

INGENIERITZA GOI ESKOLA TEKNIKOA  
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA  
BILBAO

---

## TRABAJO DE FIN DE MASTER

ALGORITMOS EVOLUTIVOS MULTIOBJETIVO  
CON CODIFICACIÓN DANDELION PARA EL  
TRAZADO DE ESTRATEGIAS DE DESCARGA DE  
TRÁFICO OPORTUNISTA EN REDES  
MULTIOPERADOR

**Alumna:** Jone Consul Godoy

**Director:** Javier Del Ser Lorente



## Resumen

Con el aumento en el número de operadores de telecomunicaciones, tanto grandes operadores que cuentan con infraestructuras propias y espectro de radio con licencia, como operadores virtuales que explotan las infraestructuras de los primeros para ofrecer servicios, actualmente varios ofrecen servicios simultáneamente en la misma área geográfica. Cada uno de ellos, con sus propias políticas y estrategias en cuanto a clientes y precios.

En el mercado actual, todos los operadores compiten entre sí para conseguir el mayor número de clientes posible, sin tener en cuenta la limitación que puedan tener en los recursos disponibles, ya que cada usuario conectado a la red, supone reservar un ancho de banda específico y mantener esa conexión a pesar de que no esté en uso constante.

En este proyecto, se plantea una nueva manera de conexión de dispositivos; *tethering* iniciado por el operador. El *tethering* es una estrategia de compartición de recursos, en la que se permite que un nodo se conecte a través de otro a la red, reservando así únicamente el ancho de banda necesario para uno de ellos. Además, añadiéndole el mecanismo *content catching* permite que el dispositivo intermedio pueda funcionar a modo de caché, acercando ciertos contenidos a los usuarios.

En este proyecto, se estudia el efecto que las diferentes políticas relacionales de los operadores tienen tanto en los beneficios de los propios operadores como en la calidad de servicio recibida en los clientes. Estas políticas aplicadas, condicionarán los resultados obtenidos, tanto económicamente como en la degradación de la calidad de servicio debida al *tethering*.

Se establecen relaciones de coalición y de competencia entre operadores y se proponen diferentes topologías Pareto-óptimas vinculadas a ellas, obtenidas mediante un algoritmo de optimización multiobjetivo: *Harmony Search*.

## Laburpena

Telekomunikazio operadoreen hazkuntzarekin, bai operadore handiak, azpiegitura, antena eta irrati espektro propioak dituztenak, bai operadore birtualak zerbitzuak aurrekoei baliabideak alokatuz eskaintzen dituztenak, gaur egun, operadore asko daude azalera geografiko berean zerbitzuak ematen. Bakoitzak bere politika eta estrategia propioekin, bezeroei zein prezioei dagokionez.

Gaur egungo merkatuan, operadore guztiak elkarrekin lehiatzen dute ahalik eta bezero kopuru handiena lortzeko asmotan eduki dezaketen errekurtsu muga kontuan hartu gabe. Bezero bat operadorearen estaldura eremuan sartzen denean, automatikoki beharrezko baliabideak erreserbatzen dira berarentzat eta konexioa uneoro mantentzen da nahiz eta erabilera konstantea ez izan.

Proiektu honetan, dispositibo edo gailuen konexio modu berri bat aurkezten da: operadoreak hasitako *tethering*-a. *Tethering*-a, errekurtsoen banaketa estrategia bat da, zeinak dispositibo bat tarteko nodo bezala funtzionatzea ahalbidetzen duen. Modu honetan, operadoreak kudeatuta gailu bakarrarentzat errekurtsu erreserba egin daiteke nahiz eta gehiago zerbitzatu. Gainera, *content catching* mekanismoak gehituta, tarteko dispositiboak cache bezala erabiltzea ahalbidetu dezake sareko edukiak bezeroengana hurbilduz.

Proiektu honetan operadoreek euren artean aplikatzen dituzten erlazio-politika ezberdinen efektua aztertzen da, operadoreak izango dituen irabazietan eta *tethering* konexioen eraginez erabiltzaileak jasandako kalitate murrizketan. Politika hauek sortutako sarea eta emaitzak baldintzatzen dituzte, aztertutako bi parametroetan.

Koalizio eta konpetentzia erlazioak zehaztuko dira hauei zuzenean erlazionatutako eta optimizazio multiobjektibo algoritmo baten bidez, *Harmony Search*, lortutako pareto-optimoak diren sare topologia ezberdinak proposatuko dira.

## Abstract

With the increase of the number of telecommunication operators, nowadays there are many operators offering services over the same geographical area, those operators can be big ones, who have physical infrastructure and spectrum license, or virtual operators who offer services exploiting the infrastructures of the big operators. Each one has its own politics and strategies regarding to clients and pricing.

In the actual market, all the operators fight with each other for achieving the maximum number of clients as possible, without taking into account the physical resources limitations due to the maintenance of the connection of each user in spite of they aren't using it constantly.

In this Project a new connection way is laid out; operator started tethering connection. Tethering is a connection sharing strategy in which a node is allowed to reach the net through other node's connection saving this way the resources needed for one of the nodes. Besides, it allows the intermediate node work as a cache, bringing closer some contents to other users.

In this work the application of bi-objective heuristics for the design of Pareto-optimal network topologies is studied depending of the different relational policies applied by the operators. This policies, will condition the obtained results, influencing the revenue of the incumbent operators in the scenario and the quality of service degradation experienced by the end users as a result of tethering

Coalition and competition relationships are established between operators and different network topologies are proposed. These topologies will be obtained by the heuristic Harmony Search Algorithm.



## Contenido

Resumen.....	i
Laburpena.....	ii
Abstract.....	iii
Lista de ilustraciones.....	x
Lista de acrónimos.....	xi
1. Introducción y contexto.....	2
2. Objetivos y alcance.....	7
3. Beneficios.....	9
3.1. Beneficios técnicos.....	9
3.2. Beneficios sociales.....	9
3.3. Beneficios económicos.....	10
4. Estado del arte.....	11
5. Análisis de alternativas.....	13
5.1. Introducción.....	13
5.2. Algoritmo evolutivo.....	13
Definición de las alternativas.....	13
Criterios de ponderación.....	14
Ponderación de las alternativas.....	14
Elección final.....	14
5.3. Codificación.....	14
Definición de las alternativas.....	14
Criterios de ponderación.....	15
Ponderación de las alternativas.....	15
Elección final.....	15
5.4. Lenguaje de programación.....	16
Definición de las alternativas.....	16
Criterios de ponderación.....	16
Ponderación de las alternativas.....	17
Elección final.....	17
5.5. Resumen.....	17
6. Análisis de riesgos.....	18
6.1. Incumplimiento de plazos.....	18

6.2.	Costes imprevistos.....	18
6.3.	Falta de ayuda/manuales.....	18
6.4.	Simulaciones pesadas.....	19
6.5.	Errores en la codificación.....	19
6.6.	Problemas con la heurística.....	19
6.7.	Conclusiones no significativas.....	19
6.8.	Parámetros del escenario no realistas.....	19
6.9.	Problemas con la codificación Dandelion.....	20
7.	Metodología.....	21
7.1.	HS y codificación Dandelion.....	21
	Algoritmos evolutivos:.....	21
	Algoritmo HS.....	22
	Optimización multiobjetivo.....	25
	Dandelion:.....	28
7.2.	Reparación del árbol.....	31
7.3.	Evaluación de la calidad.....	34
7.4.	Evaluación del beneficio.....	35
7.5.	Creación de los escenarios diferenciados.....	35
7.6.	Pruebas realizadas.....	38
	7.2.1. Optimización conjunta (backhaul).....	38
	7.2.2. Optimización independiente.....	38
8.	Cálculos.....	40
8.2.	Descripción del problema.....	40
8.3.	Descripción de los escenarios.....	41
8.4.	Pruebas realizadas.....	45
8.5.	Parámetros estudiados.....	46
8.	Descripción de las tareas y Gantt.....	47
9.1.	Descripción de tareas.....	47
9.1.	Diagrama de GANTT.....	49
10.	Presupuesto.....	51
10.1.	Mano de obra.....	51
10.2.	Gastos.....	51
10.3.	Amortizaciones.....	52
10.4.	Total.....	52
11.	Conclusiones.....	53



12.	Referencias .....	55
1.	Resultados .....	60
1.1.	Resultados gestión conjunta .....	61
1.2.	Resultados gestión independiente .....	61
1.2.1.	Coalición .....	61
1.2.2.	Competencia.....	62
1.3.	Comparación de los 3 casos .....	63
1.4.	Resultado simulaciones movilidad .....	65
1.5.	Variación en el número de usuarios .....	67
2.	Artículo publicado .....	69



## Lista de tablas

Tabla 1: Ponderación algoritmo evolutivo .....	14
Tabla 2: Ponderación codificación topología .....	15
Tabla 3: Ponderación del lenguaje de programación .....	17
Tabla 4: Tareas realizadas .....	47
Tabla 5: Horas internas .....	51
Tabla 6: Gastos .....	51
Tabla 7: Amortizaciones .....	52
Tabla 8: Presupuesto .....	52
Tabla 9: Resultados beneficio 75 nodos .....	64
Tabla 10: Resultados QoS 75 nodos .....	64
Tabla 11: resultados backhaul movimiento - Beneficio .....	65
Tabla 12: resultados backhaul movimiento – QoS .....	65
Tabla 13: Resultados beneficio para 20 usuarios .....	67
Tabla 14: Resultados beneficio para 50 usuarios .....	68
Tabla 15: Resultados retardo para 20 usuarios .....	68
Tabla 16: Resultados retardo para 50 usuarios .....	68

## Lista de ilustraciones

Ilustración 1: Previsión del tráfico móvil mensual en2020-según Cisco .....	2
Ilustración 2: Previsión de porcentaje de conexiones inalámbricas y fijas- según Cisco .....	
Ilustración 3: Topología en estrella y árbol .....	
Ilustración 4: Evolución de los precios de las tarifas móviles en España según el País .....	5
Ilustración 5: Diagrama de riesgos .....	
Ilustración 6: Red base para las pruebas.....	40
Ilustración 7: Solución gestión conjunta de la red .....	43
Ilustración 8: Solución gestión independiente de la red .....	44
Ilustración 9: Funcionamiento HS .....	
Ilustración 10: Frente de pareto.....	<b>¡Error! Marcador no definido.</b>
Ilustración 11: Convergencia Pareto.....	26
Ilustración 12: Gráficas de Rangos.....	27
Ilustración 13: Crowding distance.....	27
Ilustración 14: Topología árbol. $C=(6, 2, 1, 7, 3, 7, 3, 8)$ .....	29
Ilustración 15: Tronco árbol .....	30
Ilustración 16: Árbol inicial.....	32
Ilustración 17: Árbol corregido.....	34
Ilustración 18: Escenario optimización conjunta .....	36
Ilustración 19: Escenario gestión independiente.....	37
Ilustración 20: Diagrama de gantt.....	
Ilustración 21: Resultados gestión conjunta.....	61
Ilustración 22: Resultados gestión independiente - coalición .....	62
Ilustración 23: Resultados gestión independiente - competencia.....	62
Ilustración 24: Comparación resultados conjunta e independiente .....	63
Ilustración 25: Comparación resultados gestión independiente .....	64
Ilustración 26: Resultados gestión conjunta con movimiento .....	66
Ilustración 27: Resultados gestión independiente movimiento – coalición y competencia .....	66
Ilustración 28: Comparación resultados movimiento.....	67
Ilustración 29: Comparación resultados movimiento gestión independiente .....	<b>¡Error! Marcador no definido.</b>
Ilustración 32: Comparación max/min/media con diferente cantidad de nodos – beneficio y QoS .....	<b>¡Error! Marcador no definido.</b>

## Lista de acrónimos

HS: Harmony Search

BS: Base-Station. Estación base

EA: Evolutive algorithm- Algoritmo evolutivo.

TIC: Tecnologías de la información y comunicaciones.

3G,4G,5G: Tecnologías móviles

LTE: 4G

GSM: 3G

IoT: Internet of things

M2M: Machine to machine

P2P: Peer to peer



# MEMORIA

---

# 1. Introducción y contexto

En la última década, el sector de las Telecomunicaciones o tecnologías de la información y la comunicación (TIC) ha experimentado una gran evolución. Hace unos años, era impensable prever la situación actual; el crecimiento y desarrollo se ha dado en todos los ámbitos de este sector ha hecho que su papel sea cada día más importante y actualmente el uso de los diferentes sistemas de telecomunicación sea algo necesario para el día a día. El aumento en la cantidad de dispositivos, en el uso que se les da, en el volumen de datos y tipo de conexiones generadas... son hoy en día el motor para continuar evolucionando.

Esta evolución se ha dado de forma progresiva; a medida que el proceso de desarrollo tecnológico avanzaba y los dispositivos (sobre todo móviles) eran capaces de soportar más prestaciones, el uso de los mismos ha experimentado un rápido crecimiento, lo que se ha traducido directamente en un aumento de la demanda del ancho de banda y tráfico generado y tanto en cantidad como en calidad de las conexiones establecidas. Uno de los factores más relevantes en esta evolución ha sido la aparición de los Smartphones, convirtiendo las redes inalámbricas en el ámbito que más afecta a las TICS.

La aparición de los smartphones, ha supuesto una revolución en el uso de los teléfonos móviles. Actualmente, casi todo el mundo dispone de uno y el uso de éste no se limita únicamente a las funciones de un teléfono. Hoy en día, un smartphone es una herramienta de trabajo, una manera de estar al día, consultar las noticias y el correo, y de estar conectado a las redes sociales. Este uso en diversos entornos provoca una cantidad elevada de tráfico móvil e inalámbrico y además del uso masivo actual, se prevé que los smartphones, tablets y otros elementos inalámbricos, serán los dispositivos más usados en los próximos años.

Este incremento en el uso, según cisco [1] ha producido que en los últimos 10 años el tráfico de datos móviles se haya duplicado alrededor de 4000 veces. Además, analizando las tendencias evolutivas del sector, este aumento se está produciendo de forma exponencial. Mirando a las estadísticas, en el año 2015 el volumen de datos se incrementó un 74 % con respecto al año anterior, es decir, 3.7 exabytes al mes frente a los 2.1 exabytes al mes a finales de 2014. En la siguiente imagen se puede observar la evolución del volumen de datos en los últimos años y la esperada hasta 2020:

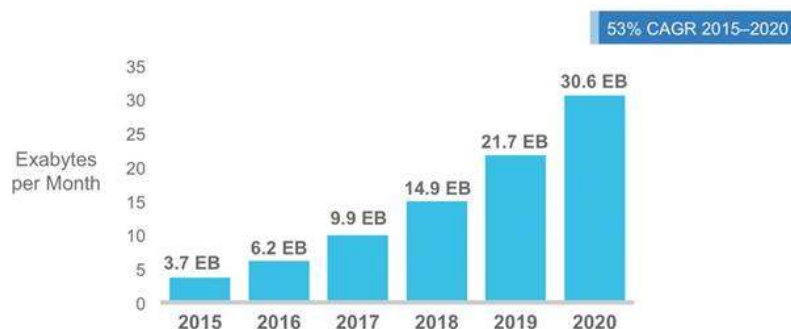
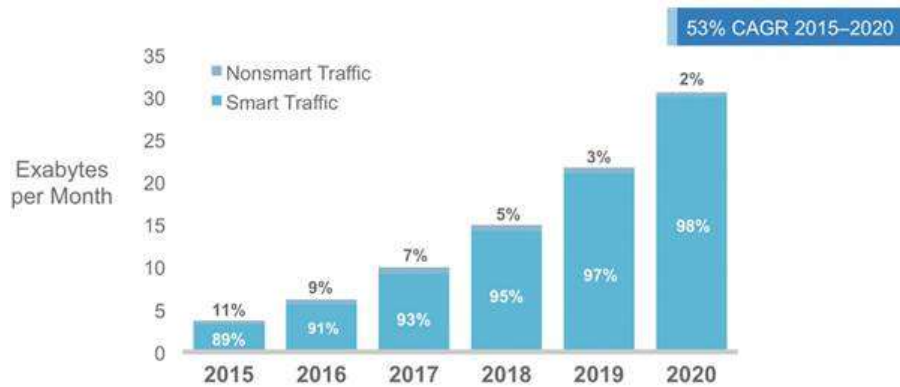


Ilustración 1: Previsión del tráfico móvil mensual en 2020-según Cisco



Como se aprecia en la ilustración anterior, se espera que estas cantidades sigan en aumento de forma exponencial y que para el año 2020 prácticamente todas las conexiones realizadas a internet (98%) se establezcan desde un dispositivo inalámbrico, smartphone o Tablet.



**Ilustración 2: Previsión de porcentaje de conexiones inalámbricas y fijas- según Cisco**

A parte del crecimiento mencionado en la cantidad de tráfico por el mayor uso de los dispositivos, se espera que las velocidades inalámbricas ofrecidas por los operadores sigan aumentando para mejorar la calidad del servicio y poder cubrir la demanda, lo que generará a su vez un mayor incremento del tráfico. Este aumento en las velocidades ofrecidas, viene dado sobre todo por las diferentes tecnologías implementadas por los operadores. Hoy en día, en cuanto a datos móviles, están en uso las tecnologías 3G y 4G, aunque esta última todavía está en fase de despliegue en España.

Esta mejora en la velocidad, está suponiendo también un cambio en las aplicaciones usadas, ya que con mayores velocidades, muchas aplicaciones en tiempo real (videollamadas, VoIP, streaming...) se podrían usar desde dispositivos inalámbricos sin percibir pérdida o deterioro de la calidad. Algunas de estas aplicaciones en tiempo real, tienen requerimientos en cuanto a pérdidas y retraso o mueven una gran cantidad de datos y pueden tener limitaciones con las tecnologías actuales.

Todo esto, hace que las investigaciones en torno a este área estén en alza para buscar diferentes soluciones y mejoras y conseguir así mayores prestaciones. Por tanto, a pesar de que varios operadores están todavía despegando sus infraestructuras 4G, en previsión de la evolución, ya se ha comenzado a investigar la siguiente generación de tecnologías inalámbricas: 5G. La definición de esta nueva generación no es clara todavía, ya que se busca mejorar en diferentes aspectos respecto a 4G y se está debatiendo cuales de esos aspectos son los más relevantes. Los estudios sobre 5G, no solo se centran en proporcionar velocidades de vértigo, sino que también buscan hacer que las conexiones ganen en calidad, mejorando por una parte el número de terminales al que puede dar servicio cada estación base simultáneamente y por otra parte, reduciendo la latencia.

Actualmente y para las tecnologías que se están implantando, la red se compone de diferentes nodos conectados directamente a una antena o estación base (BS) que pertenece a un operador. Este tipo de conexiones crea una topología de estrella. De esta manera, todos los nodos que están en el área de cobertura de la antena de su operador, se intentan conectar automáticamente, teniendo que competir por los recursos disponibles. A pesar de que un dispositivo no esté usando el ancho de banda, éste está reservado y no permite que otro usuario se conecte y lo aproveche. Con esta política, se puede perder capacidad de servicio y es posible que algún usuario se quede sin conexión, especialmente en el caso de masificación alrededor de una antena.

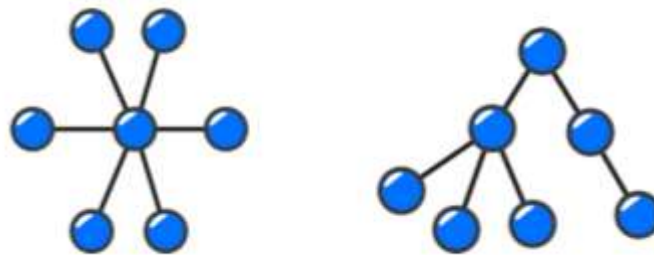


Ilustración 3: Topología en estrella y árbol

En este proyecto se plantea una nueva forma de conexión de nodos, lo que conlleva una arquitectura en árbol mediante el uso del *tethering* iniciado por el operador que permitirá aprovechar de manera más eficiente los recursos disponibles. El *tethering*, es una estrategia de compartición de recursos en la que un nodo crea un área de cobertura a su alrededor y permite que otros dispositivos se conecten a través de él a la red. Este *tethering*, nos permitiría crear picoceldas en las que los repetidores en lugar de ser antenas, estaciones base u otros elementos del operador, serían los propios smartphones de los usuarios y de esta manera, con los recursos reservados para una sola conexión, se puede dar servicio a varios dispositivos ahorrando parte del espectro al operador. Con esta conexión a través de otros dispositivos y añadiendo otros mecanismos como se explicará en los siguientes párrafos, se podrían conseguir algunos de los valores establecidos en 5G en cuestión de latencias.

Las picoceldas mencionadas (también definidas en 4G para frecuencias móviles) según se está definiendo, usarían ondas milimétricas (*mmwaves*) que permitirían una conexión con un alto ancho de banda. Además de estas altas velocidades entre dispositivos, la topología en árbol permitiría el uso de *network catching*, es decir, los propios equipos de los usuarios que hacen *tethering*, también harían funciones de caché y guardarían los contenidos consumidos por ellos mismos o por algún nodo posterior. De este modo, en lugar de conectarse a un *proxy* o caché que esté a varios saltos, si un nodo anterior tiene el contenido solicitado, se descargaría directamente desde éste, ahorrándose así unos cuantos saltos, tiempo y tráfico a la red.

Además de lo referente a las mejoras en las tecnologías, otra de las cuestiones que se plantea en este proyecto, es el impacto de las diferentes relaciones entre los operadores en el beneficio económico y en la calidad de servicio ofrecida a los usuarios.

Con la liberalización del mercado de las telecomunicaciones, la cantidad de operadores que ofrecen servicios en la misma área geográfica ha crecido notablemente. A día de hoy, además de los grandes operadores móviles que cuentan con antenas e infraestructuras propias, existen operadores virtuales que no cuentan con antenas ni licencias de espectro pero ofrecen servicios alquilándoselos a los operadores anteriores (*Over The Top services*) [2]. De esta manera tienen usuarios móviles que se pueden conectar a la red sin contar con la infraestructura física propia para ello.

El aumento en la cantidad de operadores, tanto los que cuentan con infraestructura propia como virtuales, ha provocado que el mercado de las TIC sea un mercado mucho más competitivo que hace unos años y por tanto, que los precios también lo sean. En los últimos años, el precio de los contratos ha descendido de manera considerable en todo el sector, tanto en los servicios fijos como móviles, mientras que los servicios prestados y la calidad de éstos han mejorado.

En la siguiente imagen se puede ver la evolución de los precios de las cuotas (en Euros) de alta y tarifas de datos desde 2007. Como se puede apreciar, la tendencia es que los precios sigan bajando. Este descenso hace que los operadores obtengan menos beneficios por usuario a pesar de que la cantidad de servicios y la calidad de éstos sean cada vez mayores.

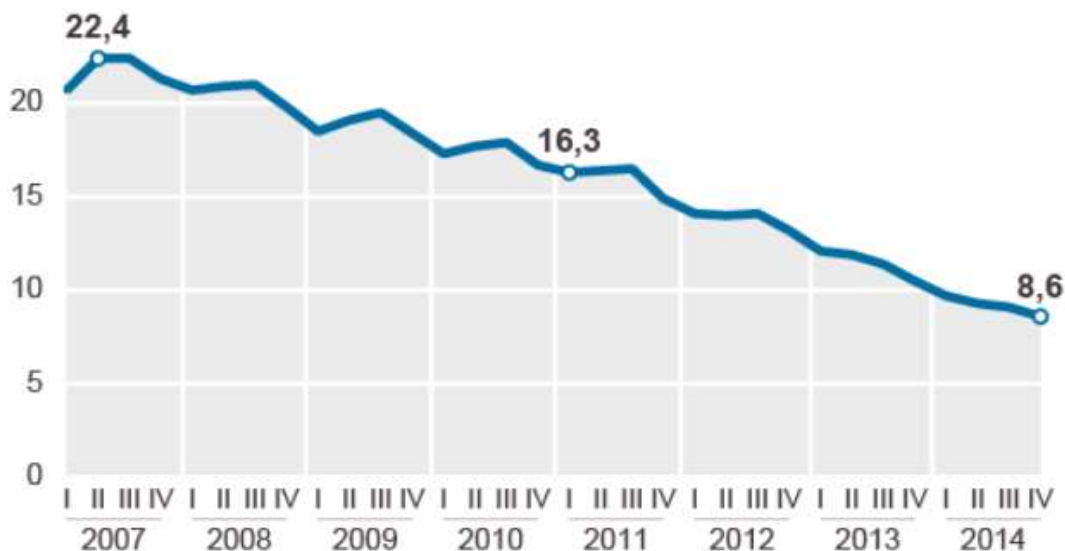


Ilustración 4: Evolución de los precios de las tarifas móviles en España según el País

Esta bajada en los precios, no solo se debe a la competitividad del mercado, otro de los factores que influye es la fluctuación de clientes móviles de los operadores. Hoy en día, una vez acabada la permanencia, los usuarios se cambian de operador buscando mejores ofertas produciendo que los operadores bajen los precios para poder captar un mayor número de usuarios.

En la actualidad, cada operador proporciona servicios a sus propios usuarios, o en el caso de alquilar servicios a operadores virtuales, a los de este segundo operador también, pero la relación entre operadores es una relación de competencia. Cada uno tiene sus propias estaciones y con ellas da servicio a sus usuarios. Esta política de competencia entre operadores, puede hacer que los operadores no obtengan el beneficio máximo posible, ya que cada operador depende únicamente de sus usuarios y recursos.

En este escenario en el que los recursos son limitados y la demanda de ellos es cada vez mayor se sitúa este proyecto que propone una manera de construir redes inalámbricas en árbol mediante el algoritmo de optimización *Harmony Search (HS)*. Este algoritmo, buscará que se mejore la calidad de usuario a la vez que maximiza el beneficio del propio operador consiguiendo un conjunto de soluciones Pareto-óptimas. Se plantea el efecto que las políticas de relaciones entre operadores pueden tener en las topologías de árbol de las redes creadas y en los parámetros mencionados fijando relaciones de coalición y de competencia entre ellos.

## 2. Objetivos y alcance

En este apartado se hablará sobre el alcance del proyecto y los objetivos que se tendrán que cumplir para su realización.

El objetivo principal del trabajo es el análisis del efecto de las diferentes políticas o estrategias que pueden seguir los operadores respecto a las relaciones entre ellos y la repercusión que éstas tendrían en diferentes aspectos de la red. Para cumplir el objetivo principal marcado, se establecerán los siguientes objetivos secundarios o parciales:

- Programación del algoritmo *Harmony Search* multiobjetivo para la evolución de árboles con criterio de coste y calidad de servicio (QoS).

Se programará un módulo para crear y optimizar las arquitecturas de red. Este módulo contendrá el algoritmo evolutivo *Harmony Search* que se modificará para una optimización multiobjetivo. Mediante este algoritmo, se analizarán posibles topologías de red y conexiones entre nodos manteniendo las más válidas hasta conseguir un conjunto de soluciones Pareto-óptimas.

Los dos objetivos que se tendrán en cuenta para la evaluación de la aptitud de cada topología serán por una parte, el beneficio del operador en unidades monetarias ya que este es un factor relevante a la hora de la posible implantación del proyecto y por otra, se evaluará la calidad de servicio experimentada por el usuario, medida en el número de slots de tiempo que necesita para alcanzar la estación base. Este segundo criterio se ha establecido atendiendo a las posibles limitaciones futuras en la red.

- Extensión del módulo para la evaluación simultánea de varios grafos en árbol

El siguiente paso, será adaptar la codificación del algoritmo HS programado para evaluar la aptitud de diferentes topologías de red simultáneamente. Para ello, se usará la codificación *Dandelion* que permite la creación y posterior evaluación de diferentes grafos en árbol de manera eficaz mediante una cadena de dígitos o código. Se evaluarán varias topologías de red para cada iteración manteniendo las más aptas en base a los criterios establecidos en el punto anterior para las siguientes iteraciones.

- Definición y creación de escenarios de simulación que cubran fielmente diferentes características en la operación de red

Se busca estudiar las diferentes políticas relacionales que puedan darse entre operadores y el efecto que tienen en las topologías de las redes creadas y en los parámetros ésta, por lo que tras la creación y adaptación del algoritmo principal, se crearán diferentes escenarios que puedan ser aplicables a la situaciones reales actuales. En estos escenarios reales, se plasmarán las arquitecturas de los operadores, así como los dispositivos móviles de los usuarios y las conexiones entre ellos.

Para esta definición realista de los escenarios, se tendrán en cuenta tanto parámetros referentes a la red como dimensión, arquitectura, áreas de cobertura, coste de las conexiones... como parámetros propios de cada operador como podrían ser el número de usuarios, precio de los contratos, políticas de calidad... y se intentará acercarlos lo máximo posible a la realidad. De esta manera, se conseguirán escenarios ficticios que reproduzcan fielmente las circunstancias actuales y las diferentes topologías propuestas.

- Simulación, análisis de resultados y obtención de conclusiones respecto al módulo de colaboración más ventajoso para todas las partes

Se quiere analizar el efecto que las diferentes políticas de relación aplicadas entre operadores tienen en la topología de red creada lo que influirá por tanto en el ámbito económico y en la calidad de servicio. Para esto, tras la construcción de los escenarios, se realizarán varias simulaciones con una cantidad elevada de usuarios en la que se probarán diferentes arquitecturas y políticas de relación entre los operadores.

Una vez obtenidos los resultados, se llevará a cabo un análisis exhaustivo tanto de los valores económicos como de la calidad de servicio y se plantearán las situaciones más ventajosas por una parte para el operador y por otra para los usuarios.

- Validación del desarrollo en un escenario realista con integración de los resultados por movilidad de los nodos

Las simulaciones anteriores, a pesar de emular un escenario realista, mantienen la posición fija de todos los usuarios inalámbricos. Esta situación, no se adapta fielmente a la realidad, por lo que para terminar, se añadirá movilidad a los nodos y se realizarán las simulaciones en las mismas condiciones que las mencionadas en el apartado anterior. De esta manera, se podrá comparar el impacto que tiene la movilidad de los usuarios tanto en la calidad experimentada como en las ganancias de los operadores.

## 3. Beneficios

En este apartado, se explicarán los beneficios que este proyecto aporta, dividiéndolos en tres tipos diferentes: Beneficios técnicos, beneficios sociales y beneficios económicos.

### 3.1. Beneficios técnicos

En cuanto a los beneficios técnicos, que este proyecto aporta, el más relevante es la utilización de un mecanismo de optimización para el cálculo de las topologías de redes móviles. Garantizando así una topología que maximice la utilización de los recursos disponibles para poder asegurar una red en árbol óptima en cada momento.

Se propone una manera diferente a la actual de crear arquitecturas de conexión de nodos o dispositivos móviles: *Tethering* iniciado por el operador. De manera que se aprovechan mejor los recursos disponibles y se maximiza el ancho de banda disponible, el área de cobertura de cada operador y la cantidad de nodos a los que se les puede dar servicio.

Las topologías que se plantean, buscan además del máximo aprovechamiento de los recursos, minimizar el retardo experimentado por los usuarios, permitiendo así tener unas redes móviles con menores latencias que soporten muchas más aplicaciones y tráfico con requerimientos de calidad de servicio.

Asimismo, para la construcción de las diferentes topologías se emplea un algoritmo evolutivo asegurando de esta manera un resultado eficaz y flexible que permite una rápida adaptación a la situación de la red en cada momento, manteniendo la calidad y los beneficios, en caso de movilidad de los nodos. Se asegurarán topologías que cumplan con los parámetros necesarios para una gestión eficiente de los recursos y una mejor calidad de servicio.

### 3.2. Beneficios sociales

Como se ha mencionado en la introducción, el aumento de la demanda de cobertura y conexión de los usuarios en todo momento es una realidad. En este proyecto se proponen diferentes arquitecturas que permiten la conexión a la red a través de nodos cercanos geográficamente, de manera que se pueda ampliar el radio de cobertura de los operadores; en el caso en el que un usuario se encuentre en un área sin cobertura, si se encuentra en el radio de *tethering* de otro nodo, podría conectarse a la red a través de éste.

Se podrá dar cobertura en áreas en las que actualmente no llega la estación base en el caso de que haya un nodo cerca al que se pueda conectar. Un ejemplo de esta situación, pueden ser las viviendas. Actualmente, hay viviendas que no disponen de cobertura en todas las habitaciones. Con esta propuesta, si el usuario que no tiene

cobertura se encuentra dentro del área de cobertura de otro nodo, podrá conectarse a la red a través de éste.

Además de la mejora en la cobertura, se pretende aumentar la percepción del usuario en cuanto a la calidad y el precio de los servicios, ya que se propone que para el caso en el que un dispositivo funcione como nodo intermedio y permita conexiones en *tethering*, se incentive al usuario de manera económica.

### 3.3. Beneficios económicos

En cuanto a los beneficios económicos, el más destacable es el beneficio del operador dado que es uno de los motivos principales para la evolución en el sector. Para un operador, cada conexión establecida con un nodo supone una reserva de recursos, por lo que conlleva un gasto además de un mal aprovechamiento de recursos. El gasto asociado al mantenimiento de la conexión con un usuario será mayor si no se está generando tráfico desde éste.

Mediante el *tethering* iniciado por el operador, se propone una estrategia de compartición de recursos en la que se podría aprovechar la conexión de uno de los nodos para la conexión de varios reduciendo así los costes asociados a las conexiones al equipamiento del operador. Una conexión mediante *tethering* no le supone ningún gasto en cuanto a recursos reservados al operador. En este proyecto, se plantea un incentivo a los usuarios que permiten a sus dispositivos funcionar como nodos intermedios, por lo que los gastos se reducirían notablemente. En este caso, se ha decidido incentivar a los usuarios que funcionan como *hotspot* por lo que sí que habrá un coste asociado al *tethering*, en cualquier caso, este coste estará determinado por el operador y será considerablemente menor que el de una conexión directa.

De esta manera, la cantidad de nodos conectados directamente a la BS se reduciría abaratando el gasto de mantenimiento de las conexiones con los dispositivos móviles.

Como beneficio económico, también se podría tener en cuenta el beneficio del usuario, ya que el dispositivo que funcione tenga otros conectados mediante *tethering* y funcione como nodo intermedio, recibirá un incentivo por los recursos que está cediendo al operador, traduciéndose en una reducción en el precio del contrato.



## 4. Estado del arte

Este trabajo es un proyecto de investigación en el que se estudia una nueva forma de interconexión para dispositivos móviles y un cambio en las relaciones actuales entre los operadores. Para el análisis además del beneficio económico que mueve el sector se ha tenido en cuenta uno de los parámetros más valorados actualmente para las propuestas de nuevas tecnologías móviles, la latencia. Por ello en este apartado se hablará sobre la situación actual de la tecnología 5G para poder entender mejor el contexto en el que se plantea este proyecto

Se denomina generación móvil a cada nueva tecnología que se ha desplegado para ofrecer servicios móviles. Cada nueva generación ha sido motivada por la necesidad de saciar una demanda o falta generada e identificada en la generación anterior. Para el caso del cambio de 2G a 3G, el avance generacional vino dado por la necesidad de mayor conectividad y datos móviles (banda ancha) en los dispositivos de los usuarios. La tecnología 2g permitía enviar datos pero el tráfico de datos fue ganando importancia por lo que se cambió a 3G.

La transmisión de datos ha seguido creciendo y adquiriendo mayor importancia por lo que para el caso de 4G, el cambio vino dado por el aumento de la velocidad y bajada de las latencias, lo que hizo que los usuarios 4G empezaran a consumir el doble de datos que los usuarios no-4G. Se espera que el caso de cambio de generación para 5G sea similar, es decir, una mayor subida de la velocidad (especialmente de bajada) y una disminución de la latencia.

En el caso de la tecnología 4G o LTE, aún está en fase de despliegue en España, por lo que todavía no se ha advertido ninguna carencia o requisito que no se satisfaga. A pesar de esto, teniendo en cuenta las previsiones vistas en el apartado 1 de esta memoria, ya se ha empezado a investigar la siguiente generación móvil. Estas investigaciones dirigidas por el organismo internacional GSMA (*GSM Association*) buscan unos estándares para cumplir en la siguiente generación. Como asociación representante de la industria móvil, GSMA además de trabajar en la definición y estandarización de 5G, también jugará un papel importante en el modelado de la estrategia de despliegue tanto en el ámbito comercial como regulatorio. Una vez definidas las características principales, GSMA además trabajará con todos sus miembros para poder identificar y desarrollar aplicaciones viables para 5G. Entre ellas se encuentran principalmente las comunicaciones M2M (*machine to machine*).

Hoy en día, todavía no están claros los parámetros/criterios necesarios para definir una nueva generación, aunque se plantean los siguientes:

- Conexiones *end-to end* de 1-10Gbps reales
- latencias de 1 milisegundo (*end-to-end round trip delay*)
- 1000x *bandwidth* por unidad de área
- Límite de dispositivos conectados
- (percepción de) 99.999% disponibilidad
- (percepción de) 100% cobertura
- 90% reducción de consumo de energía de la red
- Más de 10 años de batería para dispositivos para comunicaciones M2M

Estos requisitos, son los propuestos actualmente, aunque no es necesario el cumplimiento de todos ellos para la definición de 5G, sí que sería necesario cumplir los dos primeros (y más importantes). El hecho de que los requerimientos no estén establecidos, crea diferentes opiniones sobre el camino que debe seguir esta nueva generación y las necesidades que debe cumplir.

Por una parte está la visión consolidación de las tecnologías anteriores (2G,3G y 4G, Wifi...). Se espera que mediante 5G se consigan todas las resuelvan todas las carencias que se han presentado anteriormente, dando lugar a una hiperconexión de todos os dispositivos. Esta visión, busca una conectividad total tanto en tiempo como en espacio demandando una mayor cobertura y calidad.

Por otra parte, está la opinión de que el cambio generacional debe venir por un cambio en la tecnología radio de acceso, de manera que se consigan unas velocidades reales cerca de los 10Gbps y unas latencias reducidas.

Estas dos visiones, pueden ser contradictorias en algunos puntos, aunque ambas coinciden en el aumento de la velocidad y la reducción de la latencia. Es por esto que se han tomado estos dos parámetros como base para la elaboración de este proyecto.

## 5. Análisis de alternativas

### 5.1. Introducción

En esta sección se realizará un análisis de las diferentes alternativas que se plantean para llevar a cabo el proyecto y cumplir con los objetivos fijados anteriormente de la forma más efectiva posible.

En el proyecto se busca un conjunto de soluciones optimicen los objetivos marcados. Para ello se usará un algoritmo evolutivo, de esta manera, partiendo de una solución inicial, se realizarán diferentes iteraciones hasta conseguir el mejor resultado. Por tanto, es importante la elección de este algoritmo. En este caso, se han valorado tres tipos diferentes de algoritmos que realizan diferentes optimizaciones.

Además del algoritmo evolutivo, para la definición de la red, se usará un código que permita un fácil mapeo entre la topología y el propio código. Para esta elección, se han valorado 3 tipos de codificación diferentes.

Tras la elección del algoritmo y el código a utilizar, la siguiente alternativa será el lenguaje de programación. En este caso, se busca un lenguaje interpretado y de fácil aprendizaje y se valorará que sea un lenguaje usado actualmente, por tanto, las opciones a valorar serán python, matlab y perl.

### 5.2. Algoritmo evolutivo

#### Definición de las alternativas

Como se ha expuesto previamente, se busca optimizar una topología de red. Para ello, mediante un algoritmo evolutivo y el código seleccionado en el siguiente apartado, se realizarán diferentes iteraciones y se irán guardando los mejores resultados. Para poder encontrar el mejor resultado, es necesario elegir el algoritmo de optimización adecuado. Entre los diferentes algoritmos, se analizarán las siguientes opciones:

- Algoritmos genéticos: Son algoritmos de optimización inspirados en la evolución de las especies. Se basan en someter a los individuos o resultados a procesos de mutación aleatorios y de acuerdo con el proceso de selección natural, pasar los mejores a la siguiente generación
- *Swarm intelligence* (Inteligencia de enjambre): Son algoritmos inspirados por la naturaleza, especialmente por enjambres y colonias de hormigas. Las interacciones o mutaciones en los resultados se consiguen mediante interacciones locales y no mediante un proceso centralizado.

- Harmony Search: Es un algoritmo de optimización que simula la improvisación a la hora de crear música. Se crean diferentes notas o soluciones con el fin de conseguir la armonía.

#### Criterios de ponderación

- Eficiencia del algoritmo(4/10):
- Simplicidad de programación del algoritmo(3/10)
- Uso del algoritmo en procesos similares (3/10)

#### Ponderación de las alternativas

En la siguiente tabla se muestra la puntuación que se le ha asignado a cada uno de los criterios definidos en el apartado anterior.

	Algoritmos genéticos	Inteligencia de enjambre	de Harmony search
Eficiencia (4/10)	8	8	9
Simplicidad (3/10)	7	6	7
Trabajo anterior(3/10)	5	6	8
<b>TOTAL:</b>	<b>6.8</b>	<b>6.8</b>	<b>8.1</b>

Tabla 1: Ponderación algoritmo evolutivo

#### Elección final

Una vez ponderadas todas las alternativas, vemos que la que mayor puntuación ha obtenido es Harmony Search. Este algoritmo ha conseguido la puntuación más alta debido a la eficiencia y a que no ha sido usado anteriormente en estudios similares. Permite una búsqueda de soluciones global mediante cambios aleatorios y local, mediante cambios en las soluciones más válidas. Siendo así más eficiente que las otras dos opciones.

Por otra parte, no se ha usado nunca en procesos similares, por lo que además aportaría una novedad en el estudio que los dos otros dos algoritmos no pueden proporcionar.

## 5.3.Codificación

#### Definición de las alternativas

Como se ha expuesto anteriormente, se va a optimizar la topología de una red en función de los parámetros establecidos. Para poder optimizar esta topología, es necesario poder almacenar la información de la arquitectura de una manera rápida y eficaz. Para ello, existen diferentes codificaciones que ofrecen mapeos rápidos entre un array con la información y la topología. Entre las diferentes opciones planteadas, todas ellas códigos de Cayley, se encuentran las siguientes:

- Happy code Este código identifica de manera unívoca un código con un grafo en árbol. Los procesos de codificación y decodificación son difíciles de comprender ya que se basa en procesos matriciales. Para este código, el proceso de obtención del grafo a partir del código, es rápido pero el proceso contrario, es bastante lento y computacionalmente lento.
- Blob code Computacionalmente es más sencillo que *Happy Code* y más sencillo de comprender. También identifica de manera unívoca un árbol y un código.
- *Dandelion* code: Es un código que identifica unívocamente el código con la topología en árbol. Los procesos de codificación y decodificación se basan en los procesos de *Happy Code* pero son sencillos de comprender y más sencillos computacionalmente.

### Criterios de ponderación

- Eficiencia del mapeo código-topología (4/10)
- Facilidad de comprensión (3/10)
- Facilidad de programación (3/10)

### Ponderación de las alternativas

En la siguiente tabla se muestra la puntuación que se le ha asignado a cada uno de los criterios definidos en el apartado anterior.

	Código Dandelion	Código Blob	Código Happy
Eficiencia (4/10)	10	9	7
Facilidad de comprensión (3/10)	8	8	6
Facilidad de programación (3/10)	6	6	6
<b>TOTAL:</b>	8.2	7.8	6.6

Tabla 2: Ponderación codificación topología

### Elección final

Entre las diferentes codificaciones posibles, se ha elegido un código Cayley, que poseen una única correspondencia entre árbol y string o código. Entre los diferentes códigos Cayley, se ha optado por una codificación *Dandelion*.

Esta codificación, es la más apta de las propuestas, debido a que por una parte es la más eficaz de las tres y por otra, la modificación de uno solo de sus elementos, puede cambiar la topología entera del árbol, convirtiéndose así más eficiente en un algoritmo evolutivo.

## 5.4.Lenguaje de programación

### Definición de las alternativas

La elección de un lenguaje de programación adecuado es un factor clave en este proyecto, ya que se basa en la programación del algoritmo, las funciones necesarias para el código y los diferentes escenarios y topologías. A la hora de elegir el lenguaje de programación, se ha buscado un lenguaje actual y potente que sea independiente de la plataforma usada. Entre los posibles lenguajes, valorado los siguientes:

- Matlab: Es un entorno de programación potente con un lenguaje de alto nivel interpretado propio. Contiene muchas herramientas para el cálculo científico y visualización gráfica además de una amplia ayuda en línea. Requiere una licencia para su uso.
- Python: Es un lenguaje de programación interpretado diseñado para ser fácil de comprender y simple de implementar. A diferencia de Matlab, es de código abierto. Es uno de los lenguajes más usados hoy en día en la explotación y análisis de datos y debido esto, también cuenta con una extensa ayuda en Internet.
- Perl: es un lenguaje de programación interpretado orientado a scripting que se basa en el estilo de bloques de C aunque también toma características de programación en Shell.

### Criterios de ponderación

Para la elección del lenguaje de programación, se valoraron y ponderaron los siguientes criterios:

- Simplicidad/facilidad de aprendizaje (30)
- Coste (10)
- Conocimientos anteriores(20)
- Capacidad de procesado de datos(40)

### Ponderación de las alternativas

En la siguiente tabla se muestra la puntuación que se le ha asignado a cada uno de los criterios definidos en el apartado anterior.

	python	Matlab	Perl
Simplicidad (3/10)	8	8	6
Coste (1/10)	10	3	10
Conocimientos anteriores (2/10)	3	8	3
Procesado de datos (4/10)	9	6	4
<b>TOTAL:</b>	7.6	6.7	5

Tabla 3: Ponderación del lenguaje de programación

#### Elección final

Se buscaba un lenguaje interpretado, debido a que permite una mayor independencia de la plataforma y aunque en principio sea más lento a la hora de ejecutar que el compilado, con la capacidad de procesado actual, esto no supone un problema.

La elección con mayor peso es python. Por una parte, por la simplicidad y facilidad de aprendizaje, además de la ayuda que hay en línea debido a la actualidad del lenguaje, y por otra, y más importante, por la capacidad de procesado y explotación de analítica de datos. Python es hoy en día el lenguaje más usado en este ámbito, y de cara al posible análisis de logs de la calidad de servicio de los usuarios, se considera el más adecuado.

## 5.5. Resumen

Tras este análisis de alternativas, se trabajará en lenguaje python, mediante el que se programará el algoritmo de optimización Harmony-Search para poder optimizar topologías descritas mediante codificación *Dandelion*.

## 6. Análisis de riesgos

Durante la realización del proyecto es inevitable que surjan problemas que no están previstos y que puedan poner en riesgo el proyecto. Debido a esto, en este apartado de la memoria se enumeran los posibles riesgos y se analiza y valora el efecto que puedan tener. Para esto, se definen dos parámetros. Por una parte la probabilidad de que pase y por otra el impacto que tendría, de esta manera se podrán categorizar y evaluar. En este apartado se identificarán y se ponderarán estos riesgos de la siguiente manera:

- Probabilidad: 1 (bajo), 2(medio), 3(alto)
- Impacto: 1 (bajo), 2(medio), 3(alto)

### 6.1.Incumplimiento de plazos

Debido a una mala planificación, o algún imprevisto, es posible que se incumplan los plazos acordados. Esto puede ser un riesgo alto, ya que al ser un proyecto elaborado por una sola persona, un retraso en una de las tareas puede provocar que todo el proyecto se ve comprometido.

Riesgo: probabilidad: 3. Impacto: 3

### 6.2.Costes imprevistos

Al realizar el proyecto es posible encontrarse con gastos no previstos. La probabilidad de que esto pase es alta, pero en este caso, el impacto que ello produciría en la realización del proyecto es baja.

Riesgo: probabilidad: 3. Impacto: 1

### 6.3.Falta de ayuda/manuales

Como se ha dicho, el proyecto consiste la realización de un programa, por lo que el conocimiento del lenguaje es importante. El hecho de conocer un lenguaje, no garantiza que no vaya a haber ningún problema con la codificación. Un problema de conocimiento del lenguaje, puede traer retrasos, y la falta de manuales o soluciones, puede hacer que este retraso sea significativo.

Riesgo: probabilidad: 3. Impacto: 1



## 6.4. Simulaciones pesadas

Durante el proyecto, se llevarán a cabo varias simulaciones para poder obtener resultados. Debido a la capacidad de computación necesaria para el volumen de datos usado en la implementación del código, las simulaciones pueden resultar pesadas y lentas en los equipos o incluso imposibles de realizar

Riesgo: probabilidad: 3. Impacto: 2

## 6.5. Errores en la codificación

Al ser muchas líneas de código es posible encontrar errores no percibidos anteriormente, que puedan comprometer los resultados.

Riesgo: probabilidad: 3. Impacto: 2

## 6.6. Problemas con la heurística

En este caso, se ha atacado el problema con una heurística simple (Harmony Search multiobjetivo) ya que la estrategia de codificación utilizada permite trabajar con heurísticos básicos. Es posible que tras la codificación, los resultados obtenidos no sean satisfactorios y que sea necesario modificar el algoritmo utilizado

Riesgo: probabilidad: 1 Impacto: 3

## 6.7. Conclusiones no significativas

Al tratarse de un proyecto de investigación y no de la descripción o implantación un procedimiento, se corre el riesgo de que las conclusiones no sean significativas. Es posible que los resultados obtenidos no aporten información sobre la validez del algoritmo a la hora de evaluar las redes.

Riesgo: probabilidad: 2. Impacto: 2

## 6.8. Parámetros del escenario no realistas

Como se trata de una simulación en un entorno creado, se corre el riesgo de que los parámetros utilizados no sean realistas, lo que afectaría a las conclusiones obtenidas de los resultados. Se tratará de implementar la simulación en un escenario real, pero es posible cometer errores en la elección de los valores de los parámetros.

Riesgo: probabilidad: 2. Impacto: 2

## 6.9. Problemas con la codificación Dandelion.

El código *Dandelion* (Explicado más adelante) es un código conceptualmente sencillo, pero difícil de programar. Esto hace que la programación sea un proceso de prueba y error en el que pueda faltar algún caso.

Riesgo: probabilidad: 3. Impacto: 2

Teniendo en cuenta todos los riesgos y su impacto en función de la ponderación que se les ha dado, se muestra el diagrama en el que se clasifican:

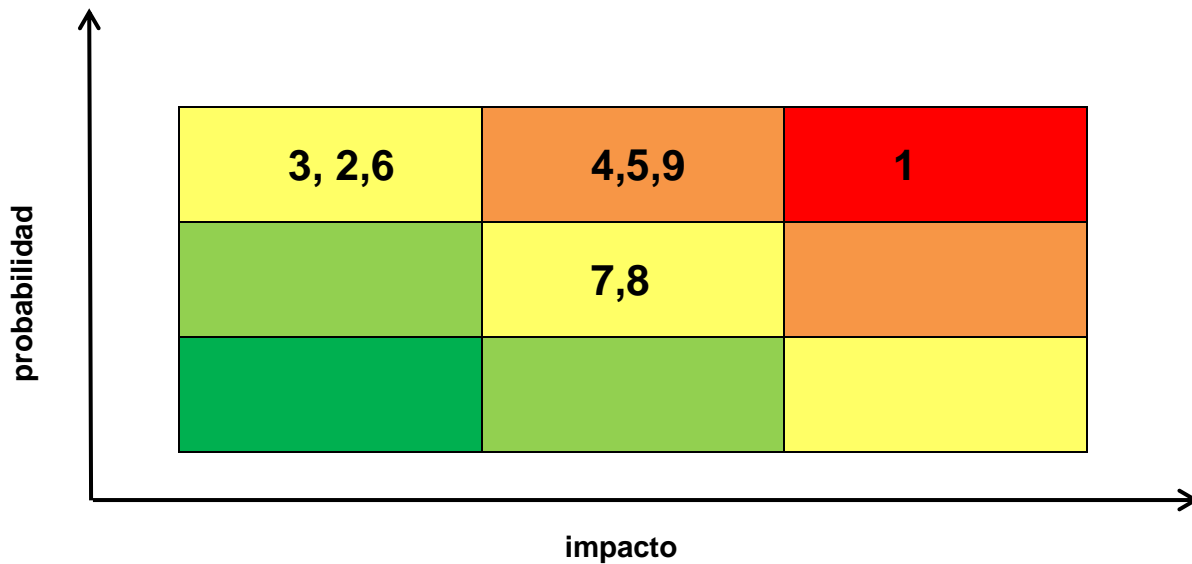


Ilustración 5: Diagrama de riesgos

## 7. Metodología

En este apartado se describirá el procedimiento llevado a cabo y la explicación teórica necesaria para su comprensión, separándolo en las diferentes partes en las que se divide el proyecto.

### 7.1. HS y codificación Dandelion

#### Algoritmos evolutivos:

Los algoritmos evolutivos son métodos de optimización de soluciones iterativos basados en la teoría de la evolución. Su funcionamiento se basa en la optimización de un conjunto de soluciones, de las cuales se evalúa el grado de aptitud y a medida que el algoritmo avanza se mantienen las soluciones más válidas, desechando las no válidas con el objetivo de conseguir el mejor resultado posible.

Se suelen usar para solucionar problemas que no tienen una solución inmediata ya que al ser iterativos, a medida que avanza el algoritmo nos permiten mejorar el resultado obtenido para el problema enunciado. Este tipo de algoritmo es aplicable en todos los ámbitos, pero en este proyecto se va a utilizar para optimizar una topología de red en base a dos criterios establecidos.

Como se ha mencionado, estos algoritmos son especialmente útiles cuando no es fácil hallar una solución inmediata a un problema. Cada iteración nos permite encontrar una mejor solución hasta cumplir un objetivo, que se establecerá antes de ejecutar el algoritmo y generalmente suele ser un máximo de iteraciones o un valor que cumpla las expectativas.

Para encontrar los resultados, los algoritmos evolutivos trabajan con una lista de posibles soluciones a un problema denominada población. Esta población, se modifica y transforma durante el algoritmo para ir mejorando el conjunto de soluciones. Cada ciclo de mejora y transformación se denomina generación y cada una de las soluciones, individuo. De esta manera, tras varias generaciones se espera que la aptitud de la población sea mejor y conseguir el individuo óptimo.

#### Funcionamiento de los algoritmos evolutivos:

Los algoritmos evolutivos reciben ese nombre porque son estrategias de búsqueda estocástica basados en el mecanismo de selección natural. Se parte de una población inicial que está compuesta de una serie de soluciones o individuos. Esta población inicial, puede ser escogida de manera aleatoria o partiendo de un conjunto de soluciones que ya se consideran aptas. La evolución en el algoritmo se aplica en los individuos a través de las diferentes generaciones y cada una de estas generaciones corresponde a una iteración.

Para obtener las diferentes generaciones se combinan la búsqueda aleatoria y la búsqueda dirigida. Una nueva generación se consigue partiendo de la población anterior. Se clasifican los individuos en función de su validez sobre la solución buscada. Es decir, se evalúa la aptitud de la solución respecto al objetivo o criterio establecido inicialmente. Las soluciones más válidas son las que pasarán a la

siguiente generación mediante una operación aleatoria que se llama transformación o mutación.

El mecanismo que se usará para esta mutación dependerá del algoritmo utilizado, ya que cada uno tiene su propio procedimiento de generación de nuevos individuos. Esto es lo que diferencia la potencia y validez de un algoritmo.

El número de individuos tiene que ser constante durante todo el proceso de obtención de la solución. Por tanto, a través de las diferentes generaciones, los individuos con mayor grado de aptitud tendrán mayores posibilidades de ser seleccionados para la posterior generación. Mejorando así la calidad global de la población.

En cada iteración se genera un conjunto nuevo de soluciones aleatorias y se vuelve a evaluar la aptitud de todas ellas. Muchas de estas soluciones, creadas aleatoriamente, no serán válidas o serán menos aptas que las iniciales por lo que se descartarán directamente. Sin embargo, algunas serán válidas y se guardarán para transformarlas para generaciones posteriores. Las soluciones que han empeorado o no han mejorado con los cambios serán eliminadas

Este proceso se realiza varias veces hasta que se consiguen los resultados esperados o se cumple el máximo de iteraciones establecido. Se espera que la calidad de las soluciones de la población sea mejor a medida que se va avanzando en el algoritmo. De esta manera, con muchas repeticiones, se puede conseguir un resultado óptimo.

### Algoritmo HS

El algoritmo *Harmony Search* se encuentra dentro de los algoritmos evolutivos y recibe su nombre porque el mecanismo que usa para generar las nuevas soluciones se asemeja a la improvisación musical.

Este algoritmo apareció en 2001 como un algoritmo meta-heurístico nuevo y en los últimos años ha conseguido una alta relevancia en diferentes campos. La mayor diferencia con otros algoritmos evolutivos es la manera de conseguir nuevos valores. La mayoría de algoritmos evolutivos pertenecen a la familia de algoritmos bio-inspirados y consiguen los nuevos valores simulando procesos biológicos que suceden de manera natural. El algoritmo HS por el contrario, usa un mecanismo parecido al que usan los músicos para buscar armonía y el proceso de obtención de nuevos valores se describe mediante la siguiente analogía:

En el algoritmo HS, cada músico (variable a optimizar) toca (genera) una nota (valor) para conseguir una armonía perfecta (solución óptima).

La improvisación musical es un proceso en el que se busca la mejor armonía probando varias combinaciones de notas que deben seguir las siguientes reglas:

1. Tocar una nota de la memoria
2. Tocar una nota adyacente a una nota de la memoria
3. Tocar una nota elegida aleatoriamente del rango disponible

Este proceso se imita en cada selección del algoritmo HS. Por tanto, las tres reglas anteriores se podrían traducir de la siguiente manera:

1. Coger un valor de HS *memory*
2. Coger un valor adyacente a un valor de HS *memory*
3. Coger un valor aleatorio dentro del rango disponible

Funcionamiento de HS:

Para la ejecución de este algoritmo será necesario definir los siguientes parámetros:

- HM (*Harmony memory*): Matriz que guarda las diferentes soluciones candidatas.
- hms (*harmony memory size*): tamaño de la memoria en la que se guardan los resultados.
- hmcr (*harmony memory considering rate*): Probabilidad de elegir un valor del *harmony memory*.
- Par: probabilidad de coger un valor adyacente.

**Paso 1:** Inicialización de *Harmony memory (HS memory)*. La memoria inicial se inicializa con unas soluciones generadas de forma aleatoria dentro del rango posible. Estas soluciones candidatas serán vectores con valores aleatorios  $x_n=(x_1, \dots, hms)$ . Para una memoria de tamaño hms y dimensión n, la matriz inicial se representará de la siguiente manera.

$$\begin{bmatrix} x_1^1 & \dots & x_n^1 \\ \vdots & \ddots & \vdots \\ x_1^{hms} & \dots & x_n^{hms} \end{bmatrix}$$

**Paso 2:** Generar una nueva solución  $x'=(x'_1, x'_2, \dots, x'_n)$  por cada componente x de la memoria HS. Esta solución se obtendrá en función del parámetro *hmcr*. Como se ha mencionado, *hmcr* es la probabilidad de seleccionar un elemento de HS *memory*, y por tanto,  $1 - hmcr$  la probabilidad de generarlo de manera aleatoria. Matemáticamente se expresaría de la siguiente manera.

$$x'_i \leftarrow x_i^{int(u(0,1)*hms)+1}$$

Si  $x'_i$  proviene de HM, se selecciona un vector correspondiente a una solución que puede ser modificado en función del parámetro par. Este parámetro par determina la probabilidad de que una solución candidata mute. Una solución con una mutación es un vector seleccionado de la memoria anterior al que se le aplicará un cambio, es decir, se sustituirá por un valor adyacente:  $x'_i \leftarrow x'_i \pm \delta$

**Paso 3:** Actualizar HS memory. Para actualizar el conjunto de soluciones, primero se evalúan las soluciones conseguidas en el paso anterior. Si alguna de las soluciones es mejor que la peor solución anterior, se reemplazará la peor solución por la nueva. En el caso contrario se descartará la nueva solución.

**Paso 4:** Repetir los pasos 2 y 3 hasta llegar al criterio establecido. Puede ser un número máximo de iteraciones o un valor de solución que cumpla las expectativas.

Para poder entenderlo de una manera más clara, se muestra un diagrama de flujo para explicar el funcionamiento del algoritmo:

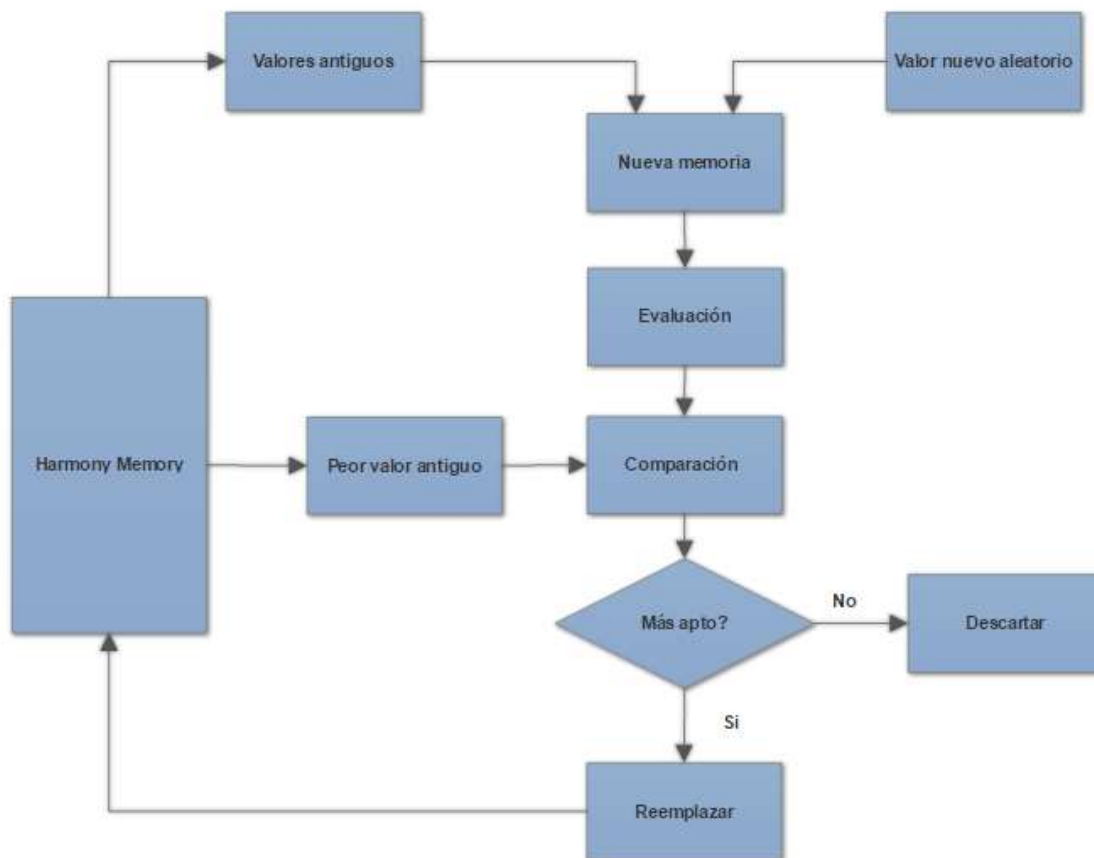


Ilustración 6: Funcionamiento de Harmony Search

En este proyecto para el algoritmo HS descrito, cada solución candidata será un código *Dandelion* que corresponda a una topología concreta de la red. En cada iteración se evaluará la validez de la red propuesta teniendo en cuenta dos parámetros. Como criterio para finalización del algoritmo se ha establecido un máximo de iteraciones. Por tanto, en cada iteración se irán modificando los códigos *Dandelion* de la memoria en función de los parámetros establecidos y su aptitud respecto a ellos; y se finalizará cuando se llegue al máximo de las iteraciones.

Para la evaluación de la aptitud de las soluciones se tendrá en cuenta el beneficio del operador y el retardo experimentado por el usuario. Por lo cual, como se dispone de dos parámetros para establecer la calidad o validez de una solución, la optimización que se llevará a cabo será multiobjetivo.

### Optimización multiobjetivo.

En el caso de un algoritmo con un parámetro a optimizar, se tiene en cuenta un solo criterio a la hora de valorar la aptitud de una solución concreta. En el caso de que haya dos o más variables que evaluar (para este proyecto el beneficio del operador y la calidad del servicio para el usuario), la optimización debe ser multiobjetivo. Cuando se tiene más de un parámetro a optimizar, no se busca una solución concreta, sino un conjunto de soluciones óptimas. Esta optimización es la que dentro del algoritmo *Harmony Search* nos permite evaluar las soluciones candidatas mencionadas en el paso 3.

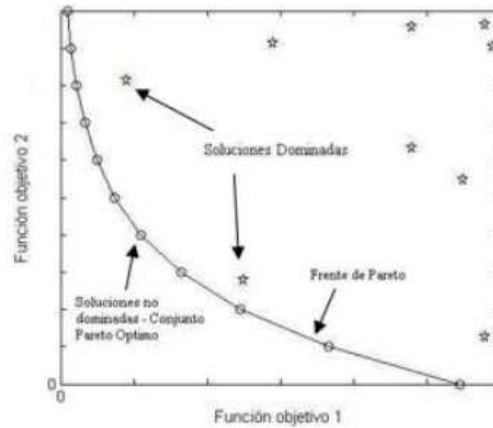
Cada uno de los parámetros a optimizar se llama objetivo. Hay algunos casos en los que una solución es la óptima para todos los objetivos, pero la mayoría de las veces, no coinciden. Es decir, no existe una solución  $x$  en la que todos los objetivos sean optimizados simultáneamente; y generalmente, cuando uno de los objetivos se maximiza el otro se minimiza. Debido a esto es necesario contar con un conjunto de soluciones aunque no todas las soluciones serán igual de válidas para resolver el problema planteado.

En el caso de que se tengan varias soluciones, es necesario decidir cuáles son las más aptas y así descartar las demás. Para determinar la aptitud de una solución, se define el concepto de dominancia, que permite identificar las mejores soluciones. Para saber si una solución es dominante sobre otras se aplica el siguiente criterio:

Teniendo dos soluciones,  $x_1$  y  $x_2$ ,  $x_1$  será dominante sobre  $x_2$  si:

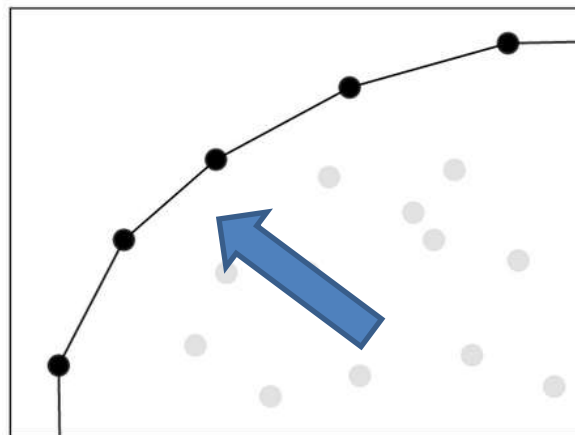
- $x_1$  no es peor que  $x_2$  en todos los objetivos
- La solución  $x_1$  es mejor que  $x_2$  en al menos un objetivo.

Para conseguir tener un conjunto de soluciones que satisfaga las necesidades del problema, es imprescindible que ese conjunto de soluciones sean dominantes o no dominadas. A este conjunto de soluciones no dominadas se le llama frente de Pareto. Este frente de Pareto, se forma uniendo las diferentes soluciones no dominadas, creando una curva como la que se puede ver en la ilustración 7.



**Ilustración 7: Gráfico frente de Pareto**

El cálculo de la dominancia entre las soluciones en el algoritmo, se aplica en la evaluación de la aptitud de cada una de ellas, de manera que podamos mantener las más válidas y descartar el resto para la siguiente iteración. Este conjunto de soluciones óptimas o frente de Pareto no siempre mantiene la misma curvatura que la imagen del ejemplo. Esta curva variará en función de los parámetros a evaluar y del valor de cada uno de los ejes de la gráfica. En este caso, se buscará maximizar el beneficio (eje y) y minimizar el retardo (eje x). Por tanto, el frente de Pareto convergerá de la siguiente manera.

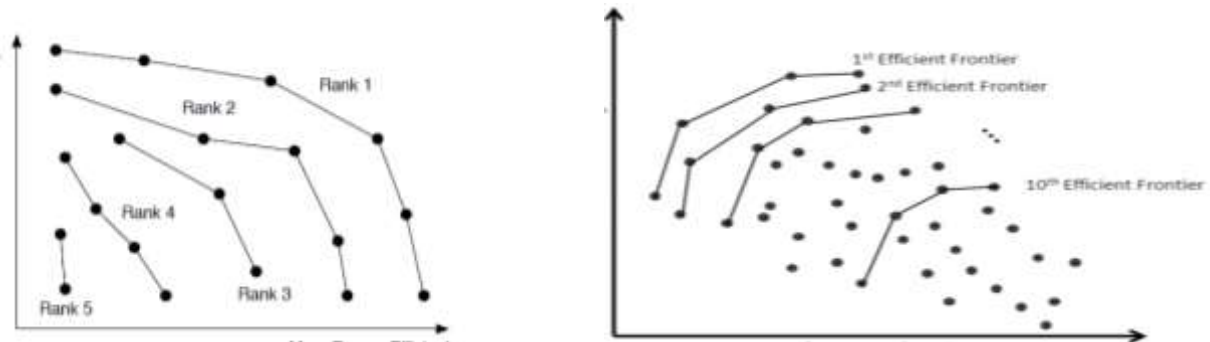


**Ilustración 8: Convergencia de frente de Pareto**

A medida que el algoritmo va avanzando, se irán obteniendo diferentes soluciones que irán formando este frente de Pareto y convergiendo cada vez más hacia un conjunto óptimo. En la memoria del algoritmo, se guardarán más resultados además de los que forman el frente. Para clasificar estos resultados, uno de los criterios utilizados son los rangos. Estos rangos están formados por soluciones no dominadas entre ellas y se nombran de forma ascendente en función de lo válidas que sean las soluciones que lo forman. Es decir, Las soluciones con  $r = 1$  serán más válidas en todos los aspectos que las de  $r = 2$  y así sucesivamente hasta clasificar



todas las soluciones. En las siguientes imágenes se muestra un gráfico con diferentes soluciones en las que se ve el rango asignado a cada una de ellas.

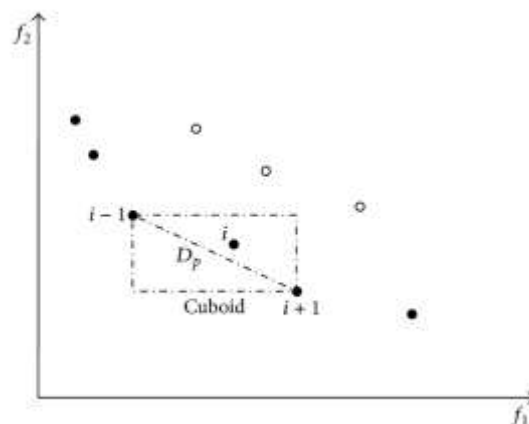


**Ilustración 9: Gráficas con clasificación por rangos**

En función de lo que convenga para cada problema en la memoria del algoritmo se guardarán las diferentes soluciones. En algunos casos interesará mantener todas las soluciones de  $\text{rango}=1$  y en el caso de que no se complete toda la memoria, se irán guardando el resto de soluciones en orden ascendente de rango hasta llenarla. Otras veces, interesa guardar soluciones de diferentes rangos para realizar una mayor exploración. En este trabajo, se han priorizado las soluciones con menor rango.

En el caso en el que haya más soluciones de un rango concreto que memoria, será necesario categorizarlas para poder conseguir las que más información y capacidad de exploración nos puedan aportar sobre la solución ideal. Para esto, se utiliza la *crowding distance*. Este parámetro se define como la distancia entre dos posibles soluciones.

Si las soluciones tienen una mayor distancia entre ellas, la información que aportan para la formación del frente es mayor que si todos los puntos se encuentran cerca. Es decir, a la hora de trazar la línea que forma el frente de Pareto, en el caso de que todas las soluciones sean próximas entre ellas no se podrá trazar una curva. En cambio, si se encuentran a mayor distancia será más sencillo. Para comprender esto, se muestra la siguiente imagen.



**Ilustración 10: Crowding distance**

Como se puede ver en la ilustración 10, hay seis soluciones de rango=1 y tres de rango=2. Si el tamaño de la memoria fuera 9, se guardarían todos los resultados. En el caso de que el tamaño sea 6, únicamente se guardarían los de rango=1. Pero si el tamaño de la memoria fuera menor de 6, habría que descartar alguna solución. En estos casos se usa la *crowding distance*.

Esta distancia es la distancia entre dos puntos próximos. En el ejemplo de la imagen, DP sería la *crowding distance* entre  $(i-1)$  e  $(i+1)$ . En este caso, si se quiere descartar una solución, se calcularían las distancias entre todos los puntos y se suprimirían las que tienen una distancia menor de manera que se pueda trazar una curva más exacta. Los dos resultados de los límites tienen una *crowding distance*=infinito, por lo que siempre estarán en el frente de Pareto formado.

### Dandelion:

Como ya se ha mencionado, las soluciones que habrá en la memoria HS serán códigos *Dandelion*. En este apartado se explicará qué es este código y por qué se ha usado en este proyecto, además de los procesos de codificación y decodificación.

#### 7.1.1. Definición

El código *Dandelion* es un tipo de código *Cayley* que tiene como característica principal el mapeo biyectivo entre un conjunto de árboles etiquetados de  $n$  nodos con  $n-2$  nodos etiquetas.

Gracias a su característica biyectiva en general, la familia de códigos *Cayley* tiene varias virtudes que hacen que sea especialmente adecuada para crear estructuras mediante algoritmos evolutivos. Aunque solo hay algunos códigos de *Cayley* que tengan una única correspondencia árbol - string. Estos códigos con correspondencia única son los vistos en el análisis de alternativas.

Los tres códigos, *Blob code*, *Happy code* y *Dandelion code* son totalmente aptos para su utilización en algoritmos evolutivos porque al ser representados mediante un *string* son fácilmente modificables, lo que permite poder evaluar la validez de diferentes árboles de una manera sencilla.

Entre estos tres códigos, el más eficiente computacionalmente es *Dandelion*. Que como ya se ha mencionado, representa un árbol de  $n$  nodos mediante un código de una longitud de  $n-2$  y este código, se corresponde de manera unívoca con una topología en árbol, que se puede obtener mediante los siguientes procedimientos:

### 7.1.2. Decodificación

En la siguiente imagen, se muestra una topología en árbol con su correspondiente código *Dandelion*.

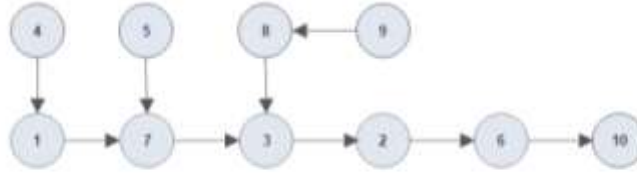


Ilustración 11: Topología árbol.  $C=(6, 2, 1, 7, 3, 7, 3, 8)$

#### Paso 1:

Partiendo del código *Dandelion*, se crea una matriz  $A_c$  de  $2 \times n-2$  siendo  $n$  la cantidad de nodos. La primera fila debe contener números sucesivos,  $\{2, 3, 4, \dots, n-1\}$ , La segunda fila contiene el código *Dandelion* asociado al árbol. El resultado será una matriz así:

$$A_c = \begin{bmatrix} 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 6 & 2 & 1 & 7 & 3 & 7 & 3 & 8 \end{bmatrix}$$

#### Paso 2:

Se definirá de la siguiente manera:  $f_c: [2, n-1] \rightarrow [1, n]$  de manera que  $f_c(i) = C_i$  para cada  $i \in [2, n-1]$ .

#### Paso 3:

Se denominará a los ciclos de la siguiente manera:  $\{Z_1, Z_2, Z_3, \dots, Z_L\}$ .

En el ejemplo, se pueden encontrar dos ciclos: (7) y (6, 3, 2).

Teniendo en cuenta que  $b_l$  denota el máximo elemento en  $Z_l$  (siendo  $l \in \{1, 2, \dots, L\}$ ), los ciclos se reordenan de manera que  $b_l$  sea el elemento más a la derecha de  $Z_l$  y que  $b_l > b_{l'}$  si  $l < l'$ . Dicho de otra manera, se girarán los ciclos de manera que el ciclo con el elemento mayor se coloque a la izquierda y dentro de los ciclos, el elemento mayor deberá estar a la derecha del ciclo.. Por tanto, quedarían los ciclos de esta manera:

$$\{Z_2, Z_1\} = \{(7), (3, 2, 6)\}$$

**Paso 4:**

Se listan los ciclos identificados en el paso anterior en el orden correspondiente, añadiéndoles el primer y último nodo, de la siguiente manera:

$$\pi = \{(1)(7)(3\ 2\ 6)(10)\}$$

Con esto, se consigue la estructura principal del árbol:



**Ilustración 12: Tronco árbol**

**Paso 5:**

Crear el árbol T correspondiente a C, añadiendo los nodos que faltan, según el matriz Ac, hasta conseguir el árbol de la ilustración 11

**7.1.3. Codificación**

Aplicando el algoritmo de forma inversa, es posible conseguir el código a partir de un árbol. En este ejemplo, se usará la misma topología que en el ejemplo anterior.

**Paso 1:**

Listar los nodos intermedios del *path* de 1 hasta N de T.

Siguiendo el mismo ejemplo que en el apartado anterior, conseguiríamos la siguiente lista: 7, 3, 2, 6.

**Paso 2:**

Partir la lista en ciclos buscando los elementos límite de los ciclos (mencionados en el paso3 de la decodificación). Los elementos límites para  $\pi$  será por tant 7 y 6, consiguiendo así 2 ciclos:

(7) y (3, 2, 6)

**Paso 3:**

Se construye la matriz Ac, en la que se insertarán los ciclos del apartado anterior. Obteniendo la siguiente matriz:

$$A_c = \begin{bmatrix} 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 6 & 2 & - & - & 3 & 7 & - & - \end{bmatrix}$$

#### Paso 4:

Añadir el resto de nodos al código. En este último paso, mirando a las conexiones del árbol, se rellenarán los huecos que faltan en la matriz  $A_c$ . Consiguiendo así la misma matriz que en la decodificación.

$$A_c = \begin{bmatrix} 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 6 & 2 & 1 & 7 & 3 & 7 & 3 & 8 \end{bmatrix}$$

La segunda fila de la matriz  $A_c$  es la que se corresponde el código. De esta manera se consiguen las codificaciones *Dandelion*.

### 7.2.Reparación del árbol

En el algoritmo HS, los *Dandelion* se crean de manera aleatoria (tanto el código entero como la variación respecto del código anterior), por lo que es necesario comprobar que el código creado se corresponde con una topología en árbol.

Para esto, se creará una función que compruebe que cuando el código se transforma en grafo, éste es un árbol y que cumple las características necesarias fijadas para cada uno de los escenarios.

En todos los escenarios, por ejemplo, es necesario que las estaciones Base de los operadores no se conecten entre ellas pero que estén conectadas a la estación base principal. Para esto, en la decodificación se creará una matriz de conexiones en la que las posiciones conectadas se pondrán a '1' y el resto a '0'. Para cada código creado se comprobará que los valores correspondientes a las conexiones entre estaciones base sea '0' y la conexión entre BS y la estación de *backhaul* sea '1'. Además de este requisito, es necesario que ninguno de los nodos correspondientes a los dispositivos de usuario se conecten directamente a la estación de *backhaul* y que la topología creada sea válida, por lo que también se corregirán estos casos.

Para la corrección, se crearán varias funciones; por una parte, la correspondiente a la corrección de la topología, es decir, se asegura que no haya bucles en el código creado y que las conexiones establecidas son factibles en cuanto a los radios de cobertura. Y por otra parte, la corrección de las conexiones entre las estaciones base y la estación de *backhaul*. Para las dos funciones se definirá una matriz adicional en la que se marcarán las conexiones posibles con '1' y las obligatorias con un '2'. El resto de conexiones serán '0'.

Primero se comprobará que las conexiones establecidas son posibles y se creará una lista de las conexiones con error. Para estas conexiones, se buscará aleatoriamente en la matriz de conexiones viables una conexión posible y se modificará el código. Una vez modificados todos los nodos de usuarios, se comprobarán las conexiones de las estaciones base y en el caso de que sea necesario se modificarán. Se pasará ahora a explicar mediante un ejemplo el procedimiento de corrección de árboles seguido:

Suponiendo una red de 10 nodos numerados de 0 a 9 en la que el nodo 9 será la estación de *backhaul* y los nodos 7 y 8 las estaciones base de los operadores, teniendo la siguiente matriz de conexiones, se formaría el árbol de la ilustración 13:

$$M_{CX} = \begin{bmatrix} 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

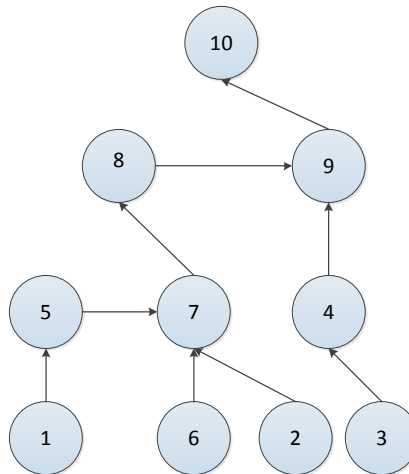


Ilustración 13: Árbol inicial

En la matriz de conexiones, las posiciones con '1' corresponden con las conexiones establecidas. La posición  $M_{CX_{i,j}}$  es la conexión del nodo  $i+1$  en sentido ascendente con el nodo  $j+1$ . En este caso, el nodo 1 está conectado con el 5, como se muestra en el árbol.

Además de la matriz de conexión, para saber si las conexiones establecidas son posibles, se define la matriz de conexiones viables. En esta matriz, cuando un nodo se encuentra dentro del radio de cobertura de otro la posición correspondiente se marca con un '1'. Por tanto, esta matriz será simétrica. Además, en esta matriz hay varias conexiones que son obligatorias, por lo que se marcarán con un '2'. Para continuar con el ejemplo se definirá de la siguiente manera:

$$M_{viable} = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 2 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 0 \end{bmatrix}$$

Si comparamos las dos matrices se encuentran posiciones para las que en la matriz de conexiones hay un valor '1' sin embargo, en la matriz de conexiones viables, hay un valor '0'. Para estos casos, se procederá a reparar el árbol.

Primero, se buscará la conexión no válida. En este caso, la correspondiente a los nodos 2,5 y 8. Después, para cada uno de los nodos se hará una lista de las posibles conexiones:

Para el nodo 5 esta lista será [1,6,8,9]. En este caso el nodo 1 está conectado al 5, por lo que se descartaría para no formar bucles. En cuanto a los tres nodos restantes, se cogerá aleatoriamente uno de la lista. Por ejemplo, 8. El mismo proceso se repetirá para el nodo 2.

Para el nodo 8, la situación es diferente, En este caso, está conectado a la otra estación base en vez de a la estación de backhaul, por lo que directamente se cambiará la conexión.

Una vez realizados estos cambios, se conseguirá la siguiente matriz de conexiones:

$$M'_{CX} = \begin{bmatrix} 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

En cuanto a la topología en árbol asociada a la nueva matriz, es la siguiente:

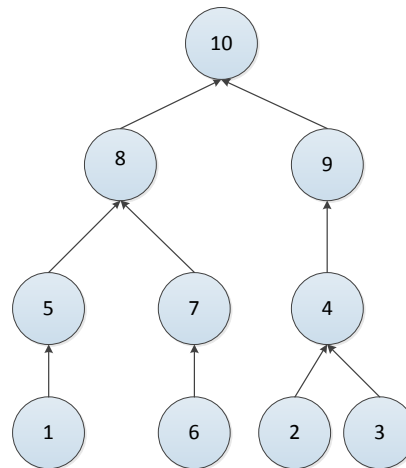


Ilustración 14: Árbol corregido

### 7.3. Evaluación de la calidad.

La calidad es uno de los parámetros usados para la optimización. Por ello, es necesario evaluarla y darle un valor para poder comparar los Para poder calcular la calidad experimentada por el usuario, se ha creado otra función, que devolverá la calidad medida en el número de slots medio que tiene que esperar un usuario desde que tiene la información lista para transmitir hasta que llega a la estación base de uno de los dos operadores.

Para obtener este valor se calcularán el mejor (*Best Case*) y el peor (*Worst Case*) caso y se obtendrá la media de ambos, de manera que:

- *Best Case*: En este caso, el nodo tendrá los recursos libres para poder transmitir en el momento en el que tiene preparada la información. Al igual que este nodo, todos los nodos siguientes hasta llegar a la estación base tendrán el slot correspondiente libre. Por tanto,

$$r = \text{Número de saltos hasta la BS.}$$

- *Worst Case*: En este caso se supondrá que los recursos asignados al nodo estarán ocupados y que el tiempo de espera será el mayor posible. Por tanto,

$$r = \begin{cases} \text{Si } BC = 1: r = \text{num nodos del nivel} * (\text{numHijos} + 1) \\ \text{Si } BC > 1: r = r_{\text{padre}} * (\text{numHijos} + 1) \end{cases}$$

Haciendo la media de los dos valores obtenidos, se conseguirá el retardo de cada uno de los nodos, y para poder cuantificar la calidad de la red entera, se calculará la media de todos los nodos de la red.



## 7.4. Evaluación del beneficio

En el caso del beneficio, se calcularán las ganancias del operador. Para esto, por una parte se obtendrán en cuenta los ingresos del operador, que se obtendrán por los contratos de los clientes y por otra parte, tendremos los gastos que le supone al operador dar servicio a sus usuarios. Estos gastos dependerán de tres factores:

- **Costes de conexión directa:** Cada usuario conectado directamente a la antena del operador, le supone un coste en debido a los recursos que tiene que mantener activos para poder dar servicio al usuario.
- **Costes/Incentivos por tethering:** Cada usuario que tiene un dispositivo conectado mediante *tethering*, le ahorra al operador los gastos por conexión directa, pero le presta sus propios recursos por lo que recibirá un incentivo. Este incentivo, será menor que el gasto por conexión directa.
- **Coste de conexión a otro operador:** Se va a permitir conectar usuarios a otros operadores y esta conexión también le supondrá un coste al operador, ya que se le cobrará por usuario conectado.

Teniendo en cuenta los parámetros definidos, el beneficio del operador se calculará de la siguiente manera:

$$\text{Beneficio} = \text{PrecioContrato} * \text{Num\_contratos} - (\text{CosteCxDirecta} * \text{nodos\_directos} + \text{CosteNodoTethering} * \text{nodos\_tehtering} + \text{CosteCxOtrosOperadores} * \text{Nodos\_a\_otro\_operador}).$$

## 7.5. Creación de los escenarios diferenciados

Tras tener un código en el que se calculen los beneficios y retardo de cada uno de los nodos, y el algoritmo de optimización, se pasará a crear un escenario multioperador.

Como se ha comentado en la descripción de la solución, se ha optado por crear dos escenarios diferenciados para poder recrear las diferentes situaciones que se pueden dar en la realidad. En ellos, se analizarán diferentes topologías y alternativas de precios, nodos y políticas de relaciones entre operadores.

### Escenario 1: Optimización conjunta

Una de las opciones que se ha planteado, es una topología en la que exista un elemento central que tenga conocimiento sobre la situación de toda la red que tiene por debajo, y que por tanto, pueda gestionarla en su totalidad. En otras palabras, se creará una estación base principal a la que se conectarán las dos BS de los operadores. Y debajo de estas dos, se conectarán todos los usuarios, tanto los correspondientes al operador 1 como al operador 2.

En este escenario, las estaciones Base se han colocado en el escenario de forma fija. La estación base principal, se coloca en el centro del área creada (Las dimensiones de esta área, son parámetros que se indican en el inicio del programa) y las dos BS de los operadores, a la misma altura, a media distancia entre el límite del área y la estación principal. Los nodos que simularán usuarios de ambos operadores (N1 y N2), estarán distribuidos al azar por todo el escenario. La cantidad total de nodos desplegada es también un parámetro modificable al inicio del programa, aunque la cantidad de nodos correspondientes a cada operador y la posición de éstos, sea aleatoria.

Este escenario se ha planteado de esta manera, porque el elemento que ejecutará el algoritmo será la estación principal, de manera que se intentará optimizar toda la red. Es decir, se buscará una optimización de los parámetros de ambos operadores, consiguiendo así un beneficio máximo en conjunto.

En la siguiente imagen se puede ver un diagrama de la arquitectura de este escenario. En gris se indicarían los nodos correspondientes al operador 1 y en azul los nodos del operador 2.

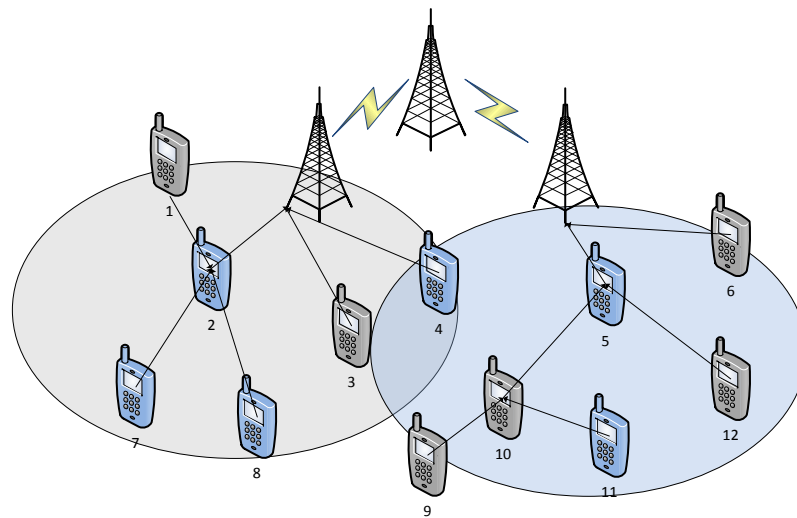


Ilustración 15: Escenario optimización conjunta

Como se ve en la imagen, La posición de los nodos, es independiente al área de cobertura de la estación base de su operador; hay nodos del operador 1 que se puedan encontrar en el radio de cobertura del operador 2 y viceversa.

## Escenario 2

En este segundo escenario, al contrario que en el anterior, no hay ningún elemento que controle toda la red. En este caso, se crearán dos topologías totalmente independientes la una de la otra, de manera que los dos operadores estén diferenciados.

Las estaciones Base de los operadores se colocarán en el mismo sitio que en el caso anterior, y los nodos se colocarán de manera aleatoria. En este caso, es posible fijar la cantidad de nodos correspondiente al operador 1, ya que es el que ejecutará el algoritmo. Una vez asignada la posición de cada nodo, se construirán dos topologías independientes, una para cada operador.

El operador 1 ejecutará el algoritmo de optimización sobre sus propios nodos, asumiendo que el operador 2 mantiene una topología fija. Esto, permite que el operador 1 modifique su arquitectura pero no pueda influir en la arquitectura del operador 2, aunque se le permita poder conectar nodos a cambio de un coste establecido.

Como se ha dicho, el operador 1 no tendrá información sobre topología del operador 2, pero sabrá la calidad que les garantiza a sus nodos, es decir, se asumirá una calidad fija para los nodos conectados al operador 2, dependiendo de la arquitectura que tenga en el momento inicial. Se ha diseñado de esta manera, porque al ser dos operadores que gestionan sus redes independientemente, el operador 1, no debe tener conocimiento sobre el estado de la red del operador 2. Por tanto, en una aproximación se asumirá una calidad fija para todos los nodos conectados a otro operador.

En la siguiente imagen, vemos un diagrama de esta red. En este caso, sería la arquitectura inicial, como se aprecia, únicamente se tiene conocimiento de las conexiones establecidas de los propios usuarios, la tipología de la red del otro operador es totalmente desconocida.

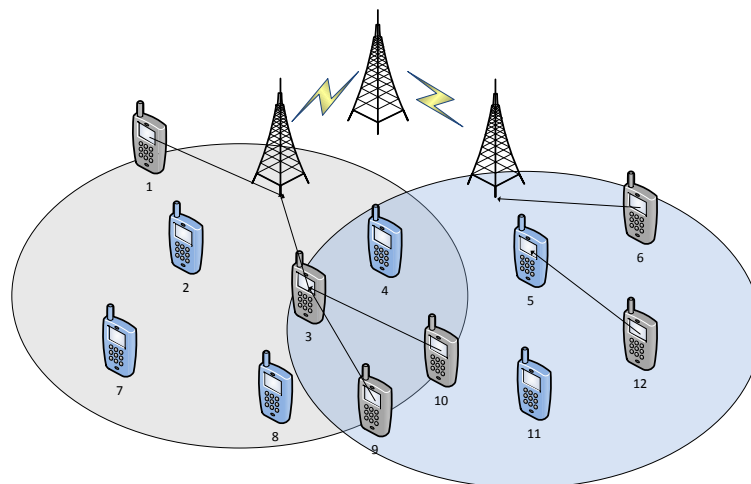


Ilustración 16: Escenario gestión independiente

Debido a la naturaleza de código *Dandelion*, debe tener una longitud fija para todas las iteraciones, por tanto, se añadirá un nodo más a la topología del operador 1, al que se conectarán todos los nodos que deberían estar conectados a otro operador. Analíticamente los nodos estarán conectados a otro operador pero debido a esta cualidad, en lo que se refiere a la codificación y la topología, estarán conectados a este nodo extra. Los nodos conectados al comodín tendrán, por tanto una calidad fija, correspondiente al operador 2 y un coste fijo, dependiendo del precio establecido por el otro operador.

Además, debido a que hay dispositivos conectados a otros operadores que pueden tener en *tethering* otro nodo, el precio por la conexión se contará como conexión a otro operador.

## 7.6. Pruebas realizadas.

Se han diseñado dos pruebas diferentes para poder aplicar a los dos escenarios credos. Por una parte, se realizará una prueba en la que la red se gestionará de manera conjunta entre los operadores desplegados en el área. Por otra parte, se realizará una optimización independiente en la que se estudiará uno solo de los operadores.

Además, se realizarán pruebas con diferente cantidad de usuarios en los operadores, para poder evaluar la influencia de éstos en los resultados. Es decir, si la calidad y el beneficio aumentan de manera lineal o proporcional al aumento de usuarios o de una manera no lineal.

### 7.2.1. Optimización conjunta (backhaul)

Esta prueba es la correspondiente al escenario 1 del apartado anterior. En este caso, se ha decidido que los dos operadores gestionarán la red de manera conjunta, buscando así el máximo beneficio para los ambos. Los parámetros que se intentarán optimizar, son el retardo medio y la calidad media de los dos operadores. Por tanto, en lugar de ejecutar el algoritmo en las estaciones de los operadores, se ejecutará en el elemento central, ya que tiene conocimiento del estado de toda la red y es el único que puede gestionarlo.

Esta optimización conjunta trabajará con la media de las ganancias de los dos operadores así como con la calidad media de todos los nodos desplegados en el escenario.

### 7.2.2. Optimización independiente

Para esta segunda prueba, no hay ningún elemento que tenga conocimiento de toda la red. Además, como se ha mencionado en el apartado de la creación de escenarios, solamente se tendrá control sobre uno de los operadores. Se sabrá la posición de los nodos del otro operador y se podrá conectar a ellos, pero no se podrá modificar su arquitectura en beneficio del operador que ejecuta el algoritmo. En este

caso, se definirán dos estrategias diferentes en función del valor del parámetro  $\alpha$  definido de la siguiente manera:

$$\alpha = \frac{\text{Coste usuario propio}}{\text{Coste usuario externo}}$$

En el caso de que  $\alpha \approx 1$ , la estrategia a seguir será coalición. Es decir, el coste por cada usuario conectado a otro operador será similar a conectarlo en la propia red. De esta manera, se facilita la conexión con otros operadores.

En el caso de que  $\alpha \ll 1$ , la estrategia a seguir será competencia. En este caso el coste por cada usuario conectado a otro operador será notablemente más alto que conectarlo a red propia.

Mediante estas dos pruebas, se pretende analizar el efecto que las diferentes políticas de relación que siguen los operadores tienen en las redes creadas y en los parámetros evaluados. Por una parte, se comparará la optimización conjunta con una optimización independiente, y por otra, dentro de las optimizaciones independientes, se estudiarán las consecuencias que tienen las relaciones de coalición y competencia tanto en los aspectos económicos como cualitativos y en el tipo de redes que se creen.

## 8. Cálculos

En este apartado se explicará el problema analizado y la solución elegida para llevar el proyecto a cabo. En esta sección, primero se describirá el problema que se plantea y posteriormente los escenarios creados y la diferencia entre ellos, después se pasará a explicar las pruebas realizadas y los parámetros estudiados.

### 8.2. Descripción del problema

Se supone un área de cobertura de dimensiones  $X_{MAX} \times Y_{MAX}$  en la que  $N$  operadores diferentes ofrecen sus servicios. De manera que el operador  $n$ -ésimo con  $n \in \{1, 2, \dots, N\}$  tiene  $M(n)$  clientes. De manera que, el número total de usuarios en cada área,  $M$ , se define de la siguiente manera:  $M \triangleq \sum_{n=1}^N M(n)$ . Cada uno de estos operadores, cuenta con una estación base con un área de cobertura circular de radio  $R(n)$ . Todos los operadores del área descrita están conectados a una estación de *backhaul* tal y como se muestra en la ilustración 6.

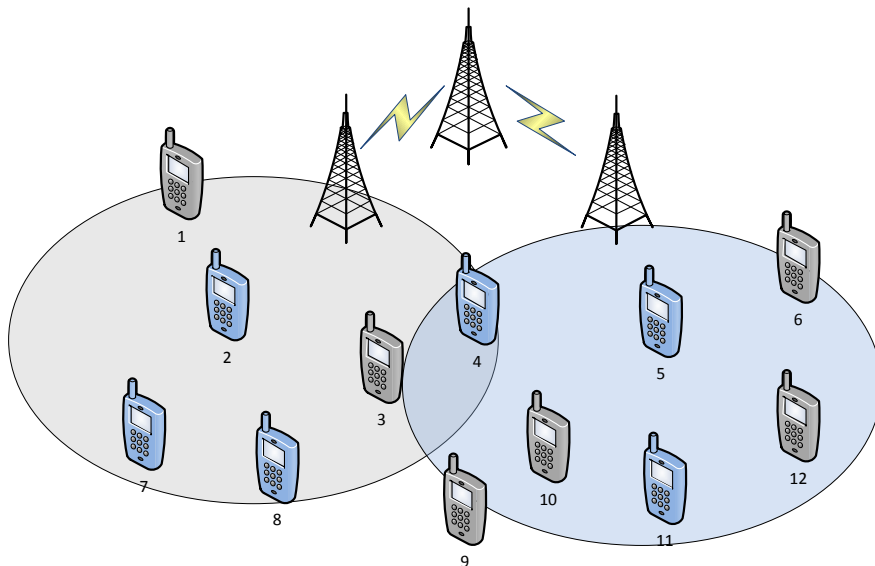


Ilustración 17: Red base para las pruebas

En esta imagen además de los nodos de operador descritos, también se muestran los nodos correspondientes a los usuarios finales. Estos nodos están colocados en las coordenadas  $\{(x_m, y_m)\}_{m=0}^M$  y cada uno de ellos tendrá un contrato con el operador  $O(m) \in \{1, 2, \dots, N\}$ . Los nodos o dispositivos de usuarios, deben estar equipados por lo menos con dos interfaces de red inalámbricas diferentes; una de ellas para la conexión directamente con la estación base del operador (Es decir, usará el protocolo 3G o 4G) y la otra interfaz de red, utilizará protocolos inalámbricos de corto alcance (WiFi, Bluetooth...) y se usará para las conexiones mediante *tethering* y compartición de la conexión con otros usuarios. Todos estos dispositivos además

deben operar en full dúplex, para poder enviar y recibir datos a la BS del operador y mediante *tethering* en el caso de que esté activado.

Los usuarios a los que se les está dando servicio, sean del operador que sean, pueden conectarse a la red de tres maneras diferentes a la red:

1. **Vía el operador con el que tienen el contrato.** En este caso, el dispositivo del usuario se conectará directamente a la BS del operador con el que tiene el contrato. Este tipo de conexión es igual que la que se usa actualmente. Para esta opción directa, el coste de reserva de recursos y mantenimiento de la conexión por cada usuario será  $C_D$
2. **Vía otro operador:** En este segundo caso, el usuario se conectará directamente a la estación base de otro operador con el que no tiene un contrato. Cuando la conexión es a través de otro operador, el coste de la conexión será de  $C_{OO}$  y dependerá del tipo de relación que haya entre el operador con el que tiene el contrato y el operador al que se conecta. Para establecer esta relación, se ha definido el parámetro  $\alpha$  de manera que  $\alpha \triangleq C_D/C_{OO} \in [0,1]$ . En función del valor de  $\alpha$  la relación entre operadores se clasificará de la siguiente manera:
  - $\alpha \cong 1$ : La relación será de coalición, ya que los precios de coste directo y los costes de conexión a otros serán similares  $C_D \cong C_{OO}$ . Se permite la conexión de otros usuarios sin penalizar económicamente al operador.
  - $\alpha \ll 1$ : La relación entre los operadores será de competencia, ya que se penaliza económicamente poder dar servicio a los nodos conectados a otros operadores, en este caso,  $C_D \ll C_{OO}$
3. **Vía *tethering*:** En este último caso, el usuario se conectará a la red a través de otro nodo de usuario mediante la interfaz de corto alcance. En este caso, el coste asociado será  $C_T$ . Este coste es notablemente menor que el coste de conexión directa y será la cantidad que se les abone como incentivo a los nodos que permitan que se conecten a ellos. Estos nodos intermedios pueden ser tanto del mismo operador con el que tiene el contrato como de otro operador.

### 8.3. Descripción de los escenarios

Para solventar el problema descrito en el apartado anterior, se han creado dos escenarios diferentes en los que se ha buscado plasmar las posibles situaciones y casos que se puedan encontrar hoy en día. Los dos escenarios creados, describen dos arquitecturas de red similares en las que se pueden encontrar 3 tipos de nodos diferentes. Por una parte, están todos los nodos que representan usuarios móviles y clientes de los operadores y por otra está el equipamiento de los operadores, formado

por la estaciones base y antenas que dan servicio a los usuarios. Dentro de este segundo grupo de equipamiento del proveedor, se diferencian las estaciones base que funcionan como antenas a las que se conectan los usuarios y las estaciones base principales, que en este caso, funcionarían como *backhaul* de varias BS de operador.

La diferencia principal entre los dos escenarios creados, es el punto de gestión de la red, es decir, el elemento que ejecutará el algoritmo propuesto y descrito en el siguiente apartado. En el primer escenario, hay un elemento principal que tiene conocimiento y control de toda la red de los diferentes operadores conectados y éste será el que ejecute el algoritmo creando una red de topología en árbol de la que será el nodo raíz. En el segundo escenario en cambio, no hay ningún nodo central que pueda gestionar la red de varios operadores; el algoritmo se ejecutará en las propias estaciones base de los operadores y únicamente podrá controlar sus propios nodos. En este caso, el nodo raíz del árbol creado, será la BS del operador.

En ambos escenarios, tanto el nodo principal como las estaciones base de los operadores, están situadas de manera fija dentro del área creada, ya que esos elementos suelen requerir grandes inversiones y suelen ser infraestructuras pesadas que se mantienen fijas. En los modelos creados, el nodo de *backhaul* se sitúa en el centro del espacio definido y las estaciones base de los operadores alrededor de ésta repartidas de manera equidistante. Los elementos correspondientes a los usuarios móviles, se colocan de manera aleatoria por todo el área definida.

Cada uno de estos nodos de usuario, se caracteriza en función de dos parámetros: el operador al que pertenecen y las coordenadas X e Y correspondientes en el área. Tanto el operador asignado a cada usuario como la posición de éstos, se establecerá de manera aleatoria, intentando reproducir una situación realista, ya que no es posible predecir la cantidad de nodos que puede haber en el área de cobertura de una antena o su posición y se ha considerado que los resultados obtenidos de esta manera serán más próximos a la realidad.

En el primer escenario, el nodo principal que tiene conocimiento de toda la red será el que ejecute el algoritmo de optimización, creando como se ha dicho una topología en árbol en la que él mismo será la raíz. Éste será el único elemento que tendrá conocimiento de la red al completo y será por tanto capaz de gestionar todos los nodos y conexiones consiguiendo una administración conjunta de los diferentes operadores conectados.

Para este proyecto, se han configurado los escenarios para dos operadores. En la siguiente imagen, se puede ver la red que se crea al ejecutar el algoritmo. En la ilustración número 7, se puede ver una imagen en la que se muestra una topología creada con una gestión conjunta de la red.



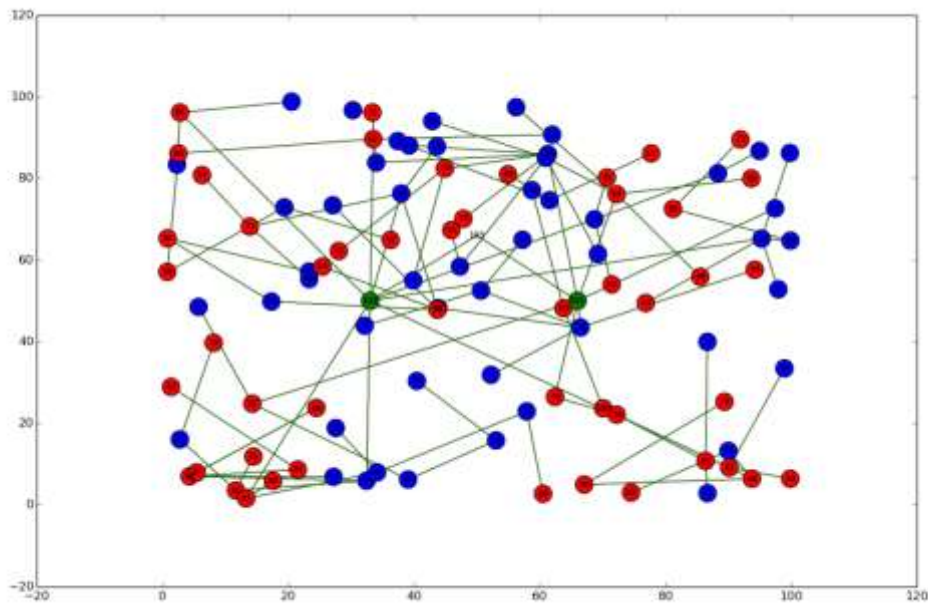


Ilustración 18: Solución gestión conjunta de la red

En la imagen, los nodos en verde corresponden al equipamiento del operador, el nodo central es la estación que ejecuta el algoritmo, y los dos restantes, las antenas. El resto de nodos que se ven en la imagen, son los dispositivos móviles o usuarios de los operadores. Se han usado dos colores diferentes para diferenciar el operador al que pertenece cada uno de ellos.

Con este escenario, se busca simular la situación de un operador móvil virtual que tiene los recursos alquilados a un gran operador y que mantiene una relación de coalición con el operador al que alquila los servicios. que por tanto no dispone de una infraestructura propia que poder gestionar. En este caso, desde una misma estación base se daría cobertura a más de un operador, de manera que el elemento de gestión conjunta se encuentre en el elemento físico, y las estaciones base de los operadores sean lógicas. Ese elemento físico tendría una visión global de la red creada, y sería capaz de gestionarla en su totalidad.

En el segundo escenario, a diferencia del primero, no hay ningún elemento principal que gestione todos los dispositivos. En este caso, se han creado dos operadores que trabajarán de forma totalmente independiente y cada uno dispondrá de su propia red y usuarios. Cada operador, se encargará de crear su topología de red correspondiente y de gestionar a sus propios clientes, sin tener influencia sobre la topología del otro. A pesar de no influenciar en la creación de la red, se le permitirá a nodos de otros operadores conectarse.

En este caso, los elementos del operador están colocados de manera similar al anterior, pero sin haber un elemento principal. Las estaciones base se sitúan en el centro del área definida, y los nodos, al igual que en el caso anterior, están repartidos y se les asigna un operador de forma aleatoria.

Como se ha dicho, en este segundo modelo no hay ningún elemento principal que ejecute el algoritmo. El algoritmo de optimización para la creación de la red, lo ejecutará la estación base de uno de los dos operadores, pudiendo mover sus conexiones mientras el otro operador mantiene su red fija. En la gestión de sus nodos, también se tendrá en cuenta la posibilidad de conexión a la estación base del otro operador o nodos de otro operador.

En este segundo caso, debido a la naturaleza del código *Dandelion*, se ha añadido un nodo más a modo de comodín al que se conectarán todos los nodos marcados como conectados a otro operador. De esta manera, aunque de forma gráfica no sea apreciable, se podrán realizar los cálculos correctamente.

Una vez ejecutado el algoritmo, se puede ver la topología en la siguiente imagen. Como se puede apreciar, es similar al caso anterior, aunque sin una estación principal.

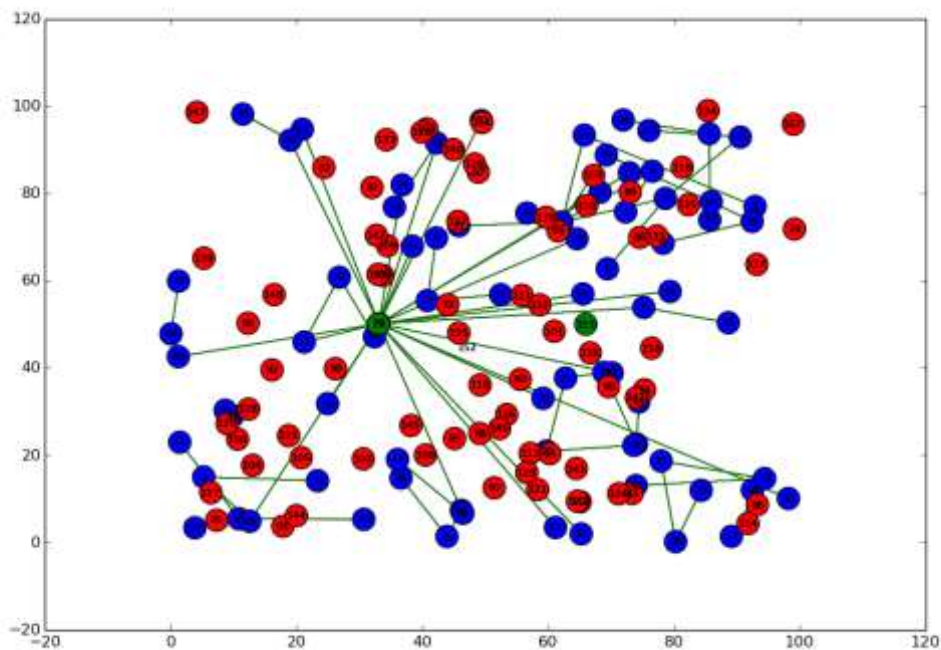


Ilustración 19: Solución gestión independiente de la red

En la imagen anterior, se pueden ver de azul los elementos correspondientes al operador optimizado, y en rojo los elementos de otro operador. Las estaciones base son los elementos en verde. Como vemos, hay una gran cantidad de nodos conectados al elemento nº76, esto se debe a que el nodo comodín está localizado justo debajo de la BS del primer operador.

Este segundo caso, podría corresponder a dos grandes operadores, que disponen de recursos suficientes para mantener su propia infraestructura y gestionar sus redes de forma independiente. Con esta solución, se les permite conectarse a la

red de otro operador para poder maximizar sus beneficios y dar cobertura en sitios en los que ellos no lleguen, pero sin tener una visión de la red del otro operador.

## 8.4. Pruebas realizadas

Tras describir los escenarios creados, se pasará a explicar las pruebas realizadas y los parámetros y datos analizados.

Como ya se ha mencionado en apartados anteriores del informe, el objetivo principal de este proyecto es el análisis del efecto de las políticas de relaciones entre operadores. Para poder determinar el tipo de relación que existe entre dos operadores, se ha definido un parámetro que lo cuantifique en función de los costes asociados a las conexiones entre operadores. El parámetro usado se define de la siguiente manera:

$$\alpha = \frac{\text{Coste conexión usuario propio}}{\text{Coste conexión usuario a otro operador}}$$

Se ha establecido que la relación entre ambos operadores puede ser de coalición o de competencia y se fijará en función al valor de  $\alpha$ :

- Si  $\alpha \approx 1$  : La relación entre los operadores será de coalición
- Si  $\alpha \ll 1$ : La relación entre los operadores será de competencia

Las diferentes políticas de relación que sigan los operadores, tendrán un efecto directo en los resultados tanto económicos como tecnológicos, debido a esto, se realizarán las pruebas para diferentes valores del parámetro  $\alpha$ . Con esto, se pretende simular todas las posibles relaciones entre los operadores, comparándolas con la situación actual: una situación de competencia en la que cada operador gestiona sus propios nodos.

En el caso del primer escenario, hay una gestión conjunta de toda la red, lo que lleva a realizar una única prueba con una relación de coalición. Para este caso, se considera que no es realista hacer una gestión conjunta de la red buscando el máximo beneficio común si la relación entre los operadores es de competencia.

En el segundo caso, por el contrario, la gestión independiente permite establecer diferentes políticas de relaciones entre los operadores, por consiguiente se realizarán pruebas para varios valores de  $\alpha$  y se compararán los resultados obtenidos.

Además, se calcularán los resultados que se obtendrían hoy en día con la situación y arquitectura actual. Una situación de competencia en la que cada operador gestiona sus propios nodos y no permite conexiones a otros operadores.

## 8.5. Parámetros estudiados

Para este proyecto, se estudiarán los dos parámetros más significativos a la hora de evaluar la aptitud de la topología de la red creada. Por una parte, el beneficio del operador, evaluado en unidades monetarias y por otra, la calidad de servicio experimentada por el usuario, medida como el número de saltos hasta la estación base.

Se han seleccionado estos dos parámetros porque se consideran los más relevantes a analizar en una topología de red para los *stakeholder* principales. En el caso de los operadores de telefonía, buscarán maximizar sus beneficios, haciendo este parámetro clave para la posible implantación. En el caso del usuario, buscará una mejor calidad de servicio al menor coste posible. Para la evaluación de la calidad de servicio, se ha decidido tener en cuenta el número de saltos hasta la estación base, ya que puede ser una aproximación al *delay* experimentado.

Además, tal y como se ha estudiado en el apartado 5 de estudio del arte, el retardo experimentado por los usuarios es un factor clave para la evolución a las siguientes generaciones móviles. Entre los diferentes criterios que se han valorado como necesarios para 5G, todos los organismos oficiales están de acuerdo en que el retardo experimentado por el usuario y la velocidad especialmente de bajada son dos de los criterios más relevantes, por tanto, teniendo esto en cuenta, se ha decidido valorar la calidad de los clientes en función del *delay*.

## 8. Descripción de las tareas y Gantt

### 9.1. Descripción de tareas

En la siguiente tabla, se listan las tareas que se han llevado a cabo para la realización del proyecto, junto a cada tarea, aparecen las fechas de inicio y final, la duración y el porcentaje acabado de cada una de ellas. En este caso, como el proyecto está finalizado, todas ellas están acabadas:

WBS	Nombre	Inicio	Fin	Duración	% Completado
<b>1</b>	<b>Programación de HS multiobjetivo</b>	Nov 30	Abr 8	95	100
1.1	Estudio de la teoría	Nov 30	Feb 19	60	100
1.2	Análisis del código inicial	Ene 18	Feb 26	30	100
1.3	Adaptación a multiobjetivo	Feb 29	Abr 8	30	100
<b>2</b>	<b>Adaptación de HS</b>	Abr 11	Abr 22	10	100
2.1	Adaptación a multioperador	Abr 11	Abr22	10	100
<b>3</b>	<b>Definición y creación de los escenarios</b>	Abr 25	Jun 13	46	100
3.1	Definición de los escenarios	Abr 28	Abr 25	1	100
3.2	Creación del primer escenario	Abr 26	May 16	15	100
3.3	Creación del segundo escenario	Abr 26	May 9	10	100
3.4	Adaptación de las funciones	May 17	May 23	5	100
3.5	Incluir movilidad	May 24	Jun 13	15	100
<b>4</b>	<b>Simulación y análisis de los resultados</b>	Jun 14	Ago 4	38	100
4.1	Diseño de las pruebas	Jun 14	Jun 14	1	100
4.2	Realización de las simulaciones	Jun15	Jul 26	30	100
4.3	Análisis de los resultados	Jul 27	Ago 4	7	100
<b>5</b>	<b>Validación de los resultados</b>	Ago 5	Ago 29	17	100
5.1	Análisis profundo de los resultados	Ago 5	Ago 18	10	100
5.2	Validación de los resultados	Ago 19	Ago 4	7	100
<b>6</b>	<b>Revisión y corrección del código</b>	Feb 29	Sep 9	140	100
<b>7</b>	<b>Escritura de Paper</b>	Sep 12	Nov 18	50	100
<b>8</b>	<b>Escritura de la memoria</b>	Sept 12	Ene3	83	100
8.1	Planteamiento	Sep 12	Sep 13	2	100
8.2	Escritura	Sep14	Nov 22	50	100
8.3	Corrección	Nov23	Ene 3	30	100

Tabla 4: Tareas realizadas

Se pasará ahora a una breve descripción de las tareas

#### 1. Programación de HS multiobjetivo

##### 1.1. Estudio de la teoría

En esta primera tarea, se estudiarán los conceptos explicados en el apartado de metodología además del estado del arte de 5G.

Se estudiarán los diferentes algoritmos de optimización y técnicas meta heurísticas, profundizando en los algoritmos evolutivos y especialmente en *Harmony Search* mediante diferentes libros y artículos.

Además de la teoría, se obtendrán conocimientos básicos de Python, ya que es el lenguaje de programación que se usará para el código.

### 1.2. Análisis del código inicial

Para este proyecto, se ha partido de un código anterior [20] en el que también se usaba la codificación/decodificación *Dandelion*. Se ha analizado este código para poder usarlo como base y modificarlo en las siguientes tareas para poder alcanzar los objetivos marcados.

### 1.3. Adaptación a multiobjetivo

En esta tarea, se han creado todas las funciones necesarias para poder realizar una optimización multiobjetivo. Se han programado por una parte las funciones para el cálculo del valor de los parámetros a optimizar y por otra, las funciones necesarias para que la optimización sea multiobjetivo y obtener los frentes de Pareto.

## 2. Adaptación a multioperador

En esta tarea, se adaptará el código actual para poder hacer cálculos con más de un operador. Se parametrizará el número de operadores para la creación de las diferentes situaciones propuestas y se crearán las funciones correspondientes para cálculo de beneficio y calidad para varios operadores. Estas funciones se realizarán para el primer escenario.

## 3. Definición de los escenarios y creación de los escenarios

### 3.1. Definición de los escenarios

En esta tarea se definirán los dos escenarios que se crearán posteriormente. Se fijarán las diferencias entre ambos y se decidirán los parámetros que caracterizarán a cada uno de los nodos desplegados.

### 3.2. Creación del primer escenario

Se creará el escenario para una gestión conjunta de dos operadores. Partiendo del código anterior, se añadirán los diferentes nodos de operador y programarán todas las funciones correspondientes explicadas en la metodología del informe.

### 3.3. Creación del segundo escenario

Se creará el escenario para una gestión independiente de dos operadores. Se modificará el código del escenario anterior para un escenario en el que no se tenga conocimiento de la red en su conjunto y se crearán las funciones necesarias.

### 3.4. Adaptación de las funciones

Se adaptarán las funciones creadas para el primer escenario para los cálculos necesarios para el segundo escenario.

### 3.5. Añadir movilidad

Se añadirá movilidad a los nodos de los usuarios para representar una situación más realista y poder comprobar el efecto de la movilidad en los dos parámetros estudiados.

## 4. Simulación y análisis de los resultados

### 4.1. Diseño de las pruebas

En esta tarea, se han decidido las diferentes pruebas a realizar en los escenarios creados. Se han analizado los factores a tener en cuenta y establecido las pruebas.

### 4.2. Realización de las simulaciones

En esta tarea, se realizarán las diferentes pruebas y simulaciones descritas en los apartados anteriores. Se concretarán los valores de los parámetros, y se guardarán todos los resultados obtenidos para su posterior estudio y valoración.

### 4.3. Análisis de los resultados

Se analizan los resultados obtenidos en las simulaciones realizadas para confirmar su validez.

## 5. Validación de los resultados

### 5.1. Análisis profundo de los resultados

Tras confirmar la validez de los resultados, se procederá a un análisis más profundo de los resultados para poder sacar conclusiones.

### 5.2. Validación de los resultados

Se comprobará que los resultados obtenidos son realistas y se sacarán las conclusiones pertinentes.

## 6. Revisión y corrección del código

Esta tarea se alarga durante toda la elaboración del proyecto, ya que va a ser necesario un continuo análisis y corrección de errores de codificación.

## 7. Paper

Se escribirá un artículo para la conferencia ICHSA 2017 con los resultados obtenidos del proyecto.

## 8. Redacción del informe

En esta última tarea, se redactará el informe que recoge toda la información del proyecto. En él, se plasmará la metodología seguida y los resultados y conclusiones obtenidas tras las pruebas.

## 9.1. Diagrama de GANTT

En la siguiente página del documento, se muestra el diagrama Gantt con las tareas del proyecto.

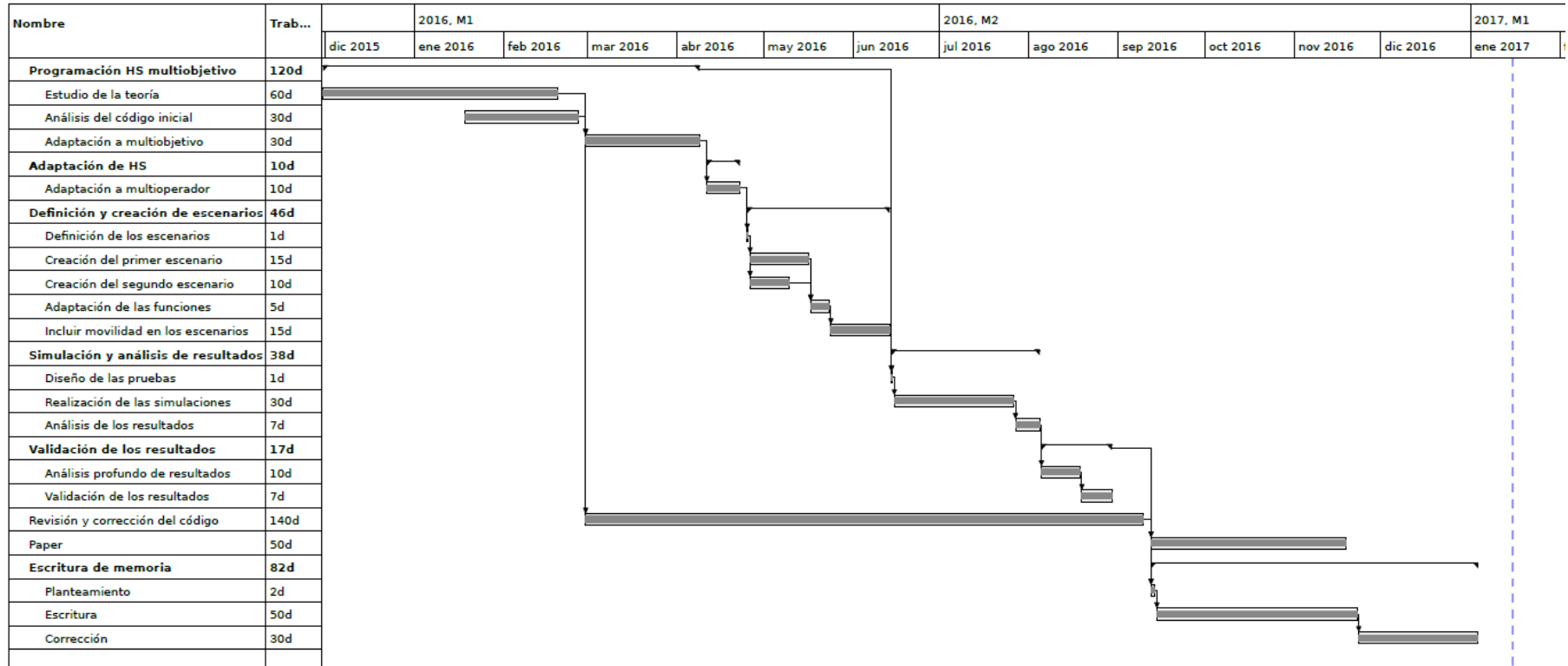


Ilustración 20: Diagrama de Gantt



## 10. Presupuesto

En este apartado, se realizará el presupuesto del proyecto. Para esto, se dividirá el presupuesto total en 3 sub-apartados de lo que se calculará un subtotal. Finalmente, se sumará todo para conseguir el coste total del proyecto.

### 10.1. Mano de obra

En este apartado, se tienen en cuenta las horas de los trabajadores, es decir, el precio de los recursos humanos utilizados para la realización del proyecto. En este caso, se contará con un ingeniero senior y uno junior para la realización del código. Además del propio director del proyecto. Por tanto, en la siguiente tabla se puede ver el coste total de las horas internas:

	€/h	H	Total
Director del proyecto	60	200	12.000
Ingeniero senior	45	300	13.500
Ingeniero junior	30	600	18.000
		SUBTOTAL:	43.500€

Tabla 5: Horas internas

### 10.2. Gastos

En este apartado, tendremos en cuenta los gastos realizados durante el proyecto. En este caso, debido a que no se ha hecho uso de ningún material desechable concreto para el proyecto, el total de los gastos será en material fungible:

	Total
Material fungible	3000
Registro conferencia	600
	SUBTOTAL: 3600 €

Tabla 6: Gastos

### 10.3. Amortizaciones

En cuanto a las amortizaciones, se tendrán en cuenta la amortización del material que se ha utilizado durante el proyecto, es decir, la devaluación de este material, ya que no se ha comprado únicamente para este proyecto:

	Coste inicial	Vida útil (horas)	Uso (horas)	amortizado
Equipo	600	1000	600	360
SUBTOTAL:				360€

Tabla 7: Amortizaciones

### 10.4. Total

En este apartado se suman los subtotales de los apartados anteriores para ver el presupuesto final del proyecto:

CONCEPTO	PRECIO
Horas internas	43.500 €
Gastos	3.600 €
Amortizaciones	360€
<b>TOTAL:</b>	<b>47460 €</b>

Tabla 8: Presupuesto

## 11. Conclusiones

Tras la realización del proyecto y analizando los resultados obtenidos en las pruebas, se ve que la manera más rentable para ambos operadores, es la gestión de la red de manera conjunta. Aunque la diferencia económica que hay con la gestión independiente, tanto en relación de coalición como de competencia, varía en función de la cantidad de usuarios de cada operador. A medida que la cantidad de usuarios aumenta, la diferencia económica también aumenta. El beneficio para el operador crece de manera lineal con el aumento de los usuarios pero en comparación con la ganancia actual, la inclinación de la gráfica es mayor por lo que cuantos más usuarios tenga el operador, la diferencia será mayor.

Por otra parte, cuando la red se optimiza de manera conjunta se busca el mayor beneficio para los dos operadores, de manera que el valor que se optimiza, es la media de ambas ganancias. Por tanto, aunque garantice un beneficio común mayor, puede que este beneficio sea considerable en uno de los dos operadores mientras que en el otro no haya tanto. En función de la topología de red que se cree en cada momento, el beneficio para cada uno de los operadores variará. Este aspecto es algo que se debería tener en cuenta a la hora de establecer una gestión conjunta de la red, ya que en el caso de que la diferencia de ganancia entre ambos operadores sea considerable, puede que a uno de ellos no le resulte rentable, debido que prestará más servicios de los que aprovechará con esta gestión.

En cuanto a la calidad experimentada por los usuarios, aunque se espere que con una gestión conjunta de la red sean mejores, los valores son más deseables para el caso de una gestión independiente, tanto en relación de coalición como de competencia. En la gráfica mostrada con la comparación de las soluciones (Ilustración 24) parece que la diferencia entre los valores de calidad son muy diferentes, pero esto depende de las diferentes topologías. Para los casos en los que el beneficio de la gestión conjunta es muy alto, la calidad se ve deteriorada. Sin embargo, para unos valores de beneficios no tan altos, la calidad no se ve tan afectada.

Además, al analizar los resultados, habría que tener en cuenta que para una gestión conjunta entre los operadores, la red que se ha optimizado tiene 150 nodos de usuario (75 por cada operador). En el caso de la gestión independiente, la red optimizada es de 75 usuarios, los pertenecientes al operador que realiza la optimización. Por tanto, aunque haya nodos conectados a otro operador y se penalice la calidad, es esperable que los valores sean mejores. En cualquier caso, los resultados pueden considerarse más deseables que los actuales en cuanto a conectividad se refiere; el algoritmo está programado para no dejar nodos sin conexión.

Analizando la diferencia entre las dos políticas de gestión independiente, los resultados son bastante parecidos tanto económicamente como cualitativamente. Se puede esperar que una política de coalición maximizaría los beneficios para el operador, pero estudiando los resultados, se ve que no es así. El beneficio del operador, depende del parámetro  $\alpha$  y por tanto de la relación de costes entre los

operadores. En el caso de que el coste de conexión a otro operador sea considerablemente mayor que el de una conexión propia directa, puede traer beneficios negativos para el operador pero si la diferencia de costes no es muy grande (y suponiendo que para una relación de coalición el coste de conexión a otro operados siempre será mayor que el de una conexión propia directa), puede resultar más rentable una relación de competencia aunque esto afecte de manera negativa a la calidad de servicio.

Por último, la movilidad en los nodos hace que el valor de los parámetros baje, sobre todo la calidad de servicio, ya que esta movilidad obliga a realizar diferentes recálculos para poder mantener la conectividad de los nodos y por tanto no poder realizar las iteraciones necesarias hasta conseguir una topología óptima. La movilidad de los nodos ha sido añadida de manera aleatoria y para intentar simular una movilidad real, el movimiento de los nodos entre las diferentes iteraciones no es muy significativo por lo que en varias ocasiones no es necesario.

## 12. Referencias

- [1] Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2012-2017. Cisco, San Francisco, CA, USA. Available at [http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white\\_paper\\_c11-520862.html](http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.html)
- [2] Kiiski, A.: Impacts of MVNOs on Mobile Data Service Market. 17th European regional ITS conference (2006)
- [3] Maitland, C. F., Bauer, J. M., Westerveld, R. The European Market for Mobile Data: Evolving Value Chains and Industry Structures. Telecommunications Policy 26(9), 485-504 (2002)
- [4] Greene, W., Lancaster, B.: Over the Top Services. LTC International Inc. (2007)
- [5] Gruber, H.: Competition and Innovation: The Diffusion of Mobile Telecommunications in Central and Eastern Europe. Information Economics and Policy 13(1), 19-34 (2001)
- [6] Kiiski, A., Hämmäinen, H.: Mobile Virtual Network Operator Strategies: Case Finland. ITS Conference (2004)
- [7] Smura, T., Kiiski, A., Hämmäinen, H.: Virtual Operators in the Mobile Industry: a Techno-Economic Analysis. NETNOMICS: Economic Research and Electronic Networking 8(1-2), 25-48 (2007)
- [8] Bertin, E., Noel, C., Michel, L.: A Few Myths about Telco and OTT Models. International Conference on Intelligence in Next Generation Networks (ICIN), 6-10 (2011)
- [9] Xu, X., Chen, R.: Competition, Cooperation, and Pricing: How Mobile Operators Respond to the Challenge of Over-The-Top. International Journal of Marketing Studies 7(6), 1-13 (2015)
- [10] Kibilda, J., Malandrino, F., DaSilva, L. A.: Incentives for Infrastructure Deployment by Over-The-Top Service Providers in a Mobile Network: A Cooperative Game Theory Model. IEEE International Conference on Communications, 1-6 (2016)
- [11] Wang, X., Chen, M., Taleb, T., Ksentini, A., Leung, V. C.: Cache in the Air: Exploiting Content Caching and Delivery Techniques for 5G Systems. IEEE Communications Magazine 52(2), 131-139 (2014)
- [12] Siris, V. A., Kalyvas, D.: Enhancing Mobile Data Offloading with Mobility Prediction and Prefetching. ACM SIGMOBILE Mobile Computing and Communications Review 17(1), 22-29 (2013)
- [13] Geem, Z. W., Kim, J.-H., Loganathan, G. V.: A New Heuristic Optimization Algorithm: Harmony Search. Simulation 76(2), 60-68 (2001)

- [14] Manjarres, D., Landa-Torres, I., Gil-Lopez, S., Del Ser, J., Bilbao, M. N., Salcedo-Sanz, S., Geem, Z. W.: A Survey on Applications of the Harmony Search Algorithm. *Engineering Applications of Artificial Intelligence* 26(8), 1818-1831 (2013)
- [15] Forsati, R., Haghghat, A. T., Mahdavi, M.: Harmony Search based Algorithms for Bandwidth-Delay-constrained Least-Cost Multicast Routing. *Computer Communications* 31(10), 2505-2519 (2008)
- [16] Del Ser, J., Bilbao, M. N., Gil-Lopez, S., Matinmikko, M., Salcedo-Sanz, S.: Iterative Power and Subcarrier Allocation in Rate-constrained Orthogonal Multicarrier Downlink Systems based on Hybrid Harmony Search Heuristics. *Engineering Applications of Artificial Intelligence* 24(5), 748-756 (2011)
- [17] Del Ser, J., Matinmikko, M., Gil-Lopez, S., Mustonen, M.: Centralized and Distributed Spectrum Channel Assignment in Cognitive Wireless Networks: a Harmony Search Approach. *Applied Soft Computing* 12(2), 921-930 (2012)
- [18] Landa-Torres, I., Gil-Lopez, S., Del Ser, J., Salcedo-Sanz, S., Manjarres, D., Portilla-Figueras, J. A.: Efficient Citywide Planning of Open WiFi Access Networks using Novel Grouping Harmony Search Heuristics. *Engineering Applications of Artificial Intelligence* 26(3), 1124-1130 (2013)
- [19] Thompson, E., Paulden, T., Smith, D. K.: The Dandelion Code: A new Coding of Spanning trees for Genetic Algorithms. *IEEE Transactions on Evolutionary Computation* 11(1), 91-100 (2007)
- [20] Perfecto, C., Bilbao, M. N., Del Ser, J., Ferro, A., Salcedo-Sanz, S.: Dandelion-Encoded Harmony Search Heuristics for Opportunistic Traffic Offloading in Synthetically Modeled Mobile Networks. *Harmony Search Algorithm* 133-145 (2016)
- [21] Landa-Torres, I., Manjarres, D., Gil-Lopez, S., Del Ser, J., Salcedo-Sanz, S.: A Preliminary Approach to Near-optimal Multi-hop Capacitated Network Design using Grouping-Dandelion encoded Heuristics. *IEEE International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, 85-89 (2012)
- [22] Perez-Bellido, A. M., Salcedo-Sanz, S., Ortiz-Garcia, E. G., Portilla-Figueras, A., Naldi, M.: A Dandelion-encoded Evolutionary Algorithm for the Delay-Constrained Capacitated Minimum Spanning Tree Problem. *Computer Communications* 32(1), 154-158 (2009)
- [23] Mahdavi, M., Fesanghary, M., Damangir, E.: An Improved Harmony Search Algorithm for Solving Optimization Problems. *Applied Mathematics and Computation* 188(2), 1567-1579 (2007)
- [24] Deb, K., Pratap, A., Agarwal, S., Meyarivan, T. A. M. T.: A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6(2), 182-197 (2002)
- [25] Picciotto, S.: How to Encode a Tree. Ph.D. Thesis, University of California, San Diego (1999)

**ANEXO 1:  
PLIEGO DE CONDICIONES/  
NORMATIVA APLICABLE**

---



## Introducción

Se trata de un proyecto de investigación en el que no se ha tenido en cuenta ninguna restricción técnica ni legal que pueda afectar a la elaboración del mismo.



**ANEXO 2:  
PLANOS Y ESQUEMAS**

---

# 1. Resultados

En este apartado del informe se presentarán los resultados obtenidos tras las pruebas. Como se ha explicado en el apartado de metodología, se han realizado varias pruebas en dos escenarios diferentes. Por una parte se analizará la diferencia entre una gestión conjunta (el algoritmo se ejecuta desde la estación de *backhaul*) de la red entre varios operadores y una gestión independiente (el algoritmo se ejecuta en la estación base del operador). Dentro de esta gestión independiente, se realizarán simulaciones con relaciones de competencia y coalición. Además, se repetirán las mismas simulaciones pero añadiendo movilidad de usuarios a los escenarios, para poder analizar el efecto que tiene en los dos parámetros. Finalmente, se volverán a repetir las mismas simulaciones pero con diferente número de usuarios, de manera que se pueda ver si la cantidad de usuarios marca una diferencia notable en los resultados.

Para poder realizar una mejor comparación entre las diferentes pruebas, se han establecido los mismos valores para los parámetros de HS en todas las simulaciones. Estos valores se han elegido mediante diferentes pruebas y análisis de los resultados:

- HM (*Harmony memory*): 30
- hms (*harmony memory size*): 30
- hmcr (*harmony memory considering rate*): 0.5
- Par: 0.1
- Número de Iteraciones: 500

En cuanto al resto de parámetros del escenario, se han usado los siguientes valores:

- Dimensiones del área creada: 75 x 75
- Radio de cobertura de operador: 40
- Radio de cobertura de *tethering*: 7
- Número de usuarios por operador: 75
- Coste de conexión directa (mismo operador): 1000
- Coste de conexión mediante *tethering*: 600
- Coste de conexión a otro operador: 1200 (coalición) /10000 (competencia)
- Ingreso de contrato por usuario: 1200

## 1.1. Resultados gestión conjunta

En la siguiente imagen, se muestran los resultados de las diferentes simulaciones realizadas con una gestión conjunta para los dos operadores. En este caso, como se ha dicho anteriormente, se busca optimizar la ganancia conjunta de los dos operadores que participan, y por tanto, la cantidad de nodos total de la red que se optimiza es de 150. Se ven los resultados de las diferentes realizaciones de la simulación. Cada uno de los colores se corresponde con una realización diferente. En azul está marcado el frente de Pareto correspondiente a las diferentes realizaciones:

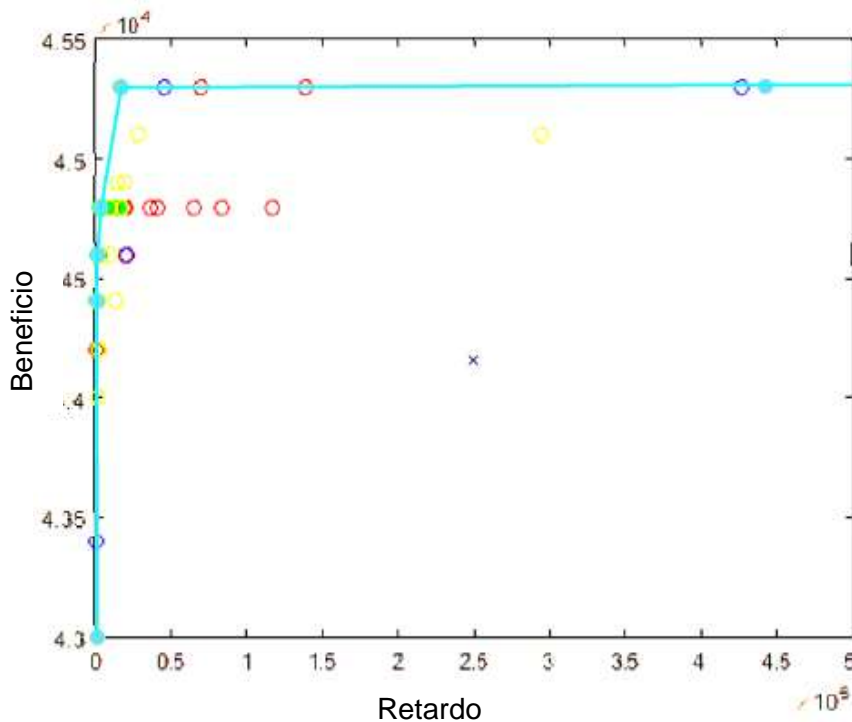


Ilustración 21: Resultados gestión conjunta

## 1.2. Resultados gestión independiente

En este segundo apartado, se van a mostrar los resultados de una gestión independiente de los nodos. Solo se presentan los valores correspondientes al operador que ejecuta el algoritmo para las dos relaciones propuestas.

### 1.2.1. Coalición

La siguiente imagen muestra una gráfica similar a la anterior con los resultados de una relación de coalición entre los operadores. Como se puede apreciar, los valores son menores que los de la optimización conjunta. Los valores numéricos correspondientes, se pueden ver en las tablas 9 y 10.

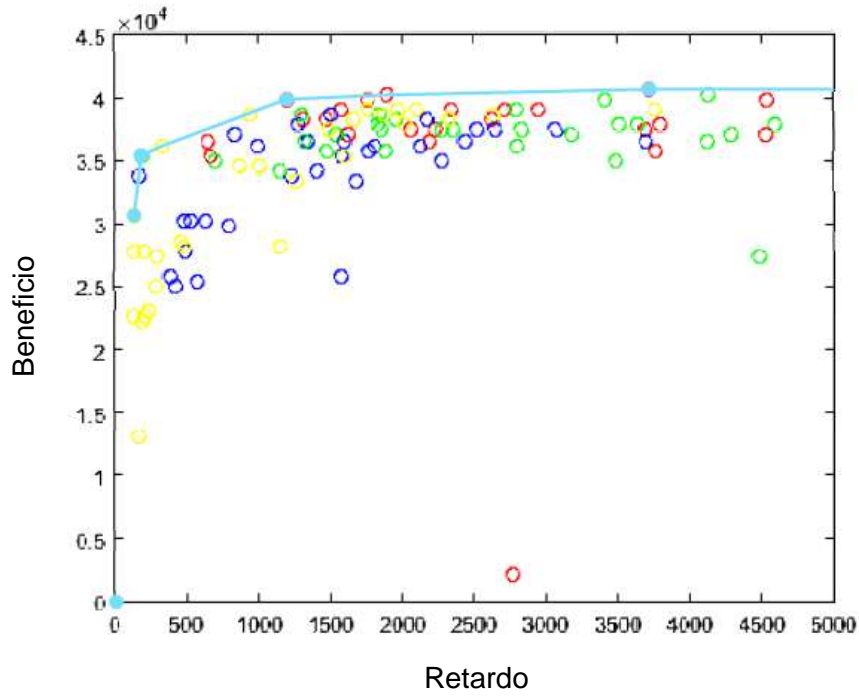


Ilustración 22: Resultados gestión independiente - coalición

### 1.2.2. Competencia

De la misma manera, se muestran los resultados de una relación de competencia:

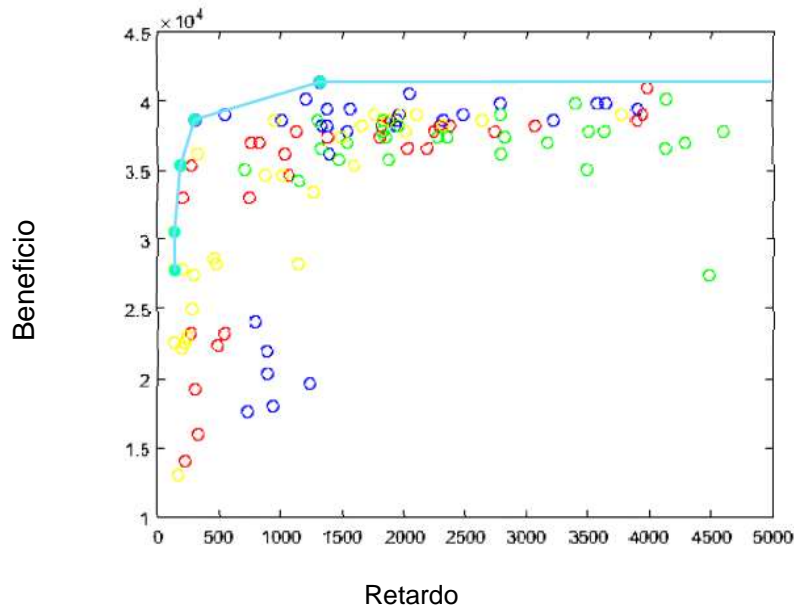
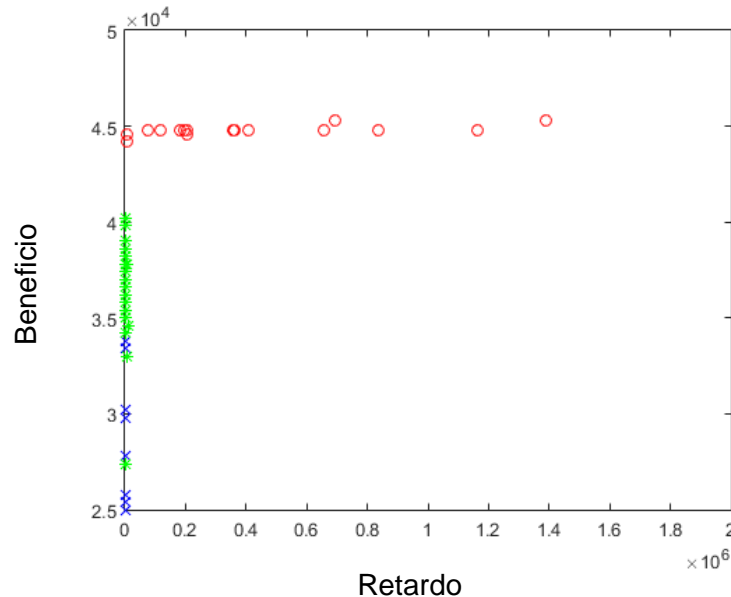


Ilustración 23: Resultados gestión independiente – competencia

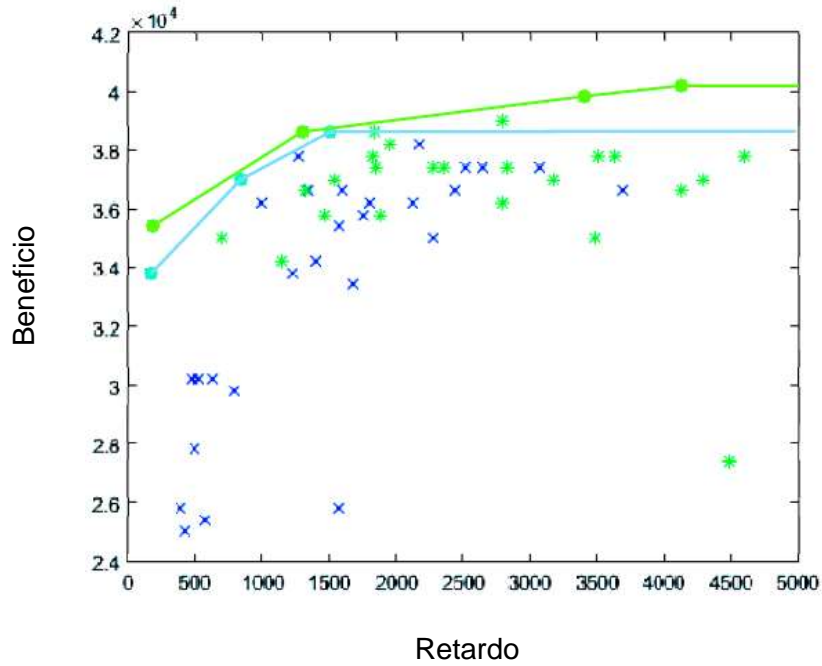
### 1.3. Comparación de los 3 casos

En la siguiente imagen, se puede ver una comparación entre las tres simulaciones que se han realizado. Como se ve en los gráficos anteriores, los valores para la gestión conjunta de la red y de la gestión independiente de la red son de diferente orden.



**Ilustración 24: Comparación resultados conjunta e independiente**

Como se observa, en lo que a valores económicos se refiere, el beneficio es notablemente mayor en el caso de la gestión conjunta para los operadores aunque analizando el retardo, los valores correspondientes a las topologías propuestas son mayores. En la imagen que se muestra, no se puede apreciar una diferencia real entre los valores para los dos casos de gestión independientes de los operadores. Por eso, en la siguiente imagen se ha hecho más zoom en esos valores del gráfico. De esta manera, se puede ver la diferencia entre dos iteraciones de gestión independiente (en verde coalición y en azul competencia).



**Ilustración 25: Comparación resultados gestión independiente**

Además de estas gráficas, en las tablas 9 y 10 se exponen los valores de las diferentes simulaciones para 75 nodos por operador. En la última fila, se ha añadido el beneficio que tendría el operador con las arquitecturas que se usan actualmente con los costes utilizados para las simulaciones.

Beneficio	Max	Min	Media
<i>Backhaul</i>	45300.0	44200.0	44966.6666667
Coalición	40600.0	2200.0	35893.3333333
Competencia	41000.0	14000.0	33786.6666667
Actual	15.000	15000	15000

**Tabla 9: Resultados beneficio 75 nodos**

Retardo	Max	Min	Media
<i>Backhaul</i>	1.15919846504e+12	5663.94771242	89139069202.1
Coalición	236675.29588	651.181818182	41519.1409514
Competencia	5188.01948052	196.519736842	1747.5102634

**Tabla 10: Resultados QoS 75 nodos**

Como se puede apreciar en la primera tabla, la diferencia entre el beneficio obtenido de la forma actual y del algoritmo propuesto es muy amplia. Suponiendo que las ganancias del operador serán aproximadas al valor medio, el beneficio sería más del doble para todos los casos.

En lo que al retardo experimentado se refiere, no se ha añadido una fila con el valor actual debido a que con las políticas actuales se contempla dejar nodos sin cobertura. Por tanto, evaluándolo del mismo modo que se ha utilizado en este proyecto para el algoritmo, el retardo para los nodos sin cobertura sería infinito. Lo que alteraría el valor de los resultados. Aunque el retardo de las topologías de red propuestas siempre será mayor que el de las actuales debido a que la calidad del usuario bajará a medida que los nodos con conexiones en *tethering* suban, es decir, a medida que suba el beneficio del operador.

#### 1.4. Resultado simulaciones movilidad

Para conseguir unos resultados que se adapten más a la realidad, se han realizado las mismas simulaciones añadiendo movilidad a los nodos. En las siguientes tablas se muestran los resultados.

Beneficio	Max	Min	Media
<i>Backhaul</i>	45300.0	42000.0	44757.5
Coalición	38200.0	9700.0	32365.0
Competencia	39800.0	33000.0	37430.0

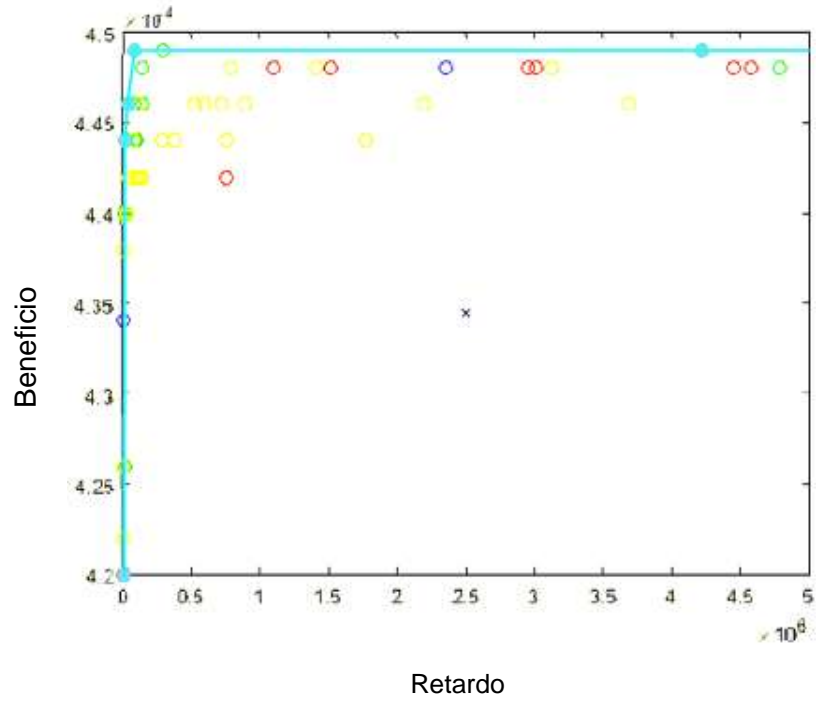
Tabla 11: resultados backhaul movimiento - Beneficio

Retardo	Max	Min	Media
<i>Backhaul</i>	14785243949.2	3965.53921569	409415661.226
Coalición	5332.42207792	887.194555055	2097.19701662
Competencia	5163.64473684	153.875	2291.51502691

Tabla 12: resultados backhaul movimiento – QoS

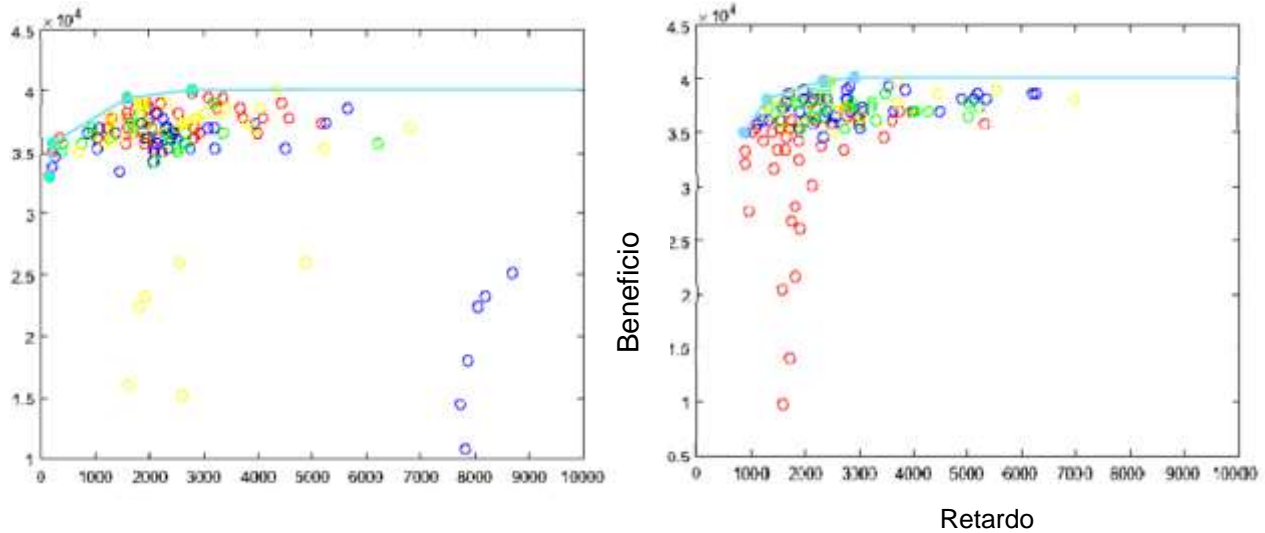
Para este caso, no se ha añadido la fila con el valor actual ya que como los parámetros son los mismos, el valor también será igual. Como se puede observar, los valores económicos son algo menores que en el caso anterior aunque la diferencia no es muy significativa. En cuanto al valor de la calidad de servicio, en general también se ve deteriorada, y es que la movilidad obliga a realizar varios recálculos y cambios de topología que impiden realizar una optimización precisa como en el caso anterior.

Los resultados para una optimización conjunta entre dos operadores se muestran en la siguiente imagen.



**Ilustración 26: Resultados gestión conjunta con movimiento**

En las siguientes imágenes se muestran los resultados de una optimización independiente, coalición y competencia respectivamente.



**Ilustración 27: Resultados gestión independiente movimiento – coalición y competencia**



En la siguiente gráfica se muestra la misma comparación de resultados que en el apartado anterior. Los resultados son similares para todos los casos.

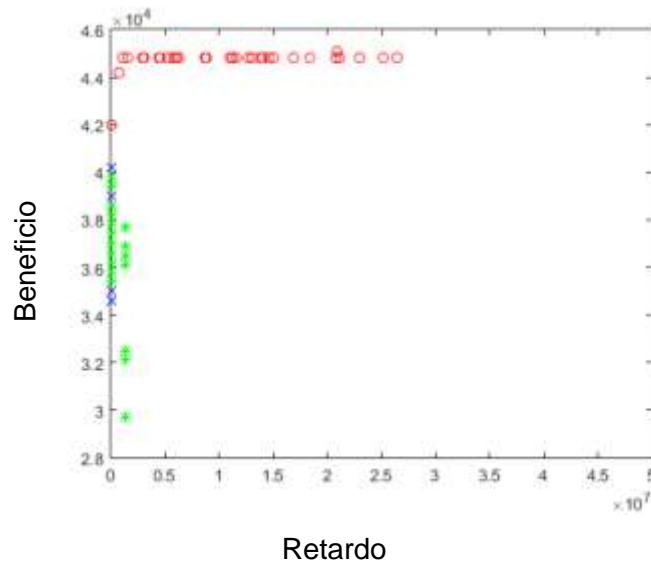


Ilustración 28: Comparación resultados movimiento

### 1.5. Variación en el número de usuarios

Se han realizado las mismas simulaciones para verificar el efecto que tiene en los resultados el número de usuarios de cada uno de los operadores. En las siguientes tablas se muestran los resultados de la calidad de usuario y del beneficio con 20 y 50 nodos. Tras estas tablas, se pueden ver los resultados gráficamente. Para estas tablas, en cuanto a los mínimos se han omitido los valores negativos, aunque se han tenido en cuenta para el cálculo de la media. De la misma manera que para los casos previos, se ha añadido una fila en el caso del beneficio con los valores actuales para poder ver aumento que tienen las topologías propuestas frente al actual.

#### 20 usuarios

Beneficio (€)	Max	Min	Media
<i>Backhaul</i>	12300.0	9000.0	11165.0
Coalición	6800.0	4400.0	5860.0
Competencia	6000.0	4400.0	5430.0
Actual	40000	4000	4000

Tabla 13: Resultados beneficio para 20 usuarios

### 50 usuarios

Beneficio (€)	Max	Min	Media
<i>Backhaul</i>	30300.0	28000.0	30223.3333333
Coalición	29200.0	24400.0	26960.0
Competencia	29600.0	24800.0	28173.3333333
Actual	10000	10000	10000

Tabla 14: Resultados beneficio para 50 usuarios

### 20 usuarios:

QoS	Max	Min	Media
Backhaul	1096467.51163	34.1627906977	28892.1398256
Coalición	81.1904761905	10.4318181818	24.0084145022
Competencia	39.7857142857	10.4318181818	17.1122564935

Tabla 15: Resultados retardo para 20 usuarios

### 50 usuarios:

QoS	Max	Min	Media
Backhaul	26749212033.2	449.199029126	1307503793.77
Coalición	9581.24509804	47.0294117647	1447.74595902
Competencia	5648.79411765	42.8039215686	629.671895425

Tabla 16: Resultados retardo para 50 usuarios

En las siguientes gráficas, se puede ver la evolución de los dos parámetros en función del número de usuarios. Se representan los valores para el Máximo, mínimo y media en rojo, azul y verde respectivamente. El primer gráfico muestra el retardo para las diferentes cantidades de usuarios y el segundo la evolución del beneficio.

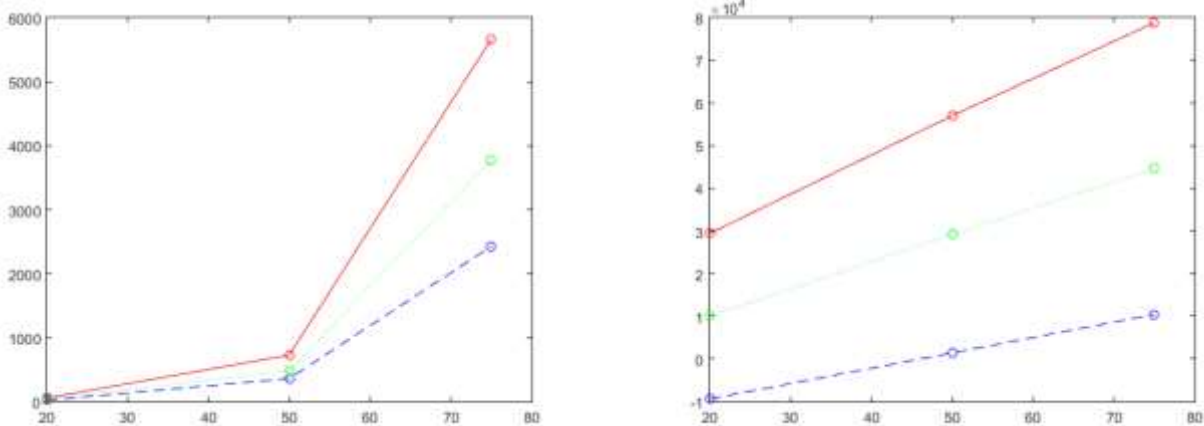


Ilustración 29: : Comparación max/min/media con diferente cantidad de nodos – Retardo y beneficio

## 2. Artículo publicado

Se adjunta el artículo publicado en la conferencia ICHSA 2017 basado en este proyecto.

# An Analysis of Coalition-Competition Pricing Strategies for Multi-operator Mobile Traffic Offloading Using Bi-objective Heuristics

Jone Consul<sup>1</sup>, Cristina Perfecto<sup>1(✉)</sup>, Miren Nekane Bilbao<sup>1</sup>,  
and Javier Del Ser<sup>1,2,3</sup>

<sup>1</sup> University of the Basque Country UPV/EHU, 48013 Bilbao, Spain  
{cristina.perfecto,nekane.bilbao,javier.delser}@ehu.eus

<sup>2</sup> TECNALIA, 48160 Derio, Spain

<sup>3</sup> Basque Center for Applied Mathematics (BCAM), 48009 Bilbao, Spain

**Abstract.** In a competitive market relationships between telecommunications operators serving simultaneously over a certain geographical area are diverse and motivated by very different business strategies and goals. Such relationships ultimately yield distinct pricing portfolios depending on the contractual affiliation of the user being served. Furthermore a key role in the last decade is the concept of tethering (connection sharing) which, when controlled by the operator, may help alleviating the consumption of network resources in densely populated scenarios. In this work we investigate the application of bi-objective heuristics for the design of Pareto-optimal network topologies leading to an optimal Pareto between the revenue of the incumbent operators in the scenario and the quality of service degradation experienced by the end users as a result of tethering. Based on computer simulation this work unveils that such a Pareto-optimal set of topologies is strongly determined by the market relationships between such operators.

**Keywords:** Traffic offloading · Pricing · Multiobjective · Competition · Coalition

## 1 Introduction

In the last years the telecommunication sector has witnessed an upsurge of the number of operators concurrently offering services over the same geographical area [1]. This sharp increase has been specially notable in the provision of mobile services, mainly due to the liberalization of this market in several economies and the decoupling between service deployment and infrastructure management, the latter introducing the role of the so-called virtual operator [2]. As such, virtual operators do not follow the conventional approach of acquiring a spectrum license to deploy a mobile network and offer their services, but rather enter the market by exploiting the infrastructure of incumbent operators to offer Over-The-Top (OTT) services [3]. The coexistence of these market stakeholders with

traditional network operators lay the basis for a strongly competitive mobile market analyzed from different techno-economic perspectives [4–9].

In this context the appearance of multiple virtual players in the telecommunication market has grown at a significantly higher pace than the deployment of new access infrastructure. Content providers and virtual operators have come along with new mobile usage patterns by end users, who have increased their demanded resources from the network as a result of lower prices of terminal devices and the ubiquitous access to high-quality contents. According to Cisco [10] an exemplifying growth of 74% in the global mobile data traffic has been registered in 2015. The need for allocating resources to accommodate the challenging data explosion in mobile networks is what has pushed operators not only to share not only infrastructure in terms of core and radio access networks (as done by the aforementioned virtual operators), but also to explore opportunistic methods that involve their users themselves.

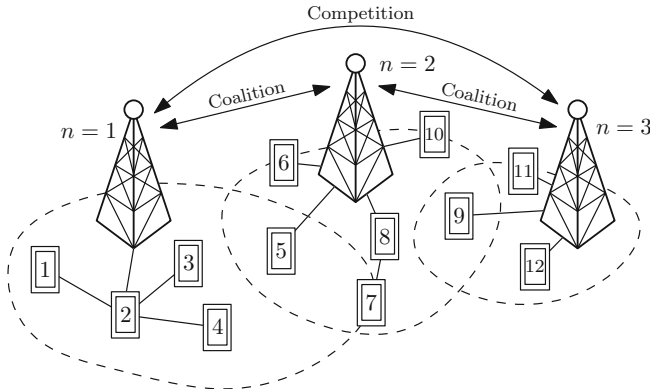
To this end, traffic offloading refers to the family of mechanisms aimed at minimizing the amount of information delivered over licensed communication resources which among other strategies can be accomplished by opportunistically resorting to local wireless communication technologies and dynamic content caching [11]. The criteria adopted in mobile traffic offloading can be very diverse, from the most straightforward policies (i.e. the maximization of the end-user satisfaction or the minimization of network operating expenses) to more elaborated schemes dealing with energy consumption, offload/upload persistence or the social centrality of the user along its predicted mobility trace [12]. Disregarding the criterion adopted to this end, the result is that traffic is shifted to opportunistically set inter-device networks.

In this paper we postulate that opportunistic traffic offloading can also provide interesting benefits when implemented between users of different mobile operators. In particular our work can be framed in the context of user incentives, by which users allow sharing their bandwidth for either offloading traffic of the operator to which they are subscribed, or providing service to subscribers of other operators. In this latter case incentives must be provided at two different levels of the scenario: between different operators and from operators to those subscribers from their client portfolio that should act as opportunistic relays and share their resources. This manuscript will explore how such incentive agreements impact on the Pareto trade-off between the quality of service delivered to end-users and the costs incurred to implement such policies. In particular we will resort to multi-objective heuristics to seek the set Pareto-optimal multi-hop network configurations – i.e. which nodes should be promoted to relays– under different incentive ratios among the operators. We will show that when operators ally together by agreeing low resource sharing fees, their overall benefit increases with respect to a competitive scenario with higher sharing fees. This increased benefit, however, yields a degraded quality of service to the end-user which, in a practical scenario, should be lower bounded in a per application basis. The simulations results obtained from different scenarios and incentive policies will be discussed so as to support these conclusions.

The remainder of this paper is structured as follows: Sect. 2 will formulate the optimization problem that models the offloading of traffic among operators, which will be efficiently solved by means of the solution encoding approach and heuristic solver described in Sect. 3 and subsections therein. The performance of the overall scheme is analyzed and discussed in Sect. 4 based on computer simulations. Section 5 ends the manuscript by drawing several conclusions.

## 2 Problem Formulation

This section presents key concepts and introduces the notation used throughout the rest of the paper. Let us suppose an area of dimensions  $X_{\max} \times Y_{\max}$ , where  $N$  different operators offer their services so that the  $n$ -th operator – where  $n \in \{1, \dots, N\}$  – has  $M(n)$  clients, i.e. each operator serves a total of  $M(n)$  users with  $\sum_{n=1}^N M(n) \triangleq M$  denoting the total number of users existing in the area. Each operator will have a constant and circular coverage area with radius  $R(n)$ , whose limits fall within the previously defined area  $X_{\max} \times Y_{\max}$ . Such operators are connected to a backhaul station as shown in Fig. 1. Nodes correspond to users' devices (e.g. smartphones, tablets, etc.) located at coordinates  $\{(x_m, y_m)\}_{m=0}^M$ , each with its contracted services signed with operator  $O(m) \in \{1, \dots, N\}$ .



**Fig. 1.** Diagram of the scenario with  $N = 3$  operators and  $M = 12$  users. In this hypothesized situation user  $n = 2$  is sharing his/her connection with users  $n \in \{1, 3, 4\}$ , which may (or may not) be clients of the same operator.

User devices are assumed to be equipped with at least two wireless network interfaces such that one of them connects to Internet contents through the operator's cellular base station (BS), whereas the other resorts to short-range wireless protocols (e.g. WiFi or Bluetooth) to share its bandwidth with other users as a tethering interface. All devices operate in full duplex mode so as to be able to send data to the BS and receive data from other nearby nodes if tethering is

enabled. Users under service can be connected to the backhaul through three different ways, each characterized by diverse requirements regarding the perceived quality of service and the cost for both the operator and the user/client:

1. Via the operator contracted by the user: in this case the node will be connected directly to the operator with whom he/she has signed the contract. In this case the quality of service delivered to the user is expected to be contracted and ensured by the operator. For this rationale this direct connection policy yields the highest cost  $C_*$  for the operator among all the cases.
2. Via other third-party operator: in this second connection mode the node at hand will access the backhaul via an external operator with whom he/she has no signed contract. In this case a cost penalty must be taken into account to model the market relation between the operators in the management of each other's clients. A parameter  $\alpha \triangleq C_*/C_{\otimes} \in [0, 1]$  with  $n \neq n'$  is defined such that
  - If  $\alpha \approx 1$  the relationship between operators is a market coalition, i.e. the operational cost associated to the provision of services to any given user is approximately the same disregarding the operator with whom the user signed his/her contract (i.e.  $C_* \approx C_{\otimes}$ ).
  - If  $\alpha \ll 1$  operators will be assumed to compete under a hostile market relationship, namely, it will be significantly more costly for an operator to serve its contracted clients through the premises of the other operator than through its own network equipment.
3. Via tethering: in this last option a node will connect to its BS through the tethered connection open by another user by virtue of the short-range wireless interface of his/her device. This case is expected to maximize the benefit of the operator and the users: operators can serve the user via tethering at a reduced cost  $C_T$  modeling an incentive paid to the user sharing his/her connection, which is assumed to be lower than the cost  $C_*$  incurred when serving the user directly. However, the relayed connection is assumed to degrade the quality of service experienced by the tethered user in the form of processing delays and/or reduced connection bandwidth. The relay node may represent a client from the same operator or instead, belong to any other provider.

The network operator  $n \in \{1, \dots, N\}$  obtains its benefit as the difference between the operational costs of providing service to its users and the monetary income yielded by their contracts  $\{\Omega_m\}_{m=1}^M$ . We simplify this cost as the sum of the costs associated to the price of the resources needed to establish the connection directly to the BS (collected in  $C_*$ ), the price of serving a user through a connection of any other operator (resp.  $C_{\otimes}$ ) and the cost of tethering a connection, in the form of an incentive  $C_T$  to the client owning the tethering device. If we denote such a profit as  $B(n)$ , we have that

$$B(n, \zeta) \triangleq \sum_{m \in M(n)} \Omega_m - C_m^{tot}(\zeta), \quad (1)$$

where the total cost per user  $C_m^{tot}(\zeta)$  is given by

$$C_m^{tot}(\zeta) \triangleq C_* I(\zeta(m) = \text{D}) + C_{\otimes} I(\zeta(m) = \text{O}) + C_T I(\zeta(m) = \text{T}), \quad (2)$$

with  $I(\cdot)$  as an auxiliary indicator function taking value 1 if its argument is true (and 0 otherwise). In the above formulae  $\zeta(m) \in \{\text{D}, \text{O}, \text{T}\}$  denotes the connection mode (DIRECT, OTHER, TETHERED) of user  $m$  as per the description above. This defined budget will encompass the first optimization goal tackled in this study via the heuristic refinement of the vector of connection modes  $\zeta \triangleq \{\zeta(m)\}_{m=1}^M$ .

When the user at hand is tethered through the shared connection of any other user in the network, we will assume that the quality of service degrades in the form of a processing delay. The delay severity will be modeled as a numerical score whose value depends on (1) the number of hops from the user to the BS; (2) the number of simultaneously tethered users at the intermediate nodes that compose the path from the user to the BS. This numerical score  $R(\zeta)$  will be set by the topology of the tree network modeling the connections between the underlying set of users and the BS deployed in the scenario, which in turn is given by the choice of connection modes  $\zeta$  for the compounding nodes of the network. This tree is rooted on the backhaul, with  $N$  first-level nodes representing the BS in the scenario, and intermediate/leaf nodes standing for the  $M$  users.

While other progression models for the delay can be assumed instinctively, in this work we will compute  $R(\zeta)$  as the average of the transmission slots to be waited by every node in the network in the *best* and *worst* cases. Assuming a round robin scheduling policy among users tethered at the same device, the *best* case stands for the case where node  $m$  transmits directly through a multihop path to the BS without awaiting for any other's transmission. The latter (*worst* case) corresponds to the case when the node is scheduled for transmitting during the last transmission slot of every intermediate tethered set of users until the BS at hand. If  $M_l^m(n)$  denotes the number of users at level  $l$  in the subtree rooted on BS  $n \in \{1, \dots, N\}$  such that  $\sum_l \sum_m M_l^m(n) = M(n)$  and all nodes in  $M_l^m(n)$  share the same parent node as  $m$ ; and  $l(m) \in \{1, \dots, L_{\max}^n\}$  (with  $n = O(m)$ ) is the level at which user  $m$  is located in the aforementioned subtree, this score particularized for BS  $n$  will be given as

$$R(n, \zeta) = \frac{1}{M(n)} \sum_{m \in M(n)} \frac{(R_{wc}^m(\zeta) + R_{bc}^m(\zeta))}{2} = \frac{1}{M} \sum_{m=1}^M \frac{l(m) + M_1(n) \prod_{l'=2}^{l(m)} M_{l'}^m(n)}{2},$$

namely, as the average between the number of hops between  $m$  and its BS (best case) and the maximum number of scheduling slots that  $m$  should await for transmission (worst case). Based on this definition, the problem tackled in this paper can be formulated as the search for a  $K$ -sized set of optimal connection modes  $\{\zeta_k^*\}_{k=1}^K$  such that the Pareto trade-off between the operators' revenue and the quality of service experienced by the users is differently balanced, i.e.

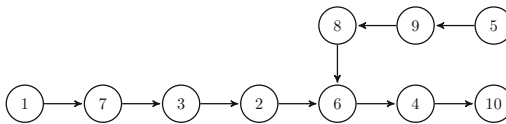
$$\{\zeta_k^*\}_{k=1}^K = \arg_{\zeta \in \{\text{D}, \text{O}, \text{T}\}^M} [\max B(n, \zeta), \min R(n, \zeta)], \quad (3)$$



which can be read as the maximization of the operators' benefit and the minimization of the delay experimented by the users. From an intuitive perspective such optimization objectives are conflicting: in order to reduce operational expenditures, an operator will prefer that their users share as much bandwidth as possible via tethering and a favorable mechanism for incentives. Likewise, direct connections will ensure a high quality of service for the users, but will go against the business goal of the operator at hand. By solving the above optimization problem the decision maker commanding the operator of the network can easily trade one objective for the other as a function of the business priorities and the contractual requirements established at the time. In the next section we will describe the heuristic solver designed to efficiently deal with the above problem.

### 3 Proposed Solver

To solve the problem formulated in the previous section we will resort to a multi-objective version of the Harmony Search (HS) algorithm, a music-based meta-heuristic optimization method first presented in [13] as a result of the observation of the music improvisation procedure undertaken by musicians. When seeking a perfect harmony musicians rely on both their memory of notes played in the past and random pitch variations, which are emulated in a computer program much alike crossover and mutation processes in Evolution are mimicked in Evolutionary Algorithms. Notes played by the musicians represent the values of the optimization variables, which are iteratively refined by applying the aforementioned operators until a stop criterion is met (e.g. a maximum number of iterations or a steady convergence of the fitness values along successive iterations). The good performance scores obtained by this heuristic has been evinced in many application scenarios [14], with several prior contributions in the telecommunication domain [15–18].



**Fig. 2.** Tree structure represented by the Dandelion string  $\mathbf{C} = \{6, 2, 4, 9, 3, 7, 6, 8\}$ .

In order to represent numerically the solutions to the problem in Expression (3) (i.e.  $\zeta$ ) the Dandelion code will be used to represent the tree-like network topology that jointly represents nodes connected to the backhaul under the three connection modes  $\{\mathbf{D}, \mathbf{0}, \mathbf{T}\}$  considered in this work. It is important to observe that by evolving this tree the connection mode is determined depending on which nodes result to be connected to each other. The Dandelion code is a bijective mapping between a tree network topology of  $M + N + 1$  nodes and an integer

string with length  $M + N - 1$ . This code has several properties in terms of inheritance and locality that make it suitable for tackling tree optimization problems by means of Evolutionary Algorithms [19], particularly in telecommunications [20–22]. A brief explanation of the encoding and decoding processes is now given:

- Dandelion encoding: Given a tree on  $n$  nodes  $T \in \Gamma_n$ , usually in the form of an adjacency list or connectivity matrix, with  $\Gamma_n$  denoting the set of possible trees interconnecting  $n$  nodes:
  - Step 1: list intermediate nodes on the path from 1 to  $n$  in  $T$ . Referring to the example tree given in Fig. 2, these are nodes 7, 3, 2, 6 and 4.
  - Step 2: find cycles in the list  $\pi$  by searching for limit elements, namely, elements larger than any other to their right. Such elements for  $\pi$  in the example list are 7, 6 and 4 and thus cycles are (7), (3, 2, 6) and 4.
  - Step 3: the array  $A_C$  for tree  $T$  is constructed by filling its first row with elements 2, 3, 4, ...,  $n - 1$  and adding cycle-related information into its second row, i.e.

$$\mathbf{A}_c = \begin{bmatrix} 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 6 & 2 & 4 & - & 3 & 7 & - & - \end{bmatrix}.$$

- Step 4:  $C_{i+1} = \succ_i$  for every  $i \in [2, n - 1]$  and  $\succ_x$  denoting  $x$ 's successor relationship. Then, the Dandelion code  $C$  corresponding to tree  $T$  is given by the contents of bottom row of  $A_C$ .

$$\mathbf{A}_c = \begin{bmatrix} 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 6 & 2 & 4 & 9 & 3 & 7 & 6 & 8 \end{bmatrix}. \tag{4}$$

By looking at the complete  $A_C$  representation of the example, the Dandelion code of the hypothesized tree  $T$  is  $\mathbf{C} = \{6, 2, 4, 9, 3, 7, 6, 8\}$ .

- Dandelion decoding: this procedure produces a tree  $T \in \Gamma_n$  as follows:
  - Step 1: a  $2 \times n - 2$  matrix  $\mathbf{A}_c$  is built by inserting the integer set  $\{2, 3, \dots, n - 1\}$  in the first row and the elements of  $C$  in the second row. For the exemplifying code  $\mathbf{C} = \{6, 2, 4, 9, 3, 7, 6, 8\}$ ,  $\mathbf{A}_c$  is given as per (4)
  - Step 2: define  $f_C : [2, n - 1] \rightarrow [1, n]$  such that  $f_C(i) = C_i$  for each  $i \in [2, n - 1]$ . Note that  $f_C(i)$  corresponds to the  $i$ -th position  $C_i$  of the code.
  - Step 3: cycles associated to  $f_C$  are computed as  $\{Z_1, Z_2, \dots, Z_L\}$ . In the example 3 cycles, namely (2 6 3), (4) and (7), are obtained. Provided that  $b_l$  denotes the maximum element in  $Z_l$  (with  $l \in \{1, \dots, L\}$ ), cycles are then reordered such that  $b_l$  is set as the rightmost element of  $Z_l$ , and that  $b_l > b_{l'}$  if  $l < l'$ . In words, cycles are circularly shifted so that the largest element is the rightmost and sorted so that cycle maxima decreases from left to right. In the example this step yields  $\{Z_2, Z_1\} = \{(7), (3\ 2\ 6), (4)\}$ .
  - Step 4: a list  $\pi$  of the elements in  $\{Z_1, Z_2, \dots, Z_L\}$  is composed in the order they occur in the cycle list, from the first element of  $Z_1$  to the last entry of  $Z_L$ , i.e.  $\pi = \{(1)(7)(3\ 2\ 6)(4)(10)\}$ .

- Step 5: the tree  $T \in \Gamma_n$  corresponding to  $C$  is constructed by arranging a set of  $n$  isolated nodes labeled with the integers from 1 to  $n$ . A path from node 1 to node  $n$  will be constructed by traversing the list  $\pi$  from left to right. An edge will be included between nodes  $i$  and  $C_i$  for every  $i \in \{2, \dots, n-1\}$  not occurring in  $\pi$ . The tree corresponding to the Dandelion code  $\mathbf{C} = \{6, 2, 4, 9, 3, 7, 6, 8\}$  is the tree given in Fig. 2.

The compounding steps of the proposed bi-objective HS solver are as follows:

1. Initialization: a pool of  $\varphi$  solutions with lengths  $M + N - 1$  is initialized uniformly at random from the alphabet  $\{2, \dots, M + N\}$ , which are evaluated according to the objectives as per (3).
2. Harmony improvisation: a new set of solutions is created from the previous set of harmonies by applying three stochastic operators: Harmony Memory Considering Rate (HMCR), Pitch Adjustment Rate (PAR) and Random Selection Rate (RSR), each driven by probabilistic parameters  $P_{\text{HMCR}}$ ,  $P_{\text{PAR}}$  and  $P_{\text{RSR}}$ , respectively. We embrace the seminal definition of these operators proposed in [13] and extended in [23].
3. Fitness evaluation and memory update: the fitness values of the newly produced solutions is obtained and compared with those of the previous ones. As we deal with a bi-objective optimization problem a non-dominated sorting criterion based on dominance rank and crowding distance (similar to the one embedded in the well-known NSGA-II evolutionary solver [24]) is selected. Only the first  $\varphi$  solutions in the list of harmonies ordered by front rank (first) and crowding distance (second) will be kept for the next iteration.
4. Termination: steps 2 and 3 are repeated until a number of iterations  $\mathcal{I}$  set beforehand are completed.

## 4 Experiments and Results

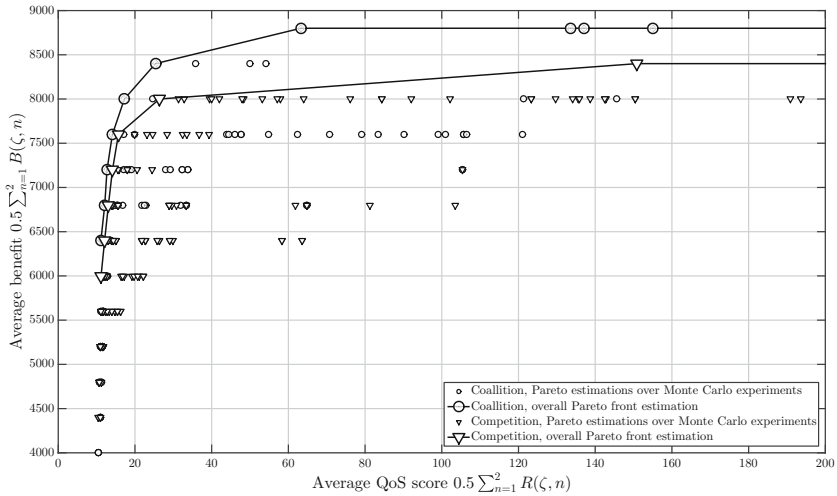
In order to assess the performance of the proposed solver when addressing the bi-objective optimization problem stated in (3), two different scenarios have been created with two operators providing services over the same geographical area. The relation between both operators is defined by  $\alpha$  as explained in Sect. 2. In all cases the area is  $50 \times 50$  with a density of 20 nodes/users per operator. Values of the coverage radii  $\{R(n)\}_{n=1}^2$  have been dynamically adjusted in order to provide service to at least 95% of the deployed users. Parameters of the HS solver are set to  $\varphi = 30$ ,  $P_{\text{HMCR}} = 0.5$ ,  $P_{\text{PAR}} = P_{\text{RSR}} = 0.1$  and  $\mathcal{I} = 500$  iterations, with results averaged over 5 Monte Carlo experiments for each scenario. Incomes from service contracts are assumed to be  $\Omega_m = 1200 \forall m \in \{1, \dots, M\}$ , whereas costs are fixed to  $C_* = 1000$  (direct connection),  $C_T = 600$  (tethering incentive) and  $C_{\otimes} = C_*/\alpha \in \{1000, 10000\}$ . (i.e.  $\alpha \in \{1, 0.1\}$ ). Regarding the latter it is important to note that  $\alpha = 1$  will emulate a coalition market agreement between operators, whereas  $\alpha = 0.1$  will correspond to a competitive market environment.

The discussion focuses on Fig. 3 and the statistics summarized in Table 1. As can be inferred from these obtained results, the income for the operators is

**Table 1.** Statistics of the obtained Pareto front estimations.

	$\min B(n, \zeta)$	$\max B(n, \zeta)$	$\min R(n, \zeta)$	$\max R(n, \zeta)$
Coalition ( $\alpha = 1$ )	4000.00	8800.00	10.43	884.98
Competition ( $\alpha = 0.1$ )	4400.00	8400.00	10.43	724.12

lower for the case when both implement a hostile pricing policy – i.e. a low  $\alpha$  – to provide services to *external* users. One would expect that this reduced profit would come along with a remarkable improvement in terms of quality of service for the end users (i.e. lower values of  $R(n, \zeta)$ ). So do the obtained values for this metric, but differences with respect to the coalition case are only found in its maximum value. Therefore, from these results it can be concluded that for the simulated scenarios, a coalition scenario between service providers is favorable for increasing their average benefit whenever the degradation of the quality of service is admissible under the contractual conditions of their users.



**Fig. 3.** Pareto Fronts trading average benefit for the quality of service offered to users under coalition and competition between operators.

### 5 Concluding Remarks

This manuscript has elaborated on analyzing the impact of different pricing agreements between service operators on the Pareto trade-off between their net income and the quality of service offered to their users. We have formulated this scenario as a bi-objective optimization problem, which relies on modeling the connection from users to the operators’ network equipment as a tree graph

that, in addition, accommodates the possibility of the operator to tether connections under incentive mechanisms. This graph is evolved towards Pareto-optimal configurations differently balancing quantitative metrics of the aforementioned optimization goals. The evolution is implemented efficiently by a heuristic solver that iteratively refines candidate solutions represented by means of the so-called Dandelion code, which possesses interesting features for evolving tree networks via crossover and mutation processes. Preliminary simulation results have evinced how such objectives behave when operators agree on a soft pricing policy for user roaming or, alternatively, impose hostile costs when processing users from any other counterpart.

**Acknowledgments.** This work is supported by the Basque Government through the ELKARTEK program, in particular the BID3A (ref. KK-2015/0000080) and BID3ABI projects.

## References

1. Maitland, C.F., Bauer, J.M., Westerveld, R.: The European market for mobile data: evolving value chains and industry structures. *Telecommun. Policy* **26**(9), 485–504 (2002)
2. Kiiski, A.: Impacts of MVNOs on mobile data service market. In: 17th European Regional ITS Conference (2006)
3. Greene, W., Lancaster, B.: Over the Top Services. LTC International Inc. (2007)
4. Gruber, H.: Competition and innovation: the diffusion of mobile telecommunications in central and Eastern Europe. *Inf. Econ. Policy* **13**(1), 19–34 (2001)
5. Kiiski, A., Hämmäinen, H.: Mobile virtual network operator strategies: case Finland. In: ITS Conference (2004)
6. Smura, T., Kiiski, A., Hämmäinen, H.: Virtual operators in the mobile industry: a techno-economic analysis. *NETNOMICS Econ. Res. Electron. Netw.* **8**(1–2), 25–48 (2007)
7. Bertin, E., Noel, C., Michel, L.: A few myths about telco and OTT models. In: International Conference on Intelligence in Next Generation Networks (ICIN), pp. 6–10 (2011)
8. Xu, X., Chen, R.: Competition, cooperation, and pricing: how mobile operators respond to the challenge of over-the-top. *Int. J. Market. Stud.* **7**(6), 1–13 (2015)
9. Kibilda, J., Malandrino, F., DaSilva, L.A.: Incentives for infrastructure deployment by over-the-top service providers in a mobile network: a cooperative game theory model. In: IEEE International Conference on Communications, pp. 1–6 (2016)
10. Cisco visual networking index: global mobile data traffic forecast update, 20122017. Cisco, San Francisco, CA, USA. [http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white\\_paper\\_c11-520862.html](http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.html). Accessed Nov 2016
11. Wang, X., Chen, M., Taleb, T., Ksentini, A., Leung, V.C.: Cache in the air: exploiting content caching and delivery techniques for 5G systems. *IEEE Commun. Magaz.* **52**(2), 131–139 (2014)
12. Siris, V.A., Kalyvas, D.: Enhancing mobile data offloading with mobility prediction and prefetching. *ACM SIGMOBILE Mobile Comput. Commun. Rev.* **17**(1), 22–29 (2013)

13. Geem, Z.W., Kim, J.-H., Loganathan, G.V.: A new heuristic optimization algorithm: harmony search. *Simulation* **76**(2), 60–68 (2001)
14. Manjarres, D., Landa-Torres, I., Gil-Lopez, S., Del Ser, J., Bilbao, M.N., Salcedo-Sanz, S., Geem, Z.W.: A survey on applications of the harmony search algorithm. *Eng. Appl. Artif. Intell.* **26**(8), 1818–1831 (2013)
15. Forsati, R., Haghghat, A.T., Mahdavi, M.: Harmony search based algorithms for bandwidth-delay-constrained least-cost multicast routing. *Comput. Commun.* **31**(10), 2505–2519 (2008)
16. Del Ser, J., Bilbao, M.N., Gil-Lopez, S., Matinmikko, M., Salcedo-Sanz, S.: Iterative power and subcarrier allocation in rate-constrained orthogonal multicarrier downlink systems based on hybrid harmony search heuristics. *Eng. Appl. Artif. Intell.* **24**(5), 748–756 (2011)
17. Del Ser, J., Matinmikko, M., Gil-Lopez, S., Mustonen, M.: Centralized and distributed spectrum channel assignment in cognitive wireless networks: a harmony search approach. *Appl. Soft Comput.* **12**(2), 921–930 (2012)
18. Landa-Torres, I., Gil-Lopez, S., Del Ser, J., Salcedo-Sanz, S., Manjarres, D., Portilla-Figueras, J.A.: Efficient citywide planning of open wifi access networks using novel grouping harmony search heuristics. *Eng. Appl. Artif. Intell.* **26**(3), 1124–1130 (2013)
19. Thompson, E., Paulden, T., Smith, D.K.: The dandelion code: a new coding of spanning trees for genetic algorithms. *IEEE Trans. Evol. Comput.* **11**(1), 91–100 (2007)
20. Perfecto, C., Bilbao, M.N., Del Ser, J., Ferro, A., Salcedo-Sanz, S.: Dandelion-encoded harmony search heuristics for opportunistic traffic offloading in synthetically modeled mobile networks. In: Kim, J.H., Geem, Z.W. (eds.) *Harmony Search Algorithm. Advances in Intelligent Systems and Computing*, vol. 382. Springer, Heidelberg (2016)
21. Landa-Torres, I., Manjarres, D., Gil-Lopez, S., Del Ser, J., Salcedo-Sanz, S.: A preliminary approach to near-optimal multi-hop capacitated network design using grouping-dandelion encoded heuristics. In: *IEEE International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, pp. 85–89 (2012)
22. Perez-Bellido, A.M., Salcedo-Sanz, S., Ortiz-Garcia, E.G., Portilla-Figueras, A., Naldi, M.: A dandelion-encoded evolutionary algorithm for the delay-constrained capacitated minimum spanning tree problem. *Comput. Commun.* **32**(1), 154–158 (2009)
23. Mahdavi, M., Fesanghary, M., Damangir, E.: An improved harmony search algorithm for solving optimization problems. *Appl. Math. Comput.* **188**(2), 1567–1579 (2007)
24. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.A.M.T.: A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **6**(2), 182–197 (2002)