

Configuración Óptima del Protocolo de Encaminamiento OLSR para VANETs Mediante Evolución Diferencial

Jamal Toutouh José M. García-Nieto Enrique Alba

Dept. LCC, ETSI Informática

Bulevar Louis Pasteur 35

Univ. de Málaga, 29071

jamal,jnieto,eat@lcc.uma.es

Resumen

Uno de los elementos más influyentes en el funcionamiento de las redes MANETs/VANETs es el protocolo de encaminamiento. Sin embargo, en la literatura actual existen pocos trabajos que traten la configuración de estos protocolos de manera óptima y automática. En este trabajo se aborda la configuración óptima del protocolo de encaminamiento OLSR para redes VANET mediante un algoritmo de Evolución Diferencial. Las configuraciones obtenidas consiguen un alto grado de entrega de paquetes y reducen sustancialmente la carga de red y el retardo medio del protocolo OLSR respecto a su configuración oficial estándar (tomado de RFC 3626), así como respecto a otras configuraciones diseñadas por expertos en el estado del arte.

1. Introducción

Las redes vehiculares ad hoc o VANETs (Vehicular Ad-hoc Networks) son redes descentralizadas entre vehículos dotados de dispositivos de comunicación inalámbrica. Así, los vehículos que pertenecen a una VANET tienen la capacidad de comunicarse entre sí y con distintos elementos de la infraestructura vial sin la necesidad de emplear una infraestructura subyacente. Este intercambio de información en tiempo real, además de poder mejorar la seguridad en la conducción, ofrece la posibilidad de optimizar el tránsito vehicular reduciéndose el consecuente impacto ambiental. En la actuali-

dad, los dispositivos inalámbricos de acceso a red empleados para el desarrollo de VANETs utilizan tecnologías que cumplen con la familia de estándares IEEE 802.11 (WiFi), principalmente IEEE 802.11b o IEEE 802.11g.

El uso de tecnologías basadas en WiFi implica que los vehículos sólo se pueden comunicar dentro de un espacio y tiempo limitados. Esto, unido al alta movilidad de los nodos de una VANET, produce que la topología cambie continuamente. De ahí que el encaminamiento de los paquetes dentro de la red sea una tarea altamente compleja que se realiza de forma descentralizada. Por tanto, una estrategia de encaminamiento eficiente es crucial para el desarrollo de las VANETs, ya que se puede así mejorar decisivamente la calidad de servicio (QoS) de la red. Se podrían obtener protocolos de este tipo mediante el diseño de protocolos específicos o por modificación de la configuración de parámetros de los ya existentes de forma que se adapten mejor a los requisitos específicos de este tipo de redes. La enorme cantidad de posibles configuraciones que se pueden dar hace prácticamente imposible la obtención de configuraciones eficientes sin el empleo de herramientas inteligentes de diseño automático. Esto justifica el empleo de las técnicas metaheurísticas [6] como herramientas que se adaptan bien para resolver este tipo de problemas. Desafortunadamente, son escasos los trabajos que se pueden encontrar en el estado del arte actual de esta materia. Alba *et al.* empleó un Algoritmo Genético Celular Multi-objetivo para encontrar una estrategia

de *broadcasting* para redes móviles ad hoc, o MANETs (Mobile Ad hoc Networks) [3], urbanas. Hung *et al.* [10] propuso un nuevo protocolo de encaminamiento que reducía el ratio de paquetes perdidos basado en optimización por cúmulos de partículas o PSO (Particle Swarm Optimization). Recientemente, García-Nieto *et al.* [8] optimizó el protocolo VDTP de transferencia de ficheros en entornos vehiculares realistas empleando cinco técnicas metaheurísticas diferentes.

En este trabajo, proponemos el empleo de una técnica metaheurística, la evolución diferencial o DE (*Differential Evolution*), para la obtención de una configuración óptima del protocolo de encaminamiento OLSR (*Optimized Link State Routing Protocol*) [7] desarrollado especialmente para redes ad hoc de alta movilidad. El rendimiento de este protocolo está determinado por la elección de sus parámetros que inicialmente vinieron definidos en el RFC 3626. Hemos escogido DE porque es un algoritmo especialmente diseñado para la optimización de parámetros con valores reales (continuos) [12] y porque no ha sido empleado antes y tiene interés científico investigarlo, que es el caso de este trabajo.

El artículo se organiza como sigue. En la siguiente sección ofrecemos una visión global del protocolo OLSR. La Sección 3 describe el algoritmo DE. La Sección 4 formula el problema de optimización del protocolo OLSR. Las secciones 5 y 6 presentan la estrategia seguida y los experimentos realizados, respectivamente. Los resultados se analizan en la Sección 7. Finalmente, la Sección 8 expone las conclusiones más importantes y plantea trabajos futuros.

2. Protocolo OLSR

OLSR [7] es un protocolo de encaminamiento proactivo diseñado expresamente para redes móviles ad hoc altamente dinámicas y con un ancho de banda limitado. Este protocolo provee de un mecanismo eficiente de disseminación (*flooding*) de información parcial del estado de la red empleando un conjunto de nodos especiales, los *Multipoint Relays (MPR)*. Periódicamente, los MPR intercambian información

sobre los enlaces con sus nodos vecinos mediante el envío de mensajes de control de la topología (*TC, Topology Control*). Cada nodo envía, también de forma periódica, mensajes *Hello* y *MID* a sus nodos vecinos (1-hop) para comprobar la existencia de enlace entre ellos.

Los mecanismos del OLSR están regulados por la configuración de un conjunto de parámetros definidos en el OLSR RFC [7] (ver Tabla 1) enfocada principalmente para MANETs de propósito general aunque se usa en VANETs. Los parámetros que más influyen en el rendimiento son: los intervalos de tiempo de espera para el reenvío de los mensajes Hello, MID y TC (*HELLO_INTERVAL*, *REFRESH_INTERVAL* y *TC_INTERVAL*, respectivamente); los tiempos límite de validez de la información recibida mediante los tres tipos de mensaje que son *NEIGHB_HOLD_TIME* (Hello), *MID_HOLD_TIME* (MID) y *TOP_HOLD_TIME* (TC); el *WILLINGNESS* o predisposición de los nodos a actuar como MPR; y el tiempo máximo que los MPR almacenan información sobre los paquetes enviados para evitar el reenvío de duplicados (*DUP_HOLD_TIME*).

Tabla 1: Principales parámetros del OLSR. Valores oficiales según el estándar RFC 3626.

Parámetros	Valores	Rango
HELLO_INTERVAL	2.0 s	1 ... 30
REFRESH_INTERVAL	2.0 s	1 ... 30
TC_INTERVAL	5.0 s	1 ... 30
WILLINGNESS	3	{0,1,3,6,7}
NEIGHB_HOLD_TIME	$3 \times \text{HELLO_INT.}$	3 ... 100
TOP_HOLD_TIME	$3 \times \text{TC_INT.}$	3 ... 100
MID_HOLD_TIME	$3 \times \text{TC_INT.}$	3 ... 100
DUP_HOLD_TIME	30.0 s	3 ... 100

3. Evolución Diferencial

La Evolución Diferencial (DE - Differential Evolution [12]) es un algoritmo estocástico poblacional diseñado para la resolución de problemas de optimización continua. En DE, la generación de nuevos individuos se realiza mediante operadores como la mutación y el cruce diferencial. Un individuo mutado w_{g+1}^i se genera mediante la siguiente ecuación (1):

$$w_{g+1}^i \leftarrow v_g^{r1} + \mu \cdot (v_g^{r2} - v_g^{r3}) \quad (1)$$

en la cual $r1, r2, r3 \in \{1, 2, \dots, i-1, i+1, \dots, N\}$ son valores enteros obtenidos aleatoriamente y diferentes entre si, además de diferentes al índice i . La constante de mutación $\mu > 0$ introduce diversidad entre los individuos v_g^{r2} y v_g^{r3} evitando así el estancamiento del proceso de búsqueda.

Para incrementar aún más la diversidad entre los individuos de la población, cada individuo mutado realiza una operación de cruce con el individuo objetivo (*target*) v_g^i , mediante la cual se genera un individuo intermedio (*trial*) u_{g+1}^i . A continuación, se obtiene una posición aleatoria del individuo mutado para evitar que el individuo intermedio replique totalmente al individuo objetivo.

$$u_{g+1}^i(j) \leftarrow \begin{cases} w_{g+1}^i(j) & \text{if } r(j) \leq Pc \text{ or } j = j_r, \\ v_g^i(j) & \text{en otro caso} \end{cases} \quad (2)$$

Tal y como se muestra en la Ecuación 2, dependiendo de la probabilidad de cruce Pc se seleccionará un elemento del individuo objetivo o del individuo intermedio. Finalmente, el individuo intermedio será aceptado para la siguiente generación si y sólo si es mejor que el individuo actual (minimiza, Ecuación 3).

$$v_{g+1}^i \leftarrow \begin{cases} u_{g+1}^i & \text{if } f(u_{g+1}^i) \leq f(v_g^i), \\ v_g^i(j) & \text{en otro caso} \end{cases} \quad (3)$$

En el Algoritmo 1 se muestra el pseudocódigo del algoritmo DE. Tras inicializar la población (Línea 1) los individuos evolucionan durante un número máximo de evaluaciones. En cada generación se realiza una operación de mutación (Línea 5) y cruce (Línea 6). Se selecciona un nuevo individuo según la Ecuación 3 (Líneas 7 y 8).

4. Configuración Óptima de OLSR

Al ser OLSR un protocolo expresamente diseñado para MANETs de alta movilidad también se está empleando en VANETs. Sin embargo, el uso de la configuración oficial están-

Algoritmo 1 Pseudocódigo de DE

```

1: inicializaPoblación()
2: mientras  $g < \text{maxGeneraciones}$  hacer
3:   para cada individuo  $v_g^i$  hacer
4:     elegir mutuamente diferentes( $r_1, r_2, r_3$ )
5:      $w_{g+1}^i \leftarrow \text{muta}(v_g^{r1}, v_g^{r2}, v_g^{r3}, \mu)$ 
6:      $u_{g+1}^i \leftarrow \text{cruza}(v_g^i, w_{g+1}^i, cp)$ 
7:     evalúa( $u_{g+1}^i$ )
8:      $v_{g+1}^i \leftarrow \text{selecciona}(v_g^i, u_{g+1}^i)$ 
9:   fin para
10: fin mientras

```

dar (ver Tabla 1) ofrece una pobre calidad de servicio en redes vehiculares [13]. Por ello, en este trabajo pretendemos obtener la configuración de los parámetros de OLSR que se adapte mejor a los requisitos específicos de las redes VANETs.

A partir de los parámetros presentados en la Tabla 1 hemos definido un *vector solución* cuyos valores son continuos (reales) dentro del rango definido. Así, podemos emplear una técnica metaheurística para obtener automáticamente los valores del vector solución que represente una configuración de OLSR cuyo rendimiento en VANETs mejore al de la parametrización que se define en su RFC 3626 original. A su vez, en este trabajo compararemos de forma analítica distintas configuraciones y su rendimiento, para ofrecerle a los expertos una ayuda que puedan emplear en el diseño de protocolos de encaminamiento específicos para VANETs.

Para evaluar el rendimiento de las distintas configuraciones de OLSR (soluciones), y así guiar al algoritmo en la búsqueda, hemos empleado tres métricas de calidad de servicio ampliamente empleadas en este área:

- *Ratio de envío de paquetes (PDR-Packet delivery ratio)*: Fracción de los paquetes enviados, que son recibidos de forma completa y correcta por el destinatario.
- *Carga de gestión de routing (NRL-Normalized routing load)*: Relación entre el número de paquetes empleados para la gestión del protocolo de encaminamiento y los paquetes de datos enviados.

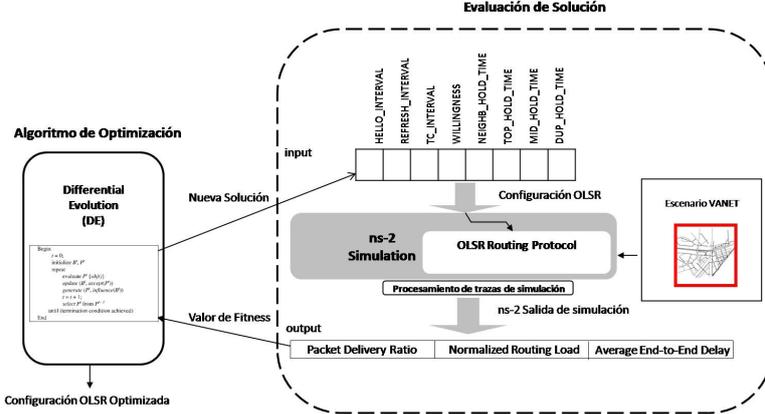


Figura 1: Modelo de optimización para la configuración de OLSR. El algoritmo DE realiza una simulación llamando a *ns-2* en cada evaluación de solución.

- *Retardo medio punto a punto (E2ED-Average End-to-End delay of a data packet):* Tiempo medio que tarda un paquete en llegar a su destinatario tras ser creado por una aplicación para su envío.

5. Modelo de Optimización

Para la resolución del problema de optimización del protocolo OLSR hemos empleado de forma conjunta un algoritmo de optimización y un proceso de simulación. El algoritmo de optimización utilizado en este trabajo ha sido la metaheurística Evolución Diferencial (DE). El proceso de simulación se ha utilizado para asignarle un valor de calidad o *fitness* a las configuraciones calculadas por el algoritmo de optimización. Para ello se ha utilizado *ns-2* [1], simulador empleado previamente para simular redes VANET que ha demostrado proveer resultados cercanos a la realidad [4]. El simulador ha sido modificado para poder interactuar de forma automática con algoritmos metaheurísticos para así poder emplearlo en este y futuros trabajos.

El método completo de optimización es un proceso evolutivo dirigido por el algoritmo DE. Como se muestra en la Figura 1, cuando DE quiere evaluar una solución invoca el proceso de simulación de dicha configuración del OLSR sobre un escenario definido para VANETs, en

el cual se tiene en cuenta las carreteras, los edificios, las señales de tráfico, la interacción entre vehículos, etc. Después de la simulación donde se analizan desde varios puntos de vista el mismo escenario VANET, *ns-2* devuelve una traza a partir de la cual se calculan las métricas (PDR, NRL, E2ED) para las comunicaciones de toda la red. Una vez obtenidos los valores de las métricas, el *fitness* de una solución se calcula como sigue:

$$fit = w_1 \cdot (-PDR) + w_2 \cdot NRL + w_3 \cdot E2ED \cdot C \quad (4)$$

Para mejorar la calidad de servicio del protocolo OLSR, el objetivo consiste en maximizar el PDR, minimizando el NRL y el E2ED. Como se puede observar en la Ecuación 4, se ha empleado una *función de minimización agregativa*, de ahí que el PDR se haya formulado con un signo negativo. En esta ecuación, los factores w_1 , w_2 y w_3 (0,8, 0,1 y 0,1, respectivamente) se emplean para introducir un peso a la influencia de cada métrica en la función de *fitness*. Así, el PDR es claramente prioritario frente a las otras dos, ya que hemos priorizado la efectividad del protocolo de encaminamiento. La constante $C = 0,01$ normaliza el valor de E2ED, para que su rango sea del mismo orden de magnitud que el de las otras dos métricas.

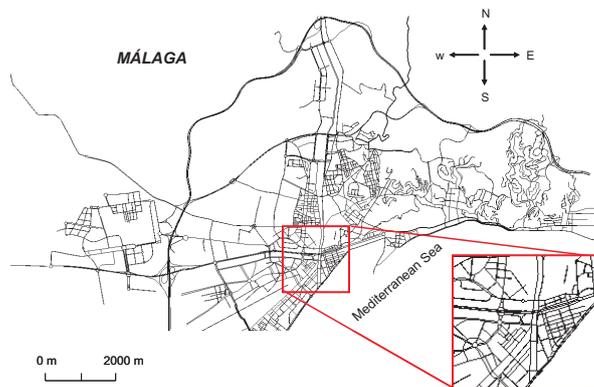


Figura 2: Área seleccionada ($2000 \times 2000 \text{ m}^2$) para el escenario VANET en el centro urbano de Málaga.

6. Marco Experimental

En esta sección se describe la instancia de red VANET generada para la evaluación de configuraciones OLSR, así como los parámetros y especificaciones que componen el procedimiento experimental.

6.1. Instancia: Escenario VANET

A la hora de evaluar el rendimiento de una red VANET, la práctica más extendida en la comunidad científica consiste en el uso de simuladores tanto de tráfico vehicular como de redes de comunicaciones. Podemos generar entornos VANETs siguiendo patrones realistas (teniendo en cuenta direcciones, sentidos, normas de circulación, semáforos, etc.) mediante la selección automática de áreas urbanas a partir de mapas digitales. Sobre dichos entornos se pueden definir modelos de movilidad y conectividad de los vehículos, y así, aproximar el funcionamiento real de la red.

En este trabajo hemos definido un escenario VANET a partir de la selección de un área metropolitana localizada en el centro de la ciudad de Málaga (Figura 2). Dicho área, de $2000 \times 2000 \text{ m}^2$ de extensión, contiene el entramado de avenidas y calles siguiendo el modelo real de direcciones y sentidos de tráfico.

Para ésta labor hemos utilizado el simulador de tráfico SUMO [11], mediante el cual se ha implementado el movimiento de 50 vehículos (nodos de red) durante un tiempo de si-

mulación de 5 minutos (300 segundos) dentro de la zona metropolitana seleccionada. De este modo, el modelo de tráfico (movimiento de vehículos) generado con SUMO mediante un fichero XML puede ser importado en el simulador de redes *ns-2* [1], convirtiendo el fichero de patrones de movimiento en formato Tcl (adecuado para éste último simulador de red). Para nuestra instancia VANET, el patrón de tráfico comprende 4050 movimientos, entre cambios de sentidos y direcciones, registrados por los vehículos. Durante el tiempo de simulación, los coches (nodos) intercambian información mediante entre 2 y 50 conexiones diferentes y la velocidad de los coches puede fluctuar entre 10 km/h y 50 km/h , teniendo en cuenta que se desarrolla en un entorno urbano.

Tabla 2: Parámetros de simulación *ns-2*

Parámetro	Valor
Tiempo de simulación	300 s
Numero de vehículos	50
Velocidad de circulación	10-50 km/h
Modelo de propagación	Two Ray Ground
Frecuencia de radio	2,47 GHz
Ancho de banda	5 Mbps
Rango de transmisión	250 m
Capa de acceso (MAC)	IEEE 802.11b
Protocolo de transporte	UDP
Flujo de datos CBR	50 sesiones

Respecto al modelo de flujo de datos de la red, se ha definido mediante Tcl (para *ns-2*) una secuencia de 50 sesiones de la aplicación de red CBR (Constant Bit Rate), las cuales operan sobre agentes de protocolo UDP (User Datagram Protocol) especificados en cada no-

Tabla 3: Resultados obtenidos por DE en la configuración óptima de OLSR. Se presentan las configuraciones: Mejor, Mediana, Peor y el cálculo de la Media en cuanto al fitness y a los indicadores de QoS. En la última fila se disponen los resultados de configuración oficial estándar (RFC 3626)

Solución	Configuración	Fitness	PDR	NRL	E2ED
Mejor		-68,35	94,00 %	68,34 kbps	8,36 ms
Mediana	Optimizada (DE)	-53,14	72,00 %	44,16 kbps	8,03 ms
Peor		-21,53	32,00 %	40,62 kbps	4,18 ms
Media		-49,88	66,00 %	49,37 kbps	6,85 ms
OLSR	FC 3626	61,22	80,00 %	328,42 kbps	1347,22 ms

do (vehículo). De esta forma, los vehículos interconectados intercambian datos mediante su respectivo agente CBR. El tamaño de paquete de datos CBR es de 512 bytes, con un ratio de envío de 4 paquetes por segundo. La Tabla 2 contiene un resumen de los parámetros de simulación utilizados.

6.2. Procedimiento Experimental

Para este estudio hemos empleado la implementación del algoritmo de Evolución Diferencial (DE) disponible en la biblioteca de metaheurísticas MALLBA [5], un framework en C++ para la resolución de problemas de optimización. La Tabla 4 contiene los valores de los principales parámetros de DE.

Tabla 4: Parámetros de DE

Parámetro	Valor
Número de individuos	10
Número de generaciones	100
Probabilidad de cruce (P_c)	0,9
Factor de mutación (μ)	0,1

En cuanto a la fase de simulación, para las evaluaciones de cada solución (configuración OLSR) se ha utilizado el simulador de redes *ns-2* en su versión *v-2.34*. De manera adicional, debido a que las versiones actuales de *ns-2* no incluyen aún ninguna implementación del protocolo OLSR, hemos incorporado para nuestros experimentos la implementación *UM-OLSR (version 0.8.8)* [2].

Para los experimentos se han llevado a cabo 30 ejecuciones independientes del algoritmo DE, siguiendo el modelo de optimización propuesto sobre el escenario VANET definido

(Málaga). Cada una de las ejecuciones independientes realiza un número de 1000 simulaciones con diferentes configuraciones de OLSR. Las máquinas utilizadas consisten en PCs Pentium IV 2.4 GHz core, 1 GB de RAM, y S.O. Linux Fedora core 6.

7. Resultados

A continuación se presentan los resultados obtenidos tras la experimentación con nuestra propuesta de optimización de OLSR.

7.1. Funcionamiento de DE

En la Tabla 3 se pueden observar los resultados obtenidos por el algoritmo DE en cuanto a los valores de fitness calculados por las configuraciones Mejor, Mediana y Peor de OLSR, en las 30 ejecuciones independientes. Además, se disponen los valores de PDR, NRL y E2ED de cada solución y se calculan las medias de todos éstos. La última fila de esta tabla contiene los resultados obtenidos tras la simulación de los parámetros de la configuración estándar del protocolo OLSR (tomados de RFC 3626).

La primera observación que podemos resaltar consiste en la mejora de los valores de fitness obtenidos por DE, los cuales son sustancialmente menores que el fitness de la configuración oficial (-49,88 de media sobre 61,22, ya que estamos minimizando), incluyendo la peor de las soluciones obtenidas por DE (-21,53 de media sobre 61,22). Si analizamos más en detalle los valores de QoS podemos ver que esta mejora se debe principalmente a la gran reducción de la carga de red (NRL) y del retardo medio (E2ED) que las configuraciones optimizadas (DE) experimentan sobre la configura-

Tabla 5: Configuraciones OLSR encontradas en el estado del arte por parte de expertos (Gómez et al. [9]), configuración estándar de RFC 3626, y mejor configuración obtenida por nuestro DE

Parámetros	Gómez <i>et al.</i> [9]			OLSR RFC	DE Mejor
	#1	#2	#3		
HELLO_INTERVAL	0,5 s	1,0 s	4,0 s	2,0 s	3,13826 s
REFRESH_INTERVAL	0,5 s	1,0 s	4,0 s	2,0 s	3,15026 s
TC_INTERVAL	1,25 s	2,5 s	10,0 s	5,0 s	45,245 s
WILLINGNESS	3	3	3	3	1
NEIGHB_HOLD_TIME	1,5 s	3,0 s	12 s	6,0 s	3,56191 s
TOP_HOLD_TIME	3,75 s	7,5 s	20,0 s	15,0 s	103,139 s
MID_HOLD_TIME	3,75 s	7,5 s	20 s	15,0 s	141,053 s
DUP_HOLD_TIME	30,0 s	30,0 s	30,0 s	30,0 s	67,7911 s

ción oficial estándar (49,37 kbps de media sobre 328,42 kbps en NRL y 6,85 ms de media sobre 1347,22 ms en E2ED). Además, la tasa de paquetes recibidos (PDR) muestra un porcentaje medio adecuado para redes VANETs (66,00 %), obteniendo incluso un porcentaje del 94 % en la mejor configuración encontrada. Curiosamente, en la peor configuración encontrada por DE se obtiene el mejor valor de NRL (40,62 kbps) y de E2ED (4,18 ms), aunque en este caso, a costa de un bajo rendimiento en el envío efectivo de paquetes (PDR de 32 %). Por tanto, podemos resaltar ya la principal contribución de este trabajo: utilizando nuestro modelo de optimización con DE, conseguimos aumentar la tasa de envío (de 80 % a 94 %) y reducir tanto la carga de red (en un 85 %) como el retardo medio (en un 99,5 %) respecto a la configuración oficial de OLSR.

Desde el punto de vista del funcionamiento interno de DE, en la Figura 3 podemos ver las trazas de evolución del mejor fitness encontrado durante las ejecuciones Mejor, Mediana y Peor. En este sentido, la principal característica consiste en la rápida convergencia que experimenta el algoritmo ya que consigue una amplia reducción de fitness durante las 60 primeras generaciones. Este es un comportamiento bastante deseable en este tipo de problemas ya que debido al alto tiempo requerido en cada evaluación de solución (37,6 segundos de media por simulación de *ns-2* con nuestro escenario VANET), el número de evaluaciones debe ser lo suficientemente reducido para abordar el problema de manera práctica y eficiente. El tiempo medio de cómputo es de 10,44 horas por ejecución de DE.

7.2. Comparación con Otras Configuraciones del Estado del Arte

En esta sección comparamos los resultados obtenidos por nuestras configuraciones OLSR optimizadas por DE con una serie de configuraciones propuestas en el estado del arte por expertos humanos en la materia.

En concreto, nos centramos en el trabajo presentado por Gómez *et al.* [9] en el que se analizan tres configuraciones de OLSR y se comparan con la configuración estándar. Para ello se centran principalmente en la latencia de cambio de ruta (RCL) y evalúan la conectividad punto a punto (End-to-End) mediante una instancia de MANET básica.

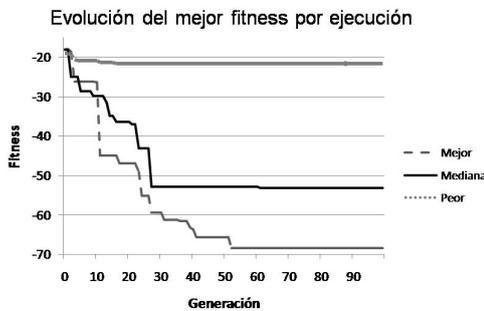


Figura 3: Evolución del mejor fitness obtenido por DE en las ejecuciones Mejor, Mediana y Peor.

Tabla 6: Comparación de los resultados obtenidos por la mejor configuración OLSR de nuestro DE con otras configuraciones analizadas por expertos (Gómez *et al.* [9]) del estado del arte

Configuración		Fitness	PDR	NRL	E2ED
Gómez <i>et al.</i> , [9]	#1	46,19	90,00 %	1170,02 kbps	1197,25 ms
	#2	-15,31	90,00 %	554,75 kbps	1208,91 ms
	#3	-29,47	66,00 %	208,84 kbps	2435,22 ms
RFC 3626		61,22	80,00 %	328,42 kbps	1347,22 ms
DE	Mejor	-68,35	94,00 %	68,34 kbps	8,36 ms

Los valores de los parámetros de dichas configuraciones (#1, #2 y #3) están recogidos en la Tabla 5 junto con los parámetros de RFC 3626 y aquellos de la mejor configuración obtenida por nuestra metaheurística DE. A partir de estas configuraciones, se disponen en la Tabla 6 los resultados obtenidos por sendas simulaciones con nuestra instancia VANET.

De nuevo podemos observar (Tabla 6) la mejora que experimenta la configuración optimizada por DE respecto a las tres configuraciones propuestas por Gómez *et al.* [9]. Si bien se obtiene mejora en cuanto al ratio de entrega de paquetes (PDR de 90 % en #1 y #2 a 94 % en DE), es en carga de red (NRL) y sobre todo en retardo medio punto a punto (E2ED) donde DE consigue el mayor beneficio, llegando a ser del 67,28 % (de 328,42 kbps a 68,34 kbps) en el caso de NRL y del 99,66 % (de 1347,22 ms a 8,36 ms) en E2ED sobre #3, la mejor configuración en términos de valor de fitness del trabajo citado.

Si analizamos los conjuntos de parámetros que conforman las configuraciones (Tabla 5) podemos destacar dos diferencias básicas. En primer lugar, el valor de WILLINGNESS es 1 en las configuraciones optimizadas y 3 en las demás. Esto produce que el proceso de selección de MPRs, definido en el RFC 3626 [7], requiera de menos tiempo de cómputo y de carga en la red. En segundo lugar, los valores temporales en los parámetros de registro de información intermedia (TOP_HOLD_TIME, MID_HOLD_TIME y DUP_HOLD_TIME) son mucho mayores en las configuraciones optimizadas que en las configuraciones de expertos y estándar. Por lo tanto, el valor de la información de gestión es tenido en cuenta por OLSR durante mucho más tiempo. Posiblemente, esta información se desaprovecha

constantemente en la configuración estándar (RFC) ya que se obliga al protocolo a efectuar actualizaciones innecesarias de registros intermedios con demasiada frecuencia. Esto último concuerda además con las principales conclusiones tomadas en Gómez *et al.* [9].

8. Conclusiones

En este trabajo se ha abordado la optimización de los parámetros del protocolo de routing OLSR para VANETs mediante el algoritmo DE. Para ello se ha generado una instancia de VANET con especificaciones de escenario realistas (área urbana de Málaga) con el que se ha evaluado el algoritmo propuesto. Las configuraciones obtenidas mejoran sustancialmente tanto la configuración oficial estándar de OLSR (RFC 3626) como otras configuraciones propuestas por expertos. Esta mejora supone aproximadamente el 67,28 % en carga de red y el 99,66 % en retardo medio.

Como trabajo futuro se pretende realizar un estudio más detallado sobre los parámetros que pueden afectar de manera significativa al funcionamiento de la VANET estudiada así como nuevos escenarios basados en entornos de autovía, áreas comerciales, etc. Además, se abordará el problema mediante formulación multiobjetivo.

Agradecimientos

Los autores están parcialmente financiados por el CICE Junta Andalucía P07-TIC-03044 (DIRICOM, <http://diricom.lcc.uma.es>) y Ministerio MICINN de España y FEDER con referencia TIN2008-06491-C04-01 (M* <http://mstar.lcc.uma.es>). José M. García-Nieto disfruta la beca BES-2009-018767 del MICINN.

Referencias

- [1] The Network Simulator Project - Ns-2. [online] <http://www.isi.edu/nsnam/ns/>.
- [2] UM-OLSR Implementation of University of Murcia. [online] Available in URL <http://masimum.dif.um.es/?Software:UM-OLSR>.
- [3] E. Alba, B. Dorronsoro, F. Luna, A. J. Nebro, and P. Bouvry. A Cellular MOGA for Optimal Broadcasting Strategy in Metropolitan MANETs. *Computer Communications*, 30(4):685 – 697, 2007.
- [4] E. Alba, S. Luna, and J. Toutouh. Accuracy and efficiency in simulating vanets. In L. Hoai et al., editor, *MCO*, volume 14, pages 568–578. Springer, 2008.
- [5] E. Alba, G. Luque, J. García-Nieto, G. Ordoñez, and G. Leguizamón. MALLBA: A software library to design efficient optimisation algorithms. *IJICA*, 1(1):74–85, 2007.
- [6] C. Blum and A. Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys*, 35(3):268–308, 2003.
- [7] T. Clausen and P. Jacquet. Optimized Link State Routing Protocol (OLSR). IETF RFC 3626, 2003 [online] in URL <http://www.ietf.org/rfc/rfc3626.txt>.
- [8] J. García-Nieto, J. Toutouh, and E. Alba. Automatic tuning of communication protocols for vehicular ad hoc networks using metaheuristics. *Eng. App. of Art. Intel.*, In Press:-, 2010.
- [9] C. Gomez, D. García, and J. Paradells. Improving performance of a real ad hoc network by tuning olsr parameters. In *ISCC '05*, pages 16–21, Washington, DC, USA, 2005. IEEE Computer Society.
- [10] C. Huang, Y. Chuang, and K. Hu. Using particle swarm optimization for QoS in ad-hoc multicast. *Eng. Appl. of Art. Intelligence*, In Press, 2009.
- [11] D. Krajzewicz, M. Bonert, and P. Wagner. The open source traffic simulation package SUMO. In *RoboCup '06*, pages 1–10, Bremen, Germany, 2006.
- [12] K. V. Price, R. Storn, and J. Lampinen. *Differential Evolution: A practical Approach to Global Optimization*. Springer-Verlag, London, UK, 2005.
- [13] F. Rango, J. Cano, M. Fotino, C. Calafate, P. Manzoni, and S. Marano. Olsr vs dsr: A comparative analysis of proactive and reactive mechanisms from an energetic point of view in wireless ad hoc networks. *Computer Communications*, 31(16):3843–3854, 2008.