UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

FACULTAD DE INGENIERÍA MEXICALI

DOCTORADO EN CIENCIAS E INGENIERÍA

Asignación de Recursos con Baja Complejidad basada en Bloques de Canales para Redes Ultra-Densas

Tesis que para cubrir parcialmente los requisitos para
obtener el grado de Doctor en Ciencias presenta:

Adolfo Eleazar Reyna-Orta

Director de Tesis:
Dr. Ángel Gabriel Andrade Reátiga

Universidad Autónoma de Baja California
Mexicali, México

6 de Mayo, 2021

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

FACULTAD DE INGENIERÍA MEXICALI

DOCTORADO EN CIENCIAS E INGENIERÍA

Low Complexity Block Channel-based Resource Allocation for satisfying Ultra-Dense Networks QoS requirements

A thesis in fulfilment of the requirements for the
degree of Doctor of Science

Adolfo Eleazar Reyna-Orta

Thesis Advisor:
Dr. Ángel Gabriel Andrade Reátiga

Universidad Autónoma de Baja California
Mexicali, México

May 6th, 2021

Tesis defendida por:

Adolfo Eleazar Reyna Orta

y aprobada por el siguiente comité:

Dr. Ángel Gabriel Andrade Reátiga
Director de tesis

Dr. Enrique René Bastidas Puga
Miembro del comité

Dr. Felix Fernando Gonzalez Navarro
Miembro del comité

Dr. Carlos Alberto Brizuela Rodríguez
Miembro del comité

Dr. Guillermo Galaviz Yáñez
Miembro del comité

Mexicali, Baja California, a 6 de Mayo del 2021.

Resumen

RESUMEN de la tesis de Adolfo E. Reyna Orta, presentada como requisito parcial para la obtención del grado de DOCTOR EN CIENCIAS. Mexicali, Baja California, México. Mayo del 2021.

Asignación de Recursos con Baja Complejidad basada en Bloques de Canales para Redes Ultra-Densas

Aprueba:____

Dr. Ángel Gabriel Andrade Reátiga
Director de tesis

La quinta y consecuentes generaciones (Beyond Fifth Generation (B5G)) de redes inalámbricas surge como una visión de una red capaz de sostener el crecimiento exponencial de los requisitos de velocidad de datos y densidad de usuarios. B5G proporcionará la infraestructura necesaria para las revolucionarias aplicaciones inteligentes como Internet de las Cosas (IoT), Cuidades Inteligenets, Vehículos Autónomos e Industria 4.0. La densificación de la red es una estrategia prometedora para abordar los objetivos propuestos de B5G de incrementar la capacidad en 100x. La densificación espacial se refiere a la adición de celdas pequeñas al despliegue de macroceldas existente para aumentar el tráfico de datos, proporcionar una alta eficiencia espectral y equilibrar la carga de las celdas. Debido a la enorme cantidad de celdas pequeñas desplegadas en redes ultra densas, los algoritmos de asignación de recursos y las estrategias de mitigación de interferencia se han convertido en desafíos importantes a superar.

La asignación de recursos es el proceso que distribuye los recursos espectrales disponibles de la red a los usuarios activos. Este proceso generalmente se expresa como un problema de optimización que maximiza el rendimiento de la red en eficiencia espectral, eficiencia energética, calidad de servicio (QoS), equidad o una combinación de estos. Dada la complejidad de asignar varios recursos de manera conjunta, este problema de optimización se divide en dos subproblemas secuenciales: asignación de canales, selecciona los canales apropiados para los usuarios utilizando una distribución de potencia uniforme; y asignación de potencia, asignando niveles óptimos de potencia sobre los canales previamente seleccionados.

En las redes ultra densas (UDN), la asignación de recursos juega un papel crucial en la minimización de la interferencia al coordinar que los nodos interferentes utilicen recursos del espectro ortogonales. Aunque mucho esfuerzo de investigación se ha centrado en la asignación de potencia para estas redes, trabajos recientes implementan algoritmos voraces de asignación de canales. Estos enfoques voraces proporcionan una solución subóptima y rápida, pero la interferencia no es considerada completamente. Dada la importancia de la coordinación de interferencia en redes densas, es fundamental realizar una asignación de canales cercana a la óptima. Sin embargo, la carga de la complejidad computacional se ha convertido en uno de los desafíos críticos a superar.

Resolver el problema de optimización de la asignación de canales con soluciones aproximadas como algoritmos metaheurísticos ofrece un equilibrio adecuado entre el rendimiento de la red y los recursos computacionales. No obstante, la elevada complejidad computacional y la alta interferencia siguen siendo los principales desafíos a superar.

Esta tesis investiga la propuesta de estrategias subóptimas eficientes para resolver el problema de la asignación de recursos en implementaciones de UDN para comunicaciones de red B5G. En este trabajo, proponemos un enfoque novedoso para la asignación de canales basado en la asignación de conjuntos predefinidos de bloques de recursos. El objetivo es mejorar la eficiencia del uso de los recursos del espectro y la capacidad de los usuarios individuales, mientras se mantiene un tiempo de respuesta corto para resolver la asignación. La reducción de la complejidad se logra reduciendo el espacio de búsqueda en subconjuntos de bloques de canales continuos, asignados directamente a usuarios o células.

Para analizar el potencial de los esquemas de asignación propuestos, resolvemos el problema de asignacion de canal (CA) utilizando varios algoritmos meta-heurísticos. Los resultados muestran que utilizando los esquemas propuestos se aumenta la probabilidad de que los métodos de optimización de aproximación encuentren soluciones adecuadas con menos recursos computacionales.

Abstract

Abstract of the thesis by Adolfo E. Reyna Orta. A thesis in fulfilment of the requirements for the degree of Doctor of Science. Mexicali, Baja California, México. May of 2021.

Low Complexity Block Channel-based Resource Allocation for satisfying Ultra-Dense Networks QoS requirements

Approved:____

Dr. Ángel Gabriel Andrade Reátiga
Thesis Advisor

The Beyond Fifth Generation (B5G) of wireless networks comes as a vision of a network capable of sustaining user density and data rate requirements exponential growth. B5G will provide the required infrastructure to revolutionary smart applications such as the Internet of Things (IoT), Smart Cities, Driverless Vehicles, Smart Grids, and Industry 4.0. Network densification is a promising strategy to address the proposed 100x capacity goal of B5G systems. Space densification refers to adding small-cells into the existing macro-cell layout to increase wireless data traffic, provide high spectral efficiency and balance the cell load. Due to the massive amount of small cells deployed in ultra-dense networks, resource allocation algorithms and interference mitigation strategies have become significant challenges to overcome.

Resource allocation is the process that distributes the available spectral resources of the network to the active users. This process is usually expressed

as an optimization problem to maximize the performance of the network on Spectral Efficiency, Energy Efficiency, Quality of Service (QoS), Fairness, or a combination of these. Given the complexity of allocating several resources jointly, this optimization problem is separated into two sequential sub-problems: channel allocation, selecting appropriate channels for the users using a uniform power distribution; and power allocation, assigning optimal power levels over the previously selected channels.

In ultra-dense networks (UDN), resource allocation plays a crucial role in minimizing interference by coordinating interfering nodes to use orthogonal spectrum resources. Although much research effort has focused on the power allocation for these networks, recent works implement centralized greedy channel allocation algorithms. These greedy approaches provide a fast sub-optimal solution, but interference is usually not fully considered. Given the importance of interference coordination in dense networks, it is crucial to perform close to optimal channel allocation. However, the computational complexity burden has become one of the critical challenges to overcome.

Solving the channel allocation optimization problem with approximated solutions such as metaheuristic algorithms offers a proper balance between the network's performance and the computational resources. Nonetheless, elevated computational complexity and high interference are still the main challenges.

This thesis investigates the proposal of efficient sub-optimal strategies for the Resource Allocation problem in UDN deployments for B5G network communications. In this work, we propose a novel approach to allocate channels based on the assignment of predefined sets of resource blocks. The objective is to improve the efficiency of spectrum resource use and individual user capacity, while maintaining a short response time to solve tje resource allocation. The reduction of complexity is achieved by reducing the search space into subsets of continuous channel blocks, allocated directly to either users or cells. To analyze the potential of the proposed allocation schemes, we solve the channel allocation (CA) problem using several metaheuristic algorithms. The results show that the proposed schemes increase the likelihood that approximation optimization methods will find suitable solutions with fewer computational resources.

Agradecimientos

Doy gracias a Dios por todas las buenas personas que me han guiado y me han empujado en la dirección correcta.

Gracias a mi asesor y amigo el Dr. Ángel, que me enseñó por el camino largo de la ciencia. No obstante, siempre trabajando juntos en cada paso.

Gracias a los profesores y al personal de la UABC que siempre me trataron bien y compartieron generosamente sus conocimientos y amabilidad.

Un agradecimiento especial a CONACYT y a mi país México que generosamente brindan esta oportunidad de educación superior. Una gran deuda, impagable en muchos sentidos, pero libre de deudas.

Gracias a mis padres Adolfo y Lucy, de quienes obtuve la alegría de aprender, la astucia y la resistencia. Gracias por todas las noches largas y por afrontar conmigo todas las dificultades debidas a los grandes sueños.

Gracias a mis mentores Dr. Santiago, Luz Lastra y familia, por recibirme como uno de ellos. Es su ayuda y apoyo lo que me motiva a seguir adelante.

Por último, pero no menos importante, gracias a mi persona favorita, mi esposa Alicia Gillespie, por ser el punto donde la pendiente de la gráfica cambia continuamente de positivo a cero, después a negativo.

Acknowledgements

I thank to God for all the good poeple that had guide me, and push me in the right direction.

Thanks to my advisor and friend Dr. Ángel, who teach me how to do science the hard way. Nonetheless, always working together trough each step.

Thanks to the UABC's professors and personal that always treated me well and generously share their knowledge and kindness.

Special thanks to Concyt and my country Mexico that generously provide for this oportunity of high education. A great debt, in many ways unpayable, but yet free.

Thanks to my parents Adolfo and Lucy, from whom I got the ichiness to learn, the cleverness and endurance. Thank you for all the long nights and for facing with me all the hardshipts that come with big dreams.

Thanks to my mentors Dr. Santiago, Luz Lastra, and family, to welcomed me as part of their family. It is your help and support that motivates me keep on going.

Last but not least, thanks to my favorite person, my wife Alicia Gillespie, you rock bae.

Dedication

To my son and daughter.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Mobile Networks have become an essential part of daily life for most people in developed countries. According to the data reported by GSMA Intelligence in [1], there are now over 9.5 billion mobile connections worldwide. In the last four years the number of connections coming from smartphones with internet capabilities has experienced a 40% growth (being 3.5 billions currently). It is predicted that this increasing rate of connections and smartphones will continue so that, by 2025, 72% of all internet users will solely use smartphones to access the web [2]. In addition to the increase of connected devices to mobile networks, new trends of data hungry applications (e.g. multimedia streaming, online gaming, Internet of Things (IoT)) will produce an exponential rate increase on the network resources demands. This explosive data rate demand is continually pushing the expansion and technological evolution of wireless networks.

The Beyond Fifth Generation (B5G) of wireless networks comes as a vision of a network capable of sustaining user density's exponential growth ($devices/km^2$) and data rate requirements. B5G will provide the required infrastructure to the revolutionary smart applications such as the Industrial IoT (IIoT) with cloudification, Holographic Type Communications (HTC), Tactile Internet for Remote Operations (TIRO), or Digital Twin (DT) [3]. The B5G vision requires an increment in the current 4G or 5G network capabilities to serve over 10x connected devices, 100x in data rate, provide a seamless connection (indoors / outdoors), low latencies, among other improvements. B5G aims to change the world by connecting anything to anything [4].

It is expected that the 100x increase of network capacity will be achieved by a combination of three symbiotic research directions (represented in Fig. 1.1):

1. Using more spectrum through additional frequency bands, such as the millimeter wave band (mmWave) [5].

2. Increasing the spectral efficiency through massive Multiple Input Multiple Output (MIMO) technology [6].

3. Densifying the network by deploying many small cells (SC) in the current network infrastructure [7].

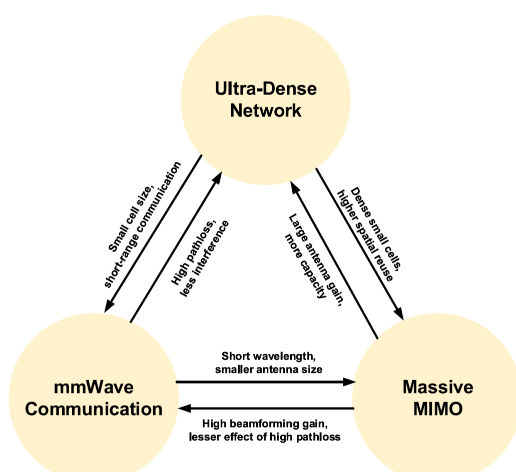


Figure 1.1: Three main research directions to improve network capacity in B5G wireless systems

The capacity of wireless networks increases linearly with the bandwidth available. Therefore, adding more bandwidth is always desirable for operators of wireless networks. However, the available spectrum for mobile networks is currently fully distributed, resulting in spectrum scarcity [8]. This spectrum shortage has led to developing novel approaches of distributing and using the spectrum, such as Cognitive Radio [9] or Edge Computing [10]. Nonetheless, while spectrum has become scarce at microwave frequencies, it is plentiful in the mmWave realm. Although far from being fully understood, researchers with diverse backgrounds study different mmWave transmission aspects [11]. Some technologies have already been standardized for short-range services (e.g., IEEE 802.11ad). However, mmWave capacity can be highly intermittent due to mmWave signals' vulnerability to blockages and delays in directional searching [12]. Such highly variable links present unique challenges for adaptive control mechanisms in transport layer protocols and end-to-end applications.

Massive MIMO technologies highly increases the number of antenna elements per device (100 or more), compared to conventional MIMO systems. These have the potential to improve the capacity of mobile networks through beamforming techniques. These techniques bring two main benefits: Precoding and Spatial multiplexing. Precoding increases the received signal gain by constructively adding the signals emitted from different antennas. Simultaneously, Spatial multiplexing forms narrow beams that allow simultaneous orthogonal communication between several users in a single cell, suppressing interference by directing energy to desired terminals only [13]. Furthermore, at higher frequencies, antennas can be miniaturized, making the application of massive MIMO in mmW even more appealing for B5G networks. The conjunction of these benefits avoids fading dips and reduces the latency on the air interface [14], [15].

Network densification is a promising strategy to address the proposed 100x capacity goals of B5G systems [7]. Space densification refers to adding small-cells (SC) into the existent macro-cell (MC) layout to increase wireless data traffic, provide high spectral efficiency and balance the cell load ($users/cell$) [16]. These networks, composed of numerous cells of varying sizes sharing the space and spectrum, are usually referred to as Ultra-Dense Heterogeneous Networks (UDN). While sparse Heterogeneous Networks, known as HetNets, are comprised of a few non-uniform spatial distributions of users or wireless nodes in conjunction with the MC, UDN is described as those networks in which the density of $cells/km^2$ is above 1,000, or is higher than the density of $users/km^2$ [17]. The main advantages of space densification are: i) improvement of link strength, ii) reduction of the cells load, and iii) extensive spectrum reutilization [18].

1.1 Ultra-Dense Networks

Network Densification is considered the key enabler for the demanding requirements of 5G [19] and beyond networks [20] to provide ubiquitous immerse connectivity. It has been demonstrated that the use of technologies such as mmWaves and MIMO should be accompanied by significant further cell densification to satisfy growing user demands [21]. Hence, it is agreed that B5G networks would be UDN [22]. However, due to the massive amount of SC deployed in these networks, resource allocation algorithms, interference mitigation strategies, deployment management, and backhaul capacity bottleneck solutions for UDN have become significant challenges to overcome.

The maximum data rate capacity provided to a mobile user depends on the bandwidth, transmission power, load of the cell associated with it, channel gain, noise (considered as a static value), and interference perceived within that bandwidth portion (usually distributed among channels). Usually, the channel gain is composed by two main parts, i) a deterministic value dependent on the signal's travel distance, and 2) a mix of various random behaviours due to physical phenomena such as multi-path and shadowing. For simplifying this analysis, it is assumed that each cell divides its bandwidth and transmission power evenly among its associated users, and the channel gain depends only on the physical distance between the cell and the users' equipment. The first assumption is commonly used to analyze network probability's scalability. The latter occurs when there is a Line of Sight (LoS) between the base station and the user equipment. Given these assumptions, one can observe that the user capacity grows linearly with the bandwidth and logarithmically with the transmission power and channel gain. Therefore, when new SC are deployed in the network, the current network layout's load is distributed among more cells. This characteristic reduces the average cell load ($users/cell$), increasing the bandwidth and transmission power allocated to each user. Furthermore, as these newly SC are deployed in the users' proximity, their channel gain is improved. On the other hand, the interference increases with each added cell, as the bandwidth is shared. Nonetheless, this relationship represents a two-fold improvement of the network capacity, compared to the logarithmic decrease rate counterpart of the interference.

Initially, it was theorized that the capacity of the HetNet would increase linearly and indefinitely as more SC were added to the network [23]. However, recent works, such as [24] and [25], show that this capacity only increases until a certain densification level is reached. Beyond this, interference rises to such a degree that it becomes the dominant factor limiting and degrading the network capacity [25]–[28]. This interference may be exacerbated when SC are randomly installed by individual users outside of planned deployment, generating an unpredictable interference distribution [29]. To continue exploiting the network densification benefits beyond these limits, efficient and reliable spectrum assignment strategies are necessary to reduce this network performance degradation [30]. Consequently, it is essential to design Resource Allocation (RA) algorithms that efficiently and intelligently manage the SC's interference [31], [32].

1.2 Resource Allocation Problem in UDN

Resource Allocation (RA) is the process that distributes the available spectral resources of the network (channel, signal transmission power, and time slot) to the active users. RA is usually expressed as an optimization problem to maximize the network's performance on Spectral Efficiency, Energy Efficiency, Quality of Service (QoS), Fairness, or a combination of these [33]. Based on the adopted approach, the RA problems can be centralized, semi-distributed, or distributed. In HetNets with a shared spectrum, RA faces the extra challenge of coordinating the interference produced between SC (co-tier interference) and the interference between SC and the MC (cross-tier interference). This optimization problem is a non-convex, NP-Hard problem with mixed variables [18]. Typically, RA consists of at least the continuous transmission power and binary channel allocation as decision variables. The dimensionality and search space grows exponentially along with the SC's density and users' demands (i.e., number of active users and QoS). Solving this optimization problem jointly, allocating the different spectral resources, results exceedingly complex [34]. It involves high computation cost, even for small network densities. This complexity makes it difficult to analyze detailed information of the UDN, such as the individual user capacity, which requires the use of extensive evaluations [35]. Hence, the RA problem is usually divided into two consecutive sub-problems:

1) Channel Allocation (CA), selecting appropriate channels for the users using a uniform power distribution.
2) Transmission Power Allocation (PA), assigning optimal transmission power levels over the previously selected channels.

Nonetheless, both sub-problems are still NP-Hard [36] and unpractical to be solved with exact methods in realistic scenarios [31], [37]. Several exact and approximation methods have been applied to solve the CA and PA problems [38], each offering different characteristics in terms of the quality of the solutions and their computational efficiency. Given the NP-Hard nature of these problems, finding the optimal solution requires enumerating all the solutions, which is intractable even for low cell densification. Exact methods, such as the Lagrangian dual decomposition method, can be applied by relaxing the CA problem with time-sharing [39] providing optimal solutions [40]. However, exact approaches are rarely used in UDN RA research [41], as its computational complexity is still an issue in these networks. Heuristics algorithms, such as greedy approaches, provide a fast sub-optimal solution, but interference is usually not fully considered in these approaches. Therefore, these greedy heuristics are generally applied to solve CA, and interference management is implemented using clustering approaches [42] or when solving PA [43].

On the other hand, both CA and PA are usually tackled using approximated methods such as Graph Coloring [31], Iterative Heuristics [44], Game Theory [45], or Metaheuristic Algorithms [46]. Metaheuristic algorithms offer a suitable compromise between the solution's response time and "quality" [47], providing fast enough sub-optimal solutions that allow the evaluation of different UDN aspects using extensive simulation. Among these methods, Genetic Algorithms (GAs) has been widely used and reported to provide a proper balance between the RA solution's efficiency and computational complexity (response time) in comparison to other heuristics [48].

In UDN, RA plays a crucial role in minimizing interference by coordinating interferent nodes to use orthogonal spectrum resources [42]. This interference coordination occurs in the frequency domain, time domain, space domain, power domain, or mixture. However, solving the RA optimization problem for UDN is a difficult task due to the problematic interference generated by many cells and the heavy computational burden involved in this process [41], [43]. In RA's centralized approaches, a central entity takes charge of the information collecting and operation control. This centralization brings disadvantages such as concentrated signaling overhead and high complexity. Therefore, distributed methods for RA have been preferred for UDN. Compared with the centralized schemes, the distributed ones do not need any central entity, and the base station makes the decision individually. However, since the distributed algorithms rely on partial knowledge, the optimal solution can hardly be approached. Comparable to most previous approaches applied to the RA problem on sparse HetNets, recent works for UDN implement a centralized greedy CA algorithm [43], [49]. These works select the channel that offers maximum gain for each user, referred to as multi-user gain, without completely modeling intra-cell or inter-cell interference [50]. However, other studies report that the improvement achieved by this strategy is reduced when applied in UDN because the small number of users per cell and the short distance between SC and user equipments (UE) increases Line-of-Sight (LoS) probability. Hence, all the available channels will theoretically have a maximum gain [51]. Consequently, the CA strategy applied for UDN should aim to reduce the interference instead of exploiting the propagation environment's temporary conditions, as this deteriorates the capacity of the UDN architecture [27].

Although, several works report that in HetNets, solving RA with approximated solutions such as metaheuristic algorithms offered a proper balance between the network's performance and the computational resources. Due to the high dimensionality, extensive search space, and aggressive frequency reuse in UDN, the RA optimization problem quickly becomes too complicated to be solved efficiently by applying these algorithms. The computational complexity burden of CA in UDN has become one of the critical challenges [41]. This situation has prevented these approaches' natural evolution from sparse HetNets to UDN [18]. However, due to the NP-Hard nature of the problem and the open issues of RA in UDN, the application of metaheuristics still seems rather appropriate. Therefore, new strategies are needed to reduce adverse effects mentioned above and find feasible and efficient solutions. These strategies must consider some insight into UDN that allows guiding the stochastic search to proper sharing structures between interfering nodes.

There are two recent trending strategies that aim to simplify solving RA in UDN: semi-distributed approaches and mean-field theory [41]. Semi-distributed techniques combine the advantages of both centralized and distributed schemes and mitigate the disadvantages. A typical way of semi-distributed scheme implementation is to divide the entities into small clusters. There is a cluster head for inter-cluster distributed and intra-cluster centralized execution in each cluster. In such a way, mixed schemes can perform the intra-cluster optimal calculation and independently carry out the inter-cluster resource allocation decision. On the other hand, the mean-field theory has received significant attention due to its ability to study large scale and interacting individuals' behavior through a simpler model. In these simplified models, the effect of all the other individuals on any given individual is approximated by a single averaged result, thus reducing the complexity.

Given the importance of interference coordination in UDN, then it is crucial to design RA strategies that: i) make use of all possible interference coordination domains, performing close to optimal CA, ii) make use of semi-distributed approaches, iii) use novel approaches that reduce problem complexity without constraining problem generalization, such as mean-field theory.

1.3 Solving Channel Allocation with Metaheuristics Algorithms

Unlike exact algorithms, which find the optimal solution of an optimization problem, metaheuristic algorithms offer no guarantee of finding an optimal solution. Still, these return an approximate solution (sub-optimal) for the problem. Metaheuristic algorithms also use stochastic search methods, usually free of derivates, intending to avoid premature convergence, escaping local optima and aiming for global optimization. Given this non-deterministic pattern, the algorithm's solutions at each execution of the algorithm travel for different paths, and the algorithm offers another final solution. This characteristic makes it non-trivial to establish a stop criterion that determines the end of the search process. Hence, these algorithms are limited to execute a finite number of fitness function evaluations. The computational resources (execution time) invested in solving an optimization problem are defined by the computational complexity of evaluating the fitness function multiplied by the maximum number of evaluations used [52]. This number of evaluations usually increases with the optimization problem complexity. Therefore, the increase in the number of fitness function evaluations is linearly bounded by the number of the algorithm's iteration, given a fixed population size..

1.3.1 Key Challenges and Problem Statement

The CA algorithm's objective is to determine the best allocation of the available channels between the network's active users to maximize the capacity of the UDN while fulfilling several restrictions, such as the QoS demands. Metaheuristic algorithms have proven to provide a proper balance between quality of solutions and computational efficiency, offering close to optimal solutions in a fraction of the time required for classical optimization techniques [47]. Population-based metaheuristic algorithms work by generating M initial random solutions. They evolve iteratively through a variety of mechanisms, usually inspired by natural phenomena. For the CA problem, these algorithms produce $X_{U\times N}$ binary matrices as the CA solutions for the network, evaluate the solutions' "quality" by calculating the network capacity and restrictions, and recombining the gathered information iteratively to explore the search space and exploit the neighborhood of promising solutions. However, there are two main issues when metaheuristic methods are applied to solve the CA problem in UDN:

1. The complexity of the optimization problem increases exponentially with the number of users, cells, and channels in the network.

2. Although cell densification increases the network capacity, co-channel in-

terference also increases, making it difficult to find an efficient channel allocation between shared channels.

1.3.2 Search Space and Dimensionality of the CA Problem

Given the CA problem's NP-hard nature, solving it will require the enumeration of all possible solutions, also known as brute force. As the complexity of a problem is not greater than the complexity of any algorithm that solves it, this work uses the complexity of brute force, i.e., the size of the search space, as the CA problem's bounded complexity. The exponential complexity of the binary allocation matrix consist of $2^{Users \cdot Channels}$ possible allocations for the network. This exponential growth of the search space is commonly known as "the curse of dimensionality," a term coined by Richard E. Bellman [53]. The main complication of this phenomenon is that the maximum number of fitness function evaluations (i.e., computational resources invested in the optimization process) can not follow the exponential rate, as exponential response times are unpractical [54]. Therefore, the divergence between these growth rates as the network scales produces increasingly inefficient solutions, constraining the network densification benefits. Furthermore, this rapidly expanding search space contains a high number of infeasible and unsuitable solutions. Infeasible solutions, those in which a cell uses one channel twice, can be adjusted by having these cells attend only the first user allocated in the channel, ignoring repetitive allocations. However, this method only transforms these infeasible solutions into redundant ones, which could mislead the optimization task. On the other hand, unsuitable solutions differ from the users' QoS requirements and poorly coordinate interfering nodes. Allocating too few channels to a user makes it impossible to fulfill their needs, while assigning too many may result in unnecessary interference for other users in the network, thereby degrading its capacity.

Several metaheuristic algorithms have been successfully used as strategies to solve the RA problem in HetNets [31]–[33], [47], [55]–[58]. However, the number of computational resources required to solve the CA problem with these previously proposed metaheuristic algorithms, while exploiting the advantages of the network densification, becomes impractical in UDN.

1.3.3 Co-channel interference management

Apart from the high complexity involved in solving the CA problem, the increasing interference experienced in UDN is a critical factor to consider since it diminishes the network densification benefits [28]. At first, each new SC added to the UDN results in improved capacity, as long as some users are offloaded from the MC to this SCB [59]. This capacity improvement comes from reusing the bandwidth in the SC, which increases the number of channels per user in the network and better signal propagation conditions between the SC and UEs, which allows fulfilling users' rate demands with fewer resources. However, the channel reuse comes at the cost of an increased interference level in the network, which opposes the improved signal propagation conditions' benefits. The interference grows along with the densification, while the probability that several users will be associated with the added cells is inversely proportional to it.

For this reason, with increasing densification, the network converges to a theoretical limit in which each cell communicates with an average of one single user [35]. The densification itself reduces the main benefits of network densification. The only way to further improve wireless network capacity depends on the RA algorithm's capability to efficiently coordinate the interference nodes. Besides, more significant densification means more unsuitable solutions are found in the search space due to the interference produced by sharing spectrum resources. These non-feasible solutions are those in which intra-cell interference results in capacity degradation, increasing the error surface's complexity. Thus, the probability of finding solutions with efficient interference coordination diminishes greatly with increasing densification.

1.4 Research Areas in UDN

It is agreed that densification is a fundamental part of reaching 5G and B5G goals. It is a straightforward mechanism to boost network capacity. Nonetheless, as mentioned before, there are still several research problems in UDN at high cell densification. The following list presents those of interest for this thesis document.

- The study of network densification limits. Current approaches do not consider the use of interference mitigation techniques, or lack the evaluation of detailed aspects of the UDN, due to the requirement of extensive simulation.
- Several of current UDN published approaches that aim to reduce the RA problem complexity, focus on the PA complexity, but CA is rarely considered and requires further experimentation.
- Given the extensive complexity of the CA problem in UDN, the application of close-to-optimal techniques to solve CA, such as metaheuristic approaches, remain stagnated.

1.5 Objectives and Thesis Scope

This thesis investigates the proposal of efficient sub-optimal strategies for the Resource Allocation problem in UDN deployments for B5G network communications. Specifically, methods based on reducing interference and the problem complexity to maximize the network's data rate while providing users QoS. This primary goal can be divided into the following partial objectives listed below.

- To identify the effect that the channel allocation process can achieve on reducing the interference in UDN with spectrum sharing when using close-to-optimal strategies.
- To reduce the CA problem's complexity in UDN to improve metaheuristic algorithms' performance to solve it efficiently.
- To determine the limits of densification and the benefits achieved in the capacity of UDN when using the block-based channel allocation schemes.

1.5.1 Main Contributions

The main contributions of this thesis are summarized as the followings:

i. Two novel CA allocation schemes are proposed that represent the allocation matrix in a compressed fashion, reducing the dimensionality and search space of the problem.

ii. The problem's complexity using the proposed schemes is evaluated using random allocation and a set of novel and well known metaheuristic algorithms to study their benefits on improving sub-optimal solvers.

iii. Using the proposed schemes and the traditional approach used in HetNets, the individual user capacity and users' outage rate are studied regarding the network densification.

iv. The proposed schemes with the metaheuristic algorithm that consistently offers the best solutions are studied in detail compared to state of the art approach.

1.6 Thesis Outline

This thesis is organized into seven chapters as follows:

- **Chapter 2** provides a state of the art for the Resource Allocation problem in HetNets and UDN.
- **Chapter 3** presents the Ultra-Dense network and Resource Allocation mathematical model used for this work.
- **Chapter 4** introduces a population based metaheuristic algorithms and the set of metaheuristics considered in this document to evaluate their performance to solve the Channel Allocation problem.
- **Chapter 5** presents the proposed Channel Allocation schemes and their computational analysis.
- **Chapter 6** evaluates the achieved performance improvement on the solution offered by the metaheuristics, and a validation and comparison of our proposal with a similar strategy recently published.
- **Chapter 7** presents the overall and specific conclusions reached in this thesis, and future ramification of research.

Chapter 2

Literature Review

This chapter provides a review of the literature, focusing on previous works that make use of PA or CA to solve the RA efficiently with non-deterministic techniques on HetNets, and the challenges that these techniques face to scale well on dense cell deployments.

2.1 Resource allocation in low cell densities

Several metaheuristic algorithms to solve the RA optimization problem in HetNets are reported. These show the importance of dealing with the high complexity involved, even for sparse SC deployments. For example, in [33], the authors maximize the HetNet capacity and the number of secondary links successfully allocated to users. The channel allocation problem is modeled as a multi-objective optimization problem and solved by the NSGA-II algorithm. In this case, the vector solution's dimensionality is relatively high, corresponding to the number of users plus the number of available channels in the system. Another approach reported in [31] deals with the high complexity of the channel allocation problem by preprocessing the availability of channels for each secondary user in terms of interference. This approach results in a vector solution with reduced dimensionality since it only allocates those channels within the appropriate interference limits with the primary users. In [31], the authors apply a Graph Coloring Technique to optimize the resulting channel allocation problem. Other works use the preprocessing technique and subsequently solve the problem with either the Genetic Algorithm (GA), Particle Swarm Optimization (PSO) [32], Ant Colony Optimization (ACO) [55] or a multi-objective Quantum PSO [56], producing results rivaling those obtained in [31]. However, these approaches are limited to allocating a single channel to each SCB user. In [47], a multi-objective approach compares the application of three heuristics to solve CA in HetNets: Convex Relaxation with Tree Pruning (CRTP), Convex Relaxation with Gradual Removal (CRGR), and GA. The vector solution dimensionality for the GA approach is equal to the number of users times the number of channels. The results show that even for a small number of nodes, users, and channels, the problem's computational burden to find feasible solutions for all the optimization techniques results in high response times. On the other hand, a spectrum partitioned model aims to maximize the system throughput by the joint optimization of user association, and RA is proposed in [57]. This work compares the proposed problem approach's performance when solved by Mixed Integer Linear Programming (MILP), Weighted Water Filling (WWF), and GA. A similar work solves the spectrum partition CA with PSO [58]. Although the spectrum partition strategy reduces the complexity of channel allocation by avoiding interference between the MC and SCBs, it also limits the potential increase in capacity due to the restricted spectrum available for the SCBs. These works show that metaheuristic algorithms offer a good trade-off between solution quality and computational effort, compared to heuristic and classical optimization methods. However, to the best of our knowledge, applying metaheuristic algorithms for solving the CA problem in UDN has not been reported.

Although metaheuristic algorithms are successfully used to solve the RA for low cell densification scenarios, as shown in Tables 2.1 and 2.2, the RA problem becomes unattainable for UDN [42]. First, because the problem complexity increases exponentially based on the number of active users, channels, and SC in the network. Also, although each cell deployed in the network enhances resource availability and network capacity, it causes an increase in co-channel interference, making it difficult to find an efficient allocation that maximizes the network capacity and spectral efficiency. Due to the high dimensionality, extensive search space, and aggressive frequency reuse in UDN, the RA optimization problem quickly becomes too complicated to be solved efficiently with approximation algorithms [60]. However, the application of metaheuristics still seems somewhat appropriate to solve the RA open issues in UDN, given their capabilities to cleverly explore the problem search space, maintaining "good quality" solutions, and avoiding premature convergence in local optima. In this sense, new strategies are needed to reduce the adverse effects mentioned above and find feasible and efficient solutions.

2.2 Resource allocation in Ultra-Dense Scenarios

Given that new strategies are required to solve the RA problem in UDN, several CA and PA approaches have been proposed. For example, the works reported in [60] and [61] solve the PA problem by applying game theory methods. However, the CA stage is tackled using either a greedy algorithm [60], random allocations

Table 2.1: Related work summary of metaheuristic algorithms applied in Non-dense scenarios.

Reference Work	Network Model	Algorithm	Objectives	One-to-One	Single-Channel	Co-Channel	Low-Complexity Strategy
31	Cognitive Radio	Graph Coloring	Fair Network Capacity	True	False	True	Network Conflict Graph
67	Cognitive Radio	Quantum Genetic Algorithm	Fair Network Capacity	True	False	rue	Network Conflict Graph + Encoding
46	Cognitive Radio	Ant Colony Optimization	Joint Primary and Secondary Capacity	One-to-One	True	With Primary	Network Conflict Graph
56	Cognitive Radio	NS-QPSO	Max-Sum, Max-Min-Fair, Proportional-Fair	One-to-One	False	True	Network Conflict Graph + Encoding
57	Spectrum Partition	Genetic Algorithm	Network Capacity with QoS	False	False	False	Approximated Methods
58	Spectrum Partition	PSO	Network Capacity with QoS	False	False	False	Approximated Methods
33	Cognitive Radio	NSGA-II	Network Capacity, Number of Secondary Links	True	True	True	None
48	Cognitive Radio	GA	Fair Network Capacity	True	False	True	Metaheuristics

[61], or considering a fixed number of channels per user as in [62]. These CA approaches offer a quick solution, but network interference and users' QoS are not considered. Like in [63], other methods solve the RA problem for a cognitive network, jointly allocating channel and power using Lagrange dual decomposition to find a sub-optimal solution. However, it presents a high computational complexity, in which response time grows exponentially with the network densification, losing practicality in UDN [48].

On the other hand, clustering techniques can considerably reduce the RA problem's computational complexity [64], [65], since RA is performed in small independent groups of SC instead of the whole network. In [42], the authors propose to form clusters of SC in a centralized manner and realize the CA process in each cluster independently. They consider that interference between groups of SC should be low enough to be neglected. A recent work reported in [50] proposes decomposing the RA problem into inter-cell and intra-cell layers and further decomposing inter-cell sub-problem in other CA and PA sub-problems. The authors use the average channel gains in the inter-cell interference calculation to obtain an approximate result and reduce the CA and PA complexity. The resources are allocated directly to the cells. Lastly, a CA interference management for UDN is proposed [66]. A centralized user-centric merge-and-split coalition formation game, along with a graph-based resource allocation algorithm, is applied. A supplementary CA algorithm assigns the rest subchannels to each cell. The main contribution is to group users who may generate substantial interference with each other into the same coalition and allocating them orthogonal subchannels as far as possible. The results show an increment in the spectral efficiency given by reducing the CA strategy's computational complexity. This reduction is achieved by distributing the available spectrum in a lower number of channels. However, using wide bandwidth channels increases interference, as fewer channels are available to allocate SC orthogonally, leading to excessive channel allocation to specific users when QoS is considered.

Compared to most previous approaches applied to the RA problem on sparse HetNets, recent works in UDN implement a greedy and straightforward CA algorithm [43], [49]. These select the channel that offers maximum gain for each user, referred to as multi-user gain, without modeling intra-cell or inter-cell interference completely [50]. However, other studies report that the improvement achieved by this strategy is reduced when applied in UDN because the small number of users per cell and the short distance between SC and UE increases the probability of Line-of-Sight (LoS). Hence, all the available channels will theoretically have a maximum gain [51]. Consequently, the allocation channel strategy applied for UDN should reduce the interference instead of exploiting the temporary conditions of the propagation environment. This situation would deteriorate the capacity of the UDN architecture [27]. For this reason, in these networks, path-loss should be assumed to be the dominant, and only, loss factor [47]. Using this model, the interference created by multiple cells sharing the same channel becomes the critical factor affecting each link's capacity in the network.

Table 2.2: Related work summary of metaheuristic algorithms applied in Dense scenarios.

Work	Network Model	Algorithm	Objectives	One-to-One	Single-Channel	Co-Channel	Low-Complexity Strategy
65	UDN	Min-cut clustering	Energy Efficiency with QoS	False	False	True	semi-centralized Clustering
60	UDN	Greedy CA	Energy Efficiency	One to Two	False	True	Stackelberg Game for PA
61	UDN	TORA and TCRA (Game Theory)	Network Capacity with QoS	False	True	True	Distributed PA
63	UDN	LaGrange Dual Decomposition	Network Capacity	False	False	True	None
64	UDN	Heuristic Clustering	Network Capacity	True	False	True	Clustering
42	UDN	K-Mean Clustering and Semi-Greddy CA	One to Two	False		Inside Clusters	Clustering and Heuristic CA
66	UDN	Coalition Game Theory	Network Capacity	Fix-to-One	True	True	Wide Channels
50	UDN	Layered RA	Network Capacity	yes	False	True	Multiple Layered CA Heuristics

2.3 Conclusion

These recent approaches for solving the CA for UDN, present at least one of the following issues:

- Do not consider users' QoS leading to unfair resource distributions and user data rate starvation.
- Consider that the bandwidth is distributed in a small number of channels, leading to inefficient use of the resources by allocating excessive bandwidth to users and increasing interference.
- Focused on exploiting the temporary conditions of the propagation environment instead of avoiding network interference.
- Consider a fixed number of users associated with each cell, exploring the network and user densification increase together.

However, the mechanisms proposed to reduce the CA optimization problem's complexity can be included in metaheuristic algorithms through a novel approach. For example, distributing the available spectrum in a smaller number of channels reduces the CA problem's search space and dimensionality but at the cost of increasing interference, as reported in [66]. However, suppose the users' QoS requirements are considered. In that case, if an interference threshold is set, it is possible to determine the number of channels required to reach users' demands, to bound the bandwidth allocated to them [57]. We call this approach reported in the literature as Individual Channel Allocation (ICA). ICA is one of the most used CA methods applied for low cell dense scenarios [31], [46], [56], [57], [67]. However, by applying ICA, the optimization problem's dimensionality grows exponentially with the network scale on the number of users, users' demands, and SC, which complicates the finding of CA's feasible solutions in

UDN.

This thesis aims to reduce the RA problem's dimensionality by allocating each user's entire bandwidth in a single block of channel. The solutions obtained will also enhance the QoS metrics of the UDN, such as throughput and spectral efficiency (SE). Unlike reported in [66], this approach avoids the aggregated interference by allocating channels with specific bandwidths regarding users' demands. We call this approach User Block Allocation (UBA). A layered RA in which a centralized entity distributes channels between cells minimizing overall network interference, is also implemented. Then cells autonomously allocate their available resources to their users, similarly to [50] or cluster approaches [42]. We call this proposed strategy Cell Block Allocation (CBA). This CBA approach is practical due to the small coverage area of SC, which results in a statistically equivalent level of interference perceived by each of SC's users. Therefore, assigning channels to users in UDN becomes the issue of assigning non-orthogonal spectral portions to small-cell. It reduces even more the dimensionality and search space of the optimization problem. This dimensionality reduction to the RA problem benefits the metaheuristic algorithms, such as the GA, since it would allow them to find high-quality solutions.

Chapter 3

Resource Allocation Model for Ultra-Dense Networks

3.1 Introduction

In this thesis a mathematical model on Interference Management for UDN scenarios is proposed. In UDN HetNets, the specific research directions are Interference Management, Cell Association, Offloading, Cell Range Expansion, and Trade-off between spectral efficiency (SE) and energy efficiency (EE). The proposed network model structures are based on achieving the maximum spectral efficiency (SE) and throughput. It provides a broad and straightforward scenario that can be augmented to study other aspects of B5G technologies such as Massive MIMO and Millimeter-Wave Networks.

Three main components are necessary to consider modeling the RA problem in wireless network:

- the number of cells (Q),

- the number of active users (U),
- the bandwidth distributed over N identical channels (B).

By considering these components, the capacity of the network given any RA solution (a binary $X_{Q \times U \times N}$ channel allocation (CA) and real $P_{Q \times N}$ power allocation (PA)) can be estimated using Shannon's equation as follows:

$$C^{Network}(X_{Q,U,N}, P_{Q,N}) = \sum_{q=1}^{Q} \sum_{u=1}^{U} \sum_{n=1}^{N} \frac{B}{N} log_2(1 + \underbrace{\frac{X_{q,u,n} P_{q,n} H(q,u,n)}{N_0 + I(q,u,n)}}_{\text{SINR}}) \tag{3.1}$$

where $H(q,u,n)$ and $I(q,u,n)$ are functions that calculate the channel gain and interference of the network for each link used, respectively. As denoted, these functions depend on which cell, user, and channel are being used for the communication and which other links are using the same channel.

These three components can be modeled in several ways, depending on which aspects of the network are of interest for the study. For example, for unplanned scenarios, the cells and users' positions can be modeled as a random placement, as it is expected to be the user deployments of small cells. Or the cells' positions can be part of the optimization of the network when a planned layout of cells is studied.

Cells can be either homogeneous or heterogeneous in terms of coverage, which is dependent on the maximum transmission power available P_q and the minimum SNR permissible ($Limit\_sinr$). A mix of different size cells composes an HetNet, in which usually small cells are deployed inside an MC coverage. This deployment of small cells can be performed by operators using planning strategies or as Plug-and-Play by users when and where required. A small number of these cells is usually known as sparse HetNets (or simply HetNets), while UDN is known to concentrate a high number of small cells. If the small cells are intended to co-use the same B portion with the MC, such deployment is called underlay or Sharing Spectrum. Otherwise, Spectrum Partition or overlay deployment refers to cells of different sizes separated in layouts with a specific orthogonal portion of the available B. These cells can also be configured to three common access control modes: Open, Closed and Hybrid. These access modes affect how the users and the cells are associated. Open access means that any nearby user can be associated with the cells, while closed cells only serve subscribed users. Some research on UDN drops the heterogeneous layout and analyzes homogeneous small cells deployments [35].

Users have two main characteristics: position and demands. Most works on UDN consider that each cell in the network always has a set of users associated with them. It follows the idea that new small cells are deployed in areas where these are needed. In this case, the network capacity demands grow with the densification, but other studies recommend that these two parameters be studied apart [35]. On the other hand, when users appear random over the coverage area, an association process pairs the users and cells. Random user placement allows controlling the users' and cells' densities individually. The users' demands on quality of service (QoS) affect the resources that the network requires to allocate to each user. Usually, these demands are measured in data rate or latency depending on the study evaluation. In RA works, the individual users' data rate demands depend on the application's data rate they require; for example, streaming media applications require bursts of high-speed data rates, while a sensors network might require constant but low-speed connections. However, this restriction can make the problem unsolvable if no solution can provide these demands to all users. Therefore, this is usually considered a soft-restriction, which is desirable to find solutions that fulfill most of the users' data rate requirements. This constrain is included as an objective of the optimization problem when multiple objectives are considered, as a penalty function in single-objective optimization problems, or as some sorting and fairness mechanism in heuristics approaches, such as in Rhee's algorithms [68]. Other works do not consider individual users' demands. These usually study more general aspects of the network, yet it is known that this might lead to unfairness and service starvation.

Finally, bandwidth is one of the primary resources of the network. It is usually divided by a fixed number N of Resources Blocks, composed of a certain number of narrowband channels (12 subcarriers of 15 KHz for OFDM). Although these Resources Blocks are distributed to users, instead of the individual subcarriers, it is common to refer to these resource blocks as channels, abstracting the specifics details of the signal modulation. A higher capacity can be achieved for each communication link by using wider channel bandwidth; this means a lower number of channels. As seen in Eq. 3.1, the capacity of each link is inversely proportional to N. This option also reduces the complication of the RA optimization problem, as a lower number of channels are available to allocate, but not necessary for improving network performance. However, this means that the network is restricted in channels to attend several users simultaneously. Also, using a low number of channels with wide bandwidth could produce an excessive bandwidth allocation to users, which achieves higher QoS than required limiting the available resources to others. On other hand, using narrowband channels allows a granulated allocation but increases the RA's complexity, by increasing the optimization problem's search space and dimensionality.

Another critical factor to consider is the channel model used to calculate the channel gain. The channel gain model includes several physical aspects that fade the wireless transmission signals due to the signal's travel through space. Two main categories are stochastic fading and path loss, expressed as follows:

$$H_{q,u,n} = \underbrace{\phi_{q,u,n}}_{Stochastic} \cdot \underbrace{D_{q,u}^{-\alpha}}_{PathLoss} \tag{3.2}$$

Physical phenomena like multipath fading, diffraction, reflections, or Doppler effect are affected by the channel's frequency and bandwidth, and a probabilistic distribution usually characterizes them. Then, using these probabilistic distributions, one can evaluate these factors by using random variables. On the other hand, the path loss is commonly evaluated using the inverse of the distance between the base station and UE's antennas to the power of a path loss exponent (α), independent of the channel. Then, suppose only path loss is considered in the calculation. In this case, i.e., $H_{q,u} = D_{q,u}^{-\alpha}$, only the interference is affected by the channel index n, as it depends on the allocation of other cells using the same channel.

This work is focused on Interference Management in Ultra-Dense HetNet scenarios. The network model's physical structures are chosen with the intention of providing a general scenario. Nonetheless, the proposed model can be adapted to study other aspects of Ultra-Dense HetNet, or different UDN scenarios such as Massive MIMO and Millimeter-Wave Networks.

3.2 Proposed System Model for UDN

For the rest of this work, the UDN model consists of a HetNet with Q cells, a single Macro-Cell (MC) and $Q-1$ Small-Cells (SC), and U active users over a coverage area of 200m x 200m. The MC is located in this area's center. The SC are randomly deployed at a minimum distance D_{SC}^{min} between them. The users' positions are set randomly over the space, with the minimal distance $D_{Q,U}^{min}$ between these and any given cell applied in the capacity calculation. Each cell is limited by a maximum power budget P_q^{max}, and can share the available bandwidth B, which is uniformly divided into N channels. The users demand a data rate R_u, which depends on the mobile application being used, e.g., browsing the web requires a much lower data rate than consuming a video stream.

Without loss of generality, this work focuses on downlink communication and considers that each UE and cell base station is equipped with a single transmitting antenna and receiving antenna, respectively. UEs are served by one cell at a time chosen based on the highest signal strength users receive, known as Path Loss Association [69]. Each cell is associated with a subset of active users in the network U_q, in such a way that $\bigcup_{q=1}^{Q} U_q = U$ and $\bigcap_{q=1}^{Q} U_q = \emptyset$. The RA algorithm is executed in each time slot, considering that the surrounding conditions, such as user locations and channel propagation characteristics, are known by the network and remain static during the time it takes to perform this task [31].

The total estimated user data rate r_u can be calculated using the capacity function proposed by Shannon, corresponding to the data rate sum of the individual channels that the associated cell q allocates to the user as follows:

$$r_{q,u} = \sum_{n=1}^{N} \frac{B}{N} \, log_2(1 + U_{q,u} X_{u,n} SINR_{q,u,n}) \tag{3.3}$$

where $X_{u,n}$ is the binary channel allocation matrix that determines whether channel n is assigned to user u, and $U_{q,u}$ is the association binary variable of value $1 \leftrightarrow u \in U_q$.

The achievable $SINR$ is determined by Eq. (3.4).

$$SINR_{(q,u,n)} = \frac{P_{q,n} D_{q,u}^{-\alpha}}{N_0 + I_{q,u,n}} \tag{3.4}$$

where $P_{q,n}$ is the power that cell q assigns to user u in channel n, $D_{q,u}^{-\alpha}$ is the path loss fading due to the distance between the user u and the base station in cell q, and α is the path loss exponent. N_0 is the noise level floor, which is considered constant across all channels, and $I_{q,u,n}$ is the interference generated by other cells $\dot{q} \neq q$ using the same channel n.

This binary channel allocation matrix determines the level of interference perceived by users, which is calculated as the sum of the received signal power in one channel minus the desired signal, as shown in Eq. (3.5).

$$I_{q,u,n} = -P_{q,n} D_{q,u}^{-\alpha} + \sum_{q^*=1}^{Q} \sum_{u^*=1}^{U} U_{q^*,u} X_{u^*,n} P_{q^*,n} D_{q^*,u}^{-\alpha} \tag{3.5}$$

To make the above clear, a scenario of how interference influences the channel allocation process in a UDN will be described below. Figure 3.1, shows a single MC ($q1$), two SC ($q2, q3$), and four active users ($u1, u2, u3, u4$). Cells can share the available bandwidth B, which is uniformly divided into N channels (i.e., n_1

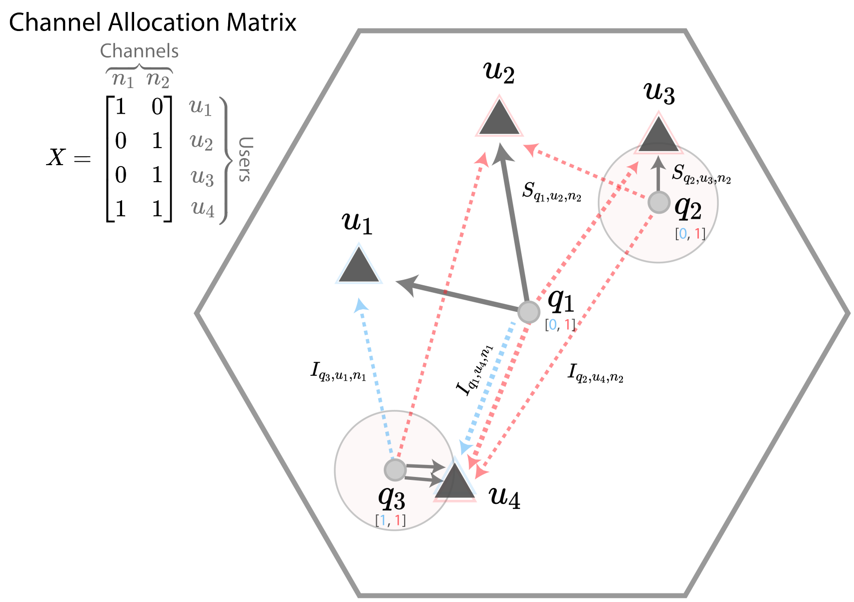


Figure 3.1: Co-existence scenario of an heterogeneous UDN: a single MC (q_1) and two SC (q_2, q_3), four active users (u_1, u_2, u_3, u_4) and two channel (n_1, n_2).

and n_2). The objective of solving the CA problem in UDN is to find a binary allocation matrix X_{UxN}, which indicates whether the channel n is assigned to user u with a value of 1 on such indexes ($X_{u,n}$). The CA matrix example shown in the figure indicates that the MC q_1 uses channels n_1 and n_2 to serve users u_1 and u_2, respectively. The SC q_2 uses only channel n_2 for its user u_3, while q_3 makes use of both channels. In the figure, the solid arrow lines represent the desired signal between cells and their associated users (i.e., $S_{q_1,u_2,,n_2}$ is the signal that the MC transmits to user u_2 over the channel n_2). In contrast, the dotted arrows represent the interference generated between shared channels. The interferences I_{q_3,u_1,n_1} and I_{q_1,u_4,n_1}, that users u_1 and u_4 perceive on the first channel from the SC q_3 and MC q_1 respectively, is known as cross-tier interference. On the other hand, users u_4 also perceive the inter-tier interference I_{q_2,u_4,n_2} of the SC q_2 in the second channel, which is added to the cross-tier interference from the MC.

The RA problem is expressed as a mixed restricted optimization problem that seeks to maximize overall system throughput while satisfying user requirements, as follows:

$$\max_{X,P} C = \sum_{q=1}^{Q}\sum_{u=1}^{U}\sum_{n=1}^{N}\frac{B}{N}log_2(1+U_{q,u}X_{u,n}SINR_{q,u,n}) \tag{3.6}$$

subject to

$$\sum_{u=1}^{U_q} x_{u,n} \leq 1,\ \forall\ (q,n) \tag{3.7}$$

$$x_{u,n}\ SINR_{q,u,n} > \sigma,\ \forall x_{u,n} = 1 \tag{3.8}$$

$$\sum_{n=1}^{N} P_{q,n} < P_q^{\max},\ \forall\ q \tag{3.9}$$

$$r_u \geq R_u,\ \forall\ u \tag{3.10}$$

Restriction (3.7) ensures that each cell serves at most one UE per channel, as this has been proven to be optimal [70] when the power is allocated over the channels with the highest gains. Meanwhile, (3.8) ensures that a minimum $SINR$ is reached over all the assigned channels. Otherwise, the bit error rate (BER) would prohibit communication. In (3.9), each cell's distributed power is limited by the maximum power available for the cell $P_q^{\max}$. Finally, constraint (3.10) ensures that each user achieves his QoS data rate requirement.

The optimization problem presented in Eq. (3.6) is non-convex, with mixed variables and NP-Hard as proven in [31]. It consists of two decision variables: a $Q \times U$ matrix representing the continuous power allocation $P_{q,n}$ and an $N \times U$ matrix for binary channel allocation $X_{u,n}$. Therefore, as the network scales and densifies, i.e., the number of users and cells increases, the RA algorithm becomes a puzzling problem to study. Usually, expression (3.6) is divided into two stages: (i) Channel Allocation (CA), and (ii) Power Allocation (PA), as reported in [36]. This strategy allows for solving the CA problem using a flat power distribution, which fulfills restriction (3.9). Then PA maximizes the performance of the network by selecting optimal power allocation over the chosen channels.

This thesis focuses on solving the CA problem by coordinating the spectral portion for each link between the base stations and UEs, attempting to reduce the overall interference of the network, achieving the highest network capacity and meeting QoS requirements for each user. It is expressed as follows:

$$\underset{X}{\operatorname{argmax}}\, C = \sum_{q=1}^{Q} \sum_{u=1}^{U} \sum_{n=1}^{N} \frac{B}{N}\, log_2 \left(1 + x_{q,u,n} \frac{P_q^{\max} D_{q,u}^{-3}}{N(N_0 + I_{u,n})} \right) \tag{3.11}$$

The main problem consists of assigning each user a set of channels that meet restrictions (3.7) and (3.10). The constrain (3.10) can make it impossible to find a feasible solution, thus it is considered a soft restriction. The CA problem cannot be solved optimally by exact methods in polynomial time as finding the global optima would require enumerating all possibilities.

3.3 Channel Allocation Solution Representation

The objective of solving the Channel Allocation problem (CA) in UDN is to find a binary allocation matrix $X_{U\times N}$ (as shown in Fig. 3.1) that maximizes network capacity (C) while meeting the QoS demands of users (c_u). However, finding highly efficient channel allocations, such as the optimal, requires an impractical computational time when exact methods are applied, given the problem's NP-Hard nature and the extensive search space ($2^{U\cdot N}$). On the other hand, greedy heuristics offer quick solutions but at a significantly lower resource use efficiency. Metaheuristic algorithms have proven to provide the right trade-off between quality solutions and computational efficiency, offering close to optimal solutions in a fraction of the time required for deterministic algorithms [47].

As it is known for metaheuristic algorithms, the computational complexity is carried by the number of fitness function calls (FFC) [71]. This FFC is calculated by multiplying the population's size and the number of iterations the algorithm evolves. In the proposed application, the computation complexity of fitness functions is equal to the complexity of calculating the capacity of each allocated channel. Therefore, for each channel in the network, the capacity of the users using it is calculated. As it is an unfeasible and inefficient solution when all the users share one channel, the worst-case allocation requires $\mathcal{O}(NQ^2)$ capacity calculations. This worst-case solution is such that each cell makes use of all the available channels.

Nevertheless, the worst-case solution is inefficient because interference will vastly degrade the signal of all the users. Therefore, the average complexity of the fitness function is $\mathcal{O}(NQlog(Q))$. This complexity calculation considers that the interference that each user receives from all the cells ($\mathcal{O}(UQ)$) is calculated before the beginning of the evolutionary process and stored in memory. Therefore, our proposal's total average complexity is $\mathcal{O}(FFC \cdot NQlog(Q) + UQ)$.

Population-based metaheuristic algorithms work by generating M initial complete solutions at random, being this the population $\vec{x}_i \in X \mid i \in [1, 2, ..., M]$, that evolve iteratively through the algorithm's operators. Therefore, a vector representation of the problem's solution should be designed to generate and manipulate these solutions. In the following sections, we will consider the worst-case scenario for the CA's solution representation (naive solution representation). We compared this naive representation with the Individual Channel Allocation (traditional approach) that uses the user's data rate or network's channel availability to limit the channels allocated to the users. Each solution's fitness value is measured through a devised function, that takes into account the optimization objective and the restrictions. For the CA optimization problem, to guarantee individual data rate requirements and maximize overall network capacity, the fitness function is proposed as follows:

$$f(x) = \frac{\sum_{u=1}^{U} r_u}{1 + \sum_{u=1}^{U} \tilde{R_u}} \tag{3.12}$$

$$\tilde{R_u} = \begin{cases} 1 & \text{if } r_u < R_u \\ 0 & \text{otherwise} \end{cases} \tag{3.13}$$

where in Eq. 3.12 $\sum_{u=1}^{U} \tilde{R_u}$ denotes the unsatisfied number of users, i.e., those who did not reach their data rate requirements. In this way, the user QoS demand restriction translates into a penalty operator so that the solution's fitness value is inversely proportional to the number of unsatisfied users.

3.3.1 Naive Solution Representation

A naive implementation will require to flatten the $X_{U \times N}$ matrix into a vector with dimensionality of $D = U \cdot N$, on which each solution of the population $\vec{x}$ represents a binary channel allocation [46].

$$\vec{x} = [\underbrace{x_1, x_2, \ldots, x_N}_{u_1}, \underbrace{x_{N+1}, \ldots, x_{2N}}_{u_2}, \underbrace{\ldots, x_D}_{u_{3,\ldots,U}}] \mid x_d \in [0, 1] \tag{3.14}$$

where x_d indicates if corresponding pair user u and channel n is assigned or not. Given this arrangement, a random solution can be generated as follows:

$$\vec{x}_d = rand_{int}(0, 1) \; \forall \; d \tag{3.15}$$

This vector (3.15) can easily be arranged into the original shape of $X_{U \times N}$, and evaluated into the network model to calculate each user's capacity. However, the binary allocation matrix $X_{u,n}$ consists of $U \times N$ dimensions and a total of 2^{UN} possible allocations for the network. Therefore, the search space grows exponentially with the network scale. This exponential growth of the search space regarding the increased dimensionality is commonly known as "the curse of dimensionality", a term coined by Richard E. Bellman [53]. The complexity is that the number of trial solutions to solve the optimization problem can only increase linearly [54] due to exponential response times are unpractical. Therefore, the divergence between these growth rates as the network scales produces increasingly inefficient solutions, constraining the network densification's benefits.

This naive implementation is similar to the greedy heuristics that allocates each channel to the users with the higher temporary gain conditions, exploiting the multi-user gain. This allocation is based on random distributions that do not consider the interference between sharing nodes. In this strategy, the network's cells distribute the whole range of channels between their users. Therefore, each user is allocated a large number of channels. It produces a heavy reutilization of the spectrum, but it generates uncontrolled interference between all the network's nodes.

This naive implementation does not consider that the search space contains many infeasible and unsuitable solutions, those conflicting with restriction (3.7), nor those assigned with too few or many channels to a user. In this implementation, a channel can be allocated to users associated with the same cell, representing a high interference between them and double the calculated interference that others using this same channel perceive. The unsuitable solutions are those channel allocations that differ from the users' QoS requirements and poorly coordinate interfering nodes. These allocate either too few or too many channels to a user, making it impossible to fulfill their needs, or result in unnecessary interference for other users in the network, thereby degrading its capacity. Given that this two situations reduce the resultant network capacity, the optimization algorithms naturally avoid these. However, this complicates the search space surface, making it more likely that metaheuristic algorithms converge to suboptimal local optima.

3.3.2 Individual Channel Allocation (ICA) Scheme

One known and effective method to alleviate the optimization problem complexity is to propose a solution representation that reduces the dimensionality and the search space discussed above by controlling the number of channels to allocate to each user. This strategy calculates the number of required channels for each user based on their demands and surrounding conditions. It relieves the optimization algorithm of how many channels are allocated to each user and focuses on assigning a fixed number of channels. Also, it lowers the complexity of the CA optimization problem in terms of dimensionality and search space, eliminating the previously mention unsuitable solutions. We refer to this strategy as Individual Channel Allocation (ICA), which can be found in some previous works on CA for HetNets [31], [32], [55]–[57].

The CA algorithm is previously calculating the number of channels to allocate to each user in UDN by considering an interference threshold I_t. This interference threshold controls the upper limit of the expected interference over the allocated and shared channels. Therefore, the number of required channels N_u to meet the user data rate R_u can be estimated from Eq. (3.17) and (3.16).

$$N_u = \left\lceil \frac{R_u}{\hat{r}_{q,u}} \right\rceil \tag{3.16}$$

$$\hat{r}_{q,u} = \frac{B}{N} \log_2 (1 + \frac{P_q^{\max} D_{q,u}^{-\alpha}}{N N_0 + I_t}) \tag{3.17}$$

where $\hat{r}_{q,u}$ represents the average data rate per channel for the user u, considering an interference threshold of I_t.

From this number of channels needed per user N_u, the CA binary matrix allocation $X_{U \times N}$ can be constructed from the discrete allocation vector $\hat{x}_{ICA}$:

$$\hat{x}_{ICA} = [\underbrace{x_1, x_2, ..., x_{N_1}}_{u_1}, \underbrace{x_{N_1+1}, ..., x_{N_1+N_2}}_{u_2}, ..., x_{(N_1+N_2+...+N_U)}] \tag{3.18}$$

where the first N_{u_1} dimensions determine the allocated channels for the user u_1, followed by the required channels for the second user and subsequent users, in order. Given this arrangement, shown in Eq. (3.18), the optimization algorithm seeks a combination of channels to maximize network capacity. A random solution for this solution scheme can be generated as follows:

$$x_{i,d} = rand_{int}(1, N) \ \forall d \tag{3.19}$$

where for each required channel d, a random channel index is allocated. For calculating the ICA solutions' fitness, the vector $\hat{x}_{ICA}$ is transformed into an allocation matrix $X_{u,n}$ and the network capacity is calculated using the algorithm shown in Fig. 3.2.

The solution's dimensionality using ICA is defined as the sum of the number of channels required for the users in the system $\sum_{u=1}^{U} N_u$. Because cells can assign each channel to only one user, the search space for feasible solutions per cell is determined $\binom{N}{N_q}$, where N_q is the sum of required channels in cell q. Using this solution representation, a dimensionality reduction from UN to $U \cdot log(N)$ and search space from 2^{UN} to $\prod_1^Q \binom{N}{N_q}$ is achieved. However, the CA problem's complexity still grows exponentially with the number of active users, channels and SCs in the network.

3.3.3 Performance evaluation of ICA and Naive Solution Representation

A UDN with the parameters shown in Table 3.1 was used to evaluate CA's performance using metaheuristic algorithms while using the ICA scheme presented

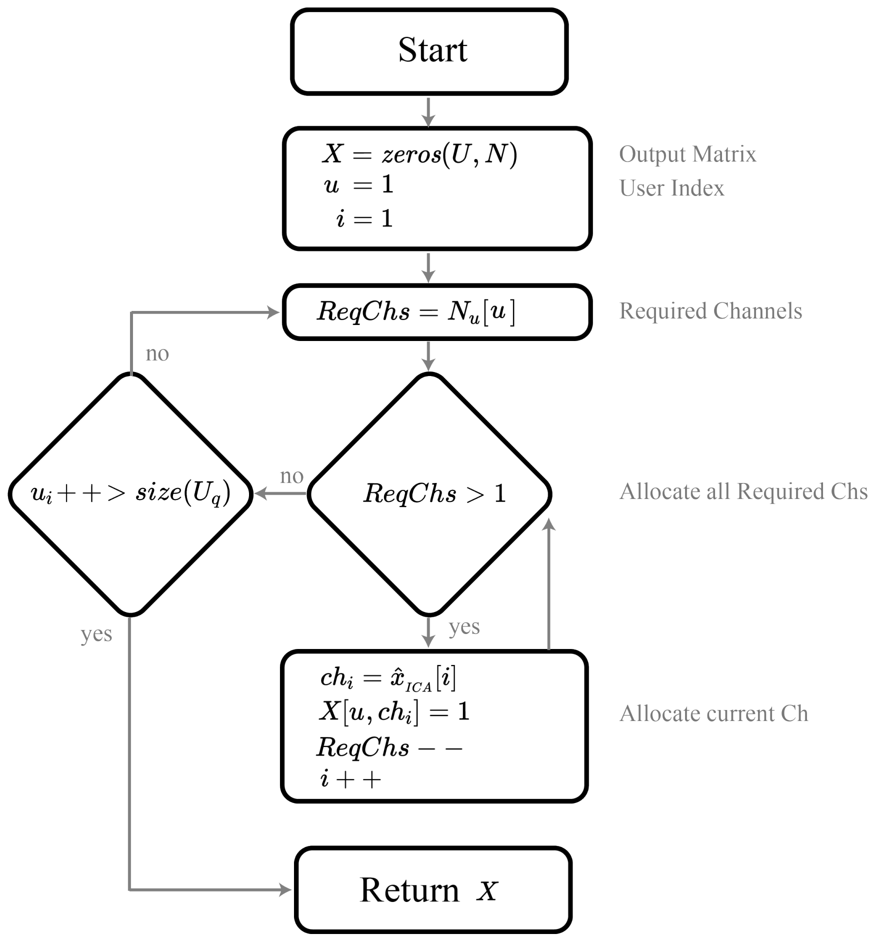


Figure 3.2: ICA Flow Chart

in this chapter. Given that the allocation's objective is to maximize the network capacity and provide individual QoS to the users, both are considered in the discussion.

The experiment was realized by generating 10,000 random solutions for the naive and ICA vector representation exposed in Eq. 3.14 and Eq. 3.18, respectively. The solution with the highest score, using the fitness function (Eq. 3.12), was selected as the representative allocation for each scheme.

The experiment results shown in Table 3.2 confirm that generating random binary allocation matrices produces a higher overall capacity than the ICA implementation. Still, the best fitness value found by random solutions is inferior to it. The naive implementation uses an unbias random generated binary vector. Therfore, the mean number of users per channel is 249, half of the total users in the network. Similarly, the mean number of channels allocated to each user is half of the total number of channels. This allocation produces a greater network capacity because of the spectrum is heavily utilized. However, it makes unfair distributions of the channels, as the link between cells and users have not homogeneous conditions. Therefore, most users do not reach their demands, and others are given more resources than they require, decreasing the fitness value of the solution. Suppose the random generator is biased to allocate a channel with a probability of 5% ($rand(0, 1) > 0.95$). In that case, the mean number of users allocated to each channel is 25. The mean number of channels per user is 50, similar to those allocated using ICA, yet the channels' unfair distribution remains. This result is shown in Fig. 3.3. The best naive random solution found for both random distributions, in terms of fitness, produces an unsatisfactory data rate for most users, while the rest are much higher than required. On the other hand, ICA manages to correct this unbalance in the

Table 3.1: Network Parameters

Network Parameters	Value
Macrocell's Power	46 dBm
Femtocells' Power	24 dBm
Pathloss α	3
Bandwidth B	200 MHz
No. Channels N	1000
Noise N_0	1e-10 watts
Limit SINR	6.3096 watts
Interference Th. I_{th}	1e-8 watts
Cells Q	200
Users U	500
D_{SC}^{min}	13 m
QoS	$[10e6, 50e6]$ Mbps

Table 3.2: Capacity and Response Time Comparison for the random exploration with the Naive and ICA vectors.

	Capacity ($Gbps$)	Time Response (s)	Fitness Value ($Gbps$)	Mean Users/Ch	Mean Ch/User
Naive 50%	30.00	485	0.12	249	499
Naive 95%	19.20	119	0.10	25	50
ICA	17.62	41	0.16	25	46

allocation. A fixed number of channels is allocated to each user, and the distribution is closer to their demands. Table 3.2 also shows that the time response is also affected by the high rate of channels allocated, given that this represents a higher interference and link capacity computations.

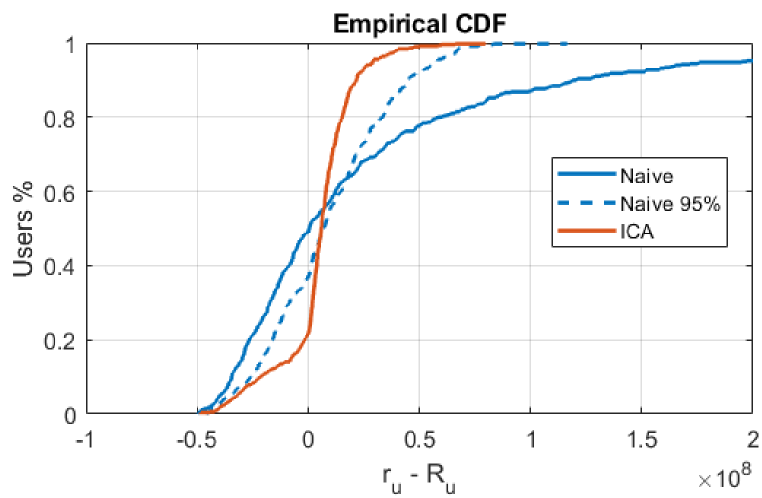


Figure 3.3: CDF users satisfaction comparison of Naive vs ICA random solutions.

3.4 Conclusions

This chapter introduced the mathematical model required to solve the RA problem for UDN. As mentioned before, several network's factors can be studied in detail by considering specific aspects of the network configuration, such as the number of antennas, central frequency, user association, or cell layout. In this thesis, a generic model with a focus on unplanned UDN is proposed. This model allows studying of the RA problem, precisely the CA stage. The chosen model configuration has the objective of analyzing the CA in UDN, its computational complexity, and the layout of the proposal, and the comparison of new approaches to solving it efficiently on highly dense networks. This model also seeks to represent a broad and simple network, which can be augmented to include later the two other leading technologies on which B5G is a vision based on MIMO and mmWaves.

The main characteristics of the proposed model are:

- Unplanned layout
- Random User Position
- Simple and easily exchangeable User Association.
- Deterministic channel model.
- Minimum SINR enforced in the capacity of link.
- Flat power allocation.
- Full spectrum sharing.
- Full interference calculation.
- Individual QoS is enforced.

By pre-calculating the potential interference that each user could perceive from each cell, the computational complexity of calculating the network capacity is $\mathcal{O}(NQlog(Q))$. Therefore, when the CA is solved using metaheuristic algorithms, the total computationally complexity is $\mathcal{O}(FFC \cdot NQlog(Q))$. It represents a much smaller and practical time response to maximize the network capacity than the required 2^{UN} necessary combinations to solve the problem optimally. Nonetheless, as studied in this chapter, the selection of the solution vector representation has an essential role in the performance of metaheuristic techniques. When comparing the naive matrix representation with the traditional ICA approach, a positive bias can be observed that produces fair assignments between users. ICA approach significantly reduces the response time due to the lower dimensionality on which the metaheuristic algorithms operate. These results show a clear advantage in designing vector solutions with lower dimensionality and consider important aspects of the network layout, which in ICA includes the users's QoS information to pre-calculate their required channels.

Chapter 4

Solving Channel Allocation with Metaheuristic Algorithms

> "A metaheuristic is formally defined as an iterative generation process which guides a subordinate heuristic by combining intelligently different concepts for exploring and exploiting the search space, learning strategies are used to structure information in order to find efficiently near-optimal solutions." [72]
>
> I. H. Osman and G. Laporte, "Metaheuristics: A bibliography"

4.1 Introduction

Metaheuristic algorithms are global optimization techniques based on intelligent random search patterns, looking for either the maximum or minimum possible value of any given objective function. Unlike complete algorithms, heuristics are algorithms that sacrifice the guarantee of finding optimal solutions for the sake of finding “good solutions” in a significantly reduced amount of time [73]. This characteristic is essential in NP-hard complexity problems, as no polynomial time complete algorithm is known to exist (if $N \neq NP$), thus requiring exponential computation time in the worst-case.

The term *heuristic* refers to “to find”, and it is associated with fast and problem specific search algorithms [73]. These algorithms generate solutions, from scratch, by adding components to an initially empty partial solution, until a solution is complete (constructive). They also start from initial complete solutions and improve them locally until a local optimum is found (local search). On the other hand, the term *meta* refers to algorithms that are “beyond, or in an upper level” from the heuristics algorithms. This term has first used by [74]. Unlike these, metaheuristics are designed to work as black boxes over any given optimization problem with none or few modifications [75] and avoid premature convergence, i.e., escape from local optima [76]. Avoiding premature convergence is achieved by either allowing worsening moves or generating new starting solutions for the local search in a more “intelligent” way than just providing random initial solutions.

The most novel and popular metaheuristics in the literature are population-based [77]. These begin by generating M initial random solutions as the population and evolving them iteratively through various mechanisms, usually inspired by natural phenomena. This initial-random-start makes an unbiased sample of the search space to recognize potentially “good” regions. Then, the information of the known solutions in the population and other possible data stored in memory is used to generate educated guesses (bias sampling), either by exploration (diversification) or exploitation (intensification). Exploration refers to the algorithm’s ability to create solutions in unknown regions of the search space, while exploitation is the previously obtained “good” solutions refining [78]. The algorithms’ skillfulness to find suitable solutions is mostly attributed to having the right balance between exploration and exploitation. However, the question of how to achieve the appropriate balance, or even measure it, is still an open issue [79]. For this reason, multiple metaheuristic methods are usually applied and compared in real-world problems to correctly select one that offers the best solutions for a given problem.

The generation of educated guesses relies on probabilistic decisions made during the search through random variables and various operators. Given the strong dependency of random values in these algorithms, it might seem that these will behave similarly to pure random search. “The main difference to pure random search is that in metaheuristic algorithms randomness is not used blindly but in an intelligent, biased form [80].”

Given these algorithms’ non-deterministic nature, each execution travels a unique search route and returns a different final solution when the stop criteria are met. These stop criteria are necessary as the optimal value is usually unknown, and there is no guarantee that it can be found. These criteria also control the algorithm time response, or computational complexity, as they limit the number of times the objective function is evaluated.

This chapter presents a general framework for implementing population-based metaheuristic algorithms and comparing some well-known and some novel metaheuristics, using a set of multi-modal benchmark functions and the Channel Allocation (CA) problem in Ultra-Dense Networks (UDN). The set of well-known algorithms chosen were Genetic Algorithm (GA), Differential Evolution (DE), and Particle Swarm Optimization (PSO), which are popular metaheuristics for their simplicity and proven efficiency. The more recent approaches, States of Matter Search (SMS) and Whale Optimization Algorithm (WOA), were elected and used to test the application of new research in Artificial Intelligence. The application of these algorithms follows the framework and specifications presented in [77], and their implementation is distributed as open-source [81].

4.2 Complexity Analysis

In computer science, the deterministic algorithms's efficiency is usually measured in their execution time using the Big O notation, which indicates the number of fundamental operations to produce the final answer and the amount of memory required to execute the algorithm. For example, the well-known sort algorithms bubble sort and merge sort have a worst-case complexity of $\mathcal{O}(N^2)$ and $\mathcal{O}(Nlog(N))$, while their memory size is $\mathcal{O}(1)$ and $\mathcal{O}(N)$ respectively. As the sorting process is deterministic, both algorithms return the same result: the sorted input array. The lowest computational complexity algorithm is considered the most time-efficient, and similarly with memory efficiency.

In contrast, most of the metaheuristic algorithms have linear computational complexity. Therefore, the most significant computation cost will be evaluating the objective function [71]. This function is evaluated several times over the algorithms' execution to obtain the final solution, with a total number of Fitness Function Calls (FFC) equal to the multiplication of the population size and the max number of iteration. Hence, the theoretical, computational complexity for most of these algorithms is $\mathcal{O}(FFC * \mathcal{O}(F(x)))$, meaning that time efficiency is not the most relevant aspect of them. On the other hand, the "quality" of such solutions plays an essential role in their performance efficiency. Unlike deterministic algorithms, each execution's final answer might differ due to these algorithms' stochastic nature. Likewise, the so-called "quality". This "quality" is problem-dependent, and the fitness function dictates its value. For this reason, the complexity analysis of metaheuristics has been a challenge since its introduction [82].

The comparison of these algorithms' performance is made by statistical tools, which utilize a set of benchmark functions to simulate the high complication that real-world problems have [83]. Then, by comparing the quality of the offered solution over multiple executions by two metaheuristic algorithms, it is possible to imply that a specific A algorithm performs statiscally superior to the B algorithm in such set of problems. However, there is a so-called 'no-free-lunch (NFL) theorem' proposed by Wolpert and Mcready (1997), which states that any algorithm will perform on average equally well as a random search algorithm for all possible functions. For this reason, this method can be applied to find the dominance of one algorithm over others in a specific problem, but remains unknown for the science which algorithm is overall predominant to all others and the reasons for such superiority, if it exists [52]. The lack of commonly accepted procedures to properly develop, study, and compare metaheuristics, triggered several critical research outputs that identified many important issues [84].

4.3 Meta-parameters Selection

The parameter tuning relevance lies in the fact that these allow changing the exploration–exploitation ratio of the algorithms, potentially improving their performance over specific tasks. However, there is no general agreement on how metaheuristic algorithms should be tuned to work optimally over particular optimization problems. Therefore, scientists and practitioners adjust metaheuristics parameters by hand, guided only by their experience and rules of thumb. Hence, tuning metaheuristics is often considered to be more of an art than a science. Although several efforts aimed to provide frameworks to mechanically and objectively select these parameters, finding the optimal tuning for these parameters is still an open problem [52], [85].

For this work, these meta-parameters were selected from the recommended range from previous published applications, shown in Table 4.1. The specific value of these parameters were selected manually, given the author previous experience and without any effort to fine tune them for the CA problem.

4.4 Solving the Channel Allocation Problem with Metaheuristics

In Chapter 3, the mathematical model to solve the Channel Allocation (CA) optimization problem in UDN was introduced. As summarized in Fig. 4.1, the network model is generated from the indicated network parameters, such as the number of users U, cells Q and channels N, to construct the information of cells and users' positions, data rate demands, calculate distances and perceived signal between users and cells. The CA problem's objective is to find the binary allocation matrix $X_{U \times N}$ that maximizes network capacity (C) while meeting the QoS demands of users (c_u). This optimization task is carried by the metaheuristic algorithm, which iteratively generates random biased solutions with the information recollected from previous attempts. It is performed for several iterations until a stop criterion is met (usually a fixed number of iterations),

Table 4.1: Parameters for the Ultra Dense Network

Algorithm	Parameter	Value	Reference
GA	Crossover	0.9	[86]
	Mutation	0.05	
PSO	Velocity Weight	0.4	[87]
	Best Personal	2	
	Best Global	2	
DE	Crossover	0.7	[88]
	Differential Weight	0.5	
SMS	Step Gas, Liquid, Solid)	0.85	[89]
	Step Liquid	0.35	
	step Solid	0.1	
WOA	b	1	[90]

finally returning the solution with the best fitness value as a solution for the problem.

As mentioned in the previous chapter, using the matrix $X_{U \times N}$ for the meta-heuristic algorithms as the solution representation entails several complications for the search, such as the immense search space full of unsuitable solutions and the unset number of channels to be allocated for the users. For this reason, the ICA scheme was introduced to reduce the optimization problem complexity. The idea is to previously calculate the number of channels to allocate to each user (N_u) to meet the user data rate R_u. It is done by considering an interference threshold limit I_t, as shown in Eqs. 3.17 and 3.16. Given this number of required channels, the $\hat{x}_{_{ICA}}$ discrete allocation vector, shown in Eq. 4.1, can be used to generate more efficient CA solutions.

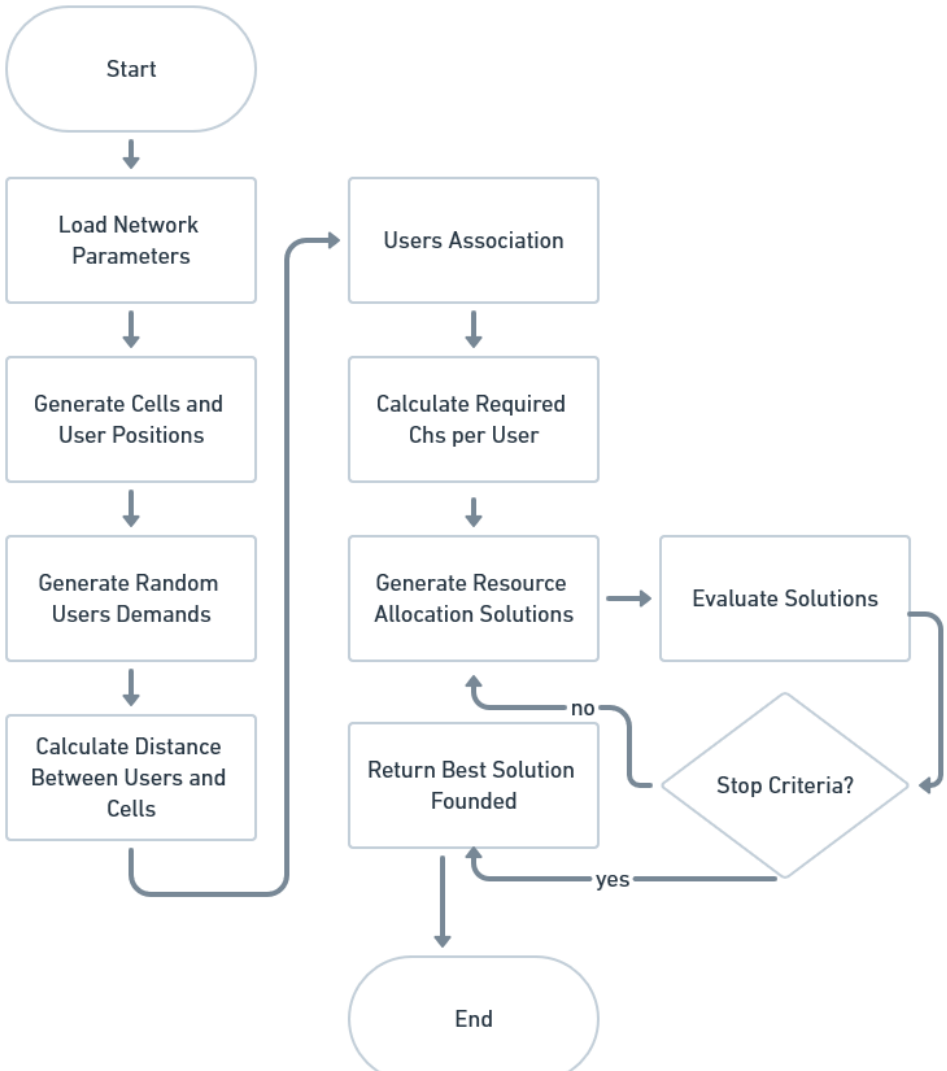


Figure 4.1: Resource Allocation Flow Chart

$$\hat{x}_{ICA} = [\underbrace{x_1, x_2, ..., x_{N_1}}_{u_1}, \underbrace{x_{N_1+1}, ..., x_{N_1+N_2}}_{u_2}, ..., x_{(N_1+N_2+...+N_U)}] \tag{4.1}$$

such that:

$$1 \leq x_d \leq N, \ \forall d$$

In the $\hat{x}_{ICA}$ vector, the first N_{u_1} dimensions determine the allocated channels for the user u_1, followed by the required channels for the second user, and the subsequent users in order. Following this logic, and as described in Fig. 3.2, the CA binary matrix allocation $X_{U \times N}$ can be constructed from any $\hat{x}_{ICA}$ vector. This procedure can be translated into a fitness function, compatible with the metaheuristic general framework introduced in this chapter, by scaling the $[0, 1]$ range into the discrete $\{1, 2, ..., N\}$ one, generating the binary CA matrix allocation and evaluating the fitness.

4.4.1 ICA and Naive schemes performance results

This section compares the performance metaheuristic algorithms described in this chapter to solve the RA problem for UDN using the ICA scheme introduced in Chapter 3. Table 4.2 shows the values of the simulation parameters used for the experiments. The evaluation considers a small area of 200mx200m (i.e., 0.04 km^2), three different active user densities (5000, 10000, 15000 $users/km^2$), and a range of SC from 50 to 400 in increments of 50, which represents densification values from 1250 to 10000 $cells/km^2$. We refer to the number of cells in the network for the results rather than the densification value. The parameters shown in Table 4.2 fulfill the two main definitions for UDN: the number of

Table 4.2: Parameters for the Ultra Dense Network

Parameter	Value
Bandwidth	180 MHz
Number of Channels (N)	1000
Coverage Area	200 x 200 m
User Data Target	10 Mbps
Interference Threshold	-50 dBm
Noise Per Channel	-70 dBm
SINR Threshold	-35 dBm
Number of SC (Q-1)	[50 - 400]
Number of Users (U)	[200, 400, 600]
MC Power	46 dBm
SC Power	24 dBm
SC Minimum Distance	10 m

deployed base stations is higher than the number of active users, and the density of SC is higher than 1000 $cells/km^2$ [17].

For each experiment configuration, 30 independent instances of the UDN were constructed, on which cells and users' positions were generated randomly, as shown in Chapter 3.3.2. Then, the corresponding metaheuristic algorithm is applied over these network instances, using the meta-parameters shown in Table 4.1 with a population of 30 solutions and 500 iterations with 15000 function calls. The network capacity values shown in the following results represent the mean network capacity obtained using the algorithms over the 30 independent executions.

Tables 4.3 and 4.4 show the resultant network capacity and unsatisfied user percent from this experiment, in which bold text remarks the algorithm with the best score of each network configuration. It shows that all the algorithms can provide increased capacity with the network densification in both cells and users' demands (either number of active users or data rate demands). In the Table 4.3 three algorithm excels over the rest, being WOA, SMS, and GA, with 8, 6, and 1 best scores (marked with bold font). These results follow the cell densification theoretical limit on which cells have an average of one user or less associated with them. Also, when 200 users in the network, the network capacity reaches a peak at 300 SC, while for 400 and 600, no clear maximum is reached. This same behavior is observed in Table 4.4, in which only the unsatisfied user percent is eliminated and minimized using the highest densification and the SMS algorithm for 400 and 600 users, respectively. This fact indicates that further densification could still be beneficial to increase the network capabilities for these demands.

Table 4.4 shows the unsatisfied users' percent from the best solution obtained by the optimization algorithms. These percents are an essential part of the proposed fitness function, as exposed in Eq. 3.12, which proportionally penalizes the solutions score regarding the number of users with lower data rate capacity than their demands. Given the relationship between these values and the fitness function, the algorithm that offers the lower unsatisfied users percent represents the best fitness value. When identical unsatisfied users percent are met, such as those solutions with all users satisfied (zero value in the Table 4.4), the best fitness is found with the higher network capacity solutions. In this case, GA is the one that mostly obtained the best results. These results show the present opposition between maximizing network capacity and providing individual QoS

Table 4.3: Performance comparison of network capacity (Gbps) using metaheuristic algorithms for solving Channel Allocation in UDN with 200, 400 and 600 active users, range of deployed cells from 50 to 400.

Users	Alg / Cells	50	100	200	300	400
200	GA	1.9231	2.3287	3.5034	3.5917	3.5415
	PSO	1.7439	2.1641	3.0814	3.3662	3.3249
	DE	1.7509	2.1525	3.1396	3.3517	3.3317
	SMS	1.7836	2.3505	3.5086	**3.6616**	**3.5886**
	WOA	**1.9575**	**2.9052**	**3.5559**	3.4774	3.3885
	RAND	1.7824	2.2200	2.9867	3.2817	3.2705
400	GA	1.9408	2.382	4.4642	5.6101	5.8974
	PSO	1.8159	2.332	3.7481	5.42	5.6014
	DE	1.8448	2.3183	4.2394	5.2717	5.5312
	SMS	1.9003	2.4528	4.3782	**5.6323**	**5.9566**
	WOA	**2.0624**	**2.5775**	**5.0757**	5.6021	5.7439
	RAND	1.7665	2.3520	4.0110	5.1633	5.4499
600	GA	2.0617	2.6673	**4.8743**	6.7851	7.4166
	PSO	1.8898	2.5762	3.8821	6.8419	7.415
	DE	1.9522	2.5866	4.4748	6.7516	7.2936
	SMS	1.9823	2.6779	4.4265	6.7948	7.587
	WOA	**2.1298**	**2.8173**	4.7639	**7.1107**	**7.6399**
	RAND	1.8545	2.6700	4.0685	6.4915	7.1904

Table 4.4: Performance comparison of unsatisfied users percent using metaheuristic algorithms for solving Channel Allocation in UDN with 200, 400 and 600 active users, range of deployed cells from 50 to 400.

Users	Alg / Cells	50	100	200	300	400
200	GA	**65.075**	36.375	**0**	**0**	**0**
	PSO	70.133	55.366	2.350	**0**	**0**
	DE	69.350	53.333	0.150	3.3517	3.3317
	SMS	70.956	42.316	0.033	**0**	**0**
	WOA	70.833	**23.300**	**0**	**0**	**0**
	RAND	69.116	51.500	2.333	**0**	**0**
400	GA	**83.312**	**79.225**	22.258	0.616	0.004
	PSO	85.050	80.666	55.400	2.375	0.308
	DE	84.633	80.058	37.308	4.450	0.591
	SMS	85.675	80.550	27.708	**0.333**	**0**
	WOA	86.191	80.091	**16.491**	0.791	0.008
	RAND	85.508	80.116	46.391	7.058	1.483
600	GA	**88.660**	**86.170**	**56.941**	**14.366**	3.850
	PSO	90.255	87.338	80.900	22.338	7.861
	DE	89.808	87.275	70.450	21.066	9.100
	SMS	90.627	86.727	71.255	17.188	**2.461**
	WOA	91.122	87.105	72.161	18.388	4.088
	RAND	90.622	86.861	78.827	30.672	12.29

to the users. It is due to higher capacities are achieved by few users with a high data rate. In UDN, lower interference means that the channel sharing strategy is reduced, rather than a well-distributed and fair channel allocation, as studied in several resource allocation studies [91]. Therefore finding solutions that represent efficient and balanced resource allocation means a complex task for the metaheuristic algorithms, especially in UDN.

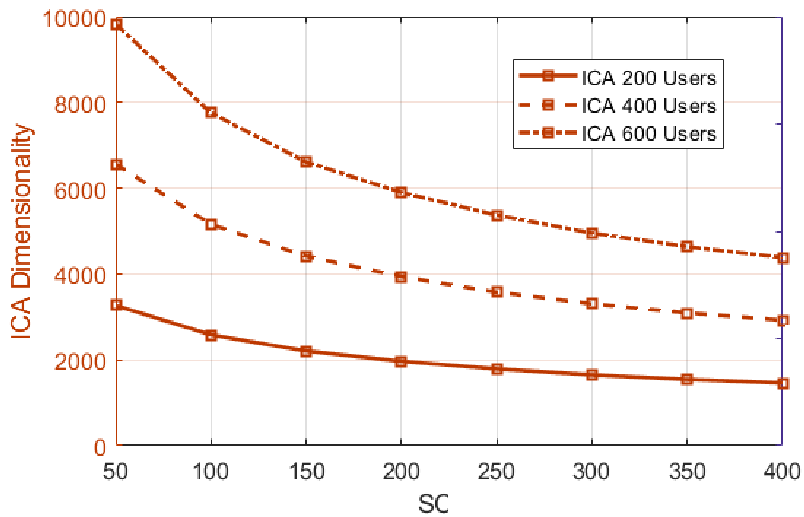


Figure 4.2: Dimensionality of the CA problem using the three ICA allocation scheme for the three levels of user density considered.

The problem complexity using the ICA scheme involves a high dimensionality (above 1,000). It depends on the sum of channels required to serve all users in the UDN, $\sum_{u=1}^{U} N_u$, and even higher exponential search space. Figure 4.2 shows that this high dimensionality is logarithmically reduced with the cell densification. It is one of the advantages of the ICA representation. Adding SC improves signal propagation conditions, so users' QoS data rates can be reached with fewer channels. Nonetheless, this signal improvement decreases concerning network densification. As Table 4.3 shows, for the three levels of user densities, WOA obtains the highest capacity values when the problem's most increased complexity occurs (i.e., 50 or 100 cells). However, GA gets the lowest unsatisfied users percent in most cases, returning the best fitness value solution. On the other hand, SMS seems to perform best with high cell densities, independently of the users' demands. In general, the results indicate that GA presents the best scaling properties for the CA problem in UDN when using the ICA solution representation.

4.5 Conclusion

This chapter presented a general framework for metaheuristic algorithm and used it to implement several population-based algorithms, which can be found as open-source [81]. Their operators' main differences can be studied and compared by abstracting population-based metaheuristics' similarities and putting aside their inspirational nature. Several categories from the algorithms' details can be devised. From these categories, the idea is that we can formulate explanations about the algorithms' performance to determine which characteristics are appropriated for certain types of problems, as proper comparison methods are still an open problem in science. This study forms part of a survey article on metaheuristic algorithms published in [77].

The metaheuristic algorithms presented in this chapter were applied to solve the CA problem for UDN introduced in Chapter 3, using the ICA solution scheme. The results show that the metaheuristic algorithms provide solutions that increase the network capacity in both cells and users' demands (either number of active users or data rate demands). Three excels on their performance over the rest from the algorithms, being GA, SMS, and WOA. Given the experiment's conditions, the results indicate that GA presents the best scaling properties for the RA problem in a UDN when using the ICA solution representation. However, the dimensionality of the problem applying ICA solution representation remains in all the cases above 1,000. This high dimensionality represents an issue, because metaheuristic algorithms are rarely designed or studied over large optimization problems [92]. Covering this immense search space would require an unfeasible number of call functions. The high-dimensionality in the CA problem is impractical given the non-linear computational complexity of evaluating a solution of $\mathcal{O}(NQlog(Q))$.

However, suppose the ICA method is applied for the channel allocation in UDN. In that case, it will be difficult for the metaheuristic algorithms to find close-to-optimal solutions in acceptable response times. For this reason, a novel solution representation based on block channel allocation is introduced in Chapter 5. It reduces the CA problem complexity in terms of dimensionality and search space by considering recent findings on UDN. Also, it allows metaheuristic algorithms to yield better solutions without increasing the computational resources invested.

Chapter 5

Block Channel Allocation Proposal for Ultra-Dense Networks

5.1 Introduction

As mentioned in Chapter 2, several novel approaches have been proposed for CA in UDN. However, these approaches present one of the following issues: (i) Do not consider users QoS, which leads to unfair resource distributions and user data rate starvation; (ii) distributes the bandwidth among a low number of channels, which can lead to inefficient use of the resources by allocating excessive bandwidth and increasing interference; (iii) aims to exploit the temporary conditions of the propagation environment instead of avoiding network interference. It is not sufficient, considering only the signals' propagation conditions to improve network performance. In UDN, there is already Line-of-Sight (LoS) between users and small base stations that improves the transmission link [51].

Nonetheless, the proposed mechanisms that manage to reduce the high-dimension CA optimization problem can be included in a novel approach to metaheuristics implementation. For example, distributing the available spectrum among a lower number of channels reduces the CA problem's search space and dimensionality at the cost of increasing interference [66]. However, by considering the user's QoS requirements, it is possible to figure out the number of channels required to reach their demands by applying an interference threshold, bounding the bandwidth allocated to them [57]. Being this the ICA approach previously mentioned, which is found as one of the most used CA methods applied in low cell densification scenarios [31], [46], [56], [57], [67]. Nonetheless, by using ICA the optimization problem's dimensionality grows exponentially with the network scale, which complicates the finding of CA's feasible solutions in UDN scenarios. Suppose the CA strategy aims to reduce the UDN interference instead of exploiting the propagation environment's temporary conditions. In that case, a path loss channel model can be used for calculating network capacity. This strategy provides a channel Signal-to-Noise-Ratio (SNR) invariance. This leaves the interference as the only factor that affects the network capacity. Similarly to [66], a dimensionality reduction can be achieved by allocating each user's entire required bandwidth in a single channel. Nonetheless, our proposed approach avoids the aggregated interference present in [66] by allocating channels with specific bandwidths regarding users' demands. We call this approach User Block Allocation (UBA). We can implement a further step in this concept by considering a layered CA where a centralized entity distributes channels among cells minimizing overall network interference. This central step is followed by cells autonomously allocating their available resources to their users, similarly to [50] or cluster approaches [42]. This goal is achieved by allocating a block of resources to each cell, considering the sum of the required channels of their associated users. We call this Cell Block Allocation (CBA). This CBA approach is practical due to the small coverage area of SC, which results in a statistically equivalent level of interference perceived by each of SC's users. Therefore, assigning channels to users in UDN becomes the issue of assigning non-orthogonal spectral portions to small-cell. It reduces, even more the high-dimensionality and search space of the optimization problem. This dimensionality reduction to the CA problem benefits the approximated algorithms since it would allow them to find high-quality solutions.

In this thesis, a resource block (RB) allocation solution vector is developed to maximize the aggregate system throughput while being subjected to interference constraints with an approximated method, such as metaheuristic algorithms. An RB is defined as a set of adjacent and applicable channels, which size depends on the user's QoS requirements. Our proposal is based on allocating a single RB to each user equipment (UE) instead of a set of disjoint channels across the available spectrum. This RB allocation allows the whole required bandwidth of either users or cells as a proper and single RB. These UBA and CBA proposals consider a filtered search space of channel allocations providing two-dimensionality reduction levels to the CA problem. This work shows that the proposed arrangements alleviate the CA optimization's high complexity and find efficient solutions for UDN scenarios with the GA.

5.2 Description of Block Channel Allocation Proposal

Given these issues, as mentioned above, finding high "quality" solutions for the CA problem in UDN using sub-optimal methods, such as metaheuristic algorithms, becomes too complicated. Therefore, new strategies are needed to reduce the CA problem complexity, allowing metaheuristic algorithms to find feasible and efficient solutions. Our block CA proposal considers three aspects to reduce the computational complexity of the CA optimization problem:

1. To consider only the Path Loss in the channel model.
2. To calculate the specific number of channels that each user requires to reach their demands.
3. To allocate the necessary channels of each user as a single block of spectrum.

By considering Path Loss as the main factor affecting the channel gain, excluding the effects of shadowing and multi-path fading [48], the interference created by multiple cells sharing the same channel becomes the critical factor affecting each link's capacity in the network. It is describerd in Chapter 3 in which only the interference factor ($I_{q,u,n}$) is affected by the channel index n. This channel SNR invariance allows our CA proposal to reduce interference during the allocation. The propagation environment's random temporary conditions can be considered later in a distributed and independent Power Allocation (PA) stage.

The second statement is related to the ICA approach. The required channels are calculated using an interferent threshold I_{th}. Nonetheless, one can further improve the CA problem's vector solution representation by considering the previous ideas. With the $\hat{x}_{_{ICA}}$ vector, each dimension represents the allocation of each one of the users' required channels. However, by considering the channel SNR invariance in the proposed model, what matters is which users share a common channel and not which specific channel is shared. Suppouse a channel is efficiently shared among a set of users in terms of interference. All the required channels by these users can be allocated for sharing any of the available channels in the network. Therefore, a vector solution representation can be designed to focus the CA problem on selecting users who could share a channel efficiently, regardless of its channel. In this sense, we propose a novel CA approach based on predefined sets of resource blocks (RB), which further reduces the high-dimensional ICA problem and search space. In this case, the N_u required channels could be assigned in a block structure of adjacent channels, either to users with UBA or cells with CBA, instead of individually assigned. This strategy reduces the solution search space into a subset of contiguous RB. For the UBA scheme, the RB size is a function of the number of channels N_u each user u_i requires, while for CBA, the size of the RB is represented by the total number of channels needed N_q for all users in the cell q_j.

The following discrete Allocation Vectors represent the CA binary Matrix $X_{u,n}$ for UBA and CBA in Eq. 5.1 and eq. 5.2, respectively.

$$\hat{x}_{_{UBA}} = [\underbrace{x_1}_{u_1}, \underbrace{x_2}_{u_2}, ..., x_i, ..., x_U] \tag{5.1}$$

For the $\hat{x}_{_{UBA}}$ vector, each dimension $\hat{x}_{i_{UBA}} \forall i \in [1, U]$ represents the the channel block's first channel for the user u_i. The value of x_1 represents the initial channel of the resource block assigned to the user u_1, x_2 for the user u_2, and so on. In contrast to ICA, only the initial channel is required to be specified to each user. This channel $\hat{x}_{i_{UBA}}$ and the $N_{u_i} - 1$ following required channels are allocated to it. Given the domain of each dimension $[1, N]$, a random solution can be generated in the same fashion (Eq. 3.19). Fig. 5.1 details the procedure to generates a binary CA Matrix $X_{u,n}$ from any given allocation vector $\hat{x}_{_{UBA}}$.

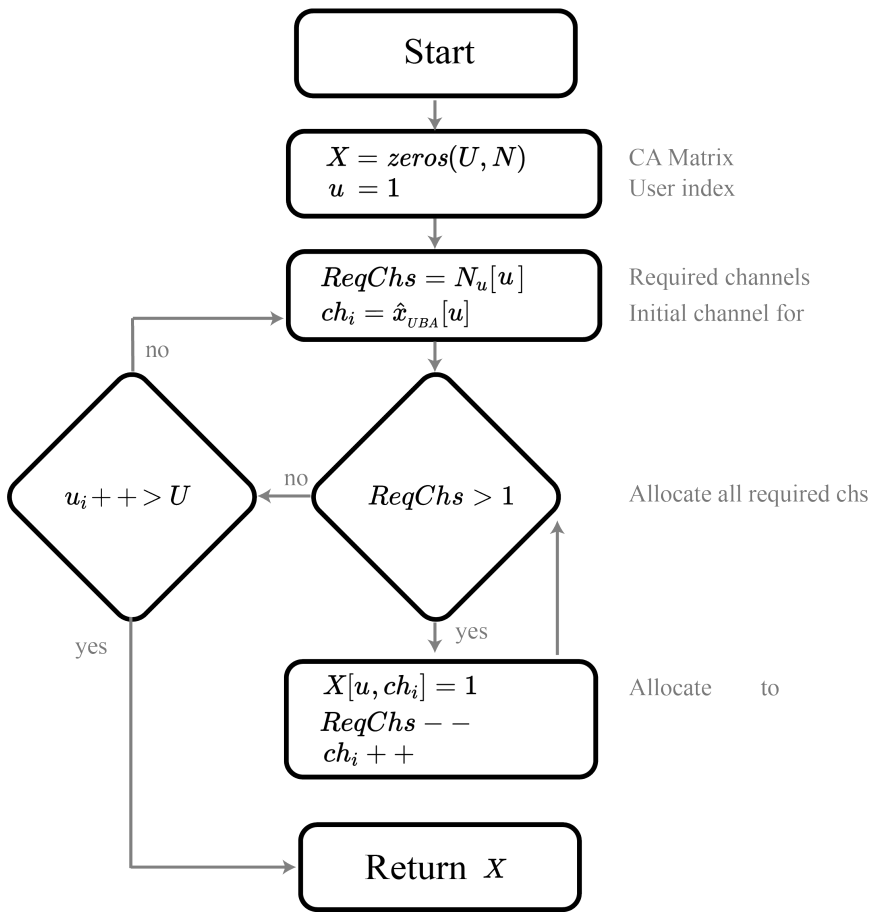


Figure 5.1: UBA Algorithm to transform $\hat{x}_{UBA}$ to X.

On the other hand, for the vector $\hat{x}_{_{CBA}}$, a specific channel is allocated directly to each active cell in the network. The CBA algorithm, detailed in Fig. 5.2, is executed to generate the binary CA Matrix $X_{u,n}$. In this case, x_1 indicates the initial channel assigned to cell q_1, where q_1 is the MC, therefore the first associated user of the MC $U_{q_1,1}$ is allocated to this x_1 channel and the $N_{U_{q_1,1}}$ consequential channels. Then, the second associated user of the MC ($U_{q_1,2}$) is allocated to the channels from $(x_1 + N_{U_{q_1,1}} + 1)$ to $(x_1 + N_{U_{q_1,1}} + N_{U_{q_1,2}})$. This process continues for all the associated users of q_1, followed by x_2 for q_2 and so on.

$$\hat{x}_{_{CBA}} = [\underbrace{x_1}_{u \in U_{q_1}}, \underbrace{x_2}_{u \in U_{q_2}}, ..., x_i, ..., x_Q] \tag{5.2}$$

The UBA and CBA schemes operate similarly, looking for a shared structure between users or cells in which interference is reduced. This method is comparable to clustering approaches like [42] and [50]. The SC with little mutual interference are grouped so that they are allocated non-orthogonal channels. However, our approach can accomplish the clustering and the channel allocation forming clusters that adjust to the cell load.

5.2.1 Case Example of ICA, UBA and CBA

To explain the proposed UBA and CBA allocation schemes' goals in more detail, we will use the example presented in Figure 5.3. A simple mobile network is shown consisting of 10 channels and an MC (q_1), two SC (q_2, q_3), and three users (u_1, u_2, u_3) displayed in the coverage area of cell q_1. In this scenario, users u_1 and u_2 are associated with cell q_1 while cell q_2 serves u_3. Cell q_3 is deactivated since there are no active users associated with it. Assuming a given user data

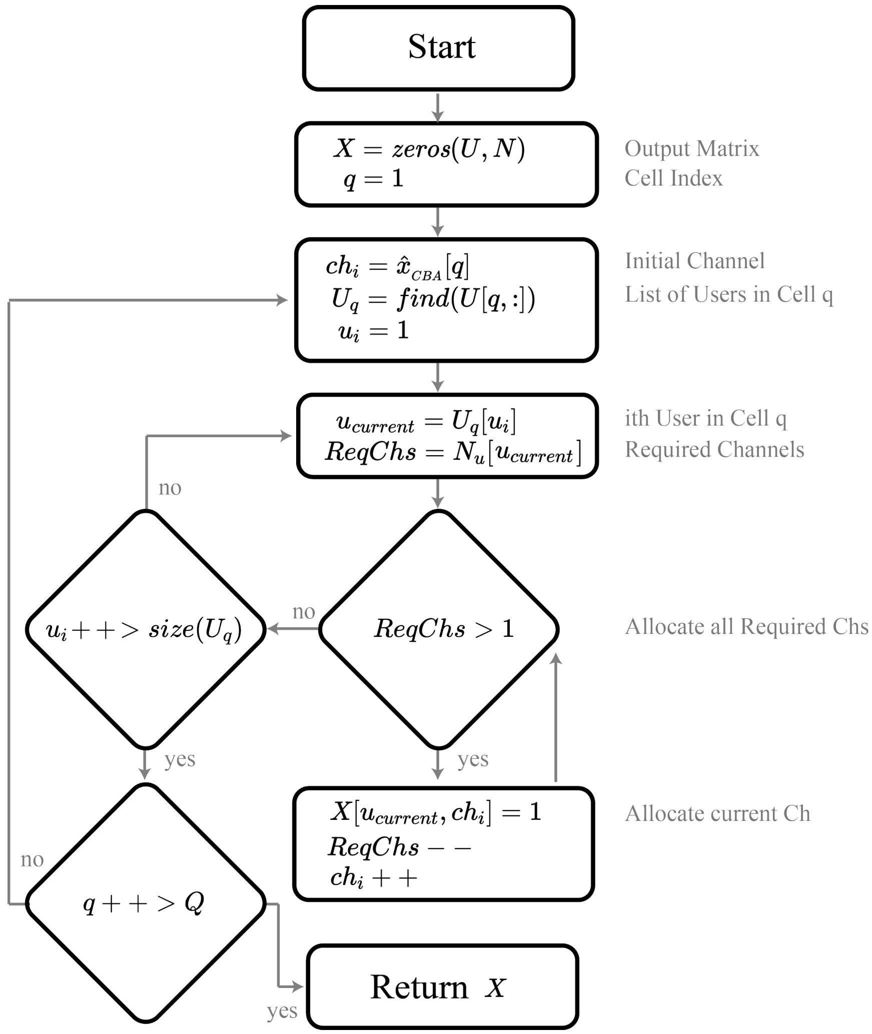


Figure 5.2: CBA Algorithm to transform $\hat{x}_{CBA}$ to X.

rate demand, the calculation of the required channels for each user N_u, using Eq. (3.16), results in 6, 3 and 4, respectively. Table 5.1 shows the structure of the Channel Allocation Vector $\hat{x}_{ICA}$, $\hat{x}_{UBA}$, and $\hat{x}_{CBA}$ for the ICA, UBA, and CBA schemes representing the coded solution. The metaheuristic algorithms will find the best solution to the CA problem. Channel Allocation Matrix $X_{U \times N}$ is used to calculate the network capacity (Eq. 3.6).

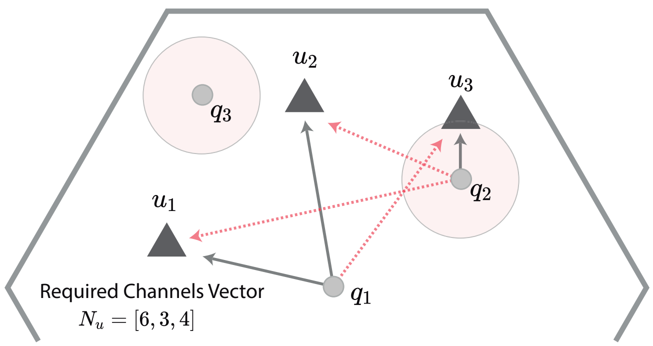


Figure 5.3: Comparison between ICA, UBA and CBA vector structures for an UDN example.

Table 5.1: Comparison between ICA, UBA, and CBA vector structures for the example network of this section.

	ICA	UBA	CBA
Channel Allocation Vector $\hat{x}$	$[\underbrace{x_1, x_2, \dots, x_6}_{u_1}, \underbrace{x_7, x_8, x_9}_{u_2}, \underbrace{x_{10}, \dots, x_{13}}_{u_3}]$	$[\underbrace{x_1}_{u_1}, \underbrace{x_2}_{u_2}, \underbrace{x_3}_{u_3}]$	$[\underbrace{x_1}_{q_1}, \underbrace{x_2}_{q_2}]$
Channel Allocation Example $\vec{S}$	$[1, 2, 3, 5, 8, 9, 10, 4, 7, 10, 4, 7, 2]$	$[1, 7, 6]$	$[1, 6]$
Channel Allocation Matrix X	$\begin{bmatrix} 1110100110 \\ 0001001001 \\ 0101001001 \end{bmatrix}$	$\begin{bmatrix} 1111110000 \\ 0000001110 \\ 0000011110 \end{bmatrix}$	$\begin{bmatrix} 1111110000 \\ 0000001110 \\ 0000011110 \end{bmatrix}$

For the ICA scheme, $\hat{x}_{_{ICA}}$ contains the individual channel x_i assigned to each user in the network. The dimensionality of $\hat{x}_{_{ICA}}$ is thirteen (that is, the sum of all the channels required in the system). The first six dimensions correspond to channels (1, 2, 3, 5, 8, and 9) assigned to user u_1; the next, three channels (4, 7, and 10) are for user u_2 and the last four channels (10, 4, 7, and 2) for user u_3. Users u_1 and u_2 cannot share any channel because they are associated with the same cell, q_1. On the other hand, u_3 is sharing channel 2 with u_1 and channels 4, 7, and 10 with u_2. Solution vector $\hat{x}_{_{ICA}}$ is transformed into the matrix $X_{U \times N}$, as explained in Figure 3.2. It is important to note that, as mentioned before, in the proposed UDN model, what matters is that users u_1 and u_2 share channels 1 and 3 with user u_3, regardless of which channels they share. From $\hat{x}_{_{ICA}}$, it is possible to generate a large number of solutions for the CA problem. Some of them could result in either the same channel allocation or network capacity. Thus, this replica of solutions and high-dimensionality of ICA scheme makes it difficult for stochastic methods to find feasible and efficient solutions during the CA process.

A similar ICA solution ($\hat{x}_{_{ICA}}$) can be constructed using the UBA and CBA schemes. As a result of the proposed dimensionality reduction, the channel allocation vector $\hat{x}_{_{UBA}}$ is decreased to three dimensions. In this case, each dimension represents the first channel of the corresponding channel block of each user. For example, the channel block for user u_1 begins with channel 1; therefore, this and the next five channels are assigned to u_1.

Similarly, channels 7, 8, and 9 are assigned to u_2. Lastly, the channel block for u_3 begins on channel 6, sharing one channel with u_1 and three channels with u_2.

Similarly, the dimensionality of the CBA vector $\hat{x}_{_{CBA}}$ is reduced to only two dimensions, based on the number of active cells in the network. For example, the channel block for cell q_1 begins with channel 1. Thus its associated users, u_1 and u_2, are assigned consecutively over the indicated channel. Then user u_1 is assigned to channels 1 through 6, user u_2 to channels 7 through 9, and user u_3, associated with q_2, to channels 6 through 9. Vectors $\hat{x}_{_{UBA}}$ and $\hat{x}_{_{CBA}}$ can be easily transformed to $X_{u,n}$ using algorithms shown in Figures 5.1 and 5.2 respectively.

The $X_{u,n}$ channel allocation matrices offered by these three schemes involve one unused channel 10, users u_1 and u_3 sharing channel 6, and users u_2 and u_3 sharing three channels, 7 through 9. These allocations produce an identical network capacity as the previous strategy due to the same level of interference generated over the shared channels. However, finding this particular solution is simplified by considering the proposed UBA and CBA structure due to the reduced number of channels allocated (dimensionality) and possible combinations (search space).

5.2.2 Search Space and Dimensionality Comparison

Our proposals UBA and CBA, were designed to reduce the CA optimization problem's dimensionality and search space. The dimensionality of UBA is equal to the number of users U, and the search space is $\prod_1^Q \binom{N}{U_q}$, where U_q represents the number of users associated with that cell. On the other hand, CBA dimensionality depends on the number of cells Q, and the search space is $\prod_1^Q \binom{N}{1} = N^Q$. CBA has the smallest search space of the three strategies. Although the search space is significantly reduced using the UBA or CBA strategies compared to the conventional ICA scheme, its increase rate remains exponential. However, the problem-dimensionality increases linearly based on the number of users U (for UBA) or cells Q (for CBA), independent of the data rate demands. In terms of the solution of the high-dimensional problem, the CBA scheme is less complex than UBA. It has a fewer dimensions and search space because the number of cells Q is usually less than the number of users U. Even if the UDN has a higher number of cells than active users, the CBA only considers cells that are actively assigned to users. Therefore, in the worst case, the number of CBA dimensions would be equal to that of UBA. An additional benefit of applying the CBA strategy is that any channels' random assignment complies with the single user per channel cells' constraint. The contiguous block contains all the channels needed to meet user demands.

Figure 5.4 shows the resultant dimensionality of the ICA, UBA, and CBA solution representation when three user densities ([200,400,600] active users) were used in the experiment. As mention before, the number of dimensions of ICA is logarithmically reduced with regard the cell densification. On the other hand, UBA's solution remains fixed with the user's densification. In contrast, the solution vector of the CBA scheme grows with the densification of cells. However, it is limited by the number of active users. For UBA and CBA allocation schemes, the highest problem dimensionality considered in the network evaluations of this work is 600 and 400 for the case of 600 active users and 400 SCs in the network, respectively. This result means that the highest dimensionality of the UBA and CBA schemes is less than half the lowest dimensionality obtained with ICA (for 200 active users and 400 SC).

It is important to notice that reducing the problem complexity is carried by restricting the solutions to allocations of contiguous RB channels for the users or cells. Figure 5.5 shows a UDN channel allocation solution example resulting

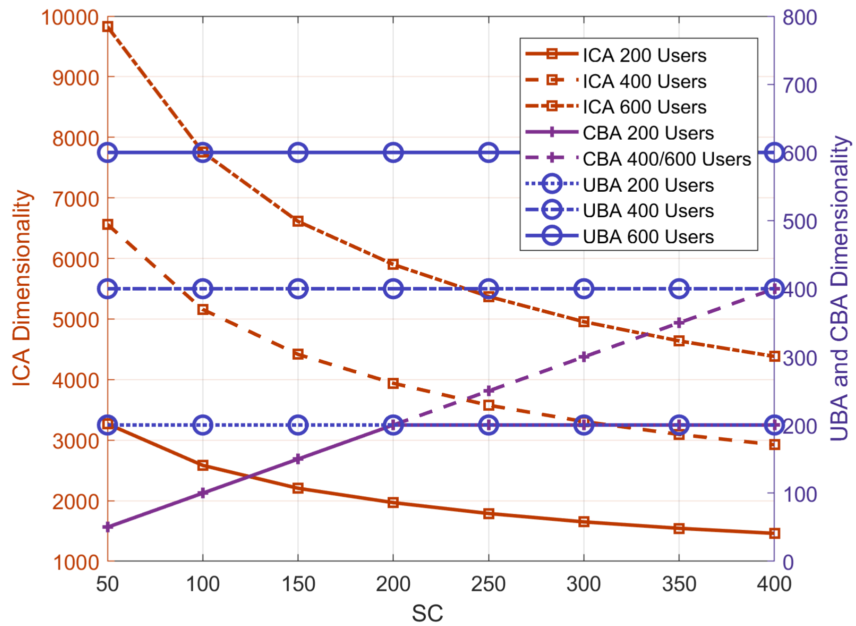


Figure 5.4: Dimensionality of the Channel Allocation problem using the three Allocation Schemes ICA, UBA and CBA for the three levels of user density considered.

from using the ICA, UBA, and CBA schemes. The rows, columns, and pixel color represent the cell index, the channel index, and relative capacity achieved, respectively. The black color represents the channels not used by the cell.

For the CBA scheme, the channels are allocated in a contiguous RB in the cells. These cells are allocated to overlapping channels (interfering with each other), as the vertical rectangle points out in the figure. Similarly, for the UBA scheme, contiguous RB are allocated directly to users. However, there is a higher probability of cells interfering with multiple other cells as the allocation is dispersed over the spectrum. It worsens with the conventional ICA scheme.

Nonetheless, all three schemes allocate the same amount of channel to each user, and any UBA solution is included in the search space of ICA. The same applies to CBA. The solution shown in the Figure 5.5 can be represented using the UBA and ICA vector solutions, but with higher dimensionality. Therefore, UBA is

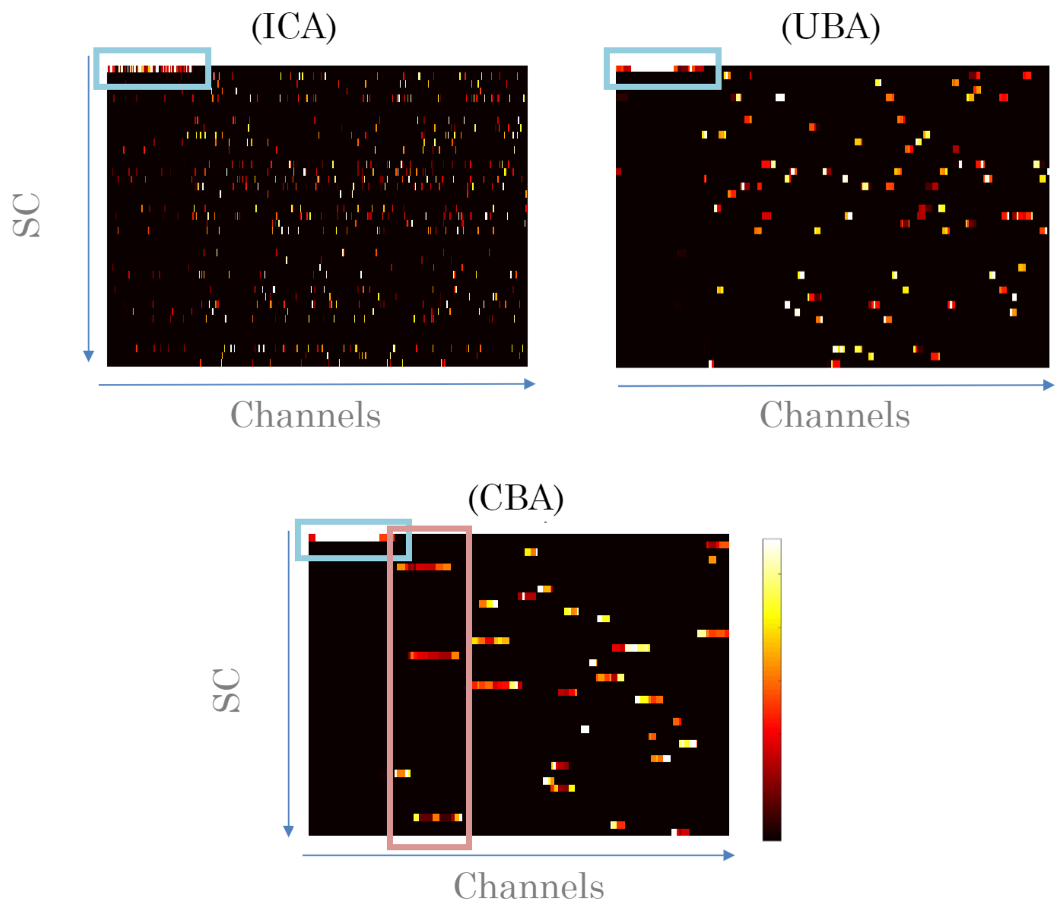


Figure 5.5: Channel allocation visualization for ICA, UBA and CBA, on which rows, cols and pixel colors represent the cell index, the channel index, and relative capacity achieved, while black means that such channel was not used by the cell.

a sub-problem of ICA. The UBA $\hat{x}_{UBA}$ search space is a subset of the original ICA $\hat{x}_{ICA}$ search space. The optimal solution available using UBA will never offer a better network performance than the optimal solution in the ICA search space. At best, these two can have the same "quality". Nonetheless, it is more likely that the optimal solution of ICA (if ever founded) would outperform the optima from the reduced search space of UBA. As the number of SC grows, the CA problem becomes more challenging to solve. UBA offers a lower complexity optimization problem, increasing the probability of finding efficient solutions by employing approximated methods. In the same way, CBA is a sub-problem of UBA, and the same inference applies. It may produce suboptimal solutions when compared to UBA or ICA optimal approaches. However, it further reduces the search space and vector dimensionality to find better solutions using the same or fewer computational resources.

5.3 Conclusion

This work proposes a CA optimization formulation that reduces the optimization problem complexity to improve UDN performance when solved using metaheuristic algorithms. The proposal is based on predefined sets of resource blocks assigned directly to users (UBA) or cells (CBA). This chapter describes that by reducing the search space into subsets of continuous channel blocks, an overall reduction of complexity is achieved. This subset of the search space is represented by a vector solution whose dimensionality is smaller than conventional approaches (like ICA) used to allocate channels in mobile networks.

In the following chapter, the performance evaluation results of the Block-based channel allocation proposed are described. For solving the challenging CA problem, the metaheuristic algorithms described in Chapter 4 are applied. The network capacity results and unsatisfied users percent achieved by the ICA, UBA and CBA schemes in a UDN scenario (more than 5000 $users/km^2$ and 1250 SC) are shown. Also, the theoretical densification limit of an average of one single user per cell is demostrated by further simulation.

Chapter 6

Channel Allocation Proposals' Evaluation

6.1 Introduction

This chapter analyses and compares the proposed CA schemes with the traditional ICA and the coalition algorithm [66]. First, the proposed UBA and CBA strategies are compared against the traditional approach ICA using some metaheuristic algorithms, composed of a balance between well-known and novel strategies. Then, an extensive analysis of the proposed CA for UDN is carried using the Genetic Algorithm (GA). Finally, the the GA/CBA's performance is compared with a recent coalition algorithm approach in [66].

When metaheuristic algorithms are applied to solve the CA problem using the conventional and proposed CA schemes, each solution of the population $\vec{S}_j$ represents a discrete Channel Allocation vector as follows:

$$\vec{S}_j = [x_1, x_2, \ldots, x_i, \ldots, x_D] \mid x_i \in [1, 2, \ldots, N] \tag{6.1}$$

where x_i either represents any one of the required channels chosen for ICA ($D = \sum_1^U N_u$), or the initial channel of the block assigned to a given user for UBA ($D = U$), or the initial channel of the block assigned to a given cell for CBA ($D = Q$). Therefore, x_i is bound between the possible indexes of the channels from 1 to N. Any channel allocation solution $\vec{S}_j$ can be expanded to a binary Channel Allocation Matrix with the corresponding algorithm. Then these matrix are used to calculate the data rate of each link in the network and the solution's fitness value. The experimentation in this chapter follows the model described in Chapter 3 and extends the experiment shown in Chapter 4.

6.2 Analysis of Network Capacity and Unsatisfied Users' performance in UDN

In this section, two of the leading Key Performance Indicators (KPI), Network Capacity and Unsatisfied Users, are evaluated in a UDN scenario. The proposed UBA and CBA schemes, evaluated through metaheuristic algorithms, are analyzed as strategies to solve the RA problem. The network parameters remain the same, as shown in Table 4.2, considering UDN with the three active user densities (5000, 10000, 15000 $users/km^2$) and a range of SC from 50 to 400 in increments of 50, which represent densification values from 1250 to 10000 $cells/km^2$. Similarly, 30 independent instances of the UDN were constructed. Cells and users' positions were generated randomly, as shown in Chapter 3. Then, the corresponding metaheuristic algorithms are applied over these network instances. The meta parameters are shown in Table 4.1 with a population of 30 solutions and 500 iterations with 15000 function calls to solve the CA problem. The network capacity and unsatisfied users' percent values shown in the following results represent the mean network capacity obtained using the algorithms over the 30 independent executions.

Tables 6.1 and 6.2 show the network capacity results and unsatisfied users percent achieved by the ICA, UBA, and CBA proposals for 200 users. This network setup allows evaluating a scenario where the number of cells is higher than the number of active users. In this case, the theoretical densification limit of an average of one single user per cell is reached when the network contains 200 SC. Therefore, at high cell densification, several SC remain inactive. Theoretically, there is a low probability of increasing the network capacity under these conditions. As shown in Table 6.1, the proposed UBA and CBA schemes achieve their maximum capacity with 200 SC in the network. Also, at this cell densification, all the users start reaching their minimum demands, as seen in Table 6.2.

On the other hand, ICA reaches its maximum capacity at 300 SC, instead of 200 SC. Even when ICA manages to allocate channels providing QoS for the users since 200 SC, the proposed UBA and CBA schemes consistently achieve higher network capacities than ICA. Therefore, these achieve better fitness values. The maximum capacity obtained under this network configuration is 4.15 Gbps, using the SMS algorithm with the proposed CBA allocation scheme and considering a UDN scenario composed of 200 SC. This capacity is twice the sum of all user data rate demands, which indicates that the allocation can maintain interference under the applied threshold. These results indicate that above 200 SC, network capacity tends to decrease with increasing cell densification. This behavior is due to the fixed interference threshold used to calculate the required channels (Eq. 3.16). The MC requires a much higher transmission power than SC. It is difficult for SC's users to efficiently share channels with MC's users, so the channel allocation algorithms naturally avoid these solutions. Due to the non-interference, the MC users reach a higher capacity than SC users. However, as more small-cells are added to the network, fewer users are associated with the MC. The spectrum is shared more efficiently among small-cell's users, although overall network capacity is reduced. Table 6.1 shows that GA and SMS algorithms offer the best solutions for the proposed UBA and CBA schemes, while WOA outperforms them with the ICA scheme.

Table 6.3 shows the network capacity results when the UDN is composed of 400 active users. With this network configuration, the theoretical densification limit is reached at 10,000 $cells/km^2$ when the maximum number SC considered is deployed. These results show a sigmoid-like increase in capacity with densification. The network capacity increases most between 100 and 200 SC deployed. When the network is composed of 50 and 100 SC, most users are associated with the MC, whose demand of required channels excels the available channels in the network, resulting in several unattended users, as shown in Table 6.4. The highest network capacity, produced by the combination of the CBA scheme and GA metaheuristic, is only 2.89 Gbps, which is lower than the sum of the users' data rate demands in this scenario (4 Gbps).

On the other hand, when 200 SC are deployed, unattended users are reassigned from the MC to the newly deployed SC. The network capacity quickly increases, successfully serving a higher number of users. At this cell densification, only CBA provides QoS for more than 90% of the users. In this scenario, densification loses importance at more than 300 cells. The highest capacity reached with this cell density (6.61 Gbps, obtained by the allocation using the SMS

Table 6.1: Performance comparison of network capacity (Gbps) using metaheuristic algorithms for solving Channel Allocation in UDN with 200 active users, range of deployed cells from 50 to 400, and allocation vectors ICA, UBA and CBA.

Scheme	Alg / Cells	50	100	200	300	400
ICA	GA	1.9231	2.3287	3.5034	3.5917	3.5415
	PSO	1.7439	2.1641	3.0814	3.3662	3.3249
	DE	1.7509	2.1525	3.1396	3.3517	3.3317
	SMS	1.7836	2.3505	3.5086	**3.6616**	**3.5886**
	WOA	**1.9575**	**2.9052**	**3.5559**	3.4774	3.3885
	RAND	1.7824	2.2200	2.9867	3.2817	3.2705
UBA	GA	**2.1003**	**3.1275**	**4.0485**	3.8933	**3.748**
	PSO	1.8462	2.1458	3.4829	3.4423	3.3811
	DE	1.8471	2.4486	3.4849	3.491	3.4494
	SMS	2.0243	3.0526	4.0204	**3.8955**	3.7358
	WOA	1.9796	3.0515	3.7798	3.684	3.5628
	RAND	1.8255	2.5374	3.2529	3.3885	3.3613
CBA	GA	**2.0847**	**3.5654**	4.1295	3.9529	3.7772
	PSO	1.8691	2.5848	3.5104	3.5205	3.3959
	DE	1.9882	2.8488	3.6257	3.5562	3.5016
	SMS	2.0468	3.5115	**4.1505**	**3.9605**	**3.7992**
	WOA	1.917	3.2461	3.9017	3.7419	3.5922
	RAND	1.8728	2.5853	3.3755	3.4596	3.3797

Table 6.2: Performance comparison of unsatisfied users percent using metaheuristic algorithms for solving Channel Allocation in UDN with 200 active users, range of deployed cells from 50 to 400, and allocation vectors ICA, UBA and CBA.

Scheme	Alg / Cells	50	100	200	300	400
ICA	GA	**65.075**	36.375	**0**	**0**	**0**
	PSO	70.133	55.366	2.350	**0**	**0**
	DE	69.350	53.333	0.150	3.3517	3.3317
	SMS	70.956	42.316	0.033	**0**	**0**
	WOA	70.833	**23.300**	**0**	**0**	**0**
	RAND	69.116	51.500	2.333	**0**	**0**
UBA	GA	**64.516**	**30.866**	**0**	**0**	**0**
	PSO	71.216	62.216	5.916	1.916	0.550
	DE	70.939	47.133	7.533	1.783	0.033
	SMS	68.816	32.180	0.016	**0**	**0**
	WOA	68.033	24.916	0.316	**0**	**0**
	RAND	70.283	46.500	16.216	5.300	1.716
CBA	GA	**64.737**	**9.133**	**0**	**0**	**0**
	PSO	69.950	46.433	6.433	0.933	0.250
	DE	66.533	35.133	3.633	0.450	**0**
	SMS	66.375	15.075	**0**	**0**	**0**
	WOA	69.583	19.150	0.266	0.033	0.016
	RAND	69.283	46.616	12.900	3.700	1.583

algorithm and CBA scheme) improves by only 0.03% with 100 more SC. The block-channel-based schemes' benefits can be observed as the best solutions found UBA outperforms ICA by an average of 8.34%. In comparison, CBA has an average improvement of 13.32% over ICA, even when both can provide QoS.

Finally, Tables 6.5 and 6.6 show the results for the case of UDN with 600 active users. This network configuration is an example in which the density of users remains higher than cell densification. Therefore the high demand justifies the cell densification, which is expected in dense urban cities. The results show that network capacity increases with cell densification without reaching a clear upper limit due to the users' high density. The theoretical limit of the densification is not reached. For this scenario, the sum of the users' data rate demands is 6 Gbps. Only densifications of 300 SC and above are able to reach this network capacity and provide most users with their service demands.

This scenario better shows the block-based allocation proposal's benefits due to the high complexity of solving the CA optimization problem when considering these high user and cell densities. The average capacity improvements over ICA are 11.82% when using UBA and 17.89% with the CBA scheme.

These results show that using the unbiased random algorithm; the capacity is increased with the UBA and CBA proposals compared to ICA, independently of the users and cells densification. However, ICA tends to offer lower unsatisfied users percent at high densification, and therefore higher value of the fitness function. It seems to indicate that the ICA's solution set contains several local optima that produce a low percent of unsatisfied users and network capacity. Simultaneusly, UBA and CBA are biased to prefer high network capacity than reducing the unsatisfied users percentage. However, for GA and SMS algorithms, the unsatisfied users' percent is always improved using the block-channel-based proposals while also providing high network capacities. However, this is not

Table 6.3: Performance comparison of network capacity (Gbps) using metaheuristic algorithms for solving Channel Allocation in UDN with 400 active users, range of deployed cells from 50 to 400, and the three types of allocation vectors: ICA, UBA and CBA.

Scheme	Alg / Cells	50	100	200	300	400
ICA	GA	1.9408	2.382	4.4642	5.6101	5.8974
	PSO	1.8159	2.332	3.7481	5.42	5.6014
	DE	1.8448	2.3183	4.2394	5.2717	5.5312
	SMS	1.9003	2.4528	4.3782	**5.6323**	**5.9566**
	WOA	**2.0624**	**2.5775**	**5.0757**	5.6021	5.7439
	RAND	1.7665	2.3520	4.0110	5.1633	5.4499
UBA	GA	**2.2662**	**2.7139**	**5.4765**	**6.2548**	6.3371
	PSO	1.9032	2.3218	4.7222	5.5076	5.7554
	DE	1.9182	2.4134	4.6772	5.5817	5.7953
	SMS	2.2053	2.7067	5.1758	6.2434	**6.3651**
	WOA	2.1445	2.5573	5.3671	5.9265	5.9286
	RAND	1.8966	2.4384	4.2796	5.3892	5.6644
CBA	GA	**2.2854**	**2.8987**	5.6008	6.5198	6.5517
	PSO	1.9583	2.4529	4.7543	5.7389	5.8314
	DE	2.0633	2.5142	5.0906	5.7779	5.8979
	SMS	2.2096	2.7615	5.751	**6.6192**	**6.6217**
	WOA	2.0681	2.4933	**5.7874**	6.1061	6.1282
	RAND	1.9431	2.4841	4.6057	5.5939	5.7306

Table 6.4: Performance comparison of unsatisfied users percentage using metaheuristic algorithms for solving Channel Allocation in UDN with 400 active users, range of deployed cells from 50 to 400, and the three types of allocation vectors: ICA, UBA and CBA.

Scheme	Alg / Cells	50	100	200	300	400
ICA	GA	**83.312**	**79.225**	22.258	0.616	0.004
	PSO	85.050	80.666	55.400	2.375	0.308
	DE	84.633	80.058	37.308	4.450	0.591
	SMS	85.675	80.550	27.708	**0.333**	**0**
	WOA	86.191	80.091	**16.491**	0.791	0.008
	RAND	85.508	80.116	46.391	7.058	1.483
UBA	GA	**82.608**	**78.841**	**14.316**	**0.937**	**0.050**
	PSO	84.966	82.250	36.216	15.550	7.675
	DE	84.750	80.950	36.958	15.158	7.000
	SMS	83.350	79.400	24.983	1.116	0.058
	WOA	83.350	79.791	17.541	4.250	2.300
	RAND	84.833	80.050	47.316	20.275	11.491
CBA	GA	82.887	**77.041**	**8.783**	0.112	**0**
	PSO	85.616	80.975	36.358	11.333	6.700
	DE	84.033	78.541	27.483	11.275	5.641
	SMS	**82.691**	77.775	11.316	**0.100**	**0**
	WOA	84.433	80.191	9.650	2.875	1.075
	RAND	84.916	80.183	40.058	16.883	10.516

Table 6.5: Performance comparison of network capacity (Gbps) using metaheuristic algorithms for solving Channel Allocation in UDN with 600 active users, range of deployed cells from 50 to 400, and three schemes of allocation vectors: ICA, UBA and CBA.

Scheme	Alg / Cells	50	100	200	300	400
ICA	GA	2.0617	2.6673	**4.8743**	6.7851	7.4166
	PSO	1.8898	2.5762	3.8821	6.8419	7.415
	DE	1.9522	2.5866	4.4748	6.7516	7.2936
	SMS	1.9823	2.6779	4.4265	6.7948	7.587
	WOA	**2.1298**	**2.8173**	4.7639	**7.1107**	**7.6399**
	RAND	1.8545	2.6700	4.0685	6.4915	7.1904
UBA	GA	**2.4016**	**3.0723**	5.1339	**7.6267**	8.1501
	PSO	2.002	2.5916	5.0107	6.8558	7.571
	DE	2.0143	2.7426	4.6311	6.8029	7.589
	SMS	2.38	3.054	5.1833	7.5323	**8.2705**
	WOA	2.2484	2.8103	**5.7037**	7.5402	7.8475
	RAND	1.9685	2.7605	4.2061	6.7561	7.4382
CBA	GA	**2.4635**	**3.3174**	4.8228	8.1125	8.6097
	PSO	2.1166	2.7877	4.7534	7.244	7.7365
	DE	2.2026	2.8126	5.2856	7.3768	7.8312
	SMS	2.3898	3.1138	4.9293	**8.1843**	**8.7331**
	WOA	2.1807	2.7983	**5.948**	7.6837	8.1151
	RAND	2.0774	2.8283	4.5841	6.9337	7.6226

Table 6.6: Performance comparison of unsatisfied users percentage using metaheuristic algorithms for solving Channel Allocation in UDN with 600 active users, range of deployed cells from 50 to 400, and three schemes of allocation vectors: ICA, UBA and CBA.

Scheme	Alg / Cells	50	100	200	300	400
ICA	GA	**88.660**	**86.170**	**56.941**	**14.366**	3.850
	PSO	90.255	87.338	80.900	22.338	7.861
	DE	89.808	87.275	70.450	21.066	9.100
	SMS	90.627	86.727	71.255	17.188	**2.461**
	WOA	91.122	87.105	72.161	18.388	4.088
	RAND	90.622	86.861	78.827	30.672	12.29
UBA	GA	**88.030**	85.908	51.150	**12.286**	5.650
	PSO	90.111	88.416	62.577	32.705	18.222
	DE	90.072	87.466	66.782	32.375	16.950
	SMS	88.950	**85.394**	60.227	15.361	**1.883**
	WOA	89.105	87.250	**48.922**	17.533	10.400
	RAND	90.238	86.555	75.750	34.677	21.988
CBA	GA	**88.008**	**84.330**	66.994	5.880	1.1389
	PSO	89.655	87.300	67.500	26.150	15.883
	DE	89.022	85.911	57.455	23.472	14.888
	SMS	88.200	84.800	64.438	**5.333**	**0.283**
	WOA	89.027	86.733	**43.738**	15.344	7.022
	RAND	89.772	86.938	70.366	32.061	20.366

true for all the metaheuristic algorithms.

In order to analyze the performance of the proposed allocation schemes, Fig. 6.1 shows the network capacity as a function of cell densification. The network capacity values were calculated as the average capacities reached by the metaheuristic algorithms under the network's density setups and allocation schemes considered in the evaluations. For example, the blue lines represent the UDN scenario with 600 active users. The capacities shown at 400 cells in Fig. 6.1 represent the average capacity between all the metaheuristic algorithms results shown in Table 6.5 for each allocation scheme. In this model, the network capacity is capped by the network's demands, as a fixed number of channels is allocated to each user. Therefore, as the figure shows, the mean network capacity grows with active users and SC. It can be noted from Fig. 6.1 and Tables 6.1, 6.3 and 6.5 that in most cases, the CBA scheme outperforms ICA and UBA, without regard to the metaheuristic algorithm used. This performance follows the idea that the proposed block-channel-based allocation schemes reduce the channel allocation process's complexity. Also, increasing the likelihood that approximate optimization methods will find suitable quality solutions with fewer computational resources.

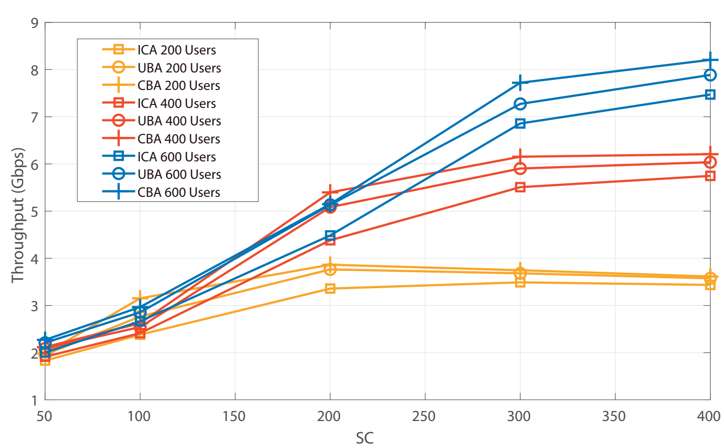
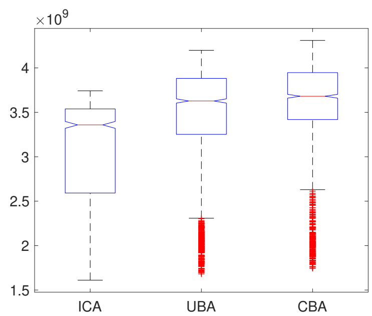


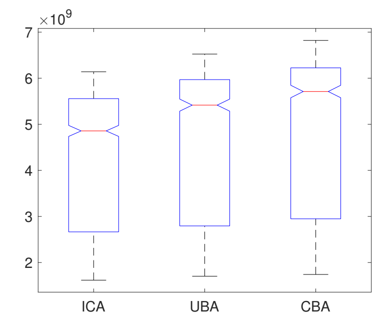
Figure 6.1: Comparison of the average capacity achieved between the applied metaheuristics algorithms vs the network densification for the CBA, UBA and ICA schemes and the three users densifications considered.

A Kruskal-Wallis test was used to obtain the statistical significance of the experimentation results shown in Fig. 6.1. This test is a non-parametric version of classical one-way ANOVA that compares the medians of the groups to determine if the samples come from the same population [93]. This test allows us to deny the null-hypothesis, i.e., that the results obtained from the metaheuristic algorithms using the proposed channel allocation schemes belong to the same distribution of the previously proposed ICA scheme. The p-value results of the Kruskal-Wallis test and the visual representation given by Matlab® for each of the three user densifications considered are shown in Fig. 6.2. It can be noted that the null hypothesis is rejected and that the UBA and CBA median network capacity is always improved in comparison with the ICA. A Tukey's Honestly test is applied to make a paired comparison of the ICA, UBA, and CBA schemes. In this test, the p-value is for the hypothesis test that the corresponding mean difference is not equal to zero. As shown in Table 6.7, all the configurations score a p-value lower than 0.05, showing the statistical significance of the mean difference. Similarly, Fig. 6.3 shows the individual metaheuristic algorithms' network capacities distribution with the three CA schemes. This result indicates that each Algorithm's solutions are improved in terms of network capacity with the proposed UBA and CBA schemes.

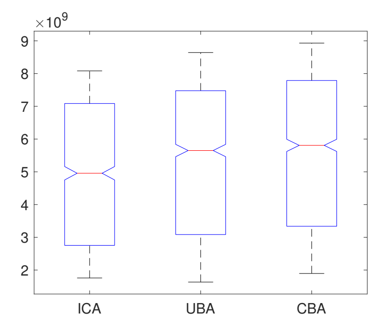
The results shown in Tables Tables 6.1, 6.3; and 6.5 show that GA generally obtains the best results for the UBA scheme. It is surpassed by the SMS algorithm only when 400 SC are considered. Similarly, with the CBA scheme, GA dominates when the optimization problem presents low dimensionality (i.e., 50 and 100 cells). In contrast, for higher cell densities, the SMS and WOA algorithms obtain better network capacity results. Based on the number of active users and 400 SC, the SMS algorithm finds the network's highest capacity for



(a) 200 Users with $p = 9e - 135$



(b) 400 Users with $p = 1e - 52$



(c) 600 Users with $p = 1e - 24$

Figure 6.2: Kruskal-Wallis test for the CBA, UBA and ICA schemes and the 200, 400 and 600 users.

the CBA scheme. It might be due to the intense exploration phase of the SMS algorithm. Because the proposed CBA scheme offers consistently offers the highest network capacities, we select it to analyze the metaheuristic algorithms' evolutionary process. Figure 6.4 shows the convergence plot of the metaheuristic algorithms for a network consisting of 600 users. The CA problem results in Fig. 6.4a represent the lowest dimensionalities (i.e., 50 SC) considered in the CA performance evaluation. The GA algorithm outperforms the others consistently throughout the evolutionary process.

Figure 6.4b shows the highest dimensionality for the CBA scheme, i.e., 400 SC. In this case, the CBA scheme using the GA algorithm offers better solution channel allocations in the first 200 iterations. However, the SMS algorithm results outperform those obtained by the GA algorithm, improving the best fitness found on subsequent iterations.

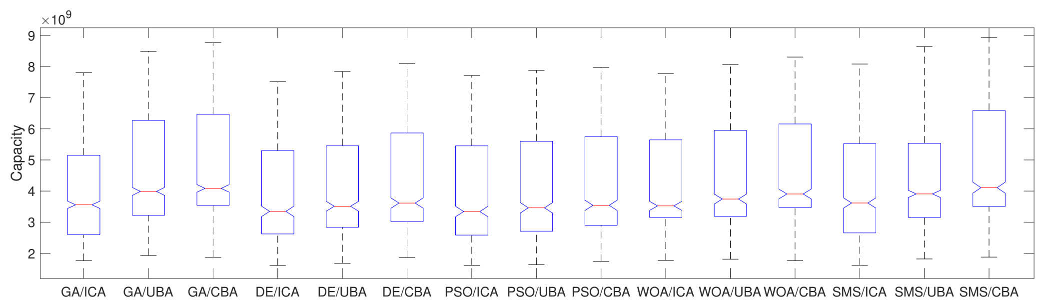


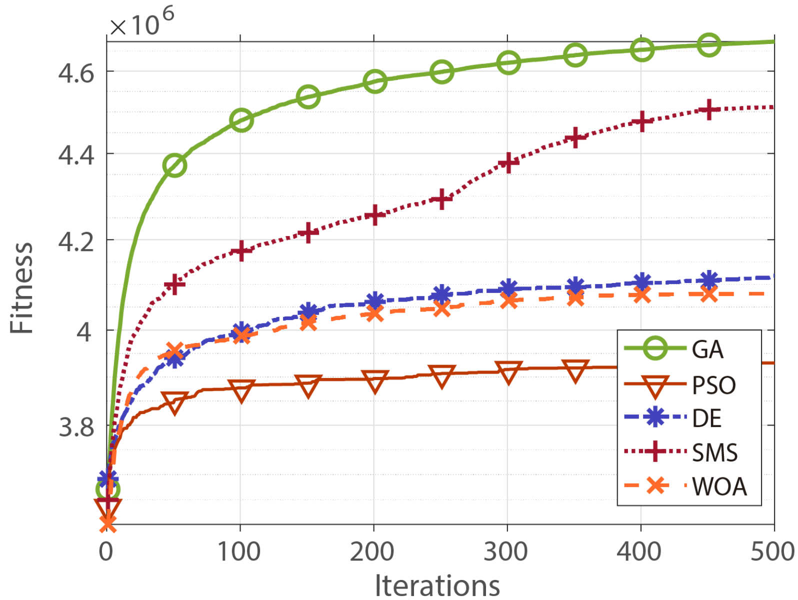
Figure 6.3: Box plot of the network capacity achieved by the metaheuristics algorithms using the CBA, UBA and ICA schemes.

It can be seen in Fig. 6.4 that for the first 200 iterations, the GA and WOA algorithms quickly evolve the initial random solutions into a potential local optima neighborhood. After this point, the information recollected in promising zones is exploited to improve the best solution found so far. By contrast, the CBA scheme solved using SMS algorithm dedicates a significant amount of the available iterations in a predominant exploration phase [89], extending over 250

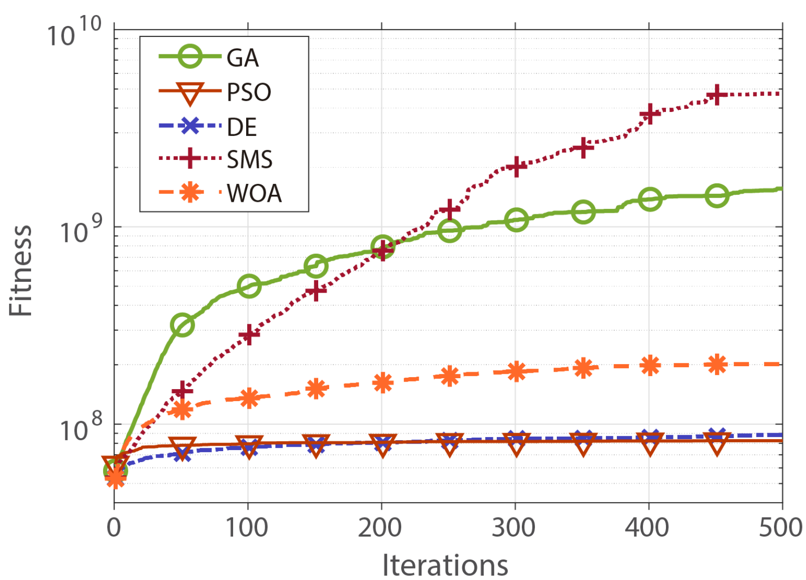
Table 6.7: Tukey's Honestly test for the CBA, UBA and ICA schemes and the 200, 400 and 600 users.

Users	Scheme	Scheme	Lower	Mean Difference	Upper	p-value
200	ICA	UBA	-1059.55	-947.54	-835.53	9.56e-10
	ICA	CBA	-1329.75	-1213.34	-1096.93	9.56e-10
	UBA	CBA	-377.77	-265.80	-153.83	8.00e-08
400	ICA	UBA	-578.05	-469.08	-360.11	9.56e-10
	ICA	CBA	-832.81	-723.75	-614.68	9.56e-10
	UBA	CBA	-363.63	-254.66	-145.69	1.30e-07
600	ICA	UBA	-373.70	-267.84	-161.98	9.99e-09
	ICA	CBA	-565.01	-460.38	-355.75	9.56e-10
	UBA	CBA	-291.81	-192.54	-93.26	1.63e-05

iterations. It can be observed in Fig. 6.4b that this produces a slower but steady increase in the fitness of the best solution through the iterations. The three stages of the SMS algorithm seem adequate for solving the CA problem when there is a high SC density, i.e., when the problem's complexity is more elevated. On the other hand, when the GA algorithm is applied, it finds quality solutions using lower computational resources, in this case, 200 or fewer iterations.



(a) 50 SC



(b) 400 SC

Figure 6.4: Comparison of the evolutionary optimization process of the CA optimization problem in UDN with 600 users, using the CBA allocation scheme for all the metaheuristic algorithms considered.

6.3 Extensive Analysis of the data rate QoS, Spectrum Efficiency, Bandwidth Reuse, and Unattended Users Percentage

A more extensive analysis of the proposed Block-Channel-based RA schemes is performed in this subsection. Some other Performance Indicators such as data rate QoS, Spectrum Efficiency, Bandwidth Reuse, and Unattended Users Percentage are discussed. The KPI results reported here were obtained applying the GA since it is the one that presented the best performance for solving the CA problem for UDN in the previous section.

The positions of cells and users, data rate demands, the user association's application and calculation of required channels follow the configuration specified in Chapter 3. Table 6.8 shows the values of the simulation parameters used for the experiments carried out. The number of deployed SC ranges from 10 to 500, representing densification values from 250 to 12500 $cell/km^2$, respectively. We consider 500 active UE uniformly distributed in the UND scenario and two UE data rate demands: (i) uniformly random distribution ranging between $[R^{\min}, R^{\max}]$, and (ii) edge case, where all UE demand a maximum data rate ($R_u = R^{\max}$). Each UE is connected with the geometrically closest SC. For better demonstration, a downlink LTE system is considered, OFDMA RB are reused by different MC and SC such that downlink UE suffer inter-cell and intra-cell interference.

Figure 6.5 shows UDN capacity regarding the network densification achieved when ICA, UBA, and CBA schemes are applied for the two cases of user data rate demands. Fig. 6.5a shows the performance evaluation regarding the net-

Table 6.8: Parameters for heterogeneous users' data rate experiment.

Parameter	Value
Bandwidth	200 MHz
User Data Target	[5, 10] Mbps
Interference Threshold	1e-8 Watts
Noise Per Channel	1e-10 Watts
Ch. SINR Threshold (σ)	8db
Path Loss (α)	3
Number of SC	[10 - 300]
Active Users in the Network	[50 - 700]
MC Power	40 Watts
SC Power	5 Watts
Minimum Distance Between SC	13 meters

work densification for the heterogeneous scenario, i.e., when UE data rate demands are distributed between [5, 10] Mbps. In contrast, Fig. 6.5b shows the performance results when the maximum UE data rate demand is considered (Edge scenario). In Figures 6.5a and 6.5b, it can be noted that the CBA scheme is the channel allocation proposal that achieves the best network capacity for UDN. Also, CBA achieves the aggregated users' demands, an average of 3.75 Gbps and 5 Gbps for the two UDN cases, respectively, with only 50 SC in the network. At the same time, ICA and UBA required a significantly higher cell density. For a high cell density (200 SC or 12500 $cell/km^2$), considered as a high cell density, the data rate obtained with CBA and UBA proposals outperforms conventional ICA by an average of 13.9% and 3.14%, respectively (see Fig. 6.5a).

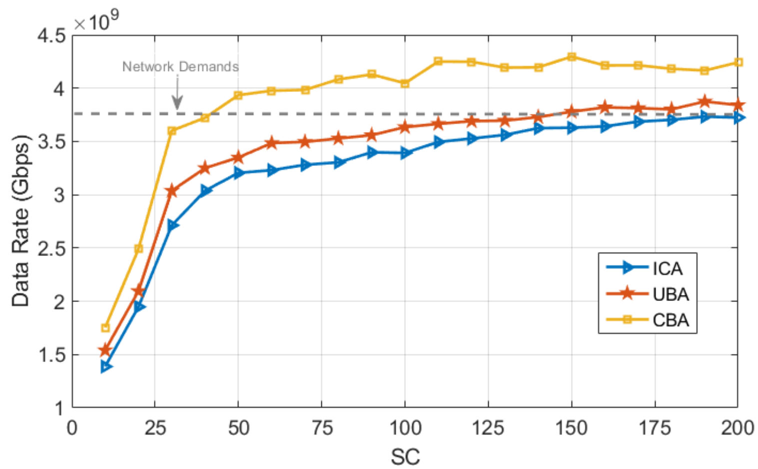
On the other hand, Fig. 6.5b shows that the CBA scheme offers an average improvement over ICA of 8.12%. In comparison, UBA provides an increase of 1.55%. For this experiment, the dimensionality of the three CA schemes is: (i) higher than 1,000 for ICA, (ii) 500 dimensions for UBA, and (iii) a range from 50-200 for CBA with regards to cell density. These results suggest that the GA is better at finding more efficient CA solutions with the dimensionality reduction achieved by the proposed UBA and CBA schemes.

Figure 6.6 shows the network capacity results for 100 and 200 SC. With users' demands of 10 Mbps as the number of active users increases. As expected, network capacity increases with the number of users and their demands. More required channels are allocated in the network to reach the demands. We can observe that the CA schemes manage to increase the network capacity regarding the network demands monotonically. The proposed CBA method consistently attains the best results. Also, it is visible that the CBA's network capacity with 100 SC is for most user densities higher than ICA with 200 SC. This result demonstrates that ICA with 100 SC could not reach the network capacity demands at high user densities, thus increasing cell densification. However, for 100 cells, the CBA scheme manages to achieve these demands even for 500 active users in the network. It shows the importance of using an efficient CA algorithm to maximize network capacity and fully take advantage of network densification.

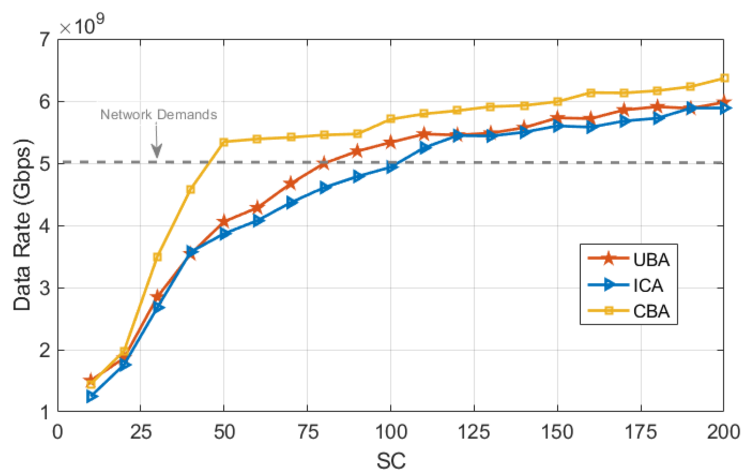
For analyzing the Spectrum Efficiency (SE), Bandwidth Reuse, interference level, and Unattended Users Percentage (UPP) in UDN, we consider a more complex UDN scenario deploying 700 users and 300 SC. Because of the problem's high complexity, none of the proposed or traditional CA schemes are likely to reach the global optima. Therefore, the quality of the solution should be related to the number of computational resources invested in the optimization process. In these evaluation, GA executes over 1,000 iterations using a population of either 10 or 100 solutions to evaluate the computational resources required to achieve efficient CA solutions. The Fitness Function Calls (FFC) measure the number of computational resources used in execution; it represents the number of times the network calculates the users' capacity. To identify the performance evaluation of each channel allocation scheme, we use the following format: Name of the channel allocation scheme and the number of solutions considered in the GA population. For example, ICA-10P indicates the results obtained using the ICA scheme and ten solutions in the population, translate into a total 10,000 FFC. The experiment is carried out 30 times for each CA method. Figure 6.7 and Table 6.9 show the mean fitness value of the best solutions.

The results in Figure 6.7 settle that the CBA-10 scheme presents the best performance, outperforming the fitness value by 400% concerning the rest of the channel assignment schemes. The UBA-10 channel allocation proposal reaches the best fitness value obtained by ICA-10 with less than a third of the FFC. At the same time, CBA-10 finds a similar solution with only 60 FFC. Similarly, CBA-10 requires 100 iterations to achieve the same best fitness value obtained by the UBA-10 scheme in 1000 iterations. The above demonstrates that the CBA channel allocation scheme offers a significant advantage over UBA and conventional ICA schemes.

Table 6.9 proves that the Spectrum Efficiency (SE), i.e., the relationship between network capacity and bandwidth, is improved with the UBA-10 and CBA-10 proposals by an average of 8% and 14% over ICA-10, respectively. Bandwidth Reuse (%) represents the percentage of channels successfully allocated concern-



(a) UE data rate demands distributed between [5, 10] Mbps



(b) Maximum UE data rate demands, 10 Mbps

Figure 6.5: Performance comparison of UDN capacity applying ICA, UBA and CBA proposals vs network densification with 500 active users.

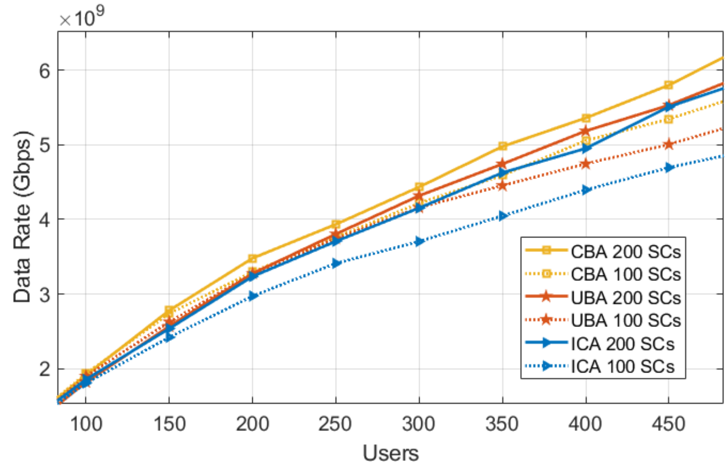


Figure 6.6: Comparison of ICA, UBA and CBA in capacity vs users density, of a UDN with 100 and 200 SC with maximum data rate requirements for all users.

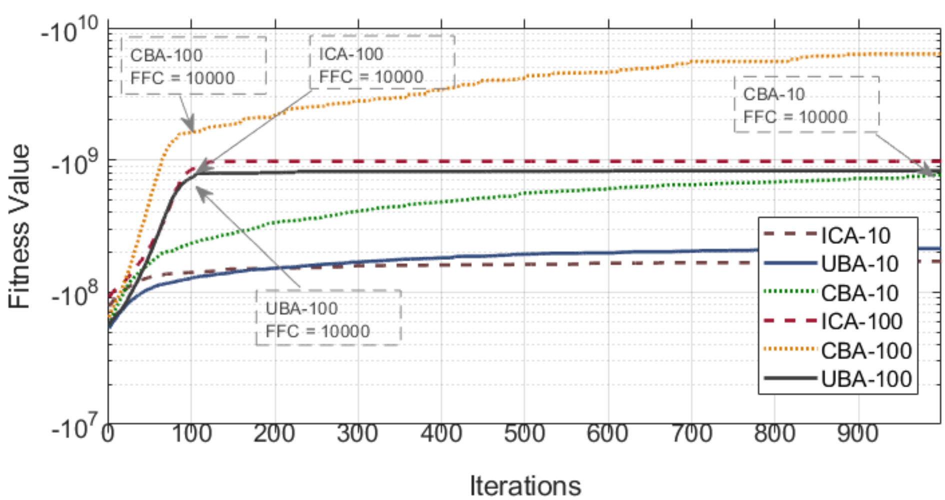


Figure 6.7: Comparison of the GA evolutionary process when using ICA, UBA and CBA schemes on a UDN with 700 users, and 300 SC.

ing the available number of channels N. In this sense, channel allocation proposals UBA-10 and CBA-10 present similar performance as ICA does. The above indicates that the network capacity's improvement is mainly due to the channel allocation strategy avoiding interference and not the maximum number of channels assigned. Therefore, using the UBA and CBA schemes highly reduce the interference in comparison to ICA. However, UBA-10 halves interference, but the UUP (Unattended Users Percentage) value is only slightly improved. The UUP (%) indicates the users' fraction that fails to achieve their data rate demands. On the other hand, using the CBA-10 scheme generates five times less interference than when using the conventional ICA-10, which significantly reduces the UUP and increases the overall fitness value, as Eq. (3.12) indicates.

It is observed in Fig. 6.7 that the ICA-100 and UBA-100 schemes reach higher fitness value using the same 10,000 FFC than the CBA-10 scheme at the end of the 1000 iterations. The above establishes that to solve the CA problem in UDN, the GA should use many solutions in the population and a low number of iterations. This behavior can be observed in CBA-100 results. It outperforms the fitness value of the CBA-10 scheme by 212% using the same amount of FFC. Unlike when ten solutions are used in the GA population, in this case, the ICA-100 scheme achieves a better fitness value than UBA-100. However, UBA-100 achieves better SE and UPP results than ICA-100. From Table 6.9, it can also be observed that by increasing the computing resources by 10x, the fitness function value is improved by 5.7x, 3.8x, and 8.2x for the ICA-100, UBA-100, and CBA-100 schemes, respectively. Figure 8 shows that the CA methods ICA-10/100 and UBA-10/100 converge prematurely to local optima. Simultaneously, CBA-10/100 finds "good quality" solutions quickly, and the fitness value continues to improve as GA continues its evolutionary process. It suggests that the proposed

Table 6.9: Numerical results of Comparison of allocation schemes on the evolutionary process with 700 users, and 300 FCs.

Metric	ICA-10	UBA-10	CBA-10	ICA-100	UBA-100	CBA-100
SE	46.5157	50.1005	52.8896	48.1405	51.7251	54.5420
Interference	1.97e-4	8.92e-5	4.04e-5	1.51e-4	5.68e-5	2.83e-5
UUP %	7.54	6.03	1.82	1.95	1.7762	0.14
B. Reuse %	587.77	591.44	593.33	583.73	595.7400	592.91
F. Value (-)	1.70e+8	2.14e+8	7.75e+8	9.76e+8	8.30+8	6.41e+9
FFC	10,000	10,000	10,000	100,000	100,000	100,000

CBA scheme offers a better opportunity for the optimization algorithms to find suitable solutions and take advantage of the computational resources invested in the optimization. The CBA scheme is considered a sub-problem of the UBA and ICA schemes. It can be concluded that the improvement in network capacity is due to the low complexity achieved in the optimization problem. From this low complexity, it is possible to find better channel assignments to users and cells, which reduces interference in the UDN, increases capacity and reduces the number of unattended users (UUP).

6.4 Performance Analysis of Cell-Channel-Block Allocation (CBA) and Coalition Algorithm

We evaluate and analyze our Block-based channel allocation proposal and the Coalition Algorithm reported in [66] for comparison purposes. The Coalition Algorithm represents a similar approach to deal with the CA problem in UDN with reduced complexity. The considered network configuration are summarized in Table 6.10. The Coalition Algorithm considers a particular case of ultra-dense scenarios like stadiums, where a close area of 200x200 meters is split using grid geometry. Each sub-area contains a single SC, and each serves from 1 to 4 users randomly placed over the space. The cells can only serve the users in their sub-areas in a closed fashion. Although it considers a single distant MC, their results show that the MC interference is insignificant compared to that generated between small cells in the SC coverage area. Our work uses a different path loss model and does not consider random behaviors for channel gains. Therefore exact calculations might be affected by it. Nevertheless, as interference is the main factor affecting the UDN performance, overall behavior should remain unaffected. In [66], the authors do not ensure QoS for each user. For this evaluation, we considered three user data requirements: 10Mbps, 50Mps, and 100Mbps. For the channel allocation, we use the CBA scheme, since it is the one that presented the best results in the previous evaluations. For the GA, we consider 30 solutions in the initial population and 1000 iterations. The results shown in Figures 6.8 and 6.10 were averaged over a total of 30 independent realizations.

Figure 6.8 shows the comparative results of the Spectral Efficiency (Fig. 6.8a) and average cell capacity (Fig. 6.8b) obtained from the CBA scheme and Coalition Algorithm for the three QoS requirements considered. As shown in Fig. 6.8a, Spectral efficiency (SE) increases with the number of cells for both approaches. This increase is due to each new SC in the network serves only 1 to 4 users, which increases user diversity, and network demand. However, Fig. 6.8b shows that the average cell capacity decreases concerning network densification. This capacity lessening is because when the spectrum is shared between more SC, the interference increases accordingly. Therefore, the

Table 6.10: Parameters for the coalition reference work in [94].

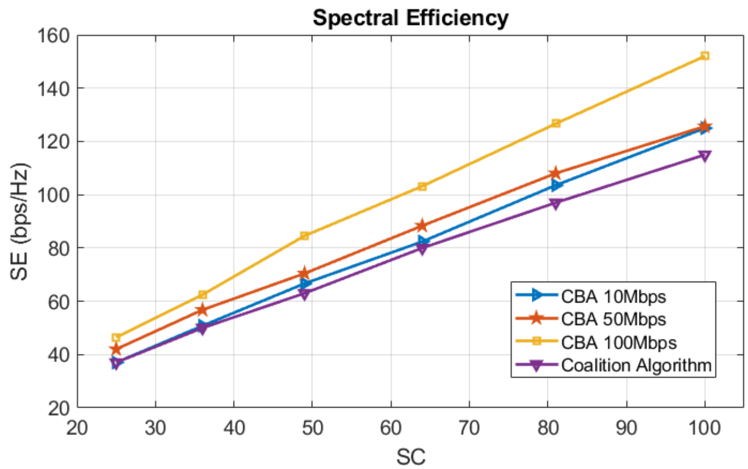
Parameter	Value
Bandwidth	100 MHz
Number of Channels	10
User Data Target	[10, 50, 100] Mbps
Interference Threshold	1e-5 Watts
Noise Per Channel	1e-10 Watts
Ch. SINR Threshold	-5db
Number of SC	[25, 100]
Number of Users per SC	[1, 4]
SC Power	0.2 Watts

individual capacity of users is reduced as the network scales in cells and users. Also, suppose QoS requirements are considered for a certain densification point. In that case, the network will not serve all users, as shown in Fig. 6.9.

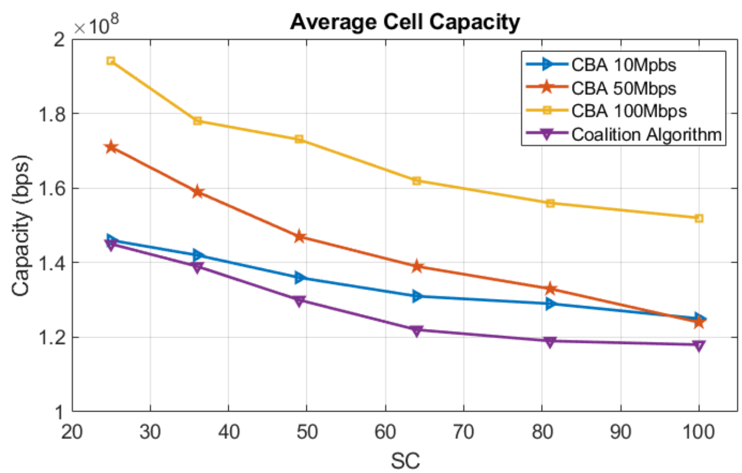
Finally, 6.10 shows the performance of the Sub-Channel Allocation Rate (SAR). SAR represents the ratio of allocated channels to total available channels. As each SC can reuse the whole bandwidth, the SAR is calculated as follows:

$$SAR = \frac{\sum_{q=1}^{Q} N_q}{N * Q} \tag{6.2}$$

We can observe in Fig. 6.10 that the Coalition Algorithm achieves a higher SAR than the CBA proposal, and it remains constant throughout the densification. This behavior is because the Coalition Algorithm applies a post-coalition channel allocation strategy. This strategy assigns the unassigned channels to users



(a)



(b)

Figure 6.8: Comparison of the (a) Spectral efficiency (SE) and (b) cell capacity obtained by CBA proposal and by Coalition Algorithm proposed in [66].

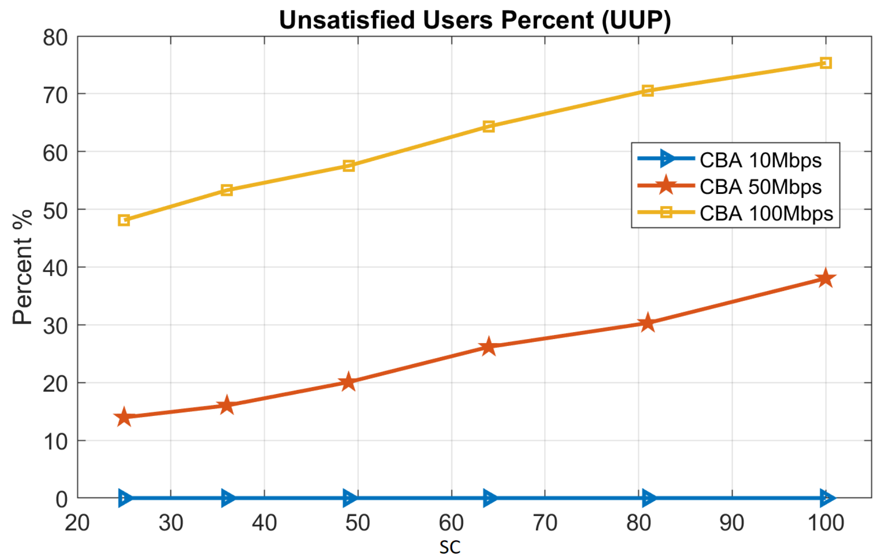


Figure 6.9: Comparison with the coalition CA in unsatisfied users percent with regards network densification.

as long as this approach increases the network's capacity. When applying the proposed CBA scheme, the SAR results decrease concerning the SC's densification when the user data rate requirements cannot be met (see 50 and 100 Mbps cases). For example, to meet the 100 Mbps of user data rate with the CBA scheme, it is necessary to allocate to users more channels resulting in high SAR. However, the average SINR of each used channel is reduced with the network densification due to the ever so high interference, as shown in Fig. 6.8b. Although it is required to assign more channels to the network (which increases the SAR index), we count only those channels that offer the highest SINR levels (i.e., SINR values are above the limit threshold depicted in Eq. (3.8)). On the other hand, when the user data rate demand is 10 Mbps, the CBA scheme assigns a smaller number of channels to each SC. Therefore, the SAR value is lower than the others. Still, the capacity values are higher than those obtained by the Coalition Algorithm (see Figure 6.8). Also, in this case, SAR performance appears to be unaffected by densification. This behavior is because the bandwidth is divided into only ten channels. Therefore, the users can reach their demands with a single channel. As each new cell in the simulations has its own 1 to 4 users, the number of channels required per cell remains 2.5 on average.

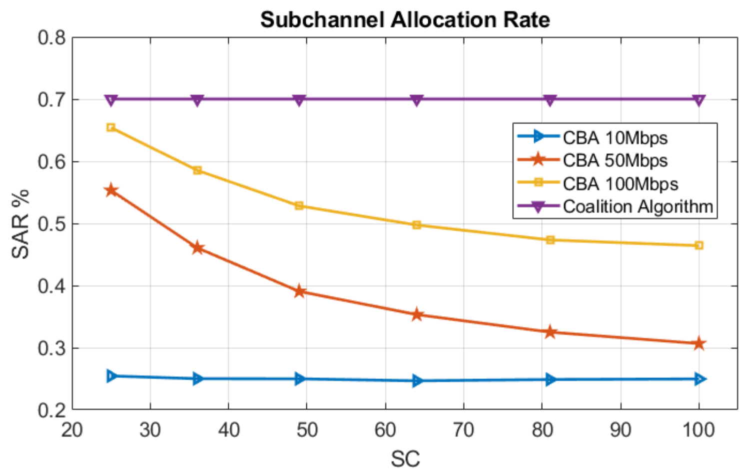


Figure 6.10: Comparison performance of Sub-Channel Allocation Rate (SAR) obtained by the CBA scheme and Coalition Algorithm proposed in [66]

6.5 Conclusion

This chapter analyses and compares the proposed CA schemes with the traditional ICA and the coalition algorithm [66]. The results show that the proposal's reduction in complexity allows the optimization algorithms to find more efficient allocation solutions without increasing the computational resources invested in the optimization process. The proposed UBA and CBA strategies are compared against the traditional approach ICA using some metaheuristic algorithms, composed of a balance between well-known and novel strategies. The results indicate that the proposed CA schemes improve the algorithms' average performance in all the network settings considered in this study.

The results show that GA with the proposed UBA and CBA makes more efficient use of the network's channels, reaching higher network capacities and using lower computational resources. As pointed out in these experimental results, GA/ICA seems unable to attain the network's aggregated demand at 500 users, which implies that further densification is advisable. Nonetheless, GA/UBA and GA/CBA network capacity shows that using this demand is achievable without increasing the densification.

The coalition algorithm's comparison results that the GA/CBA's performance to solve the CA problem in UDN is well suited to provide competitive results compared with the state of the art CA approaches in UDN. It can be observed that even if the cell densification grows linearly with the network demands, the user's capacity remains monotonically descendent. On the other side, the Sub-Channel Allocation Rate (SAR) evaluations indicate that high SAR is not necessarily correlated to high network or user capacities. Nonetheless, the post-coalition channel allocation strategy seems to improve CA solutions locally by increasing channel utilization.

Chapter 7

Conclusion

7.1 Thesis summary and General Conclusions

Network densification is crucial for achieving B5G goals. Current research efforts on UDN point out that efficient RA algorithms are required to exploit the benefits of high densification. These tailored RA efforts should deal with entangled interference and high computational complexity due to the unprecedented network scale. From the literature review, it has been noticed an opportunity to improve the RA by proposing new approaches that take advantage of state-of-the-art UDN discoveries.

The proposed approaches in this area and their potential to improve several network properties could be studied in detail through extensive simulations by solving the CA optimization problem using sub-optimal methods, such as metaheuristic algorithms. Although, the application of metaheuristic algorithms for the CA is common, the primary focus of these works is to efficiently exploit the temporary conditions of the propagation environment, such as multi-user gain.

Nonetheless, these previously proposed strategies do not devise to deal with network densification. These fail to deal with the UDN's high interference and computational burden. Furthermore, recent approaches to solve CA in UDN present at least one of the following issues:

- Unfair resource distributions and user data rate starvation because they do not consider users' QoS.
- Consider that the bandwidth is distributed in a low number of channels, leading to inefficient use of the resources by allocating excessive bandwidth to users and increasing interference.
- Focus on exploiting the temporary conditions of the propagation environment instead of avoiding network interference.
- Consider a fixed number of users associated with each cell, exploring the network and user densification increase together.
- Considers that each deployed cell has a fixed number of users associated to it, considering only the case on which the network densification and user densification increase together.

In this thesis, we proposed a mathematical model on Interference Management for UDN scenarios to study the CA optimization problem in UDN. This model circumvent the above issues' adverse effects. The chosen model is characterized using stochastic search algorithms, such as random search and a sample of five well-known and some novel metaheuristics. The objective of the CA's process is to find a binary allocation matrix that maximizes network capacity while meeting the QoS demands of users. Therefore, a vector representation of the problem-solution should be designed to generate and manipulate these solutions.

As investigated in Chapter 3, the selection of the solution vector representation has an essential function in the performance of the metaheuristic techniques. The naive implementation and the traditional ICA were studied using random allocation solutions. When comparing the naive matrix representation with the traditional ICA approach, a positive bias can be observed that produces fair assignments between users. These results show a clear advantage in designing low-dimensional vector solutions. Also, consider required aspects of the network layout, which in the case of ICA includes the users' QoS information to pre-calculate their required channels. However, when required to solve the Individually Channel Allocation problem (ICA) applying metaheuristic algorithms, the solution vector's dimensionality was above 1000 for any network configuration. This high-dimensionality solution vector represents an issue, because metaheuristic algorithms are rarely designed or studied over such large optimization problems. Covering these immense search space would require an unpractical number of call functions (FFC). The current formulation of ICA does not allow metaheuristic algorithms to solve the CA problem in UDN efficiently. That is, find solutions close to the optimum in acceptable time responses through intensive simulations.

As the network grows, the channel allocation process becomes complicated due to the high number of cells and users deployed, fewer channels available and ever-increasing interference. A Channel-block-based allocation to reduce network interference and complexity of solving the channel allocation problem for high densification levels is introduced. In this proposal, the channel block size depends on the users' data rate requirements and the interference threshold allowed. This proposal offers a more efficient way of representing the CA's solutions. At the same time, filtering the search space from infeasible solutions was designed. This subset of the search space represents a solution-vector whose dimensionality is much smaller than conventional approaches offer. An overall reduction of complexity is achieved by reducing the search space into subsets of continuous channel blocks.

In this thesis, two different solution-vector representations were proposed. One of them, called User Block Allocation (UBA), assigns the block of channels to users concerning their data rate requirements. On the other hand, the Cell Block Allocation (CBA) uses a solution-vector representation in which the channels are distributed to cells. Both strategies minimize the overall UDN interference, reduce the optimization problem's search-space, and serve the users' QoS requirements.

The proposed UBA and CBA were evaluated using the selected metaheuristic algorithms compared with the traditional ICA and the coalition algorithm (reported in the literature). The results show, that both proposals improve the optimizers' performance in highly dense scenarios to find CA solutions. It is also shown that the CBA outperforms the other approaches. Although our proposal is a restricted sub-problem of channel allocation, it presents a better trade-off between performance and computational complexity than conventional channel allocation schemes.

One benefit of solving the CA problem with the proposed UBA and CBA schemes is analyzing several technical features of the UDN through extensive simulations. A UDN with user QoS data rate demands, a path-loss only channel model, and a fixed interference threshold are studied in this work. The ICA scheme performance results reveal a constant increase in capacity with the network's densification. However, the UBA and CBA schemes reach a more significant network capacity than ICA. The results also confirm a precise capacity limit that indicates that further densification is no longer advisable. It demonstrates the importance of exploring new and efficient allocation techniques to reach conclusive findings in the study of UDN. Our results also imply that the densification limit depends on the network traffic-load demand. However, it is also affected by the fixed calculation of required channels per user.

It is worth mentioning that with the UBA and CBA schemes, the Genetic Algorithm (GA) was the one that found the best solutions on the efficient use of communication channels. The GA reached the highest network capacity values (by controlling the interference) and the lowest computational resources. The preceding establishes that by improving GA performance or other metaheuristic-algorithms, it is possible to study different deployment characteristics related to a UDN, such as spectral efficiency, Sub-Channel Allocation Rate or densification limits, or some other QoS metrics.

From the reported results, reducing the CA problem's complexity helps metaheuristic algorithms find better ways to assign channels to users. The GA-CBA combination obtained the best results compared to the other optimization strategies, such as GA-UBA, PSO-CBA, or SMS-ICA, including the Coalition Algorithm reported in the literature. Despite reducing UDN interference, unattended users percentage (UPP), or increasing spectral efficiency (SE), the assigned channel rate (SAR) is not necessarily correlated with the network or users' capacity. This result proves that higher bandwidth utilization is not always beneficial for the users' QoS achievement or the network throughput. However, although the results obtained on the channel allocation in UDN are encouraging, applying metaheuristic-algorithms in a real mobile cellular system would be infeasible. The time it would take for a mobile-entity (BS or spectrum broker) to solve the channel assignment problem using evolutionary algorithms could take several minutes or even hours.

7.2 Findings

The experimentation on this work settles the following findings. The limits on the network densification's benefits depend on the network load of active users and their data rate demands. When the network requires attending a high number of users, high network densification is beneficial, as the traffic-load is distributed among the cells. However, the RA process complexity's increase rate is proportional to the users and cell densification. On the other hand, when only the users' bandwidth demands increase, increasing cell densification offers fewer benefits. For the traditional ICA approach, the CA problem complexity increases linearly, due to more channels are allocated to each user.

In contrast, the computational complexity of the CA's problem with proposed UBA and CBA remains fixed, as the whole required bandwidth of each user is allocated as a single resource block. Nonetheless, this thesis indicates that the efficiency of the network highly depends on the allocation of the resources for both cases of network load in active users and bandwidth demands. Reducing the size of the optimization problem enhances the probability of stochastic algorithms finding quality solutions. Interference control in UDN is an effect achieved from the proposed UBA and CBA schemes. Such reduction in interference increases the network capacity and enhances user satisfaction. These results highlights the importance of aiming for sub-optimal solutions rather than fast greedy heuristics tactics the UDN properties at high cell densities.

7.3 Future work

The proposed CA schemes aimed to improve the efficiency of sub-optimal stochastic algorithms, such as metaheuristics, to study several network parameters through extensive simulations. Although this investigation focus on satisfying users' QoS, several other aspects of the network remain of studying interest such as fairness, user allocation strategies for UDN, backhaul limited RA or handovers. Furthermore, in this work, a single-stage CA strategy is applied. Some post-allocation heuristics, such as the implement in the coalition algorithm, have proven to improve the solution by using local search over the neighborhood of the best solutions known.

The objective of this thesis is to propose novel methods that improve the channel allocation efficiency. Nonetheless, further optimization could be studied to improve the proposed block-base schemes. The proposed strategy to calculate the number of required channels makes use of a fixed interference threshold. However, the selection of this value is rather important, as it controls the overall aperture for sharing the spectrum between cells. The optimization of this parameter or the proposition of dynamic ranges of interference threshold could significantly improve network efficiency. Similarly, the selected metaheuristic algorithms, their operators' versions, and meta-parameters were selected following previous publications' recommendations. However, an in-deep comparison of metaheuristic algorithms is required to establish the set of techniques that offer high efficiency in this problem. Nonetheless, the proposed model and CA schemes can be used to make this comparison and study the scaling properties of metaheuristic algorithms over the problem's dimensionality.

Finally, this work focuses on the CA stage of the RA in UDN. Further research effort is required to compare the full RA strategy with other recent approaches such as multilayered RA with cell clustering.

7.4 List of Publication

1. Fernando Fausto, Adolfo Reyna, Erik Valdemar Cueva, Ángel G. Andrade, Marco Antonio Pérez-Cisneros. “From ants to whales: Metaheuristics for all tests”, Artificial Intelligence Review, Vol.53, no.1, pp.753-810, enero 2020, DOI:10.1007/s10462-018-09676-2

2. Adolfo Reyna, Ángel G. Andrade. “Dimensionality Reduction to solve Resource Allocation Problem in 5G UDN using Genetic Algorithm”. Soft Computing, 2021, DOI: 10.1007/s00500-020-05473-8.

3. Adolfo Reyna-Orta, Ángel G. Andrade, “Algoritmo Genético para la Asignación de Recursos con equidad en redes móviles celulares”, Revista Komputer Sapiens, Vol. 8, No.2, septiembre 2016, pp.28-32, ISSN:2007-0691.

4. Adolfo Reyna-Orta, Ángel G. Andrade, Guillermo Galaviz, “Application of Genetic Algorithm for Solving fairness-based resource allocation problem in mobile systems, Encuentro Nacional de Computación, (ENC 2016), pp. 59-63, 14-16 noviembre 2016, Chihuahua, Chihuahua, México.

5. Adolfo Reyna-Orta, Ángel G. Andrade. “Block-Channel based Resource Allocation for 5G Ultra-Dense Networks”. IEEE Transactions on Wireless Communications. Submitted October 2020.

Appendix A: Introduction to Metaheuristic Algorithms

Population based metaheuristics can be implemented using a general framework, independently of the natural phenomenon inspiration. This allows their study to be focus on the mechanisms that modify the solutions in the population through the iterative process.

Usually, the first step involves the definition of a set of M randomly initialized solutions as follows:

$$\vec{x}_i \in X \mid i = 1, 2, \dots, M \tag{1}$$

such that:

$$\vec{x}_i = [x_{i,1}, x_{i,2}, \dots, x_{i,D}] \mid x_{i,d} = R(B_d^{low}, B_d^{high}) \tag{2}$$

where D is the dimensionality of the problem (number of decision variables or parameters), $\vec{x}_i$ is the i solution of the population X and $x_{i,d} = R(B_d^{low}, B_d^{high})$ is the uniform random assignation bounded by the target solution space (search space) for the dimension d. The solutions $\vec{x}_i$ (or individuals) represent candidate solutions to be tested on the optimization problem. These are assigned to a corresponding quality value (or fitness) related to the objective function $f(\cdot)$ that describes the optimization problem, such that:

$$f_i \in F \mid f_i = f\left(\vec{x}_i\right),\ i = 1, 2, \dots, M \tag{3}$$

Using the population $X_{M \times D}$ and fitness $F_{M \times 1}$ information, new candidate solutions are generated by modifying currently available solutions. This is achieved by applying the algorithm operators (usually designed by drawing inspiration from an observed natural phenomenon). For most cases, this process may be illustrated by the following expression:

$$\vec{x}_i^t = \vec{x}_i + \ \vec{x}_i \tag{4}$$

where $\vec{x}_i^t$ denotes the trial solution generated by adding up a specified update vector $\ \vec{x}_i$ to $\vec{x}_i$. This update vector $\ \vec{x}_i$ depend on the specific algorithm operators, which are explained in the rest sections of this chapter. The generated solutions are evaluated in the fitness function to obtain their corresponding score, such as $F' = f(X')$.

Given the dimensional bounds on each dimension (B_d^{low}, B_d^{high}) depend on problem in question, metaheuristic algorithms can consider a unified domain range in their operators, as $x_{i,d} \in [0, 1] \forall d$. This leaves the scaling of the solution vectors into the hands of the fitness function $f(\vec{x}_i)$, separating the metaheuristic logic from the problem logic. This consideration allows to simplify the of the algorithms equations and the comparison of these between algorithms.

Finally, most nature-inspired algorithms include some kind of selection process, in which the newly generated solutions are compared against those in the current population X^k (with k denoting the current iteration) in terms of solution quality, typically with the purpose of choosing the best individual(s) among them. As a result of this process, a new set of solutions $X^{k+1} = \{\vec{x}_1^{k+1}, \vec{x}_2^{k+1}, \ldots, \vec{x}_M^{k+1}\}$, corresponding to the following iteration (or generation) '$k+1$', is generated.

This whole process is iteratively repeated until a particular stop criterion is met (i.e., a maximum number of iterations is reached). Once this happens, the best solution found by the algorithm $\vec{x}^*$ is reported as the best approximation for the global optimum. The best solution is the one encounter in whole optimization process with the maximum fitness value, can be expressed as follows:

$$f(\vec{x}^*) \geq f(\vec{x}_i^k)\ \forall i, k \tag{5}$$

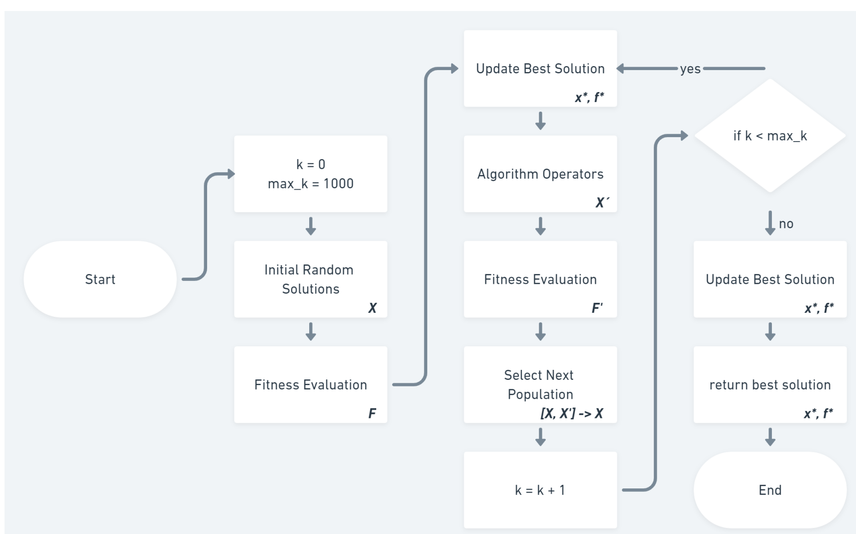


Figure 1: General framework for population-based metaheuristics.

The fig. 1 shows visually the whole general framework for the population-based metaheuristics, proposed in [77] and its open source implementation [81].

A.1 Genetic Algorithm

The Genetic Algorithm (GA) was initially developed by John Henry Holland in 1960 (and further extended in 1975) with the goal to understand the phenomenon of natural adaptation, and how this mechanism could be implemented into computers systems to solve complex problems. GA is one of the earliest metaheuristics inspired in the concepts of natural selection and evolution and is among the most successful Evolutionary Algorithms (EA) due to its conceptual simplicity and easy implementation [95]. This algorithm emulates the evolutionary process of a group of individuals in a restricted environment, in which the fittest individuals have higher opportunities for survival and through generations procreate even fitter offspring.

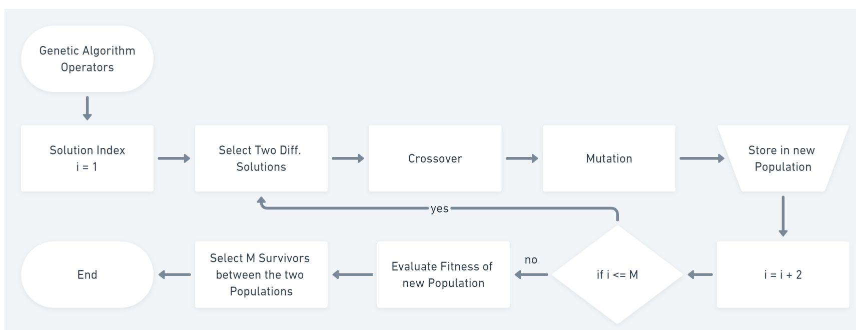


Figure 2: Genetic Algorithm Flowchart.

In GA, the solutions $\vec{x}_i$ of the population are considered individuals in the environment. These solutions are usually converted to a binary string form (this is, $x_{i,n} \in \{0, 1\}$) which represents the genetic code (chromosomes) of each individual. At each iteration, called generation, the chromosome population is modified by applying a set of evolutionary operators, namely: parents selection, crossover, mutation and survival selection. Fig. 2 shows the GA operators procedure to generate M new solution at each iteration of the evolutionary process. Two different solutions are randomly chosen (parents), usually biased to choose

Algorithm 1: Genetic Algorithm Operators (GA)

Input : D dimensions, N range of all dimensions, M solutions in population, $X_{M \times D}$ population, F_M fitness

Output: $X_{M \times D}$ new population

```
/* Tournament Selection */
for i from 1 to M do
    let r1, r2 be two different random integers from [1, M]
    let r_best be argmax(F_r1, F_r2)
    let Parent_i be X_(r_best)
end
/* Dimensional Crossover */
for i from 1 to M/2 do
    let trial1 be Parent_((i-1)*2)
    let trial2 be Parent_((i-1)*2+1)
    if rand < crossover_rate then
        for j from 1 to D do
            if rand > 0.5 then
                swap trial1_j and trial2_j
            end
        end
    end
    append [trial1, trial2] to trialPopulation
end
/* Mutation */
for i from 1 to M do
    for j from 1 to D do
        if rand < mutation_rate then
            set trialPopulation_(i,j) to a random integer from [1, N]
        end
    end
end
/* Evaluate trial population and keep the best */
let trialF to fitnessFunction(trialPopulation)
set X to selectBestM([X, trialPopulation], [F, trialF])
```

fitter solutions. The pair of solutions interchange their genetic information by the crossover operator, producing two new (offspring) solutions. New solutions have a small chance of mutating, by randomly shifting some value of the binary string. When the new population of M solutions is generated, the fitness function is applied to these to calculate their fitness value. Finally, the survival selection step chooses between the old and new generations to maintain a fixed number of solutions in the population, based on the analogy of limited resources. Several strategies have been proposed for each of the GA operators, enhancing the capabilities of this algorithm to provide a more balanced search. Form this point the chosen operators are explained with their respective math equations.

For the selection operation, the selection of M parents solutions ($\vec{P}_i \mid i = 1, 2, \ldots, M$) is performed by a tournament of two randomly selected solutions $\vec{x}_{s_1}$ and $\vec{x}_{s_2}$.

$$\vec{P}_i = \begin{cases} \vec{x}_{s_1}, \text{ if } f(\vec{x}_{s_1}) > f(\vec{x}_{s_2}) \\ \vec{x}_{s_2}, \text{ otherwise} \end{cases} \mid s_1, s_2 = \text{rand}_{int}(1, M),\ s_1 \neq s_2 \tag{6}$$

The crossover operation recombines the information of each pair of selected solutions using a binary vector $\vec{r}$ that selects the recombination pattern, such as:

$$P^*_{2i+1,d} = \begin{cases} P_{2i+1,d} & \text{if } r_n = 1 \\ P_{2i+2,d} & \text{otherwise} \end{cases}, P^*_{2i+2,d} = \begin{cases} P_{2i+1,d} & \text{if } r_d = 0 \\ P_{2i+2,d} & \text{otherwise} \end{cases}, \forall i \in [0, 1, \ldots, \frac{M}{2}-1],\ \forall d \tag{7}$$

where $\vec{r}$ is usually form by selecting a random pivot point l as:

$$r_d = \begin{cases} 1 & \text{if } d < l \\ 0 & \text{otherwise} \end{cases} \quad l = rand_{int}(1, D) \tag{8}$$

The vector $\vec{r}$ can be form by using multiple random pivot points ($l = [l_1, l_2, ...] \mid l_1 < l_2 < ...$) or by a random binary choice, as:

$$r_d = rand_{int}(0, 1) \tag{9}$$

As the crossover operator is controlled by a meta-parameter $r_{Crossover}$, the trial solution $\vec{x}_i^t$ is chosen between $\vec{P}_i^*$ and $\vec{P}_i$ with a probability of $r_{Crossover}$, as:

$$\vec{x}_i^t = \begin{cases} \vec{P}_i^* & \text{if } rand < r_{Crossover} \\ \vec{P}_i & \text{otherwise} \end{cases}$$

The mutation operation introduce a random change in the generated $\vec{x}_i^t$ solutions, usually changing a single bit from 1 to 0 or vice-versa. Mutation can occur over each dimensionality of the solutions with a probability $r_{Mutation}$ (typically as low as 0.001), as given as follows:

$$x_{i,d}^t = \begin{cases} \overline{x_{i,d}^t} & \text{if } rand(0,1) < r_{Mutation} \\ x_{i,d}^t & \text{otherwise} \end{cases} , \forall d \tag{10}$$

where $\overline{x_{i,d}^t}$ represent the bit flip for binary strings, or a random assignation between the domain of such dimension n, when working with discrete values.

As shown in Fig. 2 these three operators takes place until a population of M new solutions has been produced, and then, the M best solutions among the original and new populations are taken for the next generation, while the rest are discarded (Eq. 12). This is known as greedy selection, which requires to join the two populations X^k and X^t and sort the solution in terms of their fitness value (Eq. 11), descending for maximization or ascending for a minimization problem.

$$X' = X^k \cup X^t \mid \forall i, f(x'_i) < f(x'_{i-1}), i > 1 \tag{11}$$

$$X^{k+1} = [x'_1, x'_2, ..., x'_M]; \tag{12}$$

In practice, both selection operators seek to maintain elitism in the population; therefore, they enforce exploitation. These operators can be purely greedy or modified to admit some underfit solutions to avoid losing diversity. On the other hand, Mutation is meant to be a strictly explorative operator, which seek to prevent premature convergence to local optima. Although Crossover is usually considered as an exploitation operator, recent works have shown that it conveys a balance between exploration and exploitation when the population is dispersed, gradually shifting to exploitation as the population converges [78].

A.2 Differential Evolution (DE)

Differential Evolution is another metaheuristic inspired by the evolutionary process observed in nature, introduced by Rainer Storn & Kenneth Price in 1996 [96]. Unlike GA, DE is easier to implement, works with real value solutions, and combines more than two solutions in the generation of new ones [97]. At each generation 'k', DE applies a series of mutation, crossover and selection operators in order to evolve the solutions population, as shown in Fig. 3. In simple terms, and in the simplest form of DE, for each solution in the population $\vec{x}_i$, three other solutions are uniformly chosen to produce a mutation vector. This Mutation is created by adding to the first solution the weighted difference of the other two, as $\vec{m}_i = \vec{x}_{r_1} + F(\vec{x}_{r_2} - \vec{x}_{r_3})$. Then, the new solution is simply a random crossover of $\vec{x}_i$ and $\vec{m}_i$ over each dimension using the Crossover Rate (r_{Cross}) meta-parameter, and ensuring that at least one dimension is included. Finally, the new solution is directly compared with $\vec{x}_i$ in terms of fitness value, replacing it in the current population if an improvement is found.

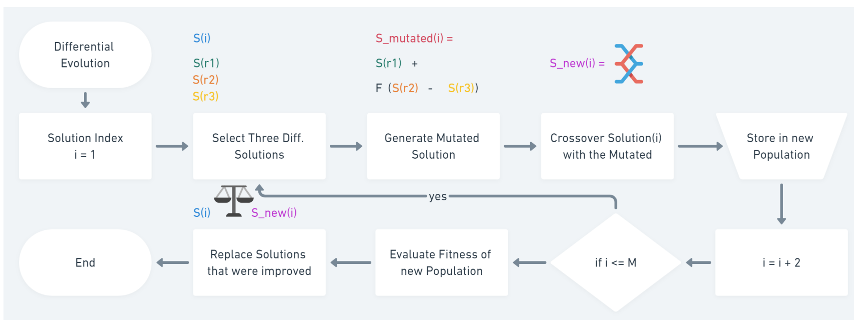


Figure 3: Differential Evolution Flowchart.

The DE mutation operation generates a new mutant solution for each solution, i.e., $\vec{m}_i \forall i \in [1, ..., M]$, as follows:

$$\vec{m}_i = \vec{x}_{r_3} + F\left(\vec{x}_{r_1} - \vec{x}_{r_2}\right) \tag{13}$$

where $r_1, r_2,\ r_3 \in \{1, 2, \ldots, M\}$ (and with $r_1 \neq r_2 \neq r_3$) each denote a randomly chosen solution index, while the meta-parameter $F \in [0, 2]$ is called differential weight, and is used to control the magnitude of the differential variation at the right side of Eq. 13.

The DE crossover operation generates a trial solution vector $\vec{x}_i^t$ composed by a

Algorithm 2: Differential Evolution (DE)

Input : D dimensions, M solutions in population, $X_{M\times D}$ population, F_M fitness,
Output: $X_{M\times D}$ new population

for i *from* 1 *to* M **do**
  let $r1$, $r2$, $r3$ be three different random integers from $[1,\ M]$
  let $base$ be $X_{r1} + F \cdot (X_{r2} - X_{r3})$
  let $trial$ be X_i
  for j *from* 1 *to* D **do**
    if $rand < crossover_{rate}$ **then**
      set $trial_j$ to $base_j$
    end
  end
  set $trial$ to $toroidalCorrection(trial)$
  let F_{trial} be $fitnessFunction(trial)$
  if $F_{trial} > F_i$ **then**
    set F_i to F_{trial}
    set X_i to $trial$
  end
end

random combination of both the candidate solution $\vec{x}_i$ and its respective mutant solution $\vec{m}_i$ as follows:

$$x_{i,d}^{t} = \begin{cases} m_{i,d} & \text{if } (\text{rand} \leq r_{Cross}) \text{ or } d = n^{*} \\ x_{i,d} & \text{otherwise} \end{cases} , \forall d \tag{14}$$

where $n^{*} \in \{1, 2, \dots, D\}$ denotes a randomly chosen dimension index. This n^{*} ensures that at least one dimension of the mutated solution is used in the trial solution.

Finally, for DE's selection process, each trail solution $\vec{x}_i^{t}$ is compared against its respective candidate solution $\vec{x}_i$ in terms of solution quality (fitness) by applying a greedy criterion. This is:

$$\vec{x}_i^{k+1} = \begin{cases} \vec{x}_i^t & \text{if } \left(f\left(\vec{x}_i^t\right) > \ f\left(\vec{x}_i^k\right)\right) \\ \vec{x}_i^k & \text{otherwise} \end{cases} \tag{15}$$

This type of selection operator, known as individual elitism, differs from the greedy approach in that it maintains the initial diversity and ensures that each solution is improved locally. This property makes DE and other algorithms using this selection prone to be used in multimodal optimization problems, where several local optima are search simultaneously. Due to the Crossover choosing between the known solution and the mutated one over the dimensionalities, this Algorithm excels on problems in which the different dimensionalities are geometrically related [77], as most of the benchmark functions used to test these algorithms are.

A.3 Particle Swarm Optimization (PSO)

Devised by James Kennedy & Russell C. Eberhart in 1995, the Particle Swarm Optimization (PSO) method draws inspiration in the behavior of flocking birds, collectively foraging for adequate food sources [98]. PSO is currently the second most popular population-based metaheuristic [77], which has some merits to its simplicity, as shown in Fig. 4. In PSO, solutions (also referred as *particles*) are composed by a set of three D-dimensional vectors: the particle's current position ($\vec{x}_i$), its historical best position ($\vec{x}_i^*$) and its velocity ($\vec{v}_i$). The exploitation is performed by attracting the solutions in the population to the global best solution $\vec{x}^*$ (note the lack of index), and the best personal $\vec{x}_i^*$, by modifying their velocity considering the distance to these two "good" known points. Therefore, for each iteration of the optimization process, the velocity vectors are re-calculated (Eq. 17), the solutions positions are updated by applying the corresponding velocity (Eq. 16), and the memory of best personal positions is updated. Unlike other approaches, PSO does not apply a Selection operator, as each newly trial solution is maintained in the population.

Figure 4: Particle Swarm Optimization Flowchart.

Algorithm 3: Particle swarm optimization (PSO)

Input : D dimensions, M solutions in population, $X_{M \times D}$ population, F_M fitness, X^* current best solution, F^* current best fitness value

Output: $X_{M \times D}$ new population

`/* Persistent Variables */`
let $X^{best}_{M \times D}$ be the best personal solution
let F^{best} be the best personal fitness value
let $V_{M \times D}$ be the velocity of each solution
`/* Update Velocity and Position */`
for i *from* 1 *to* M **do**
  set V_i to $velocity_w * V_i + personal_w(X_i^{best} - X_i) + best_w(X^* - X_i)$
  set X_i to $X_i + V_i$
  set X_i to $toroidalCorrection(X_i)$
  `/* Evaluate new solution */`
  set F_i to $fitnessFunction(X_i)$
  `/* Update Personal Best */`
  if $F_i > F_i^{best}$ **then**
    set F_i^{best} to F_i
    set X_i^{best} to X_i
  end
end
`/* Update Best */`
if $max(F) > F^*$ **then**
  set F^* to $max(F)$
  set X^* to $X_{argmax(F)}$
end

In PSO, for each iteration $(k + 1)$ a new set of M trial solutions is generated by updating the current population solutions positions as follows:

$$\vec{x}_i^{k+1} = \vec{x}_i^k + \vec{v}_i \tag{16}$$

where v_i stand for the updated velocity of particle 'i', given as follows:

$$\vec{v}_i = \vec{v}_i + c_1 \cdot r_1 \cdot \left(\vec{x}_i^* - \vec{x}_i^k\right) \ + c_2 \cdot r_2 \cdot \left(\vec{x}^* - \vec{x}_i^k\right) \tag{17}$$

where r_1 and r_2 each denote an uniform random D-dimensional vector with domain [0, 1], while the values c_1 and c_2, being the meta-parameters of PSO, are known as cognitive and social parameters, respectively. Finally, the set of personal best positions, of each solution is updated as:

$$\vec{x}_i^* = \begin{cases} \vec{x}_i^{k+1} & \text{if } f(\vec{x}_i^{k+1}) > f(\vec{x}_i^*) \\ \vec{x}_i^* & \text{otherwise} \end{cases} \quad \forall\ \vec{x}_i^{k+1} \in X^{k+1} \tag{18}$$

Because PSO does not include a Selection step, exploration is improved. However, the balance between exploration and exploitation depends mostly on the meta-parameters of the Algorithm, which dictate the amount of attraction between the global and personal best-known solutions, and the velocity inertia. Therefore the performance of PSO is heavily affected by the selection of these parameters. PSO is also known to converge quickly on local optima, for which several modification approaches have been proposed [99]. Given the vector nature of (Eq. 16) and (Eq. 17), the PSO computational implementation is efficient, and straight forward, simplifying its use for GPU parallel implementations. However, it requires 3x more memory than other algorithms, as it requires to store the values of sets X, X^* and V, which reduces the maximum amount of solutions in the population when using GPU computation or embedded devices.

A.4 States of Matter Search (SMS)

States of Matter Search (SMS) is a recently proposed algorithm, which uses an increased computational complexity operators in comparison to traditional approaches [100]. This Algorithm is inspired by the physical principles of thermal-energy motion for the three stages of matter (gas, liquid and solid), such as directional movements, collisions, and apparent random behavior. Under the SMS analogy, each solution represents a particle position whose movement is dictated by a velocity vector, similar to PSO. However, the SMS main idea is to separate the iterative search into three well-defined phases, each one with a target balance between exploration and exploitation.

Table 1: Internal parameters of SMS.

Parameter\Phase	1.Gas	2.Liquid	3.Solid
β	0.9	0.5	0.1
α	0.3	0.05	0
H	0.9	0.35	0
Iterations	50%	40%	10%

Independently of the current search stage of the SMS, each iteration three sequential operators modify the solutions in the population:

1. The velocity $\vec{v}_i$ is calculated considering the distance between the solutions and the best-known solution so far;
2. Possible collisions are identified, and velocities between particles involved are swapped;
3. Positions are updated using the velocity vector, and randomness is applied with a given probability over each dimensionality of the solutions.

The three phases in SMS are controlled by three internal parameters (differently from meta-parameters which are tuned within the specific problem), these control the participation of the previous velocity vector (β), the minimum distance on which particle collisions occurs (α) and the probability of random shifting in between dimensions of the current solutions (H). These internal parameters are listed for each search phase in the Tbl. 1. Then, SMS algorithm uses the firsts 50% of the iterations to focus on exploration (gas), with the objective of identifying the neighborhood of the global optima. Then, the next 40% of the iterations are use to converge the SMS population into the best known solution so far ($\vec{x}^*$), with a more balanced exploration and exploitation behavior (liquid). While in the rest 10% of the iterations, SMS focuses on finely exploiting the neighborhood around $\vec{x}^*$ (solid). In the last phase, attraction the best solution is the only operator active, therefore the solutions in the population converge directly to it.

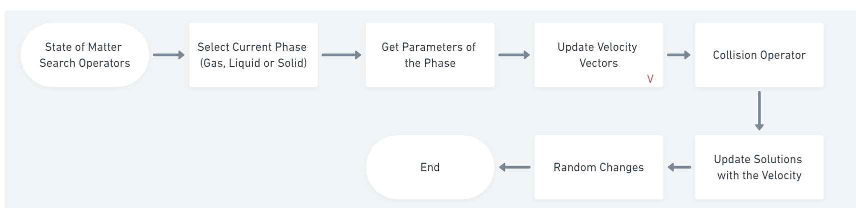


Figure 5: States of Matter Search Flowchart.

In general, at each iteration 'k', each molecule updates its position by applying the following movement rule:

$$\vec{x}_i^{k+1} = \vec{x}_i^k + \vec{v}_i \cdot \rho \cdot rand(0,1) \tag{19}$$

where $\rho \in [0,1]$ denotes a meta-parameter factor which value depend on the current SMS stage [89] and $\vec{v}_i$ denotes the velocity of the i-th solution. This

Algorithm 4: States of Matter Search (SMS)

Input : D dimensions, M solutions in population, $X_{M\times D}$ population, F_M fitness, X^* current best solution, F^* current best fitness value, Ite current iteration, $maxIte$ maximun number of iterations

Output: $X_{M\times D}$ new population

`/* Persistent Variables */`
let $V_{M\times D}$ be the velocity of each solution
for i *from* 1 *to* M **do**
    `/* Update Velocity */`
    set V_i to $0.5 \cdot V_i \cdot Ite/maxIte + (X^* - X_i)/|X^* - X_i|$
    `/* Collisions */`
    for k *from* $i+1$ *to* M **do**
        if $|X_i - X_k| < \alpha[Phase]$ **then**
            swap V_i and V_k
        end
    end
    `/* Update Positions */`
    set X_i to $X_i + rand \cdot \beta[Phase] \cdot step[Phase] \cdot V_i$
    `/* Random Behaviour */`
    if $rand < H[Phase]$ **then**
        let $r1$ be a random dimension between $[1, D]$
        set $X_{i,r1}$ to $rand$
    end
    set X_i to $toroidalCorrection(X_i)$
    `/* Evaluate new solution */`
    set F_i to $fitnessFunction(X_i)$
end
`/* Update Best */`
if $max(F) > F^*$ **then**
    set F^* to $max(F)$
    set X^* to $X_{argmax(F)}$
end
`/* Update Phase */`
if $1 - (Ite/maxIte) < Phases[Phase]$ **then**
    add 1 to $phase$
end

velocity is updated each iteration using the following expressions:

$$\vec{v}_i = \vec{v}_i \cdot \beta \cdot 0.5 \left(1 - \frac{k}{K}\right) + \frac{(\vec{x}^* - \vec{x}_i)}{\|\vec{x}^* - \vec{x}_i\|} \tag{20}$$

where the left side of the sum controls the participation of the previous velocity and the right side adds the unit attraction vector of each solution $\vec{x}_i$ to the current best solution $\vec{x}^*$. The parameter β is a scalar factor that depends on the current *phase* of the SMS ($\beta[phase]$ see Tbl. 1), while the fraction k/K, being K the maximum number of iterations, reduces the participation of the previous velocity linearly with the iterative process.

The SMS's collision mechanism causes that any two solutions, i and j (with $i \neq j$), with a distance lower than the current phase parameter $\alpha[phase]$ to exchange their velocity vectors, such that:

$$\text{if } (\|\vec{x}_i^k - \vec{x}_i^k\| < \alpha) \rightarrow \begin{array}{c} \vec{v}_i = \vec{v}_j \\ \vec{v}_j = \vec{v}_j \end{array} \ \forall (i, j) \mid i \neq j \tag{21}$$

This collision operator aims to maintain diversity in the population, by pushing solutions that got closer to opposite directions, similarly to other algorithms that considers gravity or electromagnetic analogies in their operators. However, this requires the calculation of the distance between all the solutions, which has a computational complexity of $\mathcal{O}(M^2)$, braking the computational linear complexity of these algorithms.

Finally random behavior is applied by a simple operator that assings a random position over the dimension n of each solution given the random rate $H[phase]$, as:

$$x_{i,d}^{k+1} = \begin{cases} rand(0,1) & \text{if } rand(0,1) < H[phase] \\ x_{i,d}^{k+1} & \text{otherwise} \end{cases}, \forall d \qquad (22)$$

As the SMS has a similar movement operator to PSO, with the difference that SMS only considers attraction of solutions to global best known solution $\vec{x}^*$, it could potentially be affected by the same premature convergence issues of it. However, this is addressed by using the first half of the iterations purely on exploration, aiming for the global optima neighborhood. In addition, the collision operator helps to maintain diversity in the population even during the liquid stage. The strategy of diving the optimization into stages also make the Algorithm less sensitive with regards to choosing the right meta-parameter. Given the high diversity of the first phase of the algorithm, a multimodal implementation was proposed by [89] using an additional memory that aims to maintain several "good" neighborhoods.

A.5 Whale Optimization Algorithm (WOA)

In [90], Seyedali Mirjalili & Andrew Lewis proposed a novel bio-inspired metaheuristic called Whale Optimization Algorithm (WOA). This algorithm is inspired by the predatory behavior observed in humpback whales, which employ a unique cooperative hunting maneuver known as bubble-net feeding. In this maneuver, a group of whales coordinate their efforts to trap a group of small prey by swimming in a spiral pattern around them and exhaling a burst of air to form an encircling bubble barrier, which prevents prey from escaping. Prey gets corralled into a tighter circle, until a point on which all individuals simultaneously swim to the surface with their mouths open to feed on the trapped prey.

Algorithm 5: Whale Optimization Algorithm (WOA)

Input : D dimensions, M solutions in population, $X_{M \times D}$ population, F_M fitness, X^* current best solution, F^* current best fitness value, Ite current iteration, $maxIte$ maximun number of iterations

Output: $X_{M \times D}$ new population

for i *from* 1 *to* M **do**
  if $rand < 0.5$ **then**
    let A be $2 \cdot (1 - Ite/maxIte) \cdot rand$
    if $A < 1$ **then**
      let $trial$ be $X^* - A \cdot |2 \cdot rand \cdot X^* - X_i|$
    else
      let $randMix$ be an empty vector
      for j *from* 1 *to* D **do**
        let $r1$ be a random integer between $[1, M]$
        append $X_{r1,j}$ to $randMix$
      end
      let $trial$ be $randMix - A \cdot |2 \cdot rand \cdot randMix - X_i|$
    end
  else
    let L be -(2 + Ite/maxIte) * rand + 1
    set L to $e^L \cdot cos(2\pi L)$
    let $trial$ be $|X^* - X_i| \cdot L + X^*$
  end
  set X_i to $toroidalCorrection(trial)$
  `/* Evaluate new solution */`
  set F_i to $fitnessFunction(X_i)$
end
`/* Update Best */`
if $max(F) > F^*$ **then**
  set F^* to $max(F)$
  set X^* to $X_{argmax(F)}$
end

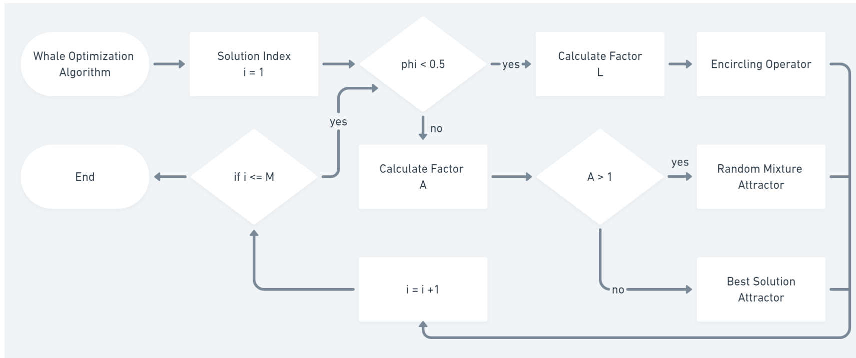


Figure 6: Whale Optimization Algorithm Flowchart.

In WOA, the whales are represented by the solutions in the population ($\vec{x}_i \in X$), while the best solution found so far ($\vec{x}^*$) represents the prey. With every iteration, each solution updates its position using one of the following three operators: encircling the prey, attacking the prey, or searching for prey. These represents three different forms of attractions between solutions, being a encircling attraction to the best solution $\vec{x}^*$, and two linear attraction to the best solution $\vec{x}^*$ and to a random solution in the population $\vec{x}_{r_1}$, this can be observed in Fig. 6. The algorithm is designed to provide the same probability between the encircling operator and the two linear ones, using an uniform random variable of φ. However, the selection between linear operators is perform by the random variable A, whose domain linearly decreases from $[0, 2]$ to $[0, 0]$ over the course of iterations as shown in Eq. 23. This favors the attraction to random solutions at the beginning of the iterations ($A > 1$), and gradually shifting to advocate for the current best solution attractor ($A \leq 1$).

$$A = 2\left(1 - \frac{k}{K}\right) rand(0, 1) \tag{23}$$

In the linear attraction operators, the solutions in the population are updated by applying the following equation:

$$\vec{x}_i^{k+1} = \begin{cases} \vec{x}^{r_1} - A \cdot \left| rand(0,1) \cdot \vec{x}^{r_1} - \vec{x}_i^k \right| & \text{if } A > 1 \\ \vec{x}^* - A \cdot \left| rand(0,1) \cdot \vec{x}^* - \vec{x}_i^k \right| & \text{otherwise} \end{cases} \tag{24}$$

where $\vec{x}^{r_1}$ denotes the mixture of a randomly chosen solution for each dimension, such as $x_d^{r_1} = x_{r,d} \mid r = rand(1, M)$, $\forall d$.

On the other hand, for the encircling operator, each solution moves in a spiraling fashion around the current best solution, updating their positions as follows:

$$\vec{x}_i^{k+1} = \left| \vec{x}^* - \vec{x}_i^k \right| \cdot \left(e^L \cdot \cos(2\pi L) \right) + \vec{x}^* \tag{25}$$

where $e^L \cdot \cos(2\pi L)$ represented the logarithmic spiral model whose shape is controlled by the random variable $l \in [-2, 1]$.

$$L = -\left(2 + \frac{k}{K} \right) \cdot rand(0,1) + 1 \tag{26}$$

In contrast to other algorithms, WOA maintains a balanced strategy between exploration and exploitation throughout the evolutionary process. An advantage of the utilization of WOA is that no meta-parameter needs to be selected. There is two interesting characteristics of WOA linear attraction operator: the mixture of solutions through dimensionalities $\vec{x}^{r_1}$ and that the random factor do not scale de distance between solutions, but the distance of the selected solution, a random one or the current best solution, in relation to the search space origin.

References

[1] GSMA-Intelligence, *Definitive data and analysis for the mobile industry.* 2020.

[2] A. Turner, *How many smartphones are in the world?* 2020.

[3] E. J. Oughton, Z. Frias, S. van der Gaast, and R. van der Berg, "Assessing the capacity, coverage and cost of 5G infrastructure strategies: Analysis of the netherlands," *Telematics and Informatics*, vol. 37, pp. 50–69, Apr. 2019.

[4] A. Imran and A. Zoha, "Challenges in 5G: How to empower SON with big data for enabling 5G," *IEEE Network*, vol. 28, no. 6, pp. 27–33, Nov. 2014.

[5] Y. Niu, Y. Li, D. Jin, L. Su, and A. V. Vasilakos, "A survey of millimeter wave communications (mmWave) for 5G: Opportunities and challenges," *Wireless Networks*, vol. 21, no. 8, pp. 2657–2676, Apr. 2015.

[6] S. A. Busari, K. M. S. Huq, S. Mumtaz, L. Dai, and J. Rodriguez, "Millimeter-wave massive MIMO communication for future wireless systems: A survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 2, pp. 836–869, 2018.

[7] D. Lopez-Perez, M. Ding, H. Claussen, and A. H. Jafari, "Towards 1 gbps/UE in cellular systems: Understanding ultra-dense small cell deployments," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2078–2101, 2015.

[8] E. Z. Tragos, S. Zeadally, A. G. Fragkiadakis, and V. A. Siris, "Spectrum assignment in cognitive radio networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 1108–1135, 2013.

[9] N. Muchandi and R. Khanai, “Cognitive radio spectrum sensing: A survey,” in *2016 international conference on electrical, electronics, and optimization techniques (ICEEOT)*, 2016.

[10] W. Z. Khan, E. Ahmed, S. Hakak, I. Yaqoob, and A. Ahmed, “Edge computing: A survey,” *Future Generation Computer Systems*, vol. 97, pp. 219–235, Aug. 2019.

[11] F. Boccardi, R. W. Heath, A. Lozano, T. L. Marzetta, and P. Popovski, “Five disruptive technology directions for 5G,” *IEEE Communications Magazine*, vol. 52, no. 2, pp. 74–80, Feb. 2014.

[12] M. Zhang, M. Polese, M. Mezzavilla, J. Zhu, S. Rangan, S. Panwar, and M. Zorzi, “Will TCP work in mmWave 5G cellular networks?” *IEEE Communications Magazine*, vol. 57, no. 1, pp. 65–71, Jan. 2019.

[13] E. G. Larsson, O. Edfors, F. Tufvesson, and T. L. Marzetta, “Massive MIMO for next generation wireless systems,” *IEEE Communications Magazine*, vol. 52, no. 2, pp. 186–195, Feb. 2014.

[14] T. L. Marzetta, “Noncooperative cellular wireless with unlimited numbers of base station antennas,” *IEEE Transactions on Wireless Communications*, vol. 9, no. 11, pp. 3590–3600, Nov. 2010.

[15] F. Rusek, D. Persson, B. K. Lau, E. G. Larsson, T. L. Marzetta, and F. Tufvesson, “Scaling up MIMO: Opportunities and challenges with very large arrays,” *IEEE Signal Processing Magazine*, vol. 30, no. 1, pp. 40–60, Jan. 2013.

[16] N. Bhushan, J. Li, D. Malladi, R. Gilmore, D. Brenner, A. Damnjanovic, R. Sukhavasi, C. Patel, and S. Geirhofer, “Network densification: The dominant theme for wireless evolution into 5G,” *IEEE Communications Magazine*, vol. 52, no. 2, pp. 82–89, Feb. 2014.

[17] M. Kamel, W. Hamouda, and A. Youssef, “Ultra-dense networks: A survey,” *IEEE Communications Surveys & Tutorials*, vol. 18, no. 4, pp. 2522–2545, 2016.

[18] L. Xu, Y. Mao, S. Leng, G. Qiao, and Q. Zhao, “Energy-efficient resource allocation strategy in ultra dense small-cell networks: A stackelberg game approach,” in *2017 IEEE international conference on communications (ICC)*, 2017.

[19] A. Gupta and R. K. Jha, “A survey of 5G network: Architecture and emerging technologies,” *IEEE Access*, vol. 3, pp. 1206–1232, 2015.

[20] M. Giordani, M. Polese, M. Mezzavilla, S. Rangan, and M. Zorzi, “Toward 6G networks: Use cases and technologies,” *IEEE Communications Magazine*, vol. 58, no. 3, pp. 55–61, Mar. 2020.

[21] N. Bhushan, J. Li, D. Malladi, R. Gilmore, D. Brenner, A. Damnjanovic, R. Sukhavasi, C. Patel, and S. Geirhofer, “Network densification: The dominant theme for wireless evolution into 5G,” *IEEE Communications Magazine*, vol. 52, no. 2, pp. 82–89, Feb. 2014.

[22] B. Romanous, N. Bitar, A. Imran, and H. Refai, “Network densification: Challenges and opportunities in enabling 5G,” in *2015 IEEE 20th international workshop on computer aided modelling and design of communication links and networks (CAMAD)*, 2015.

[23] H. S. Dhillon, R. K. Ganti, F. Baccelli, and J. G. Andrews, “Modeling and analysis of k-tier downlink heterogeneous cellular networks,” *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 3, pp. 550–560, Apr. 2012.

[24] J. Liu, M. Sheng, L. Liu, and J. Li, “Interference management in ultra-dense networks: Challenges and approaches,” *IEEE Network*, vol. 31, no. 6, pp. 70–77, Nov. 2017.

[25] J. G. Andrews, X. Zhang, G. D. Durgin, and A. K. Gupta, “Are we approaching the fundamental limits of wireless network densification?” *IEEE Communications Magazine*, vol. 54, no. 10, pp. 184–190, Oct. 2016.

[26] Z. Zhang, G. Yang, Z. Ma, M. Xiao, Z. Ding, and P. Fan, “Heterogeneous ultradense networks with NOMA: System architecture, coordination framework, and performance evaluation,” *IEEE Vehicular Technology Magazine*, vol. 13, no. 2, pp. 110–120, Jun. 2018.

[27] V. M. Nguyen and M. Kountouris, “Performance limits of network densification,” *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 6, pp. 1294–1308, Jun. 2017.

[28] B. Romanous, N. Bitar, A. Imran, and H. Refai, “Network densification: Challenges and opportunities in enabling 5G,” in *2015 IEEE 20th international workshop on computer aided modelling and design of communication links and networks (CAMAD)*, 2015.

[29] X. Zhu, Y. Xu, Y. Zhang, Y. Sun, and Z. Du, “Load-aware dynamic access for ultra-dense small cell networks: A hypergraph game theoretic solution,” in *Lecture notes in electrical engineering*, Springer Singapore, 2018, pp. 20–28.

[30] M.-C. Chuang, M. C. Chen, and Y. S., “Resource management issues in 5G ultra dense smallcell networks,” in *2015 international conference on information networking (ICOIN)*, 2015.

[31] C. Peng, H. Zheng, and B. Y. Zhao, “Utilization and fairness in spectrum assignment for opportunistic spectrum access,” *Mobile Networks and Applications*, vol. 11, no. 4, pp. 555–576, May 2006.

[32] Z. Zhao, Z. Peng, S. Zheng, and J. Shang, “Cognitive radio spectrum allocation using evolutionary algorithms,” *IEEE Transactions on Wireless Communications*, vol. 8, no. 9, pp. 4421–4425, Sep. 2009.

[33] A. Martı’nez-Vargas, J. Domı’nguez-Guerrero, Ángel G. Andrade, R. Sepúlveda, and O. Montiel-Ross, “Application of NSGA-II algorithm to the spectrum assignment problem in spectrum sharing networks,” *Applied Soft Computing*, vol. 39, pp. 188–198, Feb. 2016.

[34] R. AliHemmati, S. ShahbazPanahi, and M. Dong, “Joint spectrum sharing and power allocation for OFDM-based two-way relaying,” *IEEE Transactions on Wireless Communications*, vol. 14, no. 6, pp. 3294–3308, Jun. 2015.

[35] S. G. Colás, “Ultra dense networks deployment for beyond 2020 technologies,” PhD thesis, Universitat Politecnica de Valencia.

[36] Y. Chen, Z. Yang, and H. Zhang, “Opportunistic-based dynamic interference coordination in dense small cells deployment,” in *2017 IEEE 28th annual international symposium on personal, indoor, and mobile radio communications (PIMRC)*, 2017.

[37] H. B. Jung and D. K. Kim, “Power control of femtocells based on max-min fairness in heterogeneous networks,” *IEEE Communications Letters*, vol. 17, no. 7, pp. 1372–1375, Jul. 2013.

[38] E. Z. Tragos, S. Zeadally, A. G. Fragkiadakis, and V. A. Siris, “Spectrum assignment in cognitive radio networks: A comprehensive survey,” *IEEE Communications Surveys &*

Tutorials, vol. 15, no. 3, pp. 1108–1135, 2013.

[39] C. Y. Wong, R. Cheng, K. Lataief, and R. Murch, “Multiuser OFDM with adaptive subcarrier, bit, and power allocation,” *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 10, pp. 1747–1758, 1999.

[40] H. Zhang, C. Jiang, N. C. Beaulieu, X. Chu, X. Wen, and M. Tao, “Resource allocation in spectrum-sharing OFDMA femtocells with heterogeneous services,” *IEEE Transactions on Communications*, vol. 62, no. 7, pp. 2366–2377, Jul. 2014.

[41] Y. Teng, M. Liu, F. R. Yu, V. C. M. Leung, M. Song, and Y. Zhang, “Resource allocation for ultra-dense networks: A survey, some research issues and challenges,” *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2134–2168, 2019.

[42] W. Li and J. Zhang, “Cluster-based resource allocation scheme with QoS guarantee in ultra-dense networks,” *IET Communications*, vol. 12, no. 7, pp. 861–867, Apr. 2018.

[43] X. Tang, P. Ren, F. Gao, and Q. Du, “Interference-aware resource competition toward power-efficient ultra-dense networks,” *IEEE Transactions on Communications*, vol. 65, no. 12, pp. 5415–5428, Dec. 2017.

[44] W. Rhee and J. Cioffi, “Increase in capacity of multiuser OFDM system using dynamic subchannel allocation,” in *VTC2000-spring. 2000 IEEE 51st vehicular technology conference proceedings (cat. No.00CH37026)*.

[45] C. Jiang, Y. Chen, K. J. R. Liu, and Y. Ren, “Optimal pricing strategy for operators in cognitive femtocell networks,” *IEEE Transactions on Wireless Communications*, vol. 13, no. 9, pp. 5288–5301, Sep. 2014.

[46] Q. He and P. Zhang, “Dynamic channel assignment using ant colony optimization for cognitive radio networks,” in *2012 IEEE vehicular technology conference (VTC fall)*, 2012.

[47] P. Bhardwaj, A. Panwar, O. Ozdemir, E. Masazade, I. Kasperovich, A. L. Drozd, C. K. Mohan, and P. K. Varshney, “Enhanced dynamic spectrum access in multiband cognitive radio networks via optimized resource allocation,” *IEEE Transactions on Wireless Communications*, vol. 15, no. 12, pp. 8093–8106, Dec. 2016.

[48] P. Bhardwaj, A. Panwar, O. Ozdemir, E. Masazade, I. Kasperovich, A. L. Drozd, C. K. Mohan, and P. K. Varshney, "Enhanced dynamic spectrum access in multiband cognitive radio networks via optimized resource allocation," *IEEE Transactions on Wireless Communications*, vol. 15, no. 12, pp. 8093–8106, Dec. 2016.

[49] Y. Liu, Y. Wang, R. Sun, and R. Huang, "Hierarchical resource allocation in ultra-dense networks," in *2017 IEEE 86th vehicular technology conference (VTC-fall)*, 2017.

[50] G. Zhang, H. Zhang, Z. Han, and G. K. Karagiannidis, "Spectrum allocation and power control in full-duplex ultra-dense heterogeneous networks," *IEEE Transactions on Communications*, vol. 67, no. 6, pp. 4365–4380, Jun. 2019.

[51] M. Ding, D. Lopez-Perez, H. Claussen, and M. A. Kaafar, "On the fundamental characteristics of ultra-dense small cell networks," *IEEE Network*, vol. 32, no. 3, pp. 92–100, May 2018.

[52] X.-S. Yang, "Swarm-based metaheuristic algorithms and no-free-lunch theorems," in *Theory and new applications of swarm intelligence*, InTech, 2012.

[53] R. Bellman, R. Corporation, and K. M. R. Collection, *Dynamic programming.* Princeton University Press, 1957.

[54] S. Chen, J. Montgomery, and A. Bolufé-Röhler, "Measuring the curse of dimensionality and its effects on particle swarm optimization and differential evolution," *Applied Intelligence*, vol. 42, no. 3, pp. 514–526, Nov. 2014.

[55] Q. He and P. Zhang, "Dynamic channel assignment using ant colony optimization for cognitive radio networks," in *2012 IEEE vehicular technology conference (VTC fall)*, 2012.

[56] H.-y. Gao and J.-l. Cao, "Non-dominated sorting quantum particle swarm optimization and its application in cognitive radio spectrum allocation," *Journal of Central South University*, vol. 20, no. 7, pp. 1878–1888, Jul. 2013.

[57] H. Marshoud, H. Otrok, H. Barada, R. Estrada, A. Jarray, and Z. Dziong, "Realistic framework for resource allocation in macrofemtocell networks based on genetic algorithm," *Telecommunication Systems*, vol. 63, no. 1, pp. 99–110, Mar. 2015.

[58] R. Estrada, H. Otrok, and Z. Dziong, "Resource allocation model based on particle swarm optimization for OFDMA macro-femtocell networks," in *2013 IEEE international conference on advanced networks and telecommunications systems (ANTS)*, 2013.

[59] P.-H. Huang, H. Kao, and W. Liao, "Cross-tier cooperation for optimal resource utilization in ultra-dense heterogeneous networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 12, pp. 11193–11207, Dec. 2017.

[60] L. Xu, Y. Mao, S. Leng, G. Qiao, and Q. Zhao, "Energy-efficient resource allocation strategy in ultra dense small-cell networks: A stackelberg game approach," in *2017 IEEE international conference on communications (ICC)*, 2017.

[61] X. Tang, P. Ren, F. Gao, and Q. Du, "Interference-aware resource competition toward power-efficient ultra-dense networks," *IEEE Transactions on Communications*, vol. 65, no. 12, pp. 5415–5428, Dec. 2017.

[62] J. Chen, Z. Gao, and Q. Zhao, "Load-aware dynamic spectrum access in ultra-dense small cell networks," in *2015 international conference on wireless communications & signal processing (WCSP)*, 2015.

[63] M. Adedoyin and O. Falowo, "QoS-aware radio resource allocation for ultra-dense heterogeneous networks," in *2017 IEEE 28th annual international symposium on personal, indoor, and mobile radio communications (PIMRC)*, 2017.

[64] W. Yao, J. Li, B. Tan, and S. Hao, "Interference management scheme of ultra dense network based on clustering," in *2017 IEEE 2nd information technology, networking, electronic and automation control conference (ITNEC)*, 2017.

[65] Y. Ye, H. Zhang, X. Xiong, and Y. Liu, "Dynamic min-cut clustering for energy savings in ultra-dense networks," in *2015 IEEE 82nd vehicular technology conference (VTC2015-fall)*, 2015.

[66] J. Cao, T. Peng, Z. Qi, R. Duan, Y. Yuan, and W. Wang, "Interference management in ultradense networks: A user-centric coalition formation game approach," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 6, pp. 5188–5202, Jun. 2018.

[67] Z. Zhao, Z. Peng, S. Zheng, and J. Shang, "Cognitive radio spectrum allocation using

evolutionary algorithms," *IEEE Transactions on Wireless Communications*, vol. 8, no. 9, pp. 4421–4425, Sep. 2009.

[68] J. Gross and M. Bohge, "Dynamic mechanisms in ofdm wireless systems : A survey on mathematical and system engineering contributions . A," 2006.

[69] D. Fooladivanda, A. A. Daoud, and C. Rosenberg, "Joint channel allocation and user association for heterogeneous wireless cellular networks," in *2011 IEEE 22nd international symposium on personal, indoor and mobile radio communications*, 2011.

[70] J. Jang and K. B. Lee, "Transmit power adaptation for multiuser OFDM systems," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 2, pp. 171–178, Feb. 2003.

[71] X. S. Yang and X. He, "Firefly algorithm: Recent advances and applications," *International Journal of Swarm Intelligence*, vol. 1, no. 1, p. 36, 2013.

[72] I. H. Osman and G. Laporte, "Metaheuristics: A bibliography," *Annals of Operations Research*, vol. 63, no. 5, pp. 511–623, Oct. 1996.

[73] C. Blum and A. Roli, "Metaheuristics in combinatorial optimization," *ACM Computing Surveys*, vol. 35, no. 3, pp. 268–308, Sep. 2003.

[74] F. Glover, "Future paths for integer programming and links to artificial intelligence," *Computers & Operations Research*, vol. 13, no. 5, pp. 533–549, Jan. 1986.

[75] J. Brownlee, *Clever algorithms : Nature-inspired programming recipes.* Place of publication not identified: Lulu, 2011.

[76] X.-S. Yang and X.-S. He, "Nature-inspired algorithms," in *SpringerBriefs in optimization*, Springer International Publishing, 2019, pp. 21–40.

[77] F. Fausto, A. Reyna-Orta, E. Cuevas, Ángel G. Andrade, and M. Perez-Cisneros, "From ants to whales: Metaheuristics for all tastes," *Artificial Intelligence Review*, Jan. 2019.

[78] M. Črepinšek, S.-H. Liu, and M. Mernik, "Exploration and exploitation in evolutionary algorithms," *ACM Computing Surveys*, vol. 45, no. 3, pp. 1–33, Jun. 2013.

[79] J. D. Ser, E. Osaba, D. Molina, X.-S. Yang, S. Salcedo-Sanz, D. Camacho, S. Das, P. N.

Suganthan, C. A. C. Coello, and F. Herrera, “Bio-inspired computation: Where we stand and whats next,” *Swarm and Evolutionary Computation*, vol. 48, pp. 220–250, Aug. 2019.

[80] T. G. Stützle, “Local search algorithms for combinatorial problems-analysis,” 1998.

[81] A. Reyna-Orta, “Aeroreyna/aisearchmatlab: Alpha working.” Zenodo, 2019.

[82] F. Neumann and C. Witt, “Bioinspired computation in combinatorial optimization,” in *Proceedings of the fourteenth international conference on genetic and evolutionary computation conference companion - GECCO companion 12*, 2012.

[83] A. P. Piotrowski and J. J. Napiorkowski, “Some metaheuristics should be simplified,” *Information Sciences*, vol. 427, pp. 32–62, Feb. 2018.

[84] A. P. Piotrowski and J. J. Napiorkowski, “Searching for structural bias in particle swarm optimization and differential evolution algorithms,” *Swarm Intelligence*, vol. 10, no. 4, pp. 307–353, Nov. 2016.

[85] M. Sipper, W. Fu, K. Ahuja, and J. H. Moore, “Investigating the parameter space of evolutionary algorithms,” *BioData Mining*, vol. 11, no. 1, Feb. 2018.

[86] “Genetic algorithms,” in *Engineering optimization*, John Wiley & Sons, Inc., 2010, pp. 171–180.

[87] F. Marini and B. Walczak, “Particle swarm optimization (PSO). A tutorial,” *Chemometrics and Intelligent Laboratory Systems*, vol. 149, pp. 153–165, Dec. 2015.

[88] R. Storn, “On the usage of differential evolution for function optimization,” in *Proceedings of north american fuzzy information processing.*

[89] E. Cuevas, A. Reyna-Orta, and M.-A. Dı’az-Cortes, “A multimodal optimization algorithm inspired by the states of matter,” *Neural Processing Letters*, vol. 48, no. 1, pp. 517–556, Nov. 2017.

[90] S. Mirjalili and A. Lewis, “The whale optimization algorithm,” *Advances in Engineering Software*, vol. 95, pp. 51–67, May 2016.

[91] H. Shi, R. V. Prasad, E. Onur, and I. G. M. M. Niemegeers, “Fairness in wireless net-

works:Issues, measures and challenges," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 5–24, 2014.

[92] K. Hussain, M. N. M. Salleh, S. Cheng, and Y. Shi, "Metaheuristic research: A comprehensive survey," *Artificial Intelligence Review*, Jan. 2018.

[93] D. I. Escalona-Vargas, I. Lopez-Arevalo, and D. Gutiérrez, "Multicompare tests of the performance of different metaheuristics in EEG dipole source localization," *The Scientific World Journal*, vol. 2014, pp. 1–9, 2014.

[94] J. Cao, T. Peng, Z. Qi, R. Duan, Y. Yuan, and W. Wang, "Interference management in ultradense networks: A user-centric coalition formation game approach," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 6, pp. 5188–5202, Jun. 2018.

[95] C. R. Reeves, "Genetic algorithms," in *Handbook of metaheuristics*, Springer US, 2010, pp. 109–139.

[96] R. Storn and K. Price, "Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[97] S. Das, S. S. Mullick, and P. Suganthan, "Recent advances in differential evolution an updated survey," *Swarm and Evolutionary Computation*, vol. 27, pp. 1–30, Apr. 2016.

[98] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN95 - international conference on neural networks.*

[99] M. R. Bonyadi, Z. Michalewicz, and X. Li, "An analysis of the velocity updating rule of the particle swarm optimization algorithm," *Journal of Heuristics*, vol. 20, no. 4, pp. 417–452, Apr. 2014.

[100] E. Cuevas, A. Echavarrı'a, D. Zaldı'var, and M. Pérez-Cisneros, "A novel evolutionary algorithm inspired by the states of matter for template matching," *Expert Systems with Applications*, vol. 40, no. 16, pp. 6359–6373, Nov. 2013.