TABLA RESUMEN

PROFESOR	ARTICULO
BRUNO BARBOZA OROZCO	
ROBERTO SUAREZ ZINZUN	
RICARDO GARCÍA CRUZ	
ALBERTO DÍAZ VÁZQUEZ	2
JESÚS GARCÍA DÍAZ	2

ARTÍCULOS

Tipo	Artículo de difusión y divulgación
Autor(es)	MIDE. José Mario Miranda Ramírez, MIDE. José Luis Zaráte Lucatero. Phd Jaime Aguilar García, Dr. Alberto Díaz Vázquez
Título	Diagnóstico nutricional de producción en papaya maradol roja con aplicación móvil \
Estado actual	2
País	MÉXICO
Nombre de la Revista	Academia Journals
Editorial	Academia Journals
Volumen	1
ISSN	2380-509X
Año	2016
Propósito	Difusión
Para considerar en el currículum de cuerpo académico	Si
Miembros	0
LGACs	0

Diagnóstico nutricional de producción en papaya maradol roja con aplicación móvil “Nutri app papayita”

José Mario Miranda-Ramírez[1], Jose Luis Zárate-Lucatero[2], Jaime Aguilar-García[3], Alberto Díaz Vázquez[4]

[1]Instituto Tecnológico Superior de Apatzingán, Profesor-Investigador; Docente de Ingeniería en Innovación Agrícola Sustentable, Ingeniería en Gestión Empresarial, Doctorante en el Instituto de Innovación y Desarrollo Tecnológico. jose@itsa.edu.mx
[2]Instituto de Innovación y Desarrollo Tecnológico, Doctorante en el Instituto de Innovación y Desarrollo Tecnológico. slickand@gmail.com
[3]Instituto de Innovación y Desarrollo Tecnológico Profesor-Investigador; correo@jaimeaguilar.org
[4]Instituto Tecnológico de Estudios Superiores de Zamora, Jefe del Área de investigación, docente de la carrera de Ingeniería en Gestión Empresarial del ITESZ; alberto_diaz_vazquez@hotmail.com

México

ABSTRACT

This article has the general objective to build an App for cell phone called "papayita App" to allow farmers papaya to implement a diagnostic tool that serves to improve agricultural production processes. To make this work we were consulted two papaya farmers in the village of La Ruana municipality of Buenavista, Michoacán, México. The App is based on four major nutritional problems that arose during the process of production of papaya in this region: 1) Intoxication boron, 2) Fito toxicity nitrogen, 3) deficiency of potassium, 4) Calcium deficiency .In the analysis of technological maturity we were evaluated: 1) Methodology 2) Culture, 3) Innovation, 4) Customer Centric and 5) Technology. As part of the methodology the MIT App Inventor 2 Beta program was used is an application development environment for Android devices. The results obtained in the mobile App was generated for any type of cell with an Android operating system.

Keywords: Agricultural Producers Papaya, App, Technology Implementation.

RESUMEN

Este artículo tiene como objetivo general construir una App para el teléfono celular denominada “Papayita App” que permita a los productores agrícolas de papaya poder implementar una herramienta diagnóstico que sirva para mejorar sus procesos de producción agrícola. La App se sustenta en cuatro de los principales problemas de nutrición que se presentaron durante el proceso de producción de papaya en esta región: 1) Intoxicación por boro, 2) Fito toxicidad por nitrógeno, 3) Deficiencia de potasio, 4) Deficiencia de calcio. En el análisis de la madurez tecnológica se evaluó: 1) Metodología, 2) Cultura, 3) Innovación, 4) Customer Centric y 5) Tecnología. Como parte de la metodología se utilizó el programa MIT App Inventor 2 Beta es un entorno de desarrollo de aplicaciones para dispositivos Android. En los resultados obtenidos se generó la App móvil para cualquier tipo de celular que opere con un sistema Android.

Palabras clave: Productores Agrícolas de Papaya, App, Implementación de Tecnologías.

REVISIÓN DE LITERATURA

Para poder comprender cuál es el uso de las Tecnologías de la Información y Comunicación (TIC) en el proceso de la producción agrícola se puede definir como:

“El conjunto de herramientas, soportes y canales desarrollados y sustentados por las tecnologías (telecomunicaciones, informática, programas, computadores e internet) que permiten la adquisición, producción, almacenamiento, tratamiento, comunicación, registro y presentación de informaciones, en forma de voz, imágenes y datos, contenidos en señales de naturaleza acústica, óptica o electromagnética a fin de mejorar la calidad de vida de las personas” (Ávila, 2012, 222).

De acuerdo con investigaciones de los autores Botsiou y Dagdilelis (2013) en países como Grecia la infiltración de las nuevas tecnologías en el sector agrícola es un hecho, y constituyen una herramienta de información y de gestión de las empresas agrícolas. Mientras que en países como China el sector agrícola durante las últimas tres décadas, ha sido transformado de lo tradicional a la práctica moderna a través de la implementación efectiva de las TIC (Zhang, Wang, y Duan 2015; 17). Las TIC juegan un papel fundamental en las empresas agropecuarias en América Latina. Sus principales usos son para la toma de decisiones productivas (Zapata, 2012, 6).

La innovación tecnológica es uno de los factores esenciales para aumentar la productividad necesaria para alcanzar un crecimiento económico sostenido (Martínez y Porcelli, 2015, 5).

Los autores Ekuobase y Olutayo, (2016) describieron que en el siglo XXI, las TIC se convierten en un activo estratégico de las organizaciones para ofrecer servicios innovadores y conseguir una ventaja competitiva sostenible y la importancia de la innovación basada en las TIC en la mejora de la productividad y la competitividad que es enorme.

La agricultura es una actividad importante para el crecimiento económico de los países en desarrollo (González, Rendón, Sangerman, Cruz y Díaz, 2015, 175). A nivel mundial ha sido influida por el nuevo paradigma tecnológico; por lo que se ha favorecido con los enormes avances de la introducción de los recursos informáticos (Pérez, Martínez, López y Rendón, 2016, 11). La ciencia agronómica y la agricultura ha ganado recientemente popularidad como un medio de gestión de la producción de cultivos (Fujimoto, Satow y Kishimoto, 2016, 1). El desarrollo de las TIC en la agricultura debe estar enfocado en el pequeño productor (Espinel, 2012, 10). Sin embargo, el sector de la agricultura se ha vuelto cada vez más dependiente de la información, lo que requiere una amplia gama de información científica y técnica para la toma de decisiones eficaz (Alí y Kumar, 2016, 149).

Para los autores Pérez Martínez, López y Rendón, (2016) la adopción de innovaciones se relaciona con el uso de tecnología que permite crear un potencial productivo y mejora de la competitividad. En la agricultura, la adopción de la innovación se ha intentado contabilizar de diversas formas, sin embargo, en la mayoría de los casos se ha realizado a partir de conteos simples de innovaciones realizadas.

Las TIC son una herramienta para el acceso y la organización del conocimiento disponible para los agricultores (Jiménez, Rendón, Toledo y Aranda, 2016, 3064) y su uso puede ser un apoyo a los agricultores para aumentar su producción, contribuyen a la agricultura "inteligente", más eficiente y sostenible (Salampasis y

Theodoridis, 2013; 2) además de la disminución en los tiempos para toma de decisiones gracias a alertas climáticas y controles de plagas (Zapata, 2012, 6).

Los servicios de información para los agricultores a nivel nacional y regional son un nuevo y prometedor campo de la investigación y su aplicación en el campo emergente de la agricultura electrónica es para apoyar a los agricultores y las comunidades agrícolas a mejorar la productividad agrícola y la sostenibilidad (Zhang et al., 2015, 2).

Bajo el contexto de la producción agrícola de papaya es necesario conocer que durante el año 2014 en el municipio de Buenavista, se sembraron 132 ha y produjeron 2,426.73 t (SAGARPA, 2016).

METODOLOGÍA

Este estudio se realiza en cinco etapas principales que se describen: **1) propuesta de la investigación** donde se construyen los objetivos y preguntas de investigación, **2) diseño de la investigación** dónde se determinan las variables directas e indirectas y se determinó su naturaleza. Se centra en una investigación a) exploratoria, este tipo de estudios se realiza cuando el objetivo es examinar un tema o problema de investigación poco estudiado (Hernández, Fernández y Bapista, 2014, 91) b) no experimental de tipo c) cuantitativa su principal objetivo es examinar, medir y evaluar (Argibay, 2009, 19) d) campo e) transeccional debido a que recolecta datos en un solo momento, en un tiempo único (Hernández et al., 2014, 154) y f) descriptiva por qué parte de una investigación exploratoria (Cazau, 2006, 26), se refiere a la descripción de algún objeto, sujeto, fenómeno, acepta como perfectamente válida y original la descripción de alguna variación o modificación de algo ya descrito (Salinas, 2010, 18).

3) Evaluación de la aplicación de las TIC en el proceso de producción agrícola de papaya maradol roja se hizo un diagnóstico del análisis de nivel de la madurez digital de dos productores de papaya en la localidad de La Ruana, municipio de Buenavista Tomatlán, Michoacán, utilizando como herramienta la página web https://www.paradigmadigital.com/dtma/ Digital Transformation Maturity

Assessment (DTMA) Evaluación de la madurez de la transformación digital, una medición amplia y objetiva del nivel de madurez digital que ésta compuesta por cinco vectores 1) Metodología, 2) Cultura, 3) Innovación, 4) Customer Centric y 5) Tecnología. Con el objetivo de conocer el uso de las TIC dentro del proceso de cultivo de papaya maradol dentro de esta región.

El análisis de los cinco vectores visualiza que es bajo, los dos productores agrícolas tienen sus habilidades digitales muy poco desarrolladas, carecen de cultura sobre el uso de las TIC y de conocimiento explicito, tienen como visión producir, pierden la relación con el cliente y conocen de manera general la adopción de las innovaciones tecnológicas en su cultivo, pero no se implementan (Figura 1).

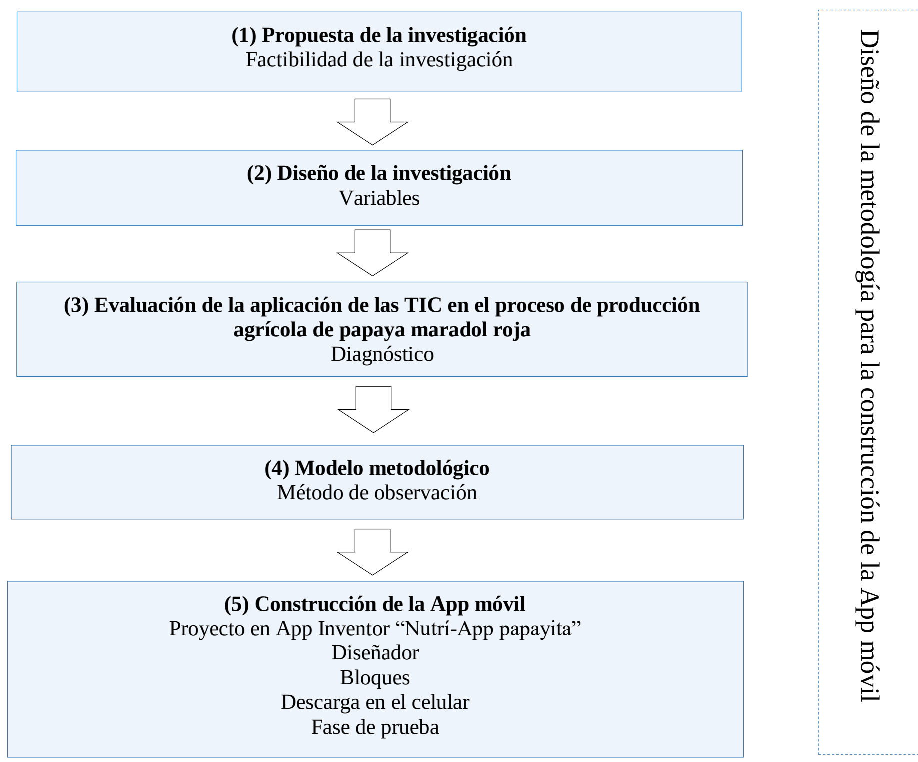
Figura 1. Gráfico de la aplicación de las TIC en el proceso de producción agrícola de papaya maradol roja

La propuesta de mejora del nivel de madurez tecnológica consiste en implementar la App móvil desarrollada, mediante las variables detectadas como problemas en el proceso de producción agrícola y la evaluación de la madurez tecnológica. Por consiguiente esta tecnología digital es estratégica, contribuye de forma directa en el proceso de producción agrícola de papaya.

La etapa **4) Modelo metodológico** analizó cuáles son las variables de tipo fitosanitarias que influyen de manera directa en el proceso de producción de papaya. De acuerdo a un diagnóstico obtenido por el método de la observación llevando un registro sistémico de recolección de datos sobre bitácoras de prevención y control de las plagas y enfermedades con mayor incidencia del cultivo en dos huertos de productores agrícolas de La Ruana, municipio de Buenavista Tomatlán, Michoacán. Las variables registradas son: *1) Intoxicación por boro, 2) Fito toxicidad por nitrógeno, 3) Deficiencia de potasio, 4) Deficiencia de calcio*.

5) Construcción de la App móvil en esta etapa a partir de estas cinco variables se desarrolla la App móvil de diagnóstico que se utilizó en el diseño, y su construcción. Para el diseño de la aplicación móvil se utilizó el programa MIT App Inventor 2 Beta es un entorno de desarrollo de aplicaciones para dispositivos Android en el cual se trabajó en la construcción del proyecto desde el Diseñador y los Bloques para enviarse a un celular y hacer las pruebas piloto y final correspondientes de su funcionamiento.

Desarrollo de la metodología para la App móvil de diagnóstico en papaya maradol roja (Figura 2).



Fuente: Elaboración propia 2016

Figura 2. Descripción de la metodología para la construcción de la App móvil.

RESULTADOS

Construcción de la App móvil. La construcción de aplicación móvil se realizó en el programa MIT App Inventor 2 Beta, para poder construir la App de Android se utilizaron dos fases: Diseñador y Bloques. El proyecto se denominó "Nutri App

Papayita” En la primera fase la pantalla inicial que se utilizó es el Diseñador. A partir de aquí se seleccionaron del menú Paleta los objetos que se necesitaron para colocarlos en el Visor. En la ventana de Componentes aparecen de forma vertical todos los objetos que se seleccionaron en la ventana de Visor y en la columna de Propiedades se seleccionaron las características del menú que se encuentra en él Visor. En esta fase se colocaron todos los objetos en el Visor para que en la segunda fase asignar todas las funciones. En la segunda fase se utilizó los Bloques esta interface está compuesta por una columna de Bloques y una ventana con el Visor. En los Bloques se encuentran los integrados y los objetos que se agregaron en el Diseñador, estos a su vez se agregaron a la ventana de Visor, se trabajó con los bloques por medio de una Condición, Lógica y Texto (Figura 3).
Se descargó la App en el celular para la primera prueba, se realizaron varias pruebas posteriores para verificar el funcionamiento correcto en un equipo móvil Samsung J7 (Figura 4).

Figura 3. Construcción de la aplicación móvil Papayita App por medio de bloques.

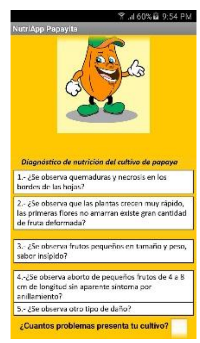


Figura 4. Aplicación móvil “Nutri App papayita” funcionando en un equipo móvil Samsung J 7.

Esta aplicación implica el uso de cualquier dispositivo móvil avanzado con sistema operativo Android a través del usuario y contactar al asesor técnico especialista en papaya para dar el apoyo técnico requerido en el momento adecuado y en cualquier etapa fenológica que se encuentre el cultivo. Para poder operar la aplicación se requiere descargarla e instalar en un dispositivo móvil avanzado. Una vez instalada, en la pantalla principal aparecen los cuatro problemas de nutrición en forma de preguntas, de los cuales el productor agrícola debe contestar de acuerdo a lo que observó en su cultivo, escribir un número en el casillero de color blanco de acuerdo al número de problemas que se presentan. Finalmente oprimir en el botón Diagnosticar Mi Cultivo Ahora!... en seguida aparece la pantalla con el diagnóstico de sugerencias.

CONCLUSIONES

Esta App móvil permite al productor agrícola informarse a tiempo si su cultivo necesita de apoyo técnico de nutrición sin necesidad de estar conectado a una red de internet. Con el uso de las TIC el productor cuenta con un elemento clave de información así como una herramienta de transferencia de conocimiento que se reflejara sin duda en su proceso de producción agrícola de papaya. En cambio el asesor técnico también puede utilizar esta herramienta para llegar a atender las necesidades de servicio que le demanden. Es recomendable trazar una estrategia dónde se involucre esta aplicación de este medio de comunicación efectiva entre las mismas fuentes de información de los productores agrícolas, redes sociales y los mismos técnicos que prestan el servicio.

Los equipos celulares se utilizan por los dos productores agrícolas para la comunicación e interacción con otros productores, pero aunado a esto se debe complementar con aplicaciones como la desarrollada si realmente el productor agrícola quiere ser más eficiente en su producción y más competitivo. Por otra parte el nivel de madurez tecnológica que mostraron es bajo por consiguiente se debe alfabetizar digitalmente al productor agrícola, aunque puede ser que un factor limitante sea la falta interés en el uso de las TIC.

REFERENCIAS BIBLIOGRÁFICAS

Ávila, D. (2012). Hacia una reflexión histórica de las TIC. ***Hallazgos,*** **(10),** 213-223 p.

Alí, J., Kumar, S. (2011). Tecnologías de la información y comunicación (TIC), los agricultores y la toma de decisiones a través de la cadena de suministro agrícola. ***International Journal of Information Managemen,*** **31** (2), 149–159.

Argibay, J. (2009). Muestra en investigación cuantitativa. ***Subjetividad y Procesos Cognitivos,*** **13** (1), 13-29.

Ávila, W. (2012). Hacia una reflexión histórica de las TIC. ***Hallazgos, 10*** (19), 213-223.

Botsiou, M., Dagdilelis, V. (2013). Aspectos de la incorporación de las TIC en las empresas agrícolas griegas: El caso de una prefectura. ***Procedia Tecnology,***

8 (2013), 387-396. Recuperado de: http://www.sciencedirect.com/science/article/pii/S2212017313001138

Cazau, P. (2006). Introducción a la investigación en ciencias sociales. Tercera edición. Argentina.

Espinel, B. (2012), “El desarrollo de las TIC debe estar enfocado en el pequeño productor”. ***In: TIC y agricultura, 18*** (2012), 1-12.

Ekuobase, G. Olutayo, V. (2016). Estudio de madurez (TIC) Tecnología de Información y Comunicación y el valor: La relación, ***Egyptian Informatics Journal, XXX*** (2016), 1-11. Recuperado de: http://www.sciencedirect.com/science/article/pii/S1110866516000025

Fujimoto, A., Satow, T., Kishimoto, T. (2016). La simulación de distribución de rociado con el rociador auge considerando efecto del viento para el análisis de la computación en la nube agrícola. ***Engineering in Agriculture, Environment and Food.*** **XXX** (2016),1-6 p. Recuperado de: http://dx.doi.org/10.1016/j.eaef.2016.04.001

González., A., Rendón, R., Sangerman, M., Cruz, G., Díaz, J. (2015). Extensionismo agrícola en el uso de tecnologías de la información y comunicación (TIC) en Chiapas y Oaxaca. ***Revista Mexicana de Ciencias Hortícolas*****, 6**(1), 175-186.

Hernández, R., Fernández, C., Bapista, P. (2014). ***Metodología de la investigación.*** (Sexta edición), México, Editorial Mc Graw Hill.

Jiménez, S., Rendón, R., Toledo, U., Aranda, G. (2016). Las tecnologías de la información y comunicación como fuente de conocimientos en el sector rural. [Ejemplar especial]. ***Revista Mexicana de Ciencias Agrícolas,*** **15** (2016), 3063-3064.

Martínez, M., Porcelli, M. (2015). Implicancias de las tecnologías informáticas en el ambiente y nuevas tendencias en el desarrollo de la informática verde como aporte al desarrollo sustentable. ***Actualidad Jurídica Ambiental.*** **50** (2015), 1-28.

Pérez, O., Martínez, H., López, J., Rendón, R. (2016). Estimación de la adopción de innovaciones en la agricultura. [Ejemplar especial]. ***Revista Mexicana de Ciencias Agrícolas.*** **15** (2016), 2909-2926.

Salampasis, M. y Theodoridis, A., (2013). Tecnología de Información y Comunicación en el Desarrollo Agrícola. ***Procedia Technology 8*** (2013) 1-3. Recuperado de: http://www.sciencedirect.com/science/article/pii/S2212017313000637

Salinas, J. (2010). ***Metodología de la investigación científica.*** Universidad de los Andes. Venezuela.

Secretaria de Agricultura, Ganadería Desarrollo Rural, Pesca y Alimentación SAGARPA., (2016). ***Servicio de Información Agroalimentaria y Pesquera (SIAP).*** México. Recuperado de: http://www.siap.gob.mx/cierre-de-la-produccion-agricola-por-estado/

Zapata, L. (2012). TIC en el sector rural y agroalimentario, el caso uruguayo. Taller TIC, Desarrollo y Políticas Públicas. Recuperado de: http://www.academia.edu/1368223/TIC_en_el_sector_rural_y_agroalimentario._El_caso_uruguayo

Zhang Y., Wang L., Duan Y. (2015). Difusión de información agrícola utilizando las TIC: Una revisión y análisis de los modelos de difusión de información en

China. ***Information Processing in Agriculture 3* (2015)**, 17-29. Recuperado de: http://dx.doi.org/10.1016/j.inpa.2015.11.002

Tipo	Artículo de difusión y divulgación
Autor(es)	Alberto díaz Vázquez, Leonel Chávez Contreras, José Jorge Rodríguez Hernández, Rubén Molina Martínez
Título	Estrategia de Tecnología Educativa, empleando un Modelo de Diseño Instruccional
Estado actual	1
País	MÉXICO
Nombre de la Revista	Alta Tecnología y la Sociedad
Editorial	Academia Journals
Volumen	7
ISSN	1940-2171
Año	2015
Propósito	Asimilación de tecnología
Para considerar en el currículum de cuerpo académico	Si
Miembros	2
LGACs	1

Estrategia de Tecnología Educativa Empleando un Modelo de Diseño Instruccional.

Dr. Alberto Díaz Vázquez[1], Dr. Rubén Molina Martínez[2],
Dr. Leonel Chávez Contreras[3] e Ing. José Jorge Rodríguez Hernández[4]

[1] Alberto Díaz Vázquez es jefe de posgrado e investigación del Instituto Tecnológico de Estudios Superiores de Zamora, Michoacán. México. alberto_diaz_vazquez@hotmail.com (**autor corresponsal)**
[2] Rúben Molina Martínez. Profesor investigador de la Universidad Michoacán de San Nicolás de Hidalgo. Michoacán. México ruben.molinam@gmail.com.
[3] Leonel Chávez Contreras profesor de tiempo completo de la academia económico administrativas del Instituto Tecnológico de Estudios Superiores de Zamora, Michoacán. México. chavezleonel63@yahoo.com.mx
[4] José Jorge Rodríguez Hernández profesor de tiempo completo de la academia de Ing en Electrónica del Instituto Tecnológico de Estudios Superiores de Zamora, Michoacán. México. tecnologico.zamora@hotmail.com

Resumen— **La presente investigación consiste en un análisis de las tecnologías de información y comunicación para ofertar educación continua a través de e-learning, el cual está orientado a un modelo de diseño instruccional para cursos en línea que el docente puede implementar utilizando la plataforma moodle. La propuesta deja claro como diseñar un curso en línea desde la introducción, planeación, estructura, hasta la bibliografía y recursos del mismo. La investigación sirve como un medio para que el docente desempeñe actividades relacionadas con el ámbito educativo. Por último se propone aplicar el diseño instruccional, así como promover y apoyar investigaciones de este tipo, utilizando las tecnologías de información y comunicación.**

***Palabras clave*— diseño instruccional, e-learning, tics, proporcione cuatro o cinco palabras que servirán para identificar el tema de su ponencia, separadas por comas.**

Introducción

En la presente investigación se detectó un problema, el cual está relacionado con el análisis de las tecnologías de información y comunicación para ofertar educación a distancia, se plantea un escenario social en el que las tic tienen un protagonismo marcado en todos los ámbitos, principalmente la educación. El amplio desarrollo del mundo tecnológico y de las comunicaciones hace posible la aparición de nuevas formas educativas, entre las que se encuentra el e-learning en educación continua.

Esta investigación explora en profundidad este contexto del e-learning para llegar al objetivo de la investigación. Se repasan algunas teorías sobre temas tales como: teorías educativas, que permiten conocer el proceso de aprendizaje y como fue evolucionando la educación; teorías sobre la educación a distancia, la cual trata de distintas formas de educación utilizando medios electrónicos, y sobre plataformas virtuales. Se realiza una trabajo de campo en donde se obtiene un universo de estudio, la determinación de la muestra, el diseño de un instrumento, así como su viabilidad y factibilidad, se hace el análisis e interpretación de la información obtenida del trabajo de campo, se presenta las conclusiones del trabajo de campo, así como las conclusiones generales y recomendaciones, se presenta la bibliografía consultada y anexos.

Descripción del Método

La investigación es de tipo Cuasi experimental, la cual consiste en probar si hay relaciones causales sin tener pleno control. En esta el investigador no tiene control total sobre el criterio empleado para asignar participantes a grupos. La descriptiva en histórica proporciona una imagen de los sucesos que están ocurriendo o que han ocurrido en el pasado.

Para cumplir con los objetivos de la investigación se determinó el siguiente método:

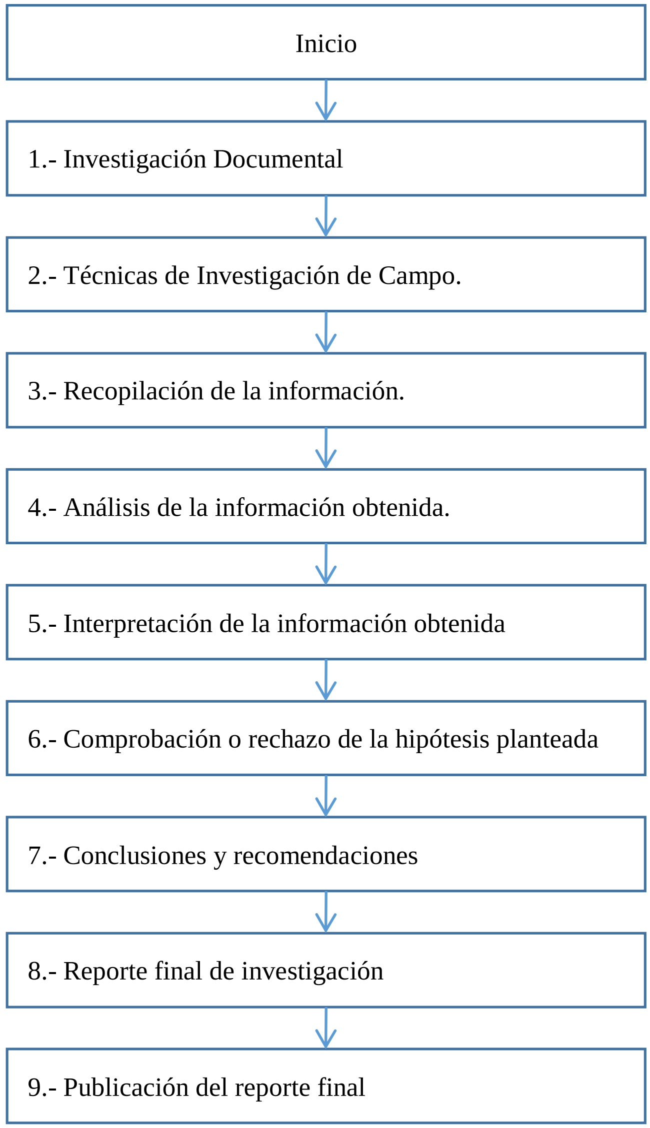


Figura 1.- Secuencia Metodológica

Modelo de e-Learning.

Con la intención de facilitar la creación efectiva de ambientes *e*-learning para diversos estudiantes, Khan (2005) desarrolló un modelo de *e*-learning. Este modelo señala los factores a considerar durante el proceso de diseño de un programa *learning*. Muchos de ellos son interdependientes, a la vez que están sistemáticamente interrelacionados.

La educación virtual se define como una institución basada en la educación formal de aprendizaje donde el grupo se separa y los sistemas de telecomunicaciones interactivos se utilizan para conectar los estudiantes, los recursos y los instructores (Simonson, 2003), es la combinación de dos factores tiempo y lugar.
En ocasiones “El título de Maestro no debe darse sino al que sabe enseñar, esto es, al que enseña a aprender, no al que manda a aprender, o indica lo que se ha de aprender, ni al que aconseja que se aprenda. El Maestro que sabe dar las primeras instrucciones, sigue enseñando virtualmente todo lo que se aprende después, porque enseño a aprender.”

Tanto la enseñanza presencial como la enseñanza a distancia o en línea requieren planificación y organización. En particular, la enseñanza a distancia, bien sea de manera sincrónica o asincrónica, requiere un mayor énfasis en la fase inicial de planificación es decir, el diseño instruccional.

Las tecnologías de la información y la comunicación (TIC) pueden contribuir al acceso universal a la educación, la igualdad en la instrucción, el ejercicio de la enseñanza y el aprendizaje de calidad y el desarrollo profesional de los docentes, así como a la gestión dirección y administración más eficientes del sistema educativo.

Modelos de diseño instruccional

De acuerdo con Siemens (2002), un modelo es una representación de hechos reales y, como tal, debe ser utilizada sólo en la medida en que es manejable para la situación o tarea en particular. Los modelos de diseño instruccional que se conocen, surgieron a partir de adaptaciones a los anteriores, de la disponibilidad y acceso a la tecnología y de las propuestas de varios teóricos de utilizarla para los procesos de enseñanza aprendizaje, y facilitar el desarrollo de la instrucción. Estos modelos tienen el objetivo de orientar hacia el diseño y presentación de contenidos educativos y sus correspondientes actividades de aprendizaje y evaluación. Algunos modelos de diseño instruccional son orientados a la tecnología educativa y desarrollo de procesos genéricos, estos modelos tienen orientación conductista. Otros están orientados a los conceptos de diseño de aprendizaje o teorías pedagógicas, tienen orientación hacia el constructivismo y el cognoscitivismo.

Los modelos de diseño instruccional además de tener un diseño instruccional, o sea trabajar en la parte del diseño, debe considerar la planeación de los cursos que se desean implementar. Se considera que el diseño instruccional comprende desde el análisis hasta la puesta en marcha del recurso, incluyendo la modalidad educativa como agente diferenciador.

Todo ello requiere una guía puntual para gestionar el conocimiento, intervenir en los procesos de enseñanza aprendizaje evaluación de la educación a distancia y provocar un aprendizaje significativo. Es precisamente en la parte de la gestión del conocimiento a distancia que surgen las modalidades sobre el aprendizaje que Holmes y Gardner (2006), Bonk y Graham (2005), Moreno (2012), Gimeno (2004) y Santamaría (2009), respectivamente, describen como:

E- learning: aprendizaje a través de medios electrónicos.
B-learning (blended learning): modalidad mixta o semipresencial de estudios que incluye tanto al e-learning como formación presencial.
M-learning: aprendizaje a través de tecnología móvil.
U-learning: aprendizaje por medio de la tecnología digital dominada por la computación ubicua, conocida también como inteligencia ambiental.
S-learning: aprendizaje por medio de las redes sociales y comunidades virtuales.

Comentarios Finales

Resultados

Aplicación del instrumento utilizando la escala tipo Likert y la forma de obtener las puntuaciones en esta investigación

La forma de obtener las puntuaciones en la escala tipo Likert fue sumando los valores alcanzados en cada pregunta. Se utilizó la forma básica de aplicar la escala tipo Likert, se le entregó la encuesta a la persona que lo respondió, creando su opinión respecto a cada categoría en el espacio que mejor describe su juicio.
Encuesta
Se llevaron a cabo las encuestas, para lo cual se les solicito a un grupo de docentes que trabajan en el ITESZ den respuesta de acuerdo a su criterio.
La aplicación del instrumento tuvo un tiempo de duración para responderlo de 15 a 20 minutos; debido a que tiene que revisar información técnica para contestar adecuadamente.

Recolección de datos

Se llevó a cabo la recolección de datos para saber la viabilidad del instrumento. Una vez que se seleccionó el diseño de investigación apropiado y la muestra adecuada de acuerdo con el problema de estudio e hipótesis, la siguiente etapa consiste en recolectar los datos pertinentes sobre las variables y dimensiones involucradas, esto es seleccionar el instrumento de medición de los disponibles en el estudio, aplicar ese instrumento de medición y preparar las mediciones obtenidas.

Confiabilidad del instrumento

La confiabilidad del instrumento de medición que se ha aplicado, ha sido a través del método del Alfa de Cronbach, el cual requiere una sola administración del instrumento de medición y produce valores que oscilan entre 1 y 0. Simplemente se aplica la medición y se calcula el coeficiente, si el valor es superior a 0.8 son considerados aceptables, para esta investigación el resultado del alfa de Cronbach fue de 0.96. La fórmula utilizada es la siguiente:

$$\propto = \frac{k}{k-1}\left[1 - \frac{\sum vi}{vt}\right]$$

Donde:
$\propto$=alfa de Cronbach
K= número de ítems
Vi= varianza de cada ítem
Vt= varianza del total
Los datos obtenidos fueron calculados en Excel 2010 quedando de la siguiente forma:

Numerador =	54,1034
Denominador =	836,1204
n =	29
n - 1=	28
k =	1

El resultado al aplicar la fórmula es

0,96

Con este resultado se demuestra la confiabilidad del instrumento de acuerdo al coeficiente de determinación aplicado en la investigación.

Modelo de Diseño instruccional. (Modelo D-I-ITESZ).

Para ofrecer educación continua a través de e-learning, es necesario contar con un diseño instruccional bien estructurado. Mientras mejor estructurado se encuentre el diseño instruccional de un curso, mejor será la eficiencia en la educación. Por tal motivo se debe capacitar a los docentes para poder implementar un diseño instruccional y de esta forma subir sus cursos.

En un diseño instruccional de e-learning existe una interacción paso a paso en los materiales de estudio. Así mismo, existe una evaluación momento a momento acerca del impacto de los materiales educacionales sobre los procesos de construcción de conocimientos por parte del estudiante en línea.

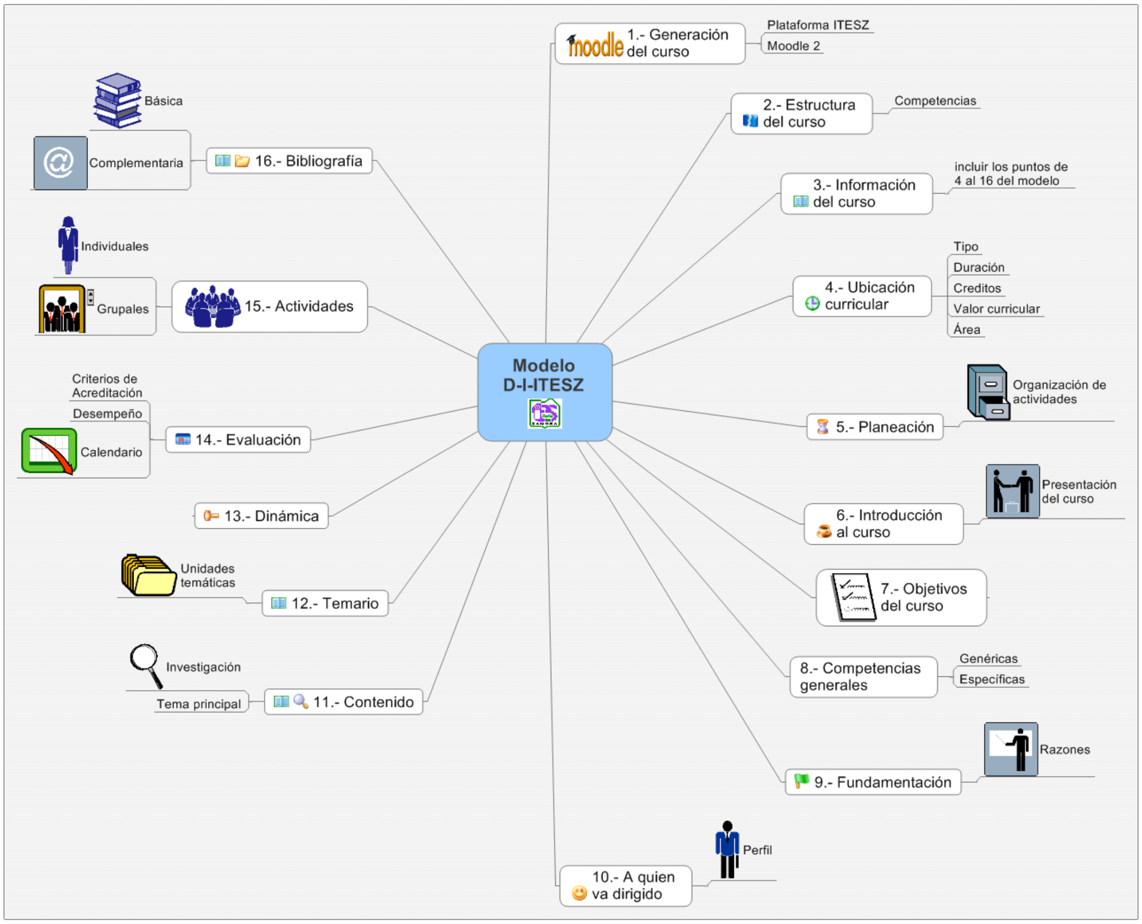


Figura 2.- Modelo D-I-ITESZ

Al seguir cada una de las etapas del modelo se pueden subir cursos a la plataforma virtual y de esta forma utilizar el diseño instruccional para dar educación continua a través de e-learning. La figura anterior fue realizada en un software llamado mind manager.

Conclusiones

Se concluye que es viable la investigación debido a que las variables que se consideraron en este estudio son las que con mayor frecuencia se mencionan en el Marco Teórico, además de haber efectuado la jerarquización analítica para conocer el peso específico que presentaba cada una de ellas. Al hacer la aplicación de las variables, a través del instrumento en el objeto de estudio, se consiguió el resultado del alfa de cronbach de 96%, siendo este el coeficiente de correlación. Y con base en las medidas de tendencia central y variabilidad de todas las variables que se aplicaron se obtuvo que: más del 50% de los docentes están por encima de la mediana. En promedio, los docentes se ubican en 104. (Se considera muy bien).

Se dio respuesta a la pregunta de investigación al informar de que manera se puede ofertar educación continua, y esto es mediante el diseño instruccional. Se cumplió el objetivo de la investigación al llevar a cabo el análisis para ofrecer educación continua a través de e-learning y desarrollar el modelo de diseño instruccional.

Por último se presenta un modelo de diseño instruccional que permite al docente desarrollar cursos en línea con todos los elementos para que el aprendizaje sea significativo. En el modelo de diseño instruccional se emplean diferentes factores los cuales se deben considerar paso a paso para dar inicio a la apertura del curso en línea, tales factores como: seleccionar la plataforma a utilizar, definir la estructura que se dará al curso, plantear los objetivos, la orientación basada en competencias, planeación, evaluación, entre otros.

Referencias

Bonk, C. & Graham, C. (2005). *The Handbook of Blended Learning: global perspectives and local designs.* San Francisco: Pfeiffer

Gimeno, J. (2004). *Computación ubicua*. Consultado el 30 de julio del 2014, del sitio: http://www.laflecha.net/canales/ciencia/articulos/computacion_ubicua/

Holmes, B. & Gardner, J. (2006). *E-learning, concepts and practice*. Londres: Sage.

Khan, B. (2005). *Web-based training: An introduction*. Englewood Cliffs, NJ:Educational Technology Publications.

Santamaría, G. (2009). Redes sociales educativas y comunidades de aprendizaje. En *La Web 2.0 como recurso para la enseñanza del francés como lengua extranjera.*España: Ministerio de Educación, pp. 73-116.

Siemens, G. (2002). Instructional design in e-learning. Consultado el día 30/07/2014, del sitio www.elearnspace.org/Articles/instructionalDesing.htm.

Simonson, M. y otros (2003). Teaching and Learning at a Distance. Foundations of Distance Education (2nd ed.). Upper Saddle River, NJ: Prentice Hall.

A Bloom Filter-Based Algorithm for Routing in Intermittently Connected Mobile Networks

Conference Paper · November 2015
DOI: [illegible]

Rolando Menchaca-Mendez
Instituto Politécnico Nacional

Jairo Javier Sánchez
Instituto Politécnico Nacional

Ricardo Menchaca-Mendez
Centro de Investigación en Computación del Instituto Politécnico Nacional

Jesús García Díaz
Existence

A Bloom Filter-Based Algorithm for Routing in Intermittently Connected Mobile Networks

Jairo Sanchez-Hernandez
Centro de Investigación en Computación
Instituto Politécnico Nacional
jairojsh@gmail.com

Rolando Menchaca-Mendez
Centro de Investigación en Computación
Instituto Politécnico Nacional
rmen@cic.ipn.mx

Ricardo Menchaca-Mendez
Centro de Investigación en Computación
Instituto Politécnico Nacional
ric@cic.ipn.mx

Jesus Garcia-Diaz
Centro de Investigación en Computación
Instituto Politécnico Nacional
jesgadiaz@gmail.com

Mario E. Rivero-Angeles
Centro de Investigación en Computación
Instituto Politécnico Nacional
erivero@cic.ipn.mx

J.J. Garcia-Luna-Aceves
University of California, Santa Cruz
jj@soe.ucsc.edu

ABSTRACT

In this paper, we present a new protocol for routing in intermittently connected mobile networks that, by periodically exchanging constant-size Counting Bloom filters, assigns to every node in the network probabilities of reaching any destination. The gradients defined by these probabilities are further used to forward data packets towards any node in the network. The proposed protocol is based on two novel operations defined over the Bloom filters, namely, the unary *degradation* operation that models the loss of topological information as it gets stale or as it is propagated away from the place where it was generated; and the binary *addition* operation that is used to acquire topological information from other nodes. These two operations are used to implement a probabilistic form of soft state that is defined in terms of the content of the Counting Bloom filters. We present a series of experimental results based on extensive detailed simulations that show that the proposed protocol outperforms the Epidemic routing protocol by delivering more data packets with less delay, while inducing less total overhead in both MANET and VANET scenarios.

Keywords

Delay Tolerant Network, MANET, VANET, Routing, Bloom filters

1. INTRODUCTION

Intermittently Connected Networks (ICNs) are a generalization of the mobile ad hoc networks (MANETs) that does not assume the existence of contemporaneous end-to-end paths connecting any pair of nodes in the network. Since sources and destinations may be disconnected for arbitrarily long periods of time, applications running on this type of networks must be delay tolerant and hence, they are also referred as Delay Tolerant Networks (DTNs) [22]. Delay tolerant networks (DTNs) are a convenient alternative to provide communication services in situations where installing a communication infrastructure is not practical [9]. Examples of such scenarios are search and rescue missions, networks used to collect migration data of Zebras in Africa [10] and information dissemination in sparse vehicular ad hoc networks (VANETs) [7, 12, 21].

An ICN can be modeled by a time varying directed multi-graph where each edge represents a particular type of link with a specific capacity. The latter models situations where two nodes can be joined by different types of network interfaces such as a satellital link or a modem-based telephonic link. In general, the capacity of a link can be seen as a time varying function that takes values from zero capacity when the link is not available, to a predefined maximum capacity that reflects the nature of the link. This way, an edge e_n connecting nodes u, v in the multi-graph indicates that there is a link between nodes u, v in the intermittently connected network. An edge (link) can be defined by $e = ((u, v), c_e(t), d_e(t))$ where $c_e(t)$ is a time varying capacity function, $d_e(t)$ is a time varying delay function and (u, v) is an ordered pair that indicates the direction of the link.

The problem of routing in ICNs can be defined as follows. Given a time-varying multigraph $G(t) = (V(t), E(t))$ and two nodes $o, d \in V(t)$ find a sequence of operations $\gamma_1\gamma_2 \ldots \gamma_n$ that take a packet generated by the origin o to the destination d. The set of γ_i operations are part of the alphabet Γ that is defined as follows.

- Transmission operation denoted by $\gamma_{(u,v)}$: A packet is transmitted from node u to node v
- Storing operation denoted by $\gamma_{store}(u)$: A packet is stored in u's local cache until a condition is met and the packet is either transmitted or deleted.
- Copying operation denoted by $\gamma_{copy}(u)$. Node u creates an exact copy of a data packet.

The problem of routing in intermittently connected networks is known to belong to the class of *NP-Hard* problems

MSWiM'15, November 2–6, 2015, Cancun, Mexico.
© 2015 ACM. ISBN 978-1-4503-3762-5/15/11 ...$15.00.
DOI: http://dx.doi.org/10.1145/2811587.2811609.

[3], even in the case where both, the history of how nodes contact each other and the traffic load are known ahead of time. The proof proceeds by reducing the problem of Edge-Disjoint Paths to the problem of routing in intermittently connected networks.

In this paper, we present a new protocol for routing in ICNs that assigns to every node in the network a probability of reaching any destination. These probabilities are computed by means of the information stored in a set of Counting Bloom Filters that condense the topological information that nodes have collected as they opportunistically encounter other nodes in the network. We say that the information stored at the filters implements a probabilistic form of soft state because it is gradually lost unless it is constantly refreshed and because it provides probabilistic clues about the likelihood of reaching a destination through any node in the network.

The remaining of this paper is organized as follows. Section 2 presents a small summary of the related work. Section 3 presents the proposed routing protocol which is based on a set of novel operations defined over Counting Bloom Filters [6, 19]. Section 4 shows the results of a detailed simulation-based performance analysis of the proposed protocol. We compare the performance of the proposed protocol against that of the Epidemic dissemination protocol which is a *de facto* baseline for performance comparisons of routing protocols for intermittently connected networks. Lastly, Section 5 presents our concluding remarks.

2. RELATED WORK

We present a small but representative sample of the papers that have addressed the problem of routing in intermittently connected networks. Our main propose is to highlight the fact that previous proposal establish individual orderings over the nodes per each of the destinations in the network, and that these orderings are either based on utility functions or probability functions. Unlike these approaches, the proposed protocol simultaneously establishes probability functions for all the destinations using a single constant-size control packet. This property makes the proposed protocol more scalable.

The Epidemic routing algorithm [20] was proposed in 2000 as a simple solution to the problem of routing in delay tolerant networks (DTN). The method used by this algorithm is similar to an infection, where packets represent the disease and the nodes that store a copy of these packets are called "carriers". This way, data packets can spread out very quickly through the connected segments of the network. From this point on, the algorithm exploits the node's mobility to reach other network segments and keep spreading the packets until they reach their intended destination. With this propagation model, packets are almost always delivered, except for the cases in which the destination is located in a completely isolated network component, when packets are discarded at the data queues or when the destination does not even exists. It is important to point out that Epidemic is a brute force algorithm that employs every possible route in the network to reach the destination and hence, is very costly in terms of bandwidth utilization and the memory space needed to store the data packets at the queues. EM-MA [8] is a recently proposed variant of epidemic routing that uses aggregation to reduce overhead, unfortunately, the scalability problems remain.

PRoPHET [13] is a probability based routing protocol for DTNs that take advantage of the repetitive behavioral patterns of the human users. For example, if a node has visited a location many times in the past, there is a high probability that this node will be there in the near future. PRoPHET employs a metric called *delivery probability* $P_{(a,b)} \in [0, 1]$ that indicates how probable is that a node a can actually deliver messages to node b. Whenever two nodes come into transmission range, they exchange summaries that contain a list of packets and their delivery probabilities. The information of the summary is then used to request those packets for which the current node is a better forwarder. In [5], the authors propose the space-content adaptive-time routing (SCaTR) framework for DTN MANETs. SCaTR is an extension of traditional on demand routing that takes action when the protocol is unable to establish a connected route. When in DTN mode, SCaTR uses contact values to select the best node that can act as proxy of the destination and store packets at that node until either the destination is discovered, or another node is selected as a better proxy. Contact values are computed based on the past connectivity information that nodes keep in their contact tables. In [18], the authors propose a series of single-copy strategies which are based on utility functions that employ last encounter timers to establish an ordering over the nodes in the network. A data packet is forwarded to a next hop if its priority is larger than that of the current node plus a constant threshold. The authors also propose a hybrid routing protocol, called "Seek and Focus" that first uses randomized forwarding and then utility-based forwarding. In [17] the same authors propose similar multiple-copy strategies for routing in intermittently connected networks. GeoSpray [16] is a routing protocol for Vehicular Delay-Tolerant Networks (VDTN) that employs a hybrid approach of single and multi-copy routing. GeoSpray starts with a multiple-copy approach where a number of copies are spread in order to exploit alternative paths. Then, it switches to a forwarding scheme, where nodes forward data packets to nodes they encounter if they have a better estimated time of delivery.

For a more comprehensive analysis of the large amount of work reporting routing protocols for intermittently connected networks, please refer to the surveys by Benamar et-al [4] and Cao and Sun[7].

Bloom filters or any of their many variants have been used in a large number of applications for distributed systems such as caching, peer-to-peer networks, routing and forwarding, monitoring and measurement and data summarization [19]. Their use as a tool to compute routes, is less extensive, mainly because of the complications introduced by false positives [15]. However, recent proposals such as [14][23] are able ameliorate these complications by either assuming or inducing a tree-like topology. As discussed in Section 3.4, the probability of false positives have no impact on the correctness of the proposed protocol.

3. ROUTING USING COUNTING BLOOM FILTERS

3.1 Overview

The proposed protocol assigns to nodes a probability of reaching every other node in the network. To establish these probabilities throughout the network, nodes periodically exchange counting Bloom filters that condense the topological information they have been able to collect. When a node receives a filter from a neighbor, it uses the *addition* operation to merge its current topological information with the one contained in the received filter. Additionally, nodes use the *degradation* operation to implement a probabilistic form of soft state, in which the information about the destinations is gradually lost as it grows stale and as it is being propagated away from the place where it was generated.

From the filters received from each neighbor, nodes compute a probability of reaching any destination D through that particular neighbor. The gradients defined by these probabilities are further used to disseminate data packets in a store-carry-and-forward fashion to their intended destinations. A given data packet will be forwarded to a neighbor when the probability of reaching its intended destination plus a constant threshold, is larger than the probability of the current node. This way, packets tend to travel only to those regions of the network where it is more likely to find their intended destination. Please note that based on the probabilities assigned to nodes, many other forwarding strategies can be defined.

In the following sections, we present more detailed descriptions of the proposed operators and of the way the probabilities of reaching a destination are computed.

3.2 Filter Operations

A *Counting Bloom Filter* F is an array of m counters that can take values from 0 to c. Given a filter F, $F[x]$ denotes the value of the x-*th* counter (with $x \in 0, ..., m-1$) contained in the filter. We use $\mathcal{F}_m^c$ to denote the set of every filter composed of m counters with capacity c. We also denote the empty filter as F_0, where all counters are set to 0. The three operations defined over the filters are as follows.

The *insert* operation is used to add the identity of a node into a Counting Bloom filter. It takes as parameters a filter F and the identifier $a \in ADDR$ of a node (where $ADDR$ is the space of node identifiers in the network) and returns a new filter that contains the identity a. To insert the identity a in a filter F, denoted as $F + a$, the value of k independent hash functions ($h_i : ADDR \rightarrow \{0 \dots m-1\}$ with $i \in \{1 \dots k\}$) is calculated to obtain the indexes of k counters that will be set to the value of c. In this way, the operation defined by $F + a$ results in $F[h_i(a)] \leftarrow c$ with $i \in \{1 \dots k\}$. Note that this operation is different from the traditional insertion operation defined for Counting Bloom Filters where the counters are incremented in one unit.

The *degradation* operation is an unary operation that consists in decrementing in a single unit the value of each counter with probability $p_{degrade}$. This operation is denoted by $\Delta_{p_{degrade}} : \mathcal{F}_m^c \rightarrow \mathcal{F}_m^c$ and is defined by the Algorithm 1. The degradation operation is used to model situations where some of the information contained in a filter is lost; either because it is growing stale or because it is information that is being propagated away from the place it was generated.

Algorithm 1 $\Delta_{p_{degrade}}(F)$

```
for i = 0 → m − 1 do
    if F[i] > 0 then
        F[i] ← F[i] − 1; with a probability p_degrade
    end if
end for
```

The third operation is the binary *addition* operation which is denoted by $\oplus : \mathcal{F}_m^c \times \mathcal{F}_m^c \rightarrow \mathcal{F}_m^c$. This operation combines the value of two filters F_1 and F_2 following the Algorithm 2.

Algorithm 2 Addition(F_1, F_2)

```
Fr is a new filter;
for l = 0 → m − 1 do
    F_R[l] ← max{F_1[l], F_2[l]}
end for
return F_R
```

3.3 Probabilistic Soft State

Each node i will keep two filters that store its own topological information, namely, F_i^* and F_i^t. F_i^* is constant through time and is defined as $F_0 + i$, whereas F_i^t is time dependent and its content changes because of the updates received from neighboring nodes and from a periodical degradation process. When a node i receives a filter F_j^t from its neighbor j, it updates the value of its own filter F_i^{t+1} according to Eq. (1). Please note that in Eq. (1) the filter received from neighbor j is first degraded to reflect the loss of information and then added to the filter of node i. Additionally, every node i stores in the structure $\mathcal{N}_i$ the latest received filters from all of its neighbors.

$$F_i^{t+1} \leftarrow \Delta_{p_{degrade}}(F_j^t) \oplus F_i^t \tag{1}$$

The filter F_i^t is periodically degraded according to Eq. (2) as a way to implement a form *probabilistic soft state*. The concept of probabilistic soft state is similar to the concept of *Weak State* proposed in [2] in the sense that both provide probabilistic hints regarding the location of the destinations. Unlike Weak State that is based on geographic information, the probabilistic soft state proposed in this work is strictly based on topological information. Moreover, the probabilistic soft state has a spatial/temporal nature because it captures the fact that information is lost as times goes by, but also as it is disseminated away from the place where it was generated.

$$F_i^{t+1} \leftarrow \Delta_{p_{degrade}}(F_i^t) \oplus F_i^* \tag{2}$$

The probability of reaching a destination through node i is calculated with Eq. (3). The probability assigned to each node in the network is precisely the one that defines the probabilistic gradient used to reach the destinations. Note that under this scheme, the probability of reaching node i through itself is always 1.

$$pr_i^D \leftarrow \frac{\sum_{\forall x | F_D^*[x] > 0} F_i^t[x]}{kc} \tag{3}$$

3.4 Protocol Description

Our protocol is defined in terms of a sequence of messages that are exchanged by nodes as they opportunistically meet each other. The messages are as follows.

- BLF: Contains filter F_i^t which condenses the topological information collected by node i.
- SUV: Array of packet identifiers. It is used to publish the set of messages stored at this node.
- REQ: Array of packet identifiers. Set of messages requested by this node.

- DAT: Message that contains a set of data packets.

Every node j broadcasts a hello message m_{HLF} containing his own Bloom filter F_j^t every interval of t_h seconds. As shown in Algorithm 3, upon reception of a message m_{HLF} from node j, node i proceeds to update its own filter by means of Eq. (1) and to compute the new delivery probabilities through node j by means of Eq. (3). Then, node i adds the identifiers of the data packets with destination D, such that the delivery probability at node j is greater than the delivery probability at node i plus a threshold ($pr_j^D \geq pr_i^D + U_e$), to the list L_{sum} which is sent in a m_{SUV} message to j. This way, the packets listed in L_{sum} are such that their destinations are more likely to be reached through i than through j.

Algorithm 3 Bloom Filter Message

```
when m_HLF is received from j do:
    L_sum ← ∅
    F_i^t ← F_i^t ⊕ F_i^*
    F_i^{t+1} ← Δp_degrade(F_j^t) ⊕ F_i^t
    for Message m ∈ BUFFER_i do
        D ← m.destination
        pr_i^D ← (Σ_{∀x|F_D^*[x]≠0} F_i^{t+1}[x]) / km
        pr_j^D ← (Σ_{∀x|F_D^*[x]≠0} F_j^t[x]) / km
        if pr_j^D ≥ pr_i^D + U_e ∨ pr_j^D = 1 then
            L_sum ← L_sum ∪ {m.id}
        end if
    end for
    send(j, L_sum)
end when
```

As described by the pseudocode of Algorithm 4, when node i receives the message m_{SUV}, it creates a new list L_{req} that contains the packets listed in L_{sum} that have not been received so far. Then, node i sends a message m_{REQ} back to j requesting the messages in L_{req}.

Algorithm 4 Summary Message

```
when m_SUV is received from j do:
    Lreq ← {∅}
    for Identifier m_i ∈ L_sum do
        if m_i ∉ BUFFER_i then
            Lreq ← Lreq ∪ {m_i}
        end if
    end for
    send(j, Lreq)
end when
```

When node j receives the list L_{req}, it sends a message m_{DAT} back to i containing every data packet requested by i. This action can be seen as a series of consecutive $\gamma_{(j,i)}$ operations. Lastly, when node i receives the data packets, it passes to upper layers those packets which are intended to the node itself and stores the others in its internal buffer. This latter action can also be seen as a series of $\gamma_{store}(i)$ operations.

As already mentioned, every node i in the network periodically degrades its own filter F_i^t by means of Eq. (2). This way, node i will eventually lose any state regarding destinations that have move far away or to a different network partition.

It is important to point out that in the presence of false positives, data packets can be disseminated towards regions of the network that do not contain the destination, however, this will not prevent the proposed routing protocol to also disseminate the packets towards the true positive, namely, towards the intended destination. Therefore, while a false positive does increase the cost of delivering a data packet, it does not prevent the data packet from reaching its destination. In the worst case, our proposed protocol will behave like a pure epidemic dissemination protocol but with the extra overhead induced by exchanging the Bloom filters.

4. EXPERIMENTAL RESULTS

In this section, we present the results of a series of simulation experiments comparing the performance of the proposed protocol against that of the Epidemic routing protocol. We selected Epidemic because it has become a *de facto* baseline for performance comparisons of routing protocols for intermittently connected networks and because both are multi-copy protocols that disseminate data packets from sources to destinations. The latter allow us to highlight the performance gains obtained by disseminating data packets only towards those regions of the network where it is likely to find the destination. Both protocols use the same tail drop policy to discard data packets when a data queue has reached its maximum capacity, and the same Hello interval of 1 second. For the proposed protocol, the values used for the degradation interval, the degradation probability ($p_{degrade}$) and the U_e threshold were hand-picked. However, a further sensitivity analysis revealed that the performance of the protocol is fairly insensitive to moderate variations (±15%) of these values. We decided to use relatively small filters (with only 64 counters) to characterize the performance of the proposed protocol under relatively high probabilities of false positives. For all the experiments, we used the NS2 simulator version 2.34 [1].

We designed two sets of experiments. The first one is intended to evaluate the performance of the protocols in MANET scenarios where nodes are sparsely spread around a large simulation area and move following the random waypoint mobility model. In the second set of experiments, we evaluate the performance of the protocols in a VANET scenario where nodes move following the street layout of the city of Murcia in Spain which is shown in Figure 1. For the two sets of experiments we use packet delivery ratio, end-to-end delay and total overhead as performance metrics. The results presented in all the figures are the average of 20 independent runs with a confidence level of 95%.

4.1 MANET Scenario

For the MANET scenario, we defined a set of three experiments to assess the impact of different variables over the performance of the proposed protocol. In Experiment 1, we vary the amount of traffic injected into the network to evaluate the effectiveness of the protocols to deliver data packets before they have to be dropped at the data queues. In Experiment 2, we decrease the node density so that the networks become more disconnected, and in Experiment 3, we evaluate the effect of decreasing the node mobility. The objective of Experiment 3 is to evaluate the effectiveness of the proposed algorithm to accurately reflect the topological changes into the counting Bloom filters. The details about the values of the simulation parameters that are fixed across the three experiments are shown in Table 1.

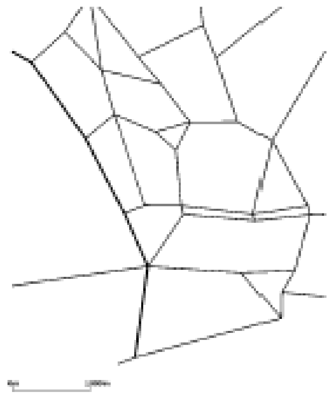


Figure 1: Street layout used in the VANET scenario.

Variable description	Value
Data flows	[1]
Simulation area (square area)	[2]
Pause time	[3]
Mobility model	Random Waypoint
Nodes in the network	100
Node placement	Random
Minimum velocity for nodes	1 m/s
Maximum velocity for nodes	20 m/s
Data flow type	CBR
Packets per data flow	1000
Tx. rate	1 packet/sec
Propagation model	Omni-directional Two-ray ground
Transmission range	250 m
Packet size	128 bytes
Simulation time	5000 seconds
Buffer capacity	2048 packets
U_a Threshold	0.1
Degradation probability ($p_{degrade}$)	0.5
Bloom filter size (m)	64
Counter capacity (c)	32
Number of hash functions (k)	4
Degradation interval	3 seconds
HELLO interval (t_h)	1 second ($\pm$0.25)

Table 1: Simulation environment.

Table 2 shows the specific values of the parameters that are varied in each experiment. The column of *reference values* corresponds to the values of the variables that are fixed during a given experiment.

Figures 2, 3 and 4 present the simulation results for the two protocols under the three different scenarios. For all the experiments, the end-points of the data flows were selected at random using a uniform probability distribution. Figure 2 shows the results of Experiment 1, where we increase the amount of traffic injected into the network by increasing the number of concurrent data flows from 10 to 20. Fig. 3 corresponds to the results of Experiment 2, where we decrease the node density by keeping constant the number of nodes and by increasing the simulation area. Lastly, Figure 4 shows the results of Experiment 3, where we reduce the mobility of the nodes by increasing the pause time from 10 seconds to 40 seconds.

Experiment	Values	Reference value
1	10,15,20 flows	10
2	2000m × 2000m 3000m × 3000m 4000m × 4000m	3000m × 3000m
3	10s,20s,40s	0s

Table 2: Specific parameters for each experiment.

Regarding packet delivery ratio, from Figures 2(a), 3(a) and 4(a) we can observe that the proposed protocol clearly outperforms Epidemic by consistently delivering more data packets. The inferior packet delivery ratio attained by Epidemic is caused by a rapid saturation of the buffers of the nodes. On the other hand, the proposed algorithm only propagates packets towards those regions of the network where it is more likely to find the destination and hence it does not waste buffer space of nodes that are located far away from the intended destinations (see Figure 2(a)). For the case of increasing traffic load, the behavior shown by the proposed protocol is as expected. When the traffic increases, the packet delivery ratio is reduced because more packets are being disseminated across the network and hence, more packets have to be dropped at the data queues. On the other hand, when the node density is increased (see Figure 3(a)), nodes tend to meet each other more often and hence, the topological information regarding the destinations percolates faster across the network. Under this situation, the probabilistic gradients are more clearly defined and the dissemination of the packets tends to be more focused towards the destinations. It is also important to point out that when the network becomes more connected, the sequence of SUV-SUV-REQ-DAT messages also becomes more costly. Moreover, we observed the appearance of the broadcast storm problem where many nodes simultaneously engage in SUV-REQ-DAT exchanges with a node transmitting a BLF packet. Lastly, Figure 4(a) shows that the two protocols are fairly insensitive to the different values used for the pause time. These results indicate that under these conditions, the probabilistic soft state used by the proposed protocol is capable of keeping up with the topological changes experienced by the network.

Regarding end-to-end delay, from Figures 2(b), 3(b) and 4(b) we can observe that the proposed protocol clearly outperforms Epidemic by consistently attaining smaller delays. This was also an expected result because Epidemic is a brute force solution that uses all possible paths to reach the intended destinations but that tend to saturate the data queues much faster. Moreover, a detailed analysis of the simulation traces revealed that the delay attained by Epidemic is also due to the fact that most of the packets were delivered to destinations that happened to be relatively close to their corresponding sources. The latter contrast with the proposed protocol that is able to deliver packets to destinations that are located far away from the sources, or in different network partitions.

Lastly, from Figures 2(c), 3(c) and 4(c) we can notice that the extra cost induced by the proposed protocol by periodically transmitting counting Bloom filters is clearly outweighed by the data overhead induced by Epidemic while disseminating data packets towards regions of the network that do not contain the intended destinations.

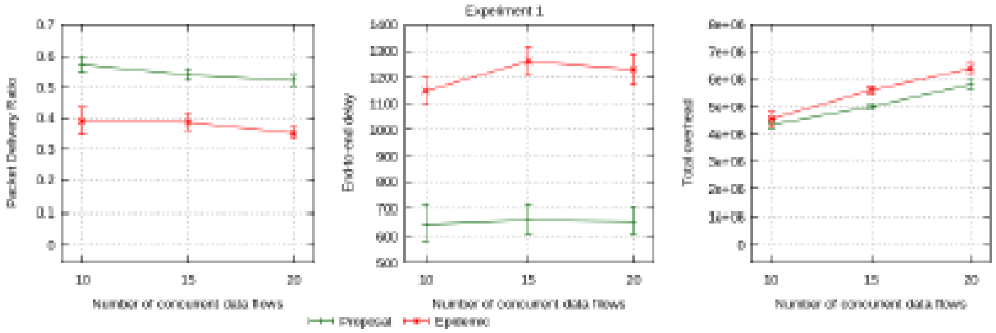


Figure 2: Performance of the protocols in a MANET scenario with increasing traffic load. (a) Packet delivery ratio. (b) End-to-end delay. (c) Total overhead.

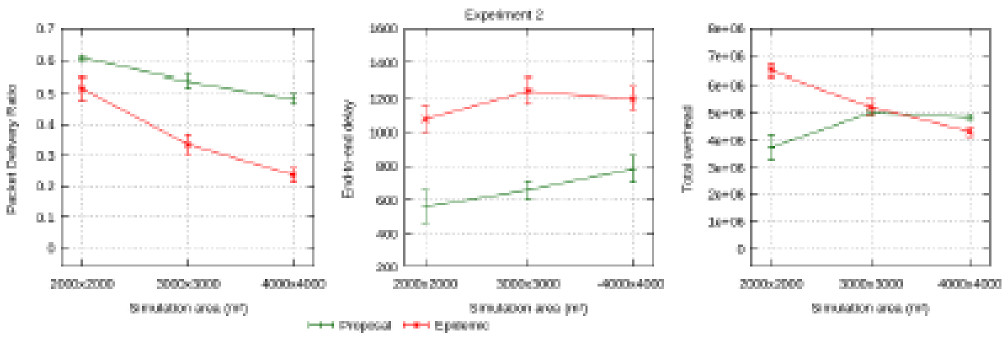


Figure 3: Performance of the protocols in a MANET scenario with decreasing node density. (a) Packet delivery ratio. (b) End-to-end delay. (c) Total overhead.

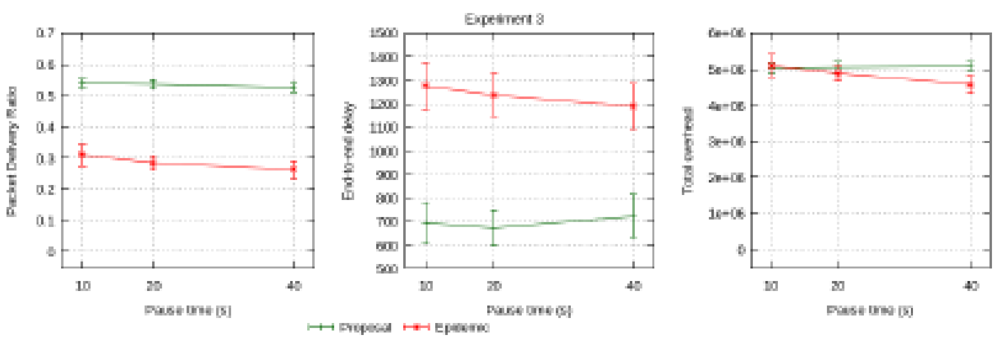


Figure 4: Performance of the protocols in a MANET scenario with decreasing node mobility. (a) Packet delivery ratio. (b) End-to-end delay. (c) Total overhead.

Vehicles entering the scenario every 45s	Flows	Routes	Avg. num. of vehicles
05	25	25	55
15	25	25	155
25	25	25	270
35	35	30	385
45	55	30	520

Table 3: Vehicle flow parameters for the VANET scenario.

Parameter	Value
Data flows	10
Packets per flow	1000
Packet size	512 bytes
Data rate	1 packet/s

Table 4: Parameters for VANET data flows

4.2 VANET Scenario

In this scenario, we evaluate the performance of the protocols when nodes move following the street layout of the city of Murcia, Spain (Figure 1). We used the SUMO simulator of urban mobility [11] to generate mobility traces of two types of vehicles, namely, *cars* and *buses*, which differ only on mobility parameters such as acceleration, maximum speed, and size. For this experiments we use the same set of values for the simulation parameters as described in Table 1, and vary the vehicle density by increasing the number of vehicles entering the simulation area per second as shown in the first column of Table 3. The second column of Table 3 corresponds to the number of different vehicle flows in the scenario. In SUMO, a flow defines the rate at which vehicles enter the scenario, as well as their type (car or bus) and the route they follow. Routes are defined in terms of a sequence of street intersections. The fourth column of Table 3 presents an estimate of the average number of active vehicles in the simulation area. This value is a function of the vehicle arrival rate and, the time vehicles remain active on the simulation, which depends on the length of the routes followed by the vehicles, the behavior of the semaphores, and the traffic congestion of the road network. As in the MANET scenario, the end-points of the CBR data flows are selected uniformly at random from the vehicles currently in the simulation. The details of the vehicle flow parameters are shown in Table 4.

Figure 5 presents the results of these experiments. From Figures 5(a) and (b), we can observe that the proposed algorithm consistently performs similar or better than Epidemic in terms of packet delivery ratio, end-to-end delay and total overhead. Our proposal delivers between 5% and 7% more packets than Epidemic, even in situations where the probability of false positives in the counting Bloom filters is quite high due to the large number of active vehicles in the simulation and the relative small size of the filters (32 counters). Under this situation, our protocol behaves similar to Epidemic because it also tend to disseminate data packets towards regions of the network that do not contain the destination but a false positive. Figure 5(b) shows that in these experiments the end-to-end delay attained by the proposed protocol is slightly better than that of Epidemic. The reason again, is the fact our protocol induces less data overhead, which reduces the waiting times at the data queues.

Lastly, Figure 5(c) shows the total overhead induced by the two protocols. As in the case of the MANET scenario, the proposed protocol incurs in far less overhead because of the reduced cost of disseminating only towards regions of the network where it is likely to find the destinations. This is true even in this scenario where the probability of false positive is high.

5. CONCLUSIONS

In this paper, we proposed a new protocol for routing in intermittently connected mobile networks that uses the concept of *probabilistic soft state* to assign to every node in the network probabilities of reaching any destination. The probabilistic soft state is defined in terms of the content of counting Bloom filters that condense the topological information that nodes have collected from other nodes as they move and that is used to compute the probabilities of reaching the destinations. These probabilities reflect the amount of information nodes have regarding the different destinations. To model the process in which nodes acquire and loss topological information, we defined two novel operations over counting Bloom filters, namely, the binary *addition* operation and the unary *degradation* operation. The addition operation is used to reflect the acquisition of new information, and the unary operation is used to reflect the loss of information, either because it is getting stale or because it is being propagated away from the place where it was generated. Then, nodes use a greedy forwarding strategy that follows the gradients defined by these probabilities to reach the intended destinations.

The proposed scheme has the advantage of establishing probabilistic gradients simultaneously for all the destinations. This contrasts with previous proposals that require the dissemination of control information for every destination node, which is not scalable. In the proposed scheme, nodes periodically exchange constant-size "hello" messages with their one-hop neighbors. Therefore, the protocol's network complexity is constant in terms of the network size. Moreover, the proposed scheme has the advantage that it combines in a simple way the concepts of probability of encounter and hop distance towards the destinations. When the network is connected, the probabilistic gradients are similar to the traditional gradients based on hop distances, but when the network is partitioned, the gradients are similar to those that compute a probability of encounter based on the time nodes last met. This way, the proposed algorithm can work in more general scenarios than previous protocols that exploit the "social behavior" of nodes.

In the presence of false positives in the Bloom filters, data packets can be disseminated towards regions of the network that do not contain the destination, however, this will not prevent the proposed protocol to also disseminate the packets towards the intended destination. We observed this situation in the VANET scenario, where due to the large number of nodes, the probability of false positive is quite high.

The results of a detailed simulation-based performance analysis showed that the proposed algorithm is a viable alternative for routing in both intermittently connected MANETs and VANETs. The proposed protocol consistently outperformed the Epidemic routing protocol in both MANET and VANET scenarios.

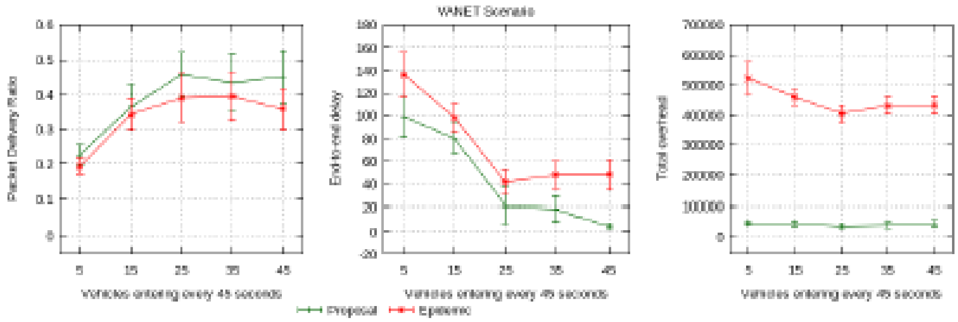


Figure 5: Performance of the protocols in a VANET scenario with increasing node density. (a) Packet delivery ratio. (b) End-to-end delay. (c) Total overhead.

6. ACKNOWLEDGMENTS

Work partially sponsored by the UC MEXUS-CONACyT program, and by the Mexican National Polytechnic Institute (IPN).

7. REFERENCES

[1] The Network Simulator, ns-2.34. http://www.isi.edu/nsnam/ns/.

[2] U. G. Acer, S. Kalyanaraman, and A. A. Abouzeid. Weak state routing for large-scale dynamic networks. *IEEE/ACM Transactions on Networking (TON)*, 18(5):1450–1463, 2010.

[3] A. Balasubramanian, B. Levine, and A. Venkataramani. DTN routing as a resource allocation problem. *SIGCOMM Comput. Commun. Rev.*, 37(4):373–384, Aug. 2007.

[4] N. Benamar, K. D. Singh, M. Benamar, D. El Ouadghiri, and J.-M. Bonnin. Routing protocols in vehicular delay tolerant networks: A comprehensive survey. *Computer Communications*, 48:141–158, 2014.

[5] J. Boice, J. Garcia-Luna-Aceves, and K. Obraczka. Combining on-demand and opportunistic routing for intermittently connected networks. *Ad Hoc Networks*, 7(1):201–218, 2009.

[6] A. Broder and M. Mitzenmacher. Network applications of bloom filters: A survey. *Internet Mathematics*, 1(4):485–509, 2004.

[7] Y. Cao and Z. Sun. Routing in delay/disruption tolerant networks: A taxonomy, survey and challenges. *IEEE Communications Surveys & Tutorials*, 15(2):654–677, 2013.

[8] T. Choksatid and S. Prabhavat. An epidemic routing with low message exchange overhead for delay tolerant networks. In *Progress in Systems Engineering*, pages 429–436. Springer, 2015.

[9] S. Jain, K. Fall, and R. Patra. Routing in a delay tolerant network. In *Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '04, pages 145–158, New York, NY, USA, 2004. ACM.

[10] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein. Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with zebranet. *SIGOPS Oper. Syst. Rev.*, 36(5):96–107, Oct. 2002.

[11] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker. Recent development and applications of SUMO - Simulation of Urban MObility. *International Journal On Advances in Systems and Measurements*, 5(3&4):128–138, December 2012.

[12] F. Li and Y. Wang. Routing in vehicular ad hoc networks: A survey. *IEEE Vehicular Technology Magazine*, 2(2):12–22, 2007.

[13] A. Lindgren, A. Doria, and O. Schelén. Probabilistic routing in intermittently connected networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 7(3):19–20, July 2003.

[14] A. Reinhardt, O. Morar, S. Santini, S. ZöĹller, and R. Steinmetz. Cbfr: Bloom filter routing with gradual forgetting for tree-structured wireless sensor networks with mobile nodes. In *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2012*, pages 1–9. IEEE, 2012.

[15] M. Sarela, C. E. Rothenberg, T. Aura, A. Zahemszky, P. Nikander, and J. Ott. Forwarding anomalies in bloom filter-based multicast. In *Proceedings of IEEE INFOCOM 2011*, pages 2399–2407. IEEE, 2011.

[16] V. N. Soares, J. J. Rodrigues, and F. Farahmand. Geospray: A geographic routing protocol for vehicular delay-tolerant networks. *Information Fusion*, 15:102–113, 2014.

[17] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Efficient routing in intermittently connected mobile networks: the multiple-copy case. *IEEE/ACM Transactions on Networking*, 16(1):77–90, 2008.

[18] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Efficient routing in intermittently connected mobile networks: The single-copy case. *IEEE/ACM Transactions on Networking (TON)*, 16(1):63–76, 2008.

[19] S. Tarkoma, C. E. Rothenberg, and E. Lagerspetz. Theory and practice of bloom filters for distributed systems. *IEEE Communications Surveys & Tutorials*, 14(1):131–155, 2012.

[20] A. Vahdat and D. Becker. Epidemic routing for partially-connected ad hoc networks. Technical report, Duke University, 2000.

[21] N. Wisitpongphan, F. Bai, P. Mudalige, V. Sadekar, and O. Tonguz. Routing in sparse vehicular ad hoc wireless networks. *IEEE Journal on Selected Areas in Communications*, 25(8):1538–1556, 2007.

[22] Z. Zhang. Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: overview and challenges. *IEEE Communications Surveys & Tutorials*, 8(1):24–37, 2006.

[23] H. Zheng and J. Wu. Up-and-down routing in mobile opportunistic social networks with bloom-filter-based hints. In *IEEE 22nd International Symposium of Quality of Service (IWQoS) 2014*, pages 1–10, May 2014.

J Heuristics (2017) 23:349–366
DOI 10.1007/s10732-017-9345-x

When a worse approximation factor gives better performance: a 3-approximation algorithm for the vertex k-center problem

Jesus Garcia-Diaz[1] · Jairo Sanchez-Hernandez[1] · Ricardo Menchaca-Mendez[1] · Rolando Menchaca-Mendez[1]

Received: 15 July 2015 / Revised: 11 March 2017 / Accepted: 22 June 2017 / Published online: 1 July 2017

Abstract The vertex k-center selection problem is a well known NP-Hard minimization problem that can not be solved in polynomial time within a $\rho < 2$ approximation factor, unless $P = NP$. Even though there are algorithms that achieve this 2-approximation bound, they perform poorly on most benchmarks compared to some heuristic algorithms. This seems to happen because the 2-approximation algorithms take, at every step, very conservative decisions in order to keep the approximation guarantee. In this paper we propose an algorithm that exploits the same structural properties of the problem that the 2-approximation algorithms use, but in a more relaxed manner. Instead of taking the decision that guarantees a 2-approximation, our algorithm takes the best decision near the one that guarantees the 2-approximation. This results in an algorithm with a worse approximation factor (a 3-approximation), but that outperforms all the previously known approximation algorithms on the most well known benchmarks for the problem, namely, the *pmed* instances from OR-Lib (Beasly in J Oper Res Soc 41(11):1069–1072, 1990) and some instances from TSP-Lib (Reinelt in ORSA J Comput 3:376–384, 1991). However, the $O(n^4)$ running time of this algorithm becomes unpractical as the input grows. In order to improve its running time, we modified this algorithm obtaining a $O(n^2 \log n)$ heuristic that outperforms

✉ Jesus Garcia-Diaz
jesgadiaz@gmail.com

Jairo Sanchez-Hernandez
jairojsh@gmail.com

Ricardo Menchaca-Mendez
ric@cic.ipn.mx

Rolando Menchaca-Mendez
rmen@cic.ipn.mx

1 Centro de Investigación en Computación, Instituto Politécnico Nacional, Av. Luis Enrique Erro S/N, Unidad Profesional Adolfo López Mateos, Mexico City, Mexico

not only all the previously known approximation algorithms, but all the polynomial heuristics proposed up to date.

Keywords k-Center problem · Approximation algorithm · Dominating set · Polynomial time heuristic

1 Introduction

The vertex k-center problem (Hakimi 1964) is a well known NP-Hard minimization problem that is defined as follows. Given a complete undirected graph $G = (V, E)$ with edge costs satisfying the triangle inequality, and a positive integer k; find a subset $C \subseteq V$ of centers, with $|C| \leq k$, that minimizes the objective function $r(C) = \max_{v \in V}\{d(v, C)\}$, where $d(v, C)$ is defined as the cost of the cheapest edge from a vertex v to the vertices in C.

The value $r(C)$ is known as the *covering radius* of the solution C, and the vertex k-center problem can be rephrased as the problem of finding a subset of at most k centers with minimum covering radius. Solving the k-center problem to optimality or within any approximation factor $\rho < 2$ is NP-Hard (Garey and Johnson 1979; Kariv and Hakimi 1979; Hochbaum and Shmoys 1985; Gonzalez 1985) .

It is important to point out that the problem of finding a set of centers of size at most k with minimum covering radius, is also an NP-Hard minimization problem even if the optimal covering radius is known ahead of time. To see this, note that the optimal covering radius must be equal to the length of one of the edges of the input graph, and hence, by running a polynomial algorithm that solves the k-center problem when the optimal covering radius is known, using the at most $(n^2 - n)/2$ different edge lengths, and picking the best solution, would solve the vertex k-center problem in polynomial time. In fact, the vertex k-center problem with known optimal covering radius is related to the Minimum Set Covering and Minimum Dominating Set problems which are also well known NP-Hard problems. The vertex k-center with known optimal covering radius problem can be stated as the problem of finding a Dominating Set of size at most k, over a pruned version of the original graph where the edges with cost greater than the optimal size are removed. Many heuristics, exact algorithms, and the approximation algorithm we present in this paper make extensive use of these relationships.

The vertex k-center problem has many practical applications, mainly in the area of facility location. Among others, we can mention applications to the location of diabetes attention units (Pacheco and Casado 2005) and to bicycle stations (Kaveh and Nasr 2011). Given its practical importance, the vertex k-center problem has been approached with a wide variety of algorithmic tools. Minieka (1970) was the first to propose a heuristic (long considered inefficient) for the k-center problem based on its relationship with the Set Covering problem. Later, Gonzalez (1985), Hochbaum and Shmoys (1985) and Dyer and Frieze (1985) independently proposed two polynomial 2-approximation algorithms. The algorithm by Gonzalez, and by Dyer and Frieze is a greedy-like farthest point algorithm, and the algorithm by Hochbaum and Shmoys is based on the parametric pruning technique and the relationship between the k-

center problem and the Dominating Set problem. Some years later, Mladenović et al. (2000) proposed a set of Tabu Search and Variable Neighborhood Search algorithms that were able to generate the best-known solutions within reasonable time for what would became the main benchmark data sets for the comparison of algorithms for the vertex k-center problem (some instances from OR-Lib and TSP-Lib). Shortly after, a set of exact algorithms inspired by Minieka's idea were developed by Daskin (2000), Ilhan and Pinar (2001), and Elloumi et al. (2004). Through experimentation, these exact algorithms proved to be practical since for some instances they converged quickly to the optimal solution; for some other instances they didn't converge, but generated near optimal solutions within a reasonable amount of time. On the other hand, the vertex k-center problem has also been addressed through other heuristic and metaheuristic methods, such as Scatter Search (Pacheco and Casado 2005), Genetic Memetic Algorithms (Pullan 2008), Bee Colony Optimization (Davidović et al. 2011), Harmony Search (Kaveh and Nasr 2011), and Structure-driven Randomization (Garcia et al. 2015). Among the heuristic approaches, the pure greedy algorithm (Gr) stands as the most natural and simple, yet inefficient method according to most experimental results, while the Scoring (Scr) algorithm, which is based on the relationship between the k-center problem and the Dominating Set problem, stands among the best (Mihelič and Robič 2005; Rana and Garg 2008).

1.1 Approximation algorithms

Approximation algorithms provide solutions for NP-Hard optimization problems that are guaranteed to be within a ρ-approximation factor. In the case of a minimization problem, such as the vertex k-center problem, a ρ-approximation algorithm produces a solution of value at most $\rho \cdot OPT$ for every instance, where OPT is the value of the optimal solution. Many authors believe that approximation algorithms with good approximation factors will perform well in practice because they take advantage of knowledge about the structure of the problem in order to keep the approximation guarantee (Vazirani 2001; Hochbaum 1997). The hope is that the solutions provided by the approximation algorithm will reach the ρ-approximation factor only on a few pathological instances. Even if the interest is in generating optimal solutions, the approximation algorithms can serve as guidance for the design of better heuristics.

For the case of the vertex k-center problem, the 2-approximation algorithms proposed so far operate in a greedy like fashion. For instance, Shmoys (1995) showed that if the optimal covering radius r is known, then an algorithm that builds a solution by selecting vertices as centers one at a time, such that the distance of the selected vertex to the current partial solution is larger than $2r$, results in a 2-approximation solution. Moreover, if the previous procedure does not produce a solution of cardinality at most k, then it implies that the proposed covering radius r was in fact smaller than the optimal. This becomes the basis of a binary search over the covering radius r for the 2-approximation HS algorithm (Hochbaum and Shmoys 1985). The algorithm by Gonzalez eliminates the need of performing a binary search by noticing that always picking the farthest vertex to the current partial solution suffices to guarantee that the distance from the selected vertex to the partial solution is larger

than 2 times the optimal covering radius (1985). However, the experimental results reported by Mihelič and Robič (2005), and Rana and Garg (2008), as well as those reported in this paper seem to indicate that forcing the next center to be at a distance greater than $2r$ from the current partial solution is sometimes too conservative and makes the algorithms miss the selection of centers that look promising at a local level. For instance, vertices with many non-dominated neighbors at a distance smaller than r from it, which are located at a more convenient distance from the current partial solution, which might be smaller than $2r$. Our hypothesis is that relaxing the farthest-vertex rule will allow the algorithm to search for solutions that are actually close to the optimal and not just settle for finding a 2-approximation. To test this hypothesis, in this paper we propose a new approximation algorithm that rather than selecting the farthest vertex to the current partial solution, selects as center a vertex that is close to the farthest vertex -at a distance not greater than r- and which is locally optimal in terms of the number of non-dominated neighboring vertices also at a distance not greater than r. This new way of selecting the next center results in an algorithm with a worse approximation factor—a 3-approximation—but that outperforms the previously known 2-approximation algorithms over the most representative benchmarks from the literature. However, the $O(n^4)$ running time of this algorithm becomes unpractical as the input grows; so, in order to improve its running time, we modified this algorithm obtaining a competitive and more efficient $O(n^2 \log n)$ heuristic. All the tests were made over the most well known benchmarks for the problem, i.e. the *pmed* instances from OR-Lib (Beasly 1990) and some instances from TSP-Lib (Reinelt 1991).

The remainder of this paper is organized as follows. In Sect. 2 we describe our 3-approximation algorithm for the vertex k-center problem. In Sect. 3 we introduce a modification to this algorithm that produces a more efficient heuristic. In Sect. 4 we show that the proposed 3-approximation algorithm is correct. In Sect. 5 we show the result of applying the 3-approximation algorithm and its heuristic version to the most well known benchmarks for the vertex k-center problem, and how it compares with the most successful polynomial algorithms. Lastly, Sect. 6 presents our concluding remarks.

2 A 3-approximation algorithm

In this section we present what we have called the *Critical Dominating Set* procedure (Fig. 1), which serves as the core procedure for the proposed 3-approximation algorithm and its heuristic versions. As shown by Lemma 1, if the *Critical Dominating Set* procedure receives as input a covering radius $r \leq r^*$, then it returns a solution of size smaller than or equal to $3r^*$, where r^* is the optimal size for the vertex k-center problem. Since the costs of the edges in the input graph are all feasible covering radius, one of them being the optimal one, we can just run the *Critical Dominating Set* procedure with all these values as input, and by returning the best found solution we get a 3-approximation algorithm.

The aim of the *Critical Dominating Set* procedure is to generate a Dominating Set of size k for the pruned input graph G_r, which is a copy of the input graph G where all the edges of cost greater than r are removed. This procedure works greedily by

```
Input: An undirected graph G = (V, E), an integer k, a covering radius r
Output: A set of vertices C ⊆ V, |C| = k
// By N(v) we mean the neighbors of v in G_r ;
C ← ∅ ;
G_r ← BottleneckGraph(G, r) ;
D ← ∅ ;
foreach v ∈ V do
  | Score[v] ← GetDegree(v, G_r) ;
end
for i = 1 to k do
  | f_i ← A vertex v ∈ V that maximizes distance(v, C) ;
  | v_{f_i} ← A vertex v ∈ N(f_i) ∪ f_i of maximum Score ;
  | S ← (N(v_{f_i}) ∪ v_{f_i}) \ D ;
  | foreach v ∈ S do
  |   | foreach u ∈ N(v) do
  |   |   | Score[u] ← Score[u] − 1
  |   | end
  | end
  | D ← D ∪ S ;
  | C ← C ∪ {v_{f_i}} ;
end
return C ;
```

Fig. 1 The *Critical Dominating Set* heuristic

selecting as center a vertex that belongs to a set of promising vertices, named *critical neighbors*, and that has a maximal $Score$. For every vertex $v \in V$, the $Score$ list stores the number of vertices, non-dominated by the current partial solution C, that would be dominated by the solution if vertex v were added to it. A *critical neighbor* is a vertex located within the one-hop neighborhood of the *critical vertex* f_i in G_r (f_i included). A *critical* vertex, as defined by Mladenovič et al. (2000), is the vertex whose distance from any given solution is maximal, and hence, the vertex that defines the size of any given solution. For the case of the *Critical Dominating Set* procedure, the critical vertex corresponds to the farthest vertex from every current partial solution C. Therefore, by selecting as center a vertex within the one-hop neighborhood of the critical vertex ($v_{f_i} \in N(f_i) \cup f_i$), the algorithm is more likely to reduce the size of the current partial solution. The purpose of this greedy strategy is to simultaneously make progress from the perspective of the vertex k-center selection problem by reducing the size $r(C)$ of the current partial solution, and from the perspective of the dominating set problem by reducing the number of non-dominated vertices. Since C is the empty set at the first iteration ($i = 1$), the vertex f_i is selected at random.

By running the *Critical Dominating Set* procedure over the up to $(n^2-n)/2$ feasible covering radius, we get a 3-approximation algorithm that we called CDS. According to our experiments, the CDS algorithm outperforms all the previously known 2-approximation algorithms and heuristics with polynomial running time.

Because of its $O(n^4)$ complexity, the CDS algorithm becomes unpractical as the input size grows. So, in order to reduce its time complexity, we can perform a binary search over the set of feasible covering radius. This way we still have a very competitive heuristic algorithm but with a significantly smaller time complexity. This is detailed in the following section.

```
Input: An undirected graph G = (V, E), an integer k, and an ordered list of the
       weight of the m edges of G: w(e_1), w(e_2), ..., w(e_m) where w(e_i) ≤ w(e_{i+1})
Output: A set of vertices C ⊆ V, |C| = k
high ← m ;
low ← 1 ;
while high − low > 1 do
    mid ← ⌈(high + low)/2⌉ ;
    C' ← CriticalDominatingSet(G, k, w(e_mid)) ;
    C ← BestSolution(C, C') ;
    r(C) ← BestSize(r(C), r(C')) ;
    if r(C) ≤ w(e_mid) then
        high ← mid ;
    else
        low ← mid ;
    end
end
return C;
```

Fig. 2 A $O(n^2 \log n)$ heuristic for the vertex k-center problem

3 An efficient heuristic

Instead of trying with all the up to $(n^2 - n)/2$ feasible covering radius, the heuristic algorithm presented in this section is based on the binary search described at Fig. 2 that assumes that the covering radius obtained by the *Critical Dominating Set* procedure (Fig. 1) is the best covering radius achievable for a given input pruned graph. This way, if the covering radius $r(C)$ obtained by the *Critical Dominating Set* procedure is smaller than or equal to the current pruning parameter $w(e_{mid})$, then it concludes that the optimum is in fact smaller than or equal to $w(e_{mid})$ and continues the search on the interval $[w(e_{low}), w(e_{mid})]$. Otherwise, it concludes that the optimum is larger than $w(e_{mid})$ and hence that it is located on the interval $(w(e_{mid}), w(e_{high})]$. Although this search does not give any approximation guarantee because the *Critical Dominating Set* procedure does not necessarily returns a Dominating Set, it has a better performance than most of the approximation algorithms and polynomial heuristics on most of the well known benchmarks and at considerable lower time cost.

4 Analysis

In this section we present a theorem that shows the correctness of the proposed 3-approximation algorithm. Definition 1 presents the notation used in the proofs.

Definition 1

$$
\begin{aligned}
C^* &= \text{an optimal solution} \\
r(C^*) &= \text{the size of } C^* \\
C_i &= \text{a partial solution with only } i \text{ elements} \\
f_i &= \text{the farthest vertex from a partial solution } C_{i-1} \\
c^*_{f_i} &= \text{the closest center } c \in C^* \text{ to vertex } f_i \\
N(f_i) &= \text{the set of vertices connected to } f_i \ (critical\ neighbors)
\end{aligned}
$$

Lemma 1 *If r is smaller than or equal to the optimal size $r^* = r(C^*)$, then the Critical Dominating Set procedure returns a solution C of covering radius $r(C) \leq 3r^*$.*

Proof First, if during any of the k iterations of the *Critical Dominating Set* procedure, the farthest vertex is at a distance less than or equal to $3r^*$, then all vertices are at a distance less than or equal to $3r^*$ from the current solution and hence, the algorithm returns a solution with covering radius $r(C) \leq 3r^*$.

Now, for the rest of the proof we will assume that during all the iterations of the *Critical Dominating Set* procedure, the farthest vertex is at a distance larger than $3r^*$ from the current solution. The overall description of this part of the proof is as follows. We have to show that the center c_i selected by the *Critical Dominating Set* procedure during the ith iteration covers within a radius of $3r^*$, all the vertices covered by the vertex $c^*_{f_i}$ in the optimal solution C^*. Then, we show that the centers $c^*_{f_i}$ considered during the k iterations of the algorithm are different, consequently, that all the k centers of the optimal solution have a vertex in the generated solution that is responsible of covering their vertices within a $3r^*$ radius and therefore, that the *Critical Dominating Set* procedure returns a solution with covering radius $r(C) \leq 3r^*$.

First, we show that the center c_i selected at iteration i, is at distance less than or equal to $2r^*$ from the center $c^*_{f_i} \in C^*$ which is the closest to f_i. The distance $d(f_i, c^*_{f_i})$ is less than or equal to r^* because $c^*_{f_i}$ is the closest center in the optimal solution C^* to f_i, and the distance $d(f_i, c_i)$ is less than or equal to r^* because c_i is in the critical neighborhood of f_i ($c_i \in N(f_i) \cup f_i$), therefore, by using the triangle inequality we get

$$d(c_i, c^*_{f_i}) \leq d(f_i, c_i) + d(f_i, c^*_{f_i}) \leq 2r^* \quad (1)$$

Using the triangle inequality again, we see that the distance between any vertex v at distance less than r^* from the center $c^*_{f_i}$ is at a distance less than $3r^*$ from the center c_i selected by the algorithm.

$$\forall v \in V : d(v, c^*_{f_i}) \leq r^*, \quad d(c_i, v) \leq d(c^*_{f_i}, c_i) + d(v, c^*_{f_i}) \leq 3r^* \quad (2)$$

Now, we proceed by contradiction to show that all the centers $c^*_{f_i}$ covered during the k iterations of the algorithm are different. Suppose that the center $c^*_{f_i}$ was already covered at an iteration $j < i$, i.e., that $c^*_{f_i} = c^*_{f_j}$. From Eq. (1) we know that $d(c_j, c^*_{f_j}) \leq 2r^*$ and from the assumption that $c^*_{f_i}$ is the closest optimal center to f_i we also have that $d(f_i, c^*_{f_i}) \leq r^*$. Using the triangle inequality and the previous equations we get

$$d(c_j, f_i) \leq d(c_j, c^*_{f_j}) + d(c^*_{f_i}, f_i) \leq 3r^* \quad (3)$$

On the other hand, by assumption we also know that $d(C_{i-1}, f_i) > 3r^*$ and therefore that $d(c_j, f_i) > 3r^*$, contradicting Eq. (3).

Theorem 1 *By running the Critical Dominating Set procedure with all the up to $(n^2 - n)/2$ feasible covering radius we get a 3-approximated solution.*

Proof Since centers are selected from the set of vertices, the size of any solution must be equal to the cost of an edge of the complete undirected input graph. Now, since at

least one of the up to $(n^2-n)/2$ feasible covering radius is smaller than or equal to the optimal covering radius, then, by Lemma 1 the *Critical Dominating Set* will return a 3-approximated solution. □

4.1 Time complexity

The *Critical Dominating Set* procedure (Fig. 1) starts by running the $BottleneckGraph$ subroutine, which removes the edges with cost greater than a given value r from the input graph. Since there are at most $(n^2-n)/2$ edges, the complexity of this subroutine is $O(n^2)$. The $GetDegree$ subroutine computes the degree of every vertex on the pruned graph G_r, which takes $O(n^2)$ steps. At the next iterations, finding f_i (line 9), v_{f_i} (line 10), and the set S (line 11) can be done in n steps. Then, the $Score$ of the neighbors of each vertex $v \in S$ is updated (line 14). The overall complexity of the k repetitions of the cycle that goes from line 12 to 16 is $O(n^2)$, because through the k iterations of the algorithm, each vertex appears only once in some of the k sets S (thanks to line 16, which inserts S into D). Thus, the overall complexity of the whole *Critical Dominating Set* procedure is $O(n^2)$.

By running the *Critical Dominating Set* procedure with the up to $(n^2-n)/2$ feasible covering radius, we get a 3-approximation algorithm of complexity $O(n^4)$. The binary search defined by Fig. 2 is performed over an ordered set of up to $(n^2-n)/2$ values; thus, it performs the *Critical Dominating Set* procedure $O(\log n)$ times, getting an overall complexity of $O(n^2 \log n)$. Notice that, although this last algorithm requires an ordered set of edges at the beginning, that does not affect its complexity, because this ordering can be done in $O(n^2 \log n)$ steps.

5 Experimental results

In this section we present the results of a series of experiments where we evaluate the performance of the proposed $O(n^4)$ 3-approximation algorithm, and its $O(n^2 \log n)$ heuristic version, against that of what we consider a representative sample of approximation algorithms and heuristics for the vertex k-center problem. The experiments were performed over some *de facto* standard benchmark data sets for this problem (Daskin 2000; Mladenovič et al. 2000; Ilhan and Pinar 2001; Elloumi et al. 2004; Mihelič and Robič 2005; Pacheco and Casado 2005; Rana and Garg 2008; Pullan 2008; Davidović et al. 2011; Kaveh and Nasr 2011).

One of the most used data sets for the comparison of the experimental approximation factor of algorithms for the vertex k-center problem is the *pmed* data set from the OR-Lib library (Beasly 1990). The *pmed* data set consists of 40 undirected graphs with a number of vertices ranging from 100 to 900, values of k ranging from 5 to 200 and edge costs that respect a non-euclidean metric. An important characteristic of this data set is that the optimal solutions are known (Daskin 2000; Ilhan and Pinar 2001; Elloumi et al. 2004). The second data set was obtained by selecting 40 relatively small instances from the TSP-Lib library. This set consists of 40 undirected graphs with a number of vertices ranging from 200 to 657, values of k ranging from 5 to 40 and edge costs that respect an euclidean metric. The optimal solutions for the 40 *small* TSP-

Lib instances are not known; however, some exact and metaheuristic algorithms have been tested over them, which provide a reliable set of best-known solutions (Pullan 2008; Ilhan and Pinar 2001). We also performed tests on the instances u1060, u1817, and pcb3038 from TSP-Lib, which respect an euclidean metric, and have 1060, 1817, and 3038 vertices, respectively; the values of k were on the range from 10 to 150 for u1060 and u1817, and on the range from 10 to 500 for pcb3038. The optimal solutions for these instances are not known; however, as in the previous case, some exact and metaheuristic algorithms have been tested over them providing a reliable set of best-known solutions (Mladenović et al. 2000; Ilhan and Pinar 2001; Pullan 2008).

The polynomial algorithms used in these experiments were the *pure greedy* Gr algorithm; the Gr+ algorithm, which is the Gr algorithm repeated n times with a different initial center; the 2-approximation Gon algorithm (Gonzalez 1985); the Gon+ algorithm, which is the Gon algorithm repeated n times with a different initial center; the 2-approximation HS algorithm (Hochbaum and Shmoys 1985); the Scr algorithm (Mihelič and Robič 2005); the proposed $O(n^4)$ approximation algorithm (named CDS), its $O(n^2 \log n)$ heuristic version CDSh (Fig. 2), and the CDSh+ variant, which is the CDSh heuristic with the *Critical Dominating Set* algorithm repeated n times, selecting every time a different vertex as initial center. Even though CDS, CDSh+ and Scr are polynomial-time algorithms, their complexity becomes unpractical on large instances; because of that, we did not test CDS and Scr over the u1060, u1817 and pcb3038 instances, and CDSh+ over the pcb3038 instance. The source code of the algorithms used in these experiments is available from https://github.com/jesgadiaz/vertex-k-center.

Table 1 shows the mean and standard deviation of the experimental approximation factors obtained by the described heuristics and approximation algorithms for the vertex k-center problem when tested over the *pmed* and the *small* TSP-Lib instances. Table 2 shows the same metrics for the u1060, and u1817 instances. Table 3 shows the same metrics for the pcb3038 instance from TSP-Lib. As shown in the tables, in all the tests, the CDS, CDSh, and CDSh+ algorithms outperformed the 2-approximated Gon and HS algorithms.

Table 4 shows the detailed results regarding the size of the solutions found by each tested algorithm over the *pmed* instances. The best results were obtained by the CDS, CDSh, and CDSh+ algorithms, all of them outperforming the Scr algorithm, which is considered one of the best heuristics for the vertex k-center problem with an overall experimental approximation factor of 1.055 and standard deviation of 0.038 over the *pmed* instances from OR-Lib. We would like to highlight the fact that the CDS, CDSh, and CDSh+ algorithms got near optimal solutions in $O(n^4)$, $O(n^2 \log n)$, and $O(n^3 \log n)$ steps, respectively. Of course, the most practical algorithm is the CDSh with a complexity of $O(n^2 \log n)$, an overall experimental approximation factor of 1.035 and standard deviation of 0.031. It is also interesting to see how the CDSh+ algorithm outperforms the 3-approximation CDS algorithm in both, execution time and quality of the generated solutions, getting an overall experimental approximation factor of 1.015 and standard deviation of 0.023, showing how important is the selection of the first center. Table 4 also shows that CDSh, CDS, and CDSh+ found 12, 24, and 26 optimal solutions out of 40.

Table 1 Mean and standard deviation of a representative set of heuristics and approximation algorithms for the vertex k-center problem tested over the *pmed* instances from OR-Lib and over 40 *small* instances from TSP-Lib

Algorithm	μ	σ	Complexity
pmed instances from OR-Lib			
Gr	1.702	0.596	$O(kn^2)$
HS	1.532	0.175	$O(n^2 \log n)$
Gr+	1.512	0.543	$O(kn^3)$
Gon	1.503	0.122	$O(kn)$
Gon+	1.303	0.122	$O(kn^2)$
Scr	1.055	0.038	$O(n^4)$
CDSh	1.035	0.031	$O(n^2 \log n)$
CDS	1.020	0.027	$O(n^4)$
CDSh+	1.015	0.023	$O(n^3 \log n)$
40 *small* instances from TSP-Lib			
Gr	2.306	1.965	$O(kn^2)$
Gr+	1.713	0.607	$O(kn^3)$
Gon	1.396	0.091	$O(kn)$
HS	1.318	0.108	$O(n^2 \log n)$
Gon+	1.223	0.068	$O(kn^2)$
CDSh	1.124	0.065	$O(n^2 \log n)$
Scr	1.078	0.050	$O(n^4)$
CDS	1.042	0.038	$O(n^4)$
CDSh+	1.035	0.039	$O(n^3 \log n)$

Table 2 Mean and standard deviation of a representative set of heuristics and approximation algorithms for the vertex k-center problem tested over the u1060 and u1817 instances from TSP-Lib

Algorithm	μ	σ	Complexity
u1060 instance			
Gr	3.366	0.521	$O(kn^2)$
HS	1.420	0.055	$O(n^2 \log n)$
Gon	1.342	0.095	$O(kn^3)$
Gon+	1.233	0.041	$O(kn^2)$
CDSh	1.161	0.055	$O(n^2 \log n)$
CDSh+	1.089	0.026	$O(n^3 \log n)$
u1817 instance			
Gr	3.599	0.473	$O(kn^2)$
HS	1.394	0.064	$O(n^2 \log n)$
Gon	1.356	0.064	$O(kn)$
Gon+	1.234	0.044	$O(kn^2)$
CDSh	1.178	0.059	$O(n^2 \log n)$
CDSh+	1.119	0.061	$O(n^3 \log n)$

Table 3 Mean and standard deviation of a representative set of heuristics and approximation algorithms for the vertex k-center problem tested over the pcb3038 instance from TSP-Lib

Algorithm	μ	σ	Complexity
pcb3038 instance			
Gr	4.566	1.539	$O(kn^2)$
HS	1.442	0.100	$O(n^2 \log n)$
Gon	1.327	0.074	$O(kn)$
Gon+	1.236	0.026	$O(kn^2)$
CDSh	1.176	0.037	$O(n^2 \log n)$

Table 4 Results obtained by each tested algorithm over the *pmed* instances from OR-Lib

Instance	n	k	OPT	Gr	Gr+	Gon	Gon+	HS	Scr	CDSh	CDS	CDSh+
pmed1	100	5	127	137	133	193	155	173	133	129	**127**	**127**
pmed2	100	10	98	117	110	166	117	130	109	102	102	**100**
pmed3	100	10	93	118	106	141	124	142	99	98	96	**95**
pmed4	100	20	74	127	92	98	92	114	83	79	79	**78**
pmed5	100	33	48	100	78	70	62	76	**48**	**48**	**48**	**48**
pmed6	200	5	84	101	89	110	98	112	90	85	**84**	**84**
pmed7	200	10	64	80	77	104	85	97	70	67	**66**	**66**
pmed8	200	20	55	73	72	82	71	73	60	60	57	**56**
pmed9	200	40	37	73	63	57	49	59	39	38	**37**	**37**
pmed10	200	67	20	56	38	31	29	33	**20**	**20**	**20**	**20**
pmed11	300	5	59	68	61	92	68	74	61	60	**59**	**59**
pmed12	300	10	51	77	56	77	66	77	53	52	52	**51**
pmed13	300	30	35	64	52	59	49	57	38	**37**	**37**	**37**
pmed14	300	60	26	59	46	39	36	42	27	27	**26**	**26**
pmed15	300	100	18	42	40	24	23	31	**18**	**18**	**18**	**18**
pmed16	400	5	47	52	47	68	52	63	48	**47**	**47**	**47**
pmed17	400	10	39	49	43	56	48	55	41	**39**	**39**	**39**
pmed18	400	40	28	50	42	40	39	45	31	30	**29**	**29**
pmed19	400	80	18	36	31	28	27	30	20	**19**	**19**	**19**
pmed20	400	133	13	32	32	19	17	22	14	14	**13**	**13**
pmed21	500	5	40	48	42	52	45	53	**40**	**40**	**40**	**40**
pmed22	500	10	38	47	43	56	47	54	40	**39**	**39**	**39**
pmed23	500	50	22	38	35	34	32	35	24	**23**	**23**	**23**
pmed24	500	100	15	42	32	23	21	26	16	16	16	**15**
pmed25	500	167	11	27	27	15	15	18	**11**	12	**11**	**11**
pmed26	600	5	38	45	39	51	43	50	41	39	**38**	**38**
pmed27	600	10	32	39	35	42	38	44	33	**32**	**32**	**32**
pmed28	600	60	18	33	27	29	25	28	19	19	**18**	**18**
pmed29	600	120	13	36	34	21	18	22	**13**	**13**	**13**	**13**
pmed30	600	200	9	29	29	14	13	18	10	10	**9**	**9**

Table 4 continued

Instance	n	k	OPT	Gr	Gr+	Gon	Gon+	HS	Scr	CDSh	CDS	CDSh+
pmed31	700	5	30	33	31	45	36	39	31	**30**	**30**	**30**
pmed32	700	10	29	35	32	43	37	44	31	30	**29**	**29**
pmed33	700	70	15	26	24	25	23	26	17	**16**	**16**	**16**
pmed34	700	140	11	30	27	17	16	20	**11**	**11**	**11**	**11**
pmed35	800	5	30	31	31	37	34	38	31	**30**	**30**	**30**
pmed36	800	10	27	34	30	49	34	38	**28**	**28**	**28**	**28**
pmed37	800	80	15	26	26	24	23	25	**16**	**16**	**16**	**16**
pmed38	900	5	29	33	31	43	31	36	**29**	**29**	**29**	**29**
pmed39	900	10	23	28	25	37	28	35	24	24	**23**	**23**
pmed40	900	90	13	22	22	21	19	21	**14**	**14**	**14**	**14**

The best found solutions are highlighted, and the optimal solutions are underlined

Table 5 shows the detailed results regarding the size of the solutions found by each tested algorithm over the *small* TSP-Lib instances. For this set, the CDSh algorithm was surpassed in quality of the generated solutions by the Scr algorithm. However, this difference was only of 4.6%, and the lower complexity of CDSh still makes it a good option. The best results were found by CDS and CDSh+. As well as in the other tests, the CDSh+ algorithm outperformed the CDS algorithm in both, execution time and quality of the generated solutions, getting an overall experimental approximation factor of 1.035 and standard deviation of 0.039. For this set of instances, CDSh, CDS, and CDSh+ found 1, 9, and 12 best-known solutions out of 40.

Table 6 shows the detailed results regarding the size of the solutions found by each tested algorithm over the u1060 instance. For this instance, the CDS algorithm and the Scr algorithm were not tested, because their complexity of $O(n^4)$ becomes unpractical with the number of vertices. The best results for these tests were obtained by CDSh and CDSh+, with an overall average experimental approximation factor of 1.161, and 1.089, respectively, and standard deviation of 0.055, and 0.026, respectively.

Table 7 shows the detailed results regarding the size of the solutions found by each tested algorithm over the u1817 instance. For this instance, the CDS algorithm and the Scr algorithm were not tested, because their running time becomes unpractical. The best results for these tests were obtained by CDSh and CDSh+, with an overall average experimental approximation factor of 1.178, and 1.119, respectively, and standard deviation of 0.059, and 0.061, respectively.

Lastly, Table 8 shows the detailed results regarding the size of the solutions found by each tested algorithm over the pcb3038 instance. For this instance, the CDS, Scr and CDSh+ algorithms were not tested, because their running time becomes unpractical. The CDSh algorithm had the best results with an overall average experimental approximation factor of 1.176 and a standard deviation of 0.037.

Table 5 Results obtained by each tested algorithm over 40 *small* TSP-Lib instances

Instance	n	k	BKS	Gr	Gr+	Gon	Gon+	HS	Scr	CDSh	CDS	CDSh+
kroA200	200	5	911.41	1130.73	1011.22	1353.13	1108.84	1078.00	1066.82	1070.63	**911.41**	**911.41**
kroA200	200	10	598.81	860.46	772.81	826.00	710.29	738.44	621.34	621.34	**599.47**	**599.47**
kroA200	200	20	389.30	595.76	568.68	512.01	466.00	515.19	416.55	427.98	423.21	**400.22**
kroA200	200	40	258.25	415.54	330.13	344.22	305.63	348.10	269.42	280.59	269.43	**262.61**
gr202	202	5	19.38	25.04	**19.38**	24.43	20.96	24.06	20.96	20.97	20.09	**19.38**
gr202	202	10	9.33	22.25	14.98	12.42	11.17	11.92	**9.83**	10.37	10.02	10.02
gr202	202	20	5.56	15.04	8.35	6.91	6.44	7.60	**5.66**	5.76	5.67	5.67
gr202	202	40	2.97	5.09	4.82	4.10	3.53	4.65	**3.11**	3.26	3.19	3.14
pr226	226	5	3720.55	5570.68	4459.26	5433.69	4601.08	4351.14	4103.65	3736.64	**3720.55**	**3720.55**
pr226	226	10	2326.48	4601.08	3434.02	3640.05	3070.01	3387.10	2439.77	2750.45	**2326.48**	**2326.48**
pr226	226	20	1365.65	3950.00	2820.01	1950.00	1703.85	1796.69	**1365.65**	1500	**1365.65**	**1365.65**
pr226	226	40	650	3434.02	2200.00	850.00	707.10	850.36	670.82	670.82	670.82	**650**
pr264	264	5	1610.12	2371.70	2071.83	2005.61	1802.77	1792.34	**1610.12**	**1610.12**	**1610.12**	**1610.12**
pr264	264	10	850	1552.41	1408.90	1350.92	1217.57	1166.19	943.39	939.41	**884.59**	**884.59**
pr264	264	20	514.78	1372.95	1372.95	728.01	707.10	696.41	550.00	538.52	538.52	**514.78**
pr264	264	40	316.22	1372.95	1297.11	424.26	360.55	509.90	360.55	360.56	**335.41**	**335.41**
pr299	299	5	1336.27	2070.17	1750.71	2042.94	1700.18	1650.18	1470.75	1456.24	1382.93	**1350.93**
pr299	299	10	888.83	1732.95	1347.68	1240.25	1033.46	1093.52	980.11	960.47	915.77	**894.43**
pr299	299	20	559.01	1212.10	1044.03	813.94	691.46	822.34	**602.07**	654.31	**602.07**	**602.07**
pr299	299	40	355.31	880.69	710.63	485.41	436.60	521.72	410.03	403.11	400.78	**400**

Table 5 continued

Instance	n	k	BKS	Gr	Gr+	Gon	Gon+	HS	Scr	CDSh	CDS	CDSh+
lin318	318	5	1101.33	1673.16	1403.98	1647.42	1371.83	1433.02	1182.75	1301.15	1129.23	**1101.33**
lin318	318	10	743.21	1454.68	1056.87	1050.16	906.53	880.30	804.87	804.87	755.17	**752.55**
lin318	318	20	496.45	1343.25	794.65	682.92	595.62	679.46	509.61	545.60	509.61	**506.75**
lin318	318	40	315.91	1343.25	622.46	422.40	388.49	450.96	**315.91**	341.23	328.95	326.79
pr439	439	5	3196.58	5692.64	3745.16	4153.46	3610.57	4069.70	3250.00	3266.97	**3220.63**	**3220.63**
pr439	439	10	1971.83	4731.60	2855.80	2780.51	2432.33	2500.00	**1971.83**	2291.96	**1971.83**	**1971.83**
pr439	439	20	1185.59	3506.15	2381.83	1746.78	1530.52	1677.98	1305.03	1360.14	1200.26	**1191.89**
pr439	439	40	671.75	1604.87	1375.00	941.07	850.00	976.28	735.69	772.17	**715.89**	**715.89**
pcb442	442	5	1024.74	1802.77	1328.53	1442.22	1252.99	1204.15	1081.66	1140.17	**1024.74**	1063.01
pcb442	442	10	670.82	1676.30	1044.03	1000.00	894.42	886.00	781.02	806.22	710.63	**707.10**
pcb442	442	20	447.21	1581.13	1000.00	640.31	565.68	588.98	500.09	538.51	**500**	**500**
pcb442	442	40	316.22	1581.13	806.22	400.00	372.02	430.11	336.00	360.55	**316.22**	326.95
d493	493	5	752.90	1133.79	887.92	1200.67	931.56	921.45	883.83	902.58	783.80	**752.90**
d493	493	10	458.30	816.36	573.07	669.79	580.47	580.47	493.10	548.61	**488.29**	489.60
d493	493	20	312.74	628.53	441.81	408.18	367.93	384.37	**328.15**	368.35	328.93	330.22
d493	493	40	206.02	457.20	355.65	264.04	243.03	259.11	217.01	230.86	218.22	**215.90**
d657	657	5	880.90	1146.17	1146.17	1331.28	1110.72	1083.13	966.84	1115.83	**880.90**	**880.90**
d657	657	10	574.74	1060.47	945.16	795.82	725.72	740.84	650.29	686.27	**601.74**	609.86
d657	657	20	374.70	829.30	735.15	516.90	464.27	513.05	440.74	447.59	420.44	**415.59**
d657	657	40	249.52	648.19	522.54	329.31	309.58	337.44	283.54	288.63	**278.36**	**278.36**

The best found solutions are highlighted, and the best-known solutions are underlined

Table 6 Results obtained by each tested algorithm over the u1060 instance from TSP-Lib

Instance	n	k	BKS	Gr	Gon	Gon+	HS	CDSh	CDSh+
u1060	1060	10	2273.08	4951.20	3750.42	3046.01	3257.34	2982.14	**2442.06**
u1060	1060	20	1580.8	4547.51	2196.36	1886.27	2108.89	1773.27	**1698.71**
u1060	1060	30	1207.77	4095.07	1679.96	1511.28	1615.16	1491.51	**1333.32**
u1060	1060	40	1020.56	3706.13	1415.31	1251.99	1458.88	1205.15	**1133.45**
u1060	1060	50	904.92	3425.46	1215.75	1070.32	1259.90	1050.75	**990.02**
u1060	1060	60	781.17	3085.34	1069.93	985.41	1097.39	922.60	**894.92**
u1060	1060	70	710.76	2817.88	921.36	900.72	999.84	851.39	**765.32**
u1060	1060	80	652.16	2552.47	825.19	806.57	900.80	738.58	**710.77**
u1060	1060	90	607.88	2315.64	800.7	751.13	863.80	706.63	**671.13**
u1060	1060	100	570.01	2056.85	738.58	706.62	863.76	652.20	**632.86**
u1060	1060	110	538.84	1842.74	700.6	651.47	780.89	603.88	**582.89**
u1060	1060	120	510.28	1650.33	667.54	632.12	710.56	570.44	**552.70**
u1060	1060	130	499.65	1441.91	632.79	582.91	706.63	551.89	**514.59**
u1060	1060	140	452.46	1352.02	583.33	565.77	696.06	510.29	**500.13**
u1060	1060	150	447.01	1275.70	570.01	538.83	652.15	500.11	**474.68**

The best found solutions are highlighted

Table 7 Results obtained by each tested algorithm over the u1817 instance from TSP-Lib

Instance	n	k	BKS	Gr	Gon	Gon+	HS	CDSh	CDSh+
u1817	1817	10	457.91	1417.40	670.56	548.31	595.16	538.81	**471.44**
u1817	1817	20	309.01	1099.25	459.99	359.20	395.12	377.60	**341.96**
u1817	1817	30	240.99	980.12	330.20	296.22	338.86	302.93	**283.97**
u1817	1817	40	209.46	853.44	283.98	255.27	279.42	250.47	**239.61**
u1817	1817	50	184.91	764.64	251.44	228.62	255.26	218.48	**204.79**
u1817	1817	60	162.65	676.44	228.94	204.76	227.18	198.38	**193.43**
u1817	1817	70	148.11	617.99	198.37	184.90	215.52	183.16	**177.80**
u1817	1817	80	136.8	527.07	179.62	163.32	198.37	162.64	**152.40**
u1817	1817	90	129.54	471.25	170.38	160.65	179.59	148.10	**148.09**
u1817	1817	100	127.01	438.46	162.61	152.38	167.03	143.66	**136.77**
u1817	1817	110	109.25	398.79	152.41	148.09	162.63	136.77	**129.50**
u1817	1817	120	107.78	359.19	148.10	136.77	154.50	127.01	**126.99**
u1817	1817	130	107.75	323.78	136.79	129.50	148.09	126.99	**109.24**
u1817	1817	140	101.61	305.85	136.77	127.01	148.09	107.78	**107.76**
u1817	1817	150	101.59	283.98	129.50	126.99	148.09	107.76	**107.74**

The best found solutions are highlighted

Table 8 Results obtained by each tested algorithm over the pcb3038 instance from TSP-Lib

Instance	n	k	BKS	Gr	Gon	Gon+	HS	CDSh
pcb3038	3038	10	728.54	1461.20	1082.57	890.99	962.95	**877.70**
pcb3038	3038	20	493.04	1391.32	698.22	594.75	651.77	**582.03**
pcb3038	3038	30	393.5	1348.39	525.75	502.31	534.35	**494.41**
pcb3038	3038	40	336.42	1276.27	474.04	426.00	463.00	**405.26**
pcb3038	3038	50	298.2	1217.16	415.39	366.00	397.50	**357.41**
pcb3038	3038	100	206.63	714.19	273.74	258.62	286.89	**238.72**
pcb3038	3038	150	164.77	609.27	213.39	202.82	232.38	**195.47**
pcb3038	3038	200	140.9	609.27	184.45	174.04	198.36	**166.78**
pcb3038	3038	250	122.78	609.27	162.63	155.41	186.07	**146.59**
pcb3038	3038	300	115.25	609.27	145.67	139.28	172.19	**127.00**
pcb3038	3038	350	104.81	609.27	128.97	124.91	158.00	**119.92**
pcb3038	3038	400	97.51	609.27	120.93	118.82	153.95	**113.44**
pcb3038	3038	450	88.96	609.27	115.03	113.53	142.88	**103.22**
pcb3038	3038	500	85.00	609.27	107.56	104.92	133.46	**96.26**

The best found solutions are highlighted

6 Conclusions

In this paper we presented an algorithm, named CDS, with non-optimal approximation factor that achieves better practical results than the known optimal approximation factor algorithms for the vertex k-center selection problem. At first, this may seem counterintuitive, however we argued that the optimal approximation algorithms proposed so far for the vertex k-center selection problem make conservative decisions that can affect the quality of the generated solutions in order to keep the approximation guarantees. In the case of the vertex k-center selection problem, the optimal approximation algorithms always chose the next center as one that is at a distance at least two times the optimal covering radius ($2r$) from the current solution even though there may be options for centers that would cover many vertices but at distance slightly smaller than $2r$. Unlike those algorithms, our CDS algorithm tries to find good partial solutions that are close to the farthest vertex to the current solution. This hurts the approximation factor, but gives better results in practice. Since the $O(n^4)$ complexity of CDS becomes unpractical as the input grows, we also defined the CDSh heuristic version of CDS, which has a more practical complexity of $O(n^2 \log n)$ and in many cases outperforms the most expensive CDS algorithm. From CDSh, we also defined the CDSh+ heuristic, which is the CDSh heuristic repeated n times, selecting every time a different vertex as initial center. All of our proposed algorithms follow a greedy approach by selecting the centers of the solution through a hierarchical optimization of two objective functions: covering radius and number of dominated vertices.

We observed through extensive experimentation that CDSh+ outperforms CDS, even though CDSh does not give any approximation guarantee. Such behavior agrees with our hypothesis that taking conservative decisions can affect the quality of the generated solutions in order to keep the approximation guarantees.

We believe the same idea can be applied to other NP-Hard optimization problems, thus, even if there are known optimal approximation algorithms for a problem there is still room for improvement. Perhaps these new algorithms will be the result of combining approximation algorithms and heuristics. In fact, the CDS, CDSh, and CDSh+ algorithms are examples of these intuitions because they exploit different ideas taken from the previous work on the vertex k-center problem, such as parametric pruning (Hochbaum and Shmoys 1985), a farthest-point heuristic (Gonzalez 1985; Dyer and Frieze 1985), and the relationship between the vertex k-center problem and the Dominating Set problem (Mihelič and Robič 2005).

Lastly, it is important to remark that the proposed CDS, CDSh, and CDSh+ algorithms outperformed not only the previously known 2-approximation Gon and HS algorithms, but also the Scr algorithm, which is considered one of the best polynomial heuristics for solving the vertex k-center problem (Mihelič and Robič 2005; Rana and Garg 2008).

Acknowledgements This work was sponsored in part by the University of California MEXUS—CONACyT program under Grant CN 15-1451, by the Mexican National Council for Science and Technology (CONACyT) and by the Mexican National Polytechnic Institute (IPN).

References

Beasly, J.E.: OR-Library: distributing test problems by electronic mail. J. Oper. Res. Soc. **41**(11), 1069–1072 (1990)

Daskin, M.: A new approach to solve the vertex p-center problem to optimality: algorithm and computational results. Commun. Oper. Res. Soc. Jpn. **45**(9), 428–436 (2000)

Davidović, T., Ramljak, D., Šelmić, M., Teodorović, D.: Bee colony optimization for the p-center problem. Comput. Oper. Res. **38**, 1367–1376 (2011)

Dyer, M.E., Frieze, A.M.: A simple heuristic for the p-centre problem. Oper. Res. Lett. **3**(6), 285–288 (1985)

Elloumi, S., Labbé, M., Pochet, Y.: A new formulation and resolution method for the p-center problem. INFORMS J. Comput. **16**, 84–94 (2004)

Garcia, J., Menchaca, R., Menchaca, R., Quintero, R.: A structure-driven randomized algorithm for the K-center problem. IEEE Latin America Trans. **13**(3), 746–752 (2015)

Garey, M., Johnson, D.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., New York (1979)

Gonzalez, T.: Clustering to minimize the maximum inter-cluster distance. Theor. Comput. Sci. **38**, 293–306 (1985)

Hakimi, S.: Optimum location of switching centers and the absolute centers and medians of a graph. Oper. Res. **12**, 450–459 (1964)

Hochbaum, D.: Approximation Algorithms for NP-Hard Problems. PWS Publishing Co., Boston (1997)

Hochbaum, D., Shmoys, D.B.: A best possible heuristic for the k-center problem. Math. Oper. Res. **10**, 180–184 (1985)

Ilhan, T., Pinar, M.Ç.: An efficient exact algorithm for the vertex p-center problem. Optimization online. http://www.optimization-online.org/DB_HTML/2001/09/376.html. Accessed 14 July 2015

Kariv, O., Hakimi, S.L.: An algorithmic approach to network location problems, I: the p-centers. J. Appl. Math. **37**(3), 513–538 (1979)

Kaveh, A., Nasr, H.: Solving the conditional and unconditional p-center problem with modified harmony search: a real case study. Sci. Iran. A **18**(4), 867–877 (2011)

Mihelič, J., Robič, B.: Solving the k-center problem efficiently with a dominating set algorithm. J. Comput. Inf. Technol. **13**(3), 225–234 (2005)

Minieka, E.: The m-center problem. SIAM Rev. **12**, 138–139 (1970)

Mladenovič, N., Labbé, M., Hansen, P.: Solving the p-center problem with Tabu search and variable neighborhood sSearch. Networks **42**, 48–64 (2000)

Pacheco, J.A., Casado, S.: Solving two location models with few facilities by using a hybrid heuristic: a real health resources case. Comput. Oper. Res. **32**(12), 3075–3091 (2005)

Pullan, W.: A memetic genetic algorithm for the vertex p-center problem. Evol. Comput. **16**(3), 417–436 (2008)

Rana, R., Garg, D.: The analytical study of k-center problem solving techniques. Int. J. Inf. Technol. Knowl. Manag. **1**(2), 527–535 (2008)

Reinelt, G.: TSPLIB-a traveling salesman problem library. ORSA J. Comput. **3**, 376–384 (1991)

Shmoys D.B.: Computing near-optimal solutions to combinatorial optimization problems. Technical report, Ithaca, NY 14853 (1995)

Vazirani, V.V.: Approximation Algorithms. Springer, New York (2001)