

# Reduced Complexity Optimal Resource Allocation for Enhanced Video Quality in a Heterogeneous Network Environment

Bharat Agarwal, *Student Member, IEEE*, Marco Ruffini, *Senior Member, IEEE*, and Gabriel-Miro Muntean, *Senior Member, IEEE*

**Abstract**—The latest Heterogeneous Network (HetNet) environments, supported by 5th generation (5G) network solutions, include small cells deployed to increase the traditional macro-cell network performance. In HetNet environments, before data transmission starts, there is a user association (UA) process with a specific base station (BS). Additionally, during data transmission, diverse resource allocation (RA) schemes are employed. UA-RA solutions play a critical role in improving network load balancing, spectral performance, and energy efficiency. Although several studies have examined the joint UA-RA problem, there is no optimal strategy to address it with low complexity while also reducing the time overhead. We propose two different versions of simulated annealing (SA): Reduced Search Space SA ( $RS^3A$ ) and Performance-Improved Reduced Search Space SA ( $PIRS^3A$ ), algorithms for solving UA-RA problem in HetNets. First, the UA-RA problem is formulated as a multiple knapsack problem (MKP) with constraints on the maximum BS capacity and transport block size (TBS) index. Second, the proposed  $RS^3A$  and  $PIRS^3A$  are used to solve the formulated MKP. Simulation results show that the proposed scheme  $PIRS^3A$  outperforms  $RS^3A$  and other existing schemes such as Default Simulated Annealing (DSA), and Default Genetic Algorithm (DGA) in terms of variability and DSA and  $RS^3A$  in terms of Quality of Service (QoS) metrics, including throughput, packet loss ratio (PLR), delay and jitter. Simulation results show that  $PIRS^3A$  generates solutions that are very close to the optimal solution.

**Index Terms**—User Association, Resource Allocation, 5G, HetNet, Simulated Annealing, Multiple Knapsack problem, Combinatorial Optimization

## I. INTRODUCTION

**T**HERE has been an extensive increase in Internet traffic in the last decade due to the significant increase in the number of devices and users along with the diversification of rich media services such as Video on Demand (VoD), Video Conferencing (VC), Augmented Reality (AR), and Virtual Reality (VR). This has contributed to a massive increase in data traffic that puts significant pressure on the current deployed communication networks capacity, yielding reduced Quality of Service (QoS) levels. For instance, the worldwide mobile traffic, which was 14 Exabytes/month at the end of 2017, is expected to increase to 110 Exabytes/month by 2023 [1]. No single technology and approach can address the challenge of accommodating this traffic increase. Instead, a Heterogeneous Network (HetNet) paradigm is employed, for example, in [2]. A HetNet consists of an existing cellular network environment enriched with a variety of smaller networks with simpler base stations (BSs), with broadly varying transmission capacities,

coverage areas, carrier frequencies, back-haul link types, and communication protocols. Indeed, the deployment of small cells within a macrocell can provide support in terms of higher communication speed and better coverage for mobile users located at the macrocell border or in regions with high traffic demand [2].

In particular, the integration of femtocell BSs (FBSs) with macro-cell BSs (MBSs) has drawn considerable attention recently. Fig. 1 illustrates such a macro-femtocell-based HetNet deployment scenario in which MBSs and FBSs collectively serve the users and can provide improved QoS levels [3]. User association (UA) to BSs and resource allocation (RA) during data transmission are primary challenges in such a HetNet environment. As a result, the dual UA-RA problem needs to be examined to enable high QoS support while considering variables such as BS capacity, user requirements, and channel quality.

We have formulated the UA-RA problem as multiple knapsack problem (MKP) in our paper, which can be broadly described as follows: Given a set of items, each of which has a weight and a value, it is essential to measure the amount of each item to be included in a knapsack so that the total weight is smaller than the capacity of the knapsack and the total value is as high as possible [4]. A collection of  $m$  knapsacks with various capacities is given in the case of the MKP. The knapsacks are represented by BSs (MBS and FBS) in this solution, and the items to fit into the knapsacks are represented by instances of user equipment (UE). The item weights are user demands, while item values are the available throughput for each knapsack. Knapsack capacity is the maximum capacity available. We use simulated annealing (SA) to solve the MKP problem, and based on the resource block (RB) utilization rate and the transport block size (TBS) index, the available throughput is determined.

This paper introduces two innovative SA solutions: Reduced Search Space Simulated Annealing ( $RS^3A$ ) and Performance-Improved Reduced Search Space Simulated Annealing ( $PIRS^3A$ ).  $RS^3A$  performs fine-tuning of some parameters such as to predict the best possible starting solution. A good choice of starting solution helps reduce the search space for SA, which as any meta-heuristic algorithm results in solving combinatorial optimization problems faster and more efficient.  $PIRS^3A$  also performs search space reduction, which helps increase the likelihood of selecting UEs with the highest estimated throughput, while also considering other

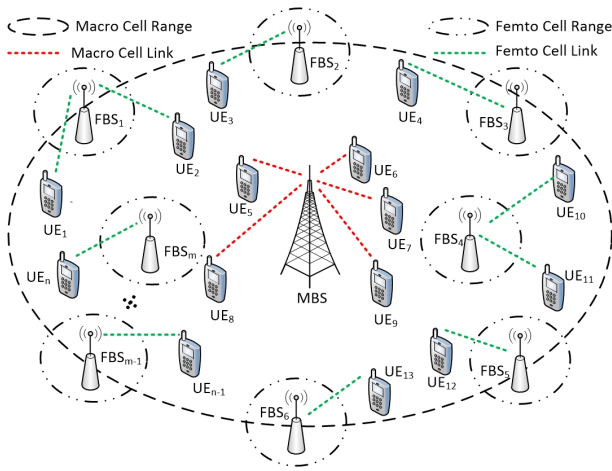


Figure 1. A two-tier macro-femtocell-based HetNet

parameters such as BS capacity, UE requirements, and channel efficiency. However,  $PIRS^3A$  goes one step further and also involves removal of ineffective solutions. The problem search space often contains many ineffective or infeasible solutions, which a typical SA algorithm considers when evaluating the target function, wasting time and effort.  $PIRS^3A$  removes the ineffective solutions from the solution space before applying SA to solve MKP and finds a good solution faster and in a more efficient manner.

This paper's principal contributions are as follows:

- 1) We propose two improved versions of SA, i.e.,  $RS^3A$  and  $PIRS^3A$ . We have shown that our proposed scheme  $PIRS^3A$  converges to a bounded near-optimal solution and outperforms two alternative SA-based approaches and Default Genetic Algorithm (DGA) in terms of variability and overhead time.
- 2) Under the maximum BS capacity and TBS index constraint, we introduced  $PIRS^3A$  as a decentralized scheme for solving the UA-RA problem in HetNets as MKP. We have also demonstrated that optimization function represents a *hyper-plane* and its *convex* and maximum BS capacity, TBS index constraints are *convex* and separately reflect *half-spaces* and intersection of *half-spaces* and *hyper-planes* form a *polyhedron*.
- 3) We present simulation results to show the effectiveness of the proposed scheme with different system parameters. The performance of  $PIRS^3A$  was compared against that of two other schemes: Single-Cell (SC) and Default Simulated Annealing (DSA). In SC, all UEs try to establish a connection with MBS only. DSA, on the other hand, corresponds to the classic SA scheme for the UA-RA problem formulated as MKP without fine-tuning in terms of parameter range and solution search space reduction.

The rest of the paper is organized as follows: Section II surveys existing related research work. Section III presents the general form of MKP, introduces SA, and explains fine-tuning of SA in terms of parameters range and solution search space. Section IV discusses the system model and formulates the problem.  $RS^3A$  and  $PIRS^3A$  algorithmic structure is presented in

Section V. Section VI includes a case study for the joint UA-RA problem and presents simulation settings. Analysis of testing results is done in Section VII. Finally, conclusions are drawn in Section VIII.

## II. RELATED WORK

The problem of UA-RA in the under-laid HetNets has been studied recently. The approaches mainly differ in terms of architecture (i.e., centralized or distributed), the number of parameters considered, and the execution time. For instance, Alnoman *et al.* [5] proposed a joint UA-RA de-centralized approach to maximize the overall network throughput using a Mamdani-type fuzzy logic controller (FLC). The users were first classified based on their data rate requirements, and the controller decides the amount of bandwidth to allocate to each class. The results were compared to greedy-based and best signal-to-interference noise ratio (SINR)-based approaches. The results showed improvements in the data rate, bandwidth usage, and blocking ratio. However, this work did not consider the network load balancing issue, which affects the overall system throughput. Wang *et al.* [6] divided the UA-RA problem into two sub-problems. The first sub-problem was solved using graph theory by fixing the power allocation (PA), UA, and RA, while the second was solved using a convex difference function in which UA-RA was fixed, and PA was solved. However, this scheme did not provide services for user equipments (UEs) with bad channel conditions. Feng *et al.* [7] proposed two schemes for the joint UA-RA problem, one centralized and one distributed. The centralized iterative scheme was broken down into two sub-problems: first, using a cutting plane approach, the UA problem was solved; second, a primary decomposition approach solved the joint frame design and the RA problem. Both sub-problems were iteratively solved to find an optimal solution. The de-centralized scheme used repeated games between users, which has shown to achieve a Nash equilibrium. Overall, the centralized scheme had better system throughput than a distributed scheme. However, the scheme incurred a large overhead, which makes its use unrealistic in large-scale networks. Additionally, the resources between wireless backhaul and small-cell UEs are assigned orthogonally; hence, the spectrum efficiency decreases as user number increases. Using the Stackelberg game method, Zhong *et al.* [8] solved the UA-RA problem while considering the back-haul potential of BSs. Sapountziz *et al.* [9] provided an optimal solution for UA in back-haul restricted HetNets by finding the optimum cell to be associated with. Without an effect on its load, the search for this optimum cell is done iteratively. Luo *et al.* [10] suggested a joint UA-RA scheme to minimize network packet delay and suggested different QoS-aware UA (QoSA) strategies: descent of block-coordinate, multiplier alternating-direction method, and multi-flow. These algorithms minimized the packet delays in a distributed way and have lower complexity than conventional UA strategies.

Barbosa *et al.* [11] proposed the use of DoE (Design of Expert) [12], RSM (Reduced Surface Methodology) and racing algorithms to improve the genetic algorithm (GA) and SA efficiency to solve the problems of classical optimization. We

have used this work as the base for tuning SA's significant hyper-parameters for solving the MKP. RSM<sup>1</sup> is suggested as a fine-tuning technique by the authors of [13] to achieve greater proximity of regions with promising settings. The racing concept was studied in [15], [16] using F-race, a racing algorithm where candidate configurations are removed using Friedman statistics. The work in [17] helps us understand how to use various indicators such as channel quality index (CQI) and modulation coding scheme (MCS) to map RB usage rate with available throughput. The authors of [18] suggested the second phase of Radio Network Planning and Optimization (RNPO), i.e., comprehensive planning in which BS is put on the geographical area of interest and attempts to optimize their position based on the SA. In summary, the above work gives us an idea that SA can be used in a wide range of HetNet applications. The authors of [19] investigate cooperative jamming in a two-tier 5G HetNet and use convex optimization techniques to find feasible solutions to non-convex problems. We have used some of the suggested convex optimization techniques to prove that restrictions on MKP are convex from this work. Some simulation configurations in [20], such as the propagation loss model and fading model, have also been used in our work. Graph theory-based management resource allocation via knapsack in cellular networks with the device to device (D2D) communication underlays is suggested in [21]. In [22] authors use GA to address RA in HetNets, but with many limitations. Some of those limitations are: 1) the problem is formulated without taking into account various parameters in HetNets such as fading, interference, channel state information (CSI) 2) the results of the proposed scheme are compared with a knapsack, but there is no information on the weights and values of items and knapsack capacity.

The authors of this paper have previously introduced a novel quality efficient femtocell offloading scheme (QEFOS) which mitigates the effect of interferences and improves QoS and user quality of experience (QoE) [23] and a preliminary version of an enhanced SA for solving the UA-RA problem as a MKP, which this paper extends [24]. Many works suggest solving UA-RA problems, but none of the proposed work considers the TBS index, user demand, and maximum BS capacity in a single problem. Overall, no work solves the UA-RA while reducing complexity and overhead time as proposed in this work.

In our work, we formulate the dual UA-RA problem as a MKP, which is an NP-complete optimization problem [4]. Many techniques based on dynamic programming, branch-and-bound, greedy, GA, ant colony, particle swarm optimization (PSO) were employed for solving MKP [25]. However, on the one hand, it is challenging to apply exact solution finding methods due to their exponential computational complexity. In contrast, approximation techniques avoid most complexity-related drawbacks and achieve good results, such as in [25]. Belonging to the latter category, SA was chosen to solve MKP. SA is a probabilistic technique used to find a global minimum of an objective function by progressing through

many local minima [11]. Other approximation meta-heuristics algorithms, such as GA and PSO, can be used to solve the MKP. However, they were not used for the following reasons: all meta-heuristics algorithms have strong searching abilities in general. However, some meta-heuristics algorithms, such as SA, are single solution-based algorithms, while others, such as GA and PSO, are population-based algorithms. This means that SA begins with just one solution and attempts to improve it. In contrast, GA and PSO would have multiple possible solutions (i.e., hundreds), depending on the population size. As a result, SA would be easier to deploy and converges faster than GA and PSO when solving a combinatorial optimization problem like MKP. However, since GA has more exploration than SA, better optimal solutions are predicted. However, this comes at the expense of a long execution and convergence time. Due to all these quintessential properties possessed by SA, it was chosen. As in solving a large real-time scale networking problem like UA-RA, an algorithm with reduced complexity, low convergence time, and low computational resource requirement should be given utmost importance.

UA-RA is a recent research topic of high interest for network academics and researchers, and solving it using a meta-heuristic algorithm is not the only solution available. Many other methods, such as Deep Reinforcement Learning (DRL) and Deep Learning (DL), can solve the UA-RA problem. However, they are not used due to some major limitations compared to meta-heuristics like SA, which is the algorithmic choice in this paper. DRL and DL are machine learning (ML) research avenues with a strong reputation for solving a wide range of learning tasks, but they are not easy to train [26]. However, lately the training cost is dropping and increasing number of such solutions are expected to be considered [27]. Therefore, it is recommended to follow closely the progress of this research avenue. On the other hand, SA is a compact and effective technique that offers excellent solutions to single and multi-objective optimization problems while reducing computation time significantly [28]. Meta-heuristic algorithms aim to solve problems faster, solve large problems, and generate robust algorithms. Unlike DL and DRL, they are versatile, easy to design, and simple to implement. Given a low enough temperature and enough perturbations, meta-heuristics algorithms like SA are theoretically guaranteed to find the optimal solution to a problem [29]. On the other hand, DRL lacks the theoretical guarantees of algorithms such as SA, which take a hill-climbing approach and are less prone to policy collapse issues. By setting the greedy criterion only to accept better solutions, SA can achieve monotonically better performance. In contrast, DRL cannot [30].

Recently, several ML or DL-based algorithms have been proposed and applied to HetNets systems, including deep neural network (DNN) [31], long short-term memory (LSTM) [32], convolutional neural network (CNN) [33], *Q-learning* [34], deep *Q-network* (DQN) [35], and deep deterministic policy gradients (DDPG) [36]. However, on one hand, the DL-based models (e.g., DNN, LSTM, and CNN) have outstanding prediction and reasoning capabilities. Still, they require a considerable amount of labelled training data [37]. On the other hand, when the HetNet systems scale grows, DRL-based

<sup>1</sup>The RSM framework is available in DoE software, and a more detailed explanation about its use can be found in [14].

Table I  
COMPARISON AMONG DIFFERENT OPTIMIZATION TECHNIQUES

Techniques		Problem Solving
Meta-heuristic	Single Solution-based	Single Solution approaches focus on modifying and improving a single candidate solution; single solution meta-heuristics include SA, etc. [44].
	Population-based	Population-based approaches maintain and improve multiple candidate solutions, often using population characteristics to guide the search; population based meta-heuristics include GA, PSO, etc. [44].
Reinforcement Learning	–	This is a branch of ML used to help an agent to learn the optimal policy when the agent has no information about the surrounding environment [34], [35], [36].
Deep Learning	–	This is branch of ML used to help an agent to learn the optimal policy when the agent has some information about surrounding environment in advance [31], [32], [37].
Deep Reinforcement Learning	–	This is an advanced model of reinforcement learning technique in which deep learning is utilized as an effective tool to improve learning rate for reinforcement learning algorithms [36], [40].

models (e.g., *Q-learning*, DQN, and DDPG) cannot converge, and the final results are unstable [38]. Slow convergence speed is a problem with *Q-learning*, mainly when the problem state space and action space are large. Additionally, the algorithms must save complete tables of an immediate value for each state-action pair, such as the *Q*-value. These tables may be too big for mobile devices to handle. DRL often causes poor performance in this regard [39].

**DL-based Schemes:** In [31], a distributed DL algorithm was proposed to make an offloading decision for MEC systems, where several DNNs were trained in parallel and the offloading decisions were made cooperatively. In [32], a LSTM network was proposed to predict the traffic of small base stations (SBS), and a cross-entropy loss function was applied to evaluate the LSTM and obtain the offloading strategy. In [37], a distributed deployment strategy for the multilayer convolutional neural network was presented, which included two parts: pre-processing and classification. The pre-processing part was deployed on the edge server for feature extraction and data compression to reduce the edge and cloud systems data transmission. In contrast, the classification part was deployed for pattern classification and recognition. These proposed DL-based methods need prior knowledge and labeled samples, which may be hard to obtain for a dynamic environment.

**Reinforcement Learning (RL)-based Schemes:** In [34], a *Q-learning*-based mobile offloading strategy was proposed for a mobile offloading game. In [35], a DQN-based approach was applied to jointly optimize the networking, caching, and computing resources in the vehicular networks. These proposed RL-based methods may be unstable and hard to converge for large search spaces with large-scale users.

**DRL-based Schemes:** In [36], a DRL-based energy-efficient UAV control method was proposed to design the trajectory of UAV by jointly considering the communications coverage, fairness, energy consumption, and connectivity. In [40], a multi-agent DRL-based method was proposed to solve the joint UA-RA optimization problem. The optimization issue is investigated to obtain the optimal long-term network utility while guaranteeing UE QoS requirements. The optimal solution is obtained by jointly associating UEs to BSs and allocating channels to UEs.

TABLE. I summarizes the different solutions. The complex-

ity of ML-based algorithms is compared in TABLE. II. The complexity of DL, RL, and DRL-based schemes depends on the number of hidden units and layers, number of inputs, and activation function [42]; on the other hand, SA's complexity depends on the cooling schedule [43]. The  $\mathcal{O}$  notation represents the growth rate of an algorithm that is less than or equal to a specific value. We can infer from this in-depth review that our proposed algorithm outperforms other ML, DL, and DRL-based algorithms for real-time wide networking problems like UA-RA in terms of implementation, computational complexity, and producing nearby global optimal solutions with low convergence time. In this context, we propose  $RS^3A$  and  $PIRS^3A$ , two novel algorithms for UA-RA in HetNets that focus on optimal UE selection with reduced complexity and low processing time. In the next sections, we concentrate on the problem formulation and solutions, present simulation-based testing and analyse the results, which demonstrate the benefit of  $RS^3A$  and  $PIRS^3A$  in comparison with alternative solutions.

### III. PROBLEM FORMULATION

#### A. Generalized MKP

The MKP is a classical 0-1 problem of combinatorial optimization that can be extended to different fields. A set of items for  $n^2$  and a set of  $m$  resources are given. The profit  $p_j$  and the resource consumption value  $r_{ij}$  are allocated to each item  $j(j = 1, \dots, n)$  for each resource  $i(i = 1, \dots, m)$ . The problem is to define a subset of all items leading to the highest total profit while not exceeding the upper bound  $b_i$  of the resource. It is possible to formulate the MKP as:

$$\text{maximize } f = \sum_{j=1}^n p_j x_j \quad (1)$$

subject to

$$\sum_{j=1}^n r_{ij} x_j \leq b_i, i \in M = \{1, \dots, m\} \quad (2)$$

$$x_j \in \{0, 1\}, j \in N = \{1, \dots, n\} \quad (3)$$

<sup>2</sup>In this context,  $n$  refers to number of items which need to be placed in the knapsack. However, in latter sections  $n$  refer to number of users.

Table II  
COMPLEXITY OF ML MODELS.(N= NUMBER OF TRAINING EXAMPLES, D= NUMBER OF DIMENSIONS OF THE DATA, K= NUMBER OF NEIGHBORS, C = FEATURE FOR EACH CLASS) [41]

Schemes	Time Complexity	Space Complexity
K Nearest Neighbours	$\mathcal{O}(K * N * D)$	$\mathcal{O}(N * D)$
Logistic Regression	$\mathcal{O}(N * D)$	$\mathcal{O}(D)$
Support Vector Machine (SVM)	$\mathcal{O}(N^2)$	$\mathcal{O}(K * D)$
Decision Tree	$\mathcal{O}(N * \log(N) * D)$	$\mathcal{O}(\text{maximum depth of tree})$
Random Forest	$\mathcal{O}(N * \log(N) * D * K)$	$\mathcal{O}(\text{depth of tree} * K)$
Naive Bayes	$\mathcal{O}(N * D)$	$\mathcal{O}(N * C)$

Variable  $x_j$  is an indicator of item  $j$ . If  $x_j$  is set to 1, item  $j$  is selected, or 0 means item  $j$  is not selected for  $j = 1, \dots, n$ . Eq.(1) represents the total profit of selection items and Eq.(2) the  $m$  resource constraints. A well stated 0/1 MKP assumes that  $p_j > 0$  and we assume that  $r_{ij} \leq b_i \leq \sum_{j=1}^n r_{ij}$  for all  $i = 1, \dots, m$  and  $j = 1, \dots, n$ .

### B. Generalized SA

SA is a local search algorithm that can circumvent the local optima problem. Its ease of implementation and its convergence properties made it an algorithm of choice for solving combinatorial optimization problems like MKP. It was named as such because of its similarity to the physical solid annealing process [11], which involves heating and controlled cooling of material by varying the temperature. If the temperature decreases very slowly, a stable state can be observed, which cannot be reached if the temperature falls quickly [11].

SA tries to evade the local optima by allowing temporal deterioration of actual solutions (i.e., moves to a solution that corresponds to a worse objective function value), where the deterioration is controlled by a parameter temperature  $t$ , which determines the mobility of the system and is reduced by a positive factor  $\rho < 1$  in the algorithm. The likelihood of accepting a deteriorated solution decreases as the algorithm progresses. For a given value of  $t$  some exchange trials  $D$  (repetitions) are performed, until the value  $t$  is less than final temperature  $\delta$ . The initial temperature  $t$  should be initialized with  $t := \alpha * \rho$ , where  $\alpha$  is defined in Algorithm 1.

Some definitions are needed to define the SA algorithm for MKP formally. Let  $\Upsilon$  be the solution space; define  $\Phi(\omega)$  to be the neighborhood function for  $\omega \in \Upsilon$ . SA starts with an initial solution,  $\omega \in \Upsilon$ . A neighboring solution  $\omega' \in \Phi(\omega)$  is then generated randomly in most cases. SA is based on Metropolis acceptance criterion<sup>3</sup>, which models how a thermodynamic system moves from its current solution  $\omega \in \Upsilon$  to a candidate solution  $\omega' \in \Phi(\omega)$ , in which the energy content is being minimized. The candidate solution,  $\omega'$ , is accepted as the current solution based on the acceptance probability

$$P\{\text{Accept } \omega' \text{ as next solution}\}$$

<sup>3</sup>This is based on the Random Walk Metropolis Algorithm (RWMA). RWMA corresponds to a way of doing dependent sampling from posterior space. RWMA is the simplest version of  $I^{st}$  order Markov Chain Monte Carlo (MCMC) algorithms. It is called  $I^{st}$  order Markov Chain because the decision to step next in parameter space only depends on our current position. It does not depend on the history of the positions we visited and Monte Carlo because of pseudo-random numbers.

$$= \begin{cases} \exp[(f(\omega') - f(\omega))/t_a], & \text{if } f(\omega') - f(\omega) < 0, \\ 1 & \text{if } f(\omega') - f(\omega) \geq 0. \end{cases} \quad (4)$$

Define  $t_a$  as temperature parameter ( $t$ ) in iteration  $a$ , such that

$$t_a > 0 \text{ for all } a \text{ and } \lim_{a \rightarrow \infty} t_a = 0 \quad (5)$$

### C. Reduced Search Space SA ( $RS^3A$ ) Phase

A is an algorithm applied to various problems (P) with parameters (V). The issue of fine-tuning an algorithm can therefore be summed up as a search space:

$$A = \{\alpha_c \leftarrow 1 \dots \eta_\alpha \ X \ \beta_c \leftarrow 1 \dots \eta_\beta \ X \ \dots \ X \ \zeta_c \leftarrow 1 \dots \eta_\zeta\}, c = 1, \dots, z, \quad (6)$$

where  $\alpha, \beta, \dots, \zeta$  are parameters of algorithm A for a given problem P and  $1 \dots \eta_\alpha, 1 \dots \eta_\beta, 1 \dots \eta_\zeta$  are the finite ranges of values assumed for each parameter. The number of parameters as well as their ranges can vary extensively according to A and P studied, such that  $\eta_\alpha \times \eta_\beta \times \dots \times \eta_\zeta$  should be possibly a number of combinations tests of A on P [11].

Our fine-tune algorithm approach can be expressed as a process that starts with an arbitrary set of instances from a class of optimization problems and follows the ranges from the algorithm for each parameter, applying  $2^k$  full factorial architecture to research the response (effect) of multiple parameters (factors). A relationship of cause and effect between factors and responses can be developed using complete factorial designs, the usual representation of which is an empirical regression model (linear or quadratic) for the mechanism under study, as follows:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_{12} X_1 X_2 + \epsilon \quad (7)$$

where Y is the response,  $\beta_0, \beta_1, \beta_2, \beta_{12}$  are coefficients,  $X_1$  and  $X_2$  are factors, and  $\epsilon$  is the experimental error.

Factorial designs help define the variables that affect the response. However, the interest is to identify the factors that can maximize the process and generate values (of factors) closer to the optimum, Eq.(7), the relationship between factors and reaction may not be sufficient. We use RSM to achieve greater proximity to an area with promising settings. The RSM is a system of statistical and mathematical techniques that employ factorial designs, regression analysis, and methods of optimization in situations where many input parameters influence the output of a process [13]. The outcome of RSM is a model of the second-order, given by

$$Y = \beta_0 + \beta_1 X_1 + \beta_{11} X_1^2 + \beta_2 X_2 + \beta_{22} X_2^2 + \beta_{12} X_1 X_2 + \epsilon \quad (8)$$

The application of a racing algorithm to describe the algorithms setup is our approach’s final stage. The *irace* package [44], is an iterated racing implementation. Iterated racing is an automated configuration process consisting of three steps: (1) sampling new configurations by specific distribution, (2) choosing the best configuration using racing from the newly sampled ones, and (3) updating the sampling distribution to bias the sampling to the best configurations. Until a termination condition is met, these three steps are repeated. In iterated racing as implemented in the *irace* package, a sampling distribution independent of the other parameters has been associated with a configurable parameter, apart from constraints and conditions between parameters. A truncated normal distribution for numerical parameters or a discrete distribution for categorical parameters is the sampling distribution. Ordinal parameters are used as numerical parameters (integers). In a normal distribution, the distribution update consists of changing the mean and the standard deviation, or the discrete probability value of the discrete distributions. The update biases the distributions to increase the probability of sampling the parameter value in the best configurations discovered so far in future iterations.

The truncated normal distribution is a significant option in the world of probability and statistics and its use is natural when a normal distribution is employed. For example, when one wants to threshold or screen values from a normally distributed dataset, the remaining data has a truncated normal distribution. There is substantial motivation to study the truncated normal distribution from a statistical perspective [46]. It was also shown in [47] that the truncated normal distribution estimators generally have a smaller mean square error than the classical non-truncated normal distribution. This property possessed by truncated normal distribution motivated us to select it. We employed a tool, i.e., *irace* package, which uses sampling based on the truncated normal distribution. The next paragraph explains how and why the truncated normal distribution is used.

The samples are drawn first from a uniform distribution. However, in subsequent iterations, they are drawn from a normal distribution based on the values of the parameters in the elite configurations (this is only for numerical parameters) so that new samples are more likely to be similar to the best values found thus far. Since parameter values are bounded to a fixed range, the truncated normal distribution is favored over the classical non-truncated distribution. The conventional non-truncated normal distribution does not accommodate out-of-bounds samples. The out-of-bounds samples can be treated by replacing them with the nearest bound values, which would result in boundary values having a very high probability, possibly greater than the distribution’s mean. With the truncated normal distribution, if the mean is close to the boundary, values near the boundary are sampled more frequently, and if the mean is far from the boundary, values near the boundary are sampled less frequently.

The best configurations are chosen using racing after new configurations are sampled. With a finite set of candidate configurations, a race begins. The Fig. 2 illustration includes ten  $C_i$  configurations. The candidate configurations are evaluated

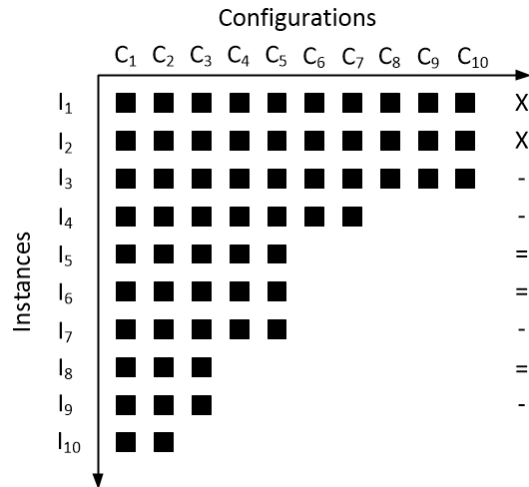


Figure 2. Racing for configuring automated algorithms [44]. Each node is an assessment of a single instance configuration. ‘x’ implies that there is no statistical test, ‘-’ implies that at least one configuration has been discarded and ‘=’ no configuration has been discarded by the test.

for a single instance at each point of the race ( $I_j$ ). Those candidate configurations that perform statistically worse than at least another one are discarded after several steps and the competition continues with the remaining configurations that survive. Since the first elimination test is critical, usually a higher number of instances are seen before conducting the first statistical test. Subsequent statistical analyses are conducted more often (by default for every instance). This procedure continues until reaching a minimum number of surviving configurations or a maximum number of instances that have been used.

No study has used this trio (DoE, RSM and *irace*) to solve MKP using SA. In Python 3 and R<sup>4</sup>, all codes are built from scratch, as no such codes are directly available for the use of the *irace* package to solve MKP using SA.

#### D. Performance Improvement

An important issue with SA is that there are too many ineffective solutions which are explored. The reason is that the SA algorithm is essentially a prescription for a partial walk in the configuration or problem space. At the end of each iteration, a random step is produced. The associated evaluation point’s target function is evaluated to determine whether or not to accept that step. As the problem search space contains many ineffective solutions, which a typical SA algorithm considers when evaluating the target function, wasting time and effort, before applying SA to solve MKP, *PIRS*<sup>3</sup>A removes the ineffective solutions to improve its performance.

To remove these ineffective solutions, the MKP items were arranged in decreasing order of their profits, i.e., the higher the profit, the lower the index. If the number of items  $n$  is greater

<sup>4</sup>A target runner must be given to execute the target algorithm with a particular configuration and instance parameter. The target runner function (or program) serves as an interface between the target algorithm execution and *irace*: it receives the instance and configuration as arguments and must return the evaluation of the target algorithm’s execution.

than  $\left\lfloor \frac{b}{n} \right\rfloor$ , we only take the front  $\left\lfloor \frac{b}{n} \right\rfloor$  items to participate in SA selection process. The reason is that the last  $n - \left\lfloor \frac{b}{n} \right\rfloor$  items are never selected as the target is to obtain the maximum profit in the knapsack. Therefore, these remaining  $n - \left\lfloor \frac{b}{n} \right\rfloor$  items are associated to ineffective solutions and are removed from the search space.

#### IV. SYSTEM MODEL

As shown in Fig. 1, consider the downlink of a HetNet consisting of fixed BSs and randomly placed UEs. The area shown in Fig. 1 is covered by two-tiers of BSs: MBS ( $M_m$ ) and FBS tier ( $M_f$ ). In the coverage area of such BSs, overlaps will occur, and each UE will be within the range of at least one BS. The set of all BSs is denoted as  $BS = \{MBS_1, \dots, MBS_{M_m}, FBS_1, \dots, FBS_{M_f}\}$ , with the set of the BSs indices  $M = \{0, 1, \dots, m-1\}$ , where  $m = M_m + M_f$ . Let  $S$  be the set of sub-channels available that can be used by each BS  $i \in M$ . These sub-channels are further divided and assigned to the UEs linked to each BS  $i$ . Each BS  $i \in M$  is assumed to transmit power  $O_{ij}^l$  on sub-channel  $l$  with a constant per sub-channel, and the total transmit power of BS  $i$  is  $\bar{P}_i = \sum_{l \in S} O_{ij}^l$ . A high-speed back-haul with minimal delay (such as high-speed fiber) is connected to all BSs. Let  $N$  be the set of UEs located inside the region  $G$  and  $\psi_j \in \Psi$  be the requested downlink rate (bits per second) of UE  $j$ , where  $\Psi$  is the discrete set of service classes. In this paper, we are interested in the video<sup>5</sup> services as it require high bandwidth. Each UE can only be associated with at most one BS at any time instance and we define  $\tilde{\mu}$  as the total path loss (*i.e.*, follows a log-distance path loss model) between BS  $i$  and UE  $j$  in decibels (dB). Other notations are given in Table. III.

The UEs might not receive high data rates because of time-variable fading channels in wireless communication (*i.e.*, LTE, 5G). Fueled by urbanization, this process is more extreme in urban areas. Another reason for low UE data rate can be congestion at BS, caused by the existence of many users. In today's wireless communication system, a decentralized scheme is typically used to solve such problems, enabling UEs to communicate with BSs that provide them with the best channel conditions and satisfy minimum QoS requirements.

##### A. Problem Formulation

We aim to build a decentralized scheme under maximum BS capacity and TBS index constraints to solve the UA-RA problem in HetNets. This kind of UA-RA problem has to be done in a self-organized way. Next, the potential objective function is considered, and the UA-RA problem is formulated as a MKP optimization problem.

<sup>5</sup>We model and simulate video streaming in NS-3, and we measure the network transport performance related to the video streaming service in terms of QoS metrics. QoS strongly influences viewer QoE, but no estimation of QoE is provided.

Table III  
TABLE OF NOTATIONS

Notation	Description
$N,  N , j$	set, cardinality, and indexes of UEs
$M,  M , i$	set, cardinality and indexes of BSs
$S,  S , l$	set, cardinality and index of sub-channels
$\tilde{\mu}$	path loss between BS $i$ and UE $j$
$O_{ij}^l, \bar{P}_i$	per sub-channel and total tx power of BS $i$
$\psi_j$	requested downlink rates of UE $j$
$b_i$	maximum capacity of BS or knapsack $i$
$r_{ij}$	resource consumption allocated to each item or UE $j$ for each BS or knapsack $i$
$x_j^i, x$	user association decision variable/vector
$p_j$	profit or available throughput for UE $j$
$Q_i$	vector of UEs having TBS index greater than $\Gamma$ with BS $i$

1) **Objective Function:** There are two kinds of entities in our system with different viewpoints and priorities: UEs and BSs. On the one side, each UE, given its QoS requirement, wants to achieve the maximum data rate. The BSs, on the other hand, want to satisfy the UEs QoS requirement within their limits. Hence, we define the objective function as the sum of available throughput (*i.e.*, estimated based on RB usage rate) under the BSs maximum capacity and TBS index constraint while considering the entity's perspective and objectives.

$$f(x) = \sum_{j=1}^n p_j x_j \quad (9)$$

subject to

$$\sum_{j=1}^n r_{ij} x_j \leq b_i, i \in M = \{1, \dots, m\} \quad (10)$$

$$x_j \in \{0, 1\}, j \in N = \{1, \dots, n\} \quad (11)$$

$$q_{ij} \geq \Gamma, i \in M = \{1, 2, \dots, m\}, j \in N = \{1, 2, \dots, n\} \quad (12)$$

$$r_{ij} \leq b_i, i \in M = \{1, \dots, m\}, j \in N = \{1, \dots, n\} \quad (13)$$

where Eq.(9) represents a *hyper-plane* hence it is a convex function with convex constraints. Eq.(10), Eq.(12) and Eq.(13) represents *half-spaces* and intersection of *half-spaces* represent *POLYHEDRA* (DEFINITION IV-A.1), which is also *convex*. In general, the UA-RA problem that is solved by formulating it as MKP has a linear objective function with linear constraints except for Eq.(11), an integer constraint, or a binary constraint as it takes either 0 or 1 value only. Hence, MKP is a *mixed-integer linear program* (MILP)<sup>6</sup> problem, solved by *PIRS<sup>3</sup>A* to achieve a near-optimal solution and enhance the QoS perceived by users.

**Lemma 1.** Generalize notion of *HYPER-PLANE* is:

$$\tilde{p}^T * \tilde{x} = Z_{max},$$

$$\tilde{p} = [p_1, p_2, \dots, p_n], \tilde{x} = [x_1, x_2, \dots, x_n].$$

$$\tilde{p}^T * \tilde{x} = \left( \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \end{bmatrix} * [x_1, x_2, \dots, x_n] \right)$$

$$p_1 * x_1 + p_2 * x_2 \dots + p_n * x_n = Z_{max}$$

*Proof* : See Appendix A

<sup>6</sup>MILP problem *i.e.* mixed-integer linear program is a simplest class of convex optimization problem. MILP can also be called a binary program (BP) because Eq.(11) can take only binary values, not all integer values.

**Definition IV-A.1.** *POLYHEDRON* is formed from the finite intersection of *HYPER-PLANES* and *HALFSPACES*.

Eq.(9) represents a *HYPER-PLANE* and Eq.(10), Eq.(12) and Eq.(13) represents *HALF-SPACES* (Fig. 14). Hence a vector  $\tilde{x}$  that satisfy all constraints and maximize Eq.(9), will form a *POLYHEDRON* and is convex.

2) **Optimization Problem:** Under QoS provisioning, we describe the Optimization Problem for the UA-RA with MBSs and FBSs as follows:

$$\begin{aligned} OP : \text{minimize} : & -f(x) \\ \text{subject to} : & (9), (10), (11), (12) \end{aligned} \quad (14)$$

Constraint Eq.(10) refers to the fact that the allocation of resources to UEs does not surpass the *maximum BS capacity* ( $b_i$ ). Constraint Eq.(11) shows the *unique association* property, where UE  $j$  can only be associated with one BS at any moment. Due to the unique association, the number of possible downlinks (UE associations) is reduced from  $2^{|N| |M|}$  to  $|M|^{|N|}$  [20]. Constraint Eq.(12) implies that each UE  $j$  should have a TBS index above a certain threshold  $\Gamma$  to participate in the UA-RA problem (i.e., *TBS index vector*). Constraint Eq.(13) indicates that each BS can serve at least one UE. OP belongs to a class of assignment problems which are proven to be combinatorial and NP-hard [4], [20], [25].

3) **PIRS<sup>3</sup>A - A Decentralized Scheme:** Our objective is to design a decentralized scheme that helps each UE  $j$  to associate with that BS  $i$  that offers the highest throughput (i.e., high CQI, better channel conditions). We deploy *PIRS<sup>3</sup>A* in an Information Service Server (ISS) near MBS, which solves UA-RA problem episodically as illustrated in Algorithm 1.

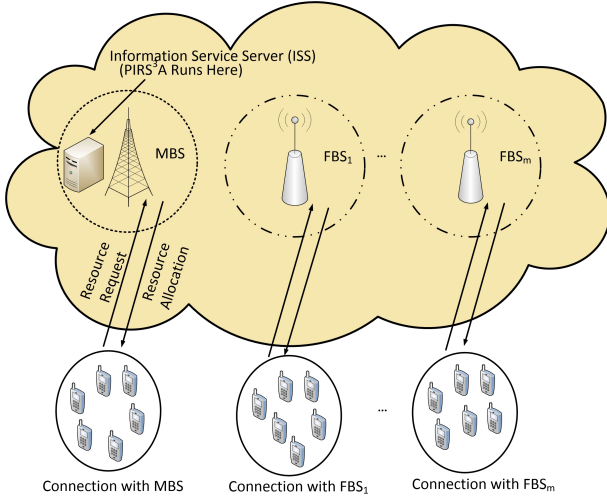


Figure 3. System model of a two tier macro-femto HetNet integrated with *PIRS<sup>3</sup>A*

We assume that MBSs are chosen first, followed by FBSs, and that UEs and BSs can communicate through control signals (CS). In each episode  $e$ , every UE  $j$  estimates the throughput they can achieve with BS  $i$  as described in Section IV.B. UE  $j$  informs ISS about this estimated throughput  $p_j$  and requested service  $r_{ij}$  demands<sup>7</sup>. Based on Eq.(12),

<sup>7</sup>In this paper we assume that UEs demands for video services only.

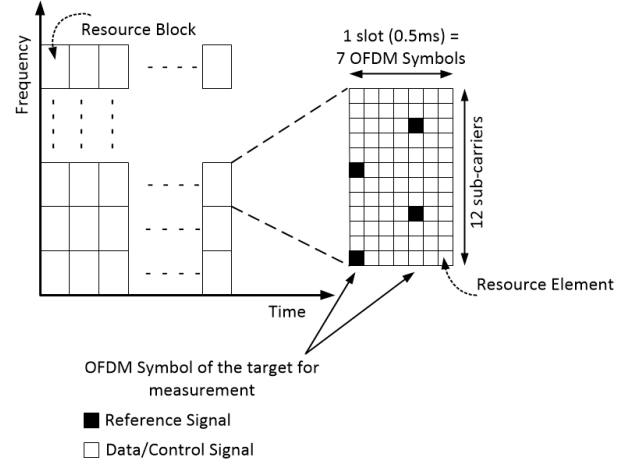


Figure 4. RB structure of LTE downlink [12].

vector  $Q_i$  ( $i \in M$ ) is formed. Using CS, BS  $i$  inform ISS about their maximum capacity ( $b_i$ )<sup>8</sup>. On receiving all required information, *PIRS<sup>3</sup>A* runs and those UEs part of an optimal solution will be associated with BS  $i$  and are flagged as "accepted" and BS  $i$  allocates resources (sub-channels) to all the "accepted" UEs (Fig. 3).

#### B. Available Throughput Estimation Method

In this subsection, we clarify our proposed method of estimating the available throughput based on metrics that can be acquired by a UE. By calculating the available throughput, the UE will determine how much data it can obtain from a base station [17]. This available throughput corresponds to  $p_j$  in OP. Available throughput means how much data transfer throughput a UE can use. It can be determined by measuring CQI and mapping it to get MCS and TBS index. The amount of data each RB (Fig. 4) may carry depends on the BSs modulation method, and this method is chosen based on the quality of the signal between the BS and the UE. The conventional method for estimating the throughput is shown in Fig. 5. The BS sends CS signal to UE. From the CS signal, UE calculates the CQI<sup>9</sup>, and UE sends back the feedback of the CSI, which includes the CQI to BS [48].

When BS receives CQI, the MCS for the downlink to the UE is chosen. The LTE framework specifies the CQI guidelines for choosing a modulation form. Using the MCS, the BS modulates data on each RB in the modulation mode. The amount of data that BS transmits to the UE depends on the MCS, and so the number of RBs allocated to the UE. The MCS and TBS indices mapping table [49] are used to get the TBS index. This TBS index and the number of RB allocated, calculated based on the method presented in [50], [51], are employed to estimate the available throughput for each UE  $j$  to each BS  $i$ .

<sup>8</sup>The maximum capacity is calculated based on the fact that for each RB we have  $12(\text{sub-carriers}) * 7(\text{OFDM symbols}) * 6(\text{bits}) = 504$  bits in the one-time slot (0.5 msec); for 100 RBs we have 100.8 Mbps for MBS and 50.4 Mbps for FBSs (Fig. 4).

<sup>9</sup>CQI is represented as a 4-bit discrete value; CQI = 1 and CQI = 15 reflect the worst, and the best signal quality, respectively.



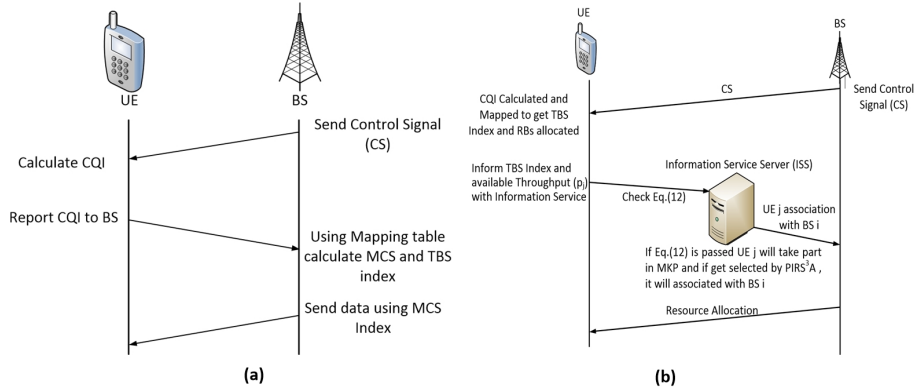


Figure 5. (a) Conventional way of determining MCS. (b) Sequence of determining MCS integrated with ISS

Fig. 5(a) has been updated for MCS determination and its shown in Fig. 5(b). In Fig. 5(b), firstly, BS  $i$  sends a CS to each UE  $j$  who wants to associate with it. On the basis of received CS each UE  $j$  calculates CQI and using CQI, MCS and TBS mapping indices table [49]  $j^{th}$  each UE calculates TBS index and RBs allocated and estimates the available throughput which UE  $j$  can receive from BS  $i$ . UE sends this information of the TBS index and available throughput to the ISS where our proposed scheme runs. If the TBS index information by UE  $j$  passed Eq.(12), it will participate in the MKP problem. If it gets selected after solving MKP using  $PIRS^3A$ , it will get an association with BS  $i$ , and resources will get allocated.

## V. ALGORITHM STRUCTURE

$PIRS^3A$  is solved in episodes, as shown in Algorithm 1. In each episode  $e \in E$ , BS  $i \in M$  inform its maximum capacity  $b_i$  (in Mbps), which represents the Knapsack capacity (the BS that participates in episode  $e$  will not participate in the next episodes) to ISS. Next, each UE  $j \in N$  will provide the TBS index (obtained by mapping) and estimated available throughput  $p_j$  and demands  $r_{ij}$  to ISS. Those UEs that satisfy Eq.(12) will be selected in the problem. Vector  $P$  (estimated available throughput (Mbps)),  $R$  (users demands (Mbps)), and vector  $Q$  (UEs satisfying Eq.(12)) are generated. Those UEs which do not satisfy Eq.(12) form a vector  $\tilde{Q}$ . The size of vectors  $Q$ ,  $\tilde{Q}$ ,  $P$ ,  $R$  vary, depending on which UEs satisfy Eq.(12).  $U_p$  denotes a set of participating UEs.  $X := [x_1, x_2, \dots, x_{\tilde{n}}]$  denotes a matrix of variable  $x_j^i$  ( $j \in U_p$ ).

Each episode  $e$  has two phases: *diversification (exploration)* and *intensification (exploitation)*. In *exploration*, only those UE  $j$  which qualify based on Eq. (12) are selected; the remaining ones are considered in the next episode  $e + 1$ . In *exploitation*, selected UEs to take part in the problem and on solving Eq.(9) subject to Eq.(10), Eq.(11) selected UEs will be associated with BS  $i$ , and resources allocated to them. After each episode, set  $N$  is updated, and only those UEs which do not meet Eq.(12) and those not part of the obtained near-optimal solution  $\omega_r$  are left over. The  $PIRS^3A$  will stop when  $N$  becomes an empty set; thus, UEs no longer participate in the *exploration*. The  $PIRS^3A$  convergence. Meta-heuristic design

Table IV  
PARAMETER SETTINGS FOR SA.

SA parameter	$\rho$	$\varrho$	$\delta$	$D$
Low	0.1	0.01	0.00001	0.1n
High	0.9	0.99	0.001	2n

leads to the concept of *probability convergence*. We give the following definitions to explain the concept.

**Definition V-1. (Normalized Performance Gap).** Let  $f(e)$  represent the utility accomplished at the episode  $e$  and  $\kappa_0$  signifies the exactness level. Then, we characterize the normalized performance gap between  $f(e)$  and  $f(e^*)$  as follows:

$$\kappa(e) = \frac{|f(e^*) - f(e)|}{f(e^*)} \quad (15)$$

**Definition V-2. (Convergence in Probability).**  $f(e)$  converges in probability to  $f(e^*)$  as  $t_a \rightarrow 0$ , if and only if  $\lim_{t_a \rightarrow 0} \Pr(\kappa(e) \leq \kappa_0) = 1$ .

## VI. CASE STUDY

### A. Fine Tuning

We have selected the following set of parameters which affect the most MKP solving efficiency using meta-heuristics SA [52]: initial temperature control parameter ( $\rho$ ), control parameter ( $\varrho$ ), final temperature ( $\delta$ ), and the number of iterations during one temperature range ( $D$ ).

We use the relative deviation from the optimum, provided by our results, to generalize the results and compare them to each other, as indicated in Definition V-1.1.

$$\kappa(e) = \frac{|f(e^*) - f(e)|}{f(e^*)} \quad (16)$$

where the computed solution is  $f(e)$  and  $f(e^*)$  is the best-known solution to the problem. Thus, the lower the value of  $\kappa(e)$  for the meta-heuristic, the better the performance of the algorithms [11]. In the first step of our approach, the parameters and their necessary corresponding levels (low and high) required by a  $2^g$  complete factorial are in TABLE. IV.

**Algorithm 1** PIRS<sup>3</sup>A for solving the UA-RA problem**Goals:** Increase  $\rightarrow$  ThroughputDecrease  $\rightarrow$  Packet Loss Ratio, Delay, Jitter**foreach** episode  $e \in E$  **do**Each BS  $i, i \in M$  informs ISS about its maximum capacity  $b_i$ **foreach**  $j \in N$  **do**Inform ISS about TBS Index  $q_{ij}$ **if**  $q_{ij}$  satisfies Eq.(12) **then**|  $q_{ij} \in Q$ **else**| UE  $j$  will not participate in the problem ( $q_{ij} \in \tilde{Q}$ )Inform ISS about estimated available throughput  $p_j$ Inform ISS about Demand  $r_{ij}$ **initialization:**  $P := [p_1, p_2, \dots, p_{\tilde{n}}]$  where  $\tilde{n} \neq n, \tilde{n} = \text{length}(Q)$  $R := [r_{i1}, r_{i2}, \dots, r_{i\tilde{n}}]$  where  $\tilde{n} \neq n, \tilde{n} = \text{length}(Q)$  $U_p := \{U_1, U_2, \dots, U_{\tilde{n}}\}, \forall j \in U_p.$  $X := 0.$ 

Arrange P in decreasing order and perform performance-improvement as explained in Section III.D.

Set parameters values from TABLE. VI i.e. an initial temp. =  $(\max\{p_j | \forall j\} - \min\{p_j | \forall j\} * \rho)$ ;  $\varrho$ ;  $\delta$ ; DSelect Initial Solution  $\omega = (x_1, x_2, \dots, x_{\tilde{n}}) \in \Upsilon$ ;Incumbent Solution  $\leftarrow f(\omega)$ ;**repeat**Set repetition counter  $a = 0$ ;**repeat**Select an integer  $\tilde{i}$  randomly from the set  $\{1, 2, \dots, \tilde{n}\}$ ;**if**  $x_{\tilde{i}} = 0$ , pick item  $\tilde{i}$ , i.e. set  $x_{\tilde{i}} = 1$ , get new solution  $\omega'$ , **then****while** solution  $\omega'$  is unusable, **do**| Drop another item from  $\omega'$  randomly; denote new solution as  $\omega'$ ;Let  $\Delta = f(\omega') - f(\omega)$ ; **while**  $\Delta \geq 0$  or  $\text{Random}(0,1) < \exp^{-\frac{\Delta}{t_a}}$ , **do**|  $\omega \leftarrow \omega'$ ;**else**Drop item  $\tilde{i}$ , and pick up another item randomly, get new solution  $\omega'$ ;Let  $\Delta = f(\omega') - f(\omega)$ ;**while**  $\Delta \geq 0$  or  $\text{Random}(0,1) < \exp^{-\frac{\Delta}{t_a}}$ , **do**|  $\omega \leftarrow \omega'$ ;**if**  $f(\omega') > \text{incumbent solution}$  **then**| Incumbent solution  $\leftarrow f(\omega')$ ; $a = a+1$ ;**until**  $a = D$ ;Set  $t = \varrho * t$ ;**until**  $t > \delta$ ; $N = N - \omega'$ ; $e = e+1$ ;

The fine-tuning of the meta-heuristics SA on MKP uses four arbitrary instances available on Github<sup>10</sup> with well designed and developed PYTHON 3 code for solving MKP using SA.

The  $2^{g=4}$  full factorial architecture is used here [11] to identify the critical factors and response effect. We used a DoE ANOVA study to identify these factors. As a consequence, in the first step of our method, we can assume the following for

the SA algorithm: (i) Out of all four factors studied, two factors are significant regardless of the instance selected i.e., Initial temperature control parameter ( $\rho$ ), control parameter ( $\varrho$ ).

(ii) Final temperature ( $\delta$ ) and number of iterations for one temperature (D) are significant in different instances.

(iii) Depending on the instance studied, there are variations between the interactions of the variables.

The next stage consists of applying the RSM to explore the neighborhood regions around a promising area and, according

<sup>10</sup><https://github.com/bharat1992-bit>

Table V  
RSM RESULTS FOR SA

SA parameters	Instance 1	Instance 2	Instance 3	Instance 4
$\rho$	0.628	0.7	0.7	0.7
$\varrho$	0.745	0.255	0.745	0.255
$\delta$	0.00041	0.00051	0.00051	0.00051
$D$	61	42	61	61

Table VI  
DEFAULT AND SUGGESTED PARAMETER SETTINGS FOR SA.

SA parameters	$\rho$	$\varrho$	$\delta$	$D$
Default	0.5	0.6	0.001	60
Suggested	0.8	0.71	0.000595	40

to the studied example, to obtain values for each parameter. RSM is a mathematical framework with statistical techniques used in problem modeling and optimization, where several factors affect the system response. In this context, RSM employs a sampling technique that finds the best match for each studied parameter to obtain a sub-optimal value that corresponds to estimated throughput Eq.(9) with low convergence time. From the RSM results presented in TABLE V, we define a range of values between the minimum and maximum of each parameter. It forms a space for the search of candidate configurations. The procedure employed consists of a simultaneous variation of all four parameters until ANOVA shows statistical significance. RSM results suggest an empirical model for the four parameters.

Following RSM, the next stage employs the *irace* package to implement the racing algorithms. There is no code in the *irace* package for solving MKP using SA. However, there are various examples of how to solve other optimization problems, such as TSP, solved by SA. The racing algorithms' efficiency and RSM's power were used.

We have used the *irace* package for racing algorithm implementation to select as good as possible configuration out of many options in our last step. For this study, the settings used for SA are  $\rho \in \{0.62, 0.7\}$ ,  $\varrho \in \{0.255, 0.5, 0.745\}$ ,  $\delta \in \{0.00041, 0.00051\}$ ,  $D \in \{39, 52, 65\}$ . The values of the  $\rho$ ,  $\varrho$ ,  $\delta$ ,  $D$  are taken from TABLE V. Because of the large difference between the minimum and maximum values obtained from the results of RSM as shown in TABLE V, three different values, including the middle point, for  $\varrho$ ,  $D$ , are considered. Every possible combination leads to a different setting of the algorithm (explained in Section III.C), so that our search space was composed of 36 different SA algorithm parameter settings. We reached the best setting (TABLE VI<sup>11</sup>) for each algorithm for solving MKP after applying the racing algorithm in the search space. In terms of parameter range, the trio lets us achieve  $RS^3A$ , i.e., Reduced Search Space SA.

### B. UA-RA problem simulation settings

In NS3, we perform comprehensive simulations to evaluate our proposed algorithm. We use the optimal  $f(e)$  solution for the benchmark, which is computed using the planned and built

<sup>11</sup>The default corresponded to settings of SA used in [11] and suggested corresponds to settings of SA obtained from fine-tuning.

Table VII  
DEFAULT SIMULATION PARAMETERS

Parameter	Value
Area of Region (G)	500m x 500m
Static UE population ( $ N $ )	[70, 85, 100]
UE traffic demand ( $\psi_j$ )	2 Mbps
# of BSs ( $ M  =  M_m \cup M_f $ )	11 = 1 + 10
Total transmit power of BSs	{46, 26} dBm
# of sub-channels ( $ S $ ) for MBSs	12 x 100
# of sub-channels ( $ S $ ) for FBSs	12 x 50
Bandwidth of each sub channel	15 Khz
BSs DL frequency	2120 Mhz
BSs UL frequency	1930 Mhz
Mobility Model	Constant Position Mobility Model
LTE Mode	FDD

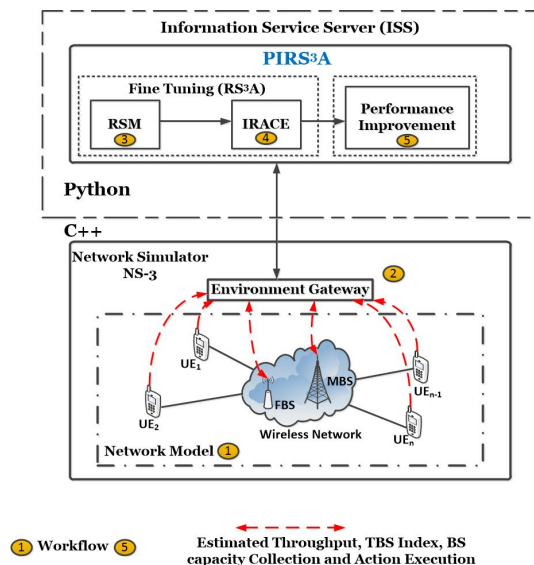


Figure 6. Various elements of the whole data simulation, optimization, and network process.

$PIRS^3A$  in PYTHON 3. In TABLE VII, the major simulation parameters are given. Fig. 6 shows the link and the chain that connects the various elements of the whole data simulation, optimization, and network process.

Firstly, for all our experiments, we assume the BSs to be deployed at fixed locations as shown in Fig. 7. Second, we

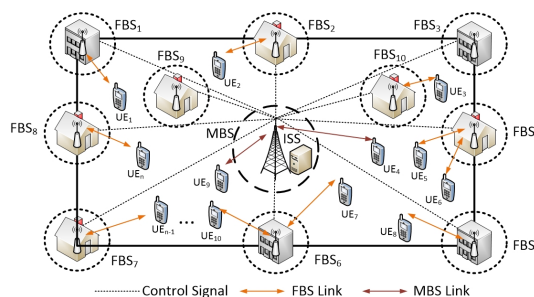


Figure 7. An example of a two-tier HetNet including one MBS:  $M_1$ , and 10 FBSs:  $M_2, M_3, \dots, M_{11}$ . The FBSs are usually located at commercial and residential buildings that constitute hotspots for wireless traffic. The UEs  $UE_1, UE_2, \dots, UE_n$  in a region  $G$  are either served by either MBS or FBSs selected by ISS.

randomly deploy UEs ( $|N| = [70, 85, 100]$ ) following a homogeneous Poisson point process (PPP) for different experiments. Third, we consider a discrete user demand (i.e., requested data rate). In this network, we consider a log distance path loss model as explained and used in [20]. We examine various QoS metrics of those UEs who qualify all constraints and are part of the near-optimal solution in each episode  $e \in E$ . We evaluate 1) throughput (Mbps), 2) packet loss ratio (PLR) (%), 3) Jitter (ms), 4) Delay (ms). We compared our proposed schemes with the other two schemes, i.e., SC and DSA.

## VII. ANALYSIS OF RESULTS

### A. Results after parameters and search space reduction

All results provided in this subsection were computed by Eq.(16) and obtained before the fine-tuning process, i.e., default SA (DSA), after fine-tuning in terms of the range of parameters, i.e.,  $RS^3A$ , and finally after further fine-tuning in terms of solution search space reduction, i.e.,  $PIRS^3A$ , for comparison purposes.

The first set of results (TABLE VIII) corresponds to ten runs of the meta-heuristics (DSA,  $RS^3A$ ,  $PIRS^3A$ ) in ten instances<sup>12</sup> of the benchmark of MKP. In TABLE. VIII, column AM is the arithmetic mean of Eq.(16) in ten runs of each instance; column  $\sigma$  represents the standard deviation in ten runs of each instance; Nopt is the number of times that algorithms reach the optimum for each instance.

Statistics showed in TABLE VIII indicate output improvements both in  $RS^3A$  and ( $PIRS^3A$ ). It should be noted that  $PIRS^3A$  for all instances is more promising.

Fig. 8 illustrates the arithmetic mean of the ten runs of meta-heuristics (DSA,  $RS^3A$ ,  $PIRS^3A$ , DGA) on ten different instances of the MKP. DSA corresponds to the default simulated annealing without any fine-tuning of search space of the parameter ranges and the solution space.  $RS^3A$  corresponds to SA with fine-tuning of search space of the parameter ranges. DGA corresponds to classical GA<sup>13</sup>. We observe that  $PIRS^3A$  is the closest to the optimal solution than DSA,  $RS^3A$ , and DGA.

In contrast, DGA produces a better optimal results compared to DSA and  $RS^3A$  due to more exploration than these SA solutions but lower optimal values compared to  $PIRS^3A$ . It can also be noted from TABLE. VIII column of Nopt that there is a greater likelihood (*probability convergence*) of obtaining optimal solutions under  $PIRS^3A$  compared to DSA and  $RS^3A$ . Indeed, DSA is a single solution-based algorithm with less exploration and high exploitation, because of which it can easily be stuck in local minima. Parameter search space reduction helps find the best starting solution, while solution search space reduction helps remove the useless regions, saving significant time in finding the global optima. By combining these two aspects,  $PIRS^3A$  achieves the global optima with low convergence time.

<sup>12</sup>In this work, Instances or Inst. refer to a problem set having a knapsack with individual capacity and items with weights and values, which are solved using a different version of SA. All the Instances selected have 40 items.

<sup>13</sup>The hyperparameters considered for evaluating MKP using DGA are [11] : number of generations = 203, crossover probability = 0.54, probability of mutation = 0.79, and population size = 110.

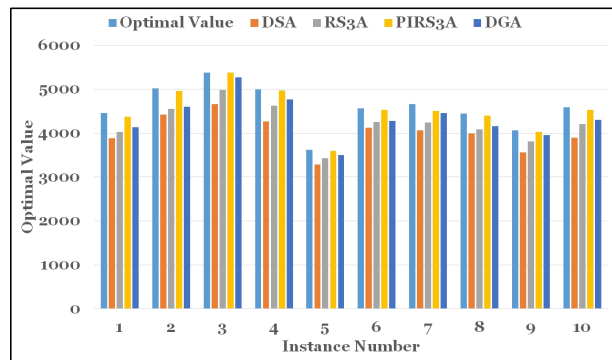


Figure 8. Comparison of optimal values for three different versions of SA.

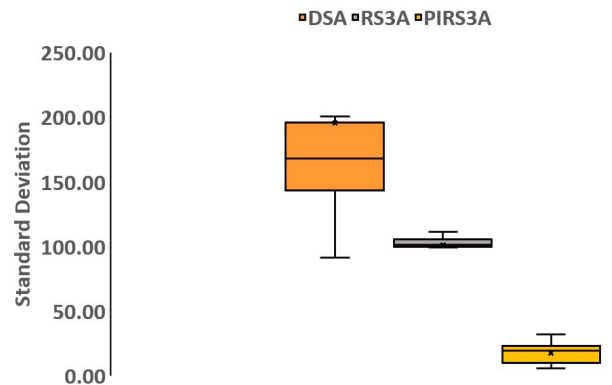


Figure 9. Variability between three different versions of SA.

Results from the  $\sigma$  column of TABLE VIII also suggest that our proposed modification in DSA makes the process more stable. Fig. 9 depicts the distribution of near-optimal value ( $f(e)$ ) per instance. We notice that, due to fine-tuning in both parameters range and solution space,  $PIRS^3A$  offers the solution close to the optimal solution than two other schemes. Through the results (TABLE VIII) and supported by the graphical analysis (Fig. 9), we can highlight that our proposed solution  $PIRS^3A$  produces better results.

Table. IX, Fig. 10(a), and Fig. 10(b) show the execution time expressed in ms and sec for DSA,  $RS^3A$ ,  $PIRS^3A$ , and DGA when varying the number of items<sup>14</sup>.  $PIRS^3A$  has an average execution time 60.71%, 15%, and 99.9% shorter than the times of DSA,  $RS^3A$ , and DGA for 50 items and around 69% ,21%, and 99.9% for 2000 items, respectively. DGA's high exploration time resulted in a long overhead time, making it unsuitable for solving real-time comprehensive network problems. This confirms that our proposed scheme has reduced complexity, while incurring a low overhead time.

### B. QoS Assessment

We have considered only one MBS ( $|M_m| = 1$ ), ten FBSs ( $|M_f| = 10$ ). All FBSs are initially off, and they will turn on if required sequentially. Under the SC scheme, all the users have an association with MBS only. In contrast, under

<sup>14</sup>Number of items refers to the different number of elements in the different instances that have been used for measuring the execution time of the different proposed algorithms. In our proposed solution, we have mapped items to UEs.

Table VIII  
STATISTICS OF THE SA AFTER 10 RUNS ON 10 DIFFERENT INSTANCES OF MKP

Inst.	$AM_{DSA}$	$\sigma_{DSA}$	$N_{opt_{DSA}}$	$AM_{RS^3A}$	$\sigma_{RS^3A}$	$N_{opt_{RS^3A}}$	$AM_{PIRS^3A}$	$\sigma_{PIRS^3A}$	$N_{opt_{PIRS^3A}}$
Inst.1	3884	170.16	0	4030	122.67	0	4374	31.78	1
Inst.2	4422	165.8	0	4553	102.02	0	4956	25.14	3
Inst.3	4656	431.07	0	4990	100.76	2	5382	5.79	7
Inst.4	4270	290.23	0	4622	106.05	1	4966	21.36	5
Inst.5	3290	91.14	0	3423	83.40	4	3594	7.08	6
Inst.6	4121	146.47	0	4250	82.32	1	4526	11.72	2
Inst.7	4058	142.29	0	4237	103.66	3	4503	20.25	6
Inst.8	3995	135.70	0	4090	111.30	0	4402	9.42	4
Inst.9	3563	181.46	0	3818	99.36	0	4028	23.78	4
Inst.10	3899	200.67	0	4207	100.50	1	4525	18.40	4

Table IX  
EXECUTION TIME OF DIFFERENT VERSIONS OF SA AND DGA

Time(ms)	Number of Items					
Scheme	50	100	200	500	1000	2000
DSA	0.00028	0.00099	0.0035	0.025	0.090	0.36
$RS^3A$	0.00013	0.00041	0.0015	0.009	0.037	0.14
$PIRS^3A$	0.00011	0.00040	0.0015	0.002	0.020	0.11
Time(s)	Number of Items					
Scheme	50	100	200	500	1000	2000
DGA	72.76	104.89	169.25	334.30	503.12	925.93

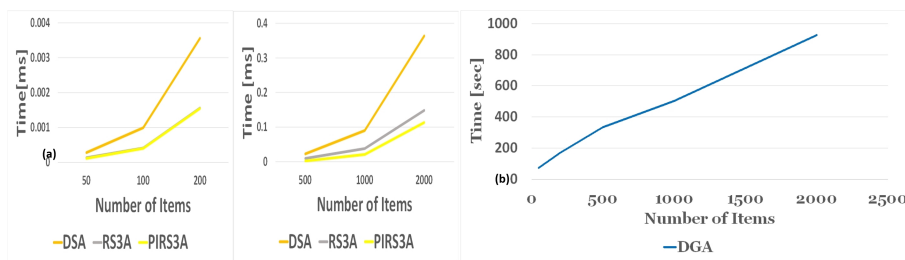


Figure 10. (a) Execution time for different versions of SA. (b) Execution time of DGA.

the DSA scheme, we formulated the UA-RA problem as the MKP problem. We solved it using DSA, i.e., without any fine-tuning in parameters range and solution space. On the other hand, under the  $PIRS^3A$  scheme, we formulated the UA-RA problem as MKP and solved it using SA, but fine-tuning both in parameters range and solution space. We assessed QoS metrics (throughput (Mbps), PLR, delay, and jitter) perceived by users in a HetNets environment and results are presented in Fig. 11, Fig. 12, and Fig. 13 with error bar which corresponds to 95 % confidence interval (CI).

1) **70 Users:**  $|\mathcal{N}| = 70$  users are randomly placed in a given area  $G$ . Fig. 11(a) results show that under the SC scheme, users achieve an average 0.45 Mbps throughput with 95% CI equals to 0.34 - 0.51 Mbps. Under the DSA scheme, users achieve an average 1.34 Mbps throughput with 95% CI equals to 1.03 - 1.64 Mbps and 2.22 Mbps with 95% CI equals to 1.80 - 2.64 Mbps under  $PIRS^3A$  scheme (TABLE. X). The  $PIRS^3A$  throughput is 40% higher than that was achieved when the DSA scheme is employed and 80% higher than achieved when the SC scheme is employed.

In terms of PLR, from Fig. 11(b), we note that under the SC scheme, users experienced more than 54.5% packet loss

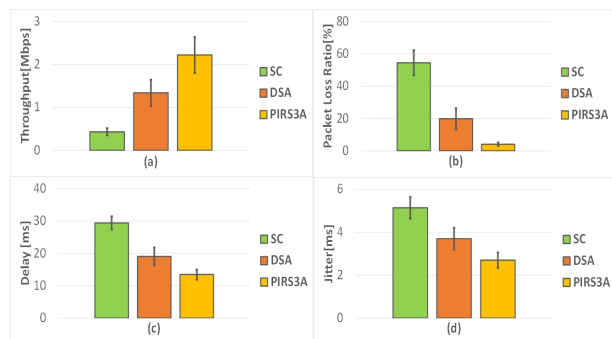


Figure 11. QoS Assessment under different schemes for 70 Users.

with CI<sup>15</sup> equals to 46.7 - 62.2 %. Under the DSA scheme, the user experienced more than 19 % packet loss with CI equals 13.45 - 26.35 % and almost 4% packet loss with CI equals 3 - 5 % under  $PIRS^3A$  scheme (TABLE. X). The  $PIRS^3A$  PLR is 78% less than that experienced when the DSA scheme is employed and 93% less than that achieved when the SC scheme is used.

In terms of delay, from Fig. 11(c), we observe that under

<sup>15</sup>Note that from now on we use 95% CI or CI interchangeably

Table X  
QoS METRICS RESULTS FOR 70 USERS

Throughput (Mbps)			
	SC	DSA	$PIRS^3A$
Average	0.4313	1.34	2.22
StdDev	0.3616	1.31	1.787
95% CI	0.34-.51	1.03-1.64	1.80-2.64
PLR (%)			
	SC	DSA	$PIRS^3A$
Average	54.5	19.9	4.1
StdDev	33.17	27.57	4.52
95% CI	46.7-62.2	13.4-26.3	3.04-5.15
Delay (ms)			
	SC	DSA	$PIRS^3A$
Average	29.43	19.06	13.47
StdDev	8.63	11.75	6.7
95% CI	27.41-31.45	16.31-21.81	11.7-15.04
Jitter (ms)			
	SC	DSA	$PIRS^3A$
Average	5.14	3.7	2.7
StdDev	2.12	2.4	1.7
95% CI	4.64-5.64	3.19-4.21	2.34-3.06

Table XI  
QoS METRICS RESULTS FOR 85 USERS

Throughput (Mbps)			
	SC	DSA	$PIRS^3A$
Average	0.3687	1.48	2.14
StdDev	0.3085	1.46	1.849
95% CI	0.30-.43	1.17-1.79	1.75-2.53
PLR (%)			
	SC	DSA	$PIRS^3A$
Average	57.3	20.2	6.02
StdDev	33.33	28	7.89
95% CI	50.21-64.39	14.32-26.22	4.34-7.70
Delay (ms)			
	SC	DSA	$PIRS^3A$
Average	29.65	19.45	15.5
StdDev	8.2	12.45	9.45
95% CI	27.91-31.39	16.81-22.09	12.99-17.5
Jitter (ms)			
	SC	DSA	$PIRS^3A$
Average	5.38	4.11	3.56
StdDev	2.15	2.43	2.95
95% CI	4.93-5.83	3.54-4.68	2.87-4.25

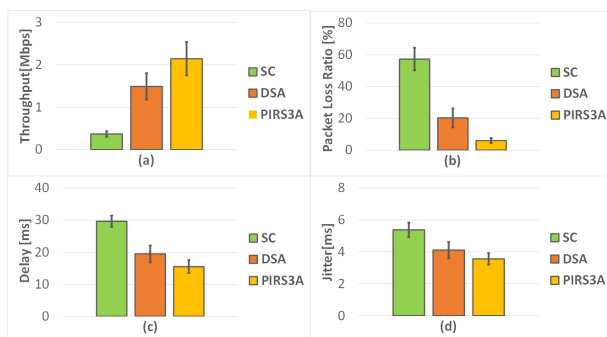


Figure 12. QoS Assessment under different schemes for 85 Users.

the SC scheme, users experienced around 29.5 ms delay with  $CI$  equals to 27.41-31.45 ms. Under the DSA scheme, the user experienced around 19 ms delay with  $CI$  equals 16.31-21.81 ms and almost 13.5 ms with  $CI$  equal to 11.7-15.04 ms under  $PIRS^3A$  scheme (TABLE. X). The  $PIRS^3A$  delay is 29.32% less than that experienced when the DSA scheme is employed and 54% less than what was achieved when the SC scheme is employed.

In terms of jitter, from Fig. 11(d), it can be seen that under the SC scheme, users experienced around 5.1 ms jitter with  $CI$  equals 4.64-5.64 ms. When the DSA scheme was employed, the user experienced around 3.7 ms jitter with  $CI$  equals 3.19-4.21 ms and almost 2.7 ms with  $CI$  equals 2.34-3.06 ms under  $PIRS^3A$  scheme (TABLE. X). The  $PIRS^3A$  jitter is 27% less than what was experienced when the DSA scheme is employed and 48% less than achieved when the SC scheme is used.

From the above results, we can conclude that our proposed scheme  $PIRS^3A$  has outperformed the other two schemes in terms of all QoS metrics considered, hence satisfying UEs  $\psi_j$ .

2) **85 Users:**  $|N| = 85$  users are randomly placed in the given area  $G$ . From Fig. 12(a) we can analyze that under the SC scheme, users achieve an average 0.36 Mbps throughput with 95%  $CI$  equals to 0.30 - 0.43 Mbps. Under the DSA

scheme, users achieve an average 1.48 Mbps throughput with 95%  $CI$  equals to 1.17- 1.79 Mbps and 2.14 Mbps with 95%  $CI$  equals to 1.75 - 2.53 Mbps under  $PIRS^3A$  scheme (TABLE. XI). The  $PIRS^3A$  throughput is 31% higher than that was achieved when the DSA scheme is employed, and 83% higher than that was achieved when SC is employed.

In terms of PLR, from Fig. 12(b), we observe that under the SC scheme, users experienced more than 57.3% packet loss with  $CI$  equals 50.21 64.39 %. Under the DSA scheme, the user experienced more than 20 % packet loss with  $CI$  equals 14.32 - 26.22 % and almost 6% packet loss with  $CI$  equals 4.34 - 7.70 % under  $PIRS^3A$  scheme (TABLE. XI).

The  $PIRS^3A$  PLR is 70% lesser than that was experienced when the DSA scheme is employed and 89% lesser than that was achieved when the SC scheme is employed.

In terms of Delay, from Fig. 12(c), we note that under the SC scheme, users experienced around 29.65 ms delay with  $CI$  equals to 27.91-31.39 ms. Under the DSA scheme, the user experience around 19.45 ms Delay with  $CI$  equals 16.81-22.09 ms, and almost 15.5 ms with  $CI$  equals 12.99-17.5 ms under  $PIRS^3A$  scheme (TABLE. XI). The  $PIRS^3A$  Delay is 20% lesser than that was experienced when the DSA scheme is employed and 48% lesser than that was achieved when the SC scheme is used.

In terms of Jitter, from Fig. 12(d), we observe that under the SC scheme, users experienced around 5.38 ms jitter with  $CI$  equals 4.93-5.83 ms. Under the DSA scheme, the user experienced around 4.11 ms jitter with  $CI$  equals 3.54-4.68 ms and almost 3.56 ms with  $CI$  equals 2.87-4.25 ms under  $PIRS^3A$  scheme (TABLE. XI). The  $PIRS^3A$  Jitter is 13.3% lesser than that was experienced when the DSA scheme is employed and 34% lesser than that was achieved when the SC scheme is employed.

From the above results, we can infer that when the number of users increased to 85, our proposed scheme still satisfies users' demand  $\psi_j$  and outperformed the other two schemes in terms of all QoS metrics. Hence, this addresses our proposed

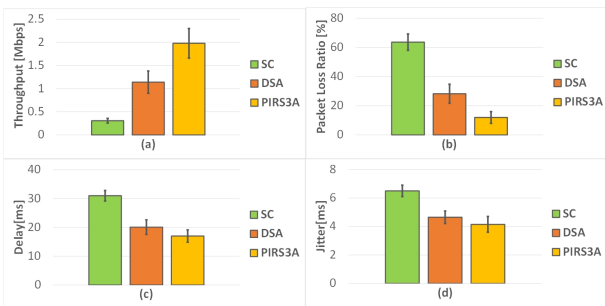


Figure 13. QoS Assessment under different schemes for 100 Users.

scheme's potential, which can satisfy users' demands for such massive traffic.

3) **100 Users:**  $|N| = 100$  users are randomly placed in a given area  $G$ . Fig. 13(a) results show that under the SC scheme, users achieve an average 0.30 Mbps throughput with 95%  $CI$  equals to 0.25 - 0.35 Mbps. Under the DSA scheme, users achieve an average 1.14 Mbps throughput with 95%  $CI$  equals to 0.90- 1.38 Mbps and 1.98 Mbps with 95%  $CI$  equals to 1.66 - 2.30 Mbps under  $PIRS^3A$  scheme (TABLE. XII). The  $PIRS^3A$  throughput is 42% higher than that was achieved when the DSA scheme is employed and 84% higher than that was achieved when the SC scheme is employed.

In terms of PLR, from Fig. 13(b), we can infer that under the SC scheme, users experienced more than 63% packet loss with  $CI$  equals 58 - 69.19 %. Under the DSA scheme, user experienced more than 20 % packet loss with  $CI$  equals to 21.77 - 34.76 % and almost 12% packet loss with  $CI$  equals to 7 - 15.98 % under  $PIRS^3A$  scheme (TABLE. XII). The  $PIRS^3A$  PLR is 57% less than what was experienced when the DSA scheme is employed and 81% less than what was achieved when the SC scheme is employed.

In terms of delay, from Fig. 13(c), it can be seen that under the SC scheme, users experienced around 31 ms delay with  $CI$  equals to 29.18-32.18 ms. Under the DSA scheme, the user experience around 20.1 ms delay with  $CI$  equals 17.58-22.62 ms and almost 17 ms with  $CI$  equal to 14.88-19.12 ms under  $PIRS^3A$  scheme (TABLE. XII). The  $PIRS^3A$  delay is 15.4% less than what was experienced when the DSA scheme is employed and 45.16% less than what was achieved when the SC scheme is used.

In terms of jitter, from Fig. 13(d), we note that under the SC scheme, users experienced around 6.5 ms jitter with  $CI$  equals 6.1-6.9 ms. Under the DSA scheme, the user experienced around 4.65 ms jitter with  $CI$  equals 4.22-5.08 ms and almost 4.15 ms with  $CI$  equals 3.61-4.69 ms under  $PIRS^3A$  scheme (TABLE. XII). The  $PIRS^3A$  jitter is 11% less than what was experienced when the DSA scheme is employed and 36% less than what was achieved when the SC scheme is used.

From the above results, we observe that when the number of users increased to 100, our proposed scheme roughly satisfies user demands  $\psi_j$  but achieves PLR higher than 10%. However,  $PIRS^3A$  outperformed well the other two schemes in terms of all QoS metrics.

Table XII  
QoS METRICS RESULTS FOR 100 USERS

Throughput (Mbps)			
	SC	DSA	$PIRS^3A$
Average	0.3072	1.14	1.98
StdDev	0.253	1.22	1.63
95% CI	0.25-0.35	0.90-1.38	1.66-2.30
PLR (%)			
	SC	DSA	$PIRS^3A$
Average	63.59	28.26	11.98
StdDev	28.7	33.18	20.65
95% CI	57.99-69.19	21.77-34.76	7-15.98
Delay (ms)			
	SC	DSA	$PIRS^3A$
Average	31	20.1	17
StdDev	9.3	12.9	10.8
95% CI	29.18-32.18	17.58-22.62	14.88-19.12
Jitter (ms)			
	SC	DSA	$PIRS^3A$
Average	6.5	4.65	4.15
StdDev	2.05	2.22	2.80
95% CI	6.1-6.9	4.22-5.08	3.61-4.69

## VIII. CONCLUSIONS AND FUTURE WORK

This paper proposed two innovative simulated annealing based solutions:  $RS^3A$ , and  $PIRS^3A$ , designed for solving the complex user association and resource allocation (UA-RA) problem in HetNets. The UA-RA problem was formulated as an MKP where BSs represent the knapsacks in this solution, and UEs are the items to be fitted into the knapsacks.

It is shown that the decentralized scheme  $PIRS^3A$  converges to an optimal solution in comparison to other two versions of SA i.e. DSA and  $RS^3A$  and DGA, with reduced complexity and low overhead time. It was also demonstrated that optimization function represents a *hyper-plane* and its *convex* and maximum BS capacity, TBS index constraints are *convex* and separately reflect *half-spaces* and intersection of *half-spaces* and *hyper-planes* form a *polyhedron*.

The numerical results demonstrate that context-awareness factors like maximum BS capacity, UE demands, and channel conditions (TBS index) significantly improve resource utilization and result in user QoS improvements. Future work will include 1) integration of dynamic interference mitigation with the UA-RA problem and proposal of a novel approach jointly solving it, and 2) considering quality of experience (QoE) estimation in the problem solving process.

### APPENDIX A PROOF OF LEMMA 1

$\tilde{p}$  represents a vector of values of items and  $\tilde{x}$  represents a vector of association i.e the values of  $\tilde{x}$  can either be 1 or 0. The problem is to maximize  $\tilde{p}^T * \tilde{x}$  and  $\tilde{p}^T * \tilde{x}$  can achieve any value in each episode and we consider that value  $Z_{max}$ .

$$\tilde{p}^T * \tilde{x} = Z_{max} \quad (17)$$

where, (17) represents a *HYPER-PLANE*(Fig. 14). In order to proof *HYPER-PLANES* are *convex*, it needs to satisfy the weighted linear combination of

$$\theta * \tilde{x}_1 + (1 - \theta) * \tilde{x}_2 = z, 0 \leq \theta \leq 1 \quad (18)$$

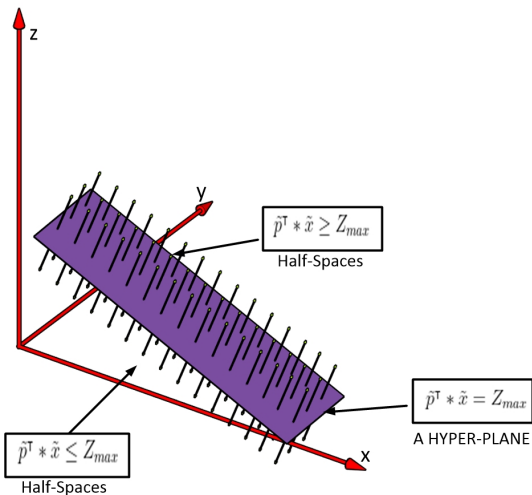


Figure 14. A three-dimension representation of Hyper-Plane and Half-Spaces

where  $\tilde{x}_1$  and  $\tilde{x}_2$  are two vectors in  $n$  dimensional space. Let

$$\tilde{p}^T * \tilde{x}_1 = Z_{max} \quad (19)$$

$$\tilde{p}^T * \tilde{x}_2 = Z_{max} \quad (20)$$

$$\begin{aligned} \tilde{p}^T * (\theta * \tilde{x}_1 + (1 - \theta) * \tilde{x}_2) &= \theta * \tilde{p}^T * \tilde{x}_1 + (1 - \theta) * \tilde{p}^T * \tilde{x}_2 \\ &= \theta * Z_{max} + (1 - \theta) * Z_{max} \\ &= Z_{max} \end{aligned} \quad (21)$$

If there is no restrictions on  $\theta$ , then it is also a *Affine* function. Hence all *Affine* functions are *convex*, but converse is not true.

#### ACKNOWLEDGMENTS

This work was conducted with the financial support of the Science Foundation Ireland Centre for Research Training in Digitally-Enhanced Reality (D-REAL) under Grant No. 18/CRT/6224. The Science Foundation Ireland (SFI) support via the research grants 16/SP/3804 (Enable) and 12/RC/2289\_P2 (Insight) is also gratefully acknowledged.

#### REFERENCES

- [1] Ericsson, *Ericsson Mobility Report November 2020*, Ericsson, Nov. 2020. [Online]. Available: <https://www.ericsson.com/en/mobility-report/reports/november-2020>
- [2] F. Guidolin, I. Pappalardo, A. Zanella and M. Zorzi, "Context-Aware Handover Policies in HetNets," in *IEEE Transactions on Wireless Communications*, vol. 15, no. 3, pp. 1895-1906, March 2016, doi: 10.1109/TWC.2015.2496958.
- [3] M. Feng, S. Mao and T. Jiang, "Joint Frame Design, Resource Allocation and User Association for Massive MIMO Heterogeneous Networks With Wireless Backhaul," in *IEEE Transactions on Wireless Communications*, vol. 17, no. 3, pp. 1937-1950, March 2018.
- [4] Z. I. Kiss, A. C. Hosu, M. Varga and Z. A. Polgar, "Load balancing solution for heterogeneous wireless networks based on the knapsack problem," *2015 38th International Conference on Telecommunications and Signal Processing (TSP)*, Prague, 2015, pp. 1-6.
- [5] A. Alnoman, L. Ferdouse and A. Anpalagan, "Fuzzy-Based Joint User Association and Resource Allocation in HetNets," *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*, Toronto, ON, 2017, pp. 1-5.
- [6] F. Wang, W. Chen, H. Tang and Q. Wu, "Joint Optimization of User Association, Subchannel Allocation, and Power Allocation in Multi-Cell Multi-Association OFDMA Heterogeneous Networks," in *IEEE Transactions on Communications*, vol. 65, no. 6, pp. 2672-2684, June 2017.
- [7] M. Feng, S. Mao and T. Jiang, "Joint Frame Design, Resource Allocation and User Association for Massive MIMO Heterogeneous Networks With Wireless Backhaul," in *IEEE Transactions on Wireless Communications*, vol. 17, no. 3, pp. 1937-1950, March 2018.
- [8] L. Zhong, M. Li, Y. Cao and T. Jiang, "Stable User Association and Resource Allocation Based on Stackelberg Game in Backhaul-Constrained HetNets," in *IEEE Transactions on Vehicular Technology*, vol. 68, no. 10, pp. 10239-10251, Oct. 2019
- [9] N. Sapountzis, T. Spyropoulos, N. Nikaiein and U. Salim, "Optimal downlink and uplink user association in backhaul-limited HetNets," *IEEE INFOCOM 2016 - IEEE International Conference on Computer Communications*, San Francisco, USA, 2016, pp. 1-9.
- [10] X. Luo, "Delay-Oriented QoS-Aware User Association and Resource Allocation in Heterogeneous Cellular Networks," in *IEEE Transactions on Wireless Communications*, vol. 16, no. 3, pp. 1809-1822, March 2017.
- [11] E. Barbosa, E. Senne and M. Silva, "Improving the Performance of Metaheuristics: An Approach Combining Response Surface Methodology and Racing Algorithms," *International Journal of Engineering Mathematics*, vol. 2015, pp. 1-9, 2015.
- [12] I. Akyar, *Wide spectra of quality control*. Rijeka, Croatia: InTech, 2011.
- [13] A. I. Khuri and S. Mukhopadhyay, "Response surface methodology," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 2, pp. 128-149, 2010.
- [14] A. I. Khuri, "A General Overview of Response Surface Methodology," *Biometrics & Biostatistics International Journal*, vol. 5, no. 3, 2017.
- [15] M. Birattari, T. Stützle, L. Paquete, and K. Varrentrapp, "A Racing Algorithm for Configuring Metaheuristics," *Proceedings of the 4th Annual Conference on Genetic and Evolutionary Computation*, pp. 11-18, Jul. 2002.
- [16] M. Birattari, Z. Yuan, P. Balaprakash, and T. Stützle, "F-Race and Iterated F-Race: An Overview," *Experimental Methods for the Analysis of Optimization Algorithms*, pp. 311-336, 2010.
- [17] K. Satoda, E. Takahashi, T. Onishi, T. Suzuki, D. Ohta, K. Kobayashi, and T. Murase, "Passive Method for Estimating Available Throughput for Autonomous Off-Peak Data Transfer," *Wireless Communications and Mobile Computing*, vol. 2020, pp. 1-12, 2020.
- [18] E. Yaacoub and Z. Dawy, "LTE radio network planning with HetNets: BS placement optimization using simulated annealing," *MELECON 2014 - 2014 17th IEEE Mediterranean Electrotechnical Conference*, Beirut, 2014, pp. 327-333.
- [19] Y. Huo, X. Fan, L. Ma, X. Cheng, Z. Tian and D. Chen, "Secure Communications in Tiered 5G Wireless Networks With Cooperative Jamming," in *IEEE Transactions on Wireless Communications*, vol. 18, no. 6, pp. 3265-3280, June 2019.
- [20] T. Z. Oo, N. H. Tran, W. Saad, D. Niyato, Z. Han and C. S. Hong, "Offloading in HetNet: A Coordination of Interference Mitigation, User Association, and Resource Allocation," in *IEEE Transactions on Mobile Computing*, vol. 16, no. 8, pp. 2276-2291, Aug. 2017.
- [21] A. Köse and B. Özbek, "Resource Allocation for Underlying Device-to-Device Communications Using Maximal Independent Sets and Knapsack Algorithm," *2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Bologna, 2018, pp. 1-5.
- [22] G. Sahu and S. S. Pawar, "Resource allocation using Genetic Algorithm in Heterogeneous Network," *2019 IEEE Pune Section International Conference (PuneCon)*, Pune, India, 2019, pp. 1-7.
- [23] B. Agarwal, M. A. Togou, M. Ruffini and G. -M. Muntean, "Mitigating the Impact of Cross-Tier Interference on Quality in Heterogeneous Cellular Networks," *2020 IEEE 45th Conference on Local Computer Networks (LCN)*, Sydney, NSW, Australia, 2020.
- [24] B. Agarwal, M. A. Togou, M. Ruffini, and G.-M. Muntean, "PIRS3A: A Low Complexity Multi-knapsack-based Approach for User Association and Resource Allocation in HetNets," *2021 IEEE International Conference on Communications Workshops (ICC Workshops)*, Montreal, Canada, 2021 pp. 1-6.
- [25] A. Liu, J. Wang, G. Han, S. Wang and J. Wen, "Improved Simulated Annealing Algorithm Solving for 0/1 Knapsack Problem," *Sixth International Conference on Intelligent Systems Design and Applications*, Jinan, 2006, pp. 1159-1164.
- [26] J. D. Lamos-Sweeney, "Deep learning using genetic algorithm," Master's thesis, Rochester Institute of Technology, Rochester, NY, May 2012.
- [27] "AlphaGo Zero: Starting from scratch," *Deepmind*. [Online]. Available: <https://deepmind.com/blog/article/alphago-zero-starting-scratch>.
- [28] L. M. R. Rere, M. I. Fanany, and A. M. Arymurthy, "Simulated annealing algorithm for deep learning," *Procedia Computer Science*, vol. 72, pp. 137-144, 2015.



- [29] L. Ingber, "Simulated annealing: Practice versus theory," *Mathematical and Computer Modelling*, vol. 18, no. 11, pp. 29–57, 1993.
- [30] Cai, Qingpeng, et al. "Reinforcement Learning Driven Heuristic Optimization," *arXiv preprint*, arXiv:1906.06639 (2019).
- [31] L. Huang, X. Feng, A. Feng, Y. Huang, and L. P. Qian, "Distributed Deep Learning-based Offloading for Mobile Edge Computing Networks," *Mobile Networks and Applications*, 2018.
- [32] H. Jiang, D. Peng, K. Yang, Y. Zeng, and Q. Chen, "Predicted mobile data offloading for mobile edge computing systems," in *Proc. Int. Conf. Smart Comput. Commun.*, 2018, pp. 153–162.
- [33] C. H. Liu, Z. Chen and Y. Zhan, "Energy-Efficient Distributed Mobile Crowd Sensing: A Deep Learning Approach," in *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1262-1276, June 2019.
- [34] L. Xiao, C. Xie, T. Chen, H. Dai and H. V. Poor, "A Mobile Offloading Game Against Smart Attacks," in *IEEE Access*, vol. 4, pp. 2281-2291, 2016.
- [35] Y. He, N. Zhao and H. Yin, "Integrated Networking, Caching, and Computing for Connected Vehicles: A Deep Reinforcement Learning Approach," in *IEEE Transactions on Vehicular Technology*, vol. 67, no. 1, pp. 44-55, Jan. 2018.
- [36] C. H. Liu, Z. Chen, J. Tang, J. Xu and C. Piao, "Energy-Efficient UAV Control for Effective and Fair Communication Coverage: A Deep Reinforcement Learning Approach," in *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 9, pp. 2059-2070, Sept. 2018.
- [37] H. Li, K. Ota and M. Dong, "Learning IoT in Edge: Deep Learning for the Internet of Things with Edge Computing," in *IEEE Network*, vol. 32, no. 1, pp. 96-101, Jan.-Feb. 2018, doi: 10.1109/MNET.2018.1700202.
- [38] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger, "Deep reinforcement learning that matters," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 1–26.
- [39] Z. Xiong, Y. Zhang, D. Niyato, R. Deng, P. Wang and L. Wang, "Deep Reinforcement Learning for Mobile 5G and Beyond: Fundamentals, Applications, and Challenges," in *IEEE Vehicular Technology Magazine*, vol. 14, no. 2, pp. 44-52, June 2019.
- [40] N. Zhao, Y. Liang, D. Niyato, Y. Pei, M. Wu and Y. Jiang, "Deep Reinforcement Learning for User Association and Resource Allocation in Heterogeneous Cellular Networks," in *IEEE Transactions on Wireless Communications*, vol. 18, no. 11, pp. 5141-5152, Nov. 2019.
- [41] P. Kumar, "Time Complexity of ML Models," *Medium*, 31-Mar-2021. [Online]. Available: <https://medium.com/analytics-vidhya/time-complexity-of-ml-models-4ec39fad2770>.
- [42] M. Bianchini and F. Scarselli, "On the Complexity of Neural Network Classifiers: A Comparison Between Shallow and Deep Architectures," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 8, pp. 1553–1565, 2014.
- [43] A. Blum, C. Dan, and S. Seddighin, "Learning Complexity of Simulated Annealing," *Proceedings of Machine Learning Research*, vol. 130, pp. 1540–1548, 2021.
- [44] E.-G. Talbi, *Metaheuristics: from design to implementation*. Hoboken, NJ: Wiley, 2009.
- [45] M. López-Ibáñez, J. Dubois-Lacoste, L. Pérez Cáceres, M. Birattari and T. Stützle, "The irace package: Iterated racing for automatic algorithm configuration," *Operations Research Perspectives*, vol. 3, pp. 43-58, 2016.
- [46] J. Pender, "The truncated normal distribution: Applications to queues with impatient customers," *Operations Research Letters*, vol. 43, no. 1, pp. 40–45, 2015.
- [47] D. R. Barr and E. T. Sherrill, "Mean and Variance of Truncated Normal Distributions," *The American Statistician*, vol. 53, no. 4, p. 357, 1999.
- [48] I. Siaud and A.-M. Ulmer-Moll, "Green-oriented multi-techno link adaptation metrics for 5G heterogeneous networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2016, no. 1, 2016.
- [49] 3GPP TR 21.916 Rel.16, Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Layer Procedures, 2020.
- [50] E. Takahashi, T. Suzuki, T. Onishi and K. Satoda, "Autonomous off-peak data transfer by passively estimating overall LTE cell load," *2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, Las Vegas, NV, 2017, pp. 754-759.
- [51] E. Takahashi, T. Onishi, T. Shiroshima and H. Sugahara, "A passive method for estimating overall LTE cell traffic load by using radio quality information of a user terminal," *2015 21st Asia-Pacific Conference on Communications (APCC)*, Kyoto, 2015, pp. 17-20.
- [52] F. Qian and R. Ding, "Simulated annealing for the 0/1 multidimensional knapsack problem," *Numerical Mathematics*, vol. 16, pp. 320–327, 2007.



Dublin City University, Ireland.

**Bharat Agarwal** received the B.Eng. degree in Electrical and Electronics Engineering from Galgotias University, India, in 2016. He was a Project Scientist at the Department of Telecommunications, India Institute of Technology, Delhi. He is currently pursuing the Ph.D. degree with the School of Electronics Engineering, Dublin City University, Dublin, Ireland. His research interests include heterogeneous networks, traffic optimization, 5G, LTE, convex optimization. He is a recipient of a SFI scholarship from the D-REAL Centre for Research Training at



**Marco Ruffini** received the M.Eng. degree in telecommunications from the Polytechnic University of Marche, Italy, in 2002, and the Ph.D. degree from Trinity College Dublin (TCD) in 2007, where he joined in 2005, after working as a Research Scientist with Philips, Germany. He is an Associate Professor and a Fellow of Trinity College, and he is a Principal Investigator of both the CONNECT Telecommunications Research Centre and the IPIC Photonics Integration Centre. He is currently involved in several Science Foundation Ireland and

H2020 projects, and leads the Optical Network Architecture Group, TCD. He has authored over 140 international publications, over ten patents and contributed to standards at the broadband forum. His main research is in the area of 5G optical networks, where he carries out pioneering work on the convergence of fixed-mobile and access-metro networks, and on the virtualization of next generation networks, and has been invited to share his vision through several keynote and talks at major international conferences across the world.



**Gabriel Miro Muntean** is a Professor with the School of Electronic Engineering, Dublin City University (DCU), Ireland, and Co-Director of the DCU Performance Engineering Laboratory. Prof. Muntean was awarded the PhD degree by DCU for research on adaptive multimedia delivery in 2004 and BEng and MEng degrees in Software Engineering by "Politehnica" University of Timisoara, Romania in 1996 and 1997, respectively. He has published over 400 papers in top-level international journals and conferences, authored four books and 21 book chapters, and edited six additional books. His research interests include quality, performance, and energy saving issues related to multimedia and multiple sensorial media delivery, technology-enhanced learning, and other data communications over heterogeneous networks. Prof. Muntean is an Associate Editor of the *IEEE Transactions on Broadcasting*, the *Multimedia Communications Area Editor* of the *IEEE Communications Surveys and Tutorials*, and chair and reviewer for important international journals, conferences, and funding agencies. Prof. Muntean is senior member of IEEE and IEEE Broadcast Technology Society.