

Comunicación eficiente entre vehículos aplicando un algoritmo multi-objetivo paralelo

Jamal Toutouh y Enrique Alba

Resumen— El encaminamiento de datos en redes vehiculares presenta un desafío debido a los frecuentes e inesperados cambios de topología. En este estudio presentamos un algoritmo de optimización multi-objetivo paralelo para tratar este tipo de problemas con el fin de maximizar la cantidad de datos que se intercambian y minimizar los tiempos de transmisión. Esta forma de abordarlo es novedosa porque en este dominio la mayoría de trabajos de optimización emplean técnicas secuenciales y/o mono-objetivas, hecho que limita su eficacia. Los resultados experimentales muestran que el algoritmo optimizado mejora de forma significativa a las otras propuestas en el estado del arte. Además, la aplicación de nuestro modelo paralelo obtiene una eficiencia computacional mayor de un 86%.

Palabras clave— optimización multi-objetivo, paralelismo, redes vehiculares, encaminamiento

I. INTRODUCCIÓN

Las redes vehiculares ad-hoc, más conocidas por su nombre en inglés *VANETs* (*Vehicular ad hoc networks*), son redes de comunicación inalámbrica sin infraestructura formadas por los dispositivos de los usuarios de las carreteras (unidades de abordo de los vehículos, teléfonos inteligentes, etc.) y distintos elementos de la infraestructura vial como semáforos y paneles informativos (ver Fig. 1). Estas redes ofrecen la oportunidad para desarrollar aplicaciones revolucionarias en el ámbito de la seguridad y eficiencia vial mediante el intercambio continuo de información del tráfico y de las maniobras de los usuarios.

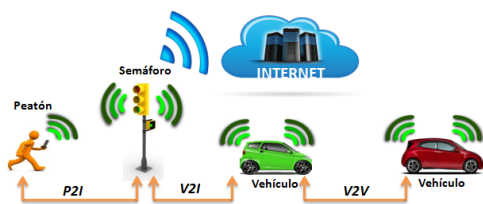


Fig. 1. Esquema básico de comunicación en VANETs.

El encaminamiento de datos (o *routing*) en VANETs presenta un desafío mayor que en redes móviles ad-hoc o MANETs (*mobile ad-hoc networks*) debido a la alta volatilidad de su topología provocada por el empleo de una tecnología de comunicación de cobertura limitada (IEEE 802.11p) y las altas velocidades de los nodos. Además, las aplicaciones suelen tener requisitos críticos de tiempo real. Así, desde entornos académicos e industriales se está invirtiendo mucho esfuerzo en tratar este problema [1].

Dept. de Lenguajes y Ciencias de la Computación. Universidad de Málaga, Málaga, España.
E-mail: {jamal,eat}@lcc.uma.es.

El protocolo de encaminamiento *Ad hoc On Demand Vector* (AODV) [2] ha sido estudiado por diversos autores para optimizar su rendimiento en VANETs [3], [4]. Una de las formas de mejorarlo es encontrando la configuración *óptima* de sus parámetros. Sin embargo, el número de parámetros como el rango de los mismos impide la aplicación de métodos de optimización exactos o enumerativos. Por ello se han analizado diferentes metaheurísticas mono-objetivo en este propósito [5], [6].

En este trabajo se aplica una versión paralela del algoritmo de cúmulo de partículas de velocidad limitada multi-objetivo o SMPSTO (*Speed-constrained Multi-objective Particle Swarm Optimization*) [7] al problema de optimización del protocolo AODV en VANETs. La metodología consiste en explorar el espacio de búsqueda formado por todas las combinaciones de los 11 parámetros (continuos y discretos) que forman configuraciones factibles de AODV, para así encontrar aquellas que optimicen su rendimiento.

Ésta es una propuesta novedosa en este dominio, puesto que la mayoría de trabajos previos en optimización de comunicaciones en VANETs emplean metaheurísticas mono-objetivo [5], [8], [9], es decir, evalúan las soluciones empleando una función agregativa en la que cada uno de los objetivos es ponderado de acuerdo a un peso dado. Este tipo de métodos tienen el problema de que los pesos elegidos influyen críticamente en la solución. Posteriormente, se ha utilizado el algoritmo multi-objetivo (MO) NSGA-II (*Non-dominated Sorting Genetic Algorithm, II*) para optimizar el protocolo de encaminamiento OLSR (*Optimized Link State Routing*) en VANETs [10].

Emplear un modelo distribuido del algoritmo permite mejorar el tiempo de búsqueda puesto que éste ejecuta en paralelo la evaluación de todas las soluciones (simulaciones de VANETs computacionalmente costosas). Así, el número de evaluaciones de la solución no se ve tan limitado como en los algoritmos secuenciales y se puede incrementar tanto el número de soluciones que emplea el algoritmo como el de generaciones que se llevan a cabo [6], [9].

Globalmente, las principales contribuciones de este trabajo son: i) introducir el uso de algoritmos paralelos multi-objetivos para tratar el problema de optimización en VANETs y ii) optimizar el rendimiento de AODV cuando se emplea en entornos vehiculares.

El texto se organiza como sigue. A continuación se introduce el *routing* en VANETs, el protocolo AODV y su problema de optimización MO. La Sección III resume la bibliografía relacionada con nuestro traba-

jo. La Sección IV describe el SMPSO y la variante utilizada en este estudio. La Sección V presenta la metodología y los detalles de implementación. Los experimentos y el análisis de resultados se muestran en la Sección VI. Finalmente, la Sección VII contiene las conclusiones y el trabajo futuro.

II. ENCAMINAMIENTO DE DATOS EFICIENTE EN VANETS

Esta sección introduce el problema del *routing* en VANETS, presenta el protocolo AODV y define el problema multi-objetivo que se trata en este trabajo.

A. Encaminamiento de datos en redes vehiculares

Las comunicaciones VANET sufren de continuas pérdidas de enlace, fragmentación de la red y continuos cambios de topología, provocado principalmente por la velocidad a la que se mueven los nodos (vehículos) y al uso de tecnologías de acceso inalámbrico de cobertura limitada (IEEE 802.11p). Por esto, cuando se define una arquitectura VANET es de vital importancia aplicar una estrategia de encaminamiento de datos eficiente que maximice la cantidad de datos que se intercambia y que minimice los tiempos de transmisión de los datos.

Existen diversas líneas de investigación en la definición de protocolos para VANETS [1], [11]. Dentro del variado conjunto de soluciones propuestas, nosotros nos centramos en un protocolo de la tipología *unicast*. La función principal de este tipo de protocolos es la de encaminar paquetes de datos a un único destino vía comunicaciones multi-salto. Los vehículos que se encuentran en la ruta de los datos deben retransmitir los paquetes hacia el nodo destino. Además, estos protocolos no deben generar una carga de control excesiva, pero tiene que ser suficiente para tener la información de las rutas actualizadas.

B. El protocolo de encaminamiento AODV

AODV [2] es un protocolo unicast para redes inalámbricas ad-hoc que pertenece a la familia de protocolos reactivos, es decir, calcula la ruta cuando se va a empezar a transmitir y se mantiene solo durante la transmisión. Diferentes variantes de AODV se han propuesto para VANETS porque al ser reactivo ofrece un rendimiento aceptable y no genera una carga de control excesiva [3], [12], [13].

El proceso de búsqueda de la ruta empieza cuando el nodo origen difunde un paquete *Routing Request* (*RREQ*) en el que se indica el destino. Entonces, los demás nodos de la red retransmiten el paquete *RREQ* a sus vecinos hasta que llega al nodo destino o se alcanza un número de saltos determinado. Cuando el nodo destino recibe el *RREQ* responde al nodo origen enviándole la información sobre la ruta en un paquete *Route Reply* (*RREP*) de forma unicast. El nodo origen espera el *RREP* hasta que lo recibe o hasta que pasa un tiempo máximo de espera.

El funcionamiento de AODV viene determinado por una serie de parámetros cuya configuración in-

fluye drásticamente en su rendimiento: cinco temporizadores y seis contadores. La Tabla I presenta estos parámetros, el valor de referencia que sugiere el AODV RFC 3561 para MANETS, los rangos de los parámetros y su tipo (continuo \mathbb{R} o discreto \mathbb{Z}).

TABLA I

PARÁMETROS DE CONFIGURACIÓN DE AODV.

Parámetro	Valor RFC	Rango	Tipo
HELLO_INTERVAL	1.0 s	[1.0, 20.0]	\mathbb{R}
ACTIVE_ROUTE_TIMEOUT	3.0 s	[1.0, 20.0]	\mathbb{R}
MY_ROUTE_TIMEOUT	6.0 s	[1.0, 40.0]	\mathbb{R}
NODE_TRAVERSAL_TIME	0.040 s	[0.01, 15.0]	\mathbb{R}
MAX_RREQ_TIMEOUT	10.0 s	[1.0, 100.0]	\mathbb{R}
NET_DIAMETER	35	[3, 100]	\mathbb{Z}
ALLOWED_HELLO_LOSS	2	[0, 20]	\mathbb{Z}
REQ_RETRIES	2	[0, 20]	\mathbb{Z}
TTL_START	1	[1, 40]	\mathbb{Z}
TTL_INCREMENT	2	[1, 20]	\mathbb{Z}
TTL_THRESHOLD	7	[1, 60]	\mathbb{Z}

C. Optimización multi-objetivo del protocolo AODV

El problema de la búsqueda de configuraciones factibles de parámetros eficientes para AODV no es un problema fácil [5]. El número y el rango de los parámetros que gobiernan el protocolo definen un espacio de búsqueda lo suficientemente grande para hacer inútil el uso de métodos exactos y enumerativos para resolverlo. Por contra, el uso de algoritmos metaheurísticos de optimización es viable puesto que calculan configuraciones optimizadas en tiempos de ejecución razonables [5], [8], [14], [15].

La evaluación de las configuraciones se realiza mediante el análisis de diferentes métricas de calidad de servicio del protocolo tras simularlo. Las métricas más utilizadas son: el porcentaje de paquetes recibidos respecto de los enviados o PDR (*packet delivery ratio*), el tiempo que emplea un paquete en llegar a su destino o E2ED (*end-to-end delay*) y la carga generada por un protocolo en relación con la cantidad de datos enviados o NRL (*normalized routing load*).

La mayoría de trabajos previos emplea metaheurísticas mono-objetivo y utiliza las tres métricas para evaluar las soluciones (configuraciones). Este tipo de métodos tienen el problema de que los pesos elegidos influyen críticamente en la búsqueda y la solución. Además, devuelven una única solución y lo deseable es tener un conjunto de soluciones óptimas que ponderen los objetivos de forma distinta.

En este trabajo hemos definido el problema de optimización multi-objetivo del protocolo AODV en VANETS. Buscamos configuraciones factibles de AODV que optimicen dos objetivos enfocados a los dos requisitos más importantes en el *routing* en VANETS: maximizar la cantidad de datos que se intercambian (PDR) y minimizar los tiempos de transmisión (E2ED). Estos dos objetivos son cotrapuestos, porque cuanto mayor es la cantidad de paquetes que viajan a través de la red mayor es la probabilidad de que aparezcan problemas de acceso al medio, por lo que los nodos tardan más en retransmitir los paquetes. Y viceversa, si la cantidad de paquetes que se transmiten es menor (bajo PDR), el tiempo que tardan los paquetes en llegar al destino es menor.

III. TRABAJO RELACIONADO

Esta sección resume la bibliografía en el diseño de protocolos de encaminamiento a partir de modificaciones sobre AODV y en la aplicación de metaheurísticas a la optimización en VANETs.

Para mejorar el rendimiento de AODV en redes vehiculares han aparecido diferentes variantes. Se han propuesto el AODV_BD [3], que reduce los tiempos de transferencia encapsulando datos en los paquetes RREQ, y AODV_OBD [16], que selecciona un subconjunto de los nodos que retransmiten los paquetes RREQ y RREP para reducir la carga de red. En el IADOV los nodos almacenan información de enrutamiento a nodos de hasta dos saltos de distancia mejorando el proceso de establecimiento de la ruta [13]. A su vez, se ha propuesto mantener rutas alternativas por si la ruta principal falla [4].

En la literatura existen diversos trabajos que emplean metaheurísticas para resolver problemas de optimización en MANETs y VANETs y se dividen en dos grandes grupos: aquellos que emplean las metaheurísticas directamente sobre elementos de la red para resolver *on-line* algún problema de eficiencia de la red [17], [18] y aquellos que las utilizan para optimizar *off-line* el funcionamiento del protocolo buscando una configuración eficiente del protocolo [19], [20], [21], como es el caso de este análisis.

Existe un número limitado de trabajos que apliquen la segunda estrategia en VANETs. El protocolo de transferencia de datos VDTP fue optimizado para mejorar la transferencia de archivos entre vehículos [22]. Los protocolos de *routing* AODV y OLSR fueron optimizados en términos de PDR, NRL y E2ED empleando un grupo de metaheurísticas mono-objetivo secuenciales en [5] y [8], respectivamente. OLSR también fue optimizado empleando un algoritmo MO, el NSGA-II [10].

El rendimiento de los algoritmos empleados en todos estos trabajos previos se ve limitado por el elevado coste computacional de la evaluación de la función objetivo. Así, diferentes análisis han reducido el coste computacional de las metaheurísticas mono-objetivo aplicando modelos paralelos [6], [9].

Nuestro trabajo trata el problema de optimización *off-line* de protocolos VANET empleando una técnica multi-objetivo paralela. Por lo que se resuelven dos de los principales problemas en este dominio: la obtención de un única solución condicionada por unos pesos preestablecidos y la evaluación de la función objetivo un número muy limitado de veces debido a su alto coste computacional.

IV. ALGORITMO PARALELO MULTI-OBJETIVO DE CÚMULO DE PARTÍCULAS PROPUESTO

Esta sección describe el algoritmo de cúmulo de partículas (SMPSO) utilizado para resolver el problema de optimización multi-objetivo del encaminamiento de datos en VANETs, así como el modelo de paralelización aplicado para mejorar la eficiencia de la búsqueda de configuraciones eficientes de AODV.

A. Algoritmo SMPSO

Las metaheurísticas MO (MMO) aparecen para resolver problemas de optimización multi-objetivo. Una MMO halla un conjunto de soluciones optimizadas *no dominadas* que se llama *frente de Pareto* y que representan diferentes compromisos de acuerdo con las funciones objetivo [23]. En este estudio se aplica la MMO SMPSO al encaminamiento de datos en VANETs. El problema consiste en optimizar dos objetivos en conflicto entre sí: maximizar la cantidad de datos que se intercambia y minimizar los tiempos de transferencia.

El pseudocódigo del SMPSO se presenta en el Algoritmo 1. Las soluciones se denominan *partículas* y se agrupan en *cúmulos* (C). Las soluciones no-dominadas se agrupan en un archivo de líderes (L) que se actualiza cada iteración de acuerdo a la *distancia de agrupamiento* o *crowding distance* (Línea 10). De forma iterativa sobre cada una de las partículas se aplican los operadores de actualización de velocidad y posición, de mutación (*operador de turbulencia*) y de evaluación (líneas 5-8). Finalmente, en la Línea 11 se actualiza el estado del cúmulo.

Algoritmo 1 Pseudocódigo de SMPSO.

```

 $C \leftarrow \text{inicializar\_cumulo}()$ 
 $L \leftarrow \text{inicializar\_archivo\_lideres}()$ 
while not condicion_de_parada() do
  for all  $particula_i \in C$  do
    calcular_velocidad( $particula_i$ )
    actualizar_posicion( $particula_i$ )
    mutar( $particula_i$ )
    evaluar( $particula_i$ )
  end for
   $L \leftarrow \text{actualizar\_archivo\_lideres}()$ 
  actualizar_estado_cumulo( $C$ )
end while

```

B. Modelo paralelo para SMPSO

La evaluación de soluciones en los problemas de optimización en VANET se realiza mediante procesos computacionalmente costosos de simulación de la red (tiempos de ejecución del orden de minutos). Esto obliga a limitar el número de evaluaciones de la función objetivo usando conjuntos de soluciones pequeños y/o aplicando un número de generaciones reducido, afectando negativamente la efectividad de la metaheurística empleada. Por ello proponemos la distribución de la evaluación de las partículas entre distintas unidades de procesamiento para que se ejecuten en paralelo. Así podemos incrementar el tamaño de los cúmulos y el número de generaciones.

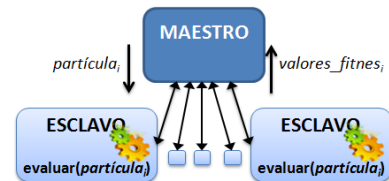


Fig. 3. Modelo maestro-esclavo aplicado en el pSMPSO.

El SMPSO paralelo (pSMPSO) propuesto está categorizado dentro del modelo *maestro-esclavo* [24].

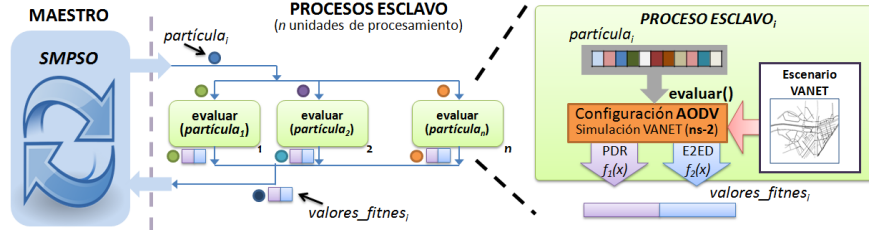


Fig. 2. Metodología empleada para la optimización multi-objetivo de AODV en VANETs.

El maestro realiza todas las operaciones del pSMPSO salvo la evaluación de las soluciones (Línea 8 del Algoritmo 2) que se distribuye a través de la arquitectura paralela (ver Fig. 3). Así, no existe ninguna diferencia entre la versión secuencial (un solo procesador) y la paralela en la forma de la exploración del espacio de búsqueda. Los algoritmos 2 y 3 presentan los pseudocódigos de los procesos maestro y esclavo del pSMPSO, respectivamente.

Algoritmo 2 Proceso maestro del pSMPSO.

```

C ← inicializar_cumulo()
L ← inicializar_archivo_lideres()
while not condicion_de_parada() do
  for all particula_i ∈ C do
    calcular_velocidad(particula_i)
    actualizar_posicion(particula_i)
    mutar(particula_i)
  end for
  for all particula_i ∈ C do in paralelo
    enviar_datos_a_esclavo_i(particula_i)
    esperar_datos_de_esclavo_i(particula_i, valores_fitness_i)
    mutar(particula_i)
  end for
  L ← actualizar_archivo_lideres()
  actualizar_estado_cumulo(C)
end while

```

Algoritmo 3 Proceso esclavo_i de pSMPSO.

```

particula_i ← esperar_datos_de_maestro()
valores_fitness_i ← evaluar(particula_i)
enviar_datos_de_esclavo_i(particula_i, valores_fitness_i)

```

V. OPTIMIZACIÓN DE AODV USANDO pSMPSO

Esta sección introduce la metodología llevada a cabo y presenta detalles sobre la codificación del problema, el método de evaluación de las soluciones y los diferentes operadores implementados.

A. Metodología general

En este estudio tratamos el problema de encontrar configuraciones de AODV de forma que maximicen el PDR y minimicen el E2ED cuando se emplea en VANETs. Para ello vamos a utilizar pSMPSO que hemos implementado utilizando las bibliotecas *jMetalCpp* [25] y *pthread*. Como se muestra en la Fig. 2 la evaluación de las soluciones (configuraciones) se realiza mediante su simulación en una VANET empleando el simulador de redes *network simulator 2 (ns-2)* [26] y analizando el PDR y el E2ED. Los valores de las dos métricas se envían al proceso maestro que los aplica para seguir con los cálculos.

B. Codificación del problema

El funcionamiento de AODV está gobernado por la configuración de 11 parámetros. Así, las partículas

se representan mediante un vector de 11 componentes, uno por cada uno de los parámetros (ver la Fig. 4). La codificación y el rango de cada uno de los componentes del vector vienen dados en la Tabla I.

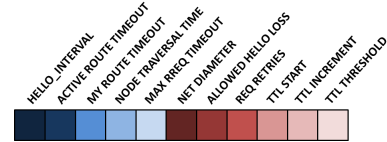


Fig. 4. Representación del vector solución.

C. Evaluación de soluciones

Este problema multi-objetivo tiene dos objetivos contrapuestos: maximizar el PDR y minimizar el E2ED. Así, hemos definido dos funciones objetivo, $f_1(s)$ y $f_2(s)$, dadas por las ecuaciones 1 y 2. En las cuales, $PDR(s)$ y $E2ED(s)$ son los PDR y E2ED medios de la VANET cuando se simula la configuración s , respectivamente. Por lo tanto, el problema de optimización multi-objetivo de AODV en VANETs viene dado por la minimización de $f_1(s)$ y $f_2(s)$.

$$f_1(s) = \frac{1}{PDR(s)} \quad f_1(s) \in [0,01, \infty) \quad (1)$$

$$f_2(s) = E2ED(s) \quad f_2(s) \in (0, \infty) \quad (2)$$

D. Operadores propuestos

En esta sección se presenta los operadores de iniciación y mutación definidos para optimizar AODV.

D.1 Inicialización de las soluciones

El operador de inicialización propuesto distribuye las soluciones de un cúmulo en diferentes áreas del espacio de búsqueda. Se divide el espacio de búsqueda en $num_particulas$ (tamaño del cúmulo) subespacios diagonales y se aloja en cada subespacio cada solución (ver Ecuación 3).

$$x_{s,i}^{(0)} = z_{(i,MIN)} + \rho^s \quad i \in [0, 10], s \in [0, num_particulas] \quad (3)$$

El valor inicial del componente i del vector solución s ($x_{s,i}^{(0)}$) viene dado por el valor ρ^s que se calcula a partir de $z_{(i,MIN)}$ y los límites de cada uno de los subespacios y un valor aleatorio $\beta \in [0, 1]$ (ver Ecuación 4). Los valores $z_{(i,MIN)}$ y $z_{(i,MAX)}$ son los valores mínimo y máximo del rango de valores que cada parámetro i pueden tomar (ver la Tabla I).

$$\rho^s = \left(\frac{s + \beta}{num_particulas} \right) \times (z_{(i,MAX)} - z_{(i,MIN)}) \quad (4)$$

D.2 Mutación o turbulencia de las partículas

El operador de mutación o turbulencia introduce nueva información generada de forma pseudo-aleatoria en el cúmulo. Para generar solo soluciones

factibles se ha definido dentro de los rangos de los parámetros de configuración de AODV.

La Ecuación 6 presenta el *movimiento* de la componente i de la solución s ($x_{s,i}^t$) que viene dado por β_i , un valor aleatorio uniformemente distribuido, y el rango de los parámetros de AODV (ver la Tabla I).

$$x_{s,i}^t = x_{s,i}^{(g)} + \beta_i \times (z_{(i,MAX)} - z_{(i,MIN)}) \quad (5)$$

$$\beta_i \in [-0,5, 0,5]$$

El nuevo valor de la componente i del vector solución s ($x_{s,i}^{(g+1)}$) viene definido por la Ecuación 6. Si el valor de $x_{s,i}^{temp}$ está dentro del rango de valores del parámetro de configuración i , entonces $x_{s,i}^{(g+1)} = x_{s,i}^{temp}$. Si $x_{s,i}^{temp} > z_{(i,MAX)}$ o $x_{s,i}^{temp} < z_{(i,MIN)}$, entonces a $x_{s,i}^{(g+1)}$ se le asigna el valor más alto o más bajo del rango de valores del parámetro i , respectivamente.

$$x_{s,i}^{(g+1)} = \begin{cases} x_{s,i}^{temp}, & \text{if } x_{s,i}^t \in [z_{(i,MAX)}, z_{(i,MIN)}] \\ z_{(i,MAX)}, & \text{if } new_x_{s,i} > z_{(i,MAX)} \\ z_{(i,MIN)}, & \text{if } new_x_{s,i} < z_{(i,MIN)} \end{cases} \quad (6)$$

VI. RESULTADOS EXPERIMENTALES

Esta sección presenta los aspectos principales de la experimentación: i) los escenarios VANET definidos para evaluar las comunicaciones, ii) los experimentos de configuración de pSMPSO, iii) los resultados en términos de eficacia y eficiencia y iv) la comparación de nuestra propuesta con el estado del arte. La plataforma paralela empleada para ejecutar los experimentos cuenta con un procesador Opteron 6172 Magny-Core de 24 núcleos de 2.1 GHz, con 24 GB de RAM y CentOS Linux como sistema operativo.

A. Escenarios VANET para la simulación

Se han definido 30 escenarios VANET a partir de los datos reales de una zona residencial de la ciudad de Málaga (España). Estos escenarios cubren tres áreas de diferente tamaño de la zona que se han llamado U1, U2 y U3 (ver la Fig. 5). En estos escenarios distintos números de vehículos se mueven siguiendo patrones inteligentes de conducción y cumpliendo con las normas de tráfico (ver la Tabla II).

TABLA II
ESPECIFICACIÓN DE LOS ESCENARIOS VANET.

Nombre	Tamaño del área	Número de vehículos	Número de transmisiones
U1	120000 m ²	20	10
		20	10
		20	10
U2	240000 m ²	30	15
		40	20
		30	15
		45	23
U3	360000 m ²	60	30
		75	38
		90	45
		105	53

Los resultados de las comunicaciones VANET se obtienen del simulando la red durante 180 segundos.

Los nodos se han configurado siguiendo los estándares para entornos vehiculares. La Tabla III resume las características principales de la red simulada. Los 30 escenarios representan una variedad de situaciones diferentes (variando tamaño de la red, densidad del tráfico y patrones de comunicación).

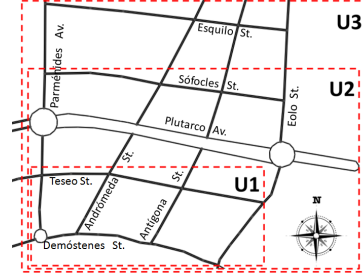


Fig. 5. Mapa de los escenarios VANET.

TABLA III
DETALLES DE LAS COMUNICACIONES VANET.

Parámetro	Valor/Protocolo
Modelo de propagación	Nakagami
Ancho de banda del canal	6 Mbps
Capas física y de acceso al medio	IEEE 802.11p
Capa de red	AODV
Capa de transporte	UDP
Tamaño de los paquetes de datos	512 bytes
Tasa de datos	64/128/256 kbps
Duración de las transmisiones	60 s

B. Configuración del pSMPSO

En este estudio se ha analizado un pSMPSO de 24 partículas que llevará a cabo las evaluaciones de todas las soluciones en paralelo (24 hebras). A continuación, se indica como se ha configurado la probabilidad de mutación y la condición de parada.

B.1 Probabilidad de mutación

Para configurar la probabilidad de mutación (p_M) se han realizado unos experimentos iniciales en los que se ha empleado un pSMPSO que realiza 300 generaciones y emplea el escenario U2 de 20 vehículos para la evaluación de las soluciones. El conjunto de candidatos a ser la probabilidad de mutación del pSMPSO en este problema es el definido por $\{\frac{1}{4L}=0,023, \frac{1}{2L}=0,045, \frac{1}{L}=0,091, \frac{1}{0,5L}=0,182\}$ siendo L the el tamaño del vector solución ($L = 11$).

Se han llevado a cabo 10 ejecuciones independientes para cada p_M y como condición de parada realizar 450 generaciones. Se ha analizado el valor del *hipervolumen* [27] del frente de Pareto obtenido para comparar los resultados. Como se muestra en la Tabla IV la configuración que obtiene el mayor *hipervolumen mediano* (0,758) es $p_M=0,091$, por lo que esa es la probabilidad elegida.

TABLA IV
HIPERVOLUMEN MEDIANO OBTENIDO PARA CADA p_M .

p_M	0,023	0,045	0,091	0,182
Hipervolumen	0,738	0,755	0,785	0,747

B.2 Condición de parada

Para poder hacer un estudio más fiable sobre la eficiencia del pSMPSO se ha decidido utilizar un criterio de parada basado en la calidad de las soluciones obtenidas. De este modo se evita poner como criterio de parada un número dado de generaciones dado porque condicionaría sustancialmente los tiempos de ejecución. Por ello, el criterio de parada se ha definido como obtener un frente de Pareto cuyo hipervolumen sea igual o superior al hipervolumen mediano obtenido con $p_M=0,091$ (0,785), o sobrepasar las 450 generaciones (evitando tiempos de ejecución excesivos).

C. Resultados del proceso de optimización MO

Se han realizado 30 ejecuciones independientes del pSMPSO para tratar el problema de optimización de AODV en VANETs y hemos evaluado los frentes de Pareto obtenidos calculando su hipervolumen (I_{HV}), ϵ (I_ϵ) y Δ (I_Δ). La Tabla V muestra los valores mínimo (Mín), mediano (Med) y máximo (Máx) obtenidos para cada una de las métricas.

TABLA V

RESULTADOS PARA LAS 30 EJECUCIONES DE pSMPSO.

Métrica	Mín	Med	Máx
Hipervolumen (I_{HV})	0,777	0,786	0,799
Epsilon (I_ϵ)	2,195	2,807	4,493
Spread (I_Δ)	0,603	0,706	0,911

El máximo valor de hipervolumen es 0,799, que es mayor que el valor de umbral de hipervolumen establecido en la condición de parada. Esto se debe a que han habido cinco ejecuciones que han parado tras haber llevado a cabo el máximo de generaciones.

La Fig. 6 muestra la distribución de las soluciones en todos los frentes de Pareto obtenidos. El frente de Pareto optimizado se obtiene de todas las soluciones no-dominadas para las 30 ejecuciones independientes. Estas soluciones presentan diferentes valores para cada uno de las funciones objetivo.

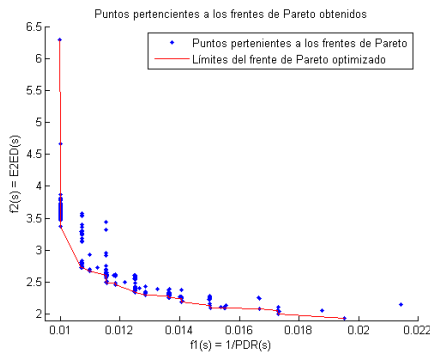


Fig. 6. Frentes de Pareto obtenidos por pSMPSO.

Para comparar las soluciones obtenidas se utiliza su distancia al *vector ideal* (vi) [28]. Las componentes del vector ideal contienen el mejor valor obtenido para cada uno de los objetivos. Para nuestro problema $vi=[0, 01, 1, 936]$. La Tabla VI muestra los resultados de la función objetivo (y los valores para las métricas PDR y $E2ED$) para las soluciones que han obtenido los mínimos de $f_1(s)$ y $f_2(s)$, es decir, el

máximo PDR ($Máx-PDR$) y mínimo $E2ED$ ($Mín-E2ED$), respectivamente. También se muestran los mismos valores para la solución que obtiene la menor distancia con vi ($Mín-Dist$), de una hipotética solución que representa la configuración que se sugiere en el AODV RFC ($AODV-RFC$) y la mediana para cada una de las funciones objetivo ($pSMPSO_{med}$).

TABLA VI

VALORES DE LAS FUNCIONES OBJETIVO Y RESULTADOS EN TÉRMINOS DE PDR Y E2ED.

	$f_1(s)$	$PDR(s)$ (en %)	$f_2(s) = E2ED(s)$ (en ms)
Mín-Dist	0,012	86,206	2,485
Máx-PDR	0,010	100,00	3,461
Mín-E2ED	0,019	51,282	1,937
AODV-RFC	0,012	80,521	28,882
pSMPSO_{med}	0,011	89,286	3,101

Al ser dos objetivos contrapuestos los que se optimizan en este problema (maximizar PDR y minimizar $E2ED$) las soluciones que obtienen los *óptimos* para cada uno de los objetivos ($Máx-PDR$ y $Mín-E2ED$) obtienen resultados muy pobres para el otro en comparación con las demás. La solución que representa la configuración de los expertos ($AODV-RFC$) obtiene PDR competitivo de 80,521 % aunque es menor que la mediana de las soluciones obtenidas con pSMPSO (89,286 %), pero el $E2ED$ es de 28,882 ms., de una orden de magnitud mayor que cualquiera de las configuraciones calculadas de forma automática. Así, la solución que minimiza la distancia al vector ideal obtiene una buena solución de compromiso ($PDR=86,201$ % y $E2ED=2,485$ ms.).

D. Coste y eficiencia computacional

En esta sección vamos a analizar tanto el coste computacional como la eficiencia del modelo paralelo utilizado. La Tabla VII muestra los valores mínimo (Mín), mediano (Med) y máximo (Máx) del número de generaciones llevadas a cabo y el tiempo requerido por las 30 ejecuciones independientes.

TABLA VII

COSTE COMPUTACIONAL DE pSMPSO OPTIMIZANDO AODV.

	Mín	Med	Máx
Número de generaciones	77	337	450
Tiempo de ejecución (minutos)	103	452	616

El tiempo máximo de ejecución supera las 10 horas (616 mins.), este tiempo se ha dado cuando el algoritmo ha parado por alcanzar el valor máximo de generaciones. La ejecución que han convergido con mayor celeridad ha tardado 103 mins. en finalizar. La mediana de los tiempos de ejecución queda establecida en 452 mins (7 horas y 32 minutos).

Las métricas más empleadas evaluar el rendimiento de un algoritmo paralelo son la *eficiencia computacional* (e_m) y el *speedup* (S_m) [24]. El speedup mide cuantas veces es más rápido un algoritmo paralelo que su versión secuencial y se define como la relación entre el tiempos que requieren un algoritmo

secuencial (T_1) y su versión paralela (T_m) que emplea m procesadores en resolver un mismo problema (Ecuación 7). La eficiencia computacional es valor normalizado del speedup con respecto al número de procesadores (Ecuación 8).

$$S_m = \frac{T_1}{T_m} \simeq \frac{E[T_1]}{E[T_m]} \quad (7)$$

$$e_m = \frac{S_m}{m} \quad (8)$$

Para poder calcular los valores de speedup se ha tenido que ejecutar SMPSO (secuencial) en las mismas condiciones que pSMPSO para poder disponer de su tiempo de ejecución (T_1). La Tabla VIII presenta los resultados experimentales sobre el rendimiento del pSMPSO sobre 24 núcleos ($m=24$), donde $E[T_{24}]$ y $E[T_1]$ son los tiempos medios empleados para ejecutar pSMPSO y SMPSO, respectivamente.

TABLA VIII

EVALUACIÓN DE LA EFICIENCIA COMPUTACIONAL Y SPEEDUP.

$E[T_{24}]$ (mins.)	$E[T_1]$ (mins.)	Speedup	Eficiencia
422,190	10383,449	20,829	0,868

La versión secuencial del algoritmo ha necesitado de media 10383 minutos (6,1 días) para terminar. Así, el speedup conseguido es de 20,829, lo que significa una eficiencia computacional de 86,8%. Estos resultados justifican la aplicación del model paralelo que se plantea en este estudio.

E. Validación de las configuraciones

Para confirmar la eficacia de la metodología de optimización que se presenta en este trabajo hemos realizado una serie de experimentos de validación. Estos experimentos consisten en la comparativa de diferentes propuestas de AODV en el estado del arte en 30 escenarios VANET (ver Sección VI-A) teniendo en cuenta tres métricas distintas: PDR, NRL y E2ED. Las configuraciones que se comparan son: la configuración obtenida que minimiza la distancia con el *vi* y que llamaremos *pSMPSO*, la configuración estándar propuesta en el RFC 3651 (*RFC*), una versión optimizada utilizando PSO (*GN*) en [5] y otra optimizada aplicando una versión paralela mono-objetivo de PSO (*pPSO*) propuesta en [6].

La Tabla IX muestra los valores medios obtenidos de cada una de las métricas agrupadas por el área geográfica (U1, U2 y U3) y para los 30 escenarios (Global). En negrita se resaltan los mejores valores para cada métrica.

En términos de cantidad de datos entregados (PDR) se observa que en general la configuración que proponemos ofrece el mejor comportamiento (en *Global* el PDR de pSMPSO es 63,643%). Si diferenciamos por áreas geográficas, pSMPSO obtiene los mejores resultados en U1 y U2 y RFC en U3.

Si analizamos la carga de red generada por el protocolo de encaminamiento (NRL), no se observa una tendencia clara aunque globalmente GN genera de media menos carga que los demás. Los resultados son dependientes del tamaño del área geográfica donde

TABLA IX

VALORES MEDIOS DE LAS MÉTRICAS DE CALIDAD DE SERVICIO PARA LAS CONFIGURACIONES DE AODV COMPARADAS.

Configuraciones	Métricas		
	PDR	NRL	E2ED
<i>Escenarios U1</i>			
pSMPSO	67,936	0,430	15,358
RFC	55,779	0,544	56,067
pPSO	62,997	0,380	36,968
GN	63,635	0,341	48,657
<i>Escenarios U2</i>			
pSMPSO	92,738	0,031	4,016
RFC	78,184	0,040	17,100
pPSO	72,737	0,032	12,926
GN	72,259	0,033	13,101
<i>Escenarios U3</i>			
pSMPSO	26,955	0,167	3,916
RFC	56,890	0,357	41,403
pPSO	26,378	0,096	8,620
GN	25,475	0,097	5,178
<i>Global</i>			
pSMPSO	63,643	0,209	7,763
RFC	63,618	0,314	38,190
pPSO	54,037	0,169	19,504
GN	53,789	0,157	22,312

se define la VANET. Así, para las áreas U1, U2 y U3 las configuraciones que han generado la menor carga de red son GN (0,341), pSMPSO (0,031) y pPSO (0,096), respectivamente.

Finalmente, en términos de tiempos de entrega de paquetes (E2ED), se observa que la configuración que hemos calculado empleando pSMPSO obtiene claramente los mejores resultados para todas las áreas geográficas (15,358, 4,016 y 3,916 ms. para las áreas U1, U2 y U3, respectivamente). Esto es realmente importante, puesto que en este tipo de redes las aplicaciones requieren de tiempos de entrega mínimos menores de 20 ms.

Así, con los resultados obtenidos en los experimentos de validación podemos concluir que la configuración obtenida por pSMPSO es la más equilibrada y ofrece claramente los mejores resultados en términos de PDR y E2ED. Porque aunque RFC entrega una mayor cantidad de paquetes en el área U3, los tiempos de entrega que requiere (E2ED de RFC es 41,4 ms.) son de una orden de magnitud mayor que los necesitados por las otras configuraciones.

VII. CONCLUSIONES Y TRABAJO FUTURO

Este trabajo trata el problema de la optimización del encaminamiento de datos en redes vehiculares. Para ello se propone el uso de pSMPSO, una metaheurística multi-objetivo paralela, para optimizar el protocolo AODV. Esta es una propuesta novedosa en este ámbito, puesto que en el estado del arte apenas se han combinado una estrategia multi-objetivo con un modelo paralelo para resolver este problema.

En este estudio, se ha implementado una versión paralela de pSMPSO que emplea 24 unidades de proceso para evaluar en paralelo 24 partículas (soluciones) empleando el paradigma maestro-esclavo. Así, se ha conseguido un speedup de 20,8 sobre la versión secuencial, lo que significa una eficiencia computacional de 86,8%. Estos resultados justifican el uso del modelo paralelo planteado.

A su vez, hemos realizado una serie de experimentos de validación para confirmar la eficacia de la metodología de optimización empleada para optimizar AODV. Los resultados obtenidos comparando distintas propuestas del estado del arte con una de las configuraciones optimizadas claramente muestran que la configuración que hemos optimizado empleando pSMPSO es la configuración más eficiente, mas equilibrada y aquella que entrega una mayor cantidad de datos en el menor tiempo posible.

Las principales líneas de trabajo futuro son: i) analizar nuevos algoritmos multi-objetivo y operadores que puedan adaptarse mejor al problema que se trata, ii) aplicar un nuevo modelo paralelo que permita la evaluación de las soluciones sobre varios escenarios distintos durante el proceso de optimización, y iii) analizar los protocolos optimizados en pruebas de concepto reales empleando vehículos equipados con dispositivos de comunicación inalámbrica.

AGRADECIMIENTOS

Jamal Toutouh es beneficiario de una beca FPU (AP2010-3108) otorgada por el Gobierno de España. A su vez, este trabajo está parcialmente financiado por el Ministerio de Economía y Competitividad y FEDER dentro del proyecto TIN 2011-28194 (roadME) y el proyecto número 8.06/5.47.4142 en colaboración con la VSB-Technical University de Ostrava. Universidad de Málaga. Campus de Excelencia Internacional Andalucía Tech.

REFERENCIAS

- [1] K.C. Lee, U. Lee, and M. Gerla, *Survey of Routing Protocols in Vehicular Ad Hoc Networks*, chapter 8, pp. 149–170, Eds. IGI Global, 2009.
- [2] C. Perkins, E. Royer, and S. Das, “Ad hoc on demand distance vector (AODV) routing (RFC 3561),” Tech. Rep., IETF MANET Working Group (August. 2003), 2003, URL: <http://tools.ietf.org/html/rfc3561>.
- [3] B. Li, Y. Liu, and G. Chu, “Improved AODV routing protocol for vehicular ad hoc networks,” in *3rd International Conference on Advanced Computer Theory and Engineering*, aug. 2010, vol. 4, pp. 337–340.
- [4] L. Chao and L. Ping’an, “An efficient routing approach as an extension of the AODV protocol,” in *Future Computer and Communication (ICFCC), 2010 2nd International Conference on*, may 2010, vol. 1, pp. 95–99.
- [5] J. García-Nieto and E. Alba, “Automatic parameter tuning with metaheuristics of the AODV routing protocol for vehicular ad-hoc networks,” in *EvoApplications (2)*. 2010, vol. 6025 of *Lecture Notes in Computer Science*, pp. 21–30, Springer.
- [6] J. Toutouh and E. Alba, “Parallel swarm intelligence for VANETs optimization,” in *P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), 2012 Seventh International Conference on*. IEEE, nov. 2012, pp. 285–290.
- [7] A.J. Nebro, J.J. Durillo, J. Garcia-Nieto, C.A. Coello Coello, F. Luna, and E. Alba, “SMPSO: A new pso-based metaheuristic for multi-objective optimization,” in *Computational intelligence in multi-criteria decision-making, 2009. mcdm’09. ieeee symposium on*. IEEE, 2009, pp. 66–73.
- [8] J. Toutouh, J. Garcia-Nieto, and E. Alba, “Intelligent OLSR routing protocol optimization for VANETs,” *Vehicular Technology, IEEE Transactions on*, vol. 61, no. 4, pp. 1884–1894, 2012.
- [9] J. Toutouh, S. Nesmachnow, and E. Alba, “Fast energy-aware OLSR routing in VANETs by means of a parallel evolutionary algorithm,” *Cluster Computing*, pp. 1–16, 2012.
- [10] J. Toutouh and E. Alba, “Multi-objective OLSR optimization for VANETs,” in *Wireless and Mobile Computing, Networking and Communications (WiMob), 2012 IEEE 8th International Conference on*, 2012, pp. 571–578.
- [11] Y. Lin, Y. Chen, and S. Lee, “Routing protocols in vehicular ad hoc networks: A survey and future perspectives,” *J. Inf. Sci. Eng.*, vol. 26, no. 3, pp. 913–932, 2010.
- [12] E. Spaho, L. Barolli, G. Mino, F. Xhafa, V. Kolici, and R. Miho, “Performance evaluation of AODV, OLSR and DYMO protocols for vehicular networks using CAVENET,” in *13th International Conference on Network-Based Information Systems (NBIS), 2010*, sept. 2010, pp. 527–534.
- [13] D. Sutariya and S. Pradhan, “An improved AODV routing protocol for VANETs in city scenarios,” in *International Conference on Advances in Engineering, Science and Management (ICAESM), 2012*, march 2012, pp. 575–581.
- [14] E. Alba, B. Dorronsoro, F. Luna, A.J. Nebro, P. Bouvry, and L. Hogie, “A cellular MOGA for optimal broadcasting strategy in metropolitan MANETs,” *Computer Communications*, vol. 30, no. 4, pp. 685–697, 2007.
- [15] J. Ledy, A.-M. Poussard, R. Vauzelle, B. Hilt, and H. Boeglen, “AODV enhancements in a realistic VANET context,” in *Wireless Communications in Unusual and Confined Areas (ICWCUCA), 2012 International Conference on*, 2012, pp. 1–5.
- [16] B. Li, Y. Liu, and G. Chu, “Optimized AODV routing protocol for vehicular ad hoc networks,” in *Mobile Congress (GMC), 2010 Global*, oct. 2010, pp. 1–4.
- [17] G.A. Di Caro, F. Ducatelle, and L. M. Gambardella, “AntHocNet: an adaptive nature-inspired algorithm for routing in mobile ad hoc networks,” *European Transactions on Telecommunications*, vol. 16, no. 5, pp. 443–455, 2005.
- [18] C. Huang, Y. Chuang, and K. Hu, “Using particle swarm optimization for QoS in ad-hoc multicast,” *Eng. Appl. Artif. Intell.*, vol. 22, no. 8, pp. 1188–1193, 2009.
- [19] J.J. Durillo, A.J. Nebro, F. Luna, and E. Alba, “A study of master-slave approaches to parallelize NSGA-II,” in *Parallel and Distributed Processing, 2008. IPDPS 2008. IEEE International Symposium on*, April 2008, pp. 1–8.
- [20] C. Segura, A. Cervantes, A. J. Nebro, M. D. Jaraíz-Simón, E. Segredo, S. García, F. Luna, J.A. Gómez-Pulido, G. Miranda, C. Luque, E. Alba, M. A. Vega-Rodríguez, C. León, and I. M. Galván, “Optimizing the DFCN broadcast protocol with a parallel cooperative strategy of multi-objective evolutionary algorithms,” in *Evolutionary Multi-Criterion Optimization*, M. Ehrgott, C.M. Fonseca, X. Gandibleux, J. Hao, and M. Sevaux, Eds., vol. 5467 of *Lecture Notes in Computer Science*, pp. 305–319. Springer Berlin Heidelberg, 2009.
- [21] H. Cheng and S. Yang, “Genetic algorithms with immigrant schemes for dynamic multicast problems in mobile ad hoc networks,” *Eng. Appl. Artif. Intell.*, vol. 23, pp. 806–819, August 2010.
- [22] J. García-Nieto, J. Toutouh, and E. Alba, “Automatic tuning of communication protocols for vehicular ad hoc networks using metaheuristics,” *Engineering Applications of Artificial Intelligence*, vol. 23, no. 5, pp. 795–805, 2010.
- [23] Ajith A., Lakhmi J., and Robert G., *Evolutionary Multiobjective Optimization. Theoretical Advances and Applications.*, Springer London, 2005.
- [24] E. Alba and M. Tomassini, “Parallelism and evolutionary algorithms,” *Evolutionary Computation, IEEE Transactions on*, vol. 6, no. 5, pp. 443–462, 2002.
- [25] J.J. Durillo and A.J. Nebro, “jMetal: A Java framework for multi-objective optimization,” *Advances in Engineering Software*, vol. 42, pp. 760–771, 2011.
- [26] S. McCanne, S. Floyd, and K. Fall, “ns2 (network simulator 2),” <http://www.nrg.ee.lbl.gov/ns/>, Retrieved October 2011.
- [27] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*, Wiley-Interscience Series in Systems and Optimization. John Wiley & Sons, Chichester, 2001.
- [28] C.A.C. Coello, G.B. Lamont, and D.A. Van Veldhuizen,

Evolutionary algorithms for solving multi-objective problems, vol. 5, Springer-Verlag New York Inc, 2007.