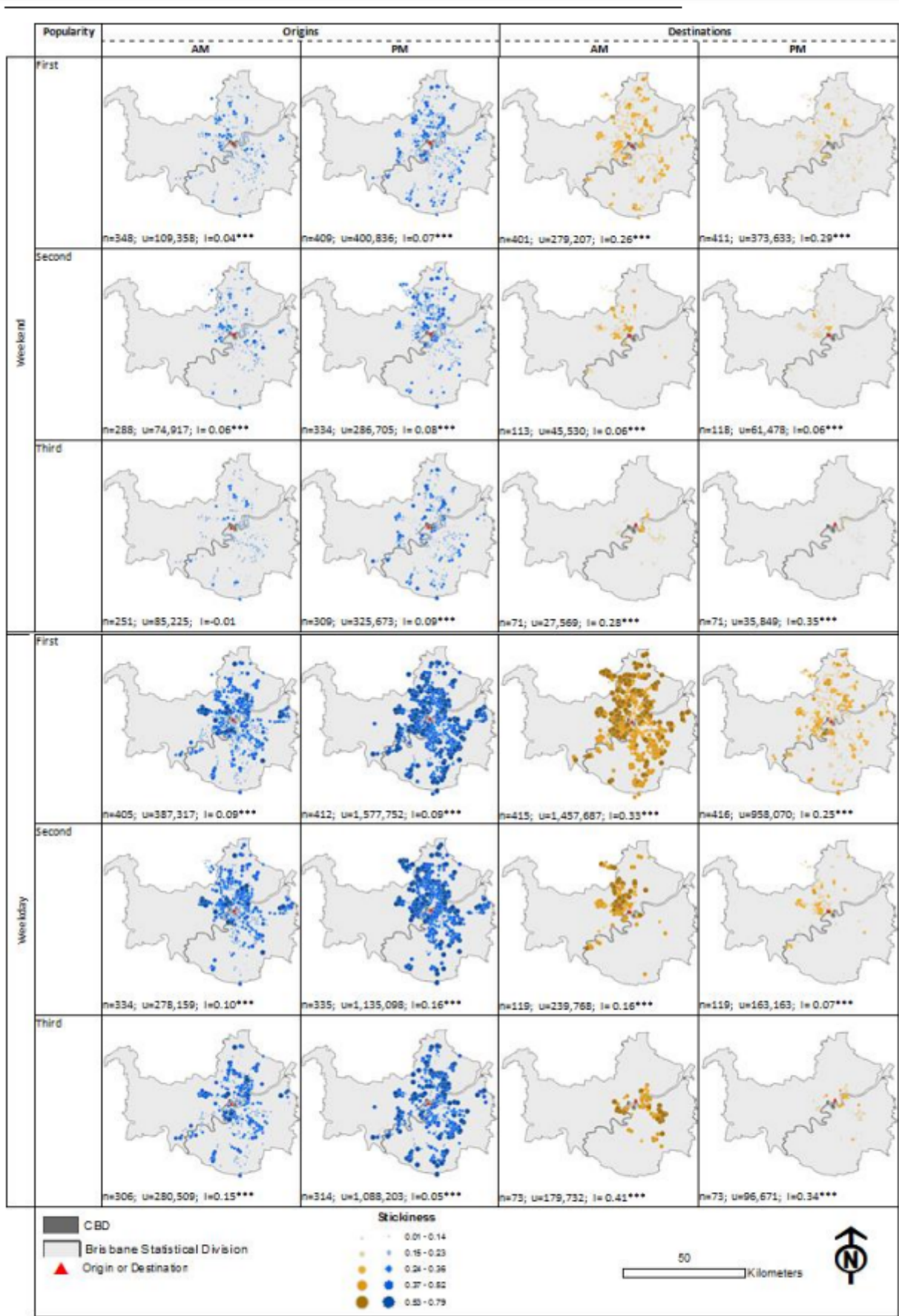
- A área de estudo escolhida para este artigo foi Brisbane, capital estatal de Queensland (Austrália), que era à data a terceira cidade mais populada do país. O tempo de estudo em questão foi de 6 meses, onde a proposta seria analisar os dados de cartões inteligentes utilizados pelos passageiros, que então dariam a informação que o estudo procurava como as trajetórias/percursos escolhidos, pares Nó Origem – Nó Destino e saídas para cada par, construção de matrizes utilizador-caminho para cada par de Nós. Foram então estudados 5460 pares Nó Origem – Nó Destino e 82706 passageiros, tendo em conta o dia da semana e a hora do dia.

8.3.3 Conclusão do Estudo.

- O estudo concluiu que o SI (*Stickiness Index*) tende a aumentar quando a diferença de tempo entre as rotas disponíveis é alta (por exemplo, quando o tempo de uma viagem por uma determinada rota é significativamente menor que as restantes), concluiu ainda que este efeito positivo torna-se ainda mais marcante quando o número de rotas alternativas é reduzido.
- Saber quais rotas estão associadas a níveis de aderência mais elevados ajuda as empresas de transportes a escalonar um maior número de recursos para essas geolocalizações de modo a satisfazer melhor a elevada enchente de passageiros que opta pelas rotas. Além disso, as rotas

identificadas como possuindo um SI reduzido podem ser um indicador de onde ajustes no cronograma podem (tendencialmente, devem) ser feitos.

- Tabela de resultados disponibilizada pelos investigadores:



○



9. Conclusões.

- O grupo mostrou-se capaz de escalonar todos os **Drivers** a **WorkBlocks**, criando assim **DriverDuties**. Apesar de mais avanço não se ter mostrado hábil o suficiente para implementar um algoritmo capaz de trabalhar estes escalonamentos de forma automático, isto é, um algoritmo que garantisse que as **Hard Constraints** pedidas são respeitadas e/ou que assegurasse algum mecanismo para tentar suavizar, no caso de não ser possível essas mesmas **Hard Constraints**.
 - O grupo, entretanto, começa o escalonamento por verificar que a Capacidade do Sistema é superior à Carga do Sistema, de modo, a que... após o escalonamento se algum motorista não respeitar as **Hard Constraints**, ou mesmo adoecer/não comparecer (caso real), um outro motorista poderá ser escalonado no seu lugar.