**EVALUACIÓN DEL DESEMPEÑO EN ALGORITMOS DE RECONFIGURACIÓN DE REDES VIRTUALES INCRUSTADAS**

Edwin Blandón Arias, Jhan Carlos Larrahondo, Néstor Álzate Mejía

[edwin.blandona@campusucc.edu.co](mailto:edwin.blandona@campusucc.edu.co), [jhan.larrahondol@campusucc.edu.co](mailto:jhan.larrahondol@campusucc.edu.co), [nestor.alzatem@campusucc.edu.co](mailto:nestor.alzatem@campusucc.edu.co)

Universidad Cooperativa de Colombia

Santiago de Cali, Colombia.

**Abstract**

En los últimos 10 años crecimiento del internet a demostrado una necesidad latente por una infraestructura de mejor calidad y con costos reducidos para la industria. Una de las tecnologías que apunta a solucionar parte del problema es la virtualización de redes, esta permite que múltiples redes heterogéneas cohabiten en una red física.

Este artículo se centra en realizar la comparación de algoritmos VNE que realizan reasignaciones en Redes Virtuales (VN), las comparaciones se llevaron a cabo realizando simulaciones de peticiones de red por medio de tres algoritmos los cuales, reciben ciertos parámetros de ejecución y realizan las simulaciones generando como resultado unas métricas las cuales son usadas para evidenciar cuales pueden tener los mejores resultados. Estas evaluaciones surgen de la necesidad de desarrollar nuevos algoritmos que se desempeñen de forma óptima en el uso de recursos de la red sustrato cuando en ella se mapean las solicitudes de VN.

El proceso de investigación realizado comprendió validar la documentación existente en las revistas especializadas, consultas de artículos científicos sobre pruebas de desempeño e investigación de nuevos algoritmos de incrustamiento.

**Introducción**

Con la evaluación de la [1] optimización en el uso de los recursos de las redes sustrato, mediante la simulación de los algoritmos propuestos, esta investigación proporcionará los resultados que sirvan de base para la evaluación de nuevos algoritmos y su implementación.

El resto de este documento tendrá el siguiente orden de temas: La Sección II hace referencia a la explicación de los algoritmos VNE con reconfiguración, esta sección contiene III sub-secciones en las cuales se explican los algoritos: BNPA-VNE, PSO y GENÉTICO. en el sección III se realiza un análisis de los algoritmos. En la Sección IV se presenta la metodología de la simulación de los algoritmos. En la Sección V se muestran los resultados de las simulaciones realizadas. En la sección VI se encuentra la discusión de los resultados y En la Sección VII las conclusiones de la investigación.

**Algoritmos VNE con reconfiguración**

Los algoritmos VNE con reconfiguración se encargan de realizar el procesos incrustamiento de los recursos de la red sustrato (SN) a las redes virtuales (VN)[ 8, 1,11, 6, 4, 7], su particularidad consiste en que evalúan el estado actual de la SN para definir si puede soportar la solicitud de recursos que se está recibiendo, en caso de no contar con disponibilidad de recursos, el algortimo de reconfiguración libera recursos para que puedan ser asignados a la petición entrante.

Los Nodos son un punto de intersección o unión de varios elementos que confluyen en el mismo lugar, en este caso en las redes fisicas y virtuales. De igual forma cohabitan con los enlaces de red que cumplen la responsabilidad de transferir datos desde un nodo a otro adyacente.

Este tipo de algoritmos usa una combinación de activo y pasivo de elementos de red (nodos de red y enlaces de red), esto ha tenido un gran avance en la división y migración de rutas, pero uno de los principales desafíos para este algoritmo es la correcta y eficiente asignación de recursos del sustrato en las redes virtuales [5,9], con lo cual en la ejecución de las asignaciones se presenta un problema conocido como Virtual Network Embedded (VNE).

En las siguientes subsecciones, se explicarán los algoritmos usados en las simulaciones realizadas:

**BNPA-VNE:**

El algoritmo BNPA-VNE fue desarrollado en MATLAB, para realizar las ejecuciones de simulación este algoritmo requiere que sean digitados datos de entrada, los cuales sirven de parámetros en el procesamiento del algoritmo el cual simula el incrustamiento en una red virtual realizando el proceso de Backtracking [2], este proceso se encarga de recibir la petición de recursos de red, validar los recursos existentes y hacer una reconfiguración de los nodos y enlaces (recursos) con el fin de que se puedan ajustar a la petición de recursos hechas inicialmente .

El funcionamiento del proceso de reconfiguración consiste en que el algoritmo analiza los recursos solicitados en la petición actual y evalúa la disponibilidad existente en la red si estas no son suficientes, el algoritmo toma las últimas tres asignaciones incrustadas y las reorganiza de manera tal que logra liberar recursos que pueden ser usados para la petición que se está procesando.

BNPA-VNE fue desarrollado tomando como base El New Paths Algebra - Virtual Network Embedding (NPA-VNE), este es un algoritmo que realiza los Mapeos de Nodos Virtuales (VnoM) y Mapeos de Enlaces Virtuales (VliM) de manera coordinada. Para la fase del VnoM, \cite{Andreas,PhuongNga} este algoritmo realiza una clasificación a partir de pares de nodos bidireccionales de la red SN ordenados por recursos disponibles en cada momento. En la fase del VliM hace uso del algoritmo Paths Algebra.

Este algoritmo evalúa la disponibilidad de un camino entre nodos y de acuerdo a la evaluación se indica si existen recursos disponibles en dicha ruta.

**PSO:**

Particle Swarm Optimization (PSO), algoritmo basado en el comportamiento de ciertas especies animales tales como las aves que vuelan en bandada y peces que nadan en cardúmenes, este algoritmo trata de abstraer la lógica del comportamiento de estas especies y lo implementa a nivel de pseudocódigo. Según las investigaciones realizadas se simula el comportamiento que tienen estas especies al momento de desplazarse y se evalúa como determinan su cercanía, su velocidad, sus movimientos y su reagrupación.

Este algoritmo se encarga de validar la partícula más cercana a la posición actual, lo que hace cierta referencia a la cercanía de partículas vecinas y su relación con el resto de sus componentes los cuales deben estar relacionados entre sí, de esta forma el algoritmo para su optimización hace uso de la relación de cercanías para ejecutar el proceso de asignación de forma más rápida y eficiente.

**GENÉTICO:**

Algoritmos genéticos [1] imitan el proceso de evolución de la vida de los animales en la naturaleza con principios de selección natural y principios de supervivencia. Cada solución candidata tiene un conjunto de propiedades (sus cromosomas o genotipo) que pueden mutarse y alterarse.

Este algoritmo, parte de la validación de las poblaciones generadas en las pruebas y la selección natural de ellas (escoger cuál es el que cuenta con mejores condiciones para la prueba) se toma la aptitud el cual corresponde al valor de la función objeto de estudio (costo de la VNE que se va a evaluar).

Los individuos son seleccionados de forma estocástica sobre la población actual y las características de cada uno se reconfigura al azar para formar una generación nueva de individuos que serán usados en la próxima iteración que el algoritmo realice.

**Análisis de los algoritmos.**

Los algoritmos a nivel de pseudocódigo y lenguaje de programación presentan diferencias en su desarrollo, el algoritmo BNPA-VNE fue desarrollado en MATLAB, los algoritmos Genetico y PSO fueron desarrollados en Python usando herramientas adicionales que ayudan a estructurar la lógica para cada uno.

Para los algoritmos desarrollados en Python su lógica requiere sean ingresados parámetros de forma manual al iniciar su ejecución, posterior al ingreso de datos, el algoritmo se encarga de hacer un llamado a distintas funciones las cuales se encargan de: generar topologías de red, generación de números aleatorios, ordenes de llegada de las peticiones, etc. concluido estos procesos, se asignan los nodos en las redes virtuales.

Estos algoritmos se encargan de realizar el incrustamiento a las redes virtuales en dos etapas, [5] incrustar los nodos virtuales en una primera etapa e incrustar los enlaces virtuales en una segunda etapa. El algoritmo de PSO en el caso de la ecuación de velocidad resta (-) y suma (+) el primero identifica la intersección de dos partículas diferentes, mientras que el segundo opera en dos sustracciones. En otras palabras, se refiere a la unión de las diferencias (sustracciones) de dos partículas diferentes con la partícula a mano. El vector producido por la ecuación de velocidad es un vector binario que indica las posiciones actuales de las partículas que se van a cambiar.

Para la ejecución de los algoritmos, se hace uso del operador de cruce ordenado, el cual es un operador utilizado para codificaciones basadas en permutación en GA. Finalmente, se hace uso de un operador de mutación uniforme. de esta manera, los nodos del sustrato se eligen aleatoriamente y si el elegido no está en otra posición del cromosoma, dicho nodo se selecciona para alojar un nodo virtual; de lo contrario, se elige otro nodo de sustrato.

Durante las simulaciones también se hizo uso de los nodos ocultos, los cuales se encargan de implementar la lógica contenida en los algoritmos descritos anteriormente pero considerando la opción de lograr el incrustamiento de los recursos solicitados en un nodo contingente es decir, si el nodo principal seleccionado para el incrustamiento no puede realizar las asignación se busca otra opción cercana que cumpla con los recursos solicitados con el fin de realizar el incrustamiento correctamente.

Dada la casuística particular de los algoritmos usados en estas pruebas se evaluó cómo unificar los criterios de simulación, cuáles parámetros usar, cantidad de ejecuciones, la posible diferencia de los resultados de cada algoritmo con el fin de llegar a un concenso en la ejecución de la prueba a realizar.

**Metodología de la simulación**

las simulaciones se ejecutaron en dos computadores con las siguientes características técnicas:\\

• El primer computador portátil con un Sistema Operativo (SO) Windows 10, las simulaciones fueron ejecutadas en una máquina virtual Ubuntu 18, procesador Intel® CoreTM i5-7200U CPU @ 2.50 GHz y 2.71 GHz, memoria RAM asignada 6GB, espacio en Disco Duro de estado sólido de 20GB.\\

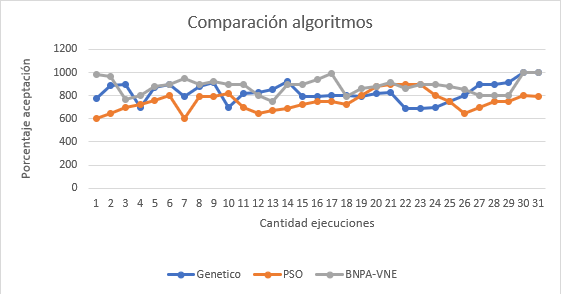
• El segundo computador portátil con un SO macOS Mojave Versión 10.14.6, procesador Dual-Core Intel® Core i5 2.30 GHz 2 núcleos, memoria RAM 8GB, Disco Duro de estado sólido de 121GB.

Al momento de considerar cuáles serían los elementos para ejecutar las pruebas se definió que se harían las mismas pruebas en los dos equipos de cómputo, para los procesos de simulación se ejecutaron 31 escenarios sin nodos ocultos y otros 31 escenarios con nodos ocultos por cada algoritmo (BNPA-VNE, GENETICO y PSO), obteniendo un total de 744 resultados ya que por cada algoritmo se ejecutaron 124 simulaciones.

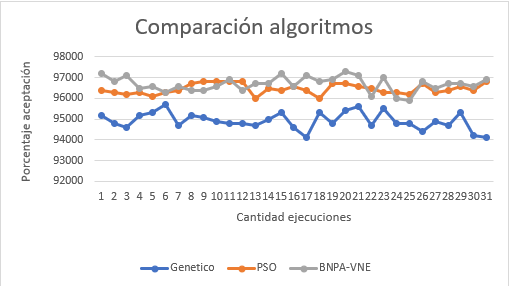
Para ejecutar las pruebas en el algoritmo BNPA-VNE fue necesario hacer uso del framework ALEVIN (ALgorithms for Embedding of VIrtual Networks) el cual es el encargado de realizar la generación de topologías de redes virtuales (VN) y redes sustrato (SN). En la ejecución de los algoritmos PSO y GENETICO se hace uso de Bibliotecas en Python tales como: Python-igraph 0.7.0, python-Numpy y python-scipy las cuales sirven para realizar los cálculos de las topologías de red, asignación de recursos por medio de vectores y matrices diseñadas en los algoritmos.

**Resultados de las simulaciones**

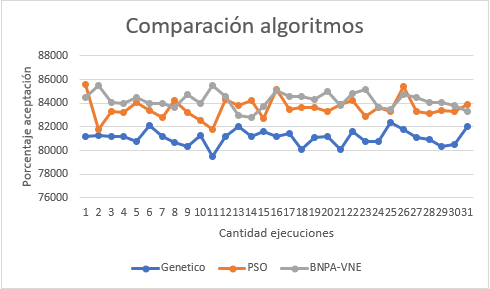
A continuación, se relacionan los resultados de las simulaciones, los primeros cuatro resultados se ejecutaron en el sistema operativo IOS y los cuatro últimos resultados fueron ejecutados en el sistema operativo Ubuntu 18.



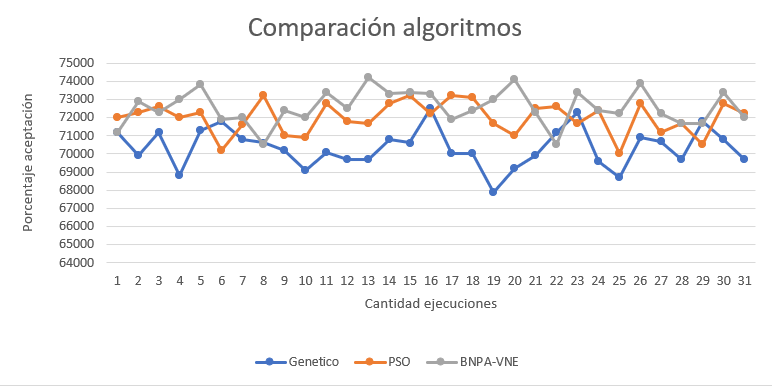
Simulación lanzando 500 peticiones a los algoritmos Genético, PSO Y BNPA-VNE. Sin nodos ocultos.



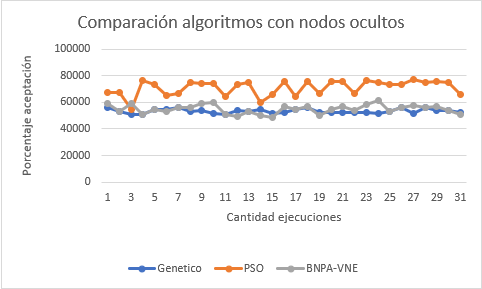
Simulación lanzando 1000 peticiones a los algoritmos Genético, PSO Y BNPA-VNE. Sin nodos ocultos.



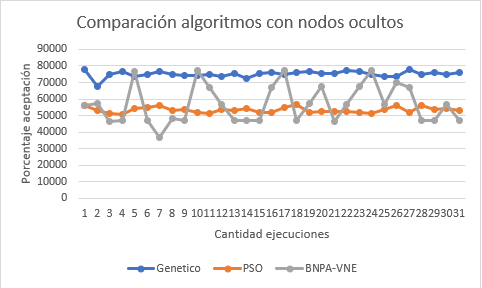
Simulación lanzando 1500 peticiones a los algoritmos Genético, PSO Y BNPA-VNE. Sin nodos ocultos.



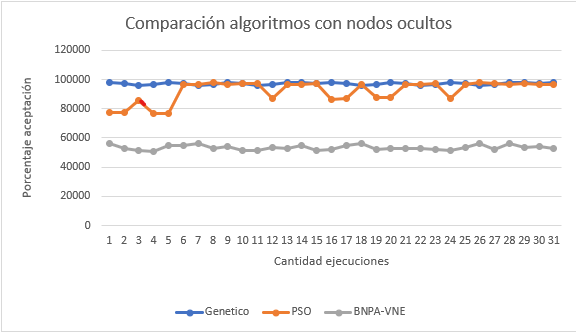
Simulación lanzando 2000 peticiones a los algoritmos Genético, PSO Y BNPA-VNE. Sin nodos ocultos.



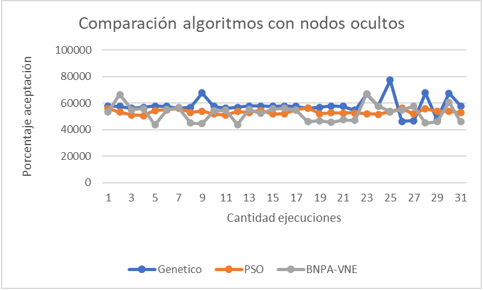
Simulación lanzando 500 peticiones a los algoritmos Genético, PSO Y BNPA-VNE. Con nodo nodos ocultos.



Simulación lanzando 1000 peticiones a los algoritmos Genético, PSO Y BNPA-VNE. Con nodo nodos ocultos.



Simulación lanzando 1500 peticiones a los algoritmos Genético, PSO Y BNPA-VNE. Con nodo nodos ocultos.



Simulación lanzando 2000 peticiones a los algoritmos Genético, PSO Y BNPA-VNE. Con nodo nodos ocultos.

**Discusión de los Resultados de las simulaciones**

Se relacionan a continuación, los análisis realizados a cada uno de los resultados obtenidos en las simulaciones ejecutadas:

Figura 1: En los resultados de la ejecución de los algoritmos, se evidencia que inicialmente con las mismas peticiones y cantidades de ejecución se generan tasas de aceptación iguales entre el algoritmo Genético y el BNPA-VNE de igual forma, se evidencia que el PSO arroja resultados inferiores pero muy cercanos a los otros dos algoritmos.

Figura 2: Los datos resultantes de la ejecución de los algoritmos sí reflejan diferencias en sus tasas de aceptación. Evidenciando que el algoritmo Genético tuvo las tasas de aceptación deficientes, en cambio, el comportamiento del PSO y BNPA-VNE presentaron un patrón de comportamiento similar en algunas de las peticiones como se puede evidenciar en las peticiones desde la 27 hasta la 31 se refleja una tendencia de incremento en sus aceptaciones.

Figura 3: En la ejecución de los algoritmos, se evidencia que el comportamiento del Genético es el que menor tasas de aceptación tiene, el algoritmo BNPA-VNE es el que mayor tasa de aceptación tiene y el PSO se encuentra en la mitad de ambos. Se evidencia que el ALGORITMO BNPA-VNE tiene las mejores tasas de aceptación y un comportamiento más estable que el PSO.

Figura 4: En esta ejecución, se evidencia nuevamente que el ALGORITMO BNPA-VNE es el que mejor comportamiento y tasas de aceptación maneja, aunque no tiene un comportamiento regular y estable seguido. Algoritmo PSO que tiende inicialmente a tener pocas aceptaciones, pero a partir de la ejecución número 7, empieza a realizar ejecuciones con tasas que tienen picos más altos, pero no es constante. El algoritmo Genético es el que peor desempeño evidencia dado que no es regular y tiene picos realmente muy bajos de aceptación en la ejecución actual.

Los siguientes son los análisis realizados a cada uno de los resultados obtenidos en las simulaciones ejecutadas con nodos ocultos:

Figura 5: Como se pude ver en la gráfica el algoritmo genético mostró un alto desempeño en un escenario este escenario a diferencia del BNPA-VNE, se mantuvo estable por debajo del rango de los otros dos algoritmos, lo cual puede interpretarse como muy estable, pero no óptimo dado que sus tasas de aceptación fueron inferiores respecto de los otros dos algoritmos esto puede deberse al hecho de que con nodos ocultos se consumen cantidades superiores de recursos, el algoritmo PSO muestra un desempeño inestable, pero al final se evidencia un patrón de comportamiento similar al algoritmo genético en el cual terminan siendo los dos muy estables en las tasas de aceptación.

Figura 6: El algoritmo Genético como en el caso anterior mostró un gran desempeño en las ejecuciones, siendo estable y teniendo muy pocas variaciones, pero el algoritmo que sorprende es el algoritmo BNPA-VNE que tuvo incrementos y decrementos repentinos en peticiones como se evidencia en la ejecución 5, 11, 17, 20 y 24 lo que evidencia que el BNPA-VNE no es estable. El algoritmo PSO se mantiene estable durante toda la ejecución por debajo de las tasas del algoritmo genético, pero con un comportamiento que se puede decir que es aceptable.

Figura 7: En la ejecución de los algoritmos con el orden de llegada de 1500, se evidencia que el comportamiento del Genético es el que menor tasas de aceptación tiene, el BNPA-VNE es el que mayor tasa de aceptación tiene y el PSO se encuentra en la mitad de ambos. Se evidencia que el BNPA-VNE tiene las mejores tasas de aceptación y un comportamiento más estable que el PSO.

Figura 8: En esta última ejecución se evidencia que el algoritmo Genético tiene la mayor tasa de aceptación de los tres ejecutados teniendo al final aceptaciones volátiles como evidencia por ejemplo en la ejecución numero 25 la cual supera en cantidad la aceptación de todas las pruebas de este proceso realizado. El algoritmo genético fue el más estable de todos tiene una tendencia inicial alta que fluctúa durante todas las ejecuciones, pero no tiene picos decrecientes importantes que generen una alerta o duda del porqué se presentó. El algoritmo BNPA-VNE tiene picos de aceptaciones en ocasiones altos y otros bajos, no es estable y termina con tendencia de baja aceptación por lo cual no se puede considerar como de los mejores en estas ejecuciones.

**Conclusiones:**

Se evidencia que los algoritmos desarrollados en Python presentan una leve diferencia con el algoritmo de Matlab en cuanto a la optimización de tiempos y recursos esto debido a la lógica empleada en el procesamiento de los algoritmos y a las herramientas usadas para su ejecución.

Los algoritmos desarrollados en Python (Genético y PSO) muestran un comportamiento estable en los momentos que se ejecutan peticiones con órdenes de llegada inferiores a los mil quinientos, pero se evidencia que tienen un patrón de comportamiento de picos altos y bajos cuando los tiempos de llegada seleccionados son superiores a lo mil quinientos las tasas de aceptación bajan y suben de forma exponencial.

El realizar las simulaciones en este tipo de algoritmos permite que los resultados sean de gran aporte para las investigaciones futuras dado que, se han documentado muy pocas investigaciones de este tipo y adicionalmente los algoritmos existentes también son pocos.

Se concluye que se deben realizar más investigaciones y desarrollos para lograr la estabilidad en las tasas efectivas de aceptación de las peticiones, si esto llega a suceder estos algoritmos lograrán un gran aporte porque pueden ayudar a mejorar los procesos de ejecución y optimización de los recursos, adicional a ello, también se hace necesario lograr realizar desarrollos que ayuden al manejo y uso de las herramientas en entornos gráficos con el fin de optimizar los tiempos de preparación de las pruebas y el ingreso de datos para ser ejecutados.

**Referencias**

[1] Zangiabady M. Zapata-Lara J. Rubio-Loyola JAguilar-Fuster C. “Online Virtual Network Em-bedding Based on Virtual Links’ Rate Require-ments”. En:IEEE Transactions on Network andService Management(2018).

[2] Botero J. F. Alzate Mejia jose. “adaptaci ́on delalgoritmo NPA-VNE para la reasignaci ́on de redesvirtuales mapeadas y la selecci ́on entre diferentestipos de m ́etricas”. En:Pereira : UniversidadTecnol ́ogica de Pereira(2017).

[3] Michael Till Beck Hermann de Meer AndreasFischer Juan Felipe Botero y Xavier Hesselbach.“Virtual Network Embedding: A Surveyg”. En:IEEE COMMUNICATIONS SURVEYSTUTO-RIALS, VOL. 15 pp. 4(2019).

[4] Ding Q. Zakia U. Li Z. Haeri S. Trajkovic L BenYedder H. “Comparison of Virtualization Algo-rithms and Topologies for Data Center Network”.En:26th International Conference on ComputerCommunication and Networks (ICCCN)(2017).

[5] J. F. Botero. “Study, evaluation and contributionsto new algorithms for the embedding problem ina network virtualization environment”. En:ThesisDoctoral, Universitat Polit`ecnica de Catalunya(2013).

[6] Prasan Kumar Sahoo Chinmaya Kumar Dehury.“DYVINE: Fitness based Dynamic Virtual Net-work Embedding in Cloud Computing”. En:IEEEJournal on Selected Areas in Communications(2019).

[7] V. Anand L. Li G. Sun H. Di H. Yu y B. Dong.“Efficient Online Virtual Network Mapping UsingResource Evaluation”. En:Journal of Networkand Systems Management(2012).

[8] Andreas Fischer Hermann de Meer Juan FelipeBotero Xavier Hesselbach. “Optimal mapping ofvirtual networks with hidden hops”. En:Telecom-munication Systems volume 51, pages 273–282(2012)(2011).

[9] M. Shaowu L. Wenzhi X. Yang y T. Xiongyan.“Completing virtual network embedding all inone mathematical programming”. En:Electronics,Communications and Control (ICECC), 2011 In-ternational Conference(2011).

[10] L. Casucci T. Phuong Nga y A. Timm-Giel. “Opti-mal mapping of virtual networks considering reac-tive reconfiguration”. En:in Cloud UniversidadCat ́olica de Pereira Networking (CLOUDNET),2012 IEEE 1st International(2012).

[11] Loganathan G. V Zong Woo Geem JoongHoon Kim. “A New Heuristic Optimization Al-gorithm: Harmony Search”. En:SIMULATION,76(2), 60–68(2001)