

IMPLEMENTACIÓN Y VIABILIDAD DE TÉCNICAS JRRM PARA SISTEMAS MÓVILES HETEROGÉNEOS

M^a Carmen Lucas Estañ y Javier Gozávez

m.lucas@umh.es, j.gozalvez@umh.es

Ubiquitous Wireless Communications Research Laboratory (Uwicare), <http://www.uwicare.umh.es/>

Universidad Miguel Hernández, Avda. de la Universidad s/n, 03202, Elche, España

Abstract- The study and design of Joint Radio Resource Management (JRRM) techniques is a key and challenging aspect in future Beyond 3G wireless communication systems, where heterogeneous Radio Access Technologies will physically coexist. In these systems, the total available radio resources need to be used in a coordinated way to guarantee adequate satisfaction levels to all users, and maximize the system revenues. In addition, JRRM algorithms need to exhibit good computational performance to be applied in real hardware systems. In this context, this work analyses the real-time computational performance of a novel JRRM proposal based on linear programming optimization using DSPs commonly used in mobile base stations in order to demonstrate its potential to be employed in Beyond 3G heterogeneous wireless systems.

I. INTRODUCCIÓN

Los futuros sistemas de comunicaciones móviles e inalámbricas estarán compuestos por distintas Tecnologías de Acceso Radio (RATs, *Radio Access Technologies*) coexistiendo físicamente y ofreciendo servicios móviles de manera coordinada a un amplio rango de usuarios con altos requisitos de QoS. Un aspecto clave en los futuros sistemas de comunicaciones móviles es la gestión coordinada de los recursos radio heterogéneos, normalmente denominada *Joint Radio Resource Management* (JRRM). Las políticas JRRM deben decidir para cada usuario, la RAT y el número de recursos radio dentro de dicha RAT que deben ser asignados al usuario para satisfacer sus requisitos de QoS obteniendo el máximo beneficio global del sistema. La mayoría de los estudios realizados sobre JRRM sólo abordan el dilema de selección de RAT (por ejemplo [1] y [2]). Una propuesta inicial que aborda simultáneamente los dilemas de selección de RAT y de asignación de recursos es realizada en [3]. El algoritmo propuesto está basado en redes neuronales y lógica difusa, aunque sólo determina la tasa de transmisión necesaria en cada RAT y no el número de recursos radio. En este contexto, [4] propone un algoritmo basado en optimización y programación lineal que simultáneamente asigna a cada usuario una combinación adecuada de RAT y número de recursos radio que garantiza los requisitos de QoS del usuario. En [4] se demostró el potencial de esta técnica para ser aplicado en sistemas móviles heterogéneos.

Dado que las decisiones JRRM están basadas en cantidades cada vez mayores de información y en algoritmos más complejos, el tiempo de procesado se ha convertido en un factor muy importante que puede determinar la viabilidad de la implementación de las técnicas en sistemas hardware

reales. Además, la mayoría de estudios en el campo de las comunicaciones móviles están basados en simulación por computador, y pocos de ellos evalúan el coste computacional de sus propuestas en sistemas reales. En este contexto, este trabajo investiga por primera vez, según el conocimiento del autor, el rendimiento en tiempo real de una política JRRM con el objetivo de analizar la viabilidad de su implementación en sistemas hardware reales.

II. TÉCNICA JRRM

La técnica JRRM analizada en este trabajo y propuesta en [4] explota la flexibilidad en términos de QoS/recursos que ofrecen los diferentes servicios en un entorno multimedia para garantizar adecuados niveles de satisfacción a los usuarios mientras se alcanza el máximo beneficio del sistema; este trabajo considera un sistema heterogéneo compuesto por GPRS, EDGE y HSDPA con usuarios de correo electrónico, web y vídeo en tiempo real. Para conseguir la solución óptima al problema, la definición de la técnica JRRM está basada en optimización y mecanismos de programación lineal, los cuales obtienen la solución óptima a problemas modelados por medio de ecuaciones lineales (Sección III).

A. Técnica MAXILOU

La técnica JRRM diseñada lleva a cabo la distribución de los recursos radio según una política de igualdad entre usuarios; la técnica intenta proporcionar niveles de satisfacción homogéneos, y los más altos posibles, a todos los usuarios en el sistema y sólo cuando no sea posible debido a la escasez de recursos radio, se aplicarán prioridades entre servicios. En este trabajo, la satisfacción del usuario es expresada por medio de valores de utilidad que identifican la asignación de recursos radio necesaria por los usuarios para alcanzar determinados niveles de QoS en función del tipo de servicio. Estos niveles de utilidad han sido establecidos a partir de las funciones de utilidad mostradas en la Fig. 1, definidas en función del *throughput* para los servicios de web y email, y del porcentaje de tramas correctamente recibidas para el servicio de vídeo en tiempo real. Estas funciones de utilidad han sido definidas en base a los niveles de QoS mostrados en la figura. Es importante resaltar que los usuarios percibirán un valor de utilidad nulo mientras no reciban recursos que satisfagan su nivel de QoS mínimo. Una vez estas funciones de utilidad han sido definidas, es necesario relacionar la utilidad con las posibles asignaciones

de recursos radio que un usuario puede recibir. Para ello, a cada combinación RAT/recursos radio le ha sido asociada una tasa de transmisión. Dado que todas las RATs implementan técnicas de adaptación al medio por las que la tasa de transmisión se adapta según la calidad del enlace, han sido consideradas tasas de transmisión correspondientes a modos de transmisión en cada RAT que proporcionen un balance entre tasa de transmisión y robustez ante errores. En particular, se han seleccionado los modos de transmisión CS2 y MCS5 que proporcionan 13.4 y 22.4 kbps por ráfaga temporal o *timeslot* en GPRS y EDGE respectivamente. En HSDPA, al existir distintos modos de transporte para cada número de códigos posible, ha sido escogido de nuevo el modo de transmisión que logra un balance entre prestaciones y fiabilidad. La Tabla 1 muestra un ejemplo de los valores de utilidad definidos para el servicio de vídeo en tiempo real a 64 kbps. En la tabla, las distintas asignaciones de recursos han sido denotadas como xY , representando x recursos radio (*timeslots* o códigos) de la RAT Y , representando GPRS, EDGE y HSDPA como G, E y H, respectivamente.

Una vez establecidos los valores de utilidad, el problema JRRM puede ser modelado. Para alcanzar su objetivo final, la técnica JRRM busca maximizar el menor valor de utilidad asignado a un usuario en cada reparto de los recursos radio. El objetivo de la técnica denominada MAXILOU (*MAXImise LOWest Utility*) es expresado de la siguiente manera:

$$\max \{ \min u_j \}, j \in [0, N-1] \quad (1)$$

donde u_j representa el valor de utilidad asignado al usuario j en el reparto de recursos actual y N corresponde a la carga de usuarios en el sistema. Si todos los usuarios presentan los mismos requisitos de servicio, (1) se obtiene asignando los mismos valores de utilidad a todos los usuarios. Para poder aplicar mecanismos de programación lineal, (1) debe expresarse de forma lineal. Para ello, se ha definido una nueva variable real z igual al menor valor de utilidad asignado a un usuario, expresando (1) como:

$$\max z, j \in [0, N-1] \text{ con } z \leq u_j \quad (2)$$

Además, u_j puede expresarse como:

$$u_j = \sum_r \sum_s u_j^r(s^r) \cdot y_j^{r,s} \quad (3)$$

siendo $u_j^r(s^r)$ el valor de utilidad obtenido por el usuario j habiendo recibido s recursos radio (códigos o *timeslots*) de la RAT r , y $s \in [1, c_r]$ con c_r igual al máximo número de recursos radio disponibles en cada RAT. $y_j^{r,s}$ es una variable binaria igual a 1 si el usuario j recibe s recursos de la RAT r y 0 en caso contrario. Por tanto, MAXILOU debe decidir que variable $y_j^{r,s}$ debe ser igual a cero para cada usuario considerando solamente variables $y_j^{r,s}$ que proporcionen al usuario valores de utilidad mayores que cero.

El planteamiento del problema debe completarse con la expresión de las restricciones impuestas por el sistema y los distintos servicios. La primera de ellas es que cada usuario sólo puede tener una variable $y_j^{r,s}$ igual a 1, no pudiendo sobrepasar el número total de recursos asignados el total de recursos disponibles en el sistema:

$$\sum_r \sum_s y_j^{r,s} = 1, \forall j \quad \text{y} \quad \sum_j \sum_r s^r y_j^{r,s} \leq c_r, \forall r \quad (4)$$

La distribución de los recursos se realiza cada vez que un usuario pide acceso al sistema o finaliza su transmisión. En

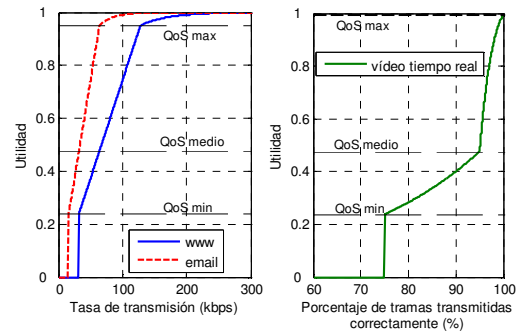


Fig. 1. Funciones de utilidad por tipo de servicio.

Rec./ RAT	Tasa datos (kbps)	Utilidad	Rec./ RAT	Tasa datos (kbps)	Utilidad	Rec./ RAT	Tasa datos (kbps)	Utilidad
1G	13.4	0.000	4E	89.6	0.298	3H	741	1.000
1E	22.4	0.000	7G	93.8	0.313	4H	1139.5	1.000
2G	26.8	0.000	8G	107.2	0.353	5H	2332	1.000
3G	40.2	0.000	5E	112	0.365	7H	4859.5	1.000
2E	44.8	0.000	1H	116.5	0.378	8H	5709	1.000
4G	53.6	0.000	6E	134.4	0.435	10H	7205.5	1.000
5G	67	0.000	7E	156.8	0.934	12H	8618.5	1.000
3E	67.2	0.000	8E	179.2	0.982	15H	11685	1.000
6G	80.4	0.000	2H	396	1.000			

Tabla 1. Valores de utilidad para el servicio de vídeo a 64kbps.

cada nuevo reparto, solamente los usuarios activos de tiempo real mantendrán el número mínimo de los recursos recibidos en el reparto anterior que garanticen su nivel de QoS mínimo si la asignación en el reparto anterior lo satisfacía o la asignación de recursos anterior en caso contrario ($s_{\min}^{r_{\min}}$), compitiendo con el resto de usuario por recursos adicionales:

$$\sum_r \sum_{s^r} u_j^r(s^r) \cdot y_j^{r,s} \geq u_j^{r_{\min}}(s_{\min}^{r_{\min}}), \forall j_{h263} \quad (5)$$

Cuando no sea posible asignar niveles de satisfacción homogéneos a todos los usuarios debido a la escasez de recursos, MAXILOU aplicará las siguientes prioridades entre servicios: vídeo en tiempo real (mayor prioridad), web y email; usuarios de vídeo con mayores tasas de transmisión serán servidos primero. Si un usuario m es un usuario de vídeo que obtiene recursos radio en el reparto anterior, la condición (5) prevalece y el usuario de vídeo recibirá los $s_{\min}^{r_{\min}}$ recursos radio de la RAT r_{\min} necesarios para satisfacer su nivel de QoS mínimo garantizado. El usuario m no recibirá recursos adicionales hasta que el usuario de mayor prioridad (usuario k) sobrepase su nivel de utilidad ($u_m^{r_{\min}}(s_{\min}^{r_{\min}})$). Esta restricción es expresada de la siguiente manera:

$$\sum_{r_a} \sum_{s_a} u_m^{r_{\min}}(s_{\min}^{r_{\min}}) \cdot y_k^{r_a, s_a} + \sum_{r_b} \sum_{s_b} u_k^r(s^r) \cdot y_k^{r, s} \geq \sum_r \sum_s u_m^r(s^r) \cdot y_k^{r, s} \quad (6)$$

donde (r_a, s_a) representa la asignación RAT/recursos radio que verifica $u_m^{r_{\min}}(s_{\min}^{r_{\min}}) > u_k^{r_a}(s_a)$ y (r_b, s_b) la asignación RAT/recursos radio que verifica $u_m^{r_{\min}}(s_{\min}^{r_{\min}}) \leq u_k^{r_b}(s_b)$, siendo la prioridad del usuario k mayor que la del usuario m . Si según todas las restricciones impuestas, el problema no tiene solución, los usuarios de menor prioridad serán eliminados del reparto hasta poder alcanzar una solución.

B. Rendimiento de MAXILOU

MAXILOU ha sido evaluada en una plataforma de simulación que emula las RATs GPRS, EDGE y HSDPA. Cargas de 10 y 20 usuarios distribuidos de forma homogénea entre los servicios de web, email y vídeo en tiempo real han sido simuladas. Según las tasas de transmisión de los

servicios de vídeo, se han considerado dos escenarios: E1 con servicios de vídeo a 16, 64 y 128 kbps, y E2 con servicios de vídeo a 64, 256 y 512 kbps.

La Fig. 2 muestra el porcentaje de usuarios que alcanzan los niveles de QoS mínimo, medio y máximo mostrados en la Fig. 1. Estos resultados muestran que MAXILOU consigue sus objetivos. En primer lugar, los usuarios reciben los niveles de satisfacción más altos y homogéneos posibles en las situaciones de escasez de recursos simuladas que no permiten alcanzar máximos niveles de satisfacción para todos los usuarios. Por otro lado, los efectos de las prioridades entre servicios son más notables en los escenarios con mayor escasez de recursos, es decir, cuando la demanda de recursos o la carga de usuarios aumentan. En estos escenarios, parte de los usuarios de menor prioridad no ven satisfecho su nivel de QoS mínimo mientras que la mayoría de los usuarios de mayor prioridad alcanzan su nivel de QoS máximo. Esto es debido a que cualquier otra distribución de los recursos radio que consiga niveles de utilidad más homogéneos incumpliría la condición de prioridades enunciada en (6). Por tanto, MAXILOU sirve el máximo número de usuarios que satisfacen las restricciones impuestas por el sistema asignándoles los niveles de QoS más homogéneos posible.

III. HERRAMIENTAS DE PROGRAMACIÓN LINEAL

MAXILOU modela el problema JRRM en sistemas heterogéneos mediante una serie de ecuaciones lineales con variables binarias y reales, el cual puede ser resuelto mediante técnicas de programación entera mixta (MIP, *Mixed Integer Linear Programming*). La mayoría de los mecanismos de resolución de problemas MIP están basados en el método simplex [5], método ampliamente utilizado para resolver problemas de programación lineal con un elevado número de variables y que requieren soluciones computacionalmente eficientes. El método simplex es un procedimiento algebraico que aprovecha el hecho de que el espacio de posibles soluciones está acotado por las restricciones del problema encontrándose la solución óptima en uno de los vértices de esta región. Dado que este método es sólo aplicable a problemas con variables reales, otros mecanismos para la resolución de problemas con variables enteras han sido desarrollados.

Uno de los métodos más utilizados para resolver problemas MIP por su rendimiento y buenas propiedades computacionales es el método *Branch and Bound* (B&B) [5]. Este método resuelve una secuencia ordenada de subproblemas de programación lineal derivados del problema

original hasta alcanzar la solución óptima. Estos subproblemas se obtienen dividiendo el conjunto de posibles valores para cada variable en conjuntos más reducidos. Para resolver cada uno de estos sub-problemas se aplica el método simplex, ignorando la condición de variables enteras y permitiendo que todas las variables alcancen valores reales. Sin embargo, este trabajo ha utilizado el método *Branch and Cout* [5] para resolver el problema JRRM. Este método mejora el método B&B incorporando planos de corte, los cuales son nuevas restricciones funcionales que reducen la región de posibles soluciones de cada sub-problema sin eliminar ninguna solución posible del problema MIP original.

Dada la complejidad y el alto coste que conlleva resolver problemas de programación lineal con un elevado número de variables y restricciones, varias herramientas software han sido desarrolladas. De entre todas ellas, este trabajo ha utilizado el software libre LP_SOLVE v.5.5 [6]. Aunque existen otras herramientas comerciales más eficientes en términos de computación, LP_SOLVE permite el acceso a su código fuente lo cual ha sido un requisito ineludible para poder incorporarlo en el emulador de DSP utilizado.

IV. PLATAFORMA HARDWARE

La evaluación del rendimiento computacional en tiempo real ha sido realizada en TASKING DSP56xxx Software Development Toolset [7], software que emula el comportamiento en tiempo real de la familia DSP563xx de Motorola basada en procesadores de punto fijo de 24 bits. De los DSPs de esta familia, ha sido utilizado el DSP56311 para estimar el rendimiento en tiempo real del algoritmo JRRM propuesto. El DSP56311 utiliza un motor de alto rendimiento que ejecuta una instrucción por ciclo de reloj con frecuencia de reloj de 150 MHz, realizando 150 MIPS (millones de instrucciones por segundo) [8].

TASKING permite escribir aplicaciones tanto en código C/C++ como en código máquina gracias al compilador C/C++ que incorpora. La principal característica de TASKING es el depurador CrossView Pro DSP563xx que permite al usuario testear y optimizar sus aplicaciones. Esta herramienta muestra el número de ciclos de reloj y el número de iteraciones ejecutadas mediante los registros CCNT (*Clock Count*) y ICNT (*Instruction Count*), lo cual permite medir el coste de computación en tiempo real de la aplicación ejecutada conociendo la frecuencia de trabajo del reloj y el número de instrucciones que el DSP ejecuta por segundo.

V. RENDIMIENTO COMPUTACIONAL EN TIEMPO REAL

La Fig. 3 muestra el tiempo de computación requerido por el DSP56311 para resolver los problemas JRRM utilizando el algoritmo MAXILOU. A pesar de que las hojas de características indican que la familia DSP563xx incorpora un motor que ejecuta una operación por ciclo de reloj, los tiempos derivados de los valores CCNT e ICNT dados en TASKING no coinciden. Esto indica que, según dicho software, no todas las instrucciones son ejecutadas en un único ciclo de reloj, sino que en media, una instrucción es ejecutada cada 7 ciclos de reloj. En este contexto, este trabajo utiliza ambos valores para establecer los límites del rendimiento en tiempo real de la solución JRRM propuesta.

Según los resultados obtenidos a partir del valor ICNT, el tiempo necesario para obtener la solución óptima cuando 5 y

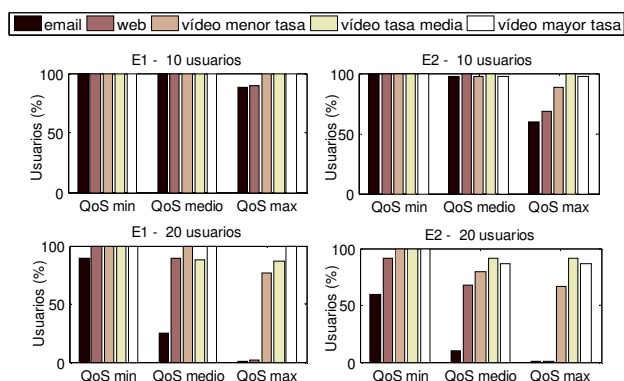


Fig. 2. Nivel de satisfacción del usuario por tipo de servicio con MAXILOU.

8 usuarios participan en cada reparto es de 0.12 y 0.58s respectivamente. Sin embargo, cuando 10 usuarios participan en cada reparto, el tiempo medio de ejecución aumenta hasta 1.15s. Aunque estos tiempos no son excesivamente altos, los resultados obtenidos comprometen la implementación de la técnica JRRM utilizando el software LP_SOLVE y la plataforma hardware DSP56311. Como muestra la Fig. 4, el aumento del tiempo de ejecución del algoritmo al aumentar la carga de usuarios en el sistema es consecuencia del mayor número de iteraciones realizadas por el método simplex, mientras que el número de nodos explorados por el algoritmo *Branch and Cut* a penas varía. Como muestra [9], el tiempo de computación podría reducirse considerablemente, hasta 100 veces según el problema, utilizando otro software de programación lineal más eficiente, como es el software comercial CPLEX [10].

Por otro lado, la aplicación de la técnica JRRM propuesta en sistemas hardware reales podría ser también posible utilizando DSP con mayores prestaciones como por ejemplo el DSP de Texas Instruments TMS320C6455 comúnmente utilizado en estaciones base 3G [11]. Este DSP trabaja con un reloj interno de 1200 MHz y palabras de 32 bits, consiguiendo gran precisión en operaciones aritméticas. Además, el TMS320C6455 utiliza 8 unidades funcionales para alcanzar máximo paralelismo en el procesado de algoritmos 3G, siendo cada unidad capaz de ejecutar una instrucción cada ciclo de reloj. Consecuentemente, el TMS320C6455 es capaz de ejecutar hasta 9600 MIPS, siendo una opción más atractiva sobre la que ejecutar la técnica JRRM propuesta y mejorar su rendimiento computacional en tiempo real. Para estimar la mejora que es posible conseguir con el TMS320C6455, ha sido considerado el peor caso en el que no es posible ejecutar instrucciones de manera simultánea por las distintas unidades funcionales, consiguiendo una ganancia igual al cociente entre las frecuencias de trabajo de los DSP (1200 MHz/150 MHz). Considerando esta potencial ganancia, la solución óptima al problema JRRM sería alcanzada en menos de 0.3s en todos los escenarios emulados, considerando los valores derivados del ICNT. La Fig. 5 muestra la función de distribución

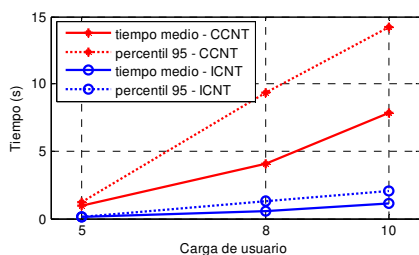


Fig. 3. Tiempo de ejecución del algoritmo MAXILOU en el DSP56311.

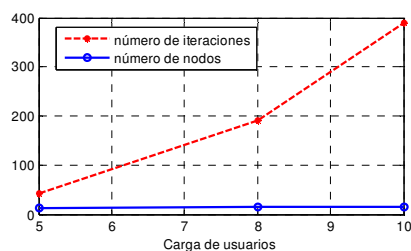


Fig. 4. Número de iteraciones ejecutadas y número de nodos explorados en la resolución del problema MIP bajo distintas situaciones de carga.

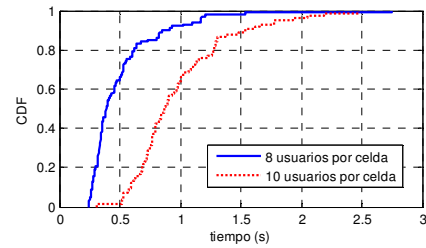


Fig. 5. CDF del tiempo de ejecución derivados del contador CCNT considerando el TMS320C6455.

acumulativa (CDF, *cumulative distribution function*) de los tiempos obtenidos a partir del contador CCNT con 8 y 10 usuarios activos en el sistema utilizando el TMS320C6455. Estos resultados muestran que tan sólo en menos del 10% y el 35% de los problemas JRRM para 8 y 10 usuarios respectivamente, MAXILOU necesita más de 1s para alcanzar la solución óptima, no superando el tiempo de computación en estos problemas los 2.5s. Es importante resaltar que mayores mejoras pueden obtenerse si varias instrucciones pudieran ser ejecutadas simultáneamente por las 8 unidades funcionales del TMS320C6455.

VI. CONCLUSIONES

Este trabajo ha investigado la potencial aplicación en sistemas hardware reales de una novedosa técnica JRRM basada en optimización y programación lineal. Los resultados obtenidos muestran que utilizando las potentes plataformas DSP actuales, los tiempos de ejecución requeridos por la política JRRM en sistemas reales serían viables. Además, mejoras adicionales son posibles utilizando herramientas de programación lineal eficientes.

AGRADECIMIENTOS

Este trabajo ha sido financiado por el Ministerio de Industria, Turismo y Comercio bajo el proyecto TSI-020400-2008-113 (propuesta CELTIC CP5-013) y por el Ministerio de Ciencia e Innovación bajo el proyecto TEC2008-06728 y la Comunidad Valenciana bajo la ayuda ACOMP/2010/111.

REFERENCIAS

- [1] Y. Guang, et al., "Joint Radio Resource Management based on the Species Competition Model", in *Proc. WCNC 2006*, Las Vegas (USA), pp. 54-57.
- [2] O. Cabral, et al., "Optimal Load Suitability Based RAT Selection for HSDPA and IEEE 802.11e", in *Proc. Wireless VITAE 2009*, Aalborg (Denmark), pp.722-726.
- [3] L. Giupponi, et al., "A novel approach for joint radio resource management based on fuzzy neural methodology", *IEEE Transactions on Vehicular Technology*, March 2008, vol. 57, pp. 1789-1805.
- [4] J. Gozálviz, M. C. Lucas-Estañ and J. Sánchez-Soriano, "Joint Radio Resource Management in Beyond 3G Heterogeneous Wireless Systems", in *Proc. WPMC 2008*, Lapland (Finland).
- [5] F. S. Hillier, G. J. Lieberman, *Introduction to operations research*, 7th edition, McGraw-Hill, 2001.
- [6] LP_SOLVE, URL: <http://lpsolve.sourceforge.net/5.5/>.
- [7] Tasking website, URL: <http://www.tasking.com/products/dsp56xxx/>.
- [8] Motorola DSP56311 Data Sheet, DSP56311/D Rev. 3, 7/2002.
- [9] S. Bayliss, et al., "An FPGA implementation of the simplex algorithm", in *Proc. IEEE FPT 2006*, Bangkok, Thailand, pp. 49-56.
- [10] IBM ILOG CPLEX, URL: <http://www-304.ibm.com/jct01003c/software/integration/optimization/cplex-dev-bundles/>.
- [11] TMS320C6455 Data Sheet, Fixed-Point Digital Signal Processor, Texas Instruments, SPRS2761, May 2005, revised April 2009.