REAL-TIME COMPUTATIONAL PERFORMANCE OF ADVANCED JRRM POLICIES IN B3G HETEROGENEOUS WIRELESS SYSTEMS

M. Carmen Lucas-Estañ and Javier Gozálvez

Ubiquitous Wireless Communications Research (Uwicore) Laboratory, http://www.uwicore.umh.es/
University Miguel Hernández
Avda. Universidad s/n, 03202, Elche, Spain
phone: + (34) 966658955, fax: + (34) 966658903,
email: m.lucas@umh.es, j.gozalvez@.umh.es

ABSTRACT

The future Beyond 3G wireless systems will be characterized by the coexistence of multiple Radio Access Technologies. In these heterogeneous systems, the coordinated use of the common radio resources is a challenging task, and efficient Joint Radio Resource Management (JRRM) techniques that guarantee adequate satisfaction levels to all users and maximize the system revenue need to be designed. To achieve the optimal solution to each situation, JRRM decisions are based on an increasing number of parameters and variables, and a key aspect in the design of these JRRM techniques is to achieve good computational performance to be applied in real hardware systems. This paper proposes a novel JRRM technique based on linear programming optimization, and its real-time computational cost is analysed to evaluate its implementation feasibility. In this study, DSPs commonly used in mobile base stations are considered to demonstrate its potential to be employed in Beyond 3G heterogeneous systems.

1. INTRODUCTION

There is a wide consensus in the research community that Beyond 3G wireless networks will be characterized by the coexistence and cooperation of a wide variety of Radio Access Technologies (RATs) with diverse but also complementary technical characteristics. This seamless integrated environment will allow network operators to better accomplish with the diverse and each time more demanding user requirements by means of a coordinate management of the common available radio resources usually referred as Joint Radio Resource Management (JRRM). In this context, the main challenge of the research community is the design of JRRM policies ensuring the RAT interoperability. The JRRM policies must decide for each incoming call the RAT over which it will be conveyed (RAT selection) and the number of radio resources within the selected RAT (intra-RAT RRM) that will be necessary to satisfy the user Quality of Service (QoS) requests. Furthermore, the JRRM policy should be capable to dynamically adapt to the current operating conditions, for example system load and multimedia user distribution, with the aim of providing adequate QoS levels to users while achieving maximum system revenue.

The JRRM concept, also referred as Common Radio Resource Management (CRRM), is defined by the 3GPP in [1], where also protocol and architecture solutions are proposed. The majority of studies only tackle the RAT selection dilemma when investigating JRRM policies. For example, [2] mapped the RAT selection dilemma into the competition between species model. In this context, each RAT adapts some parameters (price and support bandwidth) according to its current system conditions in order to attract or dismiss users from accessing this network. In [3], a RAT selection algorithm that assigns the user to the most suitable RAT is proposed. The RAT suitability is based on the current RAT load and a pre-established load threshold, which are empirically calculated with the aim of achieving the maximum throughput gain.

On the other hand, a proposal to jointly address the RAT selection and intra-RAT RRM dilemmas has been proposed in [4]. The proposed JRRM algorithm is based on neural networks and fuzzy logic, and simultaneously determines the most appropriate RAT and bit rate allocation considering among other factors the user QoS constraints. However, the proposed JRRM technique does not establish the number of radio resources that has to be assigned to each user. In this context, the work reported in [5] proposed a novel JRRM proposal that simultaneously assigns to each user an adequate combination of RAT and number of radio resources within such RAT to guarantee the user/service QoS requirements. This technique is based on linear programming and optimization techniques, and its potential to be applied in heterogeneous wireless systems was demonstrated in [5].

Given that JRRM decisions are based on an increasing number of variables and data, the JRRM processing time is becoming an important factor that can determine the feasibility of implementing the proposed techniques in real hardware systems. This is particularly the case for techniques looking for optimal or suboptimal solutions with powerful optimization tools such as those reported in [4] and [5]. In addition, the majority of studies in the mobile communication field are based on computer simulation, and few of them evaluate the computational cost of novel proposals in real systems. To the author's knowledge, there are currently no published studies of the real-time hardware computational cost of advanced JRRM policies, in particular

of JRRM policies jointly addressing the RAT selection and intra-RAT RRM dilemmas. In this context, this work investigates for the first time the real-time performance of the proposed JRRM technique in real systems with the aim of analysing its implementation feasibility.
The rest of the paper is organized as follows. Next section presents the JRRM proposal that is analysed, and its system and QoS performance. Section 3 briefly describes the linear programming resolution tools employed to solve the JRRM dilemma, while the hardware platform employed to investigate the computational cost of the proposed techniques is presented in Section 4. Finally, the real-time computational performance of the JRRM policy is presented in Section 5.

2. JRRM TECHNIQUE

The JRRM technique analysed in this work [5] is aimed at deciding the optimal RAT and number of radio resources within this RAT that should be assigned to each active user in a heterogeneous system according to the current load system conditions and service requirements. In this context, the JRRM technique exploits the QoS/resource flexibility offered by different services in a multimedia framework to achieve maximum system revenue while guaranteeing adequate QoS satisfaction level to users; this work considers a heterogeneous system composed by GPRS, EDGE, and HSDPA with email, web and real-time video users.
To achieve the optimal solution to the common radio resource distribution problem, the definition of the JRRM technique is based on linear programming and optimization techniques. The linear programming tools, which are presented in Section 3, achieve the optimal solution to problems modelled by means of linear equations. In this context, the objective of the JRRM problem, and the system and services conditions that restrict the optimum solution to the problem have to be expressed as linear functions.

2.1 MAXILOU technique

The designed JRRM technique carries out the distribution of the radio resources based on a user fairness policy: the technique tries to provide similar, and highest possible, satisfaction levels for all users in the heterogeneous system and only when it is not possible due to radio resource shortage, service priorities will be applied. In this work, the user satisfaction is represented by utility values that indentify the RAT and the number of radio resources within this RAT needed per service class to achieve certain user QoS satisfaction levels. To establish these utility values, the utility functions depicted in Figure 1 were previously defined for the considered services [5]. These utility functions try to express the perceived user satisfaction as link quality varies; the utility functions are based on user throughput for web and email transmissions, and on the percentage of correctly transmitted frames for real-time video services. The definition of the utility functions shown in Figure 1 is based on the QoS satisfaction thresholds established in Table 1. It is important to highlight that users

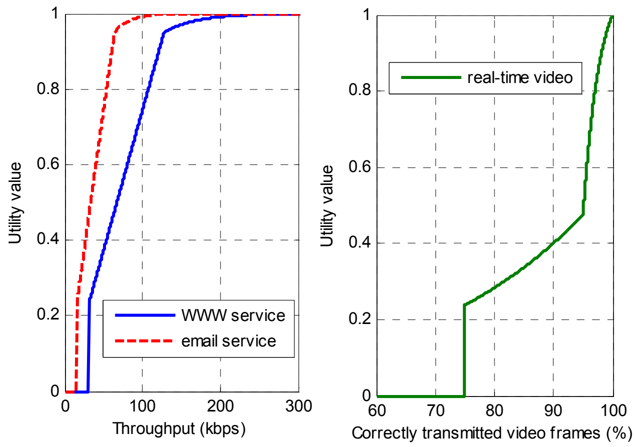

Figure 1 – Utility functions per traffic service.

	Min. QoS	Mean QoS	Max. QoS
WWW	32kbps	64kbps	128kbps
Email	16kbps	32kbps	64kbps
Established utility values	0.95/4	0.95/2	0.95
H.263 video	75%	95%	100%
Established utility values	0.95/4	0.95/2	1

Table 1 – User QoS levels per traffic service.

do not perceive a non-null utility value until its minimum QoS value is satisfied. Furthermore, the web and email utility between minimum and maximum QoS requirements linearly grow with the experienced throughput, while for real-time video users, the utility increases with the percentage of transmitted frames is slow until a high percentage of correctly transmitted frames is achieved given that an acceptable video quality requires a high percentage of correctly received video frames [6]. This definition has been established to guarantee the high demanding requirements of real-time services.
Once the utility functions are established, it is then necessary to relate the utility values with the different radio resources assignments. To establish this relation, the throughput achieved by each RAT and number of radio resources combination is considered. The simulated radio access technologies implement Link Adaptation or Adaptive Modulation and Coding schemes that dynamically vary the used transmission mode. In this case, it is difficult to estimate the throughput that could be achieved with a given number of radio resources, and an intermediate transmission mode providing a balance between data rates and error protection has been selected in each RAT to estimate the throughput. In particular, the coding scheme CS2 and the modulation and coding scheme MCS5 providing 13.4 and 22.4 kbps respectively have been selected for a timeslot in GPRS and EDGE. In HSDPA, various transmission modes can be defined per assigned code. To achieve the sought data rate/error correction balance, the selected transmission rate per assigned HSDPA code corresponds to that achieved by the 'intermediate' transmission mode out of all possible

modes for a given number of HSDPA codes. This work considers the transmission modes defined in the CQI (Channel Quality Indicator) mapping table for User Equipment category 10 [7]. In the example shown in Table 2, the intermediate transmission modes would correspond to CQI value 3 for 1 HSDPA code, and CQI 8 for 2 HSDPA codes. Table 3 shows an example of the final utility values achieved for the real-time 64 kbps H.263 video users with the RAT/resources combination shown in increasing throughput order. In this table, the resources/RAT combination is denoted as *xY*, corresponding to *x* radio resources (timeslots or codes) from RAT *Y* (GPRS is represented as *G*, EDGE as *E*, and HSDPA as *H)*. It is interesting to note that certain RAT/resources combinations cannot achieve utility values greater than zero.

Once the utility values have been established, the JRRM problem can be modelled. To achieve its final objective, the JRRM technique seeks to maximize the lowest utility value assigned to a user in a common radio resources distribution round. Therefore, the JRRM technique is referred as MAXILOU (MAXImise LOwest Utility), and its objective function is expressed as follows:

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

where u_j represents the utility value assigned to user *j* in a common radio resources distribution round, and *N* corresponds to the total system user load. Under equal service and operative constraints, (1) is achieved when utility values are equally distributed among users according to the JRRM objective of the problem. In order to apply linear programming tools, (1) has to be expressed as a linear equation. To this aim, a new real variable denoted as *z* and equal to the smallest utility value assigned to a user is

CQI value	Data Rate (kbps)	Codes	CQI value	Data Rate (kbps)	Codes
1	68.5	1	6	230.5	1
2	86.5	1	7	325	2
3	116.5	1	8	396	2
4	158.5	1	9	465.5	2
5	188.5	1			

Table 2 – Extract of HSDPA CQI Mapping Table for UE Category 10.

Res./ RAT	Data rate (kbps)	Utility value	Res./ RAT	Data rate (kbps)	Utility value	Res./ RAT	Data rate (kbps)	Utility value
1G	13.4	0.0000	4E	89.6	0.2983	3H	741	1.0000
1E	22.4	0.0000	7G	93.8	0.3127	4H	1139.5	1.0000
2G	26.8	0.0000	8G	107.2	0.3532	5H	2332	1.0000
3G	40.2	0.0000	5E	112	0.3654	7H	4859.5	1.0000
2E	44.8	0.0000	1H	116.5	0.3775	8H	5709	1.0000
4G	53.6	0.0000	6E	134.4	0.4350	10H	7205.5	1.0000
5G	67	0.0000	7E	156.8	0.9338	12H	8618.5	1.0000
3E	67.2	0.0000	8E	179.2	0.9819	15H	11685	1.0000
6G	80.4	0.0000	2H	396	1.0000			

Table 3 – 64kbps video utility values.

defined, which results in the following objective function:

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

Due to the fact that utility values assigned to users can only take specific values within a finite set, u_j is expressed as shown in (3).

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

In where, $u_j^r(s^r)$ represents the utility value obtained by user *j* when assigned *s* radio resources (codes or timeslots) of RAT *r* (*r* is equal to 0, 1 or 2 for GPRS, EDGE and HSDPA respectively), and $s \in [1, c_r]$ with c_r corresponding to the maximum number of radio resources available at each RAT. $y_j^{r,s}$ is a binary variable equal to one if user *j* is assigned *s* radio resources of RAT *r*, and equal to 0 if not. Therefore, MAXILOU focuses on deciding for each user which $y_j^{r,s}$ variable is equal to one, considering that only $y_j^{r,s}$ variables achieving a utility value greater than zero are allowed. With the defined objective function, Mixed Integer Linear Programming (MIP) mechanisms that will be presented in Section 3 can be applied to solve the JRRM problem.

Once the basic objective function has been established, the problem statement must be completed with some system and service constraints. The first one is derived from the expression (3). In this expression, it is important to note that only one $y_j^{r,s}$ variable can be equal to one for each user, which is expressed in (4).

$$\sum_r \sum_s y_j^{r,s} = 1,\ \forall j \quad (4)$$

Under high system loads, (4) might not be feasible due to the limited radio resource availability, which is expressed in constraint (5).

$$\sum_j \sum_{s^r} s^r y_j^{r,s} \le c_r,\ \forall r \quad (5)$$

In these scenarios, the number of users requesting resources will be reduced so that (4) and (5) are satisfied.

Each time a new user requests access to the system or a transmission ends, the JRRM radio resources distribution is performed among all active users. In this case, a minimum QoS level will be guaranteed to active real-time video users that were assigned resources in the previous JRRM distribution round. These real-time video users will compete for additional resources with other users. This condition is expressed in (6).

$$\sum_r \sum_{s^r} E_{s^r} y_j^{r,s} \ge E_{\min},\ \forall j_{h263} \quad (6)$$

where E_{s^r} represents the ordinal number corresponding to the assignment of *s* radio resources in RAT *r* when the possible radio resource assignments are put in increasing transmission rate order. Similarly, E_{min} represents the ordinal number of the radio resource assignment satisfying the minimum QoS level.

As previously said, MAXILOU tries to apply a user fairness policy, and only when it is not possible to fully satisfy all users due to the shortage of radio resource, the following service priorities are applied: real-time H.263 video (higher priority), web, and email; real-time video users with higher video bit rates are served first. If a user m is a video user that obtained radio resources in the previous JRRM distribution round, the condition established in (6) comes first, and the video user would be assigned the s_{min} radio resources from RAT r_{min} necessary to achieve its minimum QoS level defined by E_{min}. When such level is achieved, the user m will not be assigned additional resources until higher priority users (represented by k) surpass its utility value ($u_m^{r_{\min}}(s_{m_{\min}})$). This constraint is expressed as:

where (r_a,s_a) represent the RAT/resources assignments that verify $u_m^{r_{\min}}\left(s_{m_{\min}}\right)-u_k^{r_a}\left(s_a\right)>0$ and (r_b,s_b) the assignments that verify $u_m^{r_{\min}}\left(s_{m_{\min}}\right)-u_k^{r_b}\left(s_b\right)\le 0$; this condition is only applied when the priority of user k is higher than that of user m. According to the defined service priorities, if all active users cannot obtain their minimum QoS demand, i.e. there is no solution to the JRRM problem modelled, users with the lowest priority will be eliminated from the JRRM distribution round until a linear programming JRRM solution can be achieved.

2.2 MAXILOU performance

The performance of the MAXILOU technique is presented in Figure 2. The proposed JRRM algorithm has been evaluated in a simulation platform that emulates a heterogeneous system composed of the GPRS, EDGE and HSDPA networks. The user load is equally distributed among email, web and real-time video transmissions. Two different scenarios have been emulated: the first one (E1) simulates real-time video users with 16, 64, and 128 kbps mean bit rates, and the second one (E2) simulates 64, 256, and 512 kbps mean bit rates real-time video users. Total user loads of 10 and 20 users per cell have been emulated in both scenarios.

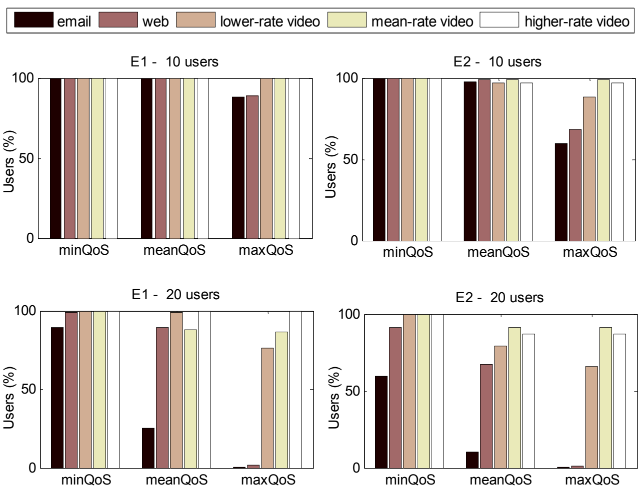

Figure 2 – Achieved user satisfaction levels per service class with the maximum JRRM policy.

Figure 2 depicts the percentage of users per service class that achieved the minimum, mean, and maximum QoS satisfaction level established in Table 1. Firstly, it is important to note that the simulated system conditions result in resource shortage situations, and therefore it is not possible to achieve high QoS satisfaction levels for all users in all scenarios. In this context, Figure 2 shows that the proposed aims have been achieved by the MAXILOU technique: all services obtain the highest and most homogeneous possible QoS levels, and services priorisation effects are most notable when the radio resources demand increases and homogeneous QoS levels cannot be achieved across all service types. When the system load or service

$$\sum_{r_a}\sum_{s_a} u_m^{r_{\min}}\left(s_{m_{\min}}\right)\cdot y_k^{r,s} + \sum_{r_b}\sum_{s_b} u_k^{r}\left(s^r\right)\cdot y_k^{r,s} \geq \sum_{r}\sum_{s} u_m^{r}\left(s^r\right)\cdot y_m^{r,s} \quad (7)$$

requirements increase, a lower number of low priority users achieve the highest QoS satisfaction levels. Under a system load of 20 users, a certain percentage of low priority users do not even receive resources to satisfy their minimum QoS level, while a high percentage of the most demanding users, the real-time video users, achieve their maximum QoS satisfaction level. This is due to two reasons. The first one is that several services require more than one radio resource from the different RATs to achieve their minimum QoS satisfaction threshold. The second reason is the fact that if low priority users, such as email, were assigned radio resources initially assigned to higher priority users to pass from their lower QoS levels to higher ones, they will obtain a utility value (or QoS satisfaction) higher than that achieved by higher priority users, which is not allowed by the service priorisation constraint (equation (7)). Therefore, MAXILOU serves the maximum possible number of users with the highest and most homogeneous possible QoS satisfaction levels satisfying system and service constraints. A more in-depth analysis of the performance of the MAXILOU JRRM technique can be found in [5].

3. LINEAR PROGRAMMING TOOLS

The MAXILOU technique models the radio resource distribution problem in a heterogeneous system by means of a series of linear functions with binary and real variables that express the objective and constraints of the studied situation. This type of problems referred as MIP problems are studied in Operations Research, and several approaches have been proposed to solve them [8]. The majority of these MIP solver mechanisms are based on the simplex method [8]. The simplex method is regularly employed in linear programming problems with a large number of real variables, and that require computationally efficient solutions. The simplex method is an algebraic procedure that makes use of the fact that the spatial region containing the set of possible solutions is limited by the linear constraints present in the problem statement. In these situations, it is demonstrated that the optimal solution that maximizes (or minimizes) the objective function is placed in a vertex of this feasible region. The simplex method exploits this fact and searches the optimal solution moving from one to another vertex of the feasible region. The simplex method is only applicable to linear programming problems with real variables, while other methodologies are then applied to also consider integer variables.

One of the most commonly used approaches to solve MIP problems due to its performance and computational properties is the Branch and Bound (B&B) method [8]. The B&B method solves an ordered sequence of reduced linear programming problems until an optimum solution is achieved. Several problems are derived from the original problem by dividing the set of possible values for every unknown variable in smaller ones. Then, for each of these problems the integer condition is relaxed allowing real values for all the variables, and the simplex method is applied to solve them. Depending on the solution achieved for each subproblem, the given branch can be ruled out if a feasible solution can not be achieved, or a better solution has been achieved in another branch (or subproblem) and no more subproblems will be derived from this branch. This process is executed until an integer solution is achieved for one of the subproblems and all the other branches are ruled out.

The methodology used in this work to solve the JRRM problem is the Branch and Cut method [8], which is an improved version of the B&B mechanism that incorporates the use of cutting planes. The cutting planes are new functional constraints that reduce the feasible solution region of the relaxed linear programming problem without eliminating feasible solutions to the original MIP problem. Additional information about the definition of the cutting planes can be obtained in [8].

3.1 Linear programming solvers

Due to the complexity and cost of solving linear programming problems with a high number of variables and constraints, several mathematic software tools have been developed to tackle this issue. The most widely used tools are ILOG CPLEX [9], a commercial software tool, and LP_SOLVE [10], a free open source solver. CPLEX is highly employed by the research community due to its high computational performance achieved through some of the fastest and most efficient algorithms to solve mathematic optimization problems with high computational requirements. On the other hand, LP_SOLVE is released under the LGPL (the GNU lesser general public license) license, and many researches have contributed to its development.

Several studies have compared the performance of both solutions. For example, the work reported in [11] analyses their performance for linear programming problems applying the simplex method to solve them. According to [11], CPLEX results 100 times faster than LP_SOLVE executing the simplex method for the evaluated linear programming problems. In [12], the CPU and user time spent to solve a high number of different MIP problems is studied for different linear programming solvers. The obtained results highlight the higher computational performance of CPLEX compared with LP_SOLVE. For example, while LP_SOLVE did not achieve the optimal solution to most of the analysed MIP problems in less than 2 hours, CPLEX solved them in less than 1 minute.

To analyse the computational performance of the MAXILOU technique in real hardware systems, the proposed JRRM technique and the tools used to solve them have to be implemented in the DSP emulator software used in this work and that is presented in Section 4. To this aim, having access to the code of the MIP solver is a needed requirement, and this is not possible with CPLEX due to its commercial nature. For this reason, and despite its worse performance, the open source LP_SOLVE solution has been used to implement in the DSP emulator platform the MIP solver needed by the proposed JRRM technique. In particular, this work employs the LP_SOLVE 5.5 version.

4. HARDWARE PLATFORM

The evaluation of the real-time computational performance of the proposed JRRM technique has been carried out using the TASKING DSP56xxx Software Development Toolset [13], which emulates the real-time behaviour of the Motorola DSP563xx family. The DSP563xx family provides between 100 and 150 million of instructions per second (MIPS), and is based on 24-bit fixed-point DSP processors. Applications in the TASKING Software Development Toolset can be written in C/C++ code, as well as in machine code, thanks to C/C++ compiler tools. The main feature of the Tasking emulator software is the CrossView Pro DSP563xx debugger that allows users to test their application code and optimize it. In the debug session, the C/C++ code, or the corresponding machine code, can be shown and also the value of memory positions, registers, or variables can be monitored. Furthermore, the CrossView Pro debugger provides some program performance analysis tools, such as the Clock Count register (CCNT) and the Instruction Count register (ICNT), enabling to measure the real-time computational cost of the application that is being executed.

To estimate the real-time performance of the proposed JRRM algorithm, the DSP56311 has been selected among the DSPs members of the DSP563xx family. The DSP56311 is widely used in network applications with general filtering operations. Like the other DSP56300 family members, the Mototola DSP56311 uses a high-performance, single-clock-cycle-per-instruction engine, a barrier shifter, 24-bit addressing, an instruction cache, and a direct memory access controller [14]. The DSP56311 incorporates an Enhanced Filter Coprocessor (EFCOP) that executes filter algorithms in parallel with core operations enhancing signal quality with no impact on channel throughput or total channels supported. The DSP56311 operates at 150 MIPS, attaining 270 MIPS when the EFCOP is in use, although the EFCOP operation has not been considered in this work. It operates with an internal 150 MHz clock with a 1.8 volt core and independent 3.3 volt input/output power, and has available a total of 128K on-chip memory.

5. REAL-TIME COMPUTATIONAL PERFORMANCE ANALYSIS

With the aim of evaluating the potential application of the proposed JRRM algorithm in real systems, its real-time computational performance is analysed in this section. This study is based on the DSP56311, which offers 150 MIPS using an internal 150MHz clock [14]. The proposed JRRM algorithm and the LP_SOLVE linear programming tool used to solve the MIP problems have been implemented in the TASKING DSP56xxx Software Development Toolset (Section 4). Before describing the obtained results, it is important to note that the Tasking software provides a measure of the computational performance by means of the CCNT and the ICNT counters. The DSP563xx family datasheet [14] indicates that the DSP incorporates a single-clock-cycle-per-instruction engine. However, in the experiments conducted using the Tasking software, the CCNT and ICNT registers provided different values, which indicates that according to the Tasking software, not all the instructions are executed in a single clock cycle. In this context, this work estimates the time needed by the DSP to solve the JRRM problem using both CCNT and ICNT counters.

Figure 3 depicts the computational time needed by the DSP56311 to resolve the JRRM problems following the MAXILOU proposal. The real-time computational performance has been estimated using the CCNT and ICNT counters. The figure shows the average performance and the 95 percentile for all the simulated JRRM distribution rounds. The real-time needed by the DSP56311 to resolve a JRRM distribution round estimated using the CCNT counter has been obtained by dividing the number of elapsed clock cycles by the frequency of the internal clock. The equivalent time estimated using the ICNT counter has been obtained dividing the value of executed instructions by the number of instructions per second that the DSP is able to execute. The illustrated results show that the ratio between the time values derived from the CCNT counter and the ones derived from the ICNT counter is in average 7 clock cycles. In this context, this work uses both estimated times to establish real-time performance bounds of the proposed and implemented JRRM solution. Using the ICNT counter, the average real-time needed by the JRRM technique to found an optimum solution when 5 and 8 users are participating in the radio resources distribution process is 0.12s and 0.58s respectively. With higher user loads, the average time increases up to 1.15s. As observed in Figure 4, as the number of users participating in a JRRM radio resources distribution round increases, the average number of iterations executed by the simplex method increases since there is a higher number of variables and constraints to be considered in the resolution of the JRRM problem. On the other hand, the average number of nodes explored by the Branch and Cut method only slightly varies with the user load, which highlights that the number of iterations is at the origin of the increase of the JRRM execution time with the user load. The results illustrated in Figure 3 also emphasize that the JRRM execution time further increase when analyzing the 95 percentile of the emulated JRRM radio resources distribution rounds. The observed time estimates compromise the implementation of the proposed JRRM policy using the LP_SOLVE tool and the DSP56311 hardware platform.

The applicability of the proposed JRRM technique can be improved with higher performance DSPs commonly used in 3G base stations where higher capacities and processing data rates are required. For example, the TMS320C6455 of Texas Instruments is one of the highest-performance fixed-point DSPs in the TMS320C60000™ DSP family [15]. It operates with an internal 1200 MHz clock and works with a 32 bit word, achieving a high accuracy in arithmetic operations. The TMS320C6455 core employs eight functional units to achieve maximum parallelism in processing 3G algorithms, with each unit capable of executing one instruction every clock cycle. Consequently,

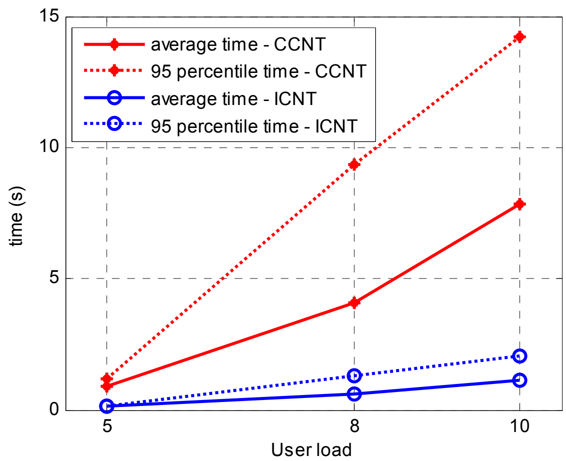

Figure 3 – Time needed by the Max_MinU algorithm to find a JRRM optimal solution using the DSP56311.

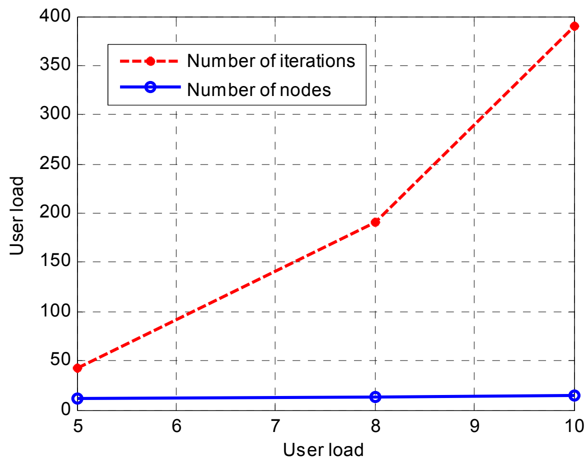

Figure 4 – Number of iterations executed and nodes explored in the MIP problem for varying user loads.

the TMS320C6455 is capable to execute up to 9600 MIPS, and represents then a more attractive solution over which to execute the proposed JRRM technique and improve its real-time computational performance. To estimate the potential improvement achieved with the TMS320C6455 hardware platform compared to the DSP56311 device, the worst case at which no code instructions can be executed simultaneously by the eight functional units is considered. In this case, only one instruction per clock cycle is executed, achieving a gain equal to the ratio between the DSP clock frequencies (1200MHz/150MHz). Figure 5 represents the execution times that could be achieved with the more efficient and powerful TMS320C6455 DSP. Figure 5 has been achieved by scaling Figure 3 with the clock frequency gain obtained when improving the DSP performance. The obtained results show that considering the ICNT counter values, the optimum solution to the JRRM problem is achieved in less than 0.3s in all the emulated scenarios. Figure 6 depicts the cumulative distribution function (CDF) of the CCNT derived computational time spent by the

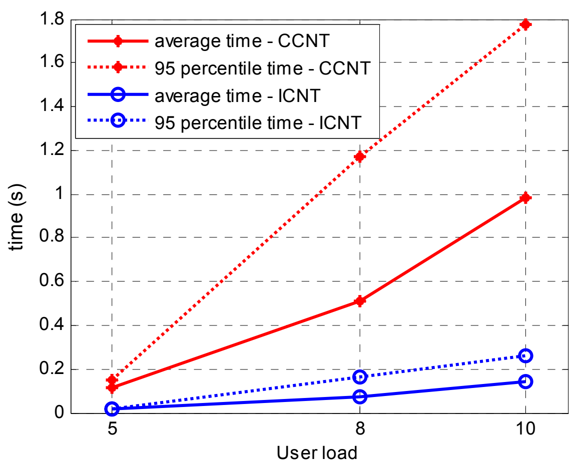

Figure 6 – Time needed by the Max_MinU algorithm to find a JRRM optimal solution using the TMS320C6455.

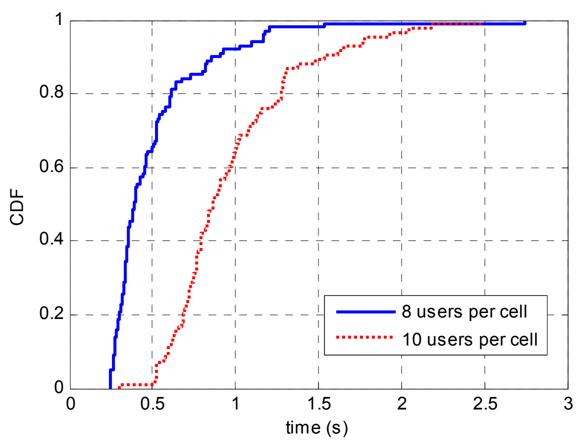

Figure 6 – CDF of the CCNT-derived real computational time (s) for the JRRM Max_MinU proposal considering the TMS320C6455.

JRRM proposal to achieve the optimum solution in the scenarios considering 8 and 10 users per cell. The data depicted in Figure 5 and 6 show that only in less than 10% and 35% of the JRRM problems under 8 and 10 user loads respectively, the proposed MAXILOU technique requires more than 1 second to achieve the optimal solution, and in these problems the time spent is not higher than 2.5s. It is important to note that further improvements could be obtained if several instructions could be executed simultaneously by the eight functional units of the TMS320C6455 hardware platform. In addition, it is important to remember that the hardware implementation of the MAXILOU JRRM proposal has been conducted using the LP_SOLVE tool, a non-optimal solution that has been considered given the possibility to access its source code. Higher real-time execution gains could be achieved when using the CPLEX algorithms to solve the proposed JRRM technique (Section 3.1).

6. CONCLUSION

This work has investigated the potential applicability in real systems of a JRRM technique based on lineal programming optimization. To this aim, the authors have implemented the technique in a platform for the real-time evaluation of hardware DSPs using a linear programming solver with accessible source code. The obtained results show that using current powerful DSP platforms, the advanced JRRM policy evaluated in this work would achieve feasible execution times to be applied in real B3G systems. Additional improvements could be further obtained with more efficient linear programming solvers.

7. ACKNOWLEDGEMENT

This work has been supported by the Ministry of Industry, Tourism and Trade under the project TSI-020400-2008-113 (CELTIC proposal CP5-013), the Ministry of Education and Science (Spain) and FEDER funds under the projects TEC2005-08211-C02-02, and the Ministry of Science and Innovation under the project TEC2008-06728.

REFERENCES

[1] 3GPP, *Improvement of Radio Resource Management (RRM) across RNS and RNS/BSS, TR 5.881, v5.0.0*, January 2002.

[2] Yang Guang, Chan Jie, Yu Kai, Zhang Ping, Victor O. K. Li, "Joint Radio Resource Management based on the Species Competition Model", in *Proc. IEEE Wireless Communications and Networking Conference 2006 (WCNC 2006)*, April 2006, Las Vegas, Nevada (USA), pp. 54-57.

[3] O. Cabral, F.J. Velez, J. Rodriguez, V. Monteiro, A. Gameiro, N.R. Prasad, "Optimal Load Suitability Based RAT Selection for HSDPA and IEEE 802.11e", in *Proc. 1st International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology (Wireless VITAE 2009)*, May 2009, Aalborg (Denmark), pp.722-726.

[4] L. Giupponi, R. Agusti, J. Pérez-Romero, O. Sallent, "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.
[5] J. Gozálvez, M. C. Lucas-Estañ and J. Sánchez-Soriano, "Joint Radio Resource Management in Beyond 3G Heterogeneous Wireless Systems", in *Proc. 11th International Symposium on Wireless Personal Multimedia Communications (WPMC)*, Lapland (Finland), 2008.
[6] L. Hanzo, P. Cherriman, J. Streit, *Wireless video communications: second to third generation systems and beyond*, IEEE Press, 2001.
[7] *3GPP, Technical Specification Group Radio Access Network; Physical layer procedures (FDD)*, 3GPP TS 22.214, version 7.3.0, 2006.
[8] F. S. Hillier, G. J. Lieberman, *Introduction to operations research*, 7th edition, McGraw-Hill, 2001.
[9] ILOG CPLEX, URL: http://www.ilog.com/products/cplex/index.cfm
[10] LP_SOLVE, URL: http://lpsolve.sourceforge.net/5.5/.
[11] S. Bayliss, C.-S. Bouganis, G. A. Constantinides, W. Luk, "An FPGA implementation of the simplex algorithm", in *Proc. 2006 IEEE International Conference on Field Programmable Technology (FPT 2006)*, Bangkok, Thailand, 2006, pp. 49-56.
[12] Benchmarks for Optimization Software by Hans Mittelmann, Arizona State University. Available: http://plato.asu.edu/bench.html.
[13] Tasking website, URL: http://www.tasking.nl/products//dsp/dsp56xxx
[14] Motorola DSP56311 Data Sheet, DSP56311/D Rev. 3, 7/2002.
[15] TMS320C6455 Data Sheet, Fixed-Point Digital Signal Processor, Texas Instruments, SPRS276I, May 2005, revised April 2009.