# Capitulo 5: Auto-organización de semáforos

*En este capitulo[[1]](#footnote-1), la metodología presentada en el capítulo previo es aplicada a la auto-organización de semáforos. Una simulación multi-agente es utilizada para estudiar tres nuevos métodos de auto-organización, que están disponibles para superar a los métodos rígidos tradicionales y adaptativos. Usando reglas simples y sin comunicación directa, los semáforos permiten la auto-organización y adaptación a cambios en las condiciones de tráfico, reduciendo los tiempos de espera, el número de autos parados e incrementando la velocidad promedio. Una simulación más realista fue desarrollada para el modelo de una avenida de Bruselas (Cools, 2006), donde los métodos de auto-organización también superan el actual método “ola verde”.*

## 5.1 Introducción

Alguno ha vivido en áreas pobladas que sufren de congestión. El tráfico consume tiempo, energía y paciencia. Lo que ha motivado a la gente a regular el flujo de tráfico con el fin de regular la congestión. La idea es simple: Si los vehículos permite viajar en una dirección, habrá una alta probabilidad de que unos obstruya a otro. Para evitar esto, se han introducido reglas para mediar (Heylighen, 2003a) entre los vehículos en conflicto, por restricciones o limitaciones de su comportamiento. La gente ha comprendido en que lado de la calle debe manejar (izquierda o derecha); los carriles permiten que los autos no tomen más espacio del necesario; las señales y códigos de tráfico regulan un comportamiento adecuado; y los semáforos regulan el cruce en las intersecciones.

Esto no es una solución al problema de congestión cuando la densidad de autos satura las calles, pero hay muchas maneras en la cual el flujo de autos puede ser limitado con el fin de mejorar el tráfico. Los semáforos no son los únicos componentes que se deben tomar en cuenta, pero si son un factor importante. Afirmamos que el sistema de semáforos puede ser más eficiente si para una densidad de carros dada, el sistema incrementa la velocidad de los vehículos. Esto se refleja en que los autos esperarán menos tiempo cuando hay luz roja, por lo que consumen menos energía y provoca menos contaminación (ya que un auto gasta más energía acelerando y frenando constantemente que si viaja a una velocidad constante).

Por décadas las personas han usado métodos matemáticos y computacionales para encontrar periodos y fases (es decir, ciclos) apropiados para los semáforos, ya que se considera que las variables deben ser optimizadas. Esto es bueno porque una sincronización exacta es mejor que no tener correlación de fases. Sin embargo muchos de los métodos que se aplican hoy en día no consideran el estado actual de tráfico. Si los autos viajan mucho más lento de lo esperado, esto puede resultar en la pérdida de fases. Si ellos van muy rápido, entonces tienen que esperar hasta que la luz verde les toque de nuevo. La optimización puede ser perjudicada por una situación “anormal”, como por ejemplo que muchos vehículos al mismo tiempo lleguen o permanezcan en cierto lugar, tal como sucede en un estadio, después de un partido, en un distrito financiero antes de la hora de entrada, o en una universidad un viernes por la noche. En la mayoría de los casos, los agentes de tránsito remplazan a los semáforos y personalmente regulan el tráfico. Sin embargo, el modelado de tráfico ha mejorado enormemente la comprensión de este fenómeno, especialmente dentro de la última década (Prigogine and Herman, 1971; Wolf et al., 1996; Schreckenberg and Wolf, 1998; Helbing et al., 2000; Helbing, 1997; Helbing and Huberman, 1998), reflejada en distintas mejoras en la infraestructura de tráfico.

Sostengo que el control de semáforos no es tanto un problema de optimización, sino más bien un problema de adaptación, debido a que el flujo y la densidad vehicular cambian constantemente. La optimización da la mejor solución posible para una configuración dada. Por ejemplo, en el diseño de alas, la optimización ofrece una buena solución, ya que las demandas de aerodinámica para un aeroplano es constante, es decir para una cierta velocidad, peso, etc. Pero dado que en el tráfico real, la configuración cambia constantemente, vemos que en algunos casos podría ser mejor un mecanismo adaptativo que un mecanismo que es óptimo solo en algunas ocasiones, en otras puede crear estragos. En realidad los sistemas modernos “inteligentes” de administración de tráfico (ATMS), utilizan métodos de aprendizaje para adaptar las fases de los semáforos, normalmente utilizan una computadora central (Federal Highway Administration, 1998; Hunt et al., 1981).[[2]](#footnote-2) Otra razón para preferir un método adaptativo es que la optimización puede ser computacionalmente costosa. Tratar de encontrar todas las posibles soluciones optimas de una ciudad, no es fácil, ya que el espacion de configuración es enorme, incierto y cambia constantemente.

En este capitulo, un trabajo reciente sobre auto-organización de semáforos (Gershenson, 2005; Cools, 2006) será utilizado para ilustrar el paso por las distintas etapas de la metodología. Estos semáforos son llamados semáforos auto-organizados porque el desempeño global es dado por reglas locales seguidas por cada semáforo: ellos desconocen el estado de las otras intersecciones y sin embargo se organizan para lograr la coordinación global. En las siguientes secciones, los pasos propuestos por la metodología son alternados con resultados de simulaciones de los experimentos llevados a cabo en un ordenador, continuamos con una discusión y conclusiones de el capitulo.

## 5.2 Aplicando la Metodología I

**Requerimientos.** La meta es desarrollar un sistema de control de semáforos eficiente y factible.

**Representación.** El sistema de control de semáforos puede ser modelado en dos niveles: al nivel de vehículo y a nivel de ciudad. Esto es fácil de identificar porque los vehículos son objetos que se mueven a través de la ciudad estableciendo claras distinciones espaciotemporales. La meta de los vehículos es fluir tan rápido como sea posible, por lo que su satisfacción σ puede ser medida en términos de su velocidad promedio y el tiempo promedio de espera en la luz roja. Los autos tendrán una σ máxima si viajan tan rápido como les sea permitido, y no se paran en las intersecciones. σ es cero si el auto se para indefinidamente. La meta del sistema de semáforos en el nivel de ciudad es permitir que los vehículos fluyan tan rápido como les sea posible, mientras median sus conflictos de espacio y tiempo en las intersecciones. Esto podría disminuir el consumo de combustible, ruido, contaminación y estrés en la población. La satisfacción de la ciudad puede ser medida en términos de la velocidad promedio y tiempo promedio de espera de todos los vehículos (es decir, promedio de ), y con el porcentaje promedio de carros estacionados. será máximo si todos los carros van tan rápido como les es posible, y el flujo en la ciudad está disponible sin paradas. Si ocurre un embotellamiento y todos los vehículos se detienen indefinidamente, entonces será mínima.

**Modelado.** Ahora puede ser formulado el problema para el control: encontrar un mecanismo que coordiné los semáforos para que mediar entre los vehicular y se pueda reducir su fricción (es decir, tratar de prevenir que arriben al mismo tiempo a un cruce).[[3]](#footnote-3) Esto maximiza la satisfacción de los vehículos y de la ciudad (’s y ). Ya que todos los vehículos contribuyen igualmente a , idealmente el control debe minimizar la fricción vía compromiso.

**Simulación.** Importantes simulaciones de tráfico usan autómatas celulares para modelar el tráfico efectivamente (Faieta and Huberman, 1993; Biham et al., 1992; Nagel and Schreckenberg, 1992; Chowdhury and Schadschneider, 1999), ya que es un método computacionalmente barato. Sin embargo el incremento del poder de computo en los últimos años ha permitido el desarrollo de simulaciones multi-agente para crear simulaciones de tráfico más realistas (Nagel, 2004; Wiering et al., 2004; Miramontes Hercog, 2004; Roozemond and Rogier, 2000). En consecuencia una simple simulación fue desarrollada en NetLogo (Wilensky, 1999), un ambiente de modelación multi-agente. El modelo de “estancamiento” (Wilensky and Stroup, 2002) fue implementado para implementar diferentes estrategias de control de tráfico. Esto consiste en un malla abstracta de tráfico con intersecciones entre ciclos de arterias de un solo carril y dos tipos: verticales u horizontales (similar a los escenarios de Biham et al. (1992) and Brockfeld et al. (2001)). Los carros fluyen en línea recta, hacía el este o sur. Cada cruce tiene un semáforo que permite fluir el tráfico en la arteria con intersección de color verde. Las luces roja y amarilla detienen el tráfico. La secuencia de luces dada para una arteria es verde-amarilla-roja-verde. Los autos simplemente tratan de ir a su máxima velocidad de 1 “pieza” por paso del tiempo, pero se detiene cuando un carro o una luz roja o amarilla esta enfrente de ellos. El tiempo es discreto pero el espacio no. Los autos usan una aceleración de 0.099 para arrancar o para frenar. Una “pieza” es un cuadro en el ambiente, del tamaño de un auto.

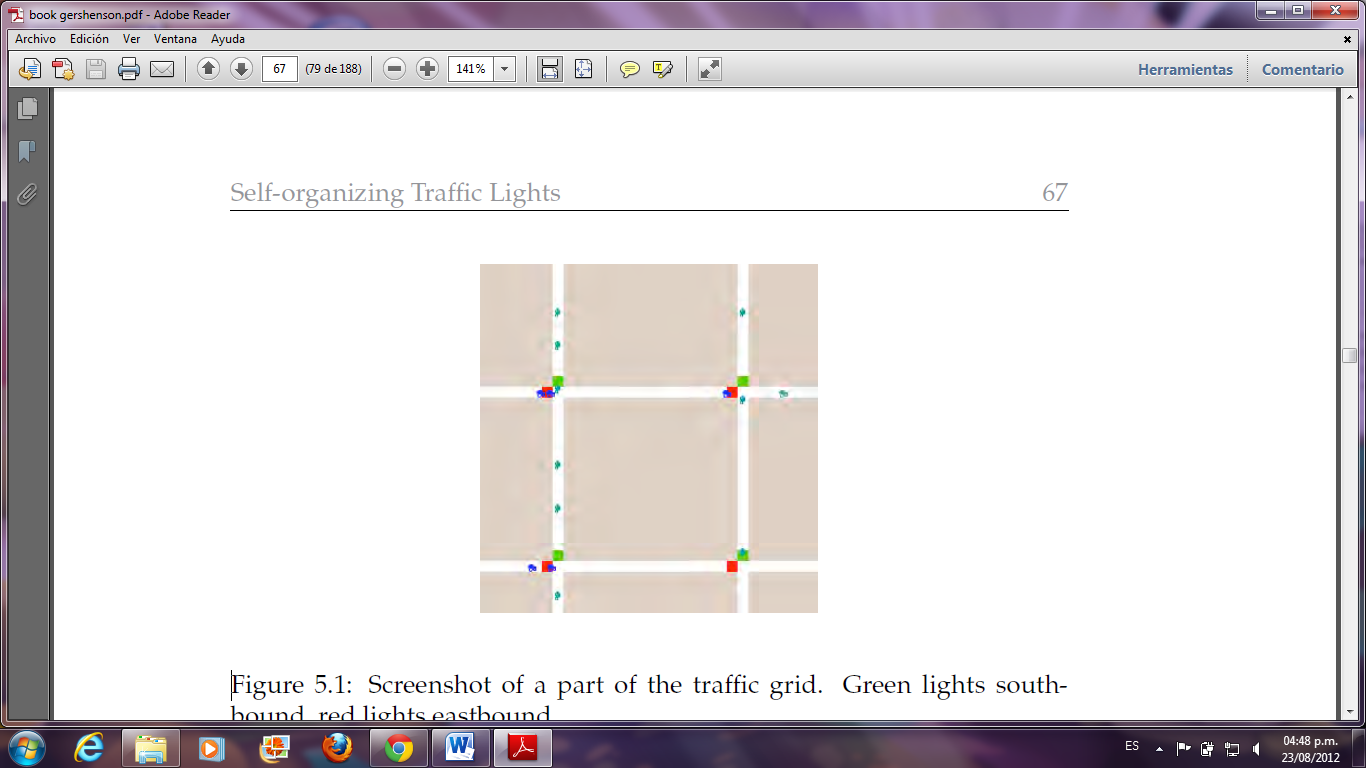


Figura 5.1: Captura de pantalla de una parte de la malla de tráfico. Las luces rojas van hacia el sur y las luces verdes hacia el este

Una captura de pantalla de una parte del ambiente se pude ver en la figura 5.1. El usuario puede cambiar distintos parámetros, como el número de arterias o de autos. Se muestran diferentes estadísticas: el número de autos detenidos. El lector está invitado a probar la simulación (incluyendo el código fuente), con ayuda de un buscador de internet con Java-activado en SOLT (2005).

A inicio, un modelo tentativo fue implementado. La idea no funciono. Sin embargo, después de refinar el modelo, se encontró un método eficiente llamado *sotl-request*.

**Modelado.** En el método sotl-request, cada semáforo mantiene un contador *ki* del número de autos que pasan por intervalo de tiempo (*c\*ts*) que se acercan a la luz roja, independientemente del estatus o la velocidad de los carros (es decir en movimiento o detenidos), desde una distancia *ρ*. *ki* puede ser vista como la integral de autos que esperan o se aproximan sobre el tiempo. Cuando *ki* alcanza un umbral θ, la luz verde opuesta cambia a amarillo, y en el siguiente paso de tiempo a rojo con *ki*=0, mientras la luz roja que contó cambio a verde. De esta forma, si hay más de un auto aproximándose o esperando antes de la luz roja, la luz cambiara a verde más rápido que si hay pocos autos. Este simple mecanismo es descrito formalmente en el algoritmo 1, logra la auto-organización de la siguiente manera: si hay sólo uno o unos pocos vehículos, estos estarán parados en la luz roja por un periodo más largo. Dando tiempo a que se les unan más autos. Mientras más autos se unan al grupo, los autos esperarán periodos más cortos en el semáforo rojo. Una vez que haya suficiente autos, la luz roja cambiará a verde incluso antes de que los coches que se paren en la intersección “lo que genera la ola verde” bajo demanda. Teniendo “pelotones” o “convoys” de carros moviéndose juntos mejora el flujo de tráfico, en comparación con una distribución homogénea de autos, ya que hay una gran área vacía entre los pelotones, que puede ser utilizada para cruzar pelotones con pocas interferencias. El método de solt-request no tiene fase de reloj interno. Los semáforos cambian solo cuando las condiciones mencionadas se reúnen. Si lo autos no se aproximan a la luz roja, la luz complementaria puede permanecer en verde.

**paraCada** (paso de tiempo) **has**

*ki* += *carrosaproximandoseAlRojo*en *ρ*;

**si** (*ki* ≥ *θ*) **entonces**

*cambiaLaLuzi*();

*ki* = 0

**fin**

**fin**

**Representación.** Se vuelve clara ahora que puede ser útil considerar también a los semáforos como agentes. Su meta es ahora deshacerse de los autos tan rápido como sea posible. Para ello, deben evitar tener luces verdes en calles vacías y rojas en calles con alta densidad de tráfico. Ya que la satisfacción de los semáforos y los vehículos es complementaría, ellos deben interactuar via Cooperación para lograr la sinergia. Además, *σsys* puede ser formulada en términos de la satisfacción de los semáforos, los vehículos o de ambos.

**Modelado.** Dos métodos clásicos fueron implementados para comparar su desempeño con *sotl-request: marching* y *optim*.

*Marching* es un método muy simple. Todos los semáforos “marchan en un paso de tiempo”: todas las luces están sincronizadas en tiempo ya sea en dirección sur ó este. Las intersecciones tienen una fase ϕi, que cuenta los pasos de tiempo. ϕi se reinicia a cero cuando la fase alcanza un periodo de valor *p*. Cuando ϕi==0, la luz roja cambia a verde, y la luz amarilla cambia a rojo. Las luces verdes cambian a amarillo en un paso de tiempo anterior, es decir cuando ϕ*==p-1*. El ciclo completo de una intersección consiste en *2p* pasos de tiempo. Las intersecciones *“Marching”* de manera que .

El método *optim* es implementado tratando de poner las fases de los semáforos de modoque, tan pronto como la luz roja cambie a verde, un carro que tuvo que detenerse pueda seguir las luces verdes. En otras palabras, una solución fija es encontrada cuando la *ola verde* fluye hacía el sureste. El ambiente de simulación tiene un radio de r parches cuadrados, que pueden ser identificados con las coordenadas . Mas aún, cada arteria consiste de 2r+1 parches. En orden para sincronizar todas las intersecciones, las luces rojas deben cambiar a verde y las luces amarillas deben cambiar a rojo cuando

y las luces verdes deben cambiar a amarillo en el paso de tiempo anterior. El periodo debe ser . El tres se agrega como un margen extra para la reacción y los tiempos de aceleración de los automóviles (el mejor se encuentra, para bajas densidades, a prueba y error).

Estos dos métodos son no adaptativos, en el sentido de que sus comportamientos son dictados de antemano, y no conciderán el actual estado de tráfico. Por lo tanto, no puede darse Cooperación en vehículos y semáforos, ya que estos últimos tienen un comportamiento fijo. Por otro lado los semáforos debajo del método de *solt-request*, son sensibles a las actuales condiciones de tráfico, y pueden por lo tanto responder a las necesidades de los vehículos entrantes.

…..

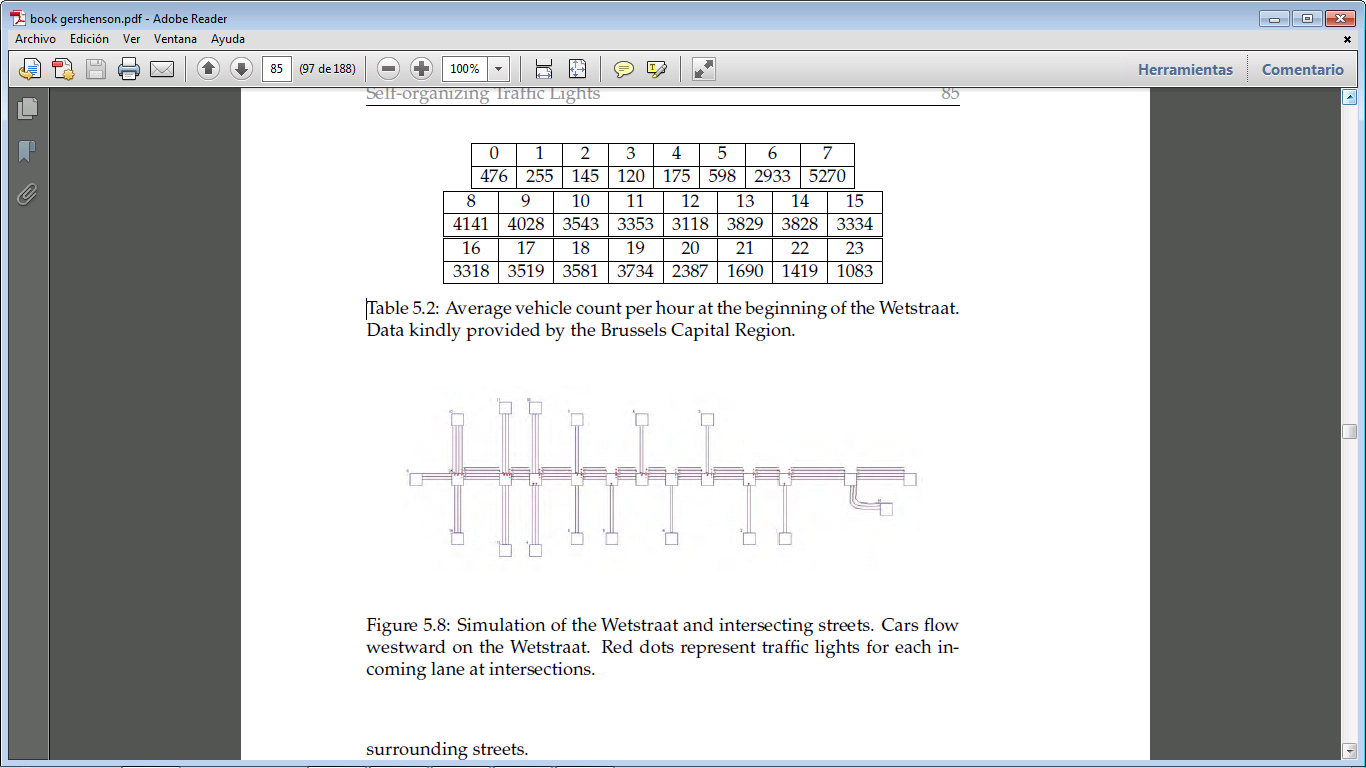
# 5.6 Aplicando la Metodología III

Los resultados mencionados a continuación fueron presentados en el gabinete del Ministerio de Movilidad y Trabajos Públicos de la Región de Bruselas, Pascal-Smet, en Agosto del 2005. El resultado fue alentador, pero las simulaciones sin embargo son abstractas. La Región de Bruselas tiene datos disponibles de la calle Loi/Wetstraat, un carril cuádruple hacia el oeste avenida de un sentido en Bruselas que reúne tráfico pesado hacía el centro de la ciudad. Para su tesis BSc, mi estudiante Seung Bae Cools presento un simulador de tráfico para pruebas de métodos solt en simulaciones más realistas (Cools, 2006). Sus resultados se resumen a continuación.

**Simulación.** Nuestro simulador moreVTS (2006) (Un Auto más Realista Simulador de Tráfico) es el tercero de una serie de proyectos open source construidos sobre otros previos, desarrollado en Java. Green Light District (GLD (2001)) fue desarrollado por el Grupo de Sistemas Inteligentes de la Universidad de Utrecht (Wiering et al., 2004). Entonces, GLD fue perfeccionado por estudiantes en Argentina dentro del proyecto iAtracos, que hemos usado como punto de partida para nuestro simulador, con una presentación física realista dentro del simulador. Entre otras cosas, la aceleración fue introducida, y la escala fue modificada de manera que un pixel representa un metro y un ciclo representa un segundo.

El simulador permite modelar configuraciones complejas de tráfico, permitiendo a los usuarios crear mapas y después correr simulaciones variando la densidad y tipos de carreteras usadas. Calles de múltiples carriles e intersecciones pueden ser organizadas, así como generar destinos y frecuencias de autos. Para más detalles de la implementación de moreVTS, favor de referirse a Cools (2006).

Los métodos *sotl* son implementados en moreVTS. Con datos dados por la Región Capital de Bruselas, hemos sido capaces de construir una simulación detallada de Wetstraat. Usamos medidas de densidades promedio de tráfico por hora en días de trabajo del 2004 (mostradas en la tabla 5.2) y el actual método “ola verde” (una adaptación de *optimo*) que tiene un periodo de 90 segundos, con 65 segundos para la luz verde en Wetstraat, 19 para fase verde en la calles de los lados, y 6 para las transiciones. Con esto, nosotros fuimos capaces de comparar nuestros controladores auto-organizados con un estándar en configuraciones realistas. La figura 5.8 muestra la vista de simulación de Westraat y sus calles circundantes.



Los datos de la tabla 5.2 son los carros entrando a Weatraat por el Este (en el puente sobre Etterbeeksesteenweg), entonces las tasas generadas por los dos nodos en la simulación, que representan esto, fueron configuradas de acuerdo a estos datos. Para los otros nodos, las frecuencias generadas y destino fueron configurados en base a un campo de estudio desarrollado en Mayo del 2006, comparando el porcentaje de autos que fluyen de Wetstraat y los que fluyen a través de la calle, entrando, o se mantienen en Wetstraat. Estos porcentajes se mantienen constantes, de manera que cuando la densidad de carros entrante a Wetstraat cambia, las demás tasas generadas cambian en la misma proporción. En promedio, por cada 5 autos fluyendo a través de la calle, cien fluyen a través de Wetstraat. Este no es el caso de Kunststraat, una avenida de dos vías en el oeste de la Wetstraat (cruce de calles segunda y tercera de izquierda a derecha figura 5.8), donde para 100 coches que circulan por Wetstraat, en 40 vueltas a la derecha e izquierda, respectivamente, y sólo 20 van en línea recta, mientras 20 más manejan a través de Kunststraat (alrededor de 10 en cada dirección). La tasa exacta generada y destinos de las frecuencias están dados en Cools (2006, pp. 55-57).

## 5.7 Experimentos: Resultados número tres

…..

# 5.8 Aplicando la Metodología IV

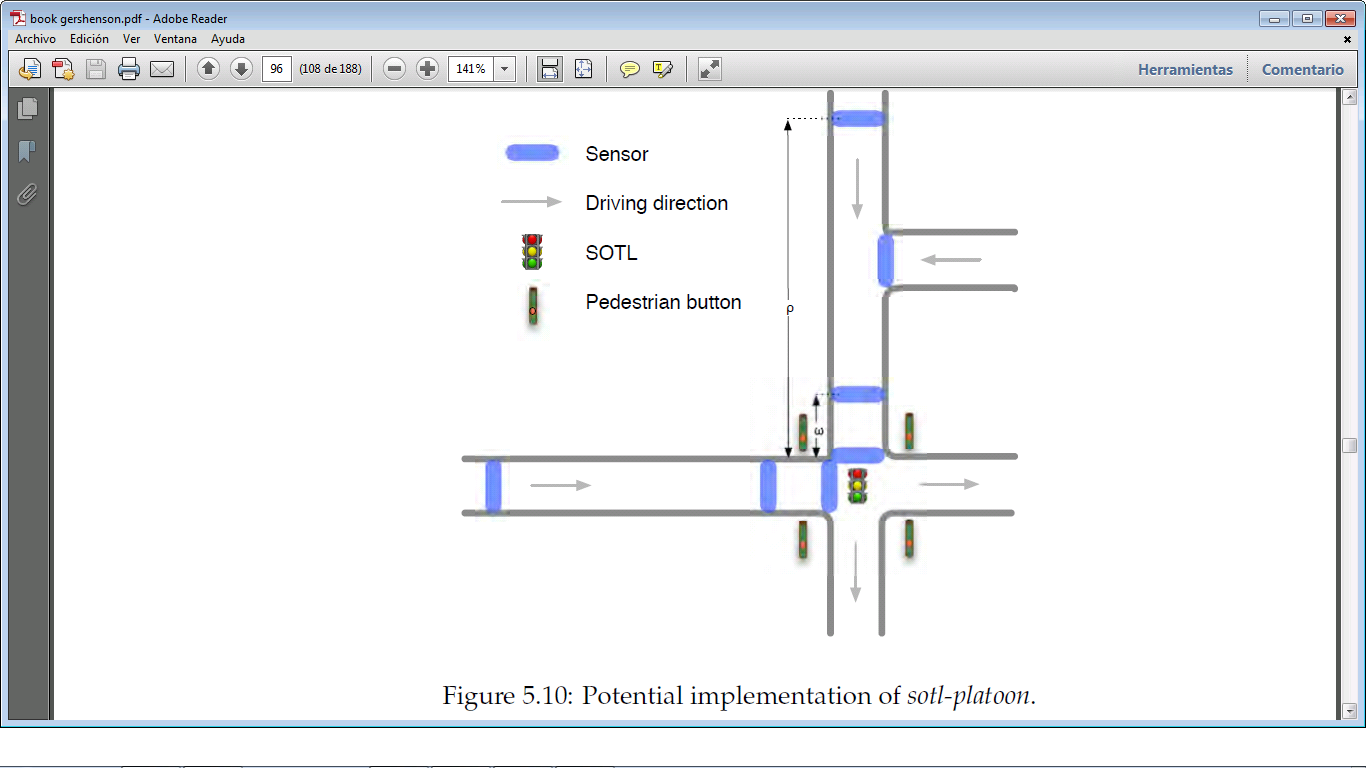
**Representación.** Si la prioridad debe darse a determinados vehículos (como transporte público o vehículos de emergencia), los pesos pueden ser agregados dando mayor importancia a algunos . El seguirá siendo el promedio de los individuales, sólo que ahora ponderado con la prioridad de los vehículos.

Un nivel medio puede ser considerado, donde las propiedades los pelotones pueden ser observadas: su comportamiento, desempeño, y satisfacción y las relaciones de sus relaciones con los vehículos y los niveles de la ciudad podría mejorarse la comprensión de la auto-organización de semáforos e incluso mejorarla.

**Aplicación.** El sistema propuesto, aun no ha sido implementado. Sin embargo, es factible hacerlo, ya que hay tecnología en sensores para implementar el método discutido de manera asequible, y las simulaciones han provisto los beneficios de los métodos de auto-organización.

La figura 5.10 muestra una posible configuración para implementar el *solt-platoon* usando simples sensores de proximidad, que deben instalarse debajo del pavimento. Sensores más complicados, tales como cámaras, también pueden ser utilizados. Un sensor a una distancia *ρ*, del semáforo puede ser usado para comenzar el conteo de autos. Un controlador podría conocer el número de autos que se aproximan a la a luz roja, incrementado un contador *c* en cada paso de tiempo que un carro pasa a través del sensor en *ρ*, y disminuir *c* cuando los carros crucen la intersección. Cada cierto tiempo, como un segundo, c sería integrado en el contador *ki* (ver algoritmo 3). La misma idea podría ser usada para contar cuantos carros están entre *ω* y la intersección. Si los carros quieren entrar o permanecer en la calle en algún lugar entre el sensor y la intersección, por ejemplo en un estacionamiento de una pequeña calle, un sensor adicional podría ser instalado para ajustar *c*.

Los peatones pueden ser tomados en cuenta, por considerarlos como autos que se acercan a un semáforos rojo. Botones como los ya disponibles en el mercado pueden ser usados para incluir a los peatones en *ki*, restaurándolo después de cada cambio de luces.



Un estudio piloto debe ser hecho antes de aplicar extensamente la auto-organización de semáforos, para afinar diferentes parámetros y métodos. Factores externos, como peatones, ciclistas, podrían también afectar el desempeño del sistema.

Una estrategia mixta, entre diferentes métodos podría considerarse, como *solt-platoon* para baja y media densidad, y *solt-phase* o *marching* para altas densidades. Otra alternativa valiosa para explorar puede ser variar *θ* de acuerdo a la actual densidad de tráfico, o probar diferentes *θ’s* para diferente direcciones para densidades de tráfico altamente sesgadas, como en Westraat (calle principal de Bruselas). La relación entre *θ* con *ρ* y *ω* también debe ser estudiada.

**Evaluación.** Si una ciudad despliega un sistema de semáforos auto-organizados, debe ser monitoreado y comparado con sistemas previos. Esto ayudará a mejorar el sistema. Si el sistema tiene beneficios económicos, su implementación en otras ciudades debe ser promovida, especialmente por que los mecanismos y simulaciones están abiertos a la comunidad.

1. Basado en Gershenson (2005); Cools et al. (2007). [↑](#footnote-ref-1)
2. El inconveniente de los ATMS es su alto costo y complejidad que requiere mantenimiento de especialistas. Aun no hay un estándar, y usualmente las compañías se contratan para diseñar soluciones particulares particulares para diferentes ciudades. La naturaleza privada de los ATMS también los hace difíciles de estudiar comparados con métodos alternativos. [↑](#footnote-ref-2)
3. Como se menciono en la sección 1.3.2, los controladores auto-organizados de semáforos han sido desarrollados antes que la metodología. A pesar de ello, el concepto de fricción es usado aquí, entendiendo mejor el mecanimos e ilustrando la metodología. [↑](#footnote-ref-3)