

Validación Inteligente para la Sincronización de Semáforos Basada en Feature Models

Javier Ferrer, José García-Nieto, Enrique Alba, and Francisco Chicano

Dept. de Lenguajes y Ciencias de la Computación,
Universidad de Málaga, E.T.S.I. Informática,
{ferrer,jnieto,eat,chicano}@1cc.uma.es

Resumen El concepto de *Smart City* o Ciudad Inteligente engloba el conjunto de acciones y servicios, basados en las tecnologías de información y comunicación, que se ofrecen en un núcleo urbano. En este sentido, el uso de técnicas bioinspiradas para la gestión del flujo del tráfico mediante la sincronización de semáforos podría constituir uno de los aspectos más innovadores en los entornos urbanos en el futuro. No obstante, la programación automática de semáforos requiere además de un proceso de validación de las soluciones generadas, dado que afectan a la seguridad de miles de usuarios. En este trabajo se propone una estrategia de validación basada en los Modelos de Características (*Feature Models*) para generar automáticamente diversos escenarios para la comprobación de la robustez de los programas de semáforos. Como caso de estudio, se realiza la validación de programas de semáforos en el área urbana de la ciudad de Málaga, generados mediante cuatro algoritmos de optimización (PSO, DE, Random Search y SCPG). El resultado es información validada sobre el programa de semáforos que mejor actúa para un mayor porcentaje de situaciones de tráfico diferentes.

Keywords: Modelos de Características, Metaheurísticas, *Pairwise Testing*, Priorización, Semáforos

1. Introduction

Hoy en día, todas las iniciativas para el desarrollo de la Ciudad Inteligente o *Smart City* [1] reciben un gran interés institucional y mediático por el impacto que generan sobre la sociedad en general. El concepto de *Smart City* engloba una serie de acciones y servicios coordinados, cuyo objetivo es conseguir la mejora de eficiencia a nivel energético, el consumo y la utilización sostenible de recursos, así como la aparición de mejores procesos operativos. En este sentido, la gestión del flujo del tráfico mediante la sincronización de semáforos constituye uno de los aspectos más sensibles en los entornos urbanos. En la actualidad existen pocas propuestas basadas en técnicas inteligentes para la sincronización de semáforos que obtienen soluciones eficientes [2,3,4]. No obstante, dichas soluciones están determinadas por instancias o escenarios de prueba con condiciones muy específicas y de variabilidad limitada. Por tanto, disponer de una estrategia de validación para diversos escenarios es crucial para este tipo de procesos, ya que determinarían la robustez de las soluciones aportadas.

Con esta motivación, proponemos en este trabajo una estrategia de validación basada en Modelos de Características (*Feature Models* - FM) [5], que sirven para representar adecuadamente sistemas de alta variabilidad. En este trabajo hemos desarrollado un modelo de gestión de tráfico, el cual recoge toda la variabilidad de características (posibles situaciones de tráfico) y la plasma en diferentes escenarios de movilidad urbana.

Entre las principales características que afectan al tráfico podemos enumerar: diferentes condiciones meteorológicas, tipología de vehículos, grado de pericia de los conductores, tiempo de simulación, etc. No obstante, el análisis de todos los posibles escenarios de tráfico es inviable debido a la gran cantidad de combinaciones de características a tener en cuenta. Por tanto, vamos a utilizar *Prioritized Genetic Solver* (PGS) [6], un algoritmo evolutivo de optimización diseñado para generar el conjunto mínimo de configuraciones o escenarios que cumplen el criterio de cobertura *pairwise*. Además, no todas las características tienen la misma probabilidad de ocurrencia o importancia, por lo que nuestro modelo de características ha sido ampliado con asignación de prioridades a características.

Como caso de estudio, vamos a centrarnos en un área urbana de Málaga. Para esto, utilizaremos el simulador de tráfico SUMO (*Simulator of Urban Mobility*) [7]. SUMO ofrece una fuente de información constante sobre el flujo de vehículos con: velocidades, consumos, emisiones, tiempos de viaje, etc.; y nos permite configurar escenarios realistas (como en el caso de Málaga) siguiendo direcciones y patrones reales de movilidad. En concreto, vamos a comprobar la robustez de programas de semáforos, previamente generados mediante cuatro algoritmos de optimización: *Particle Swarm Optimization* (PSO) [8], *Differential Evolution* (DE) [9], *Random Search* (RS) y *SUMO Cycle Programs Generator* (SCPG) [2]. Nuestro modelo de validación nos ayudará a seleccionar cual (o cuales) de estas soluciones es la más adecuada para un mayor número de escenarios diferentes de tráfico.

Las principales contribuciones de este artículo son las siguientes:

- Desarrollo de un Modelo de Características priorizado para la representación de un sistema de gestión (semafórica) de tráfico con características ponderadas acorde a su importancia o probabilidad de ocurrencia.
- Generación automática de escenarios ponderados para la validación de programas de semáforos por medio del algoritmo PGS.
- Análisis de los programas de semáforos para el caso de estudio de la ciudad de Málaga previamente generados por los algoritmos de optimización (PSO, DE, Random Search, SCPG) en dichos escenarios.

Este artículo se organiza de la siguiente manera. La Sección 2 presenta los conceptos previos sobre representación de variabilidad en Modelos de Características. La sección 3 describe el algoritmo PGS para la generación de escenarios de validación a partir del FM. La Sección 4 detalla el procedimiento experimental llevado a cabo y los análisis de resultados. Por último, en la Sección 5 se enumeran las conclusiones y el trabajo futuro.

2. Representación de Sistemas con Alta Variabilidad

Los modelos de características son un estándar *de facto* para modelar las características comunes y variables de un sistema (representadas por cajas etiquetadas) y sus relaciones (representadas con líneas) formando una estructura de tipo árbol, especificando así un conjunto de combinaciones de características que dan lugar a múltiples configuraciones diferentes [5]. Cada característica, aparte de la raíz, tiene una sola característica padre y puede tener un conjunto de características hijas. Nótese que una característica hija sólo puede ser incluida en una configuración si y sólo si, su padre es incluido también. Para ilustrar estos conceptos vamos a usar un FM (Figura 1) que hemos generado para representar la gestión de tráfico. Entre las principales características contamos con diferentes condiciones meteorológicas, tipología de vehículos, grado de pericia y reacción de los conductores, tiempo y cantidad de vehículos en la simulación. Por todo esto, el modelo representa 960 escenarios válidos diferentes, y éste nos servirá para generar un subconjunto de escenarios adecuado para testar los programas de semáforos.

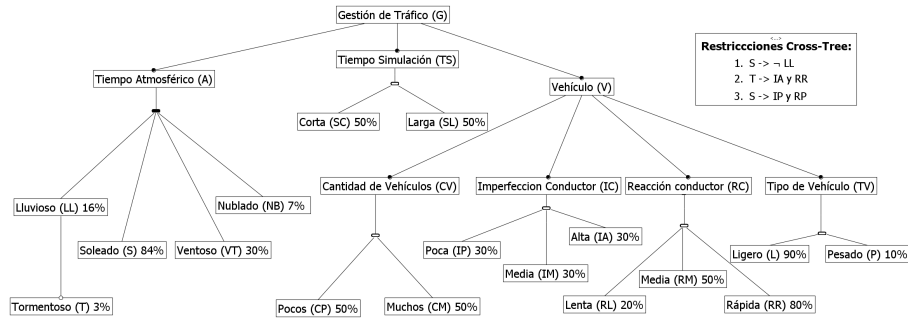


Figura 1: Modelo de gestión de tráfico con prioridades

Hay 4 tipos de relaciones jerárquicas entre características:

- *Características opcionales*: son representadas con un círculo vacío e indica que esta característica puede o no ser seleccionada si su padre es seleccionado. En nuestra instancia sería la característica “tormentoso” (T).
- *Características obligatorias*: son representadas con un círculo relleno, y deben ser seleccionadas si su padre es seleccionado. En nuestra instancia serían obligatorias: G, A, TS, V, CV, IC, RC y TV.
- *Relaciones Or-inclusivas*: son representadas como óvalos independientes rellenos que abarcan un conjunto de líneas que conectan la característica padre con las características hijas. Indican que al menos una característica debe ser seleccionada si el padre es seleccionado. En nuestra instancia, la relación de A con sus hijos es de este tipo.
- *Relaciones Or-exclusivas*: son representadas como óvalos independientes vacíos que abarcan un conjunto de líneas que conectan la característica padre con

las características hijas. Indican que exactamente una característica debe ser seleccionada si el padre es seleccionado. En nuestra instancia las relaciones de TS, CV, IC, RC y TV con sus respectivos hijos son Or-exclusivas.

Además de las relaciones padre-hijo, las características se pueden relacionar también con otras ramas del modelo de características, son las restricciones conocidas como *Cross-Tree Constraints (CTC)* [10]. Estas restricciones, así como las impuestas por las relaciones jerárquicas entre características, son expresadas y comprobadas usando lógica proposicional. Para el FM analizado en este trabajo hemos indicado las tres restricciones que se consideran en el área superior derecha de la Figura 1.

Estos modelos de características pueden ser ampliados para tener en cuenta una ponderación asociada a cada característica. En nuestra instancia, indicamos el peso de cada característica opcional en porcentaje, porque las características obligatorias siempre deben estar presentes (no necesitan ser ponderadas). A continuación resumimos la terminología básica que usamos durante este trabajo.

Una configuración o escenario es un par $[sel, \overline{sel}]$ donde sel y \overline{sel} son respectivamente los conjuntos de características seleccionadas y no seleccionadas. Con respecto a las prioridades, un par priorizado pc está compuesto por dos características f_1 y f_2 , tales que el peso del par es calculado como $pc.w = f_1.w * f_2.w$. Consecuentemente, la prioridad de una configuración es calculada como la suma de los pares priorizados que se generan de ella. Cabe recordar que una configuración es válida para un FM, *sii* dicha configuración no contradice ninguna restricción del FM.

Las pruebas de interacción combinatorias son un enfoque que construye *test suites* que nos llevan a probar de forma sistemática todas las configuraciones de un sistema [11]. En este trabajo vamos a utilizar el enfoque *pairwise*, que es el más utilizado en testing combinatorio y esta basado en la asunción de que la mayoría de errores originados en un parámetro es causado por la interacción de dos valores [12]. Así que este criterio es satisfecho si todos los pares de características $((f_1, f_2), (f_1, \overline{f_2}), (\overline{f_1}, f_2), (\overline{f_1}, \overline{f_2}))$ están presentes en al menos un escenario. Cuando se aplica esta técnica a FMs, la idea es seleccionar un conjunto de escenarios válidos, donde los errores posibles se manifiestan con alta probabilidad, sin tener que probar de forma exhaustiva todas las posibles configuraciones (960 en este caso de estudio). Además, gracias a la priorización, vamos a testar en primer lugar los escenarios con aquellas características que son más frecuentes o importantes en nuestros sistemas.

3. Generador de Escenarios: Prioritized Genetic Solver

El algoritmo *Prioritized Genetic Solver (PGS)* es un algoritmo evolutivo que construye una *test suite* (conjunto de escenarios) teniendo en cuenta prioridades y restricciones durante la generación. Este trabajo adapta y extiende el algoritmo *Genetic Solver* propuesto por Ferrer et al. [6] para trabajar con FMs y restricciones. PGS es un algoritmo genético constructivo que añade una nueva

configuración a la solución parcial en cada iteración hasta que todas las parejas de características hayan sido cubiertas. PGS se ha implementado empleando un framework de desarrollo llamado jMetal [13] y una herramienta para el manejo de FMs llamada FAMA [14]. Revisar la referencia del algoritmo [6] para más detalles.

El FM que vamos a explorar con este algoritmo es el presentado anteriormente en la Figura 1. Para fijar una prioridad a cada una de las características, se han seguido los siguientes criterios: con respecto al tiempo atmosférico se han utilizado datos del Instituto Nacional de Meteorología, concretamente de la estación meteorológica del aeropuerto de Málaga. Sobre la cantidad de vehículos ligeros y pesados, hemos recurrido a datos públicos sobre tráfico de la Delegación de Movilidad del Ayuntamiento de Málaga (<http://movilidad.malaga.eu/>). Los pesos del resto de características se han establecido atendiendo a la importancia de las mismas. Recordamos al lector que las características obligatorias tendrían un peso del 100%, puesto que son obligatorias en toda configuración válida.

Tras la ejecución del algoritmo, PGS ha generado las configuraciones o escenarios que podemos ver en la Tabla 1, cumpliendo con el criterio de cobertura *pairwise*. En dicha tabla podemos ver que se han generado 10 escenarios diferentes para testar. Se han marcado las características que están presentes en cada escenario. En la última columna tenemos el peso total que aporta cada una de las configuraciones a la test suite. Por ejemplo, vemos como el escenario 1 (E1) aporta un 63,59% de peso, mientras que los 4 últimos escenarios (E7-E10) suponen tan sólo alrededor de un 1%. Esto quiere decir que hay que tener en cuenta escenarios diversos, pero corregir las deficiencias en algunos escenarios será mas importante que en otros.

E	G	A	LL	T	S	VT	NB	TS	SC	SL	V	CV	CP	CM	IC	IP	IM	IA	RC	RL	RM	RR	TV	L	P	P %
E1	✓	✓			✓			✓		✓	✓	✓		✓		✓			✓			✓	✓	✓		63,59
E2	✓	✓			✓	✓		✓	✓		✓	✓	✓		✓	✓			✓		✓		✓	✓		22,37
E3	✓	✓	✓				✓	✓	✓		✓	✓	✓		✓			✓	✓			✓	✓	✓		7,09
E4	✓	✓	✓	✓		✓		✓		✓	✓	✓		✓		✓		✓	✓	✓			✓	✓		3,32
E5	✓	✓	✓	✓			✓	✓	✓		✓	✓		✓	✓		✓		✓		✓		✓	✓		1,41
E6	✓	✓			✓		✓	✓		✓	✓	✓	✓		✓	✓		✓	✓				✓	✓		1,21
E7	✓	✓				✓				✓	✓	✓		✓				✓		✓			✓	✓		0,43
E8	✓	✓	✓			✓	✓	✓		✓	✓	✓		✓	✓			✓				✓	✓	✓		0,34
E9	✓	✓	✓	✓		✓		✓	✓		✓	✓	✓		✓		✓		✓	✓			✓	✓		0,23
E10	✓	✓	✓	✓		✓	✓	✓		✓	✓	✓		✓			✓	✓				✓	✓	✓		0,01

Tabla 1: Escenarios computados por el PGS con criterio de cobertura *pairwise*

4. Validación Experimental

En esta sección vamos a validar de forma experimental los programas de ciclos de semáforos generados por una serie de algoritmos (PSO, DE, Random Search y SCPG) [2] para la generación de programas de ciclos en semáforos.



Figura 2: Instancia de Málaga exportada al formato de SUMO

4.1. Caso de Estudio: Málaga

La instancia a partir de la cual se generaron todos los escenarios, así como los programas de ciclos optimizados para los semáforos, corresponde a un área urbana de aproximadamente $0,75 \text{ km}^2$ físicamente localizada en la ciudad de Málaga. Esta instancia consiste en una serie de ficheros con información proveniente de mapas digitales (OpenStreetMap) como: reglas de tráfico regulado, situación de las señales y otros elementos, posición de semáforos, direcciones, calles, intersecciones, glorietas, etc. Además, se ha determinado el número de vehículos circulando, así como sus velocidades a partir de las especificaciones de la Delegación de Movilidad del Ayuntamiento de Málaga. Esta información se obtuvo a partir de puntos sensorizados en ciertas calles de la ciudad calculando así una medida de la densidad de tráfico en varios intervalos temporales.

La ilustración de la Figura 2 muestra el área seleccionada para la ciudad tratada con sus correspondientes capturas de OpenStreetMap y SUMO, describiendo el proceso de generación de la instancia. Situada en la zona entre el centro de la ciudad y el puerto, está compuesta por calles con diferentes anchuras y longitudes, además de varias glorietas. Contiene intersecciones con un número de semáforos entre 4 y 16. El número de intersecciones estudiadas es de 70, con un total de 312 semáforos y entre 250 y 500 vehículos circulando. En nuestro estudio, cada uno de los vehículos recorre su propia ruta desde el origen hasta su destino circulando con una velocidad máxima de 50 km/h (velocidad típica en las áreas urbanas). Las rutas fueron generadas a priori de manera uniforme. El tiempo de simulación es de 500 o 1000 segundos según el escenario.

En cuanto al conductor de un vehículo, hay dos aspectos principales que caracterizan su conducción; el grado de imperfección al volante y el tiempo de reacción ante los estímulos. En SUMO estas características están representadas por 2 parámetros. El parámetro σ es un número real en el rango $[0 \dots 1]$ que pondera la destreza del conductor. Un valor de $\sigma = 1$ representa un conductor con muchas imperfecciones y un valor de $\sigma = 0$ representa el comportamiento de un conductor perfecto. Por otro lado tenemos el parámetro τ , que mide el tiempo de reacción en segundos del conductor ante cualquier estímulo. Además, tenemos otros parámetros como la duración de la simulación, la cantidad de vehículos y el porcentaje de vehículos ligeros frente a pesados. Un vehículo pesado supone una aceleración, desaceleración y velocidad máxima menor (35 km/h) que un vehículo ligero. Las características opcionales seleccionadas a partir de nuestro modelo de gestión de tráfico van a generar escenarios de tráfico con distintas

características. En la Tabla 2 mostramos un resumen de lo que supone activar cada característica.

Características	LL	T	S	VT	NB	SC	SL	CP	CM
Parámetros	$\sigma+ = 0,1$ $\tau+ = 1$	$\sigma+ = 0,1$ $\tau+ = 1$	$\sigma- = 0,1$ $\tau- = 1$	$\sigma+ = 0,1$ $\tau+ = 1$	$\sigma+ = 0,1$ $\tau+ = 1$	500s	1000s	250 u.	500 u.
Características	IP	IM	IA	RP	RM	RA	L	P	-
Parámetros	$\sigma- = 0,1$	-	$\sigma+ = 0,1$	$\tau+ = 1$	-	$\tau- = 1$	95 %vl	85 %vl	-

Tabla 2: Activación de características y peso correspondiente

SUMO genera información acerca de la simulación de una instancia, a partir de la cual calculamos una serie de métricas. En este trabajo usamos las siguientes: vehículos que llegan a su destino (VLL), vehículos que no llegan a su destino (VNLL), CO2 emitido a la atmósfera (CO2), óxidos de Nitrógeno emitidos a la atmósfera (NOx), combustible consumido en el trayecto (FUEL) y tiempo global de viaje(GJT).

4.2. Configuración de los experimentos

Como comentamos anteriormente, hemos seleccionado las soluciones aportadas por 4 algoritmos [2] para someterlos a estudio: PSO en su versión *Standard 2011* [15], DE con mutación *DE/rand/1*, Random Search y SCPG [7]. Los tres primeros algoritmos son no-deterministas, por lo que hemos seleccionado 30 programas de semáforos (soluciones) para cada uno, generadas a priori para el caso de estudio. El cuarto algoritmo es el de la solución propuesta por SUMO (determinista), donde realizamos una única simulación por escenario propuesto (10 simulaciones de E1 a E10). A continuación, en el caso de los algoritmos no-deterministas, para cada escenario propuesto vamos a realizar además 30 ejecuciones independientes más por cada uno de los 30 programas de semáforos generados por los algoritmos. Todo esto suma 27.010 simulaciones independientes (3 algoritmos \times 10 escenarios \times 30 configuraciones de semáforos \times 30 ejecuciones independientes + 10 ejecuciones de SCPG).

4.3. Análisis de los resultados

En la Figura 3 podemos observar el rendimiento de los algoritmos para las métricas VLL, CO2, NOx, Fuel y GJT por vehículo y todos los escenarios. Cada una de las métricas está normalizada para su correcta visualización. En esta figura podemos apreciar que PSO obtiene el máximo número de vehículos que llegan a destino con menor consumo de combustible (Fuel), y arrojando a la atmósfera menor cantidad de CO2 y NOx. A pesar de que el tiempo global de viaje por cada vehículo es máximo para PSO, esto se debe a que también los vehículos con destinos lejanos son capaces llegar a él. En esta figura también podemos apreciar que SCPG es el peor algoritmo porque tiene consumos altos de Fuel, CO2 y NOx, a pesar de que la media de vehículos que llegan a su destino es mínima.

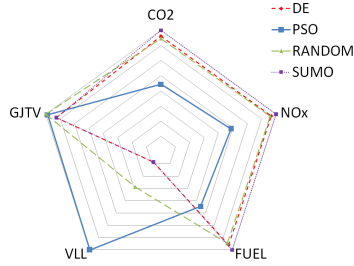


Figura 3: Diagrama normalizado de es-trella

	PSO	DE	RANDOM	SCPG
Escenario 1	455.00 (13)	269.64 (4)	333.16 (1)	248.64
Escenario 2	197.51 (3)	125.29 (26)	142.19 (4)	113.39
Escenario 3	172.45 (3)	111.00 (26)	125.00 (17)	98.67
Escenario 4	260.32 (20)	172.03 (26)	187.84 (24)	145.61
Escenario 5	133.39 (14)	96.80 (26)	100.00 (20)	78.90
Escenario 6	241.90 (2)	186.25 (22)	214.58 (0)	170.19
Escenario 7	368.90 (20)	205.90 (4)	246.39 (24)	194.13
Escenario 8	341.41 (2)	200.93 (4)	235.32 (24)	184.19
Escenario 9	147.54 (3)	96 (11)	107.87 (17)	91.26
Escenario 10	272.32 (20)	172.87 (26)	191.35 (20)	146.41

Tabla 3: Mejor media de VLL y configuración que genera este resultado.

En la Tabla 3 podemos observar la media de vehículos que llegan a su destino para cada escenario y algoritmo. Destacamos entre paréntesis el programa de semáforos de cada algoritmo que obtiene el mejor resultado promediando 30 ejecuciones independientes. Podemos observar que los programas generados por PSO siempre obtienen la mejor media de vehículos que llegan a su destino. Destacamos además que el programa generado que obtiene el mejor resultado no es siempre el mismo. Por ejemplo, para PSO, los programas 3 y 20 consiguen el mejor resultado en 3 de 10 escenarios. Esto quiere decir que no existe un programa de semáforos que sea el mejor para todos los posibles escenarios de tráfico. Incluso el programa 13, que consigue el mejor resultado para el escenario más probable (Escenario 1 con 63,59%), no vuelve a ser el mejor para ningún otro escenario. Esta misma interpretación se da para los demás algoritmos. Por ello, pensamos que es necesario el entrenamiento de estos algoritmos para considerar los aspectos variables del tráfico que hemos plasmado en 10 escenarios diferentes.

En la Figura 4 observamos los diagramas de caja para 4 métricas representativas: CO2 por vehículo, GJT por vehículo, VLL y VNLL. En estos diagramas no hemos tenido en cuenta los algoritmos, sino que hemos analizado por escenario para poner de manifiesto la variabilidad que podemos encontrar según las circunstancias del tráfico. Por un lado, analizamos el escenario más favorable atendiendo a los vehículos que llegan a destino, que es el Escenario 1. En este escenario los programas de semáforos analizados hacen que la mayoría de los vehículos lleguen al destino, arrojando una mínima cantidad de CO2 y con GJT por vehículo cercano a 500sg. Este escenario es tan favorable porque asume una las condiciones meteorológicas es soleada (S), con tiempo de simulación largo (SL), y con mayor cantidad de vehículos ligeros (VL). Por otro lado, el escenario más desfavorable es el Escenario 5, porque llegan muy pocos vehículos en comparación con los que no llegan a su destino, por lo que se arroja en proporción mucho CO2 a la atmósfera. Este escenario es tan desfavorable porque las condiciones meteorológicas son de lluvia (LL), tormenta (T) y niebla (NB), con simulación corta (SC), con 500 vehículos (CM), y un mayor porcentaje de vehículos pesados (P).

Este análisis nos da una clara idea de la necesidad de aplicación de las técnicas de priorización para tener en cuenta todos los posibles escenarios de tráfico.

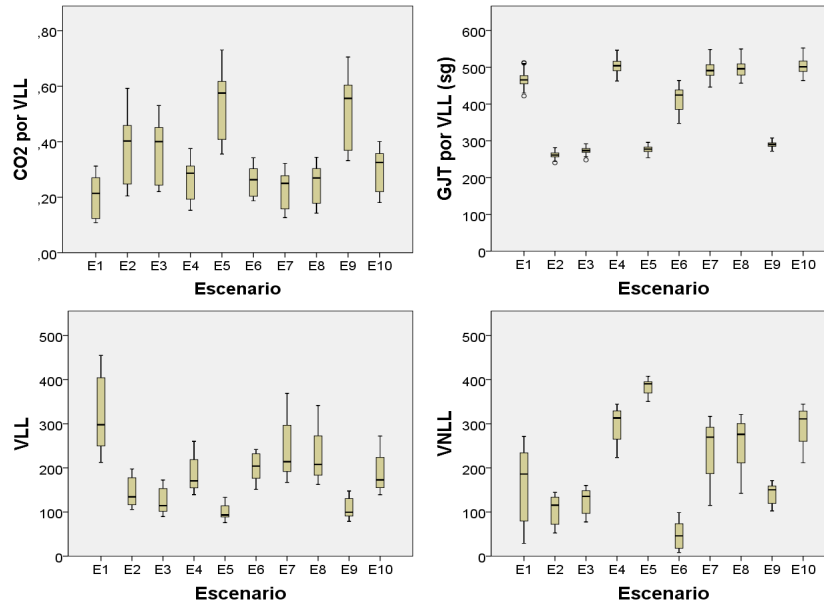


Figura 4: Diagramas de caja de CO2, GJT, VLL, y VNLL por escenario.

Anteriormente sólo se han considerado unas únicas situaciones de tráfico y eso provoca la generación de programas de semáforos limitados a condiciones ideales de tráfico. A partir de ahora podemos proponer algoritmos para generar programas de semáforos optimizados que se adapten a más situaciones diferentes de tráfico, ponderando aquellas que sean más probables en el día a día, pero sin olvidar las situaciones excepcionales.

5. Conclusiones

En este trabajo hemos propuesto una estrategia para la validación de programas de ciclo de semáforos. Hemos partido de un Modelo de Características que recoge la variabilidad que existe en un sistema complejo como es la gestión del tráfico, para, a continuación, mediante un algoritmo de optimización (PGS), generar un conjunto mínimo de escenarios que cumplen que todos los posibles pares de características se han validado. Además, hemos comprobado cómo se adaptan los programas de semáforos generados por los cuatro algoritmos propuestos PSO, DE, Random Search y SCPG. Este sistema de validación nos permite elegir un programa específico para situaciones de tráfico diferentes.

Analizando los resultados podemos concluir que PSO ha sido el mejor algoritmo generando programas de semáforos para todos los escenarios. No obstante, no existe un programa concreto, de los 30 generados por PSO, que haya sido el mejor para todos los escenarios. De hecho, para un programa generado por PSO, como máximo es el mejor en sólo 3 de 10 escenarios. Este hecho nos indica que la ponderación de cada escenario debe ser tomada en cuenta para la elección del mejor programa de semáforos generado por los algoritmos.

Una línea de trabajo futuro sería considerar varios escenarios diferentes en la evaluación del *fitness* de los algoritmos, para hacer que se generen programas de semáforos más robustos y adaptativos a las condiciones cambiantes de tráfico. Los escenarios no son igualmente probables, así que proponemos también considerar estos pesos de los escenarios para convertir el *fitness* en una función agregativa, ponderada según el resultado que obtengamos en la generación de los escenarios, a partir del Modelo de Características.

Acknowledgments Este trabajo está parcialmente financiado por el Ministerio de Economía y Competitividad (MINECO) del Gobierno de España y fondos FEDER con el proyecto roadME con TIN2011-28194 (<http://roadme.1cc.uma.es>). J. Ferrer y J. García-Nieto disfrutaron de sendas becas con códigos BES-2012-055967 y BES-2009-018767 del MINECO.

Referencias

1. Caragliu, A., Del Bo, C., Nijkamp, P.: Smart cities in europe. *Journal of Urban Technology* **18**(2) (2011) 65–82
2. García-Nieto, J., Olivera, A., Alba, E.: Optimal cycle program of traffic lights with particle swarm optimization. *IEEE Trans. on Evolutionary Computation* (2013)
3. Medina, J.S., Moreno, M.G., Royo, E.R.: Applying a Traffic Lights Evolutionary Optimization Technique to a Real Case: Las Ramblas Area in Santa Cruz de Tenerife. *IEEE Trans. on Evo. Comp.* **12**(1) (2008) 25–40
4. Nagatani, T.: Effect of speed fluctuations on a green-light path in a 2D traffic network controlled by signals. *Physica A: Statistical Mechanics and its Applications* **389**(19) (2010) 4105–4115
5. Kang, K., Cohen, S., Hess, J., Novak, W., Peterson, A.: Feature-Oriented Domain Analysis (FODA) Feasibility Study. Technical Report CMU/SEI-90-TR-21, Software Engineering Institute, Carnegie Mellon University (1990)
6. Ferrer, J., Kruse, P.M., Chicano, J.F., Alba, E.: Evolutionary algorithm for prioritized pairwise test data generation. In: GECCO, ACM (2012) 1213–1220
7. Krajzewicz, D., Bonert, M., Wagner, P.: The open source traffic simulation package SUMO. RoboCup 2006 Infrastructure Simulation Competition (2006)
8. Kennedy, J., Eberhart, R.C.: Swarm Intelligence. Morgan Kaufmann Publishers, San Francisco, California (2001)
9. Price, K., Storn, R., Lampinen, J.: Differential Evolution: A Practical Approach to Global Optimization. Natural Computing, Springer (2005)
10. Benavides, D., Segura, S., Cortés, A.R.: Automated analysis of feature models 20 years later: A literature review. *Inf. Syst.* **35**(6) (2010) 615–636
11. Cohen, M.B., Dwyer, M.B., Shi, J.: Constructing interaction test suites for highly-configurable systems in the presence of constraints: A greedy approach. *IEEE Trans. Software Eng.* **34**(5) (2008) 633–650
12. Stevens, B., Mendelsohn, E.: Efficient software testing protocols. In: Proceedings of the 1998 conference of the Centre for Advanced Studies on Collaborative research. CASCON '98, IBM Press (1998) 22–
13. Durillo, J.J., Nebro, A.J.: jMetal: A java framework for multi-objective optimization. *Advances in Engineering Software* **42**(10) (2011) 760–771
14. Trinidad, P., Benavides, D., Ruiz-Cortes, A., Segura, S., Jimenez, A.: Fama framework. In: SPLC '08. 359–359
15. Clerc, M.: Standard PSO 2011. Technical Report [online] <http://www.particleswarm.info/>, Particle Swarm Central (January 2011)