UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE AUTOMAÇÃO E SISTEMAS

Eduardo Rauh Müller

OPTIMAL ARRIVAL TIME SCHEDULING OF AUTOMATED VEHICLES AT INTERSECTIONS

Florianópolis

2018

Eduardo Rauh Müller

OPTIMAL ARRIVAL TIME SCHEDULING OF AUTOMATED VEHICLES AT INTERSECTIONS

Tese submetida ao Programa de Pós-Graduação em Engenharia de Automação e Sistemas para a obtenção do Grau de Doutor em Engenharia de Automação e Sistemas.
Orientador
Universidade Federal de Santa Catarina: Prof. Dr. Rodrigo Castelan Carlson
Coorientador
Universidade Federal de Santa Catarina: Prof. Dr. Werner Kraus Jr.

Florianópolis

2018

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Müller, Eduardo Rauh
Optimal arrival time scheduling of automated vehicles at intersections / Eduardo Rauh Müller ; orientador, Rodrigo Castelan Carlson, coorientador, Werner Kraus Junior, 2018.
178 p.

Tese (doutorado) - Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós Graduação em Engenharia de Automação e Sistemas, Florianópolis, 2018.

Inclui referências.

1. Engenharia de Automação e Sistemas. 2. Gerenciamento cooperativo de interseção. 3. Veículos automatizados. 4. Otimização. 5. Escalonamento de veículos. I. Carlson, Rodrigo Castelan. II. Kraus Junior, Werner. III. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Engenharia de Automação e Sistemas. IV. Título.

Eduardo Rauh Müller

OPTIMAL ARRIVAL TIME SCHEDULING OF AUTOMATED VEHICLES AT INTERSECTIONS

Esta Tese foi julgada aprovada para a obtenção do Título de "Doutor em Engenharia de Automação e Sistemas", e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia de Automação e Sistemas.

Florianópolis, 03 de julho 2018.

Prof. Dr. Werner Kraus Jr.
Coordenador
Universidade Federal de Santa Catarina

Banca Examinadora:

Prof. Dr. Rodrigo Castelan Carlson
Orientador
Universidade Federal de Santa Catarina

Prof. Dr. Cláudio Barbieri da Cunha
Universidade de São Paulo

Prof. Dr. Fábio Baldissera
Universidade Federal de Santa Catarina

Prof. Dr. Rômulo Silva de Oliveira
Universidade Federal de Santa Catarina

Prof. Dr. Eduardo Camponogara
Universidade Federal de Santa Catarina

To Yara

ACKNOWLEDGEMENTS

I feel like it wouldn't be fair to begin in any other way than by thanking my parents. They always had a specially large participation in my education. Being the child of two professors, this couldn't really be any different.

I am very grateful to my supervisor Rodrigo and my co-supervisor Werner. Their support goes both beyond and farther back than this thesis, and is much appreciated. I would also like to thank the other professors and staff from DAS and PGEAS for being involved in my studies since 2006, specially the members of my thesis committee.

I thank my friends and colleagues from UFSC with whom I shared many hours both inside and outside of class. There are quite a few people who went through there with an interest in traffic control or related subjects, and I hope this group keeps growing.

I also want to thank the people I worked with in PLAMUS at the very beginning of this doctorate. I learned a lot with you, and it ended up being a very fun project.

I am grateful to the people from KTH for the opportunity to study there during part of my doctorate in 2017, specially my supervisor at KTH, Bo. I would also like to thank Jonas, Pedro and Xiao for the support and advice they gave during my stay over there.

I would like to thank everyone in the automatic control department from KTH, as well as all the friends I made or met again in Europe for making me feel welcome and at home in Stockholm. I'd like to specially thank my sister Aline, my cousins Melissa and Rachel and their beautiful families, my friends from Forró in Stockholm and Jette for making my stay so enjoyable. It goes without saying that all the friends from Florianópolis are also much deserving of a thank you, for making my life here that much better.

I thank CAPES and CNPq for funding my studies.

Finally, I'd like to extend my thanks to my friends and family as a whole. Your participation in my life – not only during these four years – is much appreciated. My main motivation for working with traffic is to improve people's lives, and you are always the first people I think on. There are far too many people deserving of a special mention.

RESUMO

A perspectiva de um ambiente urbano em que os veículos sejam automatizados, conectados e cooperativos motiva o estudo de novas estratégias de coordenação que têm o potencial de trazer ganhos significativos à segurança e eficiência viária. Embora várias estratégias já tenham sido propostas, a maioria delas emprega suposições simplificadoras que tem também o efeito te torná-las excessivamente restritivas, ou não levam indicadores de desempenho em consideração explicitamente. Além disso, nota-se na literatura a falta de estudos de como parâmetros do controlador – tais como o tamanho da área de controle e headways mínimos – afetam a eficiência do tráfego. Em geral, também não há discussão suficiente sobre complexidade computacional ou capacidade viária. Para lidar com essas questões, o Escalonamento Ótimo do Horário de Chegada (OATS, Optimal Arrival Time Scheduling) é proposto como uma nova estratégia de coordenação que minimiza o horário de chegada dos veículos numa interseção isolada e garante que não haverá colisões, sem ser excessivamente restritiva. Na Estratégia OATS, o problema de coordenação é decomposto em quatro subproblemas. Em particular, o problema de definição dos horários de chegada na interseção é modelado como um programa linear inteiro misto que minimiza o tempo agregado que os veículos levam para chegar à interseção. Simulações microscópicas de tráfego são realizadas e o desempenho do OATS é avaliado em diversas demandas de tráfego e configurações do controlador. Observa-se que um aumento do headway mínimo afeta negativamente tanto a eficiência do tráfego quando a complexidade computacional. Resultados de simulação mostram que uma interseção operando com a estratégia OATS é capaz de servir mais do que o dobro da capacidade de uma interseção semaforizada convencional, enquanto os veículos ficam sujeitos a atrasos mais do que uma ordem de magnitude menores. O emprego de áreas de controle pequenas, assim como outras simplificações estudadas, reduz significativamente o tamanho do problema, em troca de um ligeiro aumento nos tempos de atraso dos veículos. Nas condições estudadas, o emprego da estratégia OATS permite uma grande eficiência viária, e o tempo de execução é pequeno o suficiente para uma implementação em tempo real mesmo com uma demanda de tráfego elevada. A decomposição em subproblemas permite também formular o problema de *motion planning* usando apenas restrições lineares. Formulações alternativas que permitem que veículos desviem do horário de chegada programado para minimizar uma função custo que leva energia em consideração também são avaliadas. Resultados de simulação mostram que, para demandas elevadas de tráfego, a alternativa que traz melhores resultados do ponto de vista energético é a que garante que os veículos chegam à interseção o mais rápido possível. Isso sugere que, sob condições de tráfego intenso, minimizar os horários de chegada dos veículos à interseção tem também o efeito de reduzir o gasto energético.

Palavras-chave: Gerenciamento cooperativo de interseção. Veículos automatizados. Otimização. Escalonamento de veículos.

RESUMO EXPANDIDO

Introdução

Sistemas de transporte são essenciais para a sociedade. A maneira como pessoas, bens e serviços se deslocam tem profundas consequências sociais, economicas e ambientais. Em todo o mundo, sistemas de transporte ineficientes causam perdas significativas. Em especial, isso ocorre no sistema viário. A perspectiva de um ambiente urbano em que os veículos sejam automatizados, conectados e cooperativos motiva o estudo de novas estratégias de coordenação que têm o potencial de trazer ganhos significativos à segurança e eficiência viária. Embora várias estratégias já tenham sido propostas, a maioria delas emprega suposições simplificadoras que tem também o efeito te torná-las excessivamente restritivas, ou não levam indicadores de desempenho em consideração explicitamente. Além disso, nota-se na literatura a falta de estudos de como parâmetros do controlador – tais como o tamanho da área de controle e headways mínimos – afetam a eficiência do tráfego. Em geral, também não há discussão suficiente sobre complexidade computacional ou capacidade viária. Para lidar com essas questões, o Escalonamento Ótimo do Horário de Chegada (OATS, *Optimal Arrival Time Scheduling*) é proposto como uma nova estratégia de coordenação de veículos automatizados numa interseção isolada que minimiza o horário de chegada dos veículos na interseção e garante que não haverá colisões, sem ser excessivamente restritiva. Na Estratégia OATS, o problema de coordenação é decomposto em quatro subproblemas. Em particular, o problema de definição dos horários de chegada na interseção é modelado como um programa linear inteiro misto que minimiza o tempo agregado que os veículos levam para chegar à interseção, tendo uma forma similar a um problema de escalonamento de tarefas.

Objetivos

Em face às deficiências observadas no estado da arte, o objetivo dessa tese é propor e estudar uma estratégia de coordenação de veículos automatizados em interseções isoladas que satisfaça uma série de requisitos desejáveis levantados – em especial, a estratégia deve ser segura e eficiente – e estudar essa estratégia em diversas configurações de controle e considerando uma demanda de tráfego elevada. Isso é motivado pelo desejo de estudar o potencial que uma estratégia desse tipo tem em melhorar as condições de tráfego, uma vez que não foi encontrada na literatura uma estratégia que satisfaça suficientemente bem os requisitos desejados. Tais requitos são: garantir a auxência de colisões entre veículos; prover uma solução, preferencialmente ótima, em relação a um indicador de desempenho relevante e em tempo razoavelmente curto para uma aplicação em tempo real; não ser excessivamente restritiva, no sentido de não se apoiar em hipóteses simplificadoras que resultem num grau de liberdade significativamente menor para o movimento dos veículos ou que limitem consideravelmente a eficiência que pode ser obtida; usar um modelo de tráfego suficientemente detalhado para obter resultados que descrevam o movimento de veículos individuais; seja generalizável para qualquer configuração viária; e seja livre de bloqueio (*deadlock*). Os seguintes tópicos são abordados nessa tese: (i) A estratégia OATS é proposta formalmente; (ii) As condições de tráfego e esforço computacional necessário para solucionar o problema considerado são analisados em simulação para diversas configurações de controlador; (iii) várias simplificações são propostas para reduzir o tamanho do problema de escalonamento que faz parte da OATS; e (iv) formulações alternativas para o subproblema de *motion planning* são estudadas, analisando-se a troca que elas proporcionam entre atraso veícular e gasto energético.

Metodologia

A estratégia OATS é modelada formalmente como composta por quatro subproblemas resolvidos em sequência. Os subproblemas 3 e 4 (SP3 e SP4, respectivamente) podem ser vistos como a parte principal da OATS, enquanto SP1 e SP2 podem ser considerados como etapas de pré-processamento necessárias para se poder resolver SP3 e SP4. O subproblema SP3 é modelado como um problema de otimização linear inteiro misto onde os horários de chegada dos veículos são minimizados, e SP4 é um problema de controle ótimo para a obtenção das sequências de estados desejados e ações de controle necessárias para que os veículos cheguem no horário agendado. A OATS é estudada em ambiente de simulação microscópica de tráfego, utilizando o simulador Aimsun. Os subproblemas SP3 e SP4 são resolvidos utilizando o otimizador Gurobi. Ambos esses softwares possuem interfaces com a linguagem Python, que foi usada para implementar a estratégia OATS. É estudado um layout viário representativo de uma interseção simples, onde quatro vias de uma faixa com 200 m de comprimento cada se encontram num cruzamento. Diversas configurações de controle são analisadas, variando-se parâmetros que afetam a qualidade da solução (e consequentemente as condições de tráfego) e esforço computacional necessário. Na maioria dos casos são consideradas duas demandas distintas e relativamente elevadas de tráfego. A maior delas corresponde a aproximadante o dobro da capacidade fornecida por um semáforo convencional. Para cada cenário, são avaliadas as condições de tráfego, especialmente o atraso médio dos veículos, assim como o esforço computacional necessário para resolver SP3, que é efetivamente o gargalo da OATS em termos de tempo de processamento.

Resultados e discussão

Observou-se em simulação que a estratégia OATS é bastante eficiente, possiblitando a passagem de demandas superiores ao dobro da capacidade de um semáforo convencional, com atrasos relativamente pequenos. De forma geral, para uma configuração de controle tomada como padrão e bem representativa do que pode ser alcançado com a OATS, foram observados atrasos veiculares médios na ordem de 2.5 s para uma demanda de tráfego bastante elevada. Para efeito de comparação, o emprego de um controlador semafórico convencional resulta em atrasos superiores a 90 s para uma demanda menor do que a metade dessa. A redução do tempo de atraso, assim como do tempo total de viagem quando se emprega a OATS é bastante expressiva. O comportamento observado é que alguns veículos, ao se aproximarem da interseção, desaceleram somente o suficiente para garantir segurança, enquanto outros veículos são capazes de atravessar sem reduzir sua velocidade. De forma geral, a velocidade média observada é bastante próxima da velocidade máxima da via. Observou-se que certas simplificações que podem ser adotadas para a estratégia OATS – em especial a redução do tamanho da região de controle e a limitação do número de veículos considerados em uma instância do problema - tem um impacto relativamente pequeno na qualidade da solução. Reduzir a distância de controle de 100 m para 30 m levou a um aumento de apenas 0.33 s (ou 0.6%) no atraso médio veicular. Em contrapartida, essa redução da região de controle leva a uma redução do esforço computacional necessário para resolver SP3 em uma ordem de magnitude. De forma geral, com regiões de controle pequenas é possível resolver o problema consistentemente em menos de um 0.1 s, tempo suficientemente curto para uma aplicação em tempo real. Mesmo nos casos em que tal tempo é extrapolado, o emprego ocasional de uma solução subótima leva a uma deterioração inexpressiva das condições de tráfego. Foram realizados também experimentos em que se estudou a troca entre o atraso veicular e o gasto energético quando se permite que veículos disviem do horário de chegada programado por uma certa margem. Observou-se que, para uma instância isolada do problema, conceder tal margem de fato leva a uma redução do gasto de energia, em troca de um aumento no atraso médio. No entanto, quanto esse aspecto foi analisado em simulação de tráfego, observou-se que quando o número de veículos é elevado isso deixa de ser verdade. Para demandas de tráfego elevadas, a estratégia mais restritiva, que força os veículos a chegarem na interseção no tempo programado sem margem para desvios resulta também no menor gasto energético.

Considerações Finais

A estratégia OATS garante a auxência de colisões devido às restrições modeladas no problema, e é bastante eficiente, levando a condições de tráfego relativamente boas mesmo em cenários de demanda elevada. A separação da OATS em subproblemas a torna mais fácil de ser resolvida, computationalmente, do que uma estratégia de controle ótimo que englobe todos os aspectos num único problema. Apesar de SP3 ser um problema NP-dificil, há uma série de simplificações possíveis pela estrutura do problema que reduzem significativamente o número de combinações que precisa ser checada. Para o layout viário e demanda consideradas, o problema é tratável em tempo real. A desvantagem da decomposição é que não é possivel levar em conta as sequêcias de estados desejados e ações de controle necessárias para veículos no momento do cálculo dos horários de chegada, uma vez que SP3 lida somente com variáveis de tempo. Não é possível incluir diretamente métricas como gasto energético ou conforto nesse subproblema. No entanto, observou-se em simulação que para demandas de tráfego elevadas, a estratégia de minimizar o horário de chegada levou a um gasto energético menor do que permitir que os veículos desviem do horário programado. Isso sugere que minimizar o tempo total é também uma estratégia eficiente do ponto de vista energético para casos de demanda elevada. Há vários possíveis futuros desdobramentos dessa pesquisa. Em especial, deseja-se estudar a estratégia OATS com diferentes layouts viários e demandas de tráfego, e também a operação de multiplos cruzamentos coordenados por uma estratégia de alto nível.

Palavras-chave: Gerenciamento cooperativo de interseção. Veículos automatizados. Otimização. Escalonamento de veículos.

ABSTRACT

The perspective of an urban environment in which vehicles are automated and cooperative motivates the investigation of new coordination strategies that have the potential to result in significant gains to road safety and efficiency. Although several coordination strategies have already been proposed, most of them either adopt simplifying assumptions that have the side effect of making them excessively restrictive, or do not take a measure of performance explicitly into account. Besides that, there is a lack in the current literature on the topic of how control parameters – such as the size of the control region or minimum headways – affect traffic efficiency. In general, there is also little discussion about computational effort or road capacity. To address these issues, Optimal Arrival Time Scheduling (OATS) is proposed as a novel intersection coordination strategy that minimizes the arrival time of vehicles at an isolated intersection and guarantees that there will be no collisions without being overly restrictive. In the OATS strategy, the coordination problem is decomposed into four subproblems. In particular, the problem of defining arrival times at the intersection is modeled as a Mixed Integer Linear Program (MILP) that minimizes the total time vehicles take to reach the intersection. Microscopic traffic simulations are performed and OATS performance is evaluated for many different traffic demands and control configurations. Increasing minimum headways is observed to affect both traffic efficiency and computational complexity in a negative way. Simulation results show that an intersection controlled by OATS is capable of servicing more than double the capacity of an intersection controlled by conventional traffic lights, while vehicles experience delays more than one order of magnitude smaller. Employing small control areas, as well as other simplifications investigated, significantly reduces the size of the problem, in exchange of a small increase in vehicle delays. In the scenarios considered, the OATS strategy allows great road efficiency and has a sufficiently small execution time to be used in a real time application, even if traffic demand is high. The decomposition in subproblems also allows the formulation of a motion planning problem using only linear constraints. Alternative formulations that allow vehicles to deviate from the scheduled arrival times in order to minimize a cost function that takes energy expenditure into account are also evaluated. Simulation results show that, for high traffic demands, the formulation that has better results in regards to energy expenditure is also the one that guarantees vehicles arrive at the intersection as soon as possible. This suggests that, under heavy traffic conditions, minimizing arrival times also has the effect of reducing energy expenditure.

Keywords: Cooperative intersection management. Automated vehicles. Optimization. Vehicle scheduling.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

ITS	Intelligent Transportation Systems
OATS	Optimal Arrival Time Scheduling
ACC	Adaptive Cruise Control
CACC	Cooperative Adaptive Cruise Control
CIM	Cooperative Intersection Management
FIFO	First-In-First-Out
SP	Subproblem
SP1	Subproblem 1
SP2	Subproblem 2
SP3	Subproblem 3
SP4	Subproblem 4
IR	Intersection Region
CR	Control Region
IC	Intersection Controller
PNR	Point of No Return
$SP4_{A1}$	Alternative formulation 1 for SP4
$SP4_{A2}$	Alternative formulation 2 for SP4
$SP4_{A3}$	Alternative formulation 3 for SP4
PI1	Problem Instance 1
PI2	Problem Instance 2

LIST OF SYMBOLS

A	The set of lanes approaching an IR
a, b	Frequently used to refer to approaching lanes ($a, b \in A$)
a_i	Current acceleration of vehicle i
$a_{i,k}$	Acceleration of vehicle i during time interval k
a_i^{D}	Target acceleration for distance control of vehicle i
$a_i^{\max}$	Maximum acceleration of vehicle i
$a_{\mathrm{net}}^{\max,\min}$	Minimum acceleration any vehicle in the network must be able to achieve
$a_i^{\min}$	Minimum acceleration of vehicle i
$a_{\mathrm{net}}^{\min,\max}$	Minimum deceleration any vehicle in the network must be able to achieve
a_i^{S}	Target acceleration for speed control of vehicle i
C	The set of conflict regions inside the IR
C_m	The set of conflicting regions which are crossed by movement $m \in M$ ($C_m \subseteq C$)
$\hat{C}$	Set of conflict regions to be locked for pedestrian movement ($\hat{C} \subseteq C$)
D_a	Control Distance in approach $a \in A$. It is the distance from the intersection up to which vehicles are controlled by the Intersection Controller
$D_{\mathrm{a}}^{\min}$	Minimum control distance in approach a
D_{NR}	Distance of No Rescheduling
d_i	Distance of vehicle i from the IR
$d_{i,k}$	Distance of vehicle i from the IR at the end of time interval k
$\hat{d}_i$	Distance from the intersection vehicle i is expected to have, if he precisely followed the last speed profile received
F	Cost function of SP4
F_1	Cost function of $\mathrm{SP4}_{\mathrm{A1}}$
F_2	Cost function of $\mathrm{SP4}_{\mathrm{A2}}$
F_3	Cost function of $\mathrm{SP4}_{\mathrm{A3}}$
F_{IR}	Flow at the intersection
F_m	Flow through movement m
$\mathrm{F}_c^{\mathrm{cap}}$	Capacity flow through conflict region c

$\mathrm{F}_{\mathrm{IR}}^{\mathrm{cap}}$ Capacity flow at the intersection

g_i Gap between vehicle i and vehicle $i-1$ in front

$g_{\min}$ minimum safety gap between vehicles approaching the IR

h Minimum headway

$h_{a,i,b,j,c}$ The minimum time headway between the arrivals of vehicle i in approach a and vehicle j in approach b at conflict region c

h_c Average minimum headway between vehicles at conflict region c

h_{L} Default minimum longitudinal headway

h_{T} Default minimum transversal headway

h_c^{cap} Capacity headway of conflict region c that is located at an exit of the IR. This value may be imposed as the minimum headway if it is desired to restrict vehicle flow through one of the intersections exits

$h_{a,i,b,j,c}^{\mathrm{safe}}$ The minimum safety headway between vehicles i and j from approaches a and b in conflict region c

$h_{a,i,b,j,c}^{\Delta\mathrm{v}}$ Speed diference headway. This value is added to the safety headway in case the vehicle in front has a lower speed than the vehicle behind

$\hat{h}^{\Delta\mathrm{v}}$ Expected speed difference headway at conflict region c

i, j Frequently used as vehicle indexes

K_1 Gain for the ACC control law

K_2 Gain for the ACC control law

k Used as time interval index for motion planning problems

k_i^{t} Target interval. The control interval during which vehicle i is expected to arrive at the IR if following its schedule

L_i Length of vehicle i

$L_{\max}$ Maximum length a vehicle can have

M The set of movements allowed inside an intersection

M_c The set of movements $m \in M$ that cross conflict region c ($M_c \subset M$)

m Frequently used to denote movement inside the intersection

m_i Movement intended by vehicle i ($m_i \in M$)

N_{U} Maximum number of vehicles with an unbounded $t_{a,i}^{\max}$ from each approach to be included in SP3

n_a Number of vehicles inside approach a in the CR

n_i Number of control intervals that compose the control horizon for vehicle i in the motion planning problem

n_a^c Number of vehicles on control region a that will cross conflict region c

P_i Set of vehicles preceding vehicle i in the arrival order

P_c^{L} Probability of longitudinal interaction at conflict reigon c

P_c^{T} Probability of transversal interaction at conflict reigon c

Q A sufficiently large constant used to model disjunctive constraints in SP3

Q_1 A (sufficiently large) constant for modeling disjunction contraints in SP4

Q_2 A (sufficiently large) constant for modeling disjunction contraints in SP4

q^{cap} Capacity flow

S_i The speed profile vehicle i intends to follow inside the intersection

T The set of vehicle arrival times $t_{a,i}$ for each $i = 1, \ldots, n_a$ and each $a \in A$

T_{d} Desired time headway for the ACC algorithm

T_{S} Control time step, or control interval

$T_{\mathrm{max}}^{\mathrm{opt}}$ Maximum allowed running time when solving a problem instance

$t_{a,i}$ The time vehicle i on approach a takes to reach the IR, counting from the current time

$t_{a,i,c}$ The time vehicle i on approach a takes to reach conflict region c, counting from the current time

t_{lock} Time to lock a set of conflict regions of an intersection for pedestrian movement

t_{free} Time to free conflict regions that were locked for pedestrian movement

$t'_{a,i}$ Approximated time of arrival of vehicle i at the IR

$\hat{t}_{a,i}$ Time instant vehicle i in approach a is scheduled to arrive at the intersection, counting from the current time, when considering that a certain amount of time has already passed since the schedule was first calculated. This is equal to $t_{a,i}$ minus the time elapsed since the schedule that resulted in $t_{a,i}$ was calculated

$t_{a,i}^{\mathrm{low}}$	Lower bound for the arrival time of vehicle i from approach a at the intersection
$t_{a,i}^{\mathrm{max}}$	Maximum feasible time of arrival at the intersection for vehicle i in approach a
$t_{a,i}^{\mathrm{min}}$	Minimum feasible time of arrival at the intersection for vehicle i in approach a
$t_{a,i}^{\mathrm{high}}$	Upper bound for the arrival time of vehicle i from approach a at the intersection
$t_{a,i}^{\mathrm{safe}}$	Earliest possible time for a safe arrival of vehicle i in approach a at the IR
v_i	Current speed of vehicle i
$v_{i,k}$	Speed of vehicle i at the end of time interval k
v_i^{in}	Speed of vehicle i when entering the intersection region
v_i^{max}	Maximum speed of vehicle i
$v_{\mathrm{L}}^{\mathrm{max}}$	Speed limit
$\hat{v}_i$	Speed vehicle i is expected to have, if he precisely followed the last speed profile received
W_{a}	Weight coefficient for acceleration
W_{d}	Weight coefficient for distance
W_i	Width of vehicle i
W_{max}	Maximum width a vehicle can have
W_{t}	Weight coefficient for time
W_{v}	Weight coefficient for speed
w_a	Weight coefficient attributed to every vehicle in approach a
$w_{a,i}$	Weight coefficient of vehicle i in approach a
α	Number of control intervals by which a vehicle is allowed to deviate from the scheduled arrival time
$\beta_{a,i,b,j,c}$	Auxiliary binary variable used to model the order of arrival at a conflict region. If vehicle i from approach a reaches conflict region c before vehicle j from approach b, then $\beta_{a,i,b,j,c} = 0$. Otherwise $\beta_{a,i,b,j,c} = 1$.
δ_i^{min}	minimum distance between vehicle i and the vehicle immediately upstream as they approach the IR, enforced by SP4
$\varepsilon^{\mathrm{d,SP4}}$	Maximum allowed deviation for final distance in SP4
$\varepsilon^{\mathrm{v,SP4}}$	Maximum allowed deviation for final speed in SP4

$\mu_{i,k}$	Binary variable associated with vehicle i at time interval k in the motion planning problem. If vehicle i has not reached the intersection by the end of interval k, then $\mu_{i,k} = 1$. Otherwise, $\mu_{i,k} = 0$
ξ^{SP2}	Minimum size of the arrival interval in SP2
$\xi^{\mathrm{d,SP4}}$	Maximum allowed position deviation from the target trajectory in SP4
$\xi^{\mathrm{t,SP3}}$	Maximum allowed time deviation from the schedule in SP3
$\xi^{\mathrm{v,SP4}}$	Maximum allowed speed deviation from the target trajectory in SP4
$\tau_{i,k}$	Duration of time interval k of vehicle i for the motion planning problem
$\tau_{i,c}^{\mathrm{arrive}}$	Time for vehicle i to travel between the IR entrance and conflict region c
$\tau_{c}^{\mathrm{cross}}$	Expected time to cross conflict region c
$\tau_{c}^{\mathrm{enter}}$	Expected time to enter conflict region c
$\tau_{i,c}^{\mathrm{inside}}$	Time between vehicle i arrival at the IR and being completely inside conflict region c
$\tau_{i,c}^{\mathrm{out}}$	Time between vehicle i arrival at the IR and leaving conflict region c

SUMMARY

1 INTRODUCTION

Transportation is of vital importance to society. The way people, goods and services move has profound social, economic and environmental implications. Around the world, inefficient transportation causes significant losses. In particular, this is the case for road traffic.

Although we as a society have been expanding the road infrastructure and improving traffic control methods for a long time, congestion is a daily reality for a large number of people. In 2014, traffic congestion was responsible for an extra 6.9 billion hours of travel time and the consumption of over 11.7 billion liters of fuel in urban areas of the United States, for a congestion cost of 160 billion dollars (SCHRANK et al., 2015).

Traffic accidents were the cause of approximately 1.25 million deaths and up to 50 million non fatal injuries in 2013 (WORLD HEALTH ORGANIZATION, 2015). Although these numbers have plateaued since 2007 while world population and motorization have increased, they are still very high. Road traffic related injuries are estimated as the ninth leading cause of death and remain the number one cause of death among people between 15 and 29 years old.

In Brazil, there were 46,935 estimated road traffic fatalities in 2013. The death rate was 23.4 per 100,000 people, and has been increasing since 2009. Road traffic crashes are estimated to have caused economic losses of 1.2% of Brazil's GDP on 2013 (WORLD HEALTH ORGANIZATION, 2015).

A large amount of traffic accidents happen at intersections. In the United States, at least 47% of vehicle crashes in 2014 were intersection related (NHTSA, 2015). Intersections are critical points in road networks, as they constitute bottlenecks for traffic flow and are regions where the interactions between road users are particularly complex.

Currently, the usual solution for managing traffic at busy intersections is the use of traffic lights to coordinate human drivers. However, significant advances in Intelligent Transportation Systems (ITS) (USDOT, 2015) and vehicle automation technology, as well as the growing interest in automated vehicles in the last decade, are giving a new direction to intersection management research. An environment where all vehicles are fully automated may soon become a reality (LEVINSON et al., 2016).

Automated vehicles should be able to react faster, more reliably and have access to much more information than human drivers.

Moreover, automated vehicles are not prone to being distracted, being reckless or subject to several other risk factors associated with accidents. Thus, it is reasonable to expect that automated vehicles can drastically reduce the number of road crashes and fatalities. Indeed, human error is currently pointed at as the critical reason of 94% of traffic accidents (NHTSA, 2014).

Moreover, automated vehicles can communicate with each other and the infra-structure in order to improve decision making. More than being autonomous, automation and communication technology enable vehicles to cooperate and make decisions in ways human drivers are not able to. A connected and cooperative vehicle environment enables sophisticated strategies for intersection management which can achieve better performance than current technology (i.e., traffic lights).

Given that (i) making traffic safer and more efficient, as well as less polluting, can bring large benefits to society; (ii) intersections are of particular interest for traffic control; and (iii) an automated, connected and cooperative vehicle environment enables new control strategies that were not possible before; there is a significant research interest on the problem of coordinating automated vehicles at intersections. Several control strategies were proposed in the last decades, with varying approaches and limitations. The different methodologies proposed include the use of multi-agent systems (DRESNER; STONE, 2008), Petri Nets (NAUMANN et al., 1997), tree search (LI; WANG, 2006), dynamic programming (YAN; DRIDI; MOUDNI, 2009), heuristics such as ant colonies (WU; ABBAS-TURKI; MOUDNI, 2012), and sophisticated optimization models (LEE; PARK, 2012; ZHU; UKKUSURI, 2015), to cite a few. The works of Li, Wen and Yao (2014), Chen and Englund (2016) and Rios-Torres and Malikopoulos (2017a) present some recent surveys on the topic.

However, while several different coordination strategies have been proposed, there has been little interest in investigating how control parameters or different configurations of such strategies affect traffic efficiency. Moreover, investigation of the traffic performance achieved and the computational effort required by these strategies is often lacking, with most papers focused on merely showing that the strategy they propose works. The problem of coordinating automated vehicles in a network consisting of multiple intersections has also received little attention.

One step in addressing these gaps in the state of the art is the development of a strategy for managing automated vehicles in an isolated intersection that is formulated in a sufficiently general way to allow the study, in simulation, of traffic performance and computational effort required under different control configurations. Additionally, an efficient and deadlock-free control strategy for the single intersection coordination problem can be employed in future research as part of a multi-level control strategy for a network composed by multiple intersections.

1.1 A BRIEF INTRODUCTION TO THE BROADER PROBLEM

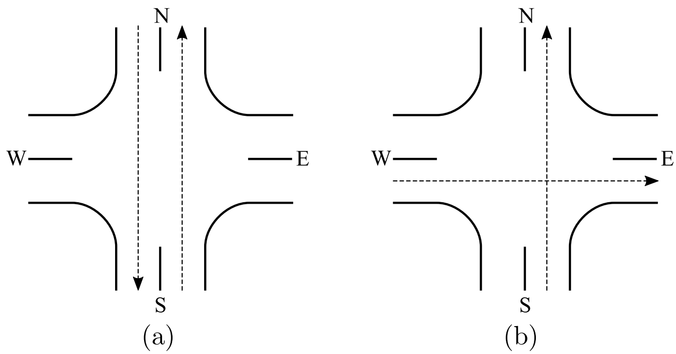
This thesis focuses on the problem of coordinating automated vehicles in an isolated intersection. One of the motivations for proposing a control scheme for a single intersection is the possibility of its use as part of a strategy for the broader problem of coordinating vehicles in a network.

It is well understood that the capacity of a typical signalized intersection depends, among other factors, of the control plan implemented. For instance, a control plan based on traffic lights with a longer cycle time generally yields a larger capacity than a control plan with the same ratios of green times, but smaller cycle length (PAPAGEORGIOU et al., 2003). Similarly, the capacity of an intersection where automated vehicles are coordinated by more sophisticated strategies can also be variable, depending on the control strategy used and its parameters. Consider the following examples:

- A common constraint for the intersection management problem is the enforcement of a minimum safety headway, which is a minimum time interval between two consecutive vehicles. The value of this headway has a direct impact on traffic capacity flow. If the minimum headway increases, capacity decreases. If vehicles follow a headway h while performing a certain movement in a road or intersection, the capacty flow q^{cap} of that movement is given by $q^{\text{cap}} = 1/h$.

- The origin and destination of the vehicles arriving at an intersection affects capacity. Suppose there are two vehicle streams approaching an intersection. Consider the case in which these two traffic streams do not have any potential conflict inside the intersection. In this situation, the movement of any vehicle that travels through one of these streams does not affect vehicles that travel through the other stream (e.g., one stream traveling from north to south and the other from south to north, not crossing at any point, as in Figure 1 (a)). Since the streams do not interact, the total capacity flow of vehicles traveling though the intersection is simply the sum of the capacity flow of each of the two traffic streams. Now, suppose that instead of two non-interacting vehicle streams, traffic demand is such that there are vehicles traveling through two conflicting traffic streams (e.g., one stream traveling from south to north and the other from west to east, crossing each other like in Figure 1 (b)). In this situation, both streams share a region of the intersection where they interact, which means the capacity flow is no longer simply the sum of the capacity flow of each of the two traffic streams. In fact, unless special circumstances apply[1], the total capacity flow in this case is lower than the sum of the capacity of each traffic stream.

Figure 1 – Example of (a) non conflicting and (b) conflicting movements at an intersection.



As a consequence, depending on the traffic state of the road network, it may be beneficial to direct vehicles in such a way that they follow less congested paths between their origin and destination in the network, taking into account the way conflicting vehicle streams interfere with each other at intersections. In this context, it could

[1] In this configuration, total capacity flow on the intersection will not be lower than the sum of the capacity flow of each movement only if the minimum headway between vehicles on the same stream is at least twice as large as the minimum allowed headway between vehicles on different streams. This hypothetical constraint is not very practical, so in the general case it is fair to assume capacity will be significantly lower.

be possible for a situation to arise in which it is beneficial for some vehicles to take paths that, albeit longer, lead to a lower number of conflicts or higher capacity. It is noteworthy that some of the possible movements inside an intersection might have more conflicts than others. For instance, in general left turns have a larger number of potential conflicts than right turns. Choosing paths for vehicles that reduce the number of potential conflicts may have a beneficial effect on network capacity.

In light of this, the research of which this thesis is inserted has the objective of developing and studying a strategy that coordinates automated vehicles inside a network, by both choosing vehicle's routes and managing vehicle movement in intersections.

The envisioned control structure consists of (i) a network level strategy for traffic assignment that takes into account the possibility that intersections can deliver different capacities depending on how traffic demand is composed (in terms of origin/destinations); as well as (ii) an intersection level coordination strategy that allows vehicles to cross the intersection in an efficient and safe manner.

The purpose of such structure is to allow each individual intersection to operate independently, requiring little to no information from the (higher level) traffic assignment strategy. This allows the system to be easily scalable, as each logical intersection can operate independently from the others.

1.2 PROBLEM STATEMENT

Ideally, each individual intersection should be controlled by an efficient strategy that guarantees safety, in the sense of avoiding collisions between vehicles, and can operate indefinitely, avoiding deadlocks. The following requirements are proposed for a local intersection management strategy for automated vehicles:

- It guarantees that no two vehicles collide with each other.
- Provides an (ideally optimal) solution in respect to a relevant and measurable metric of traffic performance, in a reasonable time.
- Is not excessively restrictive, in the sense that: (i) it should not stop vehicles from entering the intersection if it is safe and efficient to do so; and (ii) should not rely on a predefined vehicle arrival order or an heuristic for defining arrival order that does not guarantee good performance.

- Takes a realistic, microscopic traffic model into account, giving results that are in a sufficiently low abstraction level to be implemented in microscopic simulation or a real scenario by providing commands that can be directly followed by individual vehicles. A (very) high level strategy, that results in macroscopic values such as aggregated flow, is not as desirable, as translating those aggregate values into commands for individual vehicles is another problem in itself.

- Can be applied to any intersection layout.

- Is deadlock free assuming free flow downstream, and does not block the intersection area even in the case in which there is a blockage downstream.

Although coordination of automated vehicles in isolated intersections has been the subject of several studies, a control strategy that fully satisfies all of the proposed requirements was not found in the literature.

Moreover, it is also of interest to study the effects of control parameters on traffic behavior. For instance, questions such as (i) how much does the minimum headways affect traffic efficiency; or (ii) what effect does the size of the control region has on traffic efficiency; should be investigated. These issues are relevant for any intersection management strategy, but have received little attention in the literature. Evaluating an intersection management strategy under different conditions can shed some light on the impact these aspects have in traffic behavior.

1.3 OBJECTIVES AND APPROACH

In light of the previously exposed, the objective of this thesis is to propose a strategy for the coordination of automated vehicles in isolated intersections that satisfies the requirements proposed on Section 1.2; and to study this strategy under varied control configurations.

This thesis proposes and studies a novel strategy for the coordination of automated vehicles in isolated intersections: Optimal Arrival Time Scheduling (OATS). OATS coordinates vehicles by calculating a sequence of control inputs for each vehicle that allows them to cross the intersection without colliding with each other and with minimum aggregate time.

The following topics are addressed in this thesis:

- OATS is proposed as a novel strategy for managing automated vehicles arriving at an isolated intersection that satisfies the requirements proposed in Section 1.2. OATS is formulated as composed by four subproblems, the most important ones being a scheduling problem and a motion planning problem.
- Traffic efficiency and the computational effort needed to solve OATS are studied are investigated in simulation under different control configurations. A realistic, microscopic traffic simulation environment and high traffic demands are considered.
- In particular, since the scheduling problem is effectively the bottleneck of the OATS strategy in regards to computational effort, several possible configurations for making this particular problem smaller and hence easier to handle (such as reducing the size of the control region) are discussed and investigated in simulation. Some of these configurations are found to result in a trade-off between traffic efficiency and computational effort, while others are found to bring no significant benefit.
- Three different formulations are proposed for the motion planning problem and evaluated in simulation, in order to study the trade-off between vehicle delay and energy consumption when vehicles are allowed to deviate from the scheduled arrival time.

1.4 DELIMITATION

This thesis proposes OATS as a control strategy for the coordination of automated vehicles in isolated intersections. Although the proposed formulation is fairly general, allowing for any arbitrary junction to be modeled[2], throughout this thesis an urban cross intersection is used as an example and in simulations. Moreover, only small passenger vehicles (cars) are considered in simulation, even though OATS' formulation can account for heterogeneous traffic.

The possibility of vehicles having different priorities and the implementation of pedestrian crossings are discussed, although no simulations were conducted under these conditions.

OATS as proposed can only be implemented if all vehicles are automated, connected and willing to cooperate. The possibility of ha-

[2] Furtado (2017) has examined a previous, less sophisticated version of the OATS strategy (as presented in Müller, Carlson and Kraus Jr. (2016b)) in different road layouts, including a merge, one and two-lane cross intersections, and a roundabout.

ving vehicles that do not comply with the Intersection Controller is not considered.

To simplify the problem, communication is assumed to be perfect (lossless and instantaneous) and vehicles are assumed to be able to change acceleration instantly. The formulation proposed, however, includes tolerances that can be used to account for deviations. Limitations on acceleration capabilities could be accounted for with relatively straightforward modifications on the second and fourth subproblem that compose OATS.

Traffic conditions downstream the intersections are assumed as free flow. However, a mechanism is included in the formulation which can be used to limit outflow from the intersection and avoid a deadlock when vehicles can not exit the intersection.

OATS, as presented in Chapter 3, takes into account only vehicle arrival times as a measure of traffic conditions. Other metrics, such as energy efficiency, are not taken into account in the proposed formulation. Alternative formulations for the scheduling problem, capable of taking other metrics into account are discussed in Chapter 5.

1.5 THESIS OUTLINE

Chapter 2 provides a brief discussion of some concepts relevant for the work at hand and an overview on the literature on automated intersection management

Chapter 3 presents the OATS strategy. It begins with the motivation for proposing OATS, followed by an overall description of the concept behind OATS. The intersection and vehicle models are discussed, and the control goals are specified. Each of the four subproblems that compose OATS is presented, with special focus on the third (scheduling) subproblem. In particular, several possible simplifications that can be adopted to reduce the size of the scheduling problem are detailed. Optional modifications on the formulation to account for pedestrians or priority are discussed, as well as aspects such as liveness of the strategy and a simplified capacity analysis. Lastly, the control structure and the interactions between vehicles and the four subproblems are discussed.

Chapter 4 presents the results of simulations performed to investigate how traffic behaves when an intersection is managed by OATS under different control configurations and demands. The chapter begins by discussing the simulation setup. Network layout, traffic demand and parameters used are presented, as well as the different scenarios and performance indexes used. Simulations are performed for several different values of each of the parameters chosen, and results are discussed.

Chapter 5 discusses the possibility of using alternative formulations for the motion planning problem, instead of the one presented in Chapter 3 and used in Chapter 4. These formulations differ in how strict they are at keeping with the arrival schedule provided by the scheduling problem, and allow the formalization of a compromise between arrival time and energy efficiency. They provide different trade-offs between vehicle arrival time (or delay) and energy efficiency, which are examined in simulation. Each formulation is both evaluated by itself, by examining the results of solving one particular problem instance; and also evaluated in traffic simulation.

Finally, Chapter 6 concludes this thesis, with some final remarks and comments on future research.

2 BACKGROUND

This chapter presents several concepts which are relevant for the work at hand, with special focus on cooperative intersection management.

Sections 2.1 and 2.2 present the concepts of automated and cooperative vehicles, respectively. Section 2.3 explains the difference between longitudinal and transversal collisions. Section 2.4 presents Adaptive Cruise Control, an strategy for keeping automated vehicles at a safe distance from each other. Section 2.5 is a brief literature review on cooperative intersection management, and Section 2.6 discusses the motion planning problem.

2.1 AUTOMATED VEHICLES

Automated vehicles are those in which the driving task is performed (at least partially) by an automated system, instead of being completely the responsibility of an human driver.

Vehicle automation can be classified on six distinct levels according to the standard SAE J3016 (SAE, 2014). Level 0 corresponds to no automation at all, and each subsequent level has vehicles with increasing automation capabilities.

The highest automation level on the SAE J3016 scale is level 5, which corresponds to full automation. It consists of vehicles capable of performing all components of the driving task without interference from an human driver, under any environmental and roadway condition. Level 4 – high automation – corresponds to vehicles capable of executing the driving task without human interference in a limited set of driving modes. In this context, a driving mode is a set of driving scenarios with certain common characteristics and requirements. For example, some possible driving mode categories could be: high speed cruising, parking, intersection crossing, etc. By definition, if a vehicle is capable of performing every possible driving mode without human interference, it would be considered to be at level 5.

In the context of implementing an intersection control system, it is desirable for the system to have full control over vehicle behavior, otherwise it would be subject to driver's decisions and very large uncertainties. To achieve this, vehicles should be at least highly automated (level 4), and capable of all the driving modes necessary for approaching and crossing an intersection. Level 5 is not actually necessary, as the capability of the vehicle of performing others, non related driving modes – such as parking, or taking over on a freeway – is not relevant for intersection control.

It is noteworthy that the expression *autonomous* vehicle has been used, particularly in non technical media, to refer to vehicles in which the driving task can be performed solely by automated systems without the interference of human drivers. However, the term *automated* should be preferred in this context (SHLADOVER, 2009, 2017).

2.2 COOPERATIVE VEHICLES

In the context of automated vehicles, a distinction can be made among autonomous and cooperative systems (SHLADOVER, 2009, 2017). In a cooperative system, vehicles are capable of sharing information with each other and with the infrastructure in real time (via V2V – vehicle to vehicle –, and V2I – vehicle to infrastructure – communication), and are also willing to cooperate. Strictly autonomous systems, on the other hand, do not account for this type of interactions (i.e., autonomous vehicles, by this definition, do not explicitly cooperate nor share information with each other). Recent developments in communication technology, such as 5G (SHAH et al., 2018), allow for new and efficient forms of V2V and V2I communication.

Although there are some niche contexts in which an autonomous system can be preferable, specially for military applications in which communication can pose a security risk, there are several advantages for the use of cooperative systems in an urban environment (SHLADOVER, 2009), including:

1. Access to additional information, such as:
   (a) Information beyond the detection range of the sensors of a single vehicle.
   (b) Information about characteristics of other vehicles that can not be measured externally, or that are difficult to measure externally.
   (c) Information regarding road conditions, measured directly by equipment embedded in the infrastructure.
2. More reliable data in general, and the possibility of cross checking data gathered by multiple sources. This leads to less uncertainty.

3. "Misunderstandings", or errors in predicting the behavior of other vehicles are avoided by simply having vehicles share their intentions.

4. Vehicles can negotiate and plan maneuvers together.

5. It becomes possible to have some kind of "intelligence" in the infrastructure that uses global information, or at least information of the state of a large region to coordinate vehicles that individually have access to only local information.

These advantages result in the possibility of designing systems that are safer and more efficient. It becomes possible for vehicles – or a system located in the road infrastructure – to identify possible problems before they become apparent to vehicles that have access to only local information. It is also possible to implement sophisticated coordination or control strategies that would not be possible with autonomous vehicles.

Because of these advantages, in general, studies on intersection management with automated vehicles assume vehicles are cooperative, by both sharing information and being willing to cooperate with each other. This is also the case for this thesis.

2.3 COLLISION TYPES

Throughout this thesis, there is often a differentiation between two types of collisions involving a pair of vehicles: longitudinal collisions, and transversal collisions.

Longitudinal collisions are the ones that happen between vehicles traveling along the same direction on the same path (or vehicles for which their paths share a segment). Basically, it is a collision in which the front bumper of a vehicle behind touches the rear bumper of another vehicle in front. This type of collision can only happen if the vehicle behind maintains a higher speed than the vehicle in front for a sufficiently long time.

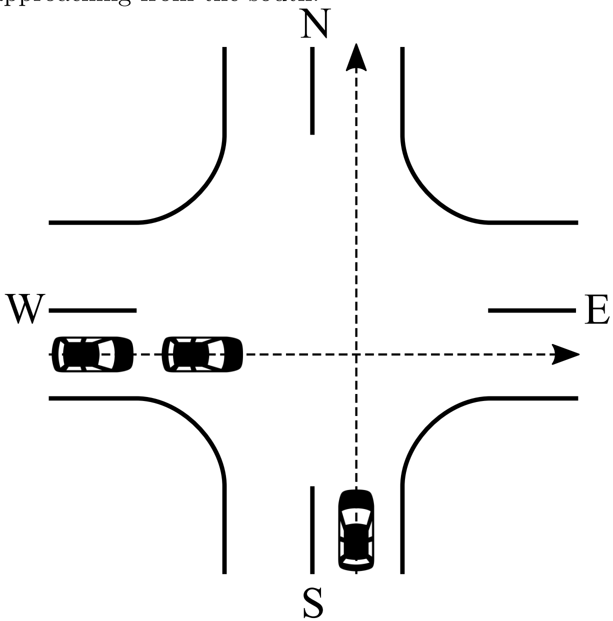
Transversal collisions, on the other hand, are the ones that involve vehicles traveling along different paths. These can happen if there is poor coordination between vehicles. Transversal collisions usually pose a higher safety risk, as the speed difference between the vehicles involved tends to be much higher than in longitudinal collisions.

In general, longitudinal collisions are easier to handle and avoid, since they depend on less factors. Avoiding transversal collisions can be a more difficult task, since it becomes necessary to consider the interaction between several vehicles instead of just each pair of adjacent vehicles on the same path.

In this thesis, if two vehicles that approach an intersection desire to perform movements that put then at risk of a transversal collision, it is said that the vehicles have a (potential) transversal conflict. Likewise, if a pair of vehicles is at risk of having a transversal collision, they are said to have a (potential) longitudinal conflict

Figure 2 depicts three vehicles at an intersection. The two vehicles traveling from west to east could suffer a longitudinal collision if the one in front travels with a lower speed than the one behind, and therefore have a longitudinal conflict. Each of them is also at risk of suffering a transversal collision with the vehicle traveling from south to north, and therefore there is a transversal conflict among the vehicle approaching the intersection from the south and each of the other vehicles.

Figure 2 – Vehicles in an intersection. The two vehicles approaching the intersection from the west have a longitudinal conflict with each other. Each of these vehicles also has a transversal conflict with the vehicle approaching from the south.



2.4 ADAPTIVE CRUISE CONTROL

Adaptive Cruise Control (ACC) is a control system that automatically adjusts the speed of a vehicle in order to maintain a safe distance from the vehicle ahead, and avoid longitudinal collisions. Even though the main goal of ACC is usually to increase safety, ACC can also increase traffic efficiency by having vehicles mantain shorter distances from each other than humans are safely capable of, which is possible since automated vehicles can react much faster and more reliably than human drivers. Typically, the response time of an ACC system is in the order of 0.1 to 0.2 s (KESTING et al., 2007).

ACC is implemented with feedback control, and can be present on autonomous vehicles, as all the needed information can be acquired from sensor data. Cooperative Adaptive Cruise Control (CACC) is an extension of ACC for cooperative vehicles. CACC also takes into account the speed of the first vehicle in a platoon, which can not be measured by vehicles other than the one just behind it.

In this thesis, it is assumed that automated vehicles which are not in the vicinity of an intersection behave as if they are being controlled by an ACC algorithm. The Control structure used is similar to the one proposed by Shladover (2012), which is equivalent to the one in Liang and Peng (1999).

In this ACC algorithm, vehicle acceleration a_i is given by

$$a_i = \min(a_i^{\mathrm{D}}, a_i^{\mathrm{S}}) \tag{2.1}$$

with a_i^{S} the acceleration calculated by the speed control algorithm, and a_i^{D} the acceleration for distance control. Speed control is implemented by a simple proportional control loop that has as reference the vehicle's desired speed (or the speed limit, whichever is lower), while distance control attempts to keep the vehicle a given (time) distance from the vehicle ahead. The resulting behavior is that a vehicle will attempt to travel at its desired speed, unless it is behind a vehicle with lower speed. In that case, it will follow the vehicle ahead with the same speed while keeping a certain distance. In practice, the values of the accelerations are saturated since vehicles have limited acceleration capability.

The distance control acceleration a_i^{D} is calculated by a control loop that aims at keeping the vehicle a desired time "distance" T_{d} from the vehicle ahead:

$$a_i^{\mathrm{D}} = K_1 \cdot (g_i - T_{\mathrm{d}} \cdot v_i) + K_2 \cdot (v_{i-1} - v_i) \tag{2.2}$$

with v_i and v_{i-1} the speeds of vehicle i and the preceeding vehicle $i-1$, g_1 the gap between the vehicles (i.e., the distance between the front bumper of the vehicle behind and the rear bumper of the vehicle in front), and K_1 and K_2 control gains. In the steady state, vehicles mantain a fixed time gap of T_d from each other, meaning vehicle i passes through the same positions of vehicle $i-1$ T_d seconds after it.

2.5 COOPERATIVE INTERSECTION MANAGEMENT

An intersection is a region in the road network where two or more roads meet or cross at the same level. Any policy or control strategy that aims at coordinating how vehicles behave at an intersection in order to promote a safe and efficient crossing can be called an intersection management strategy. Currently, the usual solution for managing busy intersection is the use of traffic lights.

Chen and Englund (2016) classify strategies for intersection management with cooperative vehicles as *cooperative intersection management* (CIM), and provide a detailed survey on the topic. The control strategy proposed in this thesis, OATS (which is detailed in Chapter 3), can be classified as a CIM strategy.

Several strategies that can be classified as CIM have been proposed in the last decades (the reader is refereed to the literature reviews provided by Li, Wen and Yao (2014) and Chen and Englund (2016) for a more in-depth discussion). Although they can differ significantly in regards to modeling approach, assumptions, and solution method, the problem considered is generally the same: allowing vehicles to safely cross an intersection, sometimes maximizing some measure of efficiency.

In general, the intersection can be viewed as a resource that is shared among vehicles. Chen and Englund (2016) classify CIM strategies in three broad categories according to the modeling approach:

1. strategies that model the intersection through space and time discretization;

2. strategies that use trajectory modeling; and

3. strategies that use collision region modeling.

The following sections present a brief characterization of each modeling approach, and discus some selected works on each category.

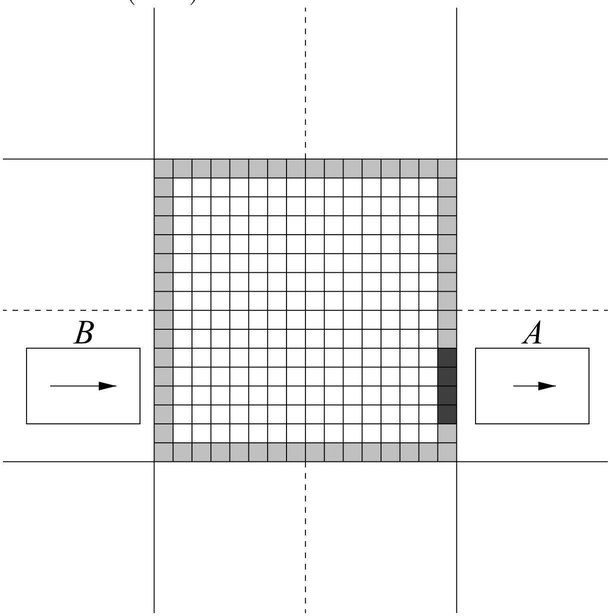
2.5.1 Space and Time Discretization

The so called space and time discretization strategies for CIM rely on modeling the intersection as composed by a set of cells, and dividing time in discrete slices. Vehicles can be granted the right to use a cell for a given time interval, and may not occupy cells for which they hold no "reservation". Only one vehicle may hold the right to use a given cell for a given time slice, and the full set of reservations held by a vehicle should allow it to cross the intersection. This transforms the CIM problem in the problem of allocating the use of discrete resources. Figure 3 illustrates an intersection divided in cells.

Notice that vehicles can not "reserve" an area smaller than a cell, or hold a cell for less time than the duration of a time slice. This means that reservations are usually "broader" than what is actually necessary, and there is usually some "waste".

The granularity of the discretization has a large effect on the

Figure 3 – Example of a intersection divided in cells. Figure from Dresner and Stone (2008).



complexity of the model and quality of the solution. Smaller cells and thinner time slices make the problem harder to solve, since the number of possibilities for the allocation of resources increases. On the other hand, having larger cells or time slices increases waste, and effectively make the problem more restrictive than necessary.

The system proposed by Dresner and Stone (2004) is based on the space and time discretization scheme, and is modeled through a multi agent system framework. A central agent is responsible for coordinating the passage of vehicle agents through the intersection. This agent can be seen as a *reservation* or *booking* agent. Vehicle agents decide the path they wish to take inside the intersection as they approach it and inform the booking agent of all the slots they need to *book* in order to perform the desired movement. If every requested cell is free on the desired time slices, the reservation is granted. Otherwise, the reservation is denied, and the vehicle must attempt to perform a new reservation in the future. If a vehicle has its requests repeatedly denied, it must stop before the intersection, as it can not enter it without the corresponding reservations.

The resulting behavior is relatively simple in that the right to cross the intersection is generally given to the vehicle that requests it first, and vehicles that have their requests denied may have to decelerate while they make new requests. One advantage of this approach is that it is very simple from a computational effort standpoint, as the central agent only needs to compare the requested reservations to a table it keeps with the current reservations. However, there is no attempt to optimize any criteria in the formulation.

The authors also explore some extensions for this control scheme in Dresner and Stone (2008). One such extension consists in making the booking agent capable of suggesting new, free routes for vehicles that have their requests denied, in order to avoid a large number of requests while vehicles try to find a free path through, basically, trial and error. Another development proposed is an extension of the system in order to control an intersection with mixed traffic, in which automated vehicles and human guided vehicles share the road. In this case, human driven vehicles follow traffic lights, while automated vehicles may cross during a red light if the booking agent determines that it is safe to do so.

Another extension for this strategy, proposed by Schepperle, Böhm and Forster (2007), allows vehicles to negotiate their reservations with each other in a process intermediated by an exchange agent.

Strategies that use a simpler intersection model and forbid any two vehicles from occupying an intersection simultaneously can be classified as a special case of space discretization, in which the intersection is composed by a single cell. This is the case of Wu, Perronnet and Abbas-Turki (2013), where the order in which vehicles are allowed to enter the intersection in defined with dynamic programming, with the goal of minimizing either an estimation of arrival time or the queue length.

The strategy proposed by Colombo and Vecchio (2015) can also be classified as modeling an intersection as composed by a single cell where all paths cross. The proposed supervisor is not concerned with optimizing performance, instead applying the minimum control action necessary to guarantee safety when vehicles are detected to be in a collision path. Vehicles (possibly human driven) are allowed to maintain whatever control action they wish as long as it does not result in an inevitable collision in the future (in which case the system overrides the driver's actions and applies a control that avoids collisions).

Another work that models the intersection as a single conflict region is the one from Oliveira et al. (2002). Vehicles are scheduled to arrive at the intersection in minimum time by a branch and bound algorithm. The problem of defining vehicle's arrival time (and order) is completely separated from the motion planning problem, which is solved by a simple heuristic.

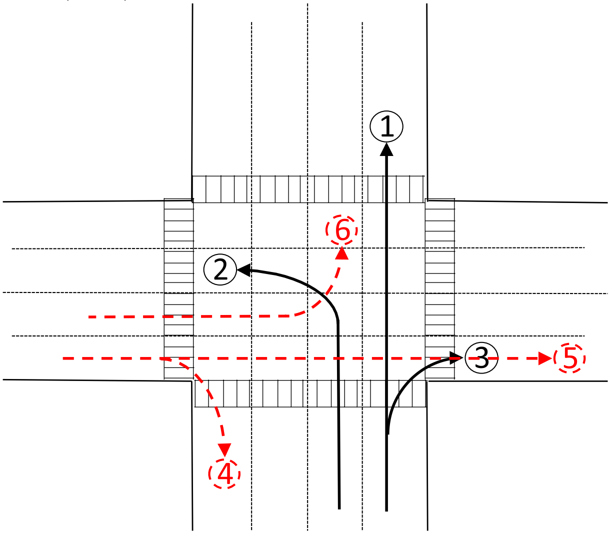
2.5.2 Trajectory Modelling

Trajectory modeling strategies take advantage of the fact that usually, vehicles cross intersections following certain patterns, or fixed paths. Figure 4 illustrates some possible movements performed by vehicles in an intersection. If the movements allowed inside the intersection are assumed to be fixed (depending only on the origin and desired turn of a vehicle), it is easy to check if any two vehicles have a potential transversal conflict.

Sets of non conflicting trajectories can be called a *safe pattern.* For example, in Figure 4, the sets of trajectories {1,2}, {1,6} and {2,4} form safe patters, unlike sets {2,5} or {2,6}. To check if two vehicles have a potential transversal conflict, it is sufficient to check if their paths form a safe pattern.

One possible strategy for avoiding transversal collisions at intersections consists in only allowing vehicles whose trajectories constitute a safe pattern to occupy the intersection area at any given time. Li and Wang (2006) use this condition to guarantee safety. Vehicles are grou-

Figure 4 – Example of trajectories in an intersection. Figure from Chen and Englund (2016).



ped according to their estimated time of arrival at the intersection, and the arrival sequence of each group is defined separately. Every possible sequence of safe patterns for the vehicles in the group is checked, and a *travel plan* with minimum time is elaborated for each sequence. The plan with the lowest travel time is chosen and followed by all vehicles in the group. This exhaustive search can be performed relatively fast if the groups are small, but since groups must cross the intersection in order, dividing vehicles in groups puts another arbitrary constraint on the problem, making it more restrictive than necessary.

Lee and Park (2012) model the problem of avoiding transversal collisions as a non linear optimization problem in which the objective function consists in minimizing the *overlap* of vehicles with potential transversal conflicts at the intersection, i.e., the time during which they occupy the intersection simultaneously. Given the formulation, any solution that avoids conflicts has the value of zero for the objective function, and usually there are multiple optimal solutions. The authors solve the problem using several tools in parallel, including solvers and genetic algorithms, and implement the first solution found. Although simulation results show that performance is better than that of traffic lights, this solution approach offers no guarantee of performance.

Allowing only safe patterns inside the intersection guarantees the absence of transversal collisions, but not of longitudinal collisions. These can be avoided by modeling additional constraints involving each vehicle and the vehicle in front on the same path. Interactions between vehicles whose trajectories merge inside the intersection are generally modeled as being a transversal conflict.

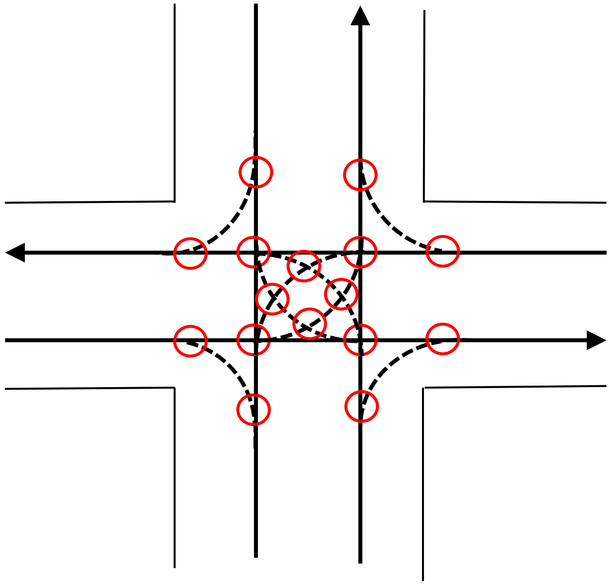
Notice that although only allowing vehicles whose trajectories constitute a safe pattern to occupy the intersection area at any given time guarantees transversal safety and is an easy condition to check, it results in a behavior that is more restrictive than necessary. Vehicle pairs whose trajectories do not form a safe pattern may actually be able to occupy the intersection area simultaneously and safely, depending on how their trajectories evolve in time. A formulation that does not allow that results in some vehicles waiting longer than what is actually needed to guarantee safety, hence some capacity is “wasted”. All things considered, this is a design decision that simplifies the problem, but results in some loss in efficiency. This simplification is present in most, if not all, works that use a trajectory modeling approach for formulating a CIM strategy, such as: Wu, Abbas-Turki and Moudni (2012), which uses an ant colony heuristic to reduce the time vehicles take to leave the intersection; Yan, Dridi and Moudni (2009), which uses dynamic programing, also with the goal of minimizing travel time; and Kowshik, Caveney and Kumar (2011), which assumes a predefined arrival order for vehicles.

2.5.3 Collision Region Modeling

The modeling of conflict regions can be seen as a combination of the other two aforementioned intersection modeling approaches. By assuming vehicles follow fixed paths inside an intersection, the intersection region can be discretized in a small number of critical points, where the paths cross and there is the possibility of a transversal collision. There is no need to check for transversal collisions outside these points. Figure 5 shows an intersection with highlighted conflict regions. If it is guaranteed that only one vehicle can occupy these regions at any given moment, then the absence of transversal collisions is also guaranteed. Longitudinal collisions can be guaranteed by controlling the distance between each pair of consecutive vehicles on the same path.

One example of this modeling approach is the strategy proposed

Figure 5 – Example of an intersection with conflict regions highlighted by red circles. Figure from Chen and Englund (2016).



by Naumann et al. (1997), possibly the first CIM strategy with conflict region modeling. The authors designed a distributed strategy in which a vehicle is only allowed to enter a conflict region if it holds a token associated to that region. When a vehicle leaves a conflict region it frees the token to be used by other vehicles. Once freed, a token passes to the highest priority vehicle that has requested that token, and the priority of a vehicle is calculated by a multi variable function that accounts for factors such as waiting time and queue size. Safety of the strategy is proven formally with the use of Petri Nets.

Zhu and Ukkusuri (2015) propose an integer program formulation for the problem and obtain an optimal solution in terms of minimizing road occupancy, although they use a macroscopic modeling approach in which traffic variables are considered constant for a road segment and individual vehicle behavior is disregarded.

Dai et al. (2016) use a modeling approach which can be categorized as either collision region modeling or space and time discretization, albeit with a small number of cells (16) and a sophisticated solution for the coordination problem. A cost function which takes metrics such as passenger comfort and deviation from a desired speed into account is optimized when calculating vehicle speed profiles. Vehicle order, however, is set by a rule that gives the right to pass first for vehicles that have been in the control region for a longer time.

The approach adopted by Gregoire, Bonnabel and Fortelle (2015, 2014) and Qian et al. (2015) is very interesting as it enforces a strong safety constraint. It is formally guaranteed that the system is always in a state in which, even if any vehicle suddenly applies maximum deceleration, there still exists a set of control inputs that avoids collisions. Vehicle order of arrival at the intersection is set according to a priority graph, which is defined using an heuristic that allow vehicles to cross in the order they request passage (i.e., First-In-First-Out).

Kamal et al. (2015) define a risk function that models the probability of a potential collision, which is taken into account in the cost function that is minimized to assign vehicle controls. Only a few vehicles very close to the intersection are optimized, and there is no guarantee of optimal solution.

The strategy proposed by Levin, Fritz and Boyles (2017) considers a more macroscopic traffic model, without explicitly modeling the state of individual vehicles; and maximizes the number of vehicles entering the intersection at the next time interval.

2.6 MOTION PLANNING

Motion planning is the problem of, given a desired movement task, finding a sequence of discrete states and control inputs for the vehicles (or agents) involved that satisfy movement constraints and reach the desired state (characterized by desired positons, speeds, etc). It often involves the optimization of some criteria. Intersection management strategies often involve motion planning to some degree.

In the context of intersection management, motion planning can be completely separated from the coordination problem and only used to elaborate a motion plan that reaches a predefined desired state, e.g., reach a given position, often at an intersection, at a given time and with a given speed. Thus, coordination aspects such as vehicle arrival order must already be set when the motion planning problem is solved.

Another possible approach is integrating the problem of choosing vehicle arrival order (or time of arrival) with the motion planning problem, instead of setting the aspects related to coordination beforehand.

Several works on automated intersection management use an optimal control framework to assign a motion plan, in the form of a sequence of control inputs (usually accelerations) for vehicles approaching an intersection (ZHANG; MALIKOPOULOS; CASSANDRAS, 2016; KAMAL et al., 2015; QIAN et al., 2015; ZANON et al., 2017; CAMPOS et al., 2017; HULT et al., 2016). Typically, vehicles are assumed to be able to keep on their lanes while they travel forward, and maneuvers such as overtaking or lane changing are not allowed. This simplifies the problem as only longitudinal motion has to be considered.

A common feature of most of these strategies is having a term in the cost function that minimizes the summation (or the integral, in the continuous time case) of the squares of the accelerations for each vehicle at each time step, as a proxy for energy (fuel) consumption. Acceleration is usually easier to model and measure than fuel consumption, specially when considering heterogeneous vehicles, and minimizing acceleration tends to produce smooth trajectories. According to Rios-Torres and Malikopoulos (2017b), under certain modeling assumptions there is a monotonic relation between acceleration and fuel consumption, leading to an almost equivalence in minimizing either of them.

Since minimizing acceleration alone could cause the optimal solution to involve vehicles taking an unreasonable time to cross the intersection, it is usual to either constrain the arrival time or include a term in the cost function that minimizes the deviation from a given desired speed.

There are various possible design decisions for modeling such motion planning problems. Vehicles may be allowed to cross the intersection in First-In-First-Out order (FIFO) (ZHANG; MALIKOPOULOS; CASSANDRAS, 2016; ZHANG; CASSANDRAS; MALIKOPOULOS, 2017), follow some pre-established priority relation (ZANON et al., 2017; CAMPOS et al., 2017; HULT et al., 2016; QIAN et al., 2015; GREGOIRE; BONNABEL; FORTELLE, 2014), or the arrival order may be optimized online (KAMAL et al., 2015; MÜLLER; CARLSON; KRAUS JR., 2016a, 2016b). The latter case, although more general, typically leads to non-convex formulations and NP-hard problems (COLOMBO; VECCHIO, 2012).

Optimization can be performed in a sequential manner with each vehicle constrained by the decisions of the previous vehicles (ZHANG; MALIKOPOULOS; CASSANDRAS, 2016; ZANON et al., 2017; CAMPOS et al., 2017; HULT et al., 2016). One possibility is to use a simple intersection model so that the entire intersection is considered a conflict region which can not be occupied simultaneously by vehicles performing conflicting movements (ZHANG; MALIKOPOULOS; CASSANDRAS, 2016; QIAN et al., 2015; ZANON et al., 2017; CAMPOS et al., 2017; HULT et al., 2016). Alternatively, elaborate models (considering conflict points or regions) allow for multiple vehicles to be inside the intersection area (GREGOIRE; BONNABEL; FORTELLE, 2014; KAMAL et al., 2015; MÜLLER; CARLSON; KRAUS JR., 2016b). In this case, vehicles with conflicting movements are allowed to be inside the intersection at the same time, as long as they do not occupy the same conflict point simultaneously. Safety is usually guaranteed by enforcing minimum (time) headways or (spatial) gaps. The approach proposed in Qian et al. (2015) and Gregoire, Bonnabel and Fortelle (2014) guarantees that the system is always on a *break safe* state, which is a system state on which even in the advent a vehicle unexpectedly starts applying maximum deceleration on the next time step, a feasible solution can still be found.

3 OPTIMAL ARRIVAL TIME SCHEDULING

This chapter presents Optimal Arrival Time Scheduling (OATS) as a novel strategy for intersection management in an environment in which vehicles are connected, highly automated and cooperative.

The problem of automated intersection management consists in coordinating the movement of vehicles as they approach and cross an intersection. OATS allows vehicles to cross the intersection efficiently and safely by minimizing the aggregated time taken to cross an intersection, while guaranteeing there are no collisions between vehicles.

The main characteristics of OATS that differentiate it from other intersection management strategies are the facts that it finds an optimal result in terms of vehicle travel time without (generally) resorting to an extensive search and makes few assumptions that restrict efficiency. In OATS, vehicle arrival order is not predefined or set by an arbitrary heuristic, but instead is optimized in order to minimize arrival times. Vehicles with potentially conflicting movements are not forbidden to be in the intersection simultaneously.

Section 3.1 presents the motivation for formulating OATS. Section 3.2 presents the OATS concept and gives a general outline of each of its components. Section 3.3 details the intersection and vehicle models, the underlying assumptions and discusses the control goals which OATS aims to accomplish. Sections 3.4, 3.5, 3.6 and 3.7 present each of the four subproblems that compose OATS. Special attention is given to SP3, with discussions on problem complexity, liveness and other relevant aspects. The necessary interactions between vehicles and the centralized intersection controller to implement OATS are detailed in Section 3.8.

3.1 THE CASE FOR OATS

None of the CIM strategies found in the literature and mentioned on Section 2.5 fully complies with all the requirements proposed in Section 1.2. More specifically, it is noteworthy to point out that no strategy was found that complies with the three following broad requirements:

1. Considers a microscopic model for traffic flow, in which the movement of each individual vehicle is considered and the results of the proposed strategy can be directly applied as commands given

to vehicles (i.e., some studies consider simplified, or macroscopic models for traffic flow).

2. Does not impose any of the following two types of simplifying constraints which may significantly impact traffic flow efficiency:

    - Constraints forbidding vehicles from occupying the intersection area simultaneously without checking if the evolution of vehicle dynamics in time and space would actually allow vehicles to share the intersection safely.
    - Arbitrary time discretization constraints, such as assuming vehicles cross an intersection in exactly one time interval or not allowing more than one vehicle inside the intersection at any interval.

3. Decides both vehicle arrival order and vehicle arrival times online, using a measurable and significant metric that relates directly to the efficiency of traffic flow (often, an heuristic is used for deciding vehicle crossing order, and it is not proven that following the heuristic optimizes traffic flow).

Among the selected works cited on Section 2.5, only six of them comply with the third requirement (WU; PERRONNET; ABBAS-TURKI, 2013; OLIVEIRA et al., 2002; LI; WANG, 2006; YAN; DRIDI; MOUDNI, 2009; ZHU; UKKUSURI, 2015; LEVIN; FRITZ; BOYLES, 2017). None of those also complies with the second requirement. In particular, Zhu and Ukkusuri (2015) and Levin, Fritz and Boyles (2017) also do not comply with requirement 1 by using a higher abstraction level for the traffic flow model, and Yan, Dridi and Moudni (2009) imposes additional constraints on traffic flow by effectively mimicking a traffic light behavior. Li and Wang (2006) considers small groups of vehicles at a time, effectively imposing a relatively small control region. Wu, Perronnet and Abbas-Turki (2013) and Oliveira et al. (2002) consider a very simple intersection model, in which there are only two traffic streams approaching an intersection and no two vehicles can be inside the intersection area at the same time.

In short, every CIM strategy found in the literature that attempts to optimize traffic flow either employs significant constraints to simplify modeling; considers only a very small number of vehicles; or assumes a very simple intersection layout.

This motivates the formulation of OATS as a strategy that complies to the proposed criteria. By optimizing a measure of traffic efficiency while not being subject to the aforementioned common simplifying constraints, it is expected that OATS should likely be more efficient than other CIM strategies. On the other hand, meeting all of the criteria may result in a formulation that has a high computational complexity. However, even if the resulting formulation turns out to be intractable for large problem instances, it is still worth investigating for the following reasons:

- It is possible that many typical, or practical problem instances are still sufficiently small to be solved quickly by such a formulation (in fact, as shown in Chapter 4, this seems to be the case).
- It can provide valuable insights for CIM in general.
- It provides a basis for comparing other CIM strategies.
- Design choices, such as the use of small control regions, can be used to ensure the problem will not involve a very large number of vehicles.

Moreover, OATS is also proposed with the goals of being sufficiently general to allow any intersection layout; and to investigate the efficiency and computational complexity of a CIM strategy under different control configurations. A CIM strategy which is viewed as a candidate for field application should be studied under various configurations, like varying headways or the size of the control region.

Finally, it is also of interest to evaluate the capacity flow, as well as traffic conditions when a CIM strategy is operating near capacity. For this reason, simulations should be performed in realistic scenarios, and with sufficiently high traffic demands. This is often not the case in the literature.

3.2 THE OATS CONCEPT

One core idea behind the OATS strategy is to decompose the overall problem of coordinating vehicles inside the intersection in four smaller subproblems that can be solved in sequence. The decomposition approach was inspired by the work of Oliveira et al. (2002), in which the scheduling and motion planning problems are also separated, albeit with very different and less general formulations for each problem[1]. OATS is composed by the following subproblems:

[1] In Oliveira et al. (2002), a very simple intersection layout is used, in which only two movements are allowed; the scheduling algorithm does not allow for more than one vehicle to be inside the intersection area at any time; vehicle access to the

1. Subproblem 1 (SP1) is the problem of defining a speed profile for each vehicle for the part of its path that is inside the intersection.

2. Subproblem 2 (SP2) is the problem of finding, for each vehicle, the feasible arrival interval during which it can reach the intersection.

3. Subproblem 3 (SP3) is the problem of scheduling vehicle arrival times at the intersection.

4. Subproblem 4 (SP4) is a motion planning problem where a suitable trajectory for reaching the intersection is defined for each vehicle.

SP1 is solved individually by each vehicle, while SP2, SP3 and SP4 are solved by a centralized controller that has information collected from all vehicles[2].

The core of the OATS strategy is the scheduling problem, SP3, in which vehicles are scheduled to arrive at the intersection with minimum time, while following constraints designed to guarantee safety. The inputs for SP3 are the time intervals during which vehicles can feasibly arrive at the intersection. When SP3 is solved, a schedule of arrival times at which vehicles must reach the intersection is produced. This schedule is used as input for the motion planning problem, SP4.

The motion planning problem, SP4, is the problem of, given a desired final state, finding a series of control inputs that allow vehicles to reach this desired state. In SP4, this state consists of arriving at the intersection at a certain time with a given target speed.

In the OATS approach, the motion planning and the scheduling problem are separated. This means that when the control inputs are calculated, vehicle arrival times (and hence, also arrival order) are already set by the scheduling problem. This greatly simplifies SP4, which can be formalized without the need for any non-linear constraint. It also means that SP3 only needs to decide vehicle arrival time (and order), without the need to take vehicle state into account, resulting in a formulation with a small number of continuous variables. Basically, by separating SP3 and SP4 in this manner, it is possible to formalize: (i) a scheduling problem (SP3) with few continuous variables (one arrival

intersection is constrained basically only by feasibility and minimum headways; and the motion planning problem consists of basically switching between minimum and maximum acceleration.

[2]This is a somewhat arbitrary choice, as it would possible to implement OATS by having SP1 and SP2 being handled by either individual vehicles or the centralized controler

time for each vehicle), and a (possibly large) number of binary variables, used to model vehicle order of arrival; and (ii) a motion planning problem with a large number of continuous variables (possibly hundreds of accelerations for each vehicle), but in which all constraints are linear. If those problems were combined into one instead of being separated, the resulting formulation would be a problem with both a very large number of continuous variables and also many binary variables.

The downside of this separation is that vehicle states and control inputs are not available to be used by SP3 when defining vehicle arrival times. This means that vehicle acceleration, energy consumption, etc, can not be taken into account when defining the schedule, and SP3 is concerned with only minimizing travel time. To some extent, however, the motion planning problem can be designed to take these metrics into account – this is investigated in Chapter 5.

For simplification reasons, it is assumed that vehicles follow fixed speed profiles while inside the intersection. This is a slight generalization of the common assumption of fixed speed inside the intersection (a fixed speed inside the intersection is actually a special case of a predefined speed profile inside the intersection). SP1 is the problem of finding such speed profiles, and is solved individually by each vehicle. The initial state of this speed profile is used as an input for both SP2 and SP4, as it corresponds to the final state of the vehicle for these problems.

Vehicles are assumed to choose profiles that allow them to cross the intersection safely and comfortably, in the sense that they will not choose a profile with a turning speed that is too high for the vehicle or passengers, or with accelerations that are too high. The vehicle should take into account its own characteristics, the speed limit, passenger preferences, and the geometry of the intended movement when defining a speed profile.

It is also assumed that vehicles choose a path that leads then to occupying the intersection for as little time as possible (i.e., vehicles are expected to chose a speed profile that minimizes the time spent inside the intersection). It can be argued that setting the speed profile of a vehicle inside the intersection is equivalent to imposing additional constraints on vehicle movement, which may lead to a less efficient solution than not imposing such constraints. However, it is reasonable to assume that having vehicles occupying the intersection area for as little time as possible is not significantly detrimental to the efficiency of a strategy that aims at minimizing overall crossing time.

The feasible arrival time intervals used as input for the scheduling problem are obtained by SP2, which solves linear motion equations to define when vehicles can possibly arrive at the intersection with a speed that is compatible with the desired profile inside the intersection.

SP3 can be seen as the core of the OATS strategy, and SP4 as the problem of translating the schedule obtained from SP3 into a motion plan that allows vehicles to effectively implement the schedule. These two problems are the focus of this thesis, while SP1 and SP2 are defined in a much simpler way, but are sufficient to guarantee that SP3 and SP4 are feasible. In fact, SP1 and SP2 can be seen as just auxiliary problems to obtain feasibility constraints for SP3 and SP4. Even so, they are classified as subproblems of their own for completeness, and they could be solved with much more sophisticated strategies than the ones presented.

The lowest level of control OATS reaches is that of sending vehicles a sequence of desired states (in the form of position, speed and acceleration) to be followed along a time horizon. Vehicles are assumed to be capable of following these trajectories by using them as reference for lower level control loops. OATS assumes vehicles are capable of instantly changing acceleration and can follow these trajectories perfectly. This is not actually the case in practice, but the deviations and uncertainties associated to trajectory following can be handled at the OATS level by sufficiently large headways and tolerances.

3.3 INTERSECTION AND VEHICLE MODEL

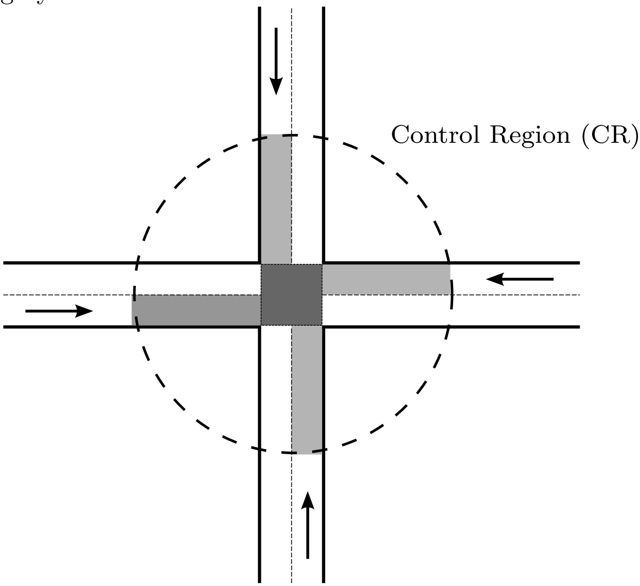
The OATS strategy manages vehicles movements while they approach and cross an intersection. Let the Intersection Region (IR) be the area where two or more roads intersect (i.e, the dark gray intersection square in Fig. 6).

Vehicles travel through fixed paths inside the IR. Given these paths and vehicle geometry, transversal collisions are only possible in areas inside the IR, called *conflict regions*. Guaranteeing that no collision occurs in any conflict region is sufficient to guarantee the absence of transversal collision in the IR. The conflict regions can ve viewed as resources shared among the vehicles.

Formally, the following definitions are used in order to describe an arbitrary intersection layout:

- A is the set of approaching lanes arriving at an intersection. They are refereed to simply as *approaches* on the text. Throughout this thesis, letters a and b are often used to refer to specific approa-

Figure 6 – Intersection layout. The Control Region is composed by the Intersection Region (the dark gray square where the roads cross) and the portion of the approaching lanes closer to the IR, depicted in lighter gray.

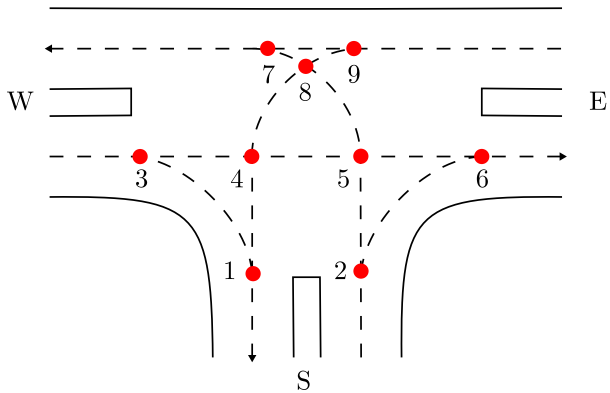


ching lanes ($a, b \in A$).

- M is the set of movements m allowed inside the intersection, each of which connects one approach to one exit along a fixed path;
- C is the set of conflict regions c inside the intersection;
- $C_m \subseteq C$ is the set of conflict regions which are intercepted by movement $m \in M$.

Figure 7 illustrates a sample intersection. Approaches are denoted according to their origin with $A = \{\text{W, E, S}\}$. Arrows depict the direction of traffic (in this case, vehicles travel on the right hand side of the road). The allowed movements are depicted by dashed lines, and can be denoted according to their origin-destination pair as: $M = \{$(W,E), (W,S), (E,W), (E,S) (S,W), (S,E)$\}$. These movements intersect in nine conflict points, depicted with red circles.

Figure 7 – An intersection with approaches $A = \{W, E, S\}$, movements (dashed lines) $M = \{(W,E), (W,S), (E,W), (E,S) (S,W), (S,E)\}$ and conflict points (red circles) $C = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$.



As the points where vehicle paths intersect inside the IR, the conflict points would be the only places where vehicles could have transversal collisions (assuming they only travel through fixed paths), in case vehicles were point-like objects. Since vehicles are not point-like objects, the concept of conflict points needs to be extended to conflict regions. Let a conflict region be a region in space where it is possible for vehicles to have a transversal collision while following the allowed movements in the intersection. Conflict regions can be defined as "tight" or "loose" as desired, (possibly) incorporating some areas where transversal collisions are not actually possible to simplify modeling. For example, conflict regions can be defined as perfect circles, even though there may be some areas in the circle in which transversal collisions are not actually possible. Conflict points and conflict regions do not need to have a one-for-one correspondence: multiple conflict points can be grouped in the same (large) conflict region for simplification purposes. Conflict regions may overlap. It is essential, however, that any region inside the IR in which a transversal collision is possible is part of (at least) one conflict region. If this is the case, guaranteeing that there are no transversal collisions inside the conflict regions is sufficient to guarantee that there are no transversal collisions inside the IR. One simple way of assigning conflict regions is to associate one conflict region to each conflict point, and make the conflict regions sufficiently large circles.

In Fig. 7, this results in nine conflict regions, with $C = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$. Assuming conflict regions as small as the circles depicted, movement (E,S) would pass through conflict regions 9, 8, 4 and 1, thus $C'_{(\mathrm{E,S})} = \{1, 4, 8, 9\}$. Note that conflict regions may be significantly larger than the red circles in the figure.

Let a Control Region (CR) be defined as an area comprised by the IR itself and a portion of each approach up to a suitable control distance D_a, $a \in A$ (light gray areas in Fig. 6). A centralized Intersection Controller (IC) coordinates movements inside the CR by calculating and sending suitable speed profiles to be followed by each vehicle while inside the CR, approaching the IR. Vehicles are assumed to follow these profiles while keeping in their lanes. There is no overtaking inside the CR.

Vehicles on each approach are indexed by $i = 1, 2, \ldots, n_a$ in order of proximity to the IR (i.e., vehicle 1 is the nearest to the intersection, and so forth), with n_a the number of vehicles in approach a inside the CR. Each vehicle inside the CR can be uniquely identified by a pair of indexes (a, i). Throughout this thesis, letters i and j are frequently used as vehicle indexes. When it is not relevant to differentiate between approaches, index a may be omitted for convenience of notation.

Each vehicle i has length $L_i \leq L_{\mathrm{max}}$ and width $W_i \leq W_{\mathrm{max}}$, with L_{max} and W_{max} the maximum length and width, respectively, of a vehicle allowed inside the CR. It is assumed larger vehicles are aware of this restriction and choose different paths in the road network. Conflict regions must be defined in such a way that any possible transversal collision between vehicles traveling on the allowed paths, even among vehicles with the maximum allowed dimensions, occurs inside conflict regions.

It is assumed that vehicles are capable of keeping on their path. Throughout this thesis, when a vehicle position, speed, or acceleration is mentioned, it refers to the vehicle's longitudinal movement along its path.

Vehicle speed v_i and acceleration a_i at any given instant are bounded by maximum speed v_i^{max}; and minimum and maximum accelerations a_i^{min} and a_i^{max}, with $v_i \leq v_i^{\mathrm{max}}$, $a_i^{\mathrm{min}} < 0 < a_i^{\mathrm{max}}$ and $a_i^{\mathrm{min}} \leq a_i \leq a_i^{\mathrm{max}}$. These values, in turn, are bounded by minimum performance criteria, and a speed limit. All vehicles in the network must not travel above the speed limit $v_{\mathrm{L}}^{\mathrm{max}}$, and must also be able to reach this limit[3]. Vehicles must also be capable of accelerating or de-

[3]It is straightforward to generalize OATS to allow vehicles that are unable to

celerating with at least $a_{\text{net}}^{\text{max,min}}$ and $a_{\text{net}}^{\text{min,max}}$, respectively. That is, for every vehicle i it must hold that: $v_i \leq v_{\text{L}}^{\text{max}}$, $a_i^{\text{min}} \leq a_{\text{net}}^{\text{min,max}}$, and $a_i^{\text{max}} \geq a_{\text{net}}^{\text{max,min}}$.

Vehicles can only travel forward and are able to communicate and share information with the IC. The IC has access to the current state information of every vehicle, as well as the following information related to vehicle movements inside the intersection:

- $m_i \in M$, the movement intended by vehicle i;
- S_i, the speed profile vehicle i intends to follow inside the intersection[4].

Vehicle behavior outside the CR is beyond the scope of this research, although in simulations the ACC control law presented in Section 2.4 is used in this situation (SHLADOVER; SU; LU, 2012). The interference of pedestrians or obstacles entering the road arbitrarily is disregarded. Free flow conditions are assumed downstream of all the intersection exits.

Once a vehicle i is inside the CR, the IC is assumed to have access to all relevant information regarding the vehicle: its intended movement m_i, the intended crossing profile S_i, all the intervals related to arrival at conflict regions (such as $tau_{i,c}^{\text{arrive}}$, besides others that will be mentioned in Section 3.6.2), and the vehicle state. Communication is assumed to be perfect (instantaneous and lossless), and vehicles are assumed to follow commands received by the IC and be capable of instantly changing their acceleration.

3.3.1 Control Goals

In order to guarantee the safe and efficient use of the intersection, the following goals are defined:

1. Vehicles must not collide with each other
   (a) No two vehicles entering the IR from different approaches can occupy the same conflict region at the same time.

achieve the speed limit, in case this becomes desirable.

[4] The IC does not actually need all the information from S_i. S_i is used by the IC to calculate additional values, such as the time a vehicle takes to reach or leave each conflict region. It is possible to have vehicles calculating and sending these values directly to the IC instead of sending S_i

(b) Vehicles traveling along the same direction must always keep a safe distance of each other. They may share the use of a conflict region.

2. Vehicles should travel through the CR and leave the IR in the minimum possible time while complying with the problem constraints.

3. All vehicles should be able to eventually leave the intersection in a finite amount of time (no deadlock).

Goals 1a and 1b guarantee the safe operation of the intersection. The first one guarantees there are no transversal collisions (which, by definition, could only happen inside conflict regions), while the second guarantees the absence of longitudinal collisions.

Goal 2 aims at maximizing efficiency in terms of travel time. Note that Goal 3 can not be guaranteed if free flow conditions downstream are not assumed, as a blockage beyond the intersection may render vehicles unable to cross.

Since the speed profiles S_i each vehicle i follows while inside the IR are decided individually for each vehicle (without regard to how this affects other vehicles) and do not change, from the IC's perspective they are fixed. As such, the IC's decisions can affect vehicle movement only while they approach the intersection. Thus, goal 2 becomes equivalent to:

2* Vehicles should travel through the CR and reach the IR in minimum time.

3.4 SUBPROBLEM 1 - DEFINING SPEED PROFILES FOR CROSSING THE INTERSECTION

SP1 is the problem of, according to the characteristics of each vehicle i and the known geometry of the movement m_i it intends to perform inside the intersection, defining a speed profile S_i for the part of its trajectory that is inside the IR.

SP1 is assumed to be solved by each vehicle without interference from the IC (although a different solution approach, with the IC solving SP1 for each vehicle, is also possible). It is expected that the speed profile obtained is feasible (i.e., vehicles are able to safely follow the speed profile) and results in the vehicle spending as little time as needed inside the intersection.

In this thesis, and particularly in simulations, it is assumed that the speed profile S_i is described by a sequence of desired vehicle speeds for each time step starting from the moment a vehicle enters the IR and until it leaves the IR[5].

The exact manner in which these profiles are defined can vary from vehicle to vehicle. For instance, one vehicle may solve an optimal control problem to find a speed profile that crosses in minimum time, while another may follow a rule that states it crosses with a constant speed that is a function of turning radius. In the simulations performed, vehicles solve SP1 by choosing an arbitrary constant speed to cross the intersection, based on the movement performed. Lower speeds are used for turning.

The exact solution method used for SP1 is not relevant for the other subproblems. In order to implement the OATS strategy it is sufficient to ensure that an (arbitrary) speed profile exists. It is assumed vehicles always have positive, non-zero speed while inside the IR.

Once a profile is chosen, it is possible to calculate, for each conflict region on the vehicle's path ($c \in C'_{m_i}$), the time interval between the instant the vehicle enters the intersection and the instant it enters or leaves each conflict region. Calculating these intervals can be done by either the vehicle or the IC (assuming it knows the speed profile). The IC is assumed to know these intervals.

3.5 SUBPROBLEM 2 - OBTAINING THE INTERVAL OF FEASIBLE ARRIVAL TIMES

SP2 is the problem of defining, for each vehicle, a feasible arrival time interval during which it is possible for the vehicle to reach the intersection.

Consider vehicle i in approach a inside the CR, with speed v_i and position d_i (measured as the distance to the end of approach a) at time instant $t_0 = 0$. Considering the following constraints:

- the vehicle must have speed v_i^{in} when it reaches the intersection, corresponding to the speed at the start of S_i given by SP1;
- the bounds in vehicle dynamics as specified in Section 3.3;

it is possible to define an interval $[t_{a,i}^{\min}, t_{a,i}^{\max}]$, with $0 \leq t_{a,i}^{\min} \leq t_{a,i}^{\max}$, as

[5] In general, this does not need to be the case, and S_i could in principle be described in a different manner, such as a function of speed in time. However, more general forms of describing S_i are not explored in this thesis.

the time interval in which it is feasible for vehicle i from approach a to reach the intersection. SP2 is the problem of finding this interval, and can be solved analytically with linear motion equations. Appendix A details how the interval $[t_{a,i}^{\min}, t_{a,i}^{\max}]$ can be calculated.

If the vehicle is sufficiently far from the intersection (or its speed is sufficiently low), $t_{a,i}^{\max}$ is potentially unbounded, as it is possible for the vehicle to come to a complete stop, stay idle for an arbitrary time, and eventually accelerate and reach the intersection while still satisfying all constraints.

The control distance D_a of each approach a must be sufficiently long so that any vehicle just entering the CR has an unbounded $t_i^{\max}$. This is necessary to guarantee that, for a single vehicle just entering the CR, SP3 always has a feasible solution, as the scheduling algorithm can delay the arrival of the vehicle at the intersection as much as needed to avoid collisions with other vehicles.

It is assumed vehicles outside the CR travel sufficiently spaced (with a certain minimum distance or headway) such that, if any vehicle decelerates or stops, vehicles behind are also able to decelerate or come to a full stop without colliding. If a vehicle is controlled by the IC in such a way that it decelerates close enough to the border of the CR to influence vehicles outside the CR, the vehicles not under the influence of the IC should decelerate accordingly to avoid a collision with the controlled vehicles.

Notice that if a vehicle i is controlled by the IC in such a way that it reaches the IR in a time greater than its minimum arrival time $t_{a,i}^{\min}$, vehicle $i+1$ behind it may be unable to reach the intersection in its own minimum arrival time $t_{a,i+1}^{\min}$ due to the presence of the vehicle in front. This is not a problem because, as will be shown in Section 3.6, when a vehicle arrival is scheduled by SP3, its arrival time is constrained to be sufficiently later than the arrival time of the vehicle in front (which must reach the intersection before it). Therefore, in such a case the minimum arrival time of the vehicle behind would not be part of an active constraint anyway. Instead, the vehicle would be constrained by the earliest possible time it could enter the intersection considering the influence of the vehicle in front. As such, there is no need to consider the influence of vehicles in front when solving SP2.

3.6 SUBPROBLEM 3 - SCHEDULING VEHICLE ARRIVALS AT THE INTERSECTION

SP3 is the problem of defining the time instant vehicles arrive at the intersection.

Considering the assumptions made in Section 3.3 and the goals stated in Section 3.3.1, as well as the known feasible arrival times obtained by solving SP2, a scheduling problem is formalized to define the time vehicles should arrive at the intersection. This is done by describing SP3 as a Mixed Integer Linear Program (MILP).

3.6.1 Decision variables and objective

Let $t_{a,i}$ be the time it takes for vehicle i in approach a to travel from its current position to the "entrance" of the IR (i.e., the time until the front bumper of the vehicle touches the nearest border of the IR). The set $T = \{t_{a,i} : i = 1, \ldots, n_a,\ a \in A\}$ corresponds to the main decision variables of SP3. Additional auxiliary decision variables are also introduced in Section 3.6.2 to model the choice of vehicle order. Recall that n_a is the number of vehicles in approach a inside the CR; vehicles are numbered $i = 1, \ldots, n_a$ for each approach $a \in A$, and are ordered by proximity to the intersection. The value of $t_{a,i}$ is bounded by $[t_{a,i}^{\min}, t_{a,i}^{\max}]$ (obtained by solving SP2) as:

$$t_{a,i}^{\min} \leq t_{a,i} \leq t_{a,i}^{\max} \qquad\qquad i = 1, \ldots, n_a; a \in A \tag{3.1}$$

Equation (3.1) ensures that for every vehicle i in each approach a the scheduled time $t_{a,i}$ is a feasible arrival time (assuming no interference with other vehicles), and therefore a feasible speed profile exists such that vehicle i is able to reach the intersection at time $t_{a,i}$.

The efficiency goal – minimizing the time vehicles take to reach the intersection – is formalized by the following objective function:

$$\min \sum_{a \in A} \sum_{i=1}^{n_a} t_{a,i}. \tag{3.2}$$

3.6.2 Safety constraints

Recall goals 1a and 1b from Section 3.3.1. For now, assume that guaranteeing the absence of collisions inside conflict regions is sufficient to guarantee the absence of both transversal and longitudinal collisions in the IR. The topic of longitudinal collisions will be addressed in Section 3.6.2.2.

Let $\tau_{i,c}^{\text{arrive}}$ be the time for vehicle i to arrive at conflict region c after it enters the IR. This is the time elapsed between the instant the front bumper of the vehicle first enters the IR and the instant it first enters the conflict region. Notice that since S_i is already defined (from SP1), $\tau_{i,c}^{\text{arrive}}$ is known. Let $t_{a,i,c}$ be the time it takes for vehicle i in approach a to travel from it's current position until it arrives at conflict reigon c. This is actually merely decision variable $t_{a,i}$ shifted by the constant value $\tau_{i,c}^{\text{arrive}}$:

$$t_{a,i,c} = t_{a,i} + \tau_{i,c}^{\text{arrive}}. \tag{3.3}$$

For any two vehicles i and j from approaches a and b that must cross a given conflict region c, the absence of collisions on this region can be modelled by a generic pair of disjunctive constraints with the form:

$$t_{a,i,c} + h_{a,i,b,j,c} \leq t_{b,j,c} \tag{3.4a}$$

$$\vee$$

$$t_{b,j,c} + h_{b,j,a,i,c} \leq t_{a,i,c} \tag{3.4b}$$

with $t_{a,i,c}$ and $t_{b,j,c}$ the times for the corresponding vehicles to reach conflict region c and $h_{a,i,b,j,c}$ a sufficiently large headway between the arrivals of vehicles i and j so as to guarantee safety at conflict region c. The disjunction models the fact that safety can (possibly) be satisfied with either arrival order. In cases where only one order is possible, only one of the constraints is needed and (3.4) can be simplified to either just (3.4a) or (3.4b).

In practice, this disjunction is modeled by introducing auxiliary binary variables $\beta_{a,i,b,j,c}$. Each of these variables describes the relative order between a pair of vehicles in a conflict region, modeling the choice of passing order. If vehicle i from approach a reaches conflict region c before vehicle j from approach b, then $\beta_{a,i,b,j,c} = 0$. Otherwise $\beta_{a,i,b,j,c} = 1$. With this and a sufficiently large constant Q, (3.4) can be rewritten as:

$$t_{a,i,c} + h_{a,i,b,j,c} \leq t_{b,j,c} + Q \cdot \beta_{a,i,b,j,c} \tag{3.5a}$$
$$t_{b,j,c} + h_{b,j,a,i,c} \leq t_{a,i,c} + Q(1 - \beta_{a,i,b,j,c}) \tag{3.5b}$$
$$\beta_{a,i,b,j} \in \{0, 1\} \tag{3.5c}$$
$$a \in A;\ b \in A;\ c \in C'_{m_i} \cap C'_{m_j}$$
$$i = 1, \ldots, n_a;\ j = 1, \ldots, n_b.$$

Notice that

$$\beta_{a,i,b,j,c} = 0 \Leftrightarrow \beta_{b,j,a,i,c} = 1. \tag{3.6}$$

Formally, the optimization problem solved by SP3 takes the form of Equation 3.2, subject to 3.1 and 3.4. Note that 3.4 can be modeled as 3.5, or simplify to either 3.4a or 3.4b depending on the case. Section 3.6.2.1 details the cases where Equation 3.5 can be simplified in such a way, and Section 3.6.2.3 discusses how headways $h_{a,i,b,j,c}$ are defined.

3.6.2.1 Special cases of arrival order

While the constraints given by (3.5) are general enough to cover any possible situation of two vehicles arriving at the intersection, in many cases only one relative order is possible and it is not actually necessary to model choice. Consider the following cases:

1. If $a = b$, i.e., vehicles come from the same approach, then the arrival order is already known, since there is no overtaking. This order is the same on all conflict regions both vehicles will pass through. This can be formalized as:

$$\beta_{a,i,a,j,c} = 0, \forall\ i < j, \tag{3.7a}$$
$$\beta_{a,i,a,j,c} = 1, \forall\ i > j. \tag{3.7b}$$

2. Depending on the intervals $[t_{a,i}^{\min}, t_{a,i}^{\max}]$ and $[t_{b,j}^{\max}, t_{b,j}^{\max}]$ and needed safety headway $h_{a,i,b,j,c}$, there may be only one feasible arrival order. If the latest time a vehicle i can arrive, plus the minimum headway, is earlier than the earliest time another vehicle j can possibly arrive, then the only possible order is for vehicle i to go through conflict region c before vehicle j. Formally:

$$t_{a,i}^{\max} + h_{a,i,b,j,c} \leq t_{b,j}^{\min} \Rightarrow \beta_{a,i,b,j,c} = 0. \tag{3.8}$$

3. If a vehicle i in approach a reaches conflict region c before vehicle j in approach b (i.e., $\beta_{a,i,b,j,c} = 0$), any vehicle $j + p$, with $p > 0$ that reaches the conflict region after vehicle j must also do so after vehicle i (i.e., $\beta_{a,i,b,j+p,c} = 0$). Formally:

$$\beta_{a,i,b,j,c} \leq \beta_{a,i+p,b,j,c}, \tag{3.9a}$$
$$\beta_{a,i,b,j,c} \geq \beta_{a,i,b,j+p,c}, \tag{3.9b}$$
$$a \in A;\ b \in A;$$
$$i = 1, \ldots, n_a;\ j = 1, \ldots, n_b;$$
$$k > 0.$$

This is equivalent to the following implications::

$$\beta_{a,i,b,j,c} = 1 \Rightarrow \beta_{a,i+p,b,j,c} = 1, \tag{3.10a}$$
$$\beta_{a,i,b,j,c} = 0 \Rightarrow \beta_{a,i-p,b,j,c} = 0, \tag{3.10b}$$
$$\beta_{a,i,b,j,c} = 0 \Rightarrow \beta_{a,i,b,j+p,c} = 0, \tag{3.10c}$$
$$\beta_{a,i,b,j,c} = 1 \Rightarrow \beta_{a,i,b,j-p,c} = 1, \tag{3.10d}$$
$$a \in A;\ b \in A;$$
$$i = 1, \ldots, n_a;\ j = 1, \ldots, n_b;$$
$$p > 0.$$

Let be n_a^c the number of vehicles on approach a that will cross conflict region c. Notice that all binary variables $\beta_{a,i,b,j,c}$ from a given approach pair a, b and conflict region c can be organized in a n_a^c x n_b^c matrix (or a n_a x n_b matrix with lines and columns matching indexes i and j, respectively, in case every vehicle in each approach goes through conflict region c). In such a matrix, all the ones would be grouped in the bottom left corner, while the zeros would be in the top right corner. In fact, knowing only the position of the "first" zero of each row (or first one in each column) is sufficient to fully describe de matrix. This means that, even though the matrix of all relative arrival orders among vehicles in a pair of approaches a and b crossing a given conflict region has $n_a^c \cdot n_b^c$ values, knowing only the values of $\min(n_a^c, n_b^c)$ specific variables is sufficient to fully describe the entire matrix. Hence, even though the number of binary variables grows proportionately to the square of the number of vehicles for a given pair of approaches (for each conflict region), the minimum amount of information needed to fully describe the arrival orders grows linearly to the number of vehicles. The number of possible combinations of relative arrival orders, however, grows exponentially to the number of "relevant" binary variables.

4. If a given vehicle i reaches a conflict region c before a vehicle j ($\beta_{a_1,i,a_2,j,c} = 0$) and after a vehicle q ($\beta_{a_1,i,a_3,q,c} = 1$), then vehicle j must also reach conflict region c after vehicle q. Formally[6]:

$$\beta_{a_1,i,a_2,j,c} = 0 \;\wedge\; \beta_{a_1,i,a_3,q,c} = 1 \Rightarrow \beta_{a_2,j,a_3,q,c} = 1. \quad (3.11)$$

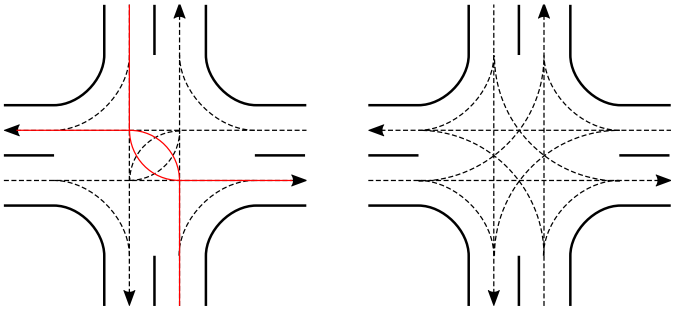
Because of the cases formalized by (3.7) – (3.11), frequently there is only one possible arrival order for a pair of vehicles. In such cases, there is no need for using binary variables to model the choice of passing order, and (3.5) can be simplified to either (3.4a) or (3.4b).

Notice that having more than one binary variable for a pair of vehicles is only necessary in case their paths cross in such a way that they have more than one conflict region in common and it is possible for them to arrive in a different order at those regions. A example of this type of situation is a pair of opposite narrow left turns, which cross each other twice, as the ones in the layout depicted in Figure 8(a). In a layout with wider turns, such as the one in 8(b), this does not happen. In cases where the paths merge, the same binary variable can be used for all associated constraints (i.e., $\beta_{a,i,b,j,c_1} = \beta_{a,i,b,j,c_2}\ \forall\ c_1, c_2 \in C$), as the relative vehicle order on all conflict regions along their shared path must be the same (this is also true for cases in which the paths diverge or are exactly the same, but as this implies vehicles coming from the same approach, this case is also covered by (3.7)). In most cases there will be (at most) one conflict region in common and hence one binary variable for each pair of vehicles with different origins and intersecting paths. If possible, intersection layouts with paths that cross each other more than once should be avoided, as this increases the number of binary variables needed in SP3.

The two first special cases of arrival order considered, formalized (3.7) and (3.8), are not actually entered as constraints of the optimization model when solving SP3, but are instead used when building the model to find out which constraints need to be inserted into it. All the relevant information for checking these two cases is already available before SP3 starts to be solved. The other two special cases, formalized by 3.9 and 3.11, are used in the same way, but could also be inserted as constraints, since they involve information on the passing order of

[6]Notice approaches a_1, a_2 and a_3 are not necessarily three different approaches.

Figure 8 – Intersection layouts (a) with narrow left turns that cross oposing left turns twice, such as the pair of movements highlighted, and (b) without movements that cross each other twice.



(a) (b)

other vehicles that is not necessarily available a priori. However, these two last cases were found, in simulation, to not significantly impact the time taken to solve a problem instance when entered as constraints, and were not used in this manner.

3.6.2.2 Collisions outside conflict regions

So far, only the possibility of collisions inside conflict regions has been addressed in SP3, even though longitudinal collisions are possible in the IR even outside these regions. However, if the headways $h_{a,i,b,j,c}$ are sufficiently large, it is possible to also guarantee there will be no longitudinal collisions inside the IR.

For a transversal collision to be possible, it is necessary that a pair of vehicles have part of their path in common. This happens if they have the same path, or their paths merge or diverge at some point. As discussed previously in Section 3.6.2, this implies both vehicles cross each of the conflict regions that are part of their shared path in the same order.

Suppose vehicle i is the first to cross the shared path, followed by vehicle j (i.e., $\beta_{a,i,b,j,c} = 0 \; \forall \; c \in C'_{m_i} \cap C'_{m_j}$). This means that, for any given conflict region $c \in C'_{m_i} \cap C'_{m_j}$, vehicle j can enter it only a time $h_{a,i,b,j,c}$ after vehicle i has arrived at the same region.

Headway $h_{a,i,b,j,c}$ can be chosen to be sufficiently large to guarantee vehicles do not suffer a longitudinal collision before reaching the next conflict region on their paths. In case the next conflict region is not the same for both vehicles (i.e., their paths diverged), they are no longer at risk of suffering a longitudinal collision, and do not need to be constrained by each other in further conflict regions. On the other hand, if the next conflict region is the same for both vehicles, the headway constraint associated with the next conflict region should ensure safety for the next segment of their path, and so forth.

Consider an intersection in which every point where vehicles can enter or exit the IR is covered by a conflict region (i.e., in the very first and very last moment a vehicle is inside the IR, it is also inside a conflict region). In this case, there is no part of the trajectory of a vehicle inside the IR where its longitudinal safety is not guaranteed by a constraint associated to a conflict region.

Hence, if:

- headways $h_{a,i,b,j,c}$ are sufficiently large, and
- every point where vehicles can enter or exit the IR is part of a conflict region,

then the headway constraints between each pair of vehicles in each conflict region they share is sufficient to guarantee there will be no longitudinal collisions.

3.6.2.3 Defining the minimum headway

Consider two vehicles traveling along the same path. If the vehicle in front has a lower speed than the one behind, there is a risk of a longitudinal collision. However, even in case of a collision, the risks and damages are relatively low if the speed difference between the vehicles is small, which is often the case for vehicles following the same path.

For vehicles following different paths that intersect, there is a risk of a transversal collision. In this type of situation it is common for the relative speed difference of the vehicles to be larger, since they move through different directions. In this case, an eventual collision may pose a much larger risk than a typical longitudinal collision. In short, side crashes tend to be more dangerous than rear end crashes.

In light of this, the minimum headway constraint between vehicles is defined in a way that is more restrictive when considering vehicles that have a potential transversal interaction.

Since it is sufficient to ensure vehicles keep a (sufficiently large) minimum headway from the vehicle in front to avoid longitudinal collision, vehicles traveling on the same direction are allowed to share a conflict region as long as they keep a safe headway. There is no need to "block" a vehicle from entering the conflict region if there is no risk of a transversal collision. On the other hand, vehicles traveling on different directions are never allowed to share the same conflict region at any moment.

More specifically, the two following safety rules are imposed for collision avoidance:

i) The time headway between two vehicles traveling in the same direction along any conflict region must be at least as large as a safety longitudinal headway h_{L};

ii) If two vehicles traveling on different directions make use of the same conflict region, there must be at least a safety time h_{T} between the instant one vehicle clears the region and the instant the other one is allowed to enter.

Given that each vehicle has information regarding its own geometry, its trajectory inside the IR, and the geometry of the conflict regions, it is possible for each vehicle i to define, for each conflict region c intercepted by its path m_i, the three following time intervals:

- $\tau_{i,c}^{\text{arrive}}$, the time to arrive at the conflict region, as already defined in section 3.6.2. Recall that this is the time elapsed between the instant the front bumper of the vehicle first enters the IR and the instant it first enters the conflict region.

- $\tau_{i,c}^{\text{inside}}$, the time to be inside the conflict region. This is the time elapsed between the instant the front bumper of the vehicle first enters the IR and the earliest moment no part of the vehicle has yet to enter the conflict region, i.e., the rear end of the vehicle has just entered the conflict region.

- $\tau_{i,c}^{\text{out}}$, the time to be completely out of the conflict region. This is the time elapsed between the instant the front bumper of the vehicle first enters the IR and the earliest moment the vehicle has completely left the conflict region (after it has already been inside it), i.e., the rear of the vehicle has just left the conflict region.

In this context, entering the IR means reaching the closest border of the IR, or the being at closest point on the vehicle path that is inside the IR. These intervals are communicated to the IR and used to formalize the headways $h_{a,i,b,j,c}$ used in (3.4) and (3.5). Let $h^{\text{safe}}_{a,i,b,j,c}$ be the minimum safety headway between vehicles i and j in conflict region c.

- If vehicles i and j have a potential longitudinal conflict on conflict region c, the time $h_{a,i,b,j,c}$ before j can safely enter after i has already entered is given by

$$h_{a,i,b,j,c} = h^{\text{safe}}_{a,i,b,j,c} + \tau^{\text{inside}}_{i,c} - \tau^{\text{arrive}}_{i,c}. \tag{3.12}$$

- If vehicles i and j have a potential transversal conflict on conflict region c, the time $h_{a,i,b,j,c}$ before j can safely enter after i has already entered is given by

$$h_{a,i,b,j,c} = h^{\text{safe}}_{a,i,b,j,c} + \tau^{\text{out}}_{i,c} - \tau^{\text{arrive}}_{i,c}. \tag{3.13}$$

In short, for vehicles at risk of a longitudinal collision, the second vehicle is allowed to enter the conflict region after the first vehicle has completely entered the conflict region and a safety headway has elapsed. For vehicles at risk of a transversal collision, the second vehicle can enter the conflict region only after the first one has completely left the region and a safety headway has elapsed.

Note that it follows from (3.13) that (unless a conflict region is sufficiently small), vehicles will not suffer transversal collisions even if $h^{\text{safe}}_{a,i,b,j,c} = 0$. The headway $h^{\text{safe}}_{a,i,b,j,c}$ is designed as an extra time that exists for safety reasons, sufficiently large to account for any uncertainties involved.

Since $\tau^{\text{inside}}_{i,c} < \tau^{\text{out}}_{i,c}$, generally (3.13) is more restrictive than (3.12) (unless $h^{\text{safe}}_{a,i,b,j,c}$ for a transversal collision is substantially larger than for a longitudinal collision, which is not the case by design). This means that if one had to choose to be conservative and model all potential conflicts as if they were transversal conflicts, this would not compromise safety, although it would be less efficient.

The safe headways $h^{\text{safe}}_{a,i,b,j,c}$ are obtained by a procedure illustrated in Algorithm 1. Notice $h^{\text{safe}}_{a,i,b,j,c}$ and $h^{\text{safe}}_{b,j,a,i,c}$ (and hence $h_{a,i,b,j,c}$ and $h_{b,j,a,i,c}$) are not necessarily the same. Unless special circumstances apply, $h^{\text{safe}}_{a,i,b,j,c}$ is generally the same as h_{L} for longitudinal conflicts and h_{T} for transversal conflicts.

Algorithm 1 implements the following procedure:

- Initially, the safety headway is set as either a default longitudinal or transversal headway, according to the conflict type. These

Algorithm 1 Procedure to obtain the minimum safe headway $h^{\text{safe}}_{a,i,b,j,c}$ between vehicles i and j at conflict region c.

procedure GetHeadway
  if conflict between (a, i) and (b, j) is longitudinal **then**
    $h^{\text{safe}}_{a,i,b,j,c} \leftarrow h_{\text{L}}$
  else
    $h^{\text{safe}}_{a,i,b,j,c} \leftarrow h_{\text{T}}$
  $h^{\text{safe}}_{a,i,b,j,c} \leftarrow h^{\text{safe}}_{a,i,b,j,c} + h^{\Delta\text{v}}_{a,i,b,j,c}$
  $h^{\text{safe}}_{a,i,b,j,c} \leftarrow \max(h^{\text{safe}}_{a,i,b,j,c}, h^{\text{cap}}_{c})$

should be large enough to account for any uncertainties involved and guarantee safety by themselves on a typical situation.

- The safety headway is then increased by the speed difference headway, $h^{\Delta\text{v}}_{a,i,b,j,c}$. This is an additional safety margin needed in case vehicle j has higher speed than vehicle i in front. This value should be sufficiently large so the front vehicle can either reach the next conflict region sufficiently before the vehicle behind (in case this is not the last conflict region on their path) or accelerate to the speed limit (in case this is the last conflict region on their path as they are about to leave the IR), and can be calculated according to the speed difference between the vehicles, or the time they take to reach the next conflict region. In this thesis, when calculating the speed difference headway a simplifying and conservative approach is adopted by assuming, for each vehicle, that there always is a vehicle behind it with maximum speed wanting to access the conflict region as soon as possible. This leads to higher headways after slow vehicles (in the order of 0.2 s for the simulated scenarios), even if the following vehicle is also slow.

- Lastly, in case $h^{\text{safe}}_{a,i,b,j,c}$ is smaller than a given capacity headway h^{cap}_{c}, it is set to be the same as the capacity headway. This value can be used to limit throughput of a conflict region located in one exit of the intersection in case it is desirable to control intersection outflow. This can be useful if congestion is detected downstream, or for coordination with other intersections. If this type of control action is not desired, then $h^{\text{cap}}_{c} = 0$.

Recapitulating the most important constratins, formally the optimization problem solved by SP3 takes the form of (3.2):

$$\min \sum_{a \in A} \sum_{i=1}^{n_a} t_{a,i} \tag{3.2}$$

subject to (3.1):

$$t_{a,i}^{\min} \leq t_{a,i} \leq t_{a,i}^{\max} \qquad i = 1, \ldots, n_a; a \in A \tag{3.1}$$

and (3.4):

$$t_{a,i,c} + h_{a,i,b,j,c} \leq t_{b,j,c} \tag{3.4a}$$
$$\vee$$
$$t_{b,j,c} + h_{b,j,a,i,c} \leq t_{a,i,c} \tag{3.4b}$$

Recall that (3.4) can either be modeled as (3.5):

$$t_{a,i,c} + h_{a,i,b,j,c} \leq t_{b,j,c} + Q \cdot \beta_{a,i,b,j,c} \tag{3.5a}$$
$$t_{b,j,c} + h_{b,j,a,i,c} \leq t_{a,i,c} + Q(1 - \beta_{a,i,b,j,c}) \tag{3.5b}$$
$$\beta_{a,i,b,j} \in \{0, 1\} \tag{3.5c}$$
$$a \in A; \; b \in A; \; c \in C'_{m_i} \cap C'_{m_j}$$
$$i = 1, \ldots, n_a; \; j = 1, \ldots, n_b.$$

or simplify to (3.4a) or (3.4b), depending on the case.

3.6.3 Problem size and simplifications

With the exception of the disjunctive constraints, all other constraints in SP3 are linear. The disjunctions, which model the choice of possible arrival order, make the problem non-convex. SP3 is actually very similar to job shop scheduling problems, which are NP-hard (GRAHAM et al., 1979; BLAZEWICZ; LENSTRA; KAN, 1983).

In fact, borrowing the terminology commonly used in job scheduling, if one views vehicles as jobs; conflict regions as machines; and the sequence of conflict regions visited by each vehicle as the operations of that job, SP3 can be viewed as an extension of the problem of job-shop scheduling with no-wait constraints (MASCIS; PACCIARELLI, 2002; LENNARTZ, 2006), in which: (i) there is an additional set of constraints (3.1) limiting the earliest and latest possible time for starting the first operation; and (ii) the processing time of an operation (which is analogous to the minimum headway) depends also on the next operation. That is, in the special case of SP3 in which $t_{a,i}^{\min} = 0 \;\forall\; a, i$; $t_{a,i}^{\max}$ is always unbounded (or sufficiently high for the constraint to never be active); and $h_{a,i,b_1,j_1,c} = h_{a,i,b_2,j_2,c} \;\forall\; b_1, b_2, j_1, j_2$; SP3 becomes equivalent to the no-wait job-shop scheduling problem. That means that the no-wait job-shop scheduling problem can be reduced to SP3. Since that problem is NP-hard[7], SP3 is also NP-hard.

In the worst case, the number of disjunctive constraints (and binary variables) associated to each conflict region can be in the order of the square of the number of vehicles that go through that conflict region. Or, more precisely, the number of binary variables when n vehicles are involved at a conflict region can be up to $\frac{n^2-n}{2}$. However, given the special cases of arrival order discussed in Section 3.6.2.1, the number of disjunctive constraints can be significantly lower in practice. It was observed in simulation that, for typical problem instances with 15 vehicles inside the CR (corresponding to the "high demand" scenarios discussed in Chapter 4), the average number of binary variables (and hence disjunctive constraints) involved is 25, with a standard deviation of 3.5 constraints. There are some significant outliers though, with some problem instances occasionally having over 120 binary variables.

The Gurobi solver (Gurobi Optimization, 2016) was used to solve SP3 in the simulations performed. It implements mainly a branch and cut algorithm (besides a pre-solving algorithm and cutting planes), meaning finding an optimal solution typically involves solving many linear relaxations of the original problem.

Problem complexity can vary significantly for different instances of the problem. The number of possible linear problems on the search tree can be up to 2 to the power of the number of binary variables. However, thanks to the branch and cut approach and the heuristics and cutting planes the Gurobi solver uses to reduce the search tree, usually the number of problems that have to be solved is much smaller[8]. As an example, for an atypically large problem, with 18 vehicles and 77 integer variables, the Gurobi solver explored 9870 nodes in the search tree. This particular case was actually the hardest problem instance encountered for a certain scenario, consisting of "high" traffic demand.

Even with these considerations, SP3 can become prohibitively expensive to solve if the number of potential conflicts (and hence bi-

[7]The no-wait job-shop scheduling problem is known to be NP-hard, except for a few special cases such as the single machine case (LENNARTZ, 2006; MASCIS; PACCIARELLI, 2002).

[8]The implications formalized by (3.9) and (3.11) on page 75 were found to have very little to no impact on this when used as constraints.

nary variables) is very large. This motivated the study of the effects of different control configurations on problem size and solution quality. Disregarding some vehicles to consider only a smaller problem has the effect of lowering both the computational complexity and solution quality (since not all available information is considered, a suboptimal solution is found). Chapter 4 discusses the results of simulations performed to investigate the effect of these simplifications.

The following strategies were studied to simplify SP3:

1. Reducing the size of the control region.
2. Optimizing vehicles in batches.
3. Setting a *distance of no re-scheduling* for deciding vehicle trajectory.
4. Approximating the arrival times of vehicles with a very narrow arrival interval.
5. Limiting the number of vehicles with unbounded maximum arrival time in a problem instance.

The size of the CR is one of the main factors related to problem complexity. The larger the control distance D_a, the larger the average number of vehicles inside the CR, and (usually) the larger the number on potential conflicts. Besides leading to potentially hard problems, a large D_a can cause modeling difficulties, as it may become impossible to disregard other intersections and conflict points outside the IR when modeling the behavior of vehicles very far from the intersection. It is interesting to notice that the layout of the CR can also have a significant impact on the average number of potential conflicts, impacting problem complexity. However, the investigation of different intersection layouts is left for future work.

Optimizing vehicles in batches (Simplification 2) is to, instead of solving SP3 each time a new vehicle enters the CR, wait for some condition to trigger and solve SP3 less frequently.

Setting a *distance of no re-scheduling* (D_{NR}) regarding vehicle trajectory (Simplification 3) can also significantly reduce the problem size. Consider that, if SP3 is solved repeatedly while vehicles travel inside de CR, one can choose to either (i) allow vehicles that were in the previous instance of the problem to update their scheduled time when a new solution is found; or (ii) force vehicles to continue using the previous solution found. A middle ground approach is to set a *distance of no re-scheduling*, and consider that once a vehicle passes this point, its speed profile can no longer change.

Simplification 4 consists in setting the arrival time of vehicles with a very narrow arrival interval as a fixed value instead of treating it as a decision variable. Consider a minimum size for the arrival interval defined by ξ^{SP2}. If the feasible arrival interval $[t_{a,i}^{\min}, t_{a,i}^{\max}]$ of vehicle i is smaller than this value, i.e.:

$$t_{a,i}^{\max} - t_{a,i}^{\min} \leq \xi^{t,\text{SP2}}, \tag{3.14}$$

then, instead of defining $t_{a,i}$ as a decision variable, it could be approximated to

$$t_{a,i} = \frac{t_{a,i}^{\max} + t_{a,i}^{\min}}{2} \tag{3.15}$$

The reasoning for this simplification is that if the interval $[t_{a,i}^{\min}, t_{a,i}^{\max}]$ is so narrow that the difference between $t_{a,i}^{\max}$ and $t_{a,i}^{\min}$ is close to the uncertainties involved, it is of little use to spend computational power choosing an exact instant inside this interval, as it is not much better than just setting a close enough value. Having narrow arrival intervals is a common occurrence among vehicles very close to the IR, since the interval narrows as vehicles approach the IR.

Finally, Simplification 5 consists in limiting the number of vehicles with an unbounded maximum arrival time to be considered in a given problem instance for each approach. This is motivated by the observation that while for some vehicles the feasible arrival time interval $[t_{a,i}^{\min}, t_{a,i}^{\max}]$ is reasonably short, for other vehicles it can be very large, or even unbounded. When these intervals are short for most vehicles, the vast majority of relative orders of arrival are not possible, which can be detected easily and significantly reduces problem size. When the feasible arrival intervals are large, however, fewer such simplifications can be made. In the case where $t_{a,i}^{\max} \to \infty$ (or an arbitrarily large maximum time) for a significant number of vehicles, almost any ordering is possible, which can lead to a huge number of feasible combinations for the binary variables. One possible approach to deal with this is to ignore the N_{U} last vehicles at the end of each approach that have an unbounded $t_{a,i}^{\max}$ (or at least a $t_{a,i}^{\max}$ larger than a defined threshold), and let then be handled only by future problem instances.

3.6.4 Vehicles inside the IR

Even though vehicles follow fixed speed profiles while inside the IR, which are beyond the IC control, it is still important to take their presence into account, as the movement of vehicles that have yet to enter the intersection can be constrained by the presence of vehicles still inside the IR. This is done by considering vehicles inside the IR in the problem constraints, but not assigning decision variables regarding their arrival time. Such vehicles have negative values for parameters such as $t_{a,i}$, signifying they have already entered the IR when the problem is being solved. In fact, this is no different than effectively setting the border of the intersection as a point of no re-scheduling (i.e., $D_{\mathrm{NR}} = 0$).

3.6.5 Liveness and priority

Recall the control goals specified in Section 3.3.1. The problem constraints of SP3 guarantee the safety goals for vehicles inside the IR, and the objective function guarantees efficiency. Lastly, it remains to show that the resulting behavior is deadlock free.

It is reasonable to assume free-flow conditions downstream, meaning it is always possible for a vehicle to leave the IR. Otherwise, if the intersection exits are blocked downstream, it is impossible to guarantee there will not be a deadlock, since vehicles may have nowhere to go. This is true for any intersection management strategy, not only OATS. If there is some level of congestion that limits the capacity of the intersection exits of receiving vehicles, the capacity headways h_c^{cap} can be used to limit the outflow through congested exits. They can be set as large as needed in order to limit the outflow of any exit to be as small as necessary to avoid queues spilling back at the intersection. In the extreme, a on/off control could be used, setting $h_c^{\mathrm{cap}} = \infty$ if heavy congestion is detected in an exit, guaranteeing no vehicles can leave the intersection while the congestion lasts, so the queue will not spill back and block the IR. Even if this blocks the intersection temporarily, if the congestion downstream eventually dissolves, the intersection should "unlock" once the capacity headway is restored to a low value. Basically, the capacity headway offers a mechanism to limit the outflow or even "lock" the intersection temporarily so queues do not spill to the IR. However, if the congestion downstream lasts indefinitely, it is not possible to guarantee that there will not be a deadlock.

Now that it was shown the intersection can eventually recover from a temporary congestion downstream by limiting (or blocking) its outflow, consider the case when there are always free-flow conditions downstream. Since the speed profiles vehicles follow when inside the IR are fixed, and with positive speed at all points, any vehicle inside the IR will eventually leave. This means that if vehicles cease entering the intersection for some time, the IR will eventually become empty.

Since each vehicle i must reach the IR with speed v_i^{in} to fulfill its desired speed profile for crossing the intersection, there is a "point of no return" (PNR) after which vehicles are no longer able to come to a complete stop and accelerate to v_i^{in} before the IR. Since the formulation of SP3 doesn't allow vehicles to reach the undesired situation of having a insufficiently low speed after the PNR (such a solution would be unfeasible, as it implies $t_{a,i} \geq t_{a,i}^{\max}$), it follows that any vehicle that is beyond the PNR must reach the intersection and cross it in a finite amount of time. Hence, no vehicle beyond the PNR will ever stop, or block the intersection.

On the other hand, it is always feasible for a vehicle that is located before the PNR (in any approach) to stop. This means that, in any possible configuration of the system, there exists a feasible solution in which every vehicle before the PNR comes to a complete stop (although this is certainly not an optimal solution). If these vehicles remain stopped for a sufficiently long time, every vehicle that has already crossed the PNR eventually leaves the CR, so only vehicles before the PNR remain in the system. This means that a state in which there are no vehicles located after their PNR is always reachable in a finite amount of time. Such state is deadlock free, because there exists a solution in which the first stopped vehicle in any approach accelerates and crosses the intersection, as there are no other vehicles imposing constraints that impede his movement.

Since this state is always reachable, and is deadlock free, the whole system is deadlock free.

More than deadlock free, it would be desirable if the system had liveness, in the sense that every vehicle entering the CR eventually leaves it. It is straightforward that, since SP3 minimizes the total arrival time, and the system is deadlock free, the resulting behavior is of constant "progress", in that vehicles continuously move forward and cross the intersection. As a matter of fact, for any given problem instance of SP3, every vehicle is scheduled to leave the intersection in a finite amount of time, implying liveness for a single problem instance.

However, as will be discussed in Section 3.8, since vehicles continuously enter the CR, SP3 is solved repeatedly, each time considering the current state of the CR, and possibly rescheduling vehicles that were previously scheduled. In such an implementation, it is possible that a vehicle is repeatedly scheduled to a later time, possibly (albeit unlikely) indefinitely. This can be avoided by changing SP3 by incorporating some sort of "memory" of how long a vehicle has already waited, giving a larger weight to vehicles that have been inside the CR for a longer time.

One possible implementation of this is the following alternative objective function for SP3

$$\min \sum_{a \in A} \sum_{i=1}^{n_a} t_{a,i} \cdot w_{a,i}, \tag{3.16}$$

with $w_{a,i}$ a weight factor attributed to vehicle i in approach a. This factor can be anything that increases strictly with the time a vehicle "waits", where "waiting" can be defined arbitrarily as the time it has been inside the CR, the time stopped, the time bellow a given speed, etc. Instead if increasing strictly, $w_{a,i}$ could also be defined as to only increase after a certain amount of time, so vehicles that are only slightly delayed do not influence the solution. For instance, one possible implementation is to set $w_{a,i} = 1$ for vehicles that have been inside the CR for less than one minute, and have it be equal to the time inside the CR for vehicles who have waited longer than that. In any case, $w_{a,i}$ should monotonically increase with the waiting time, and strictly increase after some point.

As long as $w_{a,i}$ increases as a vehicle is repeatedly re-scheduled, (3.16) should guarantee that eventually any such vehicle sufficiently out-weights the others that are competing with it and is granted passage, ensuring the system has liveness.

Besides liveness, a weight coefficient can also be used to indirectly consider vehicles beyond the CR. Consider another alternative objective function for SP3:

$$\min \sum_{a \in A} \sum_{i=1}^{n_a} t_{a,i} \cdot w_{a,i} \cdot w_a, \tag{3.17}$$

with w_a a weight coefficient attributed to every vehicle in approach a. Coefficient w_a can be defined as a measure of the vehicles upstream the CR in approach a, such as a queue length or total number of vehicles in the approach. As such, (3.17) can be used to prioritize an approach that is congested beyond the CR.

Another possible form to guarantee liveness is to modify SP2 to decrease the $t^{\max}_{a,i}$ of vehicles that were already previously scheduled, making $t^{\max}_{a,i}$ arbitrarily close to the previous scheduled time $t_{a,i}$ after each reschedule. This way, as a vehicle is repeatedly delayed the margin by which it can be delayed in the future decreases.

The alternative formulations of the objective of SP3 given by (3.16) and (3.17) can also be used to give priority to specific vehicles, such mass transit, emergency or high occupancy vehicles, by significantly increasing their weight.

3.6.6 Pedestrian crossings

The presence of pedestrians or other agents that interact arbitrarily with traffic is disregard in the formulation of OATS. However, it is straightforward to implement a mechanism to stop vehicles from occupying a conflict region for an arbitrarily long time. This is of special interest for conflict regions at the border of the IR (where the approaches end), where pedestrian crossings are usually located, and can be used to replicate the behavior of signalized crosswalks where signalization periodically stops traffic to grant pedestrian passage, or pedestrians request access by pressing a button.

Although SP3's formulation could be extended to explicitly model the possibility of "closing" a conflict region, the same behavior can be achieved by simply including a "virtual vehicle" when needed.

Consider a virtual vehicle $\hat{i}$ that has $t^{\min}_{a,\hat{i}} = t^{\max}_{a,\hat{i}} = t_{\text{lock}}$, meaning it must reach the IR exactly at t_{lock}. Such vehicle is assigned to a special virtual movement $\hat{m}$ that is composed by a set of conflict regions which are to be "locked", $\hat{C} \in C$. Arrival intervals of vehicle $\hat{i}$ are set for each conflict region $c \in \hat{C}$ such that $\tau^{\text{arrive}}_{\hat{i},c} = 0$ and $\tau^{\text{out}}_{\hat{i},c} = t_{\text{free}}$.

Finally, consider as if virtual vehicle $\hat{i}$ has a potential transversal conflict at each conflict region $c \in \hat{C}$ with all vehicles that also go through c. The resulting behavior of adding such a vehicle to a given problem instance solved by SP3 is that every control region $c \in \hat{C}$ will effectively be locked by the virtual vehicle during time interval $[t_{\text{lock}}, t_{\text{free}}]$. No other (real) vehicle will be granted access to these regions during this interval, since they have transversal conflicts with the virtual vehicle. This can be used to implement a period for pedestrian crossing in regions $c \in \hat{C}$.

It is important to chose a sufficiently large value t_{lock} to guarantee the problem is feasible. A simple approach is to check the previous solution of SP3 and set t_{lock} slightly larger than the last moment a vehicle is scheduled to occupy a conflict region $c \in \hat{C}$:

$$t_{\text{lock}} > t_{a,i} + \tau_{i,c}^{\text{out}}, \tag{3.18}$$

$$a \in A;\ i = 1, \ldots, n_a; \quad c \in \hat{C} \cap C_{m_i}. \tag{3.19}$$

Notice that when vehicles are unable to gain access to conflict regions $c \in \hat{C}$ for a long period of time, they stop a certain distance before the intersection (and before the PNR).

Another important consideration for pedestrian safety is traffic speed. Intersections are often busy places for pedestrian traffic, just as they are for vehicles. OATS formulation does not account for the presence of pedestrians that cross without waiting for a proper blocking of a conflict region. Emergency action to avoid collisions in these situations is a very relevant topic of study, but beyond the scope of this thesis.

However, a simple but effective design choice can be taken to mitigate the risks associated with traffic accidents: adopting a low speed limit.

It is well known that the severity of the consequences of crashes, specially for pedestrians, cyclists and motorcyclists rises with traffic speed. The risk of a pedestrian fatality after an impact with a vehicle is approximately 9% at a speed of 30 mph (close to 48 km/h), and approximately 50% at 40 mph (close to 64 km/h) (RICHARDS, 2010). The probability of a fatal crash is related to the fourth power of the speed (PEDEN et al., 2004). A report by the World Health Organization (2015) recommends that when motorized traffic mixes with pedestrians and cyclists, the speed limit should be under 30 km/h.

In light of this, a speed limit of 30 km/h is proposed for OATS, and used throughout this thesis.

3.6.7 A simplified capacity analysis

The theoretical capacity flow given by OATS depends on the problem constraints, intersection layout and even on the composition of traffic demand (in the sense of origins and destinations), which can be uncertain.

If demand composition and intersection layout are such that there are no potential conflicts between vehicles performing different movements (which means there is no interaction at all between different movements), then each of the allowed movements can serve up to its maximum capacity, and the total capacity of the intersection is simply the sum of the capacities of each movement.

As an example, assuming the intersection layout and vehicle parameters presented in Section 4.1, if demand composition is such that every vehicle turns right, there is no potential transversal conflict. With this type of demand, capacity flow would mostly be a function of the allowed minimum headway. Assuming $h_{\mathrm{L}} = 0.5$ s, and the conditions used for the simulations presented in Section 4.1, $h_{a,i,b,j,c}$ would, on average, be equal to 1.45 s, leading to a capacity of 2951 veh/h for each right turn movement, and 11804 veh/h in total. However, this is just a corner case.

In the vast majority of demand compositions there are transversal conflicts. In this case, overall capacity is no longer simply the sum of the capacity of each movement, as vehicle interactions have to be taken into account. Even though it is challenging to formalize these interactions to obtain a meaningful value for average capacity, it is possible to, under certain assumptions, at least obtain an upper bound for the capacity flow.

Capacity flow through an intersection controlled by OATS can be formalized as an optimization problem under the (very) simplifying assumption that the average minimum headway between vehicle pairs at a conflict region, h_c, meaningfully describes the capacity flow that crosses that conflict region.

This simplifies traffic heterogeneity by ascribing a single headway value for each conflict region. Recall that $h_{a,i,b,j,c}$ is the headway between vehicles i and j at conflict region c. This value can be different for each vehicle pair, depending mostly on the interaction type (longitudinal or transversal). On an aggregate level, it can be said that there is an average minimum headway between two vehicles that interact at conflict region c, given by h_c:

$$h_c = (h_{\mathrm{L}} + \tau_c^{\mathrm{enter}}) \cdot P_c^{\mathrm{L}} + (h_{\mathrm{T}} + \tau_c^{\mathrm{cross}}) \cdot P_c^{\mathrm{T}} + \hat{h}^{\Delta \mathrm{v}} \qquad (3.20)$$

with P_c^{L} the probability two vehicles that cross conflict region c consecutively have a longitudinal interaction, P_c^{T} the probability of a transversal interaction, τ_c^{enter} the expected time for the average vehicle involved in a longitudinal interaction to enter the conflict region, τ_c^{cross} the expected time for the average vehicle involved in a transversal interaction to cross the conflict region, and $\hat{h}^{\Delta \mathrm{v}}$ the expected speed difference headway.

For the purposes of obtaining an estimation for intersection capacity, it is assumed that the capacity flow through each conflict region c is constrained only by h_c. Formally:

$$\mathrm{F}_c^{\mathrm{cap}} = \frac{3600}{h_c}, \tag{3.21}$$

where $\mathrm{F}_c^{\mathrm{cap}}$ is the capacity flow (in vehicles per hour) through conflict region c, assuming it is possible for a vehicle to cross conflict region c every h_c seconds. This is a very significant simplification, as it implies each time h_c elapses there is a vehicle able to instantly reach conflict region c. This is clearly not the case in reality, where the time between two vehicles occupy c can be much larger than h_c due to complex interactions involving several vehicles. It is also noteworthy that maintaining h_c between vehicles, if possible, likely requires very regular and particular patterns of vehicle arrival at the CR.

Even though considering only h_c may significantly overestimate actual capacity, it can be useful to investigate the upper bound for capacity it implies. With OATS, it is not possible to have higher capacity than $\mathrm{F}_c^{\mathrm{cap}}$ at any one conflict region.

Let the flow at the intersection be defined as the sum of vehicle flows through each movement $m \in M$ in the intersection:

$$\mathrm{F}_{\mathrm{IR}} = \sum_{m \in M} \mathrm{F}_m, \tag{3.22}$$

where F_{IR} is total flow through the IR, in vehicles per hour, and F_m is the flow, also in vehicles per hour, through movement $m \in M$. The total flow of all movements that pass through each conflict region must not violate the capacity flow of the conflict region:

$$\sum_{m \in M_c} \mathrm{F}_m \leq \mathrm{F}_c^{\mathrm{cap}} \qquad \forall\, c \in C \tag{3.23}$$

where M_c is the set of movements $m \in M$ that cross conflict region c, with $M_c \subset M$.

Capacity flow at the intersection $\mathrm{F}_{\mathrm{IR}}^{\mathrm{cap}}$ is, by definition, the maximum value F_{IR} can have. As such, it can be obtained by maximizing (3.22) subject to (3.23). For the intersection layout, with OATS configuration and demand[9] as described in Section 4.1 and as used for

[9]Demand consists of 60% of vehicles traveling straight and 20% turning to each direction. This is incorporated in the problem described here through additional

simulations, the solution of this maximization problem results in an upper bound for F_{IR} of approximately 8500 vehicles per hour, which is a little under twice the maximum flow observed in simulations.

3.7 SUBPROBLEM 4 - DEFINING SPEED PROFILES FOR APPROACHING VEHICLES

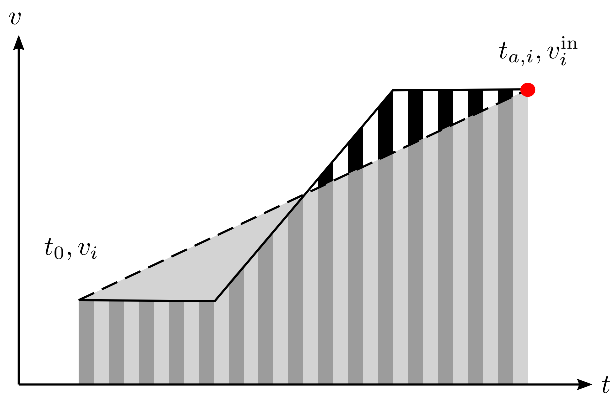
Consider vehicle i located at a distance d_i from the intersection with speed $v_{0,i}$, at a time instant $t = 0$. The solution of SP3 gives the time $t_{a,i}$ the vehicle should reach the intersection, and the speed profile S_i obtained from SP1 includes the entrance speed v_i^{in}. SP4 consists in finding, for each vehicle, a speed profile that allows it to reach the intersection with speed v_i^{in} at the scheduled time $t_{a,i}$ without colliding with other vehicles and while constrained by $v_i^{\max}$, $a_i^{\max}$ and $a_i^{\min}$.

Several such speed profiles may exist. Fig. 9 illustrates a pair of trajectories in a time / speed plot that reach the intersection at the same time, one depicted by a solid line and the other by a dashed line. The area below any curve depicting a valid trajectory is the same, and corresponds to the distance to the intersection.

There are a variety of possible strategies for finding a suitable speed profile. For instance, one could formulate an optimization problem (KATRAKAZAS et al., 2015), taking criteria such as fuel consump-

constraints.

Figure 9 – Time / speed plot of two possible trajectories for vehicle i



tion (MINETT et al., 2011) or passenger comfort (VINE; ZOLFAGHARI; POLAK, 2015) into account.

In this thesis, SP4 is divided into one problem for each approach, and modeled in a discrete way as an optimal control problem. More specifically, SP4 is formalized as the problem of finding, for each vehicle i in a given approach a, a motion plan comprised by a sequence of desired states, in the form of target positions, speeds and accelerations for each control interval to be followed by the vehicle along the time horizon stretching from the current time to the target time $t_{a,i}$ to reach the intersection.

Consider a set of n_a vehicles arriving at the intersection from approaching lane a. The control horizon of each vehicle i that just entered the CR is its scheduled arrival time at the intersection, $t_{a,i}$, assuming current time instant is $t = 0$. The horizon is divided in a number of time intervals $\tau_{i,k}$, with $k = 1, \ldots, n_i$. Each time interval has a duration given by a fixed time step T_{S}, with the exception of the last time interval, which is shorter in order to satisfy:

$$\sum_{k=1}^{n_i} \tau_{i,k} = t_{a,i} \qquad\qquad i = 1, \ldots, n_a. \tag{3.24}$$

where n_i is the number of control intervals in the control horizon of vehicle i.

For example, suppose $T_{\mathrm{S}} = 0.2$ and a given vehicle i has $t_{a,i} = 0.75$, i.e., $n_i = 4$. The time intervals for this vehicle would be $\tau_{i,1} = \tau_{i,2} = \tau_{i,3} = 0.2$ and $\tau_{i,4} = 0.15$. This allows modeling vehicles with different and arbitrary arrival times and also comparing the state of different vehicles inside the approach at any time interval, except the last for each vehicle. Safety at the last time interval is equivalent to safety upon arrival at the intersection, which is guaranteed by SP3.

The state of vehicle i at time interval k is described by

- $d_{i,k}$, the distance from the front bumper to the nearest entrance of the IR at the end of time interval k;
- $v_{i,k}$, the instant speed at the end of time interval k; and
- $a_{i,k}$, the (constant) acceleration during time interval k.

These state variables should be consistent regarding vehicle dynamics:

$$
\begin{aligned}
& d_{i,k} = d_{i,k-1} - 0.5 \cdot (v_{i,k-1} + v_{i,k}) \cdot \tau_{i,k} && (3.25a)\\
& v_{i,k} - v_{i,k-1} = a_{i,k} \cdot \tau_{i,k} && (3.25b)\\
& v_{i,k} \leq v_i^{\max} && (3.25c)\\
& a_i^{\min} \leq a_{i,k} \leq a_i^{\max}. && (3.25d)\\
& \quad k = 2, \ldots, n_i;\ i = 1, \ldots, n_a
\end{aligned}
$$

Initial conditions $d_{i,0}$ and $v_{i,0}$ are given by the current (known) state of each vehicle when SP4 is solved.

Ideally, $v_{i,n_i} = v_i^{\text{in}}$ and $d_{i,n_i} = 0$, at time $t_{a,i}$. However, by assuming a discretization scheme with constant acceleration during each time interval such conditions may become unfeasible. These requirements are relaxed by setting $\varepsilon^{\text{v,SP4}}$ and $\varepsilon^{\text{d,SP4}}$ as the maximum allowed deviation for final speed and distance, respectively, obtaining the following bounds for the final conditions:

$$
\begin{aligned}
& -\varepsilon^{\text{d,SP4}} \leq d_{i,n_i} \leq \varepsilon^{\text{d,SP4}} && (3.26a)\\
& -\varepsilon^{\text{v,SP4}} \leq v_{i,n_i} - v_i^{\text{in}} \leq \varepsilon^{\text{v,SP4}} && (3.26b)\\
& \quad i = 1, \ldots, n_a.
\end{aligned}
$$

Let $\delta_i^{\min}$ be the minimum distance between vehicle i and vehicle $i+1$ immediately upstream, given by the sum of the vehicle length L_i and a minimum safety gap $g_{\min}$:

$$\delta_i^{\min} = L_i + g_{\min}. \tag{3.27}$$

The following family of safety constraints is defined in order to keep vehicles within a safe distance from each other:

$$
\begin{aligned}
& d_{i,k} + \delta_i^{\min} \leq d_{i+1,k} && (3.28)\\
& k = 1, \ldots, n_i;\ i = 1, \ldots, n_a - 1.
\end{aligned}
$$

Finally, let F be the a cost function such that

$$F = \sum_{i=1}^{n_a} (d_{i,n_i})^2 + \sum_{i=1}^{n_a} (v_{i,n_i} - v_i^{\text{in}})^2. \tag{3.29}$$

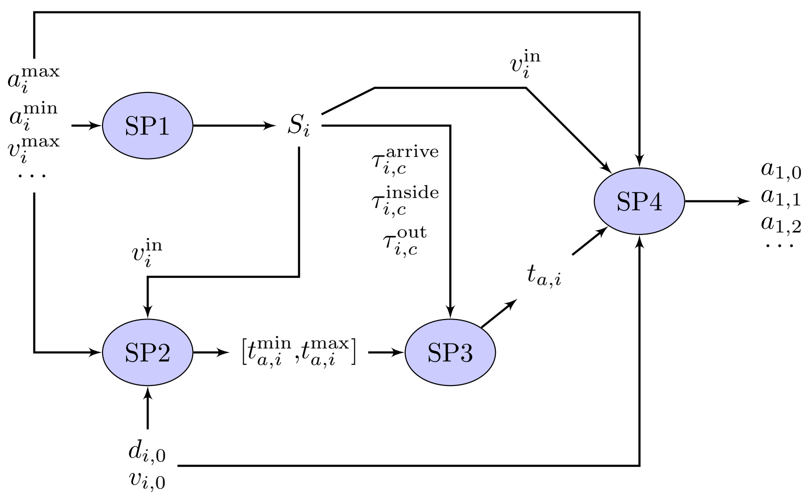
The two terms of F correspond, respectively, to minimizing the square of the distance to the intersection (which should ideally be 0) and the deviation from the desired speed at the last control interval, for each vehicle. In short, F minimizes the quadratic error in keeping to the schedule obtained from SP3. The goal of this design is to allow a safe and efficient crossing (as the schedule guarantees minimum time). Alternatively, it is possible to define a cost function that takes into account any number of arbitrary criteria, such as energy consumption, as discussed in Chapter 5.

Formally, SP4 consists of minimizing F (i.e., (3.29)), subject to (3.25)–(3.28). Note that, in order to guarantee SP4 is feasible, vehicle behavior outside the CR must be such that vehicles enter the CR with distances at least as great as $\delta_i^{\min}$ from the vehicle in front.

3.8 CONTROL STRUCTURE

Fig. 10 depicts the inputs and outputs of each of OATS' subproblems in a flowchart. Vehicle characteristics, such as $a_i^{\max}$, $a_i^{\min}$ and $v_i^{\max}$ are used as inputs for SP1, SP2 and SP4. SP1 produces a speed profile S_i for each vehicle i to cross the IR. Two types of information which can be gathered from these speed profiles are used by other subproblems: (i) the speed each vehicle has when entering the

Figure 10 – OATS flow chart, illustrating the inputs of outputs of each subproblem



intersection, v_i^{in}, is used as input for SP2 and SP4; and (ii) the time intervals a vehicle takes to travel from the intersection entrance to certain points inside the intersection, $\tau_{i,c}^{\text{arrive}}$, $\tau_{i,c}^{\text{inside}}$, $\tau_{i,c}^{\text{out}}$, are used as inputs for SP3. The current state of each vehicle i, given by its position $d_{i,0}$ and speed $v_{i,0}$, is used as an input for SP2 and SP4. SP2 generates the feasible arrival intervals for each vehicle, $[t_{a,i}^{\min}, t_{a,i}^{\max}]$. These intervals are used as an input of SP3, which schedules an desired arrival time $t_{a,i}$ for each vehicle. Finally, SP4 uses the scheduled arrival times and produces a sequence of control inputs for each vehicle, in the form of target accelerations for each following time step.

The proposed implementation of OATS relies in both the IC and the vehicles inside the CR. The IC is responsible for solving SP2, SP3 and SP4, while vehicles are responsible for solving SP1 and complying with commands received from the IC.

It is assumed that each individual vehicle has its own control loop that enables it to follow a reference trajectory. Vehicle trajectory inside the CR is composed by a path, which is already known, and a speed profile. The part of the speed profile inside the IR, S_i, is given by SP1, and the part outside the IR is given by SP4.

Vehicles are assumed capable of following the reference speed profile and keeping on their path. Although the IC calculates collision free trajectories, a vehicle can (and in a practical implementation should) have collision avoidance as part of its (lower level) control loop.

In order to implement OATS, the IC executes a number of tasks periodically. Each time a control interval T_S elapses, the following tasks are performed:

1. Any vehicle i that just entered the CR solves SP1 and informs the IC about their intended path $m_i \in M$ and speed profile S_i inside the IR, as well as its own driving capabilities (i.e., $v_i^{\max}$, $a_i^{\min}$, $a_i^{\max}$), and the intervals for arrival, entrance, and leaving each conflict region $c \in C_{m_i}$.

2. The IC gathers state data from vehicles inside the CR.

3. The IC solves SP2 for all vehicles inside the CR, or a subset of them in case some are left out for simplification reasons.

4. The IC checks if (i) a vehicle that was not in the CR or was disregarded in a previous solution needs to be taken into account now; or (ii) there exists a vehicle that was unable to keep the previous schedule. If any of these conditions is true, SP3 is solved

and a new schedule is obtained. Otherwise, the last scheduled times are maintained.

5. If (i) a new schedule has just been calculated; or (ii) the IC detects that a vehicle has failed to follow the last speed profile received, the IC solves SP4 and sends new reference trajectories to each vehicle inside the CR. Otherwise, the last calculated trajectories should still be followed.

The concepts of failing to keep schedule and failing to follow a speed profile are explained on Section 3.8.0.1. The IC is assumed to have perfect information, "knowing" the state of each vehicle at each time step. In practice, vehicles can send periodic state messages and the IC can estimate the current state based on the last messages, and possibly use sensor data as well. In that case, minimum safety headways may need to be increased accordingly to the uncertainties involved. When a vehicle first enters the CR it may be necessary for it and the IC to exchange messages in order to inform the vehicle that it is now inside a CR, and send the (geometric) description of the intersection so it can calculate a suitable speed profile (i.e., solve SP1).

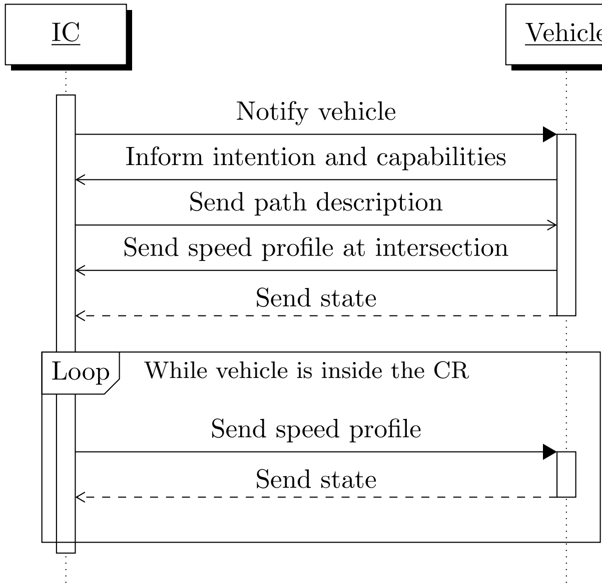
Fig. 11 illustrates the interaction between the IC and a vehicle while performing the aforementioned tasks. They first exchange information about the vehicle intention and profile inside the IR, and then enter a loop in which the vehicle continually informs its state (position, speed and acceleration) and receives speed profiles to follow. The interaction ceases once the vehicle exits the CR.

Another relevant aspect of the control structure is that as SP3 is repeatedly solved, previous solutions can still be useful. The solver used to implement OATS allows the setting of an initial "guess" value for the decision variables from which to start the optimization. Previous solutions are used as the initial guess value for the decision variables that correspond to vehicles that were already present in previous problem instances, under the assumption that it is likely that the schedule of most vehicles will usually not change, and that a previous solution should still be feasible. In practice, however, this seemed to have little to no effect on solution time.

3.8.0.1 Uncertainties in trajectory following

Consider vehicle i on approach a inside the CR, at a distance d_i from the intersection, with speed v_i and a feasible arrival interval

Figure 11 – Sequence diagram of communication between a vehicle and the IC



$[t_{a,i}^{\min}, t_{a,i}^{\max}]$. Consider that this vehicle was scheduled in a previous control interval to arrive at the intersection at a given time $t_{a,i}$, which, minus the time that has elapsed since then, would mean its current scheduled time should be $\hat{t}_{a,i}$ (in relation to the current time).

Ideally, $\hat{t}_{a,i}$ would fall inside the current feasible arrival time interval, i.e.:

$$\hat{t}_{a,i} \in [t_{a,i}^{\min}, t_{a,i}^{\max}], \tag{3.30}$$

meaning it should be possible for a vehicle to arrive at a time previously scheduled. However, due to the fact that the interval $[t_{a,i}^{\min}, t_{a,i}^{\max}]$ can be very narrow and vehicles may be unable to follow the reference trajectory exactly (due to uncertainties, disturbances, etc), a maximum allowed time deviation from the schedule is defined, $\xi^{\mathrm{t,SP3}}$. A vehicle is said to be keeping the schedule if the following holds:

$$t_{a,i}^{\min} - \xi^{\mathrm{t,SP3}} \leq \hat{t}_{a,i} \leq t_{a,i}^{\max} + \xi^{\mathrm{t,SP3}}. \tag{3.31}$$

Consider this same vehicle also received previously a reference speed profile in order to meet the schedule. Suppose that if the vehicle followed this reference precisely, it would be expected to be at distance $\hat{d}_i$ from the intersection and have speed $\hat{v}_i$.

Let $\xi^{\mathrm{v,SP4}}$ and $\xi^{\mathrm{d,SP4}}$ be the maximum allowed speed and position tolerances for trajectory following, respectively. A vehicle is said to be following its trajectory successfully (i.e., complying with the last known solution of SP4) if the following holds:

$$\hat{v}_i - \xi^{\mathrm{v,SP4}} \leq v_i \leq \hat{v}_i + \xi^{\mathrm{v,SP4}}, \tag{3.32a}$$

$$\hat{d}_i - \xi^{\mathrm{d,SP4}} \leq d_i \leq \hat{d}_i + \xi^{\mathrm{d,SP4}}. \tag{3.32b}$$

Since an eventual inability to keep the schedule can only happen if a vehicle fails to follow the reference trajectory, and SP4 is easier (computationally) to solve than SP3, it is preferable to correct eventual deviations by solving only SP4 and avoid having to reschedule vehicles. To achieve this, $\xi^{\mathrm{v,SP4}}$ and $\xi^{\mathrm{d,SP4}}$ are set significantly small in relation to $\xi^{\mathrm{t,SP3}}$, so a failure in following a speed profile is more easily "triggered" than a failure in complying with the schedule. The resulting behavior is that, when a (significant) deviation is detected, SP4 is solved again and new speed profiles are found, which allow vehicles to keep the schedule. Even though it is expected that it is never needed to solve SP3 again due to errors (and in fact this never happened in any of the simulations performed), this possibility is still implemented for completeness.

Notice that the minimum safety headways $h^{\mathrm{safe}}_{a,i,b,j,c}$ (from SP3) and minimum gap d_i^{min} (from SP4) should be large enough in relation to the maximum allowed deviations ($\xi^{\mathrm{t,SP3}}$, $\xi^{\mathrm{v,SP4}}$ and $\xi^{\mathrm{d,SP4}}$) and the maximum possible error in one time step as to guarantee safety (this assumes there are known bounds to the uncertainties involved). Similarly, uncertainties with measuring, actuation and communication delay can be addressed by imposing sufficiently large headways and gaps for the worst case scenario.

In general, the uncertainties associated to communication delays should be very small, as the time delay for transmitting vehicular information with the current technology over the considered distances is in the order of a couple milliseconds (LEVIN; FRITZ; BOYLES, 2017; MILANES et al., 2012).

4 SIMULATION RESULTS FOR DIFFERENT CONFIGURATIONS OF OATS

The OATS Strategy has several parameters which can be chosen with a certain degree of freedom, such as the control distance or minimum headways. Besides that, there are also several possible simplifications which can be adopted, as discussed in Section 3.6.3, for reducing the size of the problem. Several OATS configurations were evaluated in simulation in order to investigate the trade-off between performance and computational effort, under different traffic demands. This chapter presents and discusses these simulation results.

The OATS Strategy was implemented and used to manage an intersection modeled in the microscopic traffic simulator Aimsun (Transport Simulation Systems, 2015). The Gurobi solver (Gurobi Optimization, 2016) was used for solving both SP3 and SP4. Both Aimsun and Gurobi have an Application Programming Interface (API) for the Python programming language. OATS was implemented in Python, interfacing with both Aimsun and Gurobi.

Section 4.1 presents the intersection layout and the values used for vehicle and control parameters. Section 4.2 details the different scenarios considered and the performance indexes used to evaluate them. Section 4.3 presents and discusses simulation results.

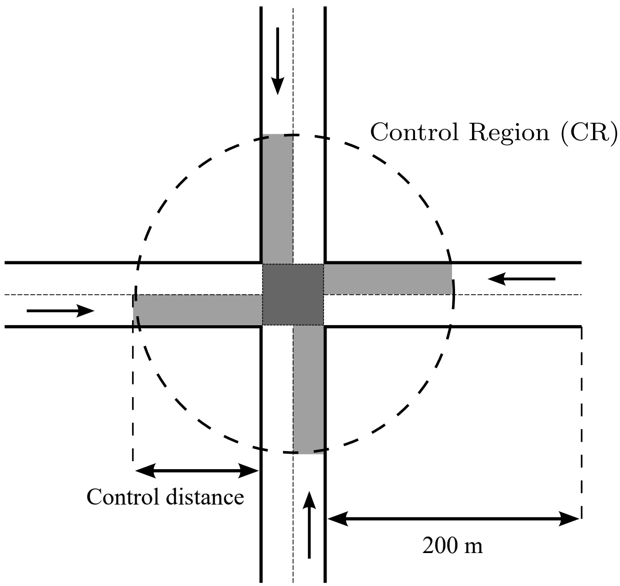
4.1 SIMULATION SETUP

The simulation scenario consists of four 200 m long, two-way roads with one lane in each direction that meet at one cross intersection, as depicted in Fig. 12. The CR is composed by the intersection itself and its approaches up to a Control distance D_{a} from the IR. By default, $D_{\mathrm{a}} = 100$ m, but this value changes for several scenarios.

Figure 13 shows a detailed layout of the IR. The gray area corresponds to the IR. Approaching lanes and exits are denoted by letters N, E, S and W. The set of approaches is defined as $A = \{\mathrm{N, E, S, W}\}$. There are 12 allowed movements connecting approaches and exits, depicted by dashed lines. They can be denoted by pairing approaching lanes and exits. The set of allowed movements is M ={WN, WE, WS, SW, SN, SE, ES, EW, EN, NE, NS, NW}.

Lanes are 3 m wide and the IR is approximately a square with side 12 m (with “cut” corners, as in Figure 13). All turns follow an arc

Figure 12 – Intersection layout. The dark gray square where the roads meet is the IR. The CR is composed by the IR and the portion of the approaching lanes less than D_a m away from the IR, depicted in lighter gray. The dashed circle represents the control distance, which actually does not need be the same for each approach. The CR fits inside this circle.

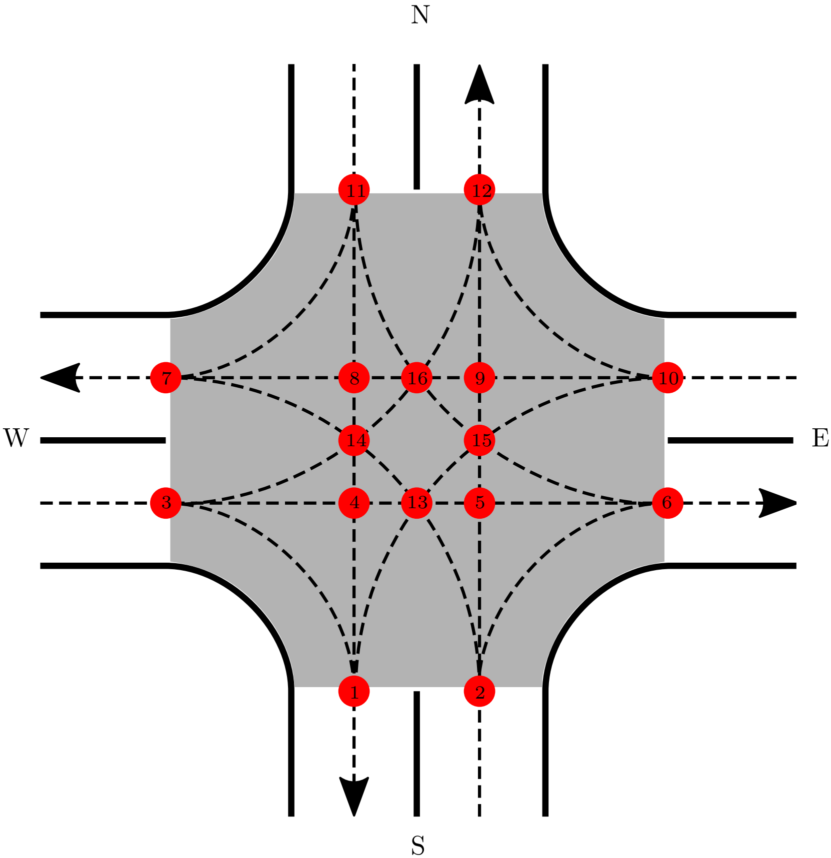


corresponding to a quarter of a circle. Right turns have a radius of 4.5 m, and left turns a radius of 7.5 m.

There are 16 conflict regions, denoted by numbers as $C = \{1, 2, ..., 15, 16\}$. Each one is associated to and centered at one of the 16 conflict points where movements cross, which are numbered and highlighted by red circles in Figure 13. Conflict regions are circles with a radius of 2.5 m, except for the ones at the edges of the IR (regions 1, 2, 3, 6, 7, 10, 11 and 12), which are half circles extending only inside the IR.

Recall $C_m \subseteq C$ denotes the set of conflict regions c which are intercepted by movement m. For this intersection, the sequence of

Figure 13 – Intersection Region layout. The IR is grayed out. The 12 allowed movements are depicted by dashed lines. The 16 conflict points are numbered and highlighted by red circles.



conflict regions crossed by each movement can be described as:

- C_{WN} ={3, 14, 16, 12}
- C_{WE} ={3, 4, 13, 5, 6}
- C_{WS} ={3, 1}
- C_{SW} ={2, 13, 14, 17}
- C_{SN} ={2, 5, 15, 9, 12}

- C_{SE} ={2, 6}
- C_{ES} ={10, 15, 13, 1}
- C_{EW} ={10, 9, 16, 8, 7}
- C_{EN} ={10, 12}
- C_{NE} ={11, 16, 15, 6}
- C_{NS} ={11, 8, 14, 4, 1}
- C_{NW} ={11, 7}

Simulation time step is $T_{\text{S}} = 0.2$ s. Speed limit and $v_i^{\max}$ for every vehicle are set to 30 km/h. Vehicles are set to solve SP1 by choosing a constant speed inside the IR with $v_i^{\text{in}} \in [25, 30]$ km/h for vehicles traveling straight and $v_i^{\text{in}} \in [15, 25]$ km/h for turning vehicles (exact values are sampled from an uniform distribution). Parameters $a_i^{\min}$ and $a_i^{\max}$ are also sampled from uniform distributions, with $a_i^{\min} \in [-5, -3]$ m/s^2 and $a_i^{\max} \in [2.5, 3.5]$ m/s^2. For all scenarios $L_i = 4$ m, $g_{\min} = 0.5$ m, $\varepsilon^{\text{d,SP4}} = 0.5$ m, $\varepsilon^{\text{v,SP4}} = 0.1$ m/s, $\xi^{\text{t,SP3}} = 0.2$ s, $\xi^{\text{v,SP4}} = 0.1$ m/s, $\xi^{\text{d,SP4}} = 0.1$ m, $Q = 1000$, $h_c^{\text{cap}} = 0$ s. Any $t_{a,i}^{\max} > 120$ s is considered to be 120 s instead. By default, $h_{\text{L}} = 0.5$ s and $h_{\text{T}} = 0.4$ s, but this changes for some scenarios.

Vehicles outside the CR are controlled by an ACC algorithm, as presented in Section 2.4, with $K_1 = 1.2$ s^{-2}, $K_2 = 1.7$ s^{-1}, and T_d sampled from an uniform distribution with $T_d \in [0.8, 1.0]$ s. Additionally, if the target distance from the vehicle ahead, given by $T_{\text{d}} \cdot v_i$ is less than 2.5 m, it is instead saturated to 2.5 m, effectively forcing 2.5 m as the minimum gap.

4.2 SCENARIOS AND PERFORMANCE INDEXES

Several scenarios were simulated for evaluating OATS. In general, different configurations of OATS were evaluated under two different traffic demands: 400 veh/h/lane and 800 veh/h/lane. These are refereed to as *low* and *high* traffic demand scenarios, respectively. In all cases vehicle entrance times are sampled from an exponential distribution. Upon entering the network each vehicle is assigned to a random destination, with 0.6 probability of traveling straight, and 0.2 probability of turning to each direction.

Note that the low demand used is actually relatively high if compared to a typical intersection. The value of 400 veh/h/lane was chosen because it is similar to the capacity of a traffic light servicing the intersection (assuming a road capacity of 1800 veh/h/lane and a fixed control plan with a cycle length of 90 s. For more details on a fixed time control for this intersection see Appendix C), allowing to investigate how effective OATS is compared to a traffic light. The value of 800 veh/h/lane was chosen as an arbitrarily high demand – much higher than what can be serviced with current technology –, to study how OATS behaves under very high demand conditions. Simulation results show that OATS achieves very good performance even at the high demand scenarios.

Since OATS must work in real time, it should provide a solution in a reasonable small amount of time. However, when performing the simulations it became apparent that a small portion of the optimization problems solved are significantly harder than most problem instances.

A maximum allowed execution time $T^{\text{opt}}_{\max}$ is set for solving SP3 in certain scenarios. In such cases, if a given problem instance takes longer than $T^{\text{opt}}_{\max}$ to be solved, the algorithm terminates the search and returns the best (suboptimal) feasible solution found so far. This mechanism was used in some experiments to limit the solution time of SP3 to $T^{\text{opt}}_{\max} = 0.1$ s (i.e., SP3 is solved by an anytime algorithm).

Notice that, assuming that vehicles that were already inside the CR are still able to keep the last schedule found, finding a new feasible (albeit non-optimal) solution for SP3 is trivial and can be done very quickly. A feasible schedule can be found by keeping all vehicles that were already inside the CR with the same schedule as previously (adjusted by the elapsed time since the previous schedule was computed), and sequentially scheduling vehicles that just entered the CR a sufficiently long time later than any other vehicle.

Each scenario was simulated 10 times (i.e., 10 replications). Each replication consists of 10 minutes of simulation. Approximately 250 problem instances are solved for most of low demand replications, and 500 for the high demand ones. The most common number of vehicles inside the CR at a given time is 7 for the low demand scenarios and 15 for the high demand scenarios (except in cases when the size of the control region or number of vehicles is decreased by some simplification). Results shown for each scenario correspond to the average values obtained across all replications.

The two main aspects of interest for evaluating OATS in simulation are traffic conditions and the computational effort needed. In order to capture these, the following metrics were chosen to evaluate the scenarios:

- average delay, which is the difference between the time a vehicle takes to cross the simulated network and the time it would take if it were the only vehicle on the network and nothing impeded its movement;
- average vehicle speed;
- average solution time for SP3, which is typically much higher than for SP4, and effectively the bottleneck in computation time; and
- percentage of timeouts, which is the percentage of problem instances for which a suboptimal solution was used because the solver reached the allowed optimization time.

For brevity, in the tables these metrics are refereed to simply as “delay”, “speed”, “solve time”, and “% timeout”, respectively. Values following a ± symbol correspond to standard deviations.

There is a significant variation in vehicle delay due to the fact that some vehicles have to decelerate while others can travel unimpeded. Although this behavior minimizes the total time to arrive at the intersection, from the perspective of a passenger in an arbitrary vehicle it may seem “unjust” that some vehicles have higher delays than others. In this context, the term “justice” refers to equality in vehicle delay. Standard deviation for vehicle speed is omitted because a significant portion of the speed variance is due to the fact that vehicles have different desired speeds for crossing the intersection, meaning the standard deviation of vehicle speed is not as reliable as the standard deviation of vehicle delay to evaluate how “unjust” traffic becomes. Standard delays for the solution time show the variation on the difficulty in solving SP3, which is mainly affected by the size of the problem. This can vary significantly depending on the number of vehicles and their origins and destinations. If the vehicles and their movements are such that there is a larger number of transversal conflicts, the problem tends to become harder.

Note that, given the specified network and vehicle parameters, vehicles that are not delayed take between 49.44 and 51.61 seconds to cross the simulated intersection (close to 50.5 seconds on average). This means that each second of delay time translates to an increase of approximately 2% in the travel time a vehicle experiences.

Most experiments were performed by varying one parameter of OATS while keeping the other fixed. The base scenario consists of the

OATS strategy without any of the simplifications discussed in Section 3.6.3, with control distance $D_{\mathrm{a}} = 100$ m, default transversal headway $h_{\mathrm{T}} = 0.4$ s and longitudinal headway $h_{\mathrm{L}} = 0.5$ s.

Experiments were performed on a laptop with a 2.4 GHz Intel i7-4700MQ CPU and 16 GB of RAM. All software used (the Aimsun simulator, the Gurobi solver, and the OATS implementation and interfaces in Python) ran on the same machine.

4.3 SIMULATION RESULTS

The experiments performed can be grouped in the following scenario sets:

1. experiments evaluating the impact of different values for the transversal headway h_{T};
2. experiments evaluating the impact of different values for the longitudinal headway h_{L};
3. experiments evaluating the impact of different values for the control distance D_{a};
4. experiments in which vehicles are scheduled in batches;
5. experiments in which vehicles near the IR are set to not update their schedule;
6. experiments in which there is a limit on the number of vehicles with unbounded maximum arrival time $t_{a,i}^{\max}$ that can be considered in a problem instance;
7. experiments in which vehicles with "tight" feasible arrival intervals have their arrival time set to an approximated value before SP3 is solved; and
8. experiments with different traffic demands.

Scenarios Sets 1–7 were evaluated under three situations:

(i) with a low traffic demand (400 veh/h/lane) and no limit on the maximum allowed execution time.

(ii) with a high traffic demand (800 veh/h/lane) and no limit on the maximum allowed execution time.

(iii) with a traffic high demand (800 veh/h/lane) and a maximum allowed execution time $T_{\max}^{\text{opt}} = 100$ ms.

Set 8 consists of scenarios for which traffic demands range from 200 veh/h/lane to 1600 veh/h/lane, and no limit on the maximum allowed execution time.

4.3.1 Experiments with different values for the transversal headway h_{T}

As discussed in Section 3.6.2, SP3 is formulated in such a way that, even if there is no default safety transversal headway ($h_{\text{T}} = 0$), no two vehicles whose trajectories cross can share a conflict region simultaneously. Hence, if the conflict regions are sufficiently large, transversal collisions are not possible even if $h_{\text{T}} = 0$. The purpose of h_{T} is to implement some measure of additional safety, increasing the minimum time between vehicles are allowed to access a conflict region to account for possible uncertainties such as measuring errors, communications delays, etc. Note that by increasing h_{T} this "safety window" increases at the cost of some efficiency, as vehicles have to wait longer to be granted the right to occupy a conflict region. As such, the choice for the value of h_{T} is a compromise between safety and efficiency. The first set of experiments is designed to examine how much efficiency is impacted by this choice. Experiments are performed with $h_{\text{T}} \in \{0, 0.2, 0.4, 0.6, 0.8, 1\}$ seconds, for the three situations mentioned in Section 4.3.

Table 1 shows simulation results, some of which – average delays and average time to solve a problem instance – are also shown in Figure 14. As expected, performance decreases as h_{T} increases, which can be seen both by the increasing vehicle delay and decreasing vehicle speed. Overall, average vehicle speeds are very close to the speed limit (30 km/h), indicating vehicles are able to travel mostly unimpeded, briefly decelerating just enough to avoid collisions with other vehicles.

Delays are very small, with vehicles being delayed, on average, less than one second for the low demand scenario (unless when $h_{\text{T}} = 1$ s, the largest value tested). For comparison, consider that the low demand is close to the capacity of an intersection controlled by conventional traffic lights, and vehicles under this type of control typically have much larger delays. Experiments were performed with a fixed time (traffic light) control for this intersection with a lower demand (see Appendix C), in which vehicles were found to experience an average delay close to 90 seconds, and average speed of approximately 12.9 km/h.

Table 1 – Simulation results varying transversal headway h_T

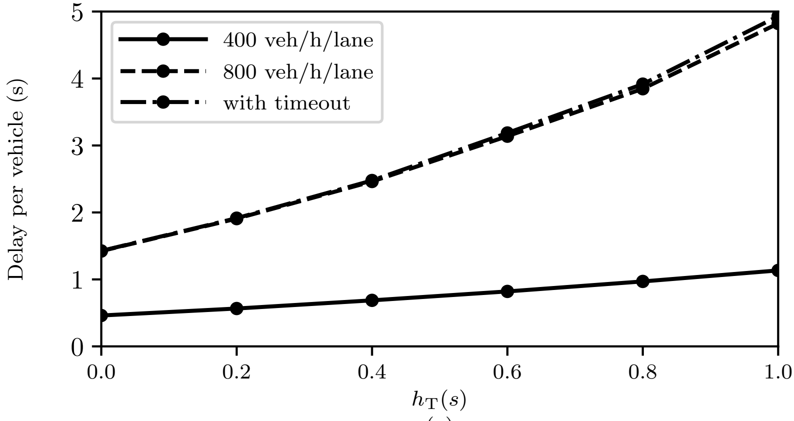
h_T (s)	delay (s/veh)	speed (km/h)	solve time (ms)	% timeout
Scenarios with low demand				
0	0.46 ± 0.84	29.46	2.3 ± 2.2	–
0.2	0.57 ± 0.99	29.40	2.7 ± 2.5	–
0.4	0.69 ± 1.15	29.33	2.8 ± 2.8	–
0.6	0.82 ± 1.34	29.26	2.8 ± 2.8	–
0.8	0.97 ± 1.49	29.18	3.3 ± 3.6	–
1	1.13 ± 1.69	29.09	3.4 ± 3.8	–
Scenarios with high demand				
0	1.42 ± 1.73	28.90	15.8 ± 112.9	–
0.2	1.91 ± 2.20	28.65	25.3 ± 188.0	–
0.4	2.46 ± 2.67	28.36	39.7 ± 346.6	–
0.6	3.14 ± 3.23	28.01	47.8 ± 355.2	–
0.8	3.85 ± 3.81	27.67	72.9 ± 483.4	–
1	4.82 ± 4.57	27.22	77.6 ± 289.2	–
Scenarios with high demand and $T_\mathrm{max}^\mathrm{opt} = 0.1$ s				
0	1.43 ± 1.72	28.90	11.0 ± 15.3	1.1 ± 0.8
0.2	1.91 ± 2.20	28.64	15.1 ± 20.4	2.6 ± 1.6
0.4	2.48 ± 2.68	28.35	19.6 ± 24.0	4.3 ± 1.9
0.6	3.19 ± 3.31	27.99	25.6 ± 28.7	7.6 ± 3.1
0.8	3.92 ± 3.84	27.64	29.6 ± 30.3	9.4 ± 4.8
1	4.93 ± 4.71	27.16	37.4 ± 34.2	15.0 ± 6.7

For the high demand cases the average vehicle delay is larger, reaching almost 5 s. This is still much lower than the delay that results from a fixed time control for a scenario with less than half that demand. Note there is no equivalent fixed time result for the high demand scenario, as a demand this high is well over the capacity of fixed time control and would result in queues spilling up, and vehicle delay would increase indefinitely as long as the simulation continues.

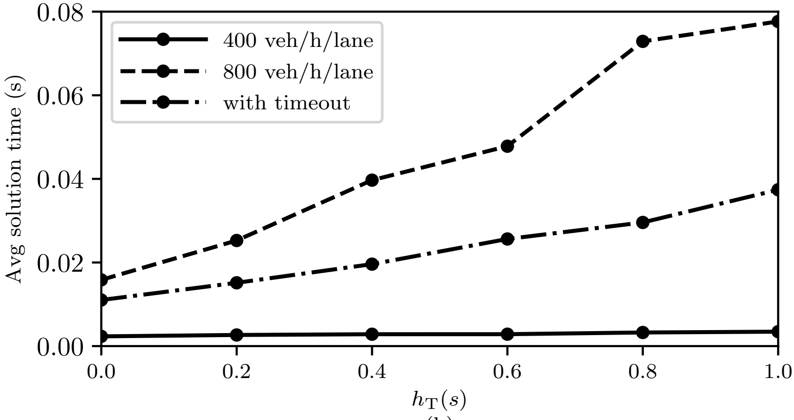
When a time limit $T_\mathrm{max}^\mathrm{opt}$ is set, up to 15 % (when $h_\mathrm{T} = 1$) of problem instances are terminated early due to timeouts and a suboptimal solution is used, substantially decreasing average solution time, and greatly decreasing the standard deviation of the solution time (which is very high for the high demand scenario without $T_\mathrm{max}^\mathrm{opt}$). The impact of occasionally using a suboptimal solution on performance, however, is very small. The largest increase in delay is just over 0.1 s, for $h_\mathrm{T} = 1$ s, and even less for the other cases.

Figure 14 makes it very apparent that the low demand scenarios have much lower delays and are significantly easier to solve than the high demand ones. It is also apparent that setting $h_\mathrm{T} = 0.1$ s has an

Figure 14 – Average vehicle delay (a) and average solution time (b) results for different transversal headways h_{T}



(a)



(b)

almost negligible effect on traffic efficiency, but significantly reduces the average solution time. Interestingly, while the average solution time is reduced by, in some cases, roughly a factor of 2, only a small portion (less than 10%, except for $h_{\mathrm{T}} = 1$) of problem instances actually take longer than 0.1 seconds to solve. The reason is that, although usually SP3 takes little time to solve, occasionally some problem instances take much longer, and these outliers significantly increase the average time (as well as the standard deviation). The variation is much smaller for the low demand scenarios, which are generally much easier to solve, as can be seen by the significantly lower solving time.

The following observations are consistent for all scenario sets evaluated in this chapter and deserve some highlight:

- The low demand scenarios are significantly more efficient (in the sense of vehicles experiencing lower delays and higher speed) and easier to solve than the high demand scenarios.

- Setting a limit on the time to solve SP3 generally affects only a small portion of problem instances and decreases traffic efficiency by very little, but has a large effect on the average solution time (and significantly decreases solution time standard deviation).

- Overall, vehicles are able to keep average speeds very close to the speed limit, and suffer very small delays. Vehicles controlled by OATS in the high demand scenarios suffer delays more than one order of magnitude smaller than vehicles under a fixed time control would suffer with a low demand[1].

- When delays increase, the standard deviation of delay tends to increase as well. This happens because when traffic conditions become worse, some vehicles are forced to decelerate more, while others are still able to travel basically unimpeded.

4.3.2 Experiments with different values for the longitudinal headway h_L

Overall, results when varying the longitudinal headway are very similar to those varying the transversal headway. The choice of h_L constitutes a compromise between safety and efficiency (just like the choice of h_L), with larger headways increasing vehicle delay and the time it takes to solve a problem instance. Experiments are performed with $h_\mathrm{L} \in \{0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8\}$ seconds, for the three situati-

[1]The only exceptions to this are the scenarios where vehicles effectively cross in a First-In-First-Out order, in which the delay is almost one order of magnitude smaller; or when traffic demand is so high queues start to form

Table 2 – Simulation results varying longitudinal headway h_L

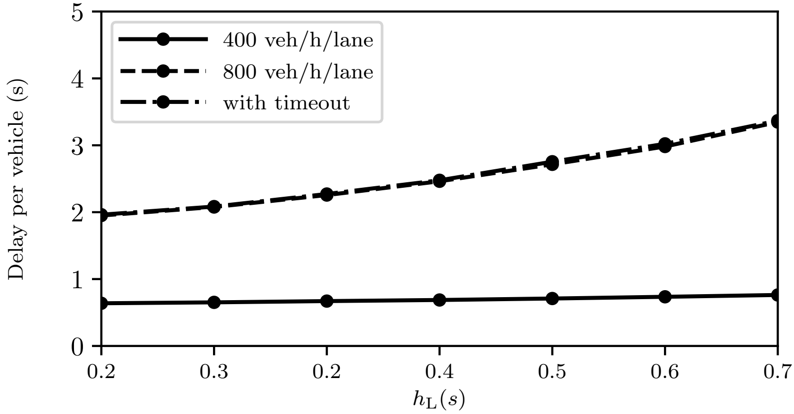
h_L (s)	delay (s/veh)	speed (km/h)	solve time (ms)	% timeout
Scenarios with low demand				
0.2	0.64 ± 1.08	29.36	2.8 ± 2.5	–
0.3	0.65 ± 1.10	29.35	2.4 ± 2.4	–
0.4	0.67 ± 1.12	29.34	2.4 ± 2.5	–
0.5	0.69 ± 1.15	29.33	2.8 ± 2.8	–
0.6	0.71 ± 1.18	29.32	2.7 ± 2.8	–
0.7	0.73 ± 1.23	29.31	2.7 ± 3.0	–
0.8	0.76 ± 1.26	29.29	2.9 ± 3.0	–
Scenarios with high demand				
0.2	1.95 ± 2.20	28.63	22.4 ± 147.0	–
0.3	2.08 ± 2.31	28.56	27.0 ± 205.8	–
0.4	2.26 ± 2.49	28.47	32.6 ± 242.6	–
0.5	2.46 ± 2.67	28.36	39.7 ± 346.6	–
0.6	2.72 ± 2.89	28.23	41.2 ± 238.9	–
0.7	2.98 ± 3.16	28.09	49.3 ± 314.6	–
0.8	3.35 ± 3.43	27.91	77.6 ± 417.4	–
Scenarios with high demand and $T_{max}^{opt} = 0.1$ s				
0.2	1.96 ± 2.21	28.62	14.2 ± 18.9	1.9 ± 1.7
0.3	2.08 ± 2.34	28.56	15.7 ± 20.8	2.4 ± 1.8
0.4	2.27 ± 2.50	28.46	17.6 ± 22.8	3.5 ± 2.5
0.5	2.48 ± 2.68	28.35	19.6 ± 24.0	4.3 ± 1.9
0.6	2.75 ± 2.95	28.21	21.9 ± 26.2	5.5 ± 2.2
0.7	3.02 ± 3.20	28.07	24.3 ± 28.3	7.1 ± 2.9
0.8	3.37 ± 3.41	27.89	27.4 ± 30.5	9.7 ± 4.4

ons mentioned in Section 4.3. Table 2 and Figure 15 show simulation results.

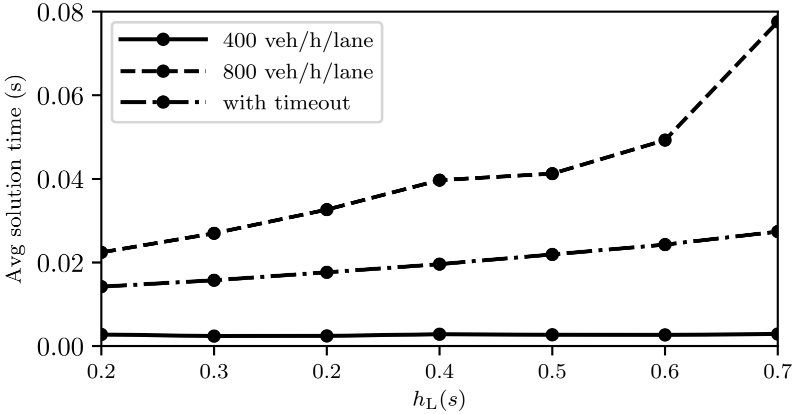
Just like in the experiments varying h_T (and all other experiments), average vehicle speed is very close to the speed limit, specially for the low demand scenario, and delays are relatively small.

Vehicle delay seems to be less sensitive to a variation in longitudinal headway than in transversal headway, possibly due to the fact vehicles are more likely to "interact" with vehicles from different approaches (since, for each vehicle, there are three times more vehicles in a different approach than on the same approach).

Figure 15 – Average vehicle delay (a) and average solution time (b) results for different longitudinal headways h_L



(a)



(b)

4.3.3 Experiments with different values for the control distance D_a

Another aspect that has a clear impact on performance is the size of the CR. Clearly, the more vehicles inside the CR, the harder it is to solve the problem, as more vehicles – and hence more variables – tend to be involved in a given problem instance. On the other hand, by decreasing the CR, vehicles have their trajectory optimized only when they are closer to the IR, and the IC has less “power” to act on vehicles, as they are under its control for less time. This means that, by decreasing the size of the CR traffic conditions should likely get worse.

Another relevant aspect is that it may not be possible to always have large CRs. The problem formulation implies that all the conflict regions inside the CR are accounted for in the model – else it would not be possible to guarantee safety. This includes not only intersections, but also any point of entry or exit of the road network, such as access points for nearby buildings or parking spots. Although it is reasonable to disregard the influence of such places close to an intersection, as the CR increases this may no longer be possible. Two approaches to deal with this problem are to either model all those interactions, or use a sufficiently small CR and disregard vehicle behavior outside it.

The trade-off between solution quality and problem size as a function of the control distance was examined by conducting experiments in which D_{a} is varied between 30 and 100 m. Notice that, given the chosen vehicle parameters, 30 m is still sufficiently large to guarantee SP3 is always feasible.

Table 3 and Figure 16 show simulation results for both low and high demand situations. As could be expected, decreasing D_{a} has an adverse effect on efficiency, both increasing vehicle delay and decreasing average speed. However, it is specially noteworthy that such effect is very small, while the decrease on the time it takes to solve a problem instance is substantial.

For the high demand situation, decreasing the control distance from 100 m to 30 m increased average delay by almost a third of a second, or 13%. Notice that since the average travel time of a non delayed vehicle is close to 50.5 s, this 0.33 s increase in delay actually represents just a 0.62 % increase in total travel time. Meanwhile, the time needed to solve a problem instance decreased by one order of magnitude. In the case where $T_{\mathrm{max}}^{\mathrm{opt}}$ is set to 0.1s, the reduction in the computational time is by (roughly) a factor of 5, which is still very substantial. Notice also that for $D_{\mathrm{a}} \leq 50$ m, timeouts are very unlikely.

Originally, the choice of $D_{\mathrm{a}} = 100$ m as a default value was motivated by the expectation that a large CR would allow for a significantly better control than a small CR, enabling the IR to act much longer on vehicles. However, the results of these experiments show that having a CR much larger than necessary has actually very little benefit to traffic

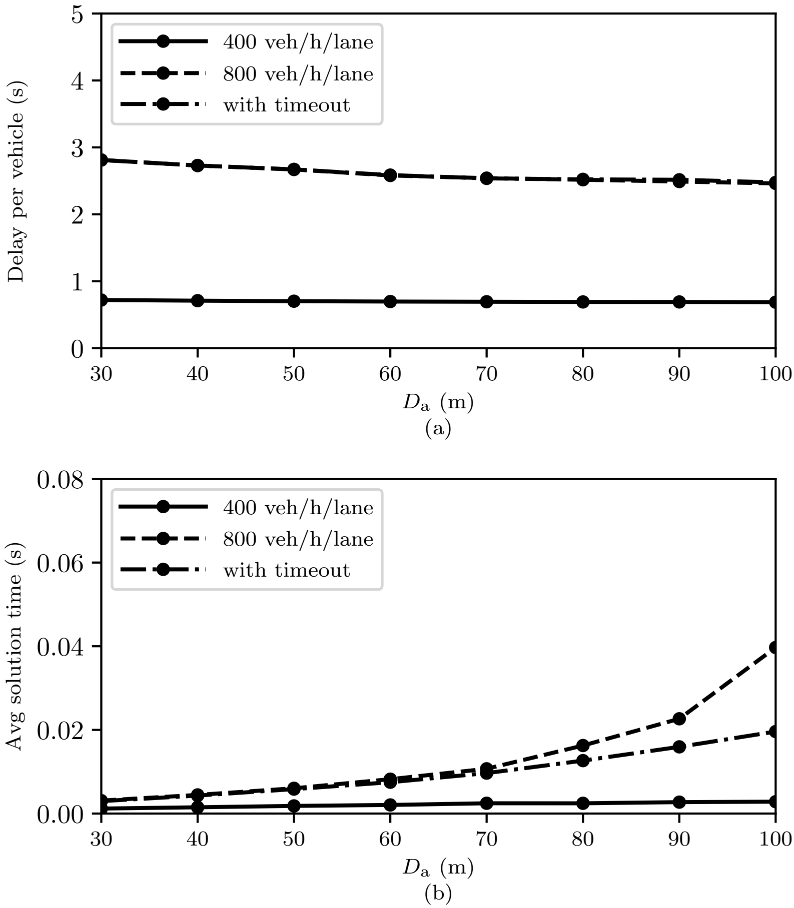
Table 3 – Simulation results for different control distances D_a

D_a (m)	delay (s/veh)	speed (km/h)	solve time (ms)	% timeout
Scenarios with low demand				
100	0.69 ± 1.15	29.33	2.8 ± 2.8	–
90	0.69 ± 1.16	29.33	2.7 ± 2.7	–
80	0.69 ± 1.16	29.33	2.5 ± 2.3	–
70	0.69 ± 1.16	29.33	2.5 ± 2.3	–
60	0.70 ± 1.17	29.33	2.1 ± 2.0	–
50	0.70 ± 1.18	29.33	1.8 ± 1.9	–
40	0.71 ± 1.21	29.32	1.5 ± 1.7	–
30	0.72 ± 1.22	29.32	1.2 ± 1.5	–
Scenarios with high demand				
100	2.46 ± 2.67	28.36	39.7 ± 346.6	–
90	2.49 ± 2.70	28.35	22.7 ± 122.6	–
80	2.52 ± 2.72	28.35	16.3 ± 80.2	–
70	2.54 ± 2.77	28.35	10.7 ± 28.9	–
60	2.58 ± 2.76	28.33	8.2 ± 12.8	–
50	2.67 ± 2.84	28.29	6.0 ± 6.7	–
40	2.73 ± 2.90	28.26	4.5 ± 4.0	–
30	2.81 ± 3.04	28.23	3.1 ± 2.7	–
Scenarios with high demand and $T^\mathrm{opt}_\mathrm{max} = 0.1$ s				
100	2.48 ± 2.68	28.35	19.6 ± 24.0	4.3 ± 1.9
90	2.52 ± 2.75	28.34	15.9 ± 21.1	3.0 ± 1.9
80	2.52 ± 2.73	28.34	12.7 ± 17.0	1.5 ± 1.7
70	2.54 ± 2.76	28.35	9.7 ± 12.9	0.7 ± 1.7
60	2.59 ± 2.76	28.33	7.5 ± 9.5	0.3 ± 0.9
50	2.67 ± 2.84	28.29	5.8 ± 6.5	0.0 ± 0.1
40	2.73 ± 2.90	28.26	4.3 ± 3.9	0.0 ± 0.0
30	2.81 ± 3.04	28.23	3.0 ± 2.6	0.0 ± 0.0

efficiency and a significant impact in computational effort, suggesting that a small CR may be more practical for field implementation.

Notice that the benefit of decreasing D_a is much larger when traffic demand is higher. The reason for this is that, when traffic is light and there are few vehicles, SP3 can generally be solved very fast, even with a large CR, so there is little benefit for reducing D_a (and hence the size of the CR). On the other hand, when traffic is high, decreasing D_a might become more relevant for ensuring SP3 is tractable. One possible method for limiting the problem complexity without needlessly reducing solution quality when demand is low is to have D_a dynamically change with traffic conditions.

Figure 16 – Average vehicle delay (a) and average solution time (b) results for different control distances D_a



4.3.4 Experiments with vehicles being scheduled in batches

OATS as designed executes the optimization process every time a new vehicle enters the CR. This guarantees that every time there is new relevant information about the system, a new solution is found, meaning vehicles can adjust quickly to a new situation. However, this does not need to be the case, as there are other forms to implement OATS.

One possible alternative is to optimize vehicles in batches. Consider that, besides the control distance D_{a}, which could be called maximum control distance, there is also a minimum control distance $D_{\mathrm{a}}^{\mathrm{min}}$, with $D_{\mathrm{a}}^{\mathrm{min}} \leq D_{\mathrm{a}}$. If a vehicle is inside the smaller radius, defined by $D_{\mathrm{a}}^{\mathrm{min}}$, it must be controlled by the IC. If it is beyond the maximum distance D_{a}, it is not controlled by the IC. And if it is located between D_{a} and $D_{\mathrm{a}}^{\mathrm{min}}$, it may or may not be controlled. The following procedure was implemented for this set of experiments, and is executed at every simulation step:

1. Categorize vehicles according to their distance to the IR. Put any vehicle that is closer than $D_{\mathrm{a}}^{\mathrm{min}}$ m to the IR on List 1, the list of vehicles that must be optimized, and any vehicle that is between D_{a} and $D_{\mathrm{a}}^{\mathrm{min}}$ m away from the IR on List 2, the list of vehicles that might be optimized.

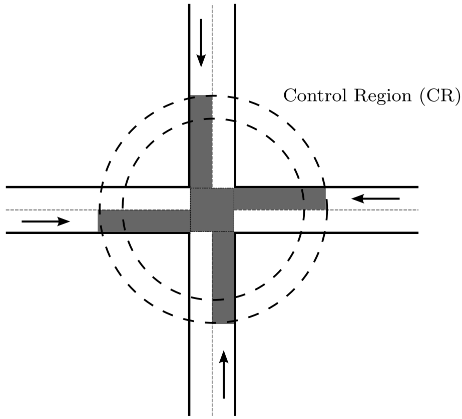
2. Check each vehicle on List 1 to see if they have been already scheduled previously. If there is any vehicle that has never been scheduled, terminate the search and solve SP3 for all vehicles in both List 1 and List 2, defining a schedule for then. Otherwise, do nothing.

The resulting behavior is that each time a vehicle that has never been optimized reaches the inner perimeter defined by $D_{\mathrm{a}}^{\mathrm{min}}$, every vehicle up to the outer perimeter defined by D_{a} is optimized. In short, SP3 is only solved each time a new vehicle crosses the distance between D_{a} and $D_{\mathrm{a}}^{\mathrm{min}}$. Figure 17 illustrates the two control perimeters around the intersection.

This decreases the number of problem instances of SP3 that have to be solved. Experiments were performed in order to test if this has a significant impact in traffic efficiency or computational effort. Results are presented in Table 4 and Figure 18. $D_{\mathrm{a}}^{\mathrm{min}}$ was varied between 30 m and 100 m. Notice that the scenarios where $D_{\mathrm{a}}^{\mathrm{min}} = 100$ m correspond to no batch optimization at all, since in this case $D_{\mathrm{a}}^{\mathrm{min}} = D_a$.

Results indicate that optimizing in batches has a negligible effect on traffic performance, with both vehicle delay and average speed almost constant across the scenarios (aside to the fact that the scenarios with low traffic demand are significantly more efficient).

Figure 17 – An Intersection with two control perimeters. The smaller circle corresponds to $D_{\mathrm{a}}^{\mathrm{min}}$, and the larger circle to D_{a}.



However, solution time increases as $D_{\mathrm{a}}^{\mathrm{min}}$ decreases (i.e., the size of the batch increases). In Figure 18 this is more visible on the time series in which $T_{\mathrm{max}}^{\mathrm{opt}} = 0.1$, as setting a maximum optimization time mitigates the effect of outliers which make results without a timeout more "spiky". Although the computational effort seems relatively flat for the low demand scenarios on the figure, it can be seen on Table 4 that it also increases as the batch size increases for these scenarios.

In short, batch optimization in OATS has a negligible effect on traffic conditions, and a (slightly) detrimental effect on computational effort for each problem instance. However, it also decreases the average number of problems solved, which goes down from 494 when there is no batch at all to 59 when $D_{\mathrm{a}}^{\mathrm{min}} = 30$, for the high demand case. This means that, even though each problem becomes slightly harder, there are much less problems to solve.

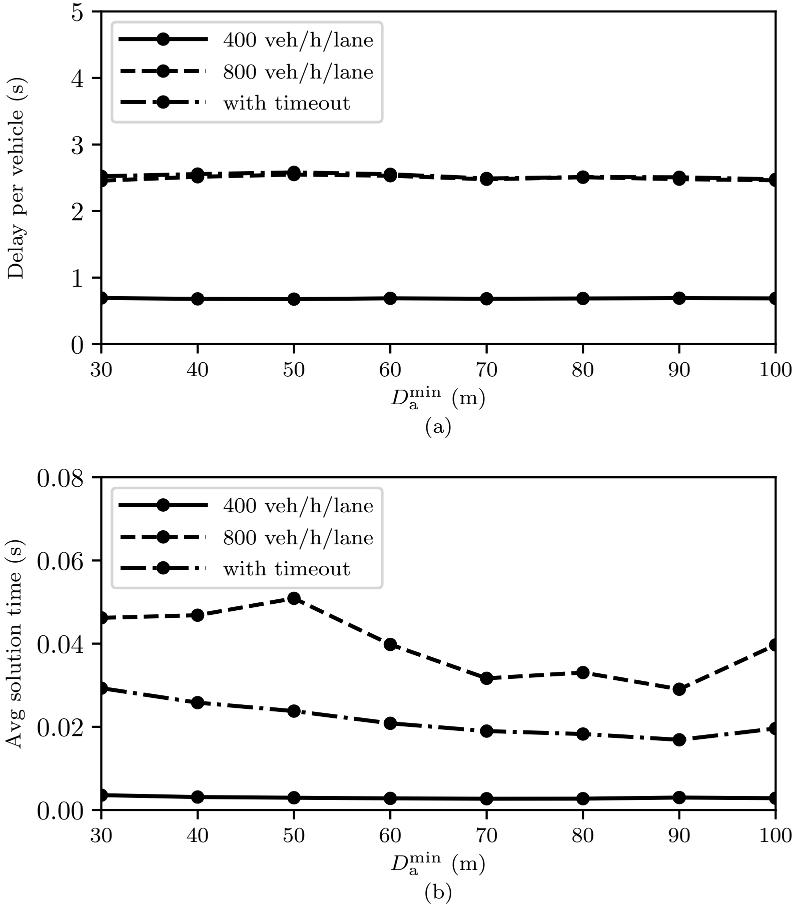
The observed increase in computational effort is not completely understood yet. It is possibly related to the fact that, when vehicles have their trajectories optimized as they enter the CR, they receive a schedule for reaching the IR in minimum time. These schedules often involve maximum acceleration or speed, meaning vehicles quickly approach the IR and, while doing so, have their feasible arrival interval $[t_{a,i}^{\mathrm{min}}, t_{a,i}^{\mathrm{max}}]$ become narrower quickly. Narrower intervals can make the problem easier to solve, as they reduce the search space (note that a

Table 4 – Simulation results for batch optimization, varying minimum control distance $D_{\mathrm{a}}^{\mathrm{min}}$

Min cont dist (s)	delay (s/veh)	speed (km/h)	solve time (ms)	% timeout
Scenarios with low demand				
100	0.69 ± 1.15	29.33	2.8 ± 2.8	–
90	0.69 ± 1.15	29.33	3.0 ± 3.0	–
80	0.69 ± 1.15	29.33	2.7 ± 2.8	–
70	0.68 ± 1.15	29.34	2.7 ± 2.7	–
60	0.69 ± 1.17	29.33	2.8 ± 3.2	–
50	0.68 ± 1.13	29.34	2.9 ± 3.5	–
40	0.68 ± 1.15	29.34	3.1 ± 3.7	–
30	0.69 ± 1.19	29.33	3.6 ± 5.8	–
Scenarios with high demand				
100	2.46 ± 2.67	28.36	39.7 ± 346.6	–
90	2.48 ± 2.65	28.36	29.0 ± 219.1	–
80	2.51 ± 2.74	28.34	33.0 ± 224.3	–
70	2.48 ± 2.72	28.36	31.7 ± 151.4	–
60	2.53 ± 2.82	28.34	39.8 ± 308.6	–
50	2.55 ± 2.82	28.33	50.9 ± 579.8	–
40	2.51 ± 2.77	28.35	46.8 ± 288.4	–
30	2.46 ± 2.71	28.37	46.2 ± 162.1	–
Scenarios with high demand and $T_{\mathrm{max}}^{\mathrm{opt}} = 0.1$ s				
100	2.48 ± 2.68	28.35	19.6 ± 24.0	4.3 ± 1.9
90	2.51 ± 2.68	28.34	16.9 ± 21.9	3.1 ± 1.3
80	2.51 ± 2.74	28.34	18.3 ± 23.5	3.9 ± 2.0
70	2.49 ± 2.74	28.35	19.0 ± 23.8	4.0 ± 1.6
60	2.55 ± 2.82	28.32	20.8 ± 24.7	4.4 ± 1.4
50	2.58 ± 2.82	28.31	23.8 ± 27.6	6.4 ± 2.9
40	2.56 ± 2.83	28.34	25.8 ± 28.5	8.2 ± 4.5
30	2.52 ± 2.88	28.35	29.3 ± 30.7	9.1 ± 4.0

narrow interval might lead to more of the implications discussed in Section 3.6.2 becoming active, meaning it can reduce the number of binary variables). On the other hand, in batch optimization some vehicles may spend a longer amount of time without receiving an optimal plan from the IC, instead following (in this implementation) an ACC algorithm. Hence, the vehicle maintains the same speed as the vehicle ahead, at a certain distance, possibly with a lower speed and larger distance from the vehicle in front than it would have if following an optimal minimum time trajectory. In short, vehicles that follow optimal, minimum time trajectories, may, on average, have their feasible arrival interval become narrower faster than vehicles who that not behave like so, and the narrowing of this interval can, possibly, contribute to making the

Figure 18 – Average vehicle delay (a) and Average solution time (b) results for batch optimization with different minimum control distances D_a^{min}



problem easier to solve.

Given the real time nature of the problem, decreasing how often SP3 has to be solved does not seem very helpful, as it still must be solved quickly when it is actually executed. As such, there appears to be no reason for batch optimization in a practical implementation, as its most relevant effect is increasing computational effort of the average problem instance.

4.3.5 Experiments with vehicles fixing their schedule after an arbitrary point

Recall that in the proposed implementation for OATS, SP3 is repeatedly solved as new vehicles enter the CR, and each time a new solution is found vehicles may receive a new schedule. In a way, this means that vehicles may have their schedule changed to better accommodate other vehicles that were not previously considered.

While this is beneficial from a perspective of minimizing total travel time, it means that some vehicles can be repeatedly delayed, being scheduled further and further as SP3 is solved repeatedly. Unless (3.16) is used as the objective function of SP3 to ensure liveness, as discussed in Section 3.6.5, it is even possible that vehicles are delayed indefinitely.

Another form to ensure liveness is to disallow vehicles to be re-scheduled, causing them to follow the first schedule they ever receive. Since any given solution of SP3 ensures every vehicle is scheduled to reach the IR in a finite amount of time (unless $h_c^{\text{cap}} = \infty$), if vehicles do not change their schedule they eventually cross the IR.

Disallowing re-scheduling also has the benefit of making SP3 significantly easier to solve, as decision variables need to be assigned only to vehicles entering the CR. However, this is equivalent to giving priority to vehicles that enter the CR first, which is not very different from a FIFO coordination strategy. Clearly, not allowing vehicles to be re-scheduled should make traffic less efficient.

It is possible to implement a middle ground, by setting a distance of no rescheduling D_{NR}, with $D_{\text{NR}} \leq D_a$ (in fact, $D_{\text{NR}} \leq D_{\text{a}}^{\text{min}} \leq D_a$ if such a configuration is used together with batch scheduling). Any vehicle i that is closer to the intersection than this distance (i.e, $d_i \leq D_{\text{NR}}$) is not scheduled again. When SP3 is executed, those vehicles are only used to model constraints for other vehicles.

Simulations were performed to investigate the effect D_{NR} has on traffic conditions and computational effort to solve SP3. D_{NR} was varied between 0 and 100 m. Notice $D_{\text{NR}} = 100$ m is equivalent to only optimizing each vehicle once as they enter the CR, which is not much different from a FIFO crossing order, and $D_{\mathrm{NR}} = 0$ m is equivalent to always allowing vehicles to be rescheduled.

Table 5 and Figure 19 show the simulation results. As expected, traffic conditions deteriorate as D_{NR} increases. The effect is much more pronounced for the high demand scenarios, with vehicle delay more than doubling when D_{NR} is increased from 80 to 100 m (the value of delay for 100 m, 10.51 s, is left out of Figure 19 for scale reasons)

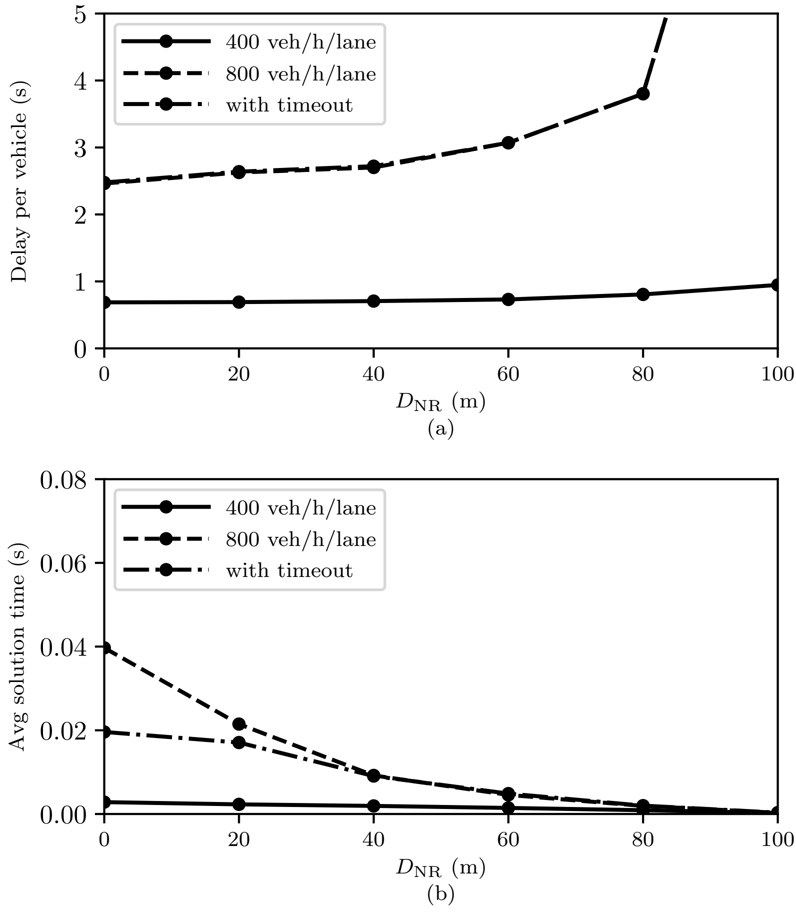
This sudden increase happens because when D_{NR} is finally increased to $D_{\mathrm{NR}} = D_a$, the IC loses almost any ability to delay a vehicle in benefit of another, effectively just prioritizing vehicles in the order they enter the CR. Interestingly, a relatively small D_{NR}, of up to 40 m, has a very small effect on traffic conditions, only increasing vehicle delay by proximately 0.25 seconds per vehicle in relation to the scenario in which $D_{\mathrm{NR}} = 0$ m. This represents an increase of approximately 10 % in vehicle delay and 0.5 % in total travel time inside the modeled region.

Average time to solve SP3 decreases significantly when increasing

Table 5 – Simulation results varying the distance of no rescheduling, D_{NR}

D_{NR}	delay (s/veh)	speed (km/h)	solve time (ms)	% timeout
Scenarios with low demand				
0	0.69 ± 1.15	29.33	2.8 ± 2.8	–
20	0.69 ± 1.16	29.33	2.3 ± 2.3	–
40	0.71 ± 1.20	29.32	1.9 ± 2.0	–
60	0.73 ± 1.24	29.31	1.4 ± 1.6	–
80	0.81 ± 1.29	29.27	0.9 ± 1.1	–
100	0.95 ± 1.43	29.19	0.2 ± 0.3	–
Scenarios with high demand				
0	2.46 ± 2.67	28.36	39.7 ± 346.6	–
20	2.63 ± 2.84	28.28	21.5 ± 69.4	–
40	2.70 ± 2.83	28.23	9.2 ± 12.7	–
60	3.07 ± 2.98	28.05	4.6 ± 4.6	–
80	3.80 ± 3.14	27.65	1.9 ± 2.1	–
100	10.51 ± 8.18	24.80	0.3 ± 0.2	–
Scenarios with high demand and $T_{\max}^{\mathrm{opt}} = 0.1$ s				
0	2.48 ± 2.68	28.35	19.6 ± 24.0	4.3 ± 1.9
20	2.64 ± 2.85	28.27	17.1 ± 21.6	2.8 ± 1.5
40	2.72 ± 2.86	28.22	9.1 ± 11.3	0.2 ± 0.3
60	3.07 ± 2.98	28.05	4.9 ± 4.8	0.0 ± 0.0
80	3.80 ± 3.14	27.65	2.0 ± 2.0	0.0 ± 0.0
100	10.51 ± 8.18	24.80	0.3 ± 0.3	0.0 ± 0.0

Figure 19 – Average vehicle delay (a) and Average solution time (b) results for different distances of no rescheduling D_{NR}



D_{NR} due to the fact that this decreases the problem size. As could be expected, when $D_{\mathrm{NR}} = 100$ m SP3 can be solved very fast, as most of the time only one vehicle is considered at a time. Increasing D_{NR} from 0 m to 40 m decreases the solution time by roughly 50% for the high demand scenarios with $T_{\max}^{\text{opt}} = 0.1$ s. It can also be seen that for $D_{\text{NR}} \geq 40$ m, having the solver timing out becomes very rare.

Results show that having a relatively small distance of no rescheduling, with $D_{\text{NR}} \leq 40$ m can make the problem significant easier to solve while impacting traffic efficiency very little. However, reducing the control distance, as discussed in Section 4.3.3 seems to be more beneficial in this regard. For instance, considering the high demand situation with $T_{\max}^{\text{opt}} = 0.1$ s, the scenario in which $D_{\text{NR}} = 40$ m (and $D_a = 100$ m) has an average vehicle delay of approximately 2.7 s and solution time of 9.1 ms. The scenario in which $D_a = 40$ m (and $D_{\text{NR}} = 0$ m) is roughly equivalent in terms of vehicle delay, but solution time is roughly half. And the scenario in which $D_a = 70$ m (and $D_{\text{NR}} = 0$ m) is roughly equivalent in terms of solution time, but has a smaller delay.

All things considered, it seems that adopting a large control distance and setting a distance of no rescheduling is a worse approach to simplifying SP3 than simply reducing the control distance, although these two simplifications could be used together.

4.3.6 Experiments with a limited number of vehicles inside the CR

Both reducing the control distance or setting a distance of no rescheduling have the effect of reducing the number of vehicles that are considered when SP3 is solved. This makes the problem easier to solve, at the cost of some reduction in traffic efficiency. In both cases, this reduction in problem size comes by effectively delimiting a smaller area for the CR.

Another possible approach for dealing with a large number of vehicles is to more directly limit how many vehicles are considered by SP3, instead of limiting the area of control. One possible approach to implement this, explored in this section, is to set a limit on the number of vehicles considered in any given problem instance.

There are multiple ways to implement such a limit. The approach chosen is to set a value, N_{U}, for the maximum number of vehicles with unbounded maximum arrival time ($t_{a,i}^{\max}$) per approach to be included in any given problem instance. For instance, if $N_{\text{U}} = 3$ veh, for each approach, the three first vehicles (ordered by proximity to the IR) that have $t_{a,i}^{\max} = \infty$ (or actually $t_{a,i}^{\max} \geq 120$ s in the simulations performed) are included in SP3, and any vehicle upstream is disregarded, to only be considered in future problem instances. This has the effect of roughly limiting the problem size by the number of vehicles, independently of the density of vehicles in the CR.

As an aditional measure, vehicles that are closer than 30 m to the IR are never excluded from SP3, no matter the value of N_{U}. This guarantees vehicles very close to the intersection are always taken into account by SP3, and SP3 is always feasible.

To investigate the effect that setting this limit has, experiments were conducted with N_{U} varying between 0 and 10, and also with no limit for N_{U} at all. Table 6 and Figure 20 show simulation results.

Overall, implementing a limit on N_{U} has, as expected, an adverse effect on traffic conditions, and a beneficial effect for the time needed to solve SP3. These effects get more pronounced as N_{U} decreases. Vehicle delay increases roughly by 0.33 s when $N_{\mathrm{U}} = 0$ veh in the high demand scenario, while the time it takes to solve SP3 decreases by a factor of 6 (for the case where $T_{\mathrm{max}}^{\mathrm{opt}} = 0.1$ s).

Since vehicles closer than 30 m to the IR are set no never be disregarded from SP3, and vehicles beyond this distance always have an unbounded maximum time, the scenarios in which $N_{\mathrm{U}} = 0$ veh are actually equivalent to the scenarios in which $D_a = 30$ m.

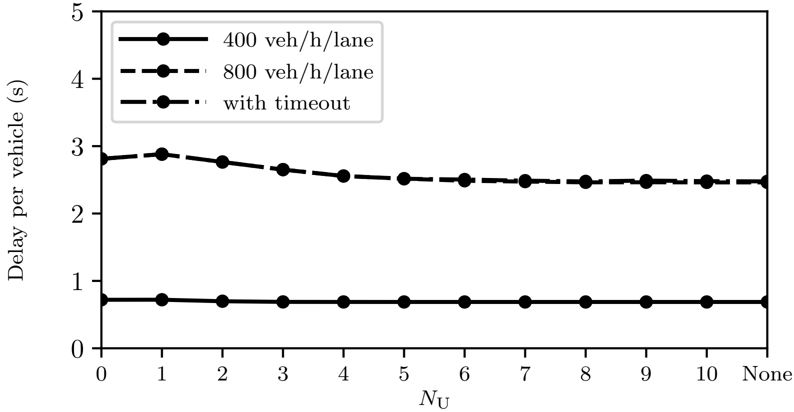
Although this applies to every scenario, In Figure 20 it is particularly evident that setting $T_{\mathrm{max}}^{\mathrm{opt}} = 0.1$ s has a very small effect on traffic performance, and a significant effect on the average solution time.

The small increase in delay and large reduction in solving time when N_{U} is small is similar to the results obtained for small control distances. This suggests that if one wishes to keep the problem size small, the simplification of setting a limit to N_{U} might be as useful as reducing the control distance.

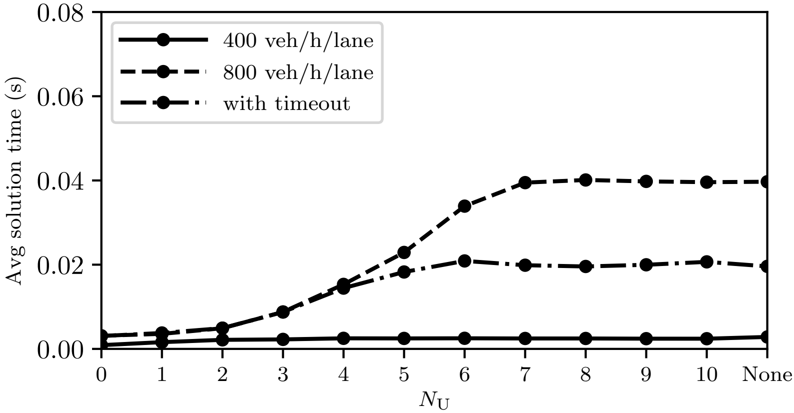
Table 6 – Simulation results varying the maximum number of vehicles with an unbounded maximum arrival time, N_{U}

N_{U}	delay (s/veh)	speed (km/h)	solve time (ms)	% timeout
Scenarios with low demand				
0	0.72 ± 1.22	29.32	0.9 ± 1.1	–
1	0.72 ± 1.22	29.32	1.6 ± 1.2	–
2	0.70 ± 1.16	29.33	2.2 ± 1.8	–
3	0.69 ± 1.16	29.33	2.3 ± 2.0	–
4	0.69 ± 1.15	29.33	2.5 ± 2.4	–
5	0.69 ± 1.15	29.33	2.5 ± 2.5	–
6	0.69 ± 1.15	29.33	2.5 ± 2.5	–
7	0.69 ± 1.15	29.33	2.5 ± 2.4	–
8	0.69 ± 1.15	29.33	2.5 ± 2.4	–
9	0.69 ± 1.15	29.33	2.4 ± 2.4	–
10	0.69 ± 1.15	29.33	2.4 ± 2.4	–
None	0.69 ± 1.15	29.33	2.8 ± 2.8	–
Scenarios with high demand				
0	2.81 ± 3.04	28.23	3.1 ± 3.0	–
1	2.88 ± 3.05	28.19	3.6 ± 2.4	–
2	2.76 ± 3.01	28.24	4.9 ± 3.3	–
3	2.65 ± 2.99	28.29	8.8 ± 8.2	–
4	2.56 ± 2.82	28.32	15.3 ± 31.5	–
5	2.52 ± 2.74	28.33	22.9 ± 79.7	–
6	2.49 ± 2.69	28.35	33.9 ± 258.0	–
7	2.47 ± 2.69	28.35	39.5 ± 341.5	–
8	2.46 ± 2.67	28.36	40.1 ± 344.0	–
9	2.46 ± 2.67	28.36	39.8 ± 349.6	–
10	2.46 ± 2.67	28.36	39.6 ± 346.9	–
None	2.46 ± 2.67	28.36	39.7 ± 346.6	–
Scenarios with high demand and $T_{\max}^{\mathrm{opt}} = 0.1$ s				
0	2.81 ± 3.04	28.23	3.1 ± 2.7	0.0 ± 0.0
1	2.88 ± 3.05	28.19	3.8 ± 2.5	0.0 ± 0.0
2	2.76 ± 3.01	28.24	4.9 ± 3.3	0.0 ± 0.0
3	2.65 ± 2.99	28.29	8.8 ± 8.1	0.0 ± 0.0
4	2.56 ± 2.85	28.33	14.4 ± 17.5	1.1 ± 0.8
5	2.52 ± 2.75	28.32	18.3 ± 22.5	3.1 ± 1.5
6	2.50 ± 2.72	28.34	20.9 ± 25.1	4.7 ± 1.7
7	2.49 ± 2.67	28.34	19.9 ± 24.5	4.4 ± 1.9
8	2.48 ± 2.68	28.35	19.6 ± 23.9	4.0 ± 1.7
9	2.49 ± 2.69	28.34	20.0 ± 24.3	4.3 ± 1.9
10	2.48 ± 2.67	28.35	20.7 ± 24.6	4.6 ± 2.3
None	2.48 ± 2.68	28.35	19.6 ± 24.0	4.3 ± 1.9

Figure 20 – Average vehicle delay (a) and Average solution time (b) results for different limits on the number of vehicles with unbounded maximum arrival time per approach, N_{U}



(a)



(b)

4.3.7 Experiments with approximated arrival times for vehicles with tight arrival intervals

As a vehicle approaches the IR, its feasible arrival interval [$t_{a,i}^{\min}$, $t_{a,i}^{\max}$] gets smaller. When this interval is very narrow, optimizing the arrival time of such vehicle may have little practical effect. This motivated Simplification 4 discussed in Section 3.6.3 in which vehicles with an arrival interval that is smaller than ξ^{SP2} have their arrival time set to the average of $t_{a,i}^{\min}$ and $t_{a,i}^{\max}$ instead of being included as a decision variable in SP3. The reasoning behind this modeling decision is the expectation that it could make SP3 smaller (by removing some vehicles from the problem) while having very little impact on the schedule, as it only affects vehicles for which the IC has very little margin to control.

Simulation results were performed to test the effect of this simplification, varying ξ^{SP2} between 0 and 0.5 s. Note that $\xi^{\mathrm{SP2}} = 0$ is equivalent to not implementing this simplification at all. Table 7 and

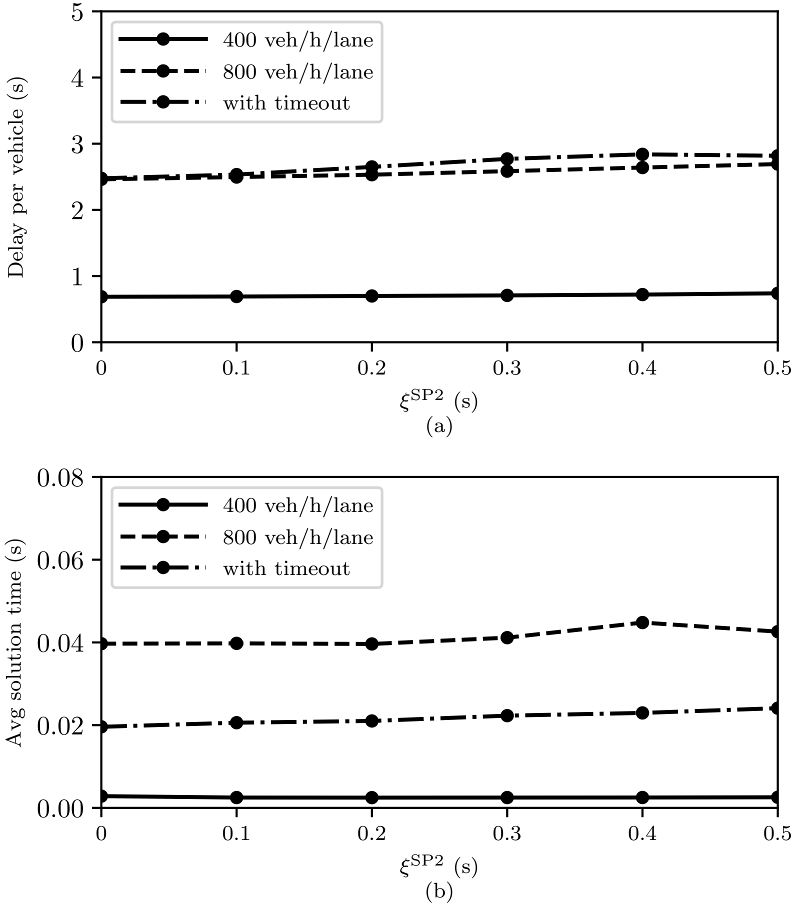
Table 7 – Simulation results varying the minimum size of the arrival interval, ξ^{SP2}

ξ^{SP2}	delay (s/veh)	speed (km/h)	solve time (ms)	% timeout
Scenarios with low demand				
0	0.69 ± 1.15	29.33	2.8 ± 2.8	–
0.1	0.69 ± 1.15	29.33	2.5 ± 2.4	–
0.2	0.70 ± 1.16	29.32	2.5 ± 2.4	–
0.3	0.71 ± 1.16	29.32	2.5 ± 2.4	–
0.4	0.72 ± 1.17	29.31	2.5 ± 2.4	–
0.5	0.74 ± 1.17	29.30	2.6 ± 2.6	–
High Demand				
0	2.46 ± 2.67	28.36	39.7 ± 346.6	–
0.1	2.50 ± 2.67	28.34	39.8 ± 304.1	–
0.2	2.53 ± 2.70	28.32	39.6 ± 261.7	–
0.3	2.59 ± 2.74	28.29	41.1 ± 294.7	–
0.4	2.64 ± 2.79	28.26	44.8 ± 318.2	–
0.5	2.69 ± 2.82	28.24	42.6 ± 333.4	–
High Demand and $T_{\max}^{\mathrm{opt}} = 0.1$ s				
0	2.48 ± 2.68	28.35	19.6 ± 24.0	4.3 ± 1.9
0.1	2.53 ± 2.72	28.32	20.6 ± 25.3	4.8 ± 2.1
0.2	2.65 ± 3.03	28.27	21.0 ± 25.6	5.3 ± 2.6
0.3	2.77 ± 3.31	28.22	22.3 ± 26.3	5.6 ± 2.9
0.4	2.84 ± 3.33	28.19	23.0 ± 27.3	6.3 ± 3.1
0.5	2.82 ± 3.16	28.18	24.1 ± 27.7	7.0 ± 3.3

Figure 21 show simulation results for different values of ξ^{SP2}.

As expected, setting a limit on the allowed size of the arrival interval has an adverse effect on traffic conditions, with delay increasing

Figure 21 – Average vehicle delay (a) and average solution time (b) results for different values for the minimum size of the arrival interval, ξ^{SP2}



(and speed decreasing) as ξ^{SP2} gets larger. This is a result of the fact that some vehicles no longer have their arrival times being optimized.

On the other hand, differently from what was expected, results also show that increasing ξ^{SP2} has an adverse effect on solution time. One possible explanation is the fact that by setting vehicles to arrive at the middle of the feasible arrival interval, they may suddenly decelerate (or accelerate) and change their speed profile in such a way that adversely affects surrounding vehicles.

Given that, as implemented, approximating the arrival time has no benefit for either traffic conditions or solution time, this simplification seems inadequate for a practical implementation.

4.3.8 Experiments with different traffic demands

The last set of experiments in this chapter aims to investigate how traffic demand affects vehicle delay and speed, as well as the computational effort necessary to solve SP3.

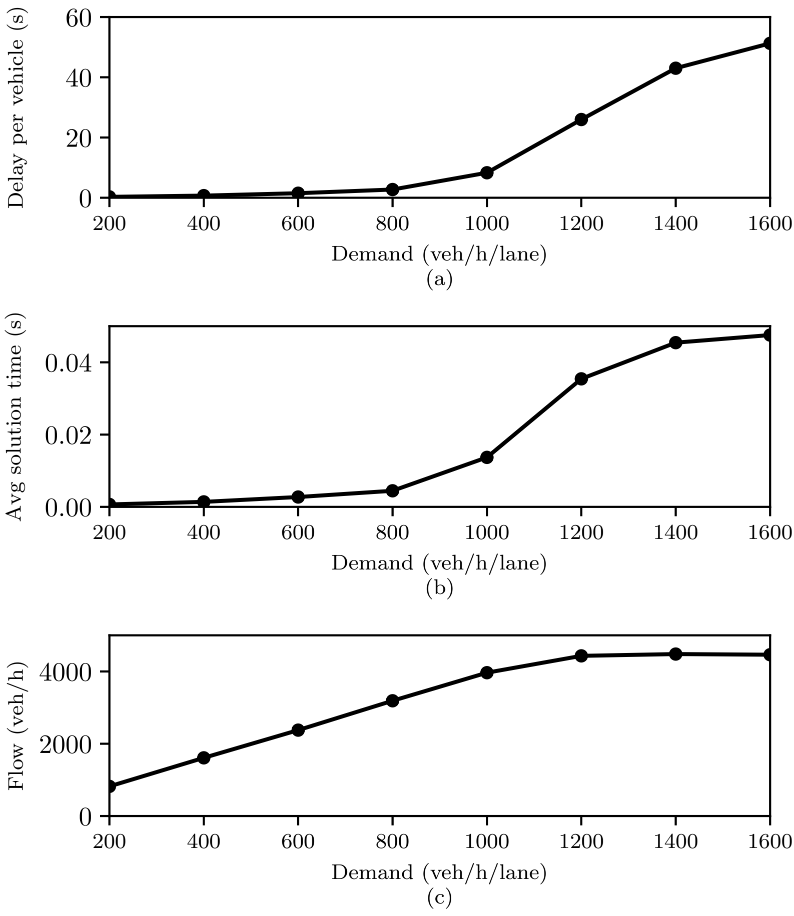
Traffic demand was varied between 200 and 1600 veh/h/lane. Besides the other performance indexes discussed, outflow from the intersection was also measured. The combined outflow from all four intersection exits is showed, together with other results, in Table 8 and Figure 22. No $T_{\max}^{\text{opt}}$ was set. Control distance was set to $D_a = 40$ m which, as discussed in Section 4.3.3, has a slightly adverse effect on vehicle delay, but keeps the solution time relatively small, which is relevant for the scenarios with very high demand.

If there is no queuing up, total outflow should be roughly four times greater than the demand per approach (recall vehicle entrance is

Table 8 – Simulation results for different traffic demands

Demand (veh/h/lane)	outflow (veh/h)	delay (s/veh)	speed (km/h)	solving time (ms)
200	820.5	0.30 ± 0.66	29.57	0.7 ± 0.9
400	1610.25	0.71 ± 1.21	29.32	1.4 ± 1.5
600	2376.75	1.50 ± 2.00	28.92	2.7 ± 2.5
800	3189	2.73 ± 2.90	28.26	4.5 ± 4.2
1000	3966	8.30 ± 9.04	25.86	13.7 ± 15.9
1200	4431.75	25.97 ± 19.66	19.69	35.4 ± 45.9
1400	4481.25	43.01 ± 25.49	15.87	45.4 ± 48.2
1600	4463.25	51.27 ± 25.54	14.44	47.5 ± 64.0

Figure 22 – Average vehicle delay (a), average solution time (b) and (c) flow results for different traffic demands



randomly generated, so the number of vehicles that actually enter the network is not exactly equal to the demand). As can be seen in Table 8, outflow is indeed roughly four times the demand up to a demand of 1000 veh/h/lane, but saturates near 4460 veh/h for demands higher than that, indicating the formation of queues when demand is 1200 veh/h/lane or more.

This suggests that the capacity of OATS under the tested conditions is close to 4460 veh/h, or about 1100 veh/h/lane, which is much more than could be achieved with a traffic lights for this intersection layout (roughly 2.5 to 3 times more).

Vehicle delay increases with demand. Delays are very short (less than 3 seconds) for demands of up to 800 veh/h/lane, as vehicles usually only have to decelerate briefly to be able to cross the IR. Delay sharply increases to 8.3 seconds when demand is 1000 veh/h/lane, and average speed decreases to just under 26 km/h. This suggest that even though the intersection can still service this demand (as the outflow is as large as the demand), vehicles often have stop completely or at least decelerate significantly. Beyond that, for demands of 1200 veh/h/lane and more, average delay and speed have little meaning. Since these demands are higher than capacity, queues start to form and grow indefinitely, so the values for delay and speed become worse as the simulation continues (i.e., if the simulation was executed for one hour instead of 10 minutes, delay and speed values for these saturated scenarios would be much worse).

Average time for solving SP3 also increases with demand, most significantly when demand goes beyond 800 veh/h/lane, and specially for the oversaturated scenarios, as there is always a large amount of vehicles inside the CR when queues form.

All things considered, OATS performs very well in the considered scenario for demands of up to 1000 veh/h/lane, and, even though queues start forming and delays increase significantly as demand reaches capacity, OATS still provides a very large outflow even for the oversaturated scenarios, at a manageable computational effort. For demands comparable to the capacity of an intersection controlled by traffic lights (400 veh/h/lane) OATS has very good performance and can be solved in very little time for all conditions tested.

5 ACCOUNTING FOR ENERGY EFFICIENCY ON THE MOTION PLANNING PROBLEM

The formulation for SP4 presented in Section 3.7, and used for all experiments on Chapter 4 finds a sequence of vehicle states that, if followed, guarantee that vehicles can keep their scheduled arrival time as best as possible. This is achieved by minimizing deviation from the schedule on the cost function of SP4.

So far, metrics such as energy, fuel efficiency or emissions have not been taken into account. In fact, due to the approach of decomposing the overall problem, SP3 does not take any metric other than time into account.

It is, however, possible to use different formulations for the motion planning problem in SP4, accounting for some form of energy efficiency. This chapter discusses possible alternative formulations for SP4 in Section 5.1, and presents related simulation results in Section 5.2. Section 5.3 provides some brief remarks about the simulation results.

5.1 ALTERNATIVE FORMULATIONS FOR THE MOTION PLANNING PROBLEM

Three different strategies are proposed and investigated for the motion planning problem described in Section 3.7. All three formulations consist of discrete optimization problems. In each strategy a combination of energy consumption and adherence to the schedule is optimized.

The main difference between the formulations is how strict they are about keeping the schedule. The motivation for their design is to evaluate how relaxing the adherence to the schedule affects energy consumption. Consider the following hypothesis: a formulation that is very strict on keeping the schedule results in little freedom to optimize other criteria, and hence should probably result in poor energy optimization. On the other hand, a formulation in which the requirement of keeping the schedule is relaxed, should have more freedom to optimize other criteria, and could possibly result in less energy consumption.

Thus, it was expected that less strict strategies would be able to better optimize energy expenditure, possibly at the cost of increased travel times (as the schedule is relaxed). Strategies with different levels of adherence to the schedule were formulated, and this trade-off was evaluated in simulation.

On all three formulations the summation of the squared accelerations of vehicles over the entire control horizon is used as a measure of energy expenditure.

The first alternative formulation for SP4, called SP4_{A1}, is similar to the formulation presented in Section 3.7, except for the fact that it has a different cost function which includes a term that minimizes energy. SP4_{A1} employs a single time step with variable duration for each vehicle, which allows the modeling of the vehicle state at the exact scheduled time of arrival. This strategy is able to obtain trajectories that follow the schedule very closely. This formulation, however, does not allow the modeling of a flexible arrival time.

In the second alternative formulation, SP4_{A2}, vehicle state at the exact scheduled time is no longer modeled. A fixed time step is adopted instead, and a flexible arrival time is accounted for in the model. The scheduled arrival time for a vehicle translates into an arrival at a desired time interval. Vehicles are allowed to deviate from the scheduled arrival time by a given number of intervals. By employing a flexible arrival time, this formulation should have more "freedom" to optimize energy consumption. The downside is that the more flexible arrival times can be safe only if the schedule accounts for the possibility of such deviations, which ultimately means using larger headways for SP3.

In the third alternative formulation, SP4_{A3}, the solution of SP3 is no longer enforced and vehicles are allowed to arrive at any feasible time. Safety is guaranteed by keeping the arrival order implied by the solution of SP3, and safety constraints included in SP4_{A3}. Vehicle trajectories are optimized sequentially, each vehicle using the trajectories of previous potentially conflicting vehicles as constraints. This formulation allows for much flexibility on vehicle arrival times, but can lead to a globally suboptimal solution and greater travel times.

5.1.1 Alternative formulation 1 - Arrival at the scheduled time

Consider SP4 as presented in Section 3.7. SP4_{A1} differs from that formulation in that instead of the cost function F defined by (3.29),

$SP4_{A1}$ uses the cost function F_1:

$$F_1 = \quad W_{\mathrm{a}} \sum_{i=1}^{n_a} \sum_{k=1}^{n_i} (a_{i,k})^2 \frac{\tau_{i,k}}{t_{a,i}} + W_{\mathrm{d}} \sum_{i=1}^{n_a} (d_{i,n_i})^2 + W_{\mathrm{v}} \sum_{i=1}^{n_a} (v_{i,n_i} - v_i^{\mathrm{in}})^2 \tag{5.1}$$

with W_{a}, W_{d} and W_{v} weights on the acceleration (energy), distance and speed components, respectively. The three terms of F_1 correspond, respectively, to the sums of the squared accelerations for each time interval (which approximates energy expenditure), the distance to the intersection at the last interval, and the deviation from the desired speed at the last interval. The acceleration term is normalized by $\frac{\tau_{i,k}}{t_{a,i}}$, i.e., according to the duration of each interval to obtain a value with a similar order of magnitude as the other terms, which facilitates tunning the weight coefficients. Note that the only differences between F and F_1 are the presence of the acceleration term and weight coefficients in F_1.

Formally, $SP4_{A1}$ consists of minimizing F_1 (i.e., 5.1), subject to (3.25)–(3.28).

5.1.2 Alternative formulation 2 - Arrival in the vicinity of the scheduled time

$SP4_{A2}$ is built on top of $SP4_{A1}$. For $SP4_{A2}$, however, $\tau_{i,k} = T_{\mathrm{S}}$ for all k, i.e., every time interval has the same duration. Vehicles are allowed to arrive earlier or later than the scheduled arrival time $t_{a,i}$. The deviation must be within α intervals of the time interval corresponding to $t_{a,i}$, which is denoted as k_i^{t}, the target interval. Because of this, the control horizon must have at least α additional time intervals exceeding k_i^{t}. Therefore, the number of intervals n_i considered for each vehicle i is given by

$$n_i = \left\lceil \frac{t_{a,i}}{T_{\mathrm{S}}} \right\rceil + \alpha. \tag{5.2}$$

Constraints (3.25) and (3.28) remain unchanged, except for the redefinition of $\tau_{i,k}$ and n_i in this section. Final conditions, however, are reformulated. Instead of (3.26a) for vehicle position, the following constraints are used:

$$d_{i,k_i^{\mathrm{t}}-\alpha-1} + \varepsilon^{\mathrm{d}} \geq 0, \tag{5.3a}$$
$$d_{i,n_i} - \varepsilon^{\mathrm{d}} \leq 0. \tag{5.3b}$$

which state that a vehicle must not have reached the intersection on the time interval just before the first allowed arrival interval, and must have reached the intersection by the last allowed interval. A vehicle is said to have reached the intersection if its distance to the IR is smaller than ε^{d}.

In addition, let $\mu_{i,k}$ be a binary variable associated with vehicle i and time interval k. If vehicle i has not reached the intersection by the end of interval k, then $\mu_{i,k} = 1$. Otherwise, $\mu_{i,k} = 0$. This is formalized with constraints

$$d_{i,k} - \varepsilon^{d,\mathrm{SP4}} \leq \mu_{i,k} \cdot Q_1, \tag{5.4a}$$
$$d_{i,k} - \varepsilon^{d,\mathrm{SP4}} \geq (\mu_{i,k} - 1) \cdot Q_1, \tag{5.4b}$$
$$k = 1, \ldots, n_i;\ i = 1, \ldots, n_a.$$

The following constraints for vehicle speed are employed instead of constraints (3.26b):

$$v_{i,k} \leq v_i^{\mathrm{in}} + \varepsilon^{v,\mathrm{SP4}} + (1 - \mu_{i,k-1} + \mu_{i,k}) \cdot Q_2, \tag{5.5a}$$
$$v_{i,k} \geq v_i^{\mathrm{in}} - \varepsilon^{v,\mathrm{SP4}} - (1 - \mu_{i,k-1} + \mu_{i,k}) \cdot Q_2, \tag{5.5b}$$
$$k = 1, \ldots, n_i;\ i = 1, \ldots, n_a$$

which guarantee vehicle speed is close to v_i^{in} at the end of the interval during which the vehicle enters the intersection. By definition $\mu_{i,0} = 1$, and Q_1 and Q_2 are sufficiently large constants.

Let $t'_{a,i}$ be an approximated time of arrival of vehicle i at the intersection taken as the middle of the arrival time step, i.e.:

$$t'_{a,i} = \Big[\sum_{k=1}^{n_i} \mu_{i,k} + 0.5 \Big] \cdot T_{\mathrm{S}}. \tag{5.6}$$

Finally, let F_2 be the cost funciton of $\mathrm{SP4}_{\mathrm{A2}}$:

$$\begin{aligned} F_2 = \quad & W_{\mathrm{a}} \sum_{i=1}^{n_a} \sum_{k=1}^{n_i} (a_{i,k})^2 \frac{1}{n_i} + W_{\mathrm{t}} \sum_{i=1}^{n_a} (t'_{a,i} - t_{a,i})^2 + \\ & W_{\mathrm{v}} \sum_{i=1}^{n_a} \sum_{k=n_i-2\cdot\alpha}^{n_i} (v_{i,k} - v_i^{\mathrm{in}})^2 \frac{1}{2\cdot\alpha+1} \end{aligned} \tag{5.7}$$

with W_{t} the weight coefficient for the term related to the arrival time error. Coefficients $\frac{1}{n_i}$ and $\frac{1}{2\cdot\alpha+1}$ on the first and last terms, respectively, normalize the terms to similar orders of magnitude. The first term, similar to the one used in (5.1), approximates energy expenditure, the second term minimizes the squared deviation from the desired arrival time, and the last term minimizes deviation from v_i^{in} for each time step during which crossing the intersection is feasible.

Formally, $\mathrm{SP4}_{\mathrm{A2}}$ consists of minimizing F_2 (i.e., (5.7)), subject to (3.25), (3.27), (3.28), (5.3)–(5.6).

5.1.3 Alternative formulation 3 - Optimizing vehicles sequentially

$\mathrm{SP4}_{\mathrm{A3}}$ is built on top of $\mathrm{SP4}_{\mathrm{A2}}$. For $\mathrm{SP4}_{\mathrm{A3}}$, however, instead of following the schedule, safety is guaranteed by keeping vehicle arrival times sufficiently apart according to safety headways, and vehicles are optimized sequentially in the arrival order implied by the solution of SP3. The states/inputs obtained for each vehicle as a result of the optimization process, as well as the safety headways, are used as constraints for subsequent vehicles.

It is assumed to not be possible for vehicles to arrive safely earlier than the scheduled times $t_{a,i}$. A sufficiently long control horizon is used for vehicles to be able to possibly arrive much later than originally scheduled, i.e., $n_i > \frac{t_{a,i}}{T_{\mathrm{S}}}$.

Vehicle position is constrained by (5.3b), but not (5.3a), guaranteeing vehicles reach the intersection by the end of the control horizon. Constraints (5.4) and (5.5) are used to model vehicle arrival interval and vehicle speed when reaching the IR.

Consider the set A of lanes approaching the intersection. Recall that, for each vehicle pair i, j from approaches a and b with potentially conflicting movements at the intersection, the minimum headway $h_{a,i,b,j}$ is the minimum time interval between them that enables entering the intersection safely, assuming they arrive with speed v_i^{in} and follow their desired speed profile thereafter. If there is no potential conflict between a vehicle pair, then $h_{a,i,b,j} = -\infty$.

Let $t^{\text{low}}_{a,i}$ be a lower bound for the arrival time of vehicle i from approach a at the intersection defined by

$$t^{\text{low}}_{a,i} = \sum_{k=1}^{n_i} \mu_{i,k} \cdot T_{\text{S}} \tag{5.8}$$

and let $t^{\text{high}}_{a,i}$ be upper bound defined by

$$t^{\text{high}}_{a,i} = t^{\text{low}}_{a,i} + T_{\text{S}}. \tag{5.9}$$

The earliest possible time $t^{\text{safe}}_{a,i}$ for a safe arrival of vehicle i at the IR, considering the known upper bounds for the arrival times of the previous vehicles and the minimum headways between them and the current vehicle is given by:

$$\begin{aligned} &t^{\text{safe}}_{a,i} = \max(t^{\text{high}}_{j} + h_{b,j,a,i}), \\ &a \in A;\ b \in A;\ i = 1, \ldots, n_a;\ j \in P_i. \end{aligned} \tag{5.10}$$

with P_i the set of vehicles preceding vehicle i in the arrival order. Notice $t^{\text{safe}}_{a,i} \geq t_{a,i}$, as $t^{\text{safe}}_{a,i}$ takes into account the (possibly later than scheduled) arrival times of previous vehicles.

Collision avoidance between vehicles at the intersection is guaranteed by

$$t^{\text{low}}_{a,i} \geq t^{\text{safe}}_{a,i}, \qquad a \in A;\ i = 1, \ldots, n_a. \tag{5.11}$$

Finally, the cost function F_3 is defined for SP4_{A3} as:

$$\begin{aligned} F_3 = \quad & W_{\text{a}} \sum_{i=1}^{n_a} \sum_{k=1}^{n_i} (a_{i,k})^2 \frac{1}{n_i} + \\ & W_{\text{v}} \sum_{i=1}^{n_a} \sum_{k^{\text{t}}_i}^{n_i} (v_{i,k} - v^{\text{in}}_i)^2 \frac{1}{n_i - k^{\text{t}}_i}. \end{aligned} \tag{5.12}$$

The difference between F_2 and F_3 is that in F_3 there is no term for minimizing arrival time deviation, and instead of minimizing the speed deviation on time intervals around interval k^{t}_i, the speed deviation for any interval $k \geq k^{\text{t}}_i$ is minimized. Penalizing a speed deviation on any time step after k^{t}_i induces earlier arrivals, since reaching the desired speed reasonably soon lowers the value of F_3.

Formally, S3 consists of minimizing F_3 (i.e., (5.12)), subject to constraints (3.25), (3.27), (3.28), (5.3b), (5.4), (5.5), (5.8)–(5.11).

5.1.4 Implementation considerations

The three formulations designed produce sequences of vehicle states to be used as references over a given time horizon, just like the original formulation presented in section 3.7. These sequences are sent to the vehicles, which follow them accordingly until they reach the intersection.

It is assumed that if vehicles arrive at the intersection within the defined bounds (constraints) and follow their crossing profiles, safety is guaranteed in the intersection. More specifically, the schedule produced by SP3 must be such that any feasible solution for the motion planning problem using that schedule as an input must be absent of collisions. This means that the safety headways of SP3 must account for possible deviations in the schedule allowed by the motion planning strategy.

Consider a scheduled time of arrivel $t_{a,i}$ that is obtained as a solution for SP3. Since the exact instant of vehicle arrival at the intersection for $\mathrm{SP4}_{\mathrm{A2}}$ may be significantly different from $t_{a,i}$, the safety headways used by SP3 must be large enough to guarantee safety with respect to this uncertainty. In this situation, a vehicle may actually arrive at any time $T_{\mathrm{S}} \cdot (\alpha + 1)$ before or after the scheduled arrival time. In the case of two consecutive vehicles (not necessarily on the same approach) arriving at the latest and earliest possible arrival times, respectively, this amounts to arrival times possibly up to $2T_{\mathrm{S}} \cdot (\alpha + 1)$ closer than expected. Therefore, headways used in SP3 should be increased accordingly to guarantee safety.

5.2 SIMULATION RESULTS

Two types of experiments are performed to evaluate the optimal control strategies for SP4 presented in this chapter: (I) Solving one arbitrary schedule, and (II) solving a series of successive schedules as vehicles enter the CR during traffic simulation (i.e., in conjunction with the rest of the OATS strategy). The Gurobi solver (Gurobi Optimization, 2016) is used for solving the optimal control problems. The arbitrary schedules used are actually the result of instances of SP3 encountered during traffic simulation.

Simulation setup is similar to the one described in Section 4.1, with the following differences:

- Control distance is set as $D_a = 80$ m $\forall\ a \in A$.

- Default longitudinal headway h_{L} and default transversal headway h_{T} are set to the same value, and are refereed to simply by minimum headway h in this chapter ($h = h_{\mathrm{T}} = h_{\mathrm{L}}$).

- Demand is set to 400 veh/h/lane per approach unless otherwise specified.

- Vehicles are set to not update their schedules when a new solution for SP3 is found if they are too close to the intersection, just as the experiments discussed in Section 4.3.5. The distance after which they no longer change their schedules is called distance of no rescheduling (D_{NR}).

The distance of no rescheduling is set to $D_{\mathrm{NR}} = 40$ m for $\mathrm{SP4_{A1}}$ and $\mathrm{SP4_{A2}}$, and $D_{\mathrm{NR}} = 80$ m for $\mathrm{SP4_{A3}}$. Since for $\mathrm{SP4_{A3}}$ $D_a = D_{\mathrm{NR}}$, vehicles are never rescheduled. For $\mathrm{SP4_{A3}}$, the control horizon is 30 s.

The reasoning for not rescheduling in $\mathrm{SP4_{A3}}$ is that, since in this formulation vehicles follow the schedule very loosely (using the result of $\mathrm{SP4_{A3}}$ mostly to decide vehicle order), rescheduling has little impact.

Several metrics are used to compare scenarios. The sum of the square accelerations for each vehicle at each interval, multiplied by the interval length is used as a proxy for energy expenditure, and refereed to as "total energy". Total arrival time for experiments in which only one schedule is solved is given by the sum of the arrival time for each vehicle (assuming, for $\mathrm{SP4_{A2}}$ and $\mathrm{SP4_{A3}}$, arrival at the middle of the corresponding time interval). Arrival time error (only relevant for $\mathrm{SP4_{A2}}$) is the absolute difference between the expected arrival time and the scheduled arrival times $t_{a,i}$. Position error (only relevant for $\mathrm{SP4_{A1}}$) is the absolute position on the last time interval. Speed error is the absolute deviation from v_i^{in} at the interval in which a vehicle reaches the intersection (which, by definition, is the last interval for $\mathrm{SP4_{A1}}$).

Vehicle delay is the difference between the time a vehicle takes to cross the network and the time it would take to do so with free flow speed. Solution time is the time taken to solve one problem instance.

5.2.1 Experiments solving one schedule

Let the turning direction of a vehicle be defined as -1 if the vehicle turns left, 0 if it travels straight, and 1 if it turns right. Each row in Table 9 shows the desired turn, characteristics, initial state, and possible schedules for one vehicle. Consider the two schedules represented by the combination of v_i^{in} and either one of the two sets of $t_{a,i}$, one calculated using $h = 0.8$ s for SP3, and the other with $h = 1.6$ s (i.e., the third-to-last and second-to-last columns of Table 9). Let the problem of finding a motion plan for the first schedule (with $h = 0.8$ s) be called Problem Instance 1 (PI1), and for the second schedule (with $h = 1.6$ s) Problem Instance 2 (PI2).

Table 10 shows the results of solving PI1 with SP4_{A1} for different weight combinations on F_1. In Scenario 1 only energy expenditure is optimized. Scenario 2 corresponds to minimizing only deviation from the schedule, and in Scenario 3 both deviation from the schedule and energy are minimized. As could be expected, Scenarios 1 and 2 show the best performance in the criteria being optimized. Scenario 3 consists of a compromise, with results fairly close to Scenario 1 in terms of energy.

It is noteworthy that when all criteria are considered, energy results are similar to when only energy is minimized, while adherence to the schedule is much better. This happens for all three formulations,

Table 9 – Motion planning problem instance

a	i	turn	$a_i^{\min}$ (m/s^2)	$a_i^{\max}$ (m/s^2)	$v_{i,0}$ (m/s)	$d_{i,0}$ (m)	$t_{a,i}$ (s) $h = 0.8$ s	$t_{a,i}$ (s) $h = 1.6$ s	v_i^{in} (m/s)
1	1	0	-4.7	3.0	8.3	3.4	0.4	0.4	8.3
1	2	0	-4.3	2.7	8.3	15.2	1.8	2.1	7.4
1	3	0	-3.9	3.0	7.7	27.4	3.3	4.0	8.2
1	4	0	-3.3	2.7	8.3	39.0	4.7	5.7	7.4
1	5	0	-4.0	3.3	8.3	50.6	6.2	7.6	8.3
1	6	1	-3.8	3.0	8.3	61.3	7.5	14.4	5.5
1	7	0	-4.7	2.5	8.3	78.8	8.4	10.2	7.0
2	1	0	-3.7	3.0	3.5	35.6	12.5	13.2	5.7
2	2	0	-3.2	3.2	6.4	45.3	5.0	5.0	8.3
2	3	-1	-4.1	2.5	8.3	78.1	8.9	9.3	4.2
3	1	0	-3.9	3.2	8.3	41.9	16.0	14.9	8.3
3	2	-1	-4.8	2.6	7.8	71.4	13.9	19.1	4.5

Table 10 – Results of solving PI1 with $SP4_{A1}$ for different weights

Scn.	W_a	W_d W_v	total energy (m^2s^{-4})	total arrival time (s)	avg. speed error (m/s)	avg. position error (m)
1	1	0	31.3	88.7	0.080	0.417
2	0	1	76.2	88.7	0.000	0.001
3	1	1	33.6	88.7	0.059	0.099

Table 11 – Results for solving PI1 with $SP4_{A2}$, varying α, and $W_a = 1$, $W_t = W_v = 0$

Scn.	α	total energy (m^2s^{-4})	total arrival time (s)	avg. arrival time error (s)	avg. speed error (m/s)
4	0	26.3	88.8	0.225	0.008
5	1	24.2	88.4	0.187	0.041
6	2	22.4	88.0	0.270	0.069
7	3	21.0	87.8	0.370	0.090

unless weight factors differ by several orders of magnitude.

Table 11 shows the results of solving PI1 with $SP4_{A2}$ with different values of α, and $W_a = 1$ and $W_t = W_v = 0$ for all scenarios, (i.e, only energy is optimized). As α increases, energy expenditure decreases, and even the worse case shows less energy expenditure than Scenario 1. This happens because the increase of the allowed margin for vehicle arrival time provides more freedom for the controller to minimize energy. Notice that even though vehicle arrival times go down as α increases, these scenarios use the same schedule (and h), meaning safety is being compromised for larger values of α.

Consider that if $\alpha = 1$, a vehicle may arrive up to 0.4 s earlier or later than scheduled. In the worst case of a late vehicle followed by an early one, they can arrive at the intersection up to 0.8 s closer than expected. As such, to meet the same safety criteria, minimum headways should be increased by at least 0.8 s for $SP4_{A2}$ in this case. Table 12 shows the results for solving PI2 with $SP4_{A2}$, $\alpha = 1$, $W_a = 1$ and $W_t = W_v = 0$, which is a fairer comparison to Scenario 1. Total arrival time increases significantly, since headways are higher. Energy

Table 12 – Results of solving PI2 with $SP4_{A2}$ and $W_a = 1$, $W_t = W_v = 0$

Scn.	α	total energy (m^2s^{-4})	total arrival time (s)	avg. arrival time error (s)	avg. speed error (m/s)
8	1	59.5	121.8	0.12	0.010

Table 13 – Results for solving PI1 with $SP4_{A3}$ and $W_a = 1$, $W_v = 0$

Scn.	total energy (m^2s^{-4})	total arrival time (s)	avg. speed error (m/s)
9	27.182	103.2	0.008

also increases, mostly due to the fact that with higher headways most vehicles are scheduled to arrive later and need to decelerate/accelerate more.

Table 13 shows the results for solving PI1 with $SP4_{A3}$, with $W_a = 1$ and $W_v = 0$, and a control horizon of 30 seconds. Results are very similar when $W_v = 1$. Energy is lower compared to scenarios with $SP4_{A1}$, at the expense of higher total arrival time. Notice that both energy and arrival times are lower than when PI2 is solved by $SP4_{A2}$, indicating that $SP4_{A3}$ may be a better energy saving strategy than $SP4_{A2}$.

5.2.2 Traffic simulation results

The experiments in Section 5.2.1 show that for a specific schedule, $SP4_{A3}$ achieves the best results with respect to energy consumption, but it does not achieve a total arrival time as low as $SP4_{A1}$. However, all three optimization strategies only consider vehicles inside the CR, which is a region fairly close to the intersection, and the state of vehicles beyond that is not taken into account. In this section, traffic simulations are performed to evaluate the state of vehicles beyond the CR.

Table 14 shows simulation results with all three strategies for the scenario described in Section 5.2. Energy expenditure takes into

Table 14 – Comparison of $SP4_{A1}$, $SP4_{A2}$ and $SP4_{A3}$ in traffic simulation.

Scn.	Strat.	α	total energy (m^2s^{-4})	avg delay (s)	avg. speed (km/h)	avg. solution time (ms)
1	$SP4_{A1}$	-	4999 ± 235.4	1.2 ± 1.7	29.1	1.8 ± 1.1
2	$SP4_{A2}$	0	5089 ± 335.4	2.1 ± 2.7	28.6	3.9 ± 1.9
3	$SP4_{A2}$	1	5595 ± 369.9	3.4 ± 3.8	28.0	8.0 ± 8.0
4	$SP4_{A2}$	2	7273 ± 889.8	6.3 ± 6.1	26.7	21.9 ± 36.3
5	$SP4_{A3}$	-	5178 ± 372.8	2.9 ± 2.5	28.1	182.2 ± 70.8

account the entire time a vehicle is in the network, instead of just at the CR like the previous experiments. SP3 uses $h = 0.8$ s for $SP4_{A1}$ and $SP4_{A3}$, and $h = 0.8 + 2 \cdot (1 + \alpha)$ for $SP4_{A2}$. Recall that SP3 minimizes vehicle arrival times, producing schedules that are very tight for a large portion of vehicles (several vehicles must arrive as soon as possible). In all scenarios $W_a = W_t = W_d = W_v = 1$. Queues did not exceed the simulated area. Simulation time was 30 minutes for each scenario, during which 785 motion planning problems were solved.

Scenario 1 has a much lower delay than any other. This was already expected, since $SP4_{A1}$ allows no flexibility on the arrival times. In scenarios with $SP4_{A2}$, the increase in α and, hence, in h is followed by an increase in vehicle delay.

Scenario $SP4_{A1}$ is also the best in terms of energy expenditure, even though scenarios 2–5 should have more freedom to optimize energy. The reason for this is that, by being very strict about complying with the schedule, $SP4_{A1}$ makes vehicles leave the CR as fast as possible, which decreases the likelihood that a vehicle outside the CR (which is not taken into account by any formulation) has to slow down because of a vehicle in front of it. Since traffic demand is relatively high, an strategy which allows vehicles to exit the CR more rapidly is also the one that has lower global energy expenditure.

Table 15 and Figure 23 show energy and delay results for $SP4_{A1}$ and $SP4_{A3}$ for different traffic demands. In all cases $SP4_{A1}$ is the strategy that best reduces vehicle delay. For lower demands, of 320 veh/h/lane or less, $SP4_{A3}$ fares better with respect to energy consumption. For demands higher than that, the use of $SP4_{A1}$ leads to a lower energy expenditure. This happens because as demand decreases, so does the likelihood of a vehicle outside the CR being delayed by the

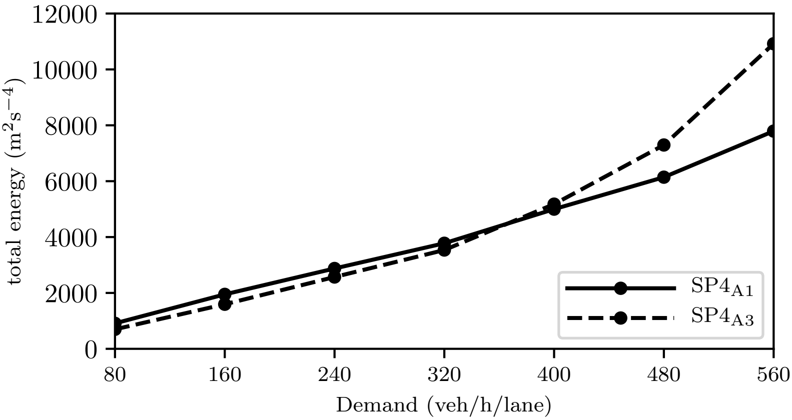
Table 15 – Comparison of $SP4_{A1}$ and $SP4_{A3}$ with varying demand

	$SP4_{A1}$		$SP4_{A3}$	
Demand (veh/h/lane)	total energy (m^2s^{-4})	avg delay (s)	total energy (m^2s^{-4})	avg delay (s)
560	7792 ± 585	2.6 ± 3.1	10919 ± 1286	8.2 ± 6.6
480	6143 ± 295	1.8 ± 2.3	7295 ± 655	4.6 ± 3.7
400	4999 ± 235	1.2 ± 1.7	5178 ± 373	2.9 ± 2.5
320	3778 ± 257	0.9 ± 1.4	3536 ± 587	2.4 ± 2.1
240	2875 ± 137	0.6 ± 1.2	2570 ± 145	1.8 ± 1.7
160	1949 ± 158	0.3 ± 0.8	1595 ± 132	1.4 ± 1.3
80	912 ± 112	0.2 ± 0.5	698 ± 86	1.1 ± 1.1

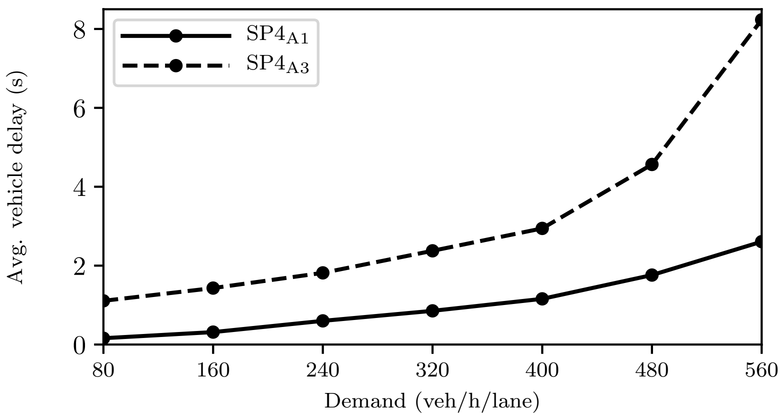
presence of a vehicle in front of it, and $SP4_{A1}$ is better at "removing" vehicles upstream from the way.

It is important to note that $SP4_{A2}$ and $SP4_{A3}$ were designed with the goal of evaluating the trade-off between allowing a deviation from the schedule and energy consumption, and the resulting formulations are not necessarily efficient (this was simply not a concern when modeling them). In fact, they are much less efficient than the formulation for $SP4_{A1}$. Solving $SP4_{A3}$ takes significantly more computational effort than the other strategies. Both $SP4_{A1}$ and $SP4_{A2}$ consist of one separate motion planning problem for each lane approaching the intersection, while $SP4_{A3}$ models all lanes in one problem instance. In $SP4_{A1}$ all constraints are linear, which makes it relatively easy (computationally) to solve. In $SP4_{A2}$ and $SP4_{A3}$, binary variables are used to model the time interval in which a vehicle arrives. In $SP4_{A2}$ the number of binary variables associated to each vehicle depends on the allowed deviation from the scheduled arrival time. When this increases, the problem becomes harder to solve. For small deviations, such as the ones used in this chapter, this effect is small, and $SP4_{A2}$ is not much harder than $SP4_{A1}$. In $SP4_{A3}$, vehicles may arrive much later than originally scheduled, and the number of binary variables associated to a vehicle is (mostly) limited by the control horizon. The larger number of binary variables is responsible for making the problem harder to solve than $SP4_{A1}$ and $SP4_{A2}$. The number of variables (both continuous and binary) is proportional to the number of vehicles for all three strategies. As such, solution time is expected to increase linearly to the number of vehicles.

Figure 23 – (a) Total energy spent and (b) average vehicle delay results for $SP4_{A1}$ and $SP4_{A3}$ under different traffic demands



(a)



(a)

5.3 REMARKS ON THE ALTERNATIVE MOTION PLANNING STRATEGIES

In this chapter, three different optimal control strategies were evaluated for solving SP4. The most flexible strategy ($SP4_{A3}$) with respect to arrival times provided the lowest energy consumption when solving a specific instance of the problem. However, when successive problems were solved as new vehicles approached the intersection during traffic simulation, no strategy outperformed strategy $SP4_{A1}$ in a high traffic situation. This happened because “clearing” the control region of vehicles as fast as possible ensures traffic conditions are better at the arriving lanes. This is beneficial for vehicles that have not yet entered the control region as it is less likely they have to decelerate when approaching, particularly if traffic demand is high.

These results suggest that, at least for high traffic demands and assuming free flow downstream, $SP4_{A1}$ is the most suitable of the three evaluated strategies for obtaining speed profiles for vehicles to approach the intersection, as it leads to both the lowest arrival times and lowest energy expenditure.

6 CONCLUSION

The results of this thesis are summarized on this chapter. Section 6.1 presents some final remarks about the research performed. Section 6.2 provides a brief summary of the main contributions of this thesis. Finally, Section 6.3 discusses possible paths for further research.

6.1 CONCLUDING REMARKS

Optimal Arrival Time Scheduling (OATS) was proposed and studied as a new intersection management strategy for automated vehicles. It provides high traffic efficiency by minimizing the time vehicles take to cross an intersection, while guaranteeing safety. OATS was studied in simulation and provides a significant improvement in traffic conditions when compared to the current usual control strategy – traffic lights. OATS allows vehicles to cross intersections with negligible delays for traffic demands comparable to the capacity achieved by current technology. Furthermore, OATS capacity is more than twice that of what can be achieved by traffic lights for the studied intersection layout, and still presents small delays even for such high traffic demand.

The OATS formulation separates the scheduling and motion planning problems. The resulting scheduling problem is NP-hard. However, several implications which result from the problem structure (detailed in Section 3.6.2.1), as well as possible simplifications (proposed in Section 3.6.3) allow the problem to be solved in a reasonable amount of time in the simulated scenarios. The resulting motion planning problem, on the other hand, has only linear constraints and a quadratic cost function, and can be solved relatively easily.

The downside of the separation of the problems is that vehicle states and control inputs can not be included in the scheduling problem, meaning vehicle arrival time (and order) is decided with no explicit regard for any measure of energy efficiency: only time is considered in the objective function of the scheduling problem. Even so, simulation results for different formulations of the motion planning problem, which sacrifice adherence to the schedule in favor of energy efficiency, suggest that having vehicles arriving as soon as possible at the intersection can actually be a good strategy in regards to energy efficiency when traffic demand is high. This is the case because by having vehicles arrive as early as possible, the likelihood they interfere with vehicles upstream – and cause them to decelerate – decreases. In short, clearing the intersection as soon as possible is a good energy saving strategy for high traffic situations.

The effects of control parameters of OATS on traffic flow were evaluated through microscopic traffic simulation. In particular, limiting the size of the scheduling problem by either using a small control region or limiting the number of vehicles with unbounded maximum arrival time in a problem instance was found to have very little impact on traffic conditions, and be a significant factor in problem size. Reducing the size of the control region from 100 to 30 m increased average travel time by only 0.6% (or 0.33 s), and decreased the time needed to solve the scheduling problem by one order of magnitude. This shows that having a control region much larger than strictly necessary comes at a significant cost and has very limited benefits, suggesting small control regions might be better suited for a practical application. Although only the OATS strategy was simulated, this result can be relevant for other intersection management strategies as well. It would be interesting to investigate wherever the observation that controlling vehicles far from the intersection has little benefits for traffic conditions is valid only for OATS, or if it also holds true for other intersection management strategies.

Minimum headway was also found to have a significant impact in vehicle delay. The choice of headway constitutes a trade-off between efficiency and safety, and also has a significant impact in computational effort, making the problem harder to solve as headways increase. Fixing a point after which vehicles can no longer update their schedule to a new one was found to have a smaller impact than simply decreasing the size of the control region. Both approximating the arrival time of vehicles with tight intervals and solving the scheduling problem in batches was found to have no beneficial impact for either solution time or traffic conditions, suggesting such considerations are inadequate for simplifying the problem.

It is important to highlight that small, easy to handle control regions are possible because the speed limit is relatively low. This puts yet another argument in favor for a low speed limit, as it both increases safety and allows a smaller problem to be considered at each control interval.

It also noteworthy that OATS, as proposed, has a very general formulation. This allows simpler strategies to be formulated as special cases of OATS. The modularity of the strategy also makes it relatively simple to change parts of it without needing to worry about the other subproblems. All things considered, it would be relatively straightforward to compare different strategies using OATS as a "framework". For instance, one could describe a scenario with only one conflict region, occupying the entire intersection. The scheduling subproblem could be changed to have vehicles arriving in a first-in-first out order, or in an order given by a predefined priority. SP1, SP2 and SP4 could be modified to set a limit on vehicle jerk (the derivative of acceleration) to formalize some level of passenger comfort. Conflict regions could be set to have no area, and transversal headways be set as large enough to account for safety by themselves. Alternatively, several conflict regions could be set to occupy the entire intersection without any overlap, obtaining a behavior akin to cell based reservation strategies. In short, the general nature of some aspects of OATS's formulation and its modularity make it suitable for the comparison of different intersection management strategies. Furtado (2017) has compared different intersection control strategies on several junction layouts, including a previous version of OATS (MÜLLER; CARLSON; KRAUS JR., 2016a). Similar research could be done using the more general version of OATS presented in this thesis as a framework to implement several control strategies without the need of implementing each strategy from scratch.

6.2 MAIN CONTRIBUTIONS

The main contributions of this thesis can be summarized as:

- A new control strategy for coordinating automated vehicles as they approach and cross a single intersection was proposed, called Optimal Arrival Time Scheduling (OATS). OATS, as proposed, guarantees minimum aggregate arrival time and the absence of vehicle collisions. It is formulated in a sufficiently general form to allow for any arbitrary intersection layout, and thanks to this and its modularity, it can be used as a framework for evaluating other control strategies to some extent. OATS makes no assumptions about vehicle order of arrival at the intersection, which is optimized online, and allows vehicles with potential conflicts to occupy the intersection at the same time, as long as they satisfy safety constraints. OATS allows for high efficiency of traffic and can be solved in a reasonably short time for the evaluated scenarios. OATS was evaluated in microscopic traffic simulation under various control configurations and traffic demands.

- Three different motion planning strategies were proposed for solving the problem of vehicles arriving at an intersection while complying to a set schedule, differing in how much they accept to deviate to the schedule.
- A greater understanding of intersection management with automated vehicles. In particular, the notions that
  - increasing the size of the control region or otherwise limiting the number of vehicles to be considered in the scheduling problem translates into very small benefits in traffic conditions and a significant cost in computational effort, meaning small control regions may be better suited for practical implementation; and
  - for high traffic demands, ensuring vehicles reach the intersection – and hence leave the control area – as soon as possible is beneficial from an energy conservation standpoint, as it minimizes the likelihood that vehicles upstream have to decelerate due to the influence of vehicles downstream.

6.2.1 Novel contributions

Although the OATS strategy as a whole, and hence much of the developments presented in this thesis are novel, not every aspect of OATS is novel in itself, as it has some elements in common with other CIM strategies. This subsection goes into a more detailed discussion in an attempt to clarify what exactly is novel about the work exposed in this Thesis[1].

The idea of decomposing the overal problem by separating the the overall coordination problem and the motion planning problem in subproblems to be solved sequentially can also be seen in, e.g., Oliveira et al. (2002), even if this is not explicitly stated and the motion planning problem is defined in a much simpler way.

Constraining vehicles to follow fixed speed profiles inside the IR is a novel idea, although it is merely a generalization of the common assumption of constant speed inside the intersection (as in, e.g., Zohdy and Rakha (2016)).

Allowing vehicles to stop while approaching the IR and setting the control distance to be sufficiently large so a vehicle entering the

[1] When something is deemed as a novelty in this context, what is actually meant is that it is not present in any literature the author is familiar with.

CR is always able to stop and later accelerate to the crossing speed is also a novel contribution of this thesis. If this was not the case, some other mechanism would have to be used to guarantee feasibility for heavy traffic flows. A large number of vehicles entering the CR in a short amount of time, without the ability to wait indefinitely, can easily result in a unfeasible scheduling problem. This is not much discussed in the literature, possibly because it is a very unlikely occurrence for small traffic volumes or large control regions.

Similarly, the observation that the behavior of vehicles outside the CR must be such that they enter the CR sufficiently spaced to be guaranteed to not lead to an unfeasible problem is also novel. This issue is very unlikely to become a problem unless traffic flows are very large and vehicles are allowed to be relatively "agressive" (i.e., keeping very short distances to the vehicle ahead) outside the CR.

The basic structure of SP3 – minimizing a metric of performance subject to disjunctive constraints that model a set of choices – is standard for scheduling problems. However, its use in the CIM context together with the additional constraints proposed is a novel contribution.

Basically everything else pertaining to SP3 and discussed on Section 3.6 is a contribution of this thesis. Namely: (i) the other three special cases of arrival order discussed; (ii) the observation that intersection layouts without movements that cross each other more than once can decrease the number of conflicts; (iii) the differentiation of how longitudinal and transversal conflicts are handled, by forbidding vehicles with transversal conflicts to share a conflict region, but allowing vehicles with longitudinal conflicts to do so, and enforcing a minimum headway in addition to this; (iv) the use of variable headways, which can be increased if the vehicle in front has a lower speed; (v) the inclusion of capacity headways h_c^{cap} to constrain exit flow; (vi) the discussion on possible considerations or simplifications to limit the size of the problem and its computational complexity[2]; and (vii) the discussion on pedestrian crossings, and how to obtain this behavior by modeling a virtual vehicle.

The concepts of guaranteeing that a CIM strategy is deadlock free; implementing priorities for vehicles; and attempting to quantify the capacity flow that can be obtained are not novel in themselves,

[2]Although some works do effectively employ the simplifications of scheduling in batches (e.g., Kamal et al. (2015)); or not re-scheduling vehicles for which a previous solution was already found (e.g., Zohdy and Rakha (2016)); there is no in-depth discussion of the effects of such simplifications, neither a comparison with scenarios in which these simplifications are not present.

being very common concerns when dealing with traffic flow. The analysis performed on these aspects, however, are particular to OATS, and therefore a contribution of the present work.

The basic structure of SP4 – minimizing some metric of performance subject to constraints on vehicle dynamics – is somewhat straightforward and very common for motion planning problems. The particular objective function chosen, however, is another contribution of this thesis at least in the context of CIM (minimizing the deviation from a scheduled position and speed at a given time). The alternative formulations presented on Chapter 5 build up on this, by also formalizing some measure of deviation from the schedule, and therefore are novel as well. Interestingly, the third alternative formulation $SP4_{A3}$ is effectively very similar to CIM strategies that optimize vehicles sequentially in a predefined arrival order, such as Zhang, Cassandras and Malikopoulos (2017), albeit with a different objective function.

The discussion on how the IC deals with uncertainties and possible deviation in trajectory following, by effectively ignoring small deviations and solving SP4 again when vehicles deviate too much from the expected path, is also novel in the context if CIM.

Finally, the simulation results and associated findings and discussions are one of the main contributions of this thesis, specially the evaluation of a CIM strategy under varied control configurations and traffic demands and the investigation of how these aspects affect traffic coditions and computation effort.

6.2.2 Publications

Preliminary versions of OATS have been the subject of two papers written by the author during his doctorate, leading up to this thesis. This document presents a much more general formulation than the ones already published, and expands on several subjects not previously analyzed.

In Müller, Carlson and Kraus Jr. (2016a) a much simplified intersection layout was used, composed by only two traffic streams, and vehicles in conflicting movements were not allowed to be at the intersection simultaneously. In Müller, Carlson and Kraus Jr. (2016b), the concept of conflict points was used to generalize the formulation for any intersection layout. In both cases, only minimum headways with a fixed value were used as constraints, the first and second subproblems that compose OATS were only conceptually explained, and the motion planning problem was solved with a simple heuristic.

Much of the discussion on Chapter 5 regarding alternative formulations for the motion planning problem that take energy consumption into account were presented by the author in Müller, Wahlberg and Carlson (2018).

6.3 FURTHER RESEARCH

As discussed in Section 1, one of the motivations for designing OATS in the first place was the need for a suitable strategy for a multi-level network coordination scheme. As such, one possible research direction is the integration of multiple intersections controlled by OATS and a network level traffic assignment algorithm. One preliminary control structure could consist of a network level supervisor which constantly monitors the congestion level of the network links (roads) and, when congestion is high, sends values of h_c^{cap} to the intersections in order to restrict the outflow to congested links. A supervisor could be designed to avoid gridlocks, or implement other suitable control strategies.

Another research direction is the further study of OATS under different conditions, specially with other road layouts. The presence of dedicated turning lanes, for instance, could have a significant impact on the strategy. The study of roundabouts seems specially interesting, as they – and specially turbo roundabouts – result in intersection layouts that significantly reduce the number of conflict points. This may be very relevant for OATS, as a smaller number of conflict points can translate into easier problems, and possibly higher capacity flow. On the other hand, other aspects such as the fact that there is usually a greater number of vehicles inside a roundabout than inside a typical intersection may also have a significant effect. It is not yet clear which junction design – cross intersections or roundabouts – is more efficient for automated traffic. More research is necessary on this direction. Furtado (2017) shows some results for an earlier version of OATS in different layouts, including roundabouts, but he does not investigates traffic demands higher than what can be achieved with current technology.

The study of the OATS strategy in simulation considering a more realistic car model is a necessary step before eventual field tests. In this thesis, vehicles are assumed to be able to change acceleration instantly, and discontinuities in acceleration are allowed. The vehicle model, as well as SP1, SP2 and SP4 can be adapted to take into account more realistic constraints on acceleration.

Finally, OATS could also be used as a framework to implement different intersection management strategies and compare them in the same environment without having to implement each one from scratch. This can be used to verify if some of the the most relevant observations made in this thesis: (i) the fact that controlling vehicles far from the intersection has little benefits for traffic conditions; and (ii) the fact that minimizing vehicle arrival time translates to a good energy saving strategy under high demand situations; are valid only for OATS, or if these results are valid for intersection management in general.

REFERENCES

BLAZEWICZ, J.; LENSTRA, J. K.; KAN, A. R. Scheduling subject to resource constraints: classification and complexity. **Discrete applied mathematics**, Elsevier, v. 5, n. 1, p. 11–24, 1983.

CAMPOS, G. R. de et al. Traffic coordination at road intersections: Autonomous decision-making algorithms using model-based heuristics. **IEEE Intelligent Transportation Systems Magazine**, v. 9, p. 8–21, 2017. ISSN 1939-1390.

CHEN, L.; ENGLUND, C. Cooperative intersection management: a survey. **IEEE Transactions on Intelligent Transportation Systems**, IEEE, v. 17, n. 2, p. 570–586, 2016.

COLOMBO, A.; VECCHIO, D. D. Efficient algorithms for collision avoidance at intersections. In: ACM. **15th ACM international conference on Hybrid Systems: Computation and Control**. Beijing, China, 2012. p. 145–154.

COLOMBO, A.; VECCHIO, D. D. Least restrictive supervisors for intersection collision avoidance: A scheduling approach. **IEEE Transactions on Automatic Control**, IEEE, v. 60, n. 6, p. 1515–1527, 2015.

DAI, P. et al. Quality-of-experience-oriented autonomous intersection control in vehicular networks. **IEEE Transactions on Intelligent Transportation Systems**, IEEE, v. 17, n. 7, p. 1956–1967, 2016.

DRESNER, K.; STONE, P. Multiagent traffic management: A reservation-based intersection control mechanism. In: IEEE COMPUTER SOCIETY. **Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 2**. 2004. p. 530–537.

DRESNER, K.; STONE, P. A multiagent approach to autonomous intersection management. **Journal of Artificial Intelligence Research**, v. 31, p. 591–656, 2008.

FURTADO, H. S. **Evaluation of Intersection Control Strategies for Automated Vehicles**. 2017.

GIPPS, P. G. A behavioural car-following model for computer simulation. **Transportation Research Part B: Methodological**, Elsevier, v. 15, n. 2, p. 105–111, 1981.

GRAHAM, R. L. et al. Optimization and approximation in deterministic sequencing and scheduling: a survey. In: **Annals of discrete mathematics**. : Elsevier, 1979. v. 5, p. 287–326.

GREGOIRE, J.; BONNABEL, S.; FORTELLE, A. D. L. Priority-based coordination of robots. 2014. 20 pages, 2014, <hal-00828976v3>.

GREGOIRE, J.; BONNABEL, S.; FORTELLE, A. de L. Robust multirobot coordination using priority encoded homotopic constraints. **arXiv preprint arXiv:1306.0785**, 2015.

Gurobi Optimization, I. **Gurobi Optimizer Reference Manual**. 2016. Disponível em: <http://www.gurobi.com>.

HULT, R. et al. Primal decomposition of the optimal coordination of vehicles at traffic intersections. In: **55th IEEE Conference on Decision and Control**. 2016. p. 2567–2573. ISBN 978-1-5090-1837-6.

KAMAL, M. A. S. et al. A vehicle-intersection coordination scheme for smooth flows of traffic without using traffic lights. **IEEE Transactions on Intelligent Transportation Systems**, IEEE, v. 16, n. 3, p. 1136–1147, 2015.

KATRAKAZAS, C. et al. Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions. **Transportation Research Part C: Emerging Technologies**, Elsevier, v. 60, p. 416–442, 2015.

KESTING, A. et al. Extending adaptive cruise control to adaptive driving strategies. **Transportation Research Record: Journal of the Transportation Research Board**, Transportation Research Board of the National Academies, n. 2000, p. 16–24, 2007.

KOWSHIK, H.; CAVENEY, D.; KUMAR, P. Provable systemwide safety in intelligent intersections. **IEEE transactions on vehicular technology**, IEEE, v. 60, n. 3, p. 804–818, 2011.

LEE, J.; PARK, B. Development and evaluation of a cooperative vehicle intersection control algorithm under the connected vehicles environment. **IEEE Transactions on Intelligent Transportation Systems**, IEEE, v. 13, n. 1, p. 81–90, 2012.

LENNARTZ, P. M. **No-Wait Job Shop Scheduling, a Constraint Propagation Approach**. Tese (Doutorado) — Utrecht University, 2006.

LEVIN, M. W.; FRITZ, H.; BOYLES, S. D. On optimizing reservation-based intersection controls. **IEEE Transactions on Intelligent Transportation Systems**, IEEE, v. 18, n. 3, p. 505–515, 2017.

LEVINSON, D. et al. **The Transportation Futures Project: Planning for Technology Change**. 2016. 141 p.

LI, L.; WANG, F.-Y. Cooperative driving at blind crossings using intervehicle communication. **IEEE Transactions on Vehicular Technology**, IEEE, v. 55, n. 6, p. 1712–1724, 2006.

LI, L.; WEN, D.; YAO, D. A survey of traffic control with vehicular communications. **IEEE Transactions on Intelligent Transportation Systems**, IEEE, v. 15, n. 1, p. 425–432, 2014.

LIANG, C.-Y.; PENG, H. Optimal adaptive cruise control with guaranteed string stability. **Vehicle system dynamics**, Taylor & Francis, v. 32, n. 4-5, p. 313–330, 1999.

MASCIS, A.; PACCIARELLI, D. Job-shop scheduling with blocking and no-wait constraints. **European Journal of Operational Research**, Elsevier, v. 143, n. 3, p. 498–517, 2002.

MILANES, V. et al. An intelligent v2i-based traffic management system. **IEEE Transactions on Intelligent Transportation Systems**, IEEE, v. 13, n. 1, p. 49–58, 2012.

MINETT, C. F. et al. Eco-routing: comparing the fuel consumption of different routes between an origin and destination using field test speed profiles and synthetic speed profiles. In: IEEE. **2011 IEEE Forum on Integrated and Sustainable Transportation System**. 2011. p. 32–39.

MÜLLER, E. R.; CARLSON, R. C.; KRAUS JR., W. Intersection control for automated vehicles with milp. In: **14th IFAC Symposium on Control in Transportation Systems**. Istanbul, Turkey: , 2016a.

MÜLLER, E. R.; CARLSON, R. C.; KRAUS JR., W. Time optimal scheduling of automated vehicle arrivals at urban intersections. In:

IEEE 19th International Conference on Intelligent Transportation Systems (ITSC). 2016b. p. 1174–1179.

MüLLER, E. R.; WAHLBERG, B.; CARLSON, R. C. Optimal motion planning for automated vehicles with scheduled arrivals at intersections. In: **17th European Control Conference**. Limassol, Cyprus: , 2018. Accepted.

NAUMANN, R. et al. Validation and simulation of a decentralized intersection collision avoidance algorithm. In: IEEE. **IEEE Conference on Intelligent Transportation System**. 1997. p. 818–823.

NHTSA. **Critical reasons for crashes investigated in the National Motor Vehicle Crash Causation Survey**. 2014. 238 p.

NHTSA. **A Compilation of Motor Vehicle Crash Data from the Fatality Analysis Reporting System and the General Estimates System**. 2015.

OLIVEIRA, R. S. d. et al. Controle ótimo de um cruzamento automatizado de tráfego urbano. In: **Anais do XIV Congresso Brasileiro de Automática**. Natal-RN: , 2002. v. 1, p. 1501–1506.

PAPAGEORGIOU, M. et al. Review of road traffic control strategies. **Proceedings of the IEEE**, IEEE, v. 91, n. 12, p. 2043–2067, 2003.

PEDEN, M. et al. **World report on road traffic injury prevention**. : World Health Organization Geneva, 2004.

QIAN, X. et al. Decentralized model predictive control for smooth coordination of automated vehicles at intersection. In: IEEE. **Control Conference (ECC), 2015 European**. 2015. p. 3452–3458.

RICHARDS, D. Relationship between speed and risk of fatal injury: pedestrians and car occupants. DfT, 2010.

RIOS-TORRES, J.; MALIKOPOULOS, A. A. A survey on the coordination of connected and automated vehicles at intersections and merging at highway on-ramps. **IEEE Transactions on Intelligent Transportation Systems**, IEEE, v. 18, n. 5, p. 1066–1077, 2017a.

RIOS-TORRES, J.; MALIKOPOULOS, A. A. Automated and cooperative vehicle merging at highway on-ramps. **IEEE Transactions on Intelligent Transportation Systems**, IEEE, v. 18, n. 4, p. 780–789, 2017b.

SAE. **J3016, Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems**. 2014.

SCHEPPERLE, H.; BÖHM, K.; FORSTER, S. Towards valuation-aware agent-based traffic control. In: ACM. **Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems**. 2007. p. 185.

SCHRANK, D. et al. **2015 Urban Mobility Scorecard**. 2015.

SHAH, S. A. A. et al. 5g for vehicular communications. **IEEE Communications Magazine**, IEEE, v. 56, n. 1, p. 111–117, 2018.

SHLADOVER, S.; SU, D.; LU, X.-Y. Impacts of cooperative adaptive cruise control on freeway traffic flow. **Transportation Research Record: Journal of the Transportation Research Board**, n. 2324, p. 63–70, 2012.

SHLADOVER, S. E. Cooperative (rather than autonomous) vehicle-highway automation systems. **Intelligent Transportation Systems Magazine, IEEE**, IEEE, v. 1, n. 1, p. 10–19, 2009.

SHLADOVER, S. E. **Recent international activity in cooperative vehicle–highway automation systems**. 2012.

SHLADOVER, S. E. Connected and automated vehicle systems: Introduction and overview. **Journal of Intelligent Transportation Systems**, Taylor & Francis, p. 1–11, 2017.

Transport Simulation Systems. **AIMSUN Users' Manual v. 8**. 2015.

USDOT. **ITS ePrimer**. 2015. URL: https://www.pcb.its.dot.gov/ePrimer.aspx (visited on 09-12-2015).

VINE, S. L.; ZOLFAGHARI, A.; POLAK, J. Autonomous cars: The tension between occupant experience and intersection capacity. **Transportation Research Part C: Emerging Technologies**, Elsevier, v. 52, p. 1–14, 2015.

WEBSTER, F. Traffic signal settings. **Road Research Technical Paper**, v. 39, 1958.

WORLD HEALTH ORGANIZATION. **WHO global status report on road safety 2015: supporting a decade of action**. : World Health Organization, 2015.

WU, J.; ABBAS-TURKI, A.; MOUDNI, A. E. Cooperative driving: an ant colony system for autonomous intersection management. **Applied Intelligence**, Springer, v. 37, n. 2, p. 207–222, 2012.

WU, J.; PERRONNET, F.; ABBAS-TURKI, A. Cooperative vehicle-actuator system: a sequence-based framework of cooperative intersections management. **IET Intelligent Transport Systems**, IET, v. 8, n. 4, p. 352–360, 2013.

YAN, F.; DRIDI, M.; MOUDNI, A. E. Autonomous vehicle sequencing algorithm at isolated intersections. In: IEEE. **12th International IEEE Conference on Intelligent Transportation Systems.** 2009. p. 1–6.

ZANON, M. et al. An asynchronous algorithm for optimal vehicle coordination at traffic intersections. In: **20th IFAC World Congress**. 2017.

ZHANG, Y.; CASSANDRAS, C. G.; MALIKOPOULOS, A. A. Optimal control of connected automated vehicles at urban traffic intersections: A feasibility enforcement analysis. In: IEEE. **American Control Conference (ACC), 2017**. 2017. p. 3548–3553.

ZHANG, Y. J.; MALIKOPOULOS, A. A.; CASSANDRAS, C. G. Optimal control and coordination of connected and automated vehicles at urban traffic intersections. In: **2016 American Control Conference (ACC)**. 2016. p. 6227–6232.

ZHU, F.; UKKUSURI, S. V. A linear programming formulation for autonomous intersection control within a dynamic traffic assignment and connected vehicle environment. **Transportation Research Part C: Emerging Technologies**, Elsevier, v. 55, p. 363–378, 2015.

ZOHDY, I. H.; RAKHA, H. A. Intersection management via vehicle connectivity: The intersection cooperative adaptive cruise control system concept. **Journal of Intelligent Transportation Systems**, Taylor & Francis, v. 20, n. 1, p. 17–32, 2016.

APPENDIX A – Defining the Feasible Arrival Interval

This appendix details how the feasible arrival interval $[t_{a,i}^{\min},t_{a,i}^{\max}]$ is obtained in SP2.

Consider vehicle i in approach a, currently with speed v_i and position d_i (measured as the distance to the end of approach a). Recall vehicle dynamics is constrained by $v_i^{\max}$, $a_i^{\min}$ and $a_i^{\max}$; and that vehicle i should arrive at the intersection with speed v_i^{in}.

The fastest possible way for vehicle i to reach the intersection is to either:

- accelerate with $a_i^{\max}$ until it reaches $v_i^{\max}$, cruise with maximum speed for as long as possible, and then decelerate with $a_i^{\min}$ reaching v_i^{in} at the intersection; or
- In case there is not sufficient space to reach $v_i^{\max}$, accelerate until a higher speed v_i^{high} and then decelerate to v_i^{in}.

Let t_i^{A} and d_i^{A} be the time and distance vehicle i needs to accelerate from v_i to $v_i^{\max}$, respectively; and t_i^{D} and d_i^{D} be the time and distance vehicle i needs to decelerate from $v_i^{\max}$ to v_i^{in}. These values are given by:

$$t_i^{\text{A}} = \frac{v_i^{\max} - v_i}{a_i^{\max}}, \tag{A.1}$$

$$t_i^{\text{D}} = \frac{v_i^{\text{in}} - v_i^{\max}}{a_i^{\min}}, \tag{A.2}$$

$$d_i^{\text{A}} = t_i^{\text{A}} \cdot \frac{v_i + v^{\max}}{2}, \tag{A.3}$$

$$d_i^{\text{D}} = t_i^{\text{D}} \cdot \frac{v_i^{\max} + v_i^{\text{in}}}{2}. \tag{A.4}$$

If $d_i^{\text{A}} + d_i^{\text{D}} \geq d_i$, then vehicle i has sufficient space to accelerate to $v_i^{\max}$, and its speed profile with the fastest arrival at the intersection involves traveling the remaining distance d_i^{H} with speed $v_i^{\max}$ for a time t_i^{H}. These are given by:

$$d_i^{\text{H}} = d_i - d_i^{\text{A}} - d_i^{\text{D}}, \tag{A.5}$$

$$t_i^{\text{H}} = \frac{d_i^{\text{H}}}{v_i^{\max}}. \tag{A.6}$$

Clearly, in this case

$$t_{a,i}^{\min} = t_i^{\text{A}} + t_i^{\text{H}} + t_i^{\text{D}}. \tag{A.7}$$

On the other hand, if $d_i^{\mathrm{A}} + d_i^{\mathrm{D}} < d_i$, then vehicle i can not reach $v_i^{\max}$. In this case, the highest speed it will reach during its trajectory, v_i^{high}, is given by the positive solution of:

$$v_i^{\mathrm{high}} = \sqrt{\frac{d_i + \dfrac{-a_i^{\min} \cdot (v_i)^2 + a_i^{\max} \cdot (v_i^{\mathrm{in}})^2}{-2 \cdot a_i^{\min} \cdot a_i^{\max}}}{\dfrac{a_i^{\max} - a_i^{\min}}{-2 \cdot a_i^{\min} \cdot a_i^{\max}}}}. \tag{A.8}$$

In this case, $t_{a,i}^{\min}$ is equal to the time vehicle i spends accelerating to v_i^{high} and then decelerating to v_i^{in}, and is given by:

$$t_{a,i}^{\min} = \frac{v_i^{\mathrm{high}} - v_i}{a_i^{\max}} + \frac{v_i^{\mathrm{in}} - v_i^{\mathrm{high}}}{a_i^{\min}}. \tag{A.9}$$

Now, in order to obtain $t_{a,i}^{\max}$, consider that vehicle i needs at least a time t_i^{C} in order to be able to change its speed to v_i^{in}. If $v_i < v_i^{\mathrm{in}}$ (i.e., it is initially traveling slower than v_i^{in}), then:

$$t_i^{\mathrm{C}} = \frac{v_i^{\mathrm{in}} - v_i}{a_i^{\max}}, \tag{A.10}$$

otherwise, if $v_i^{\mathrm{in}} \leq v_i$ (i.e., it is initially traveling faster than v_i^{in}), then:

$$t_i^{\mathrm{C}} = \frac{v_i^{\mathrm{in}} - v_i}{a_i^{\min}}. \tag{A.11}$$

In either case, the distance d_i^{C} vehicle i needs to achieve speed v_i^{in} is given by

$$d_i^{\mathrm{C}} = t_i^{\mathrm{C}} \cdot \frac{v_i + v_i^{\mathrm{in}}}{2}. \tag{A.12}$$

If vehicle i is at distance d_i^{C} from the intersection (i.e., $d_i = d_i^{\mathrm{C}}$), then it has no other choice than immediately assume either $a_i^{\min}$ or $a_i^{\max}$ to achieve v_i^{in} as soon as possible. In this case, vehicle i has only one possible instant of arrival at the intersection, which means $t_{a,i}^{\max} = t_{a,i}^{\min}$, and there is no need to calculate $t_{a,i}^{\max}$ again as $t_{a,i}^{\min}$ is already available from either (A.7) or (A.9).

If, on the other hand, $d_i > d_i^{\mathrm{C}}$, then vehicle i has sufficient space to decelerate to a certain speed v_i^{low} and then accelerate to v_i^{in}. The lowest speed v_i^{low} vehicle i can reach during this is given by the positive real solution of:

$$v_i^{\mathrm{low}} = \sqrt{(v_i)^2 + d \cdot a_i^{\min} \cdot \frac{\frac{(v_i^{\mathrm{in}})^2 - (v_i)^2}{2} - a_i^{\max} \cdot d_i}{a_i^{\min} - a_i^{\max}}}. \tag{A.13}$$

If (A.13) has no positive real solution, then it is possible for vehicle i to decelerate until it comes to a complete stop, wait for an arbitrary time, and then accelerate again. In that case, $t_{a,i}^{\max}$ is unbounded. In the experiments performed, an unbounded $t_{a,i}^{\max}$ was saturated at $t_{a,i}^{\max} = 120$ s. If, on the other hand, (A.13) has a positive real solution, then $t_{a,i}^{\max}$ is given by the time it takes to decelerate to v_i^{low} and then accelerate to v_i^{in}, and is given by:

$$t_{a,i}^{\max} = \frac{v_i^{\mathrm{in}} - v_i^{\mathrm{low}}}{a_i^{\max}} + \frac{v_i^{\mathrm{low}} - v_i}{a_i^{\min}}. \tag{A.14}$$

APPENDIX B – Defining the speed diference safety headway

This appendix presents how the speed difference headway, $h^{\Delta \mathrm{v}}_{a,i,b,j,c}$ introduced in Section 3.6.2.3 is obtained.

Consider vehicle i being followed by vehicle j traveling along the same path, crossing conflict regions c_1 and c_2, in this order. By definition, the time $\Delta_i^{c_1,c_2}$ vehicle i takes to travel from $c1$ to $c2$ is given by

$$\Delta_i^{c_1,c_2} = \tau_{i,c_2}^{\mathrm{arrive}} - \tau_{i,c_1}^{\mathrm{arrive}}. \tag{B.1}$$

If vehicle j is faster than vehicle i to cross the distance between c_1 and c_2 (i.e., $\Delta_j^{c_1,c_2} < \Delta_i^{c_1,c_2}$), then vehicle j can get closer to vehicle i while it travels between conflict regions, which might be a safety risk in case the regions are sufficiently apart and h_{L} is relatively small (which is actually not the case for the simulated scenarios). To avoid that, the speed difference safety headway is set as

$$h^{\Delta \mathrm{v}}_{a,i,b,j,c} = \Delta_i^{c_1,c_2} - \Delta_j^{c_1,c_2} \tag{B.2}$$

and is added to the headway $h_{a,i,b,j,c}$. This guarantees that the vehicles will still be keeping the minimum longitudinal headway at all points along their journey between conflict regions, assuming the speed profile the vehicles follow between conflict regions c_1 and c_2 is monotonic[1] (i.e., vehicles do not both accelerate and decelerate in between a pair of adjacent conflict regions).

Now, assume c_2 is at the border of the IR, and vehicles leave the CR when they leave c_2. For simplicity, the speed headway is calculated in a conservative manner in this case. It is assumed that every vehicle is followed by a vehicle traveling with speed $v_i^{\max}$. Vehicles are always able to accelerate to $v_i^{\max}$ after they leave the IR, since free flow conditions are assumed. The calue of $h^{\Delta \mathrm{v}}_{a,i,b,j,c}$ for conflict regions at the exits of the IR must be sufficiently large to allow a vehicle to accelerate to $v_i^{\max}$ before a potential vehicle behind approaches it too much.

Let v_i^{out} be the speed with which vehicle i leaves the IR. Vehicle i takes a time t_i^{a} to accelerate to $v_i^{\max}$ after leaving the IR, given by

[1] If speed profiles can assume any arbitrary form, only checking the time vehicles arrive at conflict regions is not sufficient, and calculating $h^{\Delta \mathrm{v}}_{a,i,b,j,c}$ becomes more complex

$$t_i^{\mathrm{a}} = \frac{v_i^{\max} - v_i^{\mathrm{out}}}{a_i^{\max}}. \tag{B.3}$$

During this time, an hypothetical follower vehicle, traveling with speed $v_i^{\max}$, gets Δ_{d} closer to vehicle i, with

$$\Delta_{\mathrm{d}} = t_i^{\mathrm{a}} \cdot \frac{v_i^{\max} - v_i^{\mathrm{out}}}{2}. \tag{B.4}$$

In order to balance that effect, $h_{a,i,b,j,c}^{\Delta\mathrm{v}}$ must be such that it allows vehicle i to cover this distance just after leaving the IR while it accelerates to $v_i^{\max}$. Hence, for conflict regions at the border of the IR:

$$h_{a,i,b,j,c}^{\Delta\mathrm{v}} = \frac{\Delta_{\mathrm{d}}}{v_i^{\mathrm{out}}}. \tag{B.5}$$

APPENDIX C – Fixed time control plan results

The usual solution for managing intersections with current technology is the use of traffic lights to coordinate human driven vehicles. This appendix shows the design and simulation results of a fixed-time control plan (i.e., traffic lights) for the same intersection layout considered throughout this thesis. This is done to obtain quantitative results representative of the traffic conditions resulting from the use of current technology, allowing the comparison of OATS with the current typical solution for the coordination problem at intersections.

Although sophisticated techniques can be used for designing a semaphoric control, possibly in real time, a fixed-time control policy is used in this section. This is representative of most intersections, and is adequate for a comparison with the scenarios of this thesis, which have a small variation in traffic demand. The benefits of real time control are more pronounced when demand varies, which is not the case for the experiments performed.

C.1 A FIXED TIME CONTROL PLAN THAT MINIMIZES VEHICLE DELAY

Webster (1958) proposed a method for designing a fixed time control plan that minimizes vehicle delay, based on a theoretical analysis and empirical data.

A semaphoric control plan is composed by a sequence of phases. During each phase, green time is granted to a subset of the allowed movements in the intersection, while controlled movements that are not part of the current phase are blocked (i.e., receive a red light). The total length of a plan (after which it is either repeated, or a different plan is applied) is called the cycle length.

The optimal cycle length c_0 for a fixed control plan that minimizes vehicle delay is given by

$$c_0 = \frac{1.5 \cdot L + 5}{1 - Y} \tag{C.1}$$

with L the total lost time in a cycle and Y the sum of the critical occupancy (or flow ratio) of each phase.

The lost time is the time during which vehicles do not cross the intersection from any approaching link. This time is a result of the fact drivers take some time to react to a change in signal, and also of the existence of yellow and clearance red times for safety reasons. Clearence red is a time during which all phases receive a red light.

The occupancy (or flow ratio) of a movement is the ratio between its demand and its capacity. Several movements can be active at the same phase. The movement that has the highest flow ratio in a phase is called the critical movement of that phase. Let y_i be the critical flow ratio of phase i, given by

$$y_i = \frac{q_i}{q_i^{\text{cap}}} \tag{C.2}$$

with q_i and q_i^{cap} the flow and capacity of the critical movement of phase i, respectively. The sum of the critical occupancies y is given by

$$Y = \sum_{i=1}^{n} y_i \tag{C.3}$$

with n the number of phases in the intersection. After defining cycle length, the green time (which is the cycle length minus the lost time) is divided among the phases in such a way that all phases should have the same saturation ratio. That is, for each phase i, the green time g_i of that phase is given by

$$g_i = (c_0 - L) \cdot \frac{y_i}{Y} \tag{C.4}$$

Notice that as $Y \to 1$, $c_0 \to \infty$. As traffic demand increases, so does the optimal cycle. However, at a certain point, increasing the cycle further has very little effect in decreasing delay, and in practice there is not much to be gained with cycles longer than 2 minutes for most intersection layouts in use. For demands near saturation, it is usually better to just saturate the cycle at a value close to 2 minutes instead of using very large cycle times. Also notice that if $Y > 1$, demand is higher than capacity, and it is impossible to avoid the formation of queues. Equation (C.1) would result in negative cycles for $Y > 1$. In practice, when the sum of flow ratios Y approaches 0.9, (C.1) results in impractically large cycles, and queues become almost inevitable unless traffic is very regular.

C.2 SIMULATION RESULTS

Experiments were performed on the same cross intersection presented in Section 4.1. Vehicles behave according to the car following model implemented on the Aimsun simulator (Transport Simulation Systems, 2015), which is similar to the Gipps car following model (GIPPS, 1981). All vehicle parameters are left at the default value.

The control plan is composed by four phases, each one allowing all movements that originate in one approach (without dedicated turning lanes, there is no reason to have left turn specific phases or other possible configurations).

In order to design a control plan, assume that drivers take 1.5 s to react to a green light. The plan is defined as having 2 s of yellow time and 1 second of clearance red after each phase. Half of the yellow time is assumed to be used by drivers. As such, there should be 3.5 seconds of lost time[1] for each phase, and total lost time is 14 s. Road capacity of each lane is assumed to be 1800 veh/h, which is in line to a typical urban road (experimental results show that, in the modeled conditions, capacity is between 1800–1820 veh/h/lane).

If demand at each approaching lane is 400 veh/h (i.e., the same as in the scenarios refereed to as low demand scenarios in Chapter 4), each phase has saturation $y_i = 0.2222$, which means $Y = 0.8889$. In this case, (C.1) results in an optimal cycle $c_0 = 288$ s, which is very long. A cycle length of 152 s was arbitrarily chosen as a maximum value for cycle time.

The resulting plan has 35 s of green time for each phase. It was evaluated in simulation by executing 10 replications, during 30 minutes each. Under these conditions, average vehicle delay was observed to be 167 seconds, and average speed was 8.66 km/h. Recall vehicles take roughly 50 seconds to cross the network if not subjected to any delay, meaning that with the semaphoric control and this demand they take more than 4 times longer to be able to cross the intersection than if they were the only vehicle in the network. Traffic performance is clearly much worse than under OATS, which achieves delays smaller than 1 s and speeds close do 30 km/h (the speed limit) for the same demand.

The demand of 400 veh/h/lane is very close to saturation (i.e., close to capacity), which can compromise the efficiency of semaphoric control. For a comparison of semaphoric control operating in unsaturated conditions, consider a slightly lower traffic demand, of 320 veh/h/lane. For this demand, $y_i = 0.1778$ for each phase, and $y_i = 0.7111$. Optimal cycle is $c_0 = 90$ s. Experimental results show that for this demand and resulting control plan (actually rounded up for a cycle length of 92 s), average vehicle delay is 91.1 s, and average vehicle speed is 12.93 km/h. This performance is much better than when demand is 400 veh/lane, but delay is still more than one order of magnitude

[1] This configuration is actually somewhat risky. Most intersections have a larger clearance red

worse than what can be achieved by OATS. Recall the "default" scenario considered for OATS results in average delays of 0.69 s and average speeds of 29.33 km/h for a demand of 400 veh/h, and average delays of 2.46 s and average speeds of 28.36 km/h for a demand of 800 veh/h. I.e., the OATS strategy results in much better traffic conditions than traffic lights, even when considering much higher demands for OATS than for traffic lights.

Table 16 summarizes the simulation results with fixed time semaphoric control.

Table 16 – Simulation results with fixed time semaphoric control

demand (veh/h/lane)	cycle (s)	Y	delay (s/veh)	speed (km/h)
400	152	0.8888	167.06	8.66
320	92	0.7111	91.1	12.93