

Un Enfoque Multiobjetivo en Redes Periódicas de Tiempo Continuo

Diego Bosch¹, Benjamín Barán^{1,2}, and Uri Yael¹

¹ Universidad Católica “Nuestra Señora de la Asunción”. Asunción, Paraguay
`{diegojbosch, ing.uriyael}@gmail.com`

² Universidad Nacional del Este. Ciudad del Este, Paraguay
`bbaran@cba.com.py`

Resumen Una red dinámica es conocida como red periódica cuando se compone de enlaces que se conectan y desconectan periódicamente. Trabajos anteriores han tratado el problema teniendo en cuenta sólo tiempos discretos. En contrapartida, el presente trabajo utiliza las mismas funciones objetivo que estos trabajos, considerando por primera vez el tiempo continuo, además de una variabilidad en el tiempo que dura cada era (lapso de tiempo entre dos cambios de topología). El trabajo modela el problema en un contexto multi-objetivo y propone el uso de un Algoritmo Evolutivo Multi-Objetivo (el *Strength Pareto Evolutionary Algorithm*), implementado y luego comparado con respecto a los principales algoritmos del estado del arte como: DSDV, DSR, AODV, EG Shortest y EG Foremost, demostrando en las simulaciones que sería el preferido en un contexto puramente multi-objetivo que considera siete funciones objetivo: tiempo de viaje y tiempo de vida (promedio, varianza y peor caso), así como cantidad de saltos promedio.

Keywords: Grafo evolutivo, redes dinámicas, redes periódicas, optimización multi-objetivo, solución preferida.

1. Introducción

Una red dinámica es conocida como red periódica cuando se compone de enlaces que se conectan y desconectan con un intervalo regular de tiempo, conocido como periodo (T) [1]. Para transmitir un paquete de datos se necesita un camino desde el nodo fuente (remitente) hasta el nodo destino (destinatario). En el camino, existen nodos intermedios que actúan reenviando el paquete hasta llegar al destino. Si un nodo intermedio está ausente en un momento dado, el nodo predecesor debe almacenar el paquete hasta que el enlace entre ellos se active, o tomar la decisión de cambiar el camino, según sea la política de encaminamiento.

Las redes arriba citadas se analizan usando un formalismo para representar la conectividad de una red en cualquier instante, conocido como *Evolving Graph* (EG) [10], [11]. Un EG puede entenderse como una secuencia indexada de subgrafos que representan la configuración de la red en cada era e , como ilustra la Figura 1, donde por ejemplo, el enlace entre los nodos A y B solo está presente en la tercera y la décima era de cada ciclo (o periodo) que dura 15 eras.

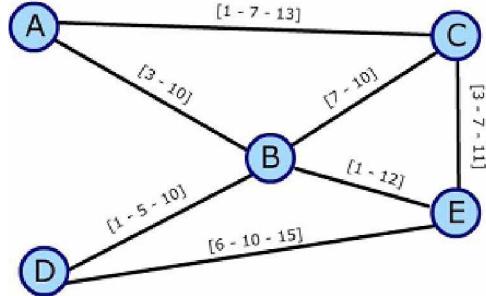


Figura 1. Grafo evolutivo que representa la topología de una red pequeña, donde en cada enlace se identifica las eras en la que se encuentra activo

Las redes periódicas han llegado a ser muy útiles en el modelado de redes de comunicación. Por citar algunos ejemplos, se las utiliza en el estudio de Satélites de Órbita Baja (*Low Earth Orbit - LEO*), redes de sensores, redes móviles del tipo *ad-hoc* (MANET) [8], así como en el proyecto *One Laptop Per Child* (OLPC) que utiliza este modelado con un fin educativo [4], entre otras aplicaciones posibles.

2. Contexto del trabajo

Este trabajo sigue la línea de investigación propuesta por Yael y Barán [16], [17], donde se aborda el problema de encaminamiento en redes dinámicas periódicas, modelándolo como un problema multi-objetivo. En dichos trabajos, Yael y Barán proponen un algoritmo evolutivo multi-objetivo, capaz de hallar un conjunto de soluciones Pareto óptimas [12], que no sólo encuentra individuos competitivos con respecto a las soluciones calculadas por otros algoritmos del estado del arte, sino que además logra calcular individuos que dominan (en el sentido Pareto [5]) a las soluciones calificadas como óptimas, cuando una sola función objetivo es considerada. Aquí corresponde recordar que una solución multi-objetivo es mejor que otra (o la domina) si no es peor en ningún objetivo y es estrictamente mejor en al menos un objetivo [5]. Trabajos anteriores [10], [11], [16] solo consideran la duración de las eras como una variable discreta t y no asignan un valor específico en segundos a dicha variable. Esto resulta en que las soluciones son comparadas, entre otras formas, por la cantidad de eras transcurridas y, por lo tanto, dichos trabajos asumen implícitamente que la duración en segundos de dos eras cualesquiera, es siempre la misma.

Los trabajos [10], [11], [16] utilizan Grafos Evolutivos que permiten implementar el concepto de camino (o viaje) compuesto por pares ordenados $\langle n, e \rangle$, donde n es un nodo y e una era en la que el nodo se encuentra conectado a la red. El presente trabajo propone por primera vez asignar a cada era una duración en segundos, considerando para ello un proceso aleatorio pues en la mayoría de los casos prácticos este valor no es necesariamente constante [3]. De esta forma,

la misma era en ciclos distintos puede tener una duración ligeramente diferente al ser considerada como una variable aleatoria. Esto permite simular diferencias en la duración de las eras, debido por ejemplo a márgenes de error en los cronogramas o fenómenos atmosféricos que impactan sobre la conectividad real de la red al momento efectivo de su utilización.

Al realizar simulaciones considerando eras cuyas duración es aleatoria, se espera que las soluciones del Algoritmo Evolutivo Multi-objetivo sean mejores en comparación a las soluciones tradicionales utilizando algoritmos mono-objetivo presentados en los trabajos de referencia [16], [17].

3. Simulaciones

3.1. Simulador

Siguiendo la línea propuesta en [16], [17], este trabajo modifica el simulador desarrollado por los autores en dichos trabajos de forma a considerar tiempos físicos (medidos en segundos) para cada era. Recordando que el tiempo de duración de una era es una variable aleatoria, esta aleatoriedad es representada mediante el uso de diferentes escenarios que reflejen el azar. Esto es, un escenario es una realización concreta de valores que puede ir tomando cada era durante el tiempo que dure una simulación. Estos escenarios pueden ser generados en forma aleatoria a partir de una función de distribución de probabilidad, como se hace en este trabajo, u obtenerse a partir de datos históricos o incluso ser propuestos por un experto.

Cabe destacar que en las simulaciones que se presentarán se considera que los paquetes siempre llegan a destino dado que si el enlace deja de estar disponible en una era, los paquetes todavía no enviados pueden ser calendarizados para ser transmitidos en el siguiente ciclo, dado que la red tiene una periodicidad T asegurada y no se consideran restricciones en los nodos.

3.2. Funciones objetivo

Trabajos anteriores [16], [17] han calculado un conjunto de soluciones óptimas teniendo en cuenta la optimización de varias funciones objetivos como: la cantidad promedio de saltos efectuados por los paquetes para llegar al nodo destinatario, la cantidad promedio de eras vividas por los paquetes hasta que todos ellos culminan su camino y la cantidad promedio de eras durante las cuales los paquetes viajan a través de la red hasta llegar a destino [16]. Este trabajo utilizará las mismas funciones objetivo de modo a comparar las simulaciones y los resultados obtenidos. La *cantidad de saltos* mide cuántos enlaces fueron atravesados por un paquete para llegar desde el nodo remitente hasta el nodo destino; el conjunto de caminos más cortos es el calculado con el algoritmo de Dijkstra [10]. El *tiempo de vida* es el tiempo transcurrido desde que se crea un paquete para ser enviado, hasta que llega a su destino. El *tiempo de viaje* es el tiempo transcurrido desde que un paquete efectivamente sale del remitente y hasta que

llega al destinatario. Es decir, no se considera el tiempo transcurrido entre que se crea un paquete y se inicia su transmisión efectiva. Notar que el tiempo de vida es igual al tiempo de viaje más el tiempo de espera.

Debido a que cada algoritmo se simulará sobre N escenarios que caracterizarán la distribución de probabilidad de las variables aleatorias del problema considerado, surgen nuevas alternativas al definir las funciones objetivo a partir de las arriba descritas. Para este trabajo se ha decidido trabajar con el promedio, varianza y peor caso para las funciones objetivo que consideran tiempos, dando como resultado siete funciones objetivo: f_1 : promedio de la cantidad de saltos, f_2 : peor caso del tiempo de viaje, f_3 : promedio del tiempo de viaje, f_4 : varianza del tiempo de viaje, f_5 : peor caso del tiempo de vida, f_6 : promedio del tiempo de vida y f_7 : varianza del tiempo de vida.

3.3. Entorno de simulación

Los algoritmos evaluados en las simulaciones que siguen son los utilizados en [16], [17]: *Destination-Sequenced Distance-Vector* (DSDV) [13], *Dynamic Source Routing* (DSR) [9], *Ad-hoc On-Demand Distance Vector* (AODV) [14], *Evolving Graph Shortest* (EG Shortest), [10] y *Evolving Graph Foremost* (EG Foremost) [10], además del *Strength Pareto Evolutionary Algorithm* (SPEA) [18], utilizando los siguiente parámetros en las simulaciones: cantidad de individuos por generación = 10, cantidad máxima de individuos en la población externa = 1000, probabilidad de cruzamiento = 0.5 y probabilidad de mutación = 0.25.

El simulador originalmente programado en Ruby [16] fue adaptado para simular el funcionamiento de los algoritmos en la red periódica de la Figura 1. Las simulaciones fueron ejecutadas en una máquina con un procesador Intel Core i5, 2.27 GHz y memoria RAM de 6.00 GB. La duración de las simulaciones, dependiendo del conjunto de escenarios utilizados fueron variando desde algunas horas, varios días y hasta un poco más de una semana. Cabe destacar que en general los algoritmos mono-objetivo necesitaron un tiempo de simulación considerablemente menor al SPEA.

Las pruebas que se presentan a continuación se realizaron utilizando la red periódica de la Figura 1. Al comenzar una simulación, lo primero que se hace es generar la cantidad de escenarios que se necesite. En la Tabla 1 se puede ver un ejemplo de una matriz generada para un escenario correspondiente a la red antes mencionada (Figura 1), donde cada fila representa un ciclo (letra C), cada columna representa una era (letra e) y cada casilla representa la duración en segundos, recordando que en una red periódica cada ciclo tiene un número constante de eras; por ejemplo, en la Tabla 1, los 5 ciclos analizados están compuestos de 15 eras, cada uno.

Todas las pruebas fueron realizadas sobre cuatro escenarios distintos que fueron generados al comenzar la simulación. Al considerar cuatro escenarios ya se tiene una razonable variedad de datos distintos, haciendo que la simulación pueda ser viable en cuanto a recursos y tiempo consumido. Una mayor cantidad de escenarios es lógicamente preferible pero requeriría un tiempo de ejecución mayor, como así también una cantidad de recursos muy grande para la realidad

Tabla 1. Ejemplo de escenario, donde E representa una era y C representa un ciclo

	e1	e2	e3	e4	e5	e6	e7	e8	e9	e10	e11	e12	e13	e14	e15
C1	0,15	0,14	0,22	0,18	0,27	0,15	0,14	0,22	0,18	0,27	0,15	0,14	0,22	0,18	0,75
C2	0,16	0,18	0,20	0,12	0,17	0,16	0,19	0,20	0,12	0,17	0,16	0,18	0,20	0,16	0,71
C3	0,44	0,29	0,26	0,27	0,15	0,24	0,29	0,26	0,27	0,18	0,24	0,29	0,26	0,27	0,15
C4	0,10	0,16	0,16	0,10	0,23	0,10	0,16	0,16	0,10	0,23	0,10	0,16	0,19	0,10	0,23
C5	0,29	0,20	0,23	0,26	0,29	0,29	0,20	0,23	0,24	0,29	0,29	0,20	0,23	0,26	0,29

concreta en que se realizó este trabajo. Para determinar la duración de cada era se utilizó una distribución de probabilidad triangular, escogida empíricamente de forma a lograr que los paquetes lleguen razonablemente a destino.

En lo que sigue, se asume que el tiempo que tarda en viajar un paquete de un nodo a otro es de 100 milisegundos (es decir 0,1 segundos). Las pruebas se realizaron con un tráfico que contenía 31 paquetes que se van creando dentro de la red ilustrada en la Figura 1. Para una red de este tamaño se puede asumir que se trata de un tráfico bastante congestionado. Entonces, de manera a que varios paquetes no puedan ser enviados dentro de una misma era por falta de tiempo, se definió que las eras duraban entre 0,1 y 0,3 segundos. De este modo se puede ver como se comportan los algoritmos simulados a medida que las eras y por ende los ciclos, van avanzando en el problema ejemplo presentado. En el caso de que un paquete no pueda ser enviado en una era debido a que el tiempo de la era finaliza, se lo calendariza para ser enviado en la misma era pero en el siguiente ciclo. De esta manera, todos los paquetes en algún momento llegarán necesariamente a su destino final para la red dinámica de la Figura 1.

3.4. Esquema de simulación

En la Figura 2 se puede apreciar el esquema a seguir en las simulaciones, donde las siglas utilizadas son: (N) número máximo de escenarios, (E_n) un escenario de los N posibles, (AB) enlace del nodo A al nodo B, (AC) enlace del nodo A al nodo C, (S_m) una solución SPEA de las m posibles, (FO) función objetivo, (t) tiempo medido en segundos, además de utilizar las siglas correspondientes a cada algoritmo arriba mencionado.

En primer lugar se utiliza la distribución de probabilidad para los escenarios, con la cual se van generando los N escenarios que se necesitan para empezar la simulación (pasos 1 y 2 de la Figura 2). Luego, el esquema de simulación se divide de acuerdo al tipo de algoritmo a simular (pasos 3 y 4 de la Figura 2). Para los algoritmos mono objetivo primeramente se tiene como entrada un escenario para luego probar la solución encontrada sobre cada uno de los N escenarios considerados. Como resultado de las simulaciones sobre los N escenarios se pueden calcular las 7 funciones objetivo para cada algoritmo, las que luego son comparadas para conformar un conjunto de soluciones no dominadas (Pareto óptimas). El algoritmo multi-objetivo SPEA realiza cada iteración sobre los N escenarios,

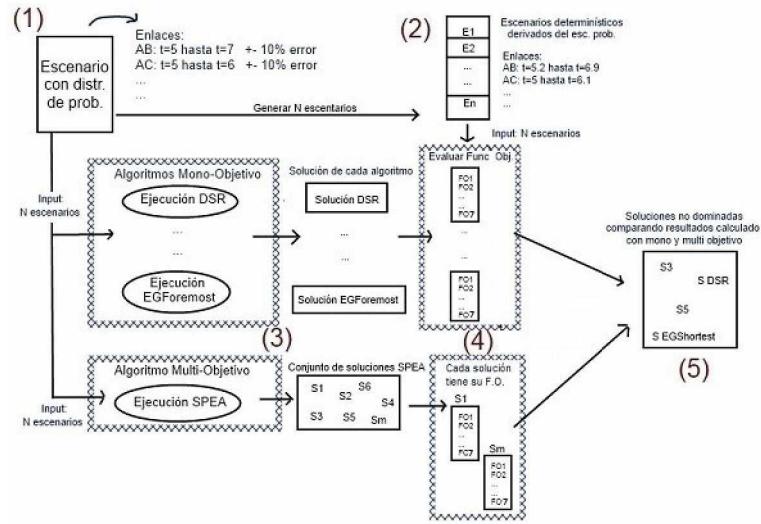


Figura 2. Esquema de simulación

para dar como resultado un conjunto de soluciones no dominadas en una sola corrida del algoritmo. Cada solución lógicamente tiene sus correspondientes funciones objetivo debidamente calculadas.

Por último, las soluciones de los algoritmos mono-objetivo y las soluciones del SPEA son comparadas entre sí para evaluar si forman parte del conjunto final de soluciones no dominadas (paso 5 de la Figura 2). De esta forma, se obtiene un conjunto de soluciones no dominadas que permite realizar un análisis final, el cual se explica en la siguiente sección.

4. Resultados experimentales

Las pruebas fueron realizadas de manera a ir comparando el algoritmo SPEA con los demás algoritmos mono-objetivo. Se hicieron pruebas del SPEA con 5, 20, 40, 100, 300 y 1000 iteraciones ya que el SPEA, al ser un algoritmo evolutivo, itera hasta que converja a un razonable conjunto de soluciones no dominadas.

Una vez terminada una simulación, se comparan los resultados de los algoritmos mono-objetivo para tener un conjunto de soluciones no dominadas. Debido a la gran cantidad de funciones objetivo que se calculan para cada solución, resulta muy difícil que una solución de un algoritmo en particular domine a todas las otras soluciones [15]. Entonces, para estos casos donde se tienen varias funciones objetivo, se puede utilizar la Relación de Preferencia introducida por Drechsler et al. en [7] quienes utilizaron un algoritmo previamente propuesto en [6] para resolver un problema de múltiples objetivos. Dicho algoritmo utiliza una relación de preferencia para comparar soluciones no dominadas entre sí [15]. La relación de preferencia define que, dadas dos soluciones x y x' , se dice que x es

Tabla 2. Conjunto de soluciones no dominadas encontradas al resolver el problema ejemplo de la Figura 1

Algoritmo	f_1	f_2	f_3	f_4	f_5	f_6	f_7
1.DSDV	1,677	8,935	10,387	0,819	25,613	28,758	4,389
3.DSDV	1,677	8,935	10,266	1,098	25,613	28,637	4,024
17.EGs	1,677	5,516	9,266	5,780	21,968	22,935	0,936
20.EGs	1,677	6	8,661	2,985	20,516	22,330	2,619
22.SPEA	1,935	11,258	11,379	0,043	24,516	26,330	1,565
32.SPEA	2,097	10,677	11,645	0,351	21,71	25,097	6,204
34.SPEA	2,129	12,387	13,113	0,292	21,806	27,008	11,869
37.SPEA	1,839	7,226	10,250	3,321	19,29	22,677	6,205

preferida a x' si y sólo si x tiene una mayor cantidad de objetivos mejores que x' .

Dentro de todas las simulaciones realizadas [2], a continuación sólo se presenta una simulación específica tomada como ejemplo para analizar los resultados obtenidos. En esta simulación en particular, el algoritmo SPEA sólo fue iterado 20 veces. Una vez que el simulador ejecutó todos los algoritmos antes mencionados sobre los cuatro escenarios, resulta un conjunto de soluciones no dominadas que se pueden apreciar en la Tabla 2. En dicha tabla cada columna f representa una función objetivo conforme fuera definido al final de la sección 3.2.

Tabla 3. Comparación del número de funciones objetivo en que una solución es mejor que la otra, considerando sólo las soluciones no dominadas.

	1.DSDV	3.DSDV	17.EGs	20.EGs	22.SPEA	32.SPEA	34.SPEA	37.SPEA	S
1.DSDV	**	1(N)	2(N)	2(N)	2(N)	3(N)	2(N)	3(N)	0
3.DSDV	3(P)	**	2(N)	2(N)	2(N)	3(N)	2(N)	3(N)	1
17.EGs	4(P)	4(P)	**	2(N)	4(P)	3(N)	3(N)	3(N)	3
20.EGs	4(P)	4(P)	4(P)	**	3(N)	4(P)	3(N)	5(P)	5
22.SPEA	5(P)	5(P)	3(N)	4(P)	**	4(P)	6(P)	3(N)	5
32.SPEA	4(P)	4(P)	4(P)	3(N)	3(N)	**	5(P)	3(N)	4
34.SPEA	5(P)	5(P)	4(P)	4(P)	1(N)	2(N)	**	3(N)	4
37.SPEA	4(P)	4(P)	4(P)	2(N)	4(P)	4(P)	4(P)	**	6

Luego, con las soluciones no dominadas de la Tabla 2 se crea una matriz de comparación de las funciones objetivo de cada solución (ver Tabla 3), donde se compara en cuantas funciones objetivo la solución de una fila es mejor que la solución de una columna. Así, al aplicar la Relación de Preferencia se asigna el valor (P) si la solución de la fila es preferida a la solución de la columna y el valor (N) si la solución de la columna es preferible a la solución de la fila. Para identificar cual sería la solución preferida entre todas las soluciones no dominadas, se puede por ejemplo contar las veces que una solución ha sido la

preferida sobre otras; es decir, los valores P para cada fila, lo que se muestra en la columna ‘S’ de la Tabla 3. Finalmente, se puede notar que la solución preferida bajo este criterio sería la solución ‘37.SPEA’ por tener el valor más alto (6) en la columna ‘S’.

Tabla 4. Relación de preferencia entre las soluciones no dominadas elegidas para realizar el grafo de la Figura 3.

	2.DSDV	3.DSDV	17.EGs	18.EGs	19.EGs	20.EGs	138.SPEA	234.SPEA
2.DSDV	**	(P)	(N)	(N)	(N)	(N)	(N)	(P)
3.DSDV	(N)	**	(N)	(N)	(N)	(N)	(N)	(N)
17.EGs	(P)	(P)	**	(P)	0	(P)	(N)	(N)
18.EGs	(P)	(P)	(N)	**	(N)	0	(N)	(N)
19.EGs	(P)	(P)	0	(P)	**	(P)	(N)	(N)
20.EGs	(P)	(P)	(N)	0	(N)	**	(N)	(P)
138.SPEA	(P)	(P)	(P)	(P)	(P)	(P)	**	0
234.SPEA	(N)	(P)	(P)	(P)	(P)	(N)	0	**

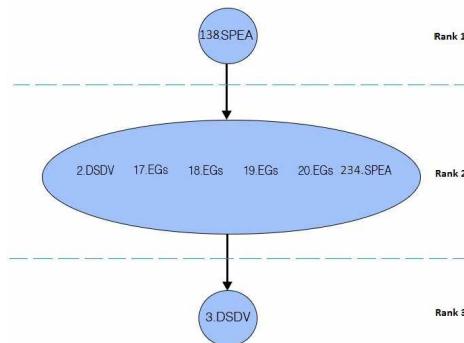


Figura 3. Grafo de relación de preferencia correspondiente a la Tabla 4, donde la solución no dominada ‘138.SPEA’ se encuentra en el primer nivel de la clasificación, quedando así demostrado que es la solución preferida.

Alternativamente, la selección de la solución preferida puede hacerse utilizando una técnica más formal como un *Grafo de Relación de Preferencia* [15]. Para ilustrar este concepto, la Figura 3 presenta el *Grafo de Relación de Preferencia* correspondiente a la Tabla 4, similar a la Tabla 3 que resulta de otra simulación que obtuvo 110 soluciones no dominadas que por un tema de espacio y didáctica solo se presenta parcialmente. En efecto, la Figura 3 muestra que la solución ‘138.SPEA’ es preferida a todas las demás soluciones por lo que se encuentra en el primer lugar de la clasificación (Rank 1); luego se observa que seis soluciones

colapsan en un solo nodo (Rank 2) debido a que todas ellas resultan preferidas a al menos una solución de este conjunto, existiendo al menos una solución del conjunto que es preferida a la solución considerada; es decir, al graficar las relaciones de preferencia entre estas seis soluciones se encontrará al menos un ciclo. Finalmente, se puede observar que la solución ‘3.DSDV’ en ningún caso es preferida a pesar de ser no dominada, por lo que queda última en la clasificación (Rank 3). Cabe destacar que la elección de una solución calculada por el algoritmo SPEA como preferida en el ejemplo no es una simple casualidad dado que consistentemente la soluciones preferidas siempre fueron calculadas por el SPEA corriendo un número suficientemente grande de iteraciones.

5. Conclusión

La primera contribución de este trabajo es el simulador implementado que permite incluir al tiempo continuo en las simulaciones y que incluso cuenta con la posibilidad de utilizar diversos escenarios para modelar la incertidumbre.

Experimentalmente se comprobó que algunos algoritmos como el AODV, DSR y EG Foremost no son buenos para resolver el problema en cuestión, dado que no han sido capaces de encontrar soluciones no dominadas. En contrapartida, se demostró experimentalmente que algunos algoritmos como el DSDV y el EG Shortest son razonablemente buenos (aún siendo algoritmos mono-objetivo) dado que lograron encontrar soluciones no dominadas, aunque estas soluciones no lleguen a ser las soluciones preferidas. Por lo tanto, si no se tiene acceso a suficientes recursos computacionales como para utilizar un algoritmo multi-objetivo, estos dos algoritmos serían los recomendables.

Cabe destacar que por primera vez se propone la utilización de la Relación de Preferencia [15] para comparar el gran número de soluciones no dominadas encontradas en el problema estudiado. Así, se demostró que mediante esta Relación de Preferencia se logró identificar la solución no dominada preferida del conjunto de soluciones Pareto óptimas, incluso al considerar un número considerable de funciones objetivo, siete en este caso.

Como principal aporte de este trabajo, se aportan datos experimentales que concluyentemente indican que un algoritmo de ruteo que considere un contexto multi-objetivo (en este caso el SPEA) es claramente superior a los algoritmos del estado del arte, típicamente mono-objetivo, para redes periódicas de tiempo continuo en contextos de incertidumbre. Como trabajo futuro los autores se encuentran experimentando con otros algoritmos multi-objetivos, diversas redes periódicas y variados conjuntos de escenarios.

Referencias

1. Ash, G. R. (1995). Dynamic network evolution, with examples from AT&T's evolving dynamic network. *Communications Magazine, IEEE*, 33(7), 26-39.
2. Bosch, D. (2015) Ingeniería de tráfico en redes dinámicas periódicas en tiempo continuo utilizando el algoritmo SPEA. Tesis de grado. Universidad Católica "Nuestra Señora de la Asunción".

3. Bosch, D., Barán, B., & Yael, U. (2012) Ingeniería de tráfico en redes dinámicas periódicas en tiempo continuo con eras de duración estocástica. *IEEE ARANDUCON*.
4. Carrano, R. C., Blestas, M., & Magalhães, L. C. (2007). Mesh networks for digital inclusion-testing OLPC's XO mesh implementation. In *8o Forum Internacional de Software Livre, 2007, Porto Alegre*.
5. Coello, C. A., Lamort, G.B., & Van Veldhuizen, D.A. (2007). Evolutionary Algorithms for Solving Multi-Objective Problems. 2da.Edición, Editorial Springer.
6. Drechsler, N., Drechsler, R., & Becker, B. (1999). Multi-objected optimisation in evolutionary algorithms using satisfyability classes. In: *Reusch, B. (ed.) International Conference on Computational Intelligence, Theory and Applications, 6th Fuzzy Days. Lecture notes in Computer Science*, vol. 1625, pp 108-117. Springer, Dortmund.
7. Drechsler, N., Drechsler, R., & Becker, B. (2001). Multi-objected optimisation based on relation favour. In: *First International Conference on Evolutionary Multi-Criterion Optimization. Lecture Notes in Computer Science*, vol. 1993, pp. 154-166. Springer, Berlin.
8. Jarry, A., & Lotker, Z. (2004). Connectivity in evolving graph with geometric properties. In *Proceedings of the 2004 joint workshop on Foundations of mobile computing* (pp. 24-30). ACM.
9. Johnson, D. B., & Maltz, D. A. (1996). Dynamic source routing in ad hoc wireless networks. In *Mobile computing* (pp. 153-181). Springer US.
10. Monteiro, J. (2007). Uso de Grafos Evolutivos no Roteamento em Redes Dinâmicas: algoritmos, fluxos e limites. Tesis de maestría. Instituto de Matemática y Estadística, Universidad de São Paulo. 2007.
11. Monteiro, J., Goldman, A., & Ferreira, A. (2007, June). Using Evolving Graphs Foremost Journey to Evaluate Ad-Hoc Routing Protocols. In *Proceedings of 25th Brazilian Symposium on Computer Networks (SBRC'07)*, Blem, Brazil.
12. Pareto, V. (1896). *Cours D'Economie Politique*,Volume 1. Lausanne: F. Rouge.
13. Perkins, C. E., & Bhagwat, P. (1994). Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *ACM SIGCOMM Computer Communication Review* (Vol. 24, No. 4, pp. 234-244). ACM.
14. Perkins, C., Belding-Royer, E., & Das, S. (2003). Ad hoc On-Demand Distance Vector (AODV) Routing. IETF. RFC 3561.
15. von Lücken, C., Barán, B., & Brizuela, C. (2014). A survey on multi-objective evolutionary algorithms for many-objective problems. *Computational Optimization and Applications*, 1-50.
16. Yael, U. (2009). Ingeniería de Tráfico en Redes Dinámicas Periódicas Utilizando el Algoritmo SPEA. Tesis de grado, Universidad Católica “Ntra. Sra. de la Asunción”.
17. Yael, U., & Barán, B. (2010) Routing in Periodic Dynamic Networks using a Multi-Objective Evolutionary Algorithm. *ALIO-INFORMS Joint International Meeting*.
18. Zitzler, E., & Thiele, L. (1999). Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *Evolutionary Computation, IEEE on*, 3(4), 257-271.