

# SmartShell: A Near-Field Reflective Surface Enhancing RSS

Linling Zhong  
Shanghai Jiao Tong University  
uranium\_zll@sjtu.edu.cn

Mingwei Ouyang  
Shanghai Jiao Tong University  
1999mrou@sjtu.edu.cn

Fengyuan Zhu  
Shanghai Jiao Tong University  
jsqdzhu@sjtu.edu.cn

Meng Jin  
Shanghai Jiao Tong University  
Jinm@sjtu.edu.cn

Xinbing Wang  
Shanghai Jiao Tong University  
xwang8@sjtu.edu.cn

Xinping Guan  
Shanghai Jiao Tong University  
xpguan@sjtu.edu.cn

Chenghu Zhou  
Chinese Academy of Sciences  
zhouch@reis.ac.cn

Xiaohua Tian\*  
Shanghai Jiao Tong University  
xtian@sjtu.edu.cn

## ABSTRACT

Reconfigurable reflective arrays can be used to program the radio propagation environment in order to form favorable wireless channel conditions. Previous designs have used large-scale arrays containing hundreds to thousands of reflecting elements located external to the receiving node, with the reflection coefficients of all array elements managed by a controller. However, these designs can be costly to deploy and are challenging to quickly adapt to the time-varying nature of wireless channels caused by mobility.

In this paper, we propose using a small-sized and low-cost reflective surface attached to the mobile device, called SmartShell, to program the micro-environment of radio propagation near the device. We provide a systematic analysis of the near-field reflective surface, laying the theoretical foundation for the feasibility and effectiveness of SmartShell. We then present insight into controlling the surface units of SmartShell, showing that this is fundamentally different from existing far-field designs and can be formulated as a binary quadratic programming (BQP) problem, with performance bounds derived. We develop a prototype of the SmartShell system and conduct comprehensive experiments. Results show that the SmartShell can increase the received signal strength (RSS) of the wireless node by 5-10 dB, which is comparable to far-field large-scale reflective surfaces but at a much lower cost.

## CCS CONCEPTS

• Hardware; • Networks → Network components;

## KEYWORDS

Near-field, Reflective array, RSS

\*Corresponding author

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

*MobiSys '23, June 18–22, 2023, Helsinki, Finland*

© 2023 Association for Computing Machinery.

ACM ISBN 979-8-4007-0110-8/23/06...\$15.00

<https://doi.org/10.1145/3581791.3596866>

## ACM Reference Format:

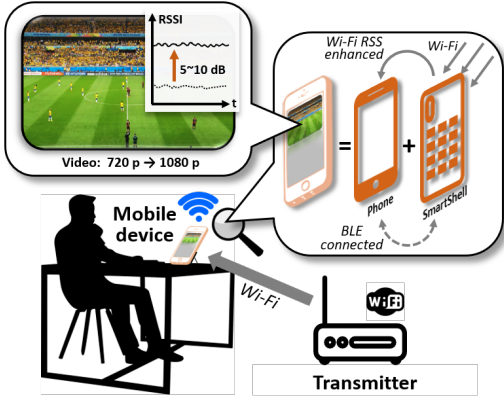
Linling Zhong, Mingwei Ouyang, Fengyuan Zhu, Meng Jin, Xinbing Wang, Xinping Guan, Chenghu Zhou, and Xiaohua Tian. 2023. SmartShell: A Near-Field Reflective Surface Enhancing RSS. In *ACM International Conference on Mobile Systems, Applications, and Services (MobiSys '23), June 18–22, 2023, Helsinki, Finland*. ACM, New York, NY, USA, 13 pages. <https://doi.org/10.1145/3581791.3596866>

## 1 INTRODUCTION

Many Internet of things (IoT) systems are based on wireless communications, where the fundamental challenge arises from unpredictable wireless channels. As the wireless channel condition is closely related to electromagnetic (EM) waves propagation environment, realizing a controllable or influenceable EM wave propagation environment has been a long-standing desire. Efforts in this vein include metasurface [1, 2], intelligent reflecting surface (IRS) [3, 4], smart reflectarray [5], reconfigurable intelligent surface (RIS) [6–8], LAIA [9, 10], RFocus [11, 12]. Although varying in terminologies, such schemes share the same essence: leveraging passive and tunable reflecting surfaces to change propagation properties of EM waves, and then change the wireless channel condition [13]. This is in contrast to conventional wireless design philosophy, which only regards channel condition as a constraint [14, 15].

The basic requirement of wireless communication is to increase the RSS, which straightforwardly can be achieved with directional or multiple antennas [16]. However, such solutions can violate the IoT devices' size, expense and power consumption constraints [11]. In order to reap equivalent benefit of antenna array, it is proposed to amplify the received signal by mounting antennas on the external reflection surface [9, 11]. The surface is composed of thousands of simple elements organized in an array, where each element can switch among different states. All the elements are centrally controlled to form a favorable wireless channel condition for increasing the RSS.

Effective utilization of the reflecting surface faces following uncertainties. First, the reflected EM wave energy is proportional to  $\frac{A}{d^2}$  [17], with  $A$  and  $d$  denote the surface area and surface-to-receiver distance, respectively. Thus the big-sized surface needs to be installed with thousands of passive antennas, which may incur installation difficulties or cost issues. Second, the receiving node has to provide feedback to the centralized controller for the surface to coordinate and optimize the states of those reflecting elements,



**Figure 1: SmartShell attached to the mobile devices and improve RSSI by 5 ~ 10dB.**

but the effective communication between the end node and the controller itself can be a challenge. Third, existing algorithm designs for controlling the reflecting surface can hardly accommodate mobility. This is because mobility makes the channel condition even more complicated, which makes it more challenging to quickly coordinate thousands of reflecting elements.

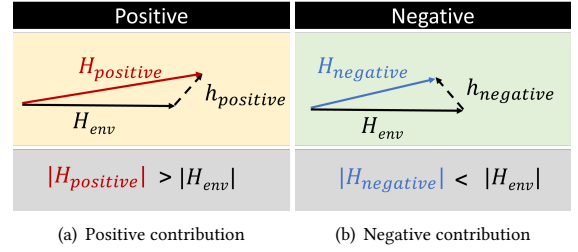
In this paper, we propose *SmartShell* as shown in Fig. 1: using small-sized and low-cost reflecting surface attached to the shell of the IoT device, in order to program the micro-environment of radio propagation near to the receiving antenna in the IoT device. *SmartShell* is made of a small piece of printed circuit board (PCB). Though with smaller reflecting area  $A$  and fewer reflecting elements compared with existing large-scale array designs, *SmartShell* can still remarkably increase the RSS, because it is near to the receiving antenna with a small  $d$ . This significantly mitigates the deployment difficulties and costs. Moreover, it is easier to control a small number of reflecting elements, in order to quickly respond to time-variability of the wireless channel in the mobile condition.

However, the following technical issues have to be figured out before the basic idea of *SmartShell* could be carried out:

- The *SmartShell* surface is inevitably close to the device’s own antenna, which could incur near-field coupling. Will such coupling neutralize the expected RSS enhancement effect from the *SmartShell*?
- How to control the *SmartShell* reflective units? Could we just apply the control algorithm used by existing far-field RIS such as RFocus to the *SmartShell*? How good can the *SmartShell* achieve in terms of RSS enhancement theoretically?

In clarifying those issues, we make the following contributions.

- We provide a mathematical analysis of the near-field reflective surface, laying the theoretical foundation for the feasibility and effectiveness of *SmartShell*. We reveal how to make key design decisions on the size of the reflective unit and the distance between the units and the device’s antenna in order to avoid potential coupling effects. We show that the near-field surface can be more effective than the existing far-field counterpart by 1-2 orders of magnitude under the same size. This is the first time that this has



**Figure 2: The signal reflected by the element contributes either (a) positively or (b) negatively to the received signal.**

been studied, providing a theoretical foundation for the design of near-field reflective surfaces.

- We then present the insight for controlling the *SmartShell* surface units. Unlike far-field RIS, which can utilize a voting method for control, controlling near-field units is more challenging because the feasible control strategies in the two scenarios form different spaces. We formulate the control of near-field units as a binary quadratic programming (BQP) problem, where the upper and lower performance bounds are obtained.
- We develop a prototype of *SmartShell*, based on which we conduct comprehensive experiments. The surface is powered by electromagnetic coil based wireless charging; an APP is running in the background to control the surface units through Bluetooth communication as used by common wireless auxiliaries. Results show that the *SmartShell* can enhance RSS by 5 ~ 10dB on average, while has no effect on the LTE/Wi-Fi signal receiving when power off.

## 2 NEAR-FIELD REFLECTIVE SURFACE

In this section, we first list a few known results in modeling the wireless channel (particularly in the case with a smart surface in the environment). Then we discuss why a near-field surface is advantageous over a far-field surface. At last, we discuss the near-field coupling effect, which is the key challenge we meet in designing a small-sized near-field surface.

### 2.1 Near-field v.s. Far-field

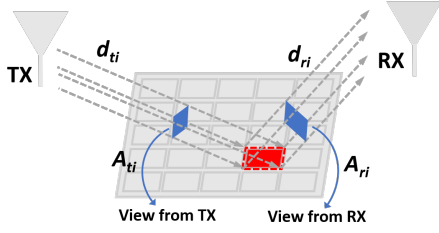
Wireless channels describe how the signal changes as it propagates from transmitter to receiver. They are a direct function of the paths along which the signal propagates. In particular, the channel of a narrowband signal traversing a single path is given by:

$$h = \alpha \cdot e^{j\phi} \quad (1)$$

where  $\alpha$  and  $\phi$  represent the amplitude change ( $\alpha^2$  is also known as channel gain) and phase shift introduced by the channel, respectively. According to Friis path loss model, we have:

$$\alpha = \sqrt{\frac{A}{4\pi d^2}} \quad \text{and} \quad \phi = \frac{2\pi d}{\lambda} \quad (2)$$

where  $d$  is the propagation distance of the signal,  $A$  is the antenna area, and  $\lambda$  is the wavelength.



**Figure 3: The project of the element’s surface to the vertical plane of the Tx-to-element direction and element-to-Rx direction.**

Then we consider the case in a indoor environment with a reflective surface. In this case, the signal travels along multiple paths, and a subset of the paths pass via the  $N$  elements on the reflective surface. Assuming that the elements are independent of each other with negligible coupling between them (we will show in Sec. 2.2 that how we can remove the inter-element coupling and demonstrate it experimentally in Sec. 6.1), the channel at the receive antenna can be written as the linear addition of the effect of all elements and that of the environment:

$$h = H + \sum_{i=1}^N h_i \cdot \Gamma_i \cdot b_i \quad (3)$$

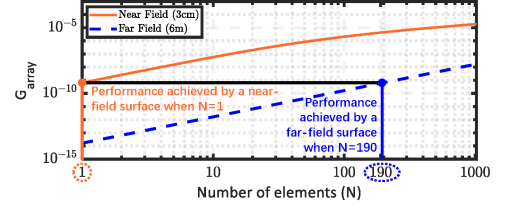
where  $H$  is the combination of the paths not going via the surface;  $h_i$  denotes the signal attenuation on the path traversing through element  $i$ . We have  $h_i = h_{ti} \cdot h_{ri}$ , where  $h_{ti}$  and  $h_{ri}$  respectively denotes the channel from source to the  $i^{th}$  element and the channel from the  $i^{th}$  element to the receiver.  $\Gamma_i$  is the reflection coefficient when the  $i^{th}$  element is on. Since the elements are passive and lack power, we have  $|\Gamma_i| < 1$ .  $b_i \in \{0, 1\}$  denotes whether the  $i^{th}$  element is off or on

To maximize the RSS, the surface has to control the on and off state of each element (i.e., each  $b_i$  in Eq.(3)), so that all the element that is turned on have positive contribution to the received signal. Fig. 2 shows two examples where the signal reflected by the element has a positive or negative contribution to the received signal. In Sec. 3 and 4.3, we will discuss the principle and method for setting the state of each element  $i$  to optimize the RSS. This will involve a detailed analysis of the mathematical problem of the SmartShell system and the algorithms that we have developed to maximize its effectiveness.

To further understand why a small-sized near-field surface (like SmartShell) is advantageous over a large-sized far-field surface (like RFocus), we provide a theoretical analysis on how the scale of the surface (e.g., its area and the number of elements on the surface) and its distance to the receiver ( $d$ ) affects the maximal achievable signal strength.

Specifically, assuming that all the elements have optimal contribution, then based on Eq. (2) and (3), the free-space channel gain obtained by the surface is upper bounded by:

$$G_{array} = \left| \sum_{i=1}^N |h_i| \cdot \Gamma_i \right|^2 = \left| \sum_{i=1}^N \sqrt{\frac{A_{ri}}{4\pi d_{ri}^2}} \cdot \sqrt{\frac{A_{ti}}{4\pi d_{ti}^2}} \cdot \Gamma_i \right|^2 \quad (4)$$



**Figure 4:  $G_{array}$  obtained with different element numbers  $N$  and different surface-to-receiver distances when the frequency band is 5.8GHz.**

where  $d_{ti}$  and  $d_{ri}$  respectively denote the distance from source to the  $i^{th}$  element and the distance from the  $i^{th}$  element to the receiver.  $A_{ti}$  and  $A_{ri}$  are the project of the element’s surface to the vertical plane of the Tx-to-element direction and element-to-Rx direction, as shown by Fig. 3. As can be seen in Eq. (4), *the upper-bound signal strength increases linearly with the number of the elements while decays quadratically with signal propagation distance*. That is to say, the propagation distance dominates the channel gain, and this is why a small-sized near-field surface is advantageous over a large-sized far-field surface.

Fig. 4 shows how the number of elements  $N$  impact the gain of the reflective surface  $G_{array}$ , under different distance between the center of the surface and the receive antenna (3cm for the near field and 6m for the far field). The results are obtained at a 5.8GHz frequency band and when the distance between the transmitter and the surface is fixed at 6m. As can be seen from the figure, a small sized near-field array can achieve a equivalent performance of a large-sized far-field array. For example, we can see that a single element on a near-field reflective array can achieve an equivalent signal gain compared with 190 elements on a far-field array. This motivates us to design a small-sized array that attached to the smartphone to enhances the signal strength around the receive antenna.

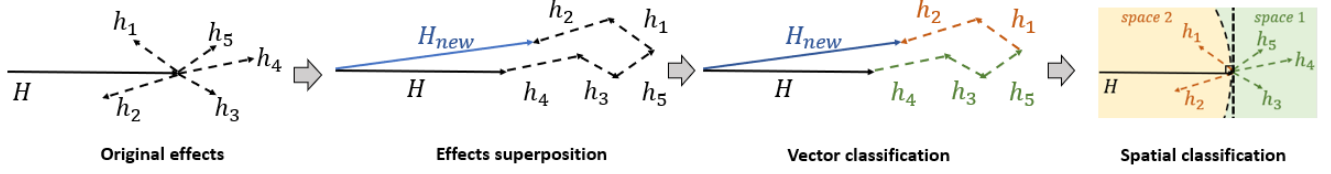
## 2.2 Near-field Coupling Effect

To design a small-sized near-field surface, a key challenge however is how to mitigate the coupling effect brought by such a design. There are basically two types of coupling effect on a small-sized near-field surface: i) *external coupling*, which refers to the coupling between the surface and the smartphone; and ii) *internal coupling*, which refers to the coupling between nearby elements on the surface. We in this section discuss how we can address these two types of coupling effect.

**External coupling effect.** The external coupling occurs when the distance from the surface to the receive antenna is shorter than the distance  $d_E$  [18, 19], which is defined as:

$$d_E = \frac{D^2 N}{5\lambda} \quad (5)$$

where  $D$  is the antenna size. For a tablet or a smartphone, the antennas are often installed at the edge of the device to avoid signal shielding, so there is enough space to ensure that the distance between the array and the antenna is greater than  $d_E$ . For example, when we use a 25-element surface which works at 5GHz band and



**Figure 5:  $H$  and  $h_i$  are in the complex number domain. The RSS depends on the superposition of  $H$  and  $h_i$ . According to the effect of  $h_i$  after superposition, the complex vector classification of  $h_i$  can be performed. Eventually, it can classify  $h_i$  into two different spaces.**

with the size of 6.5cm, the distance is only 1.6cm. Thus the coupling effect can be easily avoided by carefully arranging the location of the reflective surface on the shell.

**Internal coupling effect.** When the spacing between two elements on the surface is smaller than  $\lambda/2$ , the internal coupling effect occurs. For an array of (say  $m \times n$ ) elements, the internal coupling between them can be characterized by a  $m \times n$  matrix, denoted as  $\mathbf{H}_{\text{internal}}$ , where  $h_{ij}$  denotes the influence of the  $i_{th}$  element on the  $j_{th}$  element, which is related to the inter spacing between elements  $i$  and  $j$ . To minimize the internal coupling, we expect that  $h_{ij} = 0, i \neq j$ . We propose two metrics  $\xi_1$  and  $\xi_2$  to quantify the degree of internal coupling effect of a surface:

$$\xi_1 = \frac{\sum_{i=1}^N \sum_{j=1, i \neq j}^N |h_{ij}|}{N(N-1)} \quad (6)$$

$$\xi_2 = \frac{\sum_{i=1}^N |h_{ii}|}{N}$$

Where,  $\xi_1$  characterizes the linearity of the array. A higher  $\xi_1$  means a lower coupling effect between elements.  $\xi_2$  captures how much the surface can strength the received signal. We will show in Sec. 4.2 a element arrangement method which aims to minimize  $\xi_1$  and at the same time maximize  $\xi_2$ .

### 3 CONTROLLING NEAR-FIELD REFLECTING ELEMENTS

#### 3.1 Problem Analysis

The goal of SmartShell is to maximize signal strength at the receiver, which can be modeled as the following optimization problem:

$$\mathbf{b}^* = \arg \max_{\mathbf{b}} |h| = \arg \max_{\mathbf{b}} |H + \sum_{i=1}^N h_i \cdot b_i| \quad (7)$$

where  $\mathbf{b} = [b_1, \dots, b_N]$  is the state (on or off) of all the  $N$  elements. The maximum signal strength is achieved when all the elements that have positive contribution to the received signal are turned on. Fig. 5(d) illustrates this problem on the complex plane, where  $H$  and  $h_i$  are represented as vectors on that plane. Then if  $h_i$  forms an acute angle with  $H$  (or equivalently,  $H \cdot h_i > 0$ ), we consider that element  $i$  has a positive contribution to the received signal. Otherwise, we consider the element has a negative contribution to the received signal.

However, a challenge we meet in solving the above optimization problem is that both  $H$  and  $h_i$  are hard to measure in practice.

That is to say, we need to determine whether an element  $i$  should be turned on or off, without knowing the length and direction of  $h_i$ . A naive solution to this problem is to first turn off all the elements, and then turn on each element  $i$  individually to observe the difference between  $|H|$  and  $|H + h_i|$ . The element which makes  $|H| < |H + h_i|$  is considered as a positive element. We divide the space into two regions: *space 1* and *space 2*. All the  $h_i$  which satisfy  $|H| < |H + h_i|$  locate on *space 1*, as shown in Fig. 5. While, the others locate on *space 2*. We term the boundary between spaces 1 and 2 as the *decision boundary*. However, we meet a problem here that, in practice, the state of each element  $i$  (i.e.,  $b_i = 1$  or 0) should not be determined individually. As the example shown in Fig. 6, all the four elements positive contribution to the received signal. However, when  $h_3$  is selected, the decision boundary will change, as shown in Fig. 6(d-e). Then element 2 becomes a negative element. The above phenomenon can be mathematically explained with the following lemmas:

**Lemma 1.** For any channel vector  $H$ , if we denote the set of channel vectors  $h_i$  which satisfy  $|H + h_i| > |H|$  as  $\odot(H) = \{h_i \mid |H + h_i| > |H|\}$ , then we have:

$$\odot(H) \neq \odot(H + h_i) \quad (8)$$

Lemma 1 means that once a new element  $i$  is selected, the decision boundary between space 1 and space 2 will change. At this time, we need to re-estimate the contribution of the remaining elements based on the new  $H$ .

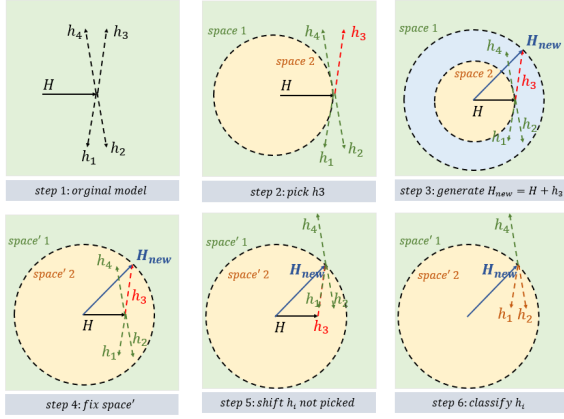
**Lemma 2.** For any  $H$ , there exists  $h_i$  and  $h_j$  that satisfy  $h_i, h_j \in \odot(H)$ , while  $h_j \notin \odot(H + h_i)$ .

*Proof.* According to the definition of  $\odot(H)$ , we can get  $\forall h_i \in \odot(H) \Leftrightarrow H + h_i \notin \odot(H)$ . Let  $\theta = \cos \langle H, h_i \rangle$ , which is the cosine between  $H$  and  $h_i$ . When  $h_i$  satisfies  $-\frac{1}{2} \frac{|h_i|}{|H|} < \theta < \frac{1}{2} \frac{|h_i|}{|H|}$ , we have

$$\begin{cases} |H + h_i| > |H| \\ |H - h_i| > |H| \end{cases} \quad (9)$$

So there exists  $h_j = h_i \cdot e^{j\pi} = -h_i$ , which satisfies  $h_j \in \odot(H)$ . Since  $|H + h_i + h_j| < |H + h_i|$ , we can get  $h_j \notin \odot(H + h_i)$ .

The above lemmas tell that: each time after selecting a  $h_i$  located in *space 1*, the decision boundary needs to be updated. Fig. 6 illustrates the process of updating the decision boundary. That is to say, we need to re-estimate the contribution (positive or negative) of all element once a new  $h_i$  is selected, which is a time-consuming process. Some existing works (e.g., RFocus [11]) can solve this problem using proposed Majority voting. However, the premise of using this algorithm is  $N \rightarrow \infty$  and  $|\mathbf{b} \cdot \mathbf{h}| \ll |H|$ , so that the change in



**Figure 6:**  $h_1, h_2, h_3, h_4$  in the original model are all located in space1. When  $h_3$  is selected, in the fixed mapping space the positions of  $h_1$  and  $h_2$  become space2. At this time, selecting these two vectors will cause the amplitude to decrease.

the decision boundary when adding a new  $h_i$  is negligible. However, SmartShell is limited in size and cannot use a large number of antenna elements, so the number of  $N$  is limited. Furthermore, due to the proximity of SmartShell to the receiving end, the impact of each  $h_i$  is non-negligible. So, we cannot use Majority voting in SmartShell to determine the state of each element.

### 3.2 Problem Formulation

To solve the above problem, we formulate the optimization problem in Eq. (7) as a BQP (Binary Quadratic Programming) problem. Specifically, we perform the following transformation on Eq. (7):

$$\mathbf{b}^* = \arg \max_{\mathbf{b}} |H| \left| e^{j\varphi_H} + \sum_{i=1}^N e^{j\varphi_i} \cdot \frac{|h_i|}{|H|} \cdot b_i \right| \quad (10)$$

Further, let

$$g(\mathbf{b}) = \left| e^{j\varphi_H} + \sum_{i=1}^N e^{j\varphi_i} \cdot \frac{|h_i|}{|H|} \cdot b_i \right|^2 \quad (11)$$

Obviously, the maximum position of  $g(\mathbf{b})$  is the same as that of Eq. (7).  $g(\mathbf{b})$  is a BQP problem. A BQP problem refers to a class of mathematical models of the form Equation (12), which has been proven to be an *NP-Hard* problem[20].

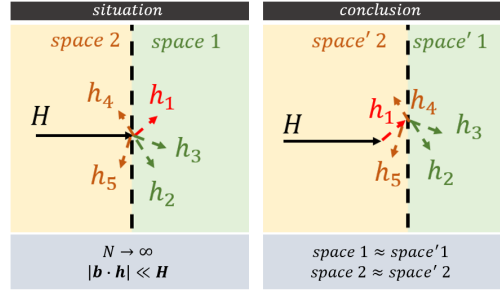
$$f(\mathbf{x}) = \mathbf{x}^T \mathbf{Q} \mathbf{x} = \sum_{i=1}^N \sum_{j=1}^N q_{ij} x_i x_j, x_i \in \{0, 1\} \quad (12)$$

*Proof.* Define an operator:

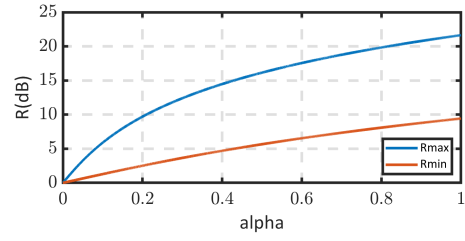
$$O(f(\varphi)) = \begin{pmatrix} f(\varphi_H) \\ \frac{|h_1|}{|H|} f(\varphi_1) \\ \vdots \\ \frac{|h_N|}{|H|} f(\varphi_N) \end{pmatrix} \quad (13)$$

Let

$$\mathbf{Q} = O(\cos(\varphi)) \cdot O(\cos(\varphi))^T + O(\sin(\varphi)) \cdot O(\sin(\varphi))^T \quad (14)$$



**Figure 7:** The voting method is used on the premise that the choice does not affect the vector space.



**Figure 8:** The performance and indicators of the reflective array.

So it can be proved that

$$g(\mathbf{b}) = (1 \quad b_1 \quad \dots \quad b_N) \mathbf{Q} \begin{pmatrix} 1 \\ b_1 \\ \vdots \\ b_N \end{pmatrix} \quad (15)$$

Therefore, Eq. (7) can be transformed into a BQP problem. In the previous work on the BQP problem, the use of greedy and local search heuristics is a more effective method, which can achieve an approximate optimal solution [21–24]. Therefore we adopt heuristics methods to solve this problem.

In addition, BQP problems can also be solved by conic relaxations [25]. However, the premise of using this method to solve is that  $\mathbf{Q}$  is known, which means the parameters of the whole system are known, so it is not suitable for this system.

### 3.3 Performance Bounds

In this section, we discuss the performance boundaries of this BQP problem. In the wireless channel environment, the multipath effect usually causes Rayleigh fading of the signal. In the Rayleigh fading model, it can be inferred that the phase of the signal is uniformly distributed over  $[-\pi, \pi]$  and independent of each other. From a statistical point of view, since the signals at the receiving end come from different paths, all phases appear with equal probability, so the uniform distribution of the phases is reasonable.

On the premise that the phase is uniformly distributed on  $[-\pi, \pi]$ , the performance that this system can improve can be estimated:

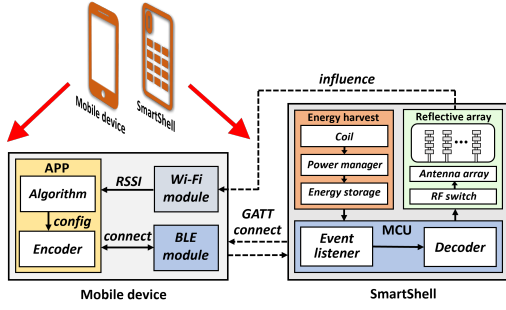


Figure 9: System workflow.

$$R = E \left( \frac{|H + \sum_{i=1}^N h_i \cdot b_i|^2}{|H|^2} \right) \quad (16)$$

$R$  represents the ratio that can be improved relative to the original signal strength. Since the phases are uniformly distributed, theoretically half of the elements in  $b$  are set to 1. Therefore, the performance of the optimal case can be calculated as

$$R_{max} = E \left( \frac{|H + \sum_{i=1}^{\frac{N}{2}} h_i|^2}{|H|^2} \right) \quad (17)$$

It can be known from Section 3.1 that the optimal selection situation should satisfy  $-\frac{\pi}{2} \leq \varphi_i - \varphi_H \leq \frac{\pi}{2}$ . Therefore, under this condition,  $E(\cos(\varphi_i - \varphi_H)) = \frac{\pi}{2} E(\sin(\varphi_i - \varphi_H)) = 0$ . The distance of each element from the transmitter is approximately equal, so ideally the values of  $|h_i|$  are approximately equal, and the effect of  $|h_i|$  on  $R_{max}$  can be calculated accordingly. Let  $\alpha = \frac{|h_i|}{|H|}$ ,  $0 < \alpha < 1$ , then

$$R_{max} = R(N) = 1 + \frac{2N}{\pi} \alpha + \frac{2N^2 - 4N}{\pi^2} \alpha^2 + \frac{N}{2} \alpha^2 \quad (18)$$

For the worst case, it can be considered that the entire system has been randomly sampled  $N$  times, assuming that  $\log_2 N$  elements are exhaustively enumerated, it can be considered that there are  $\log_2 N/2$  elements set to 1 in the same way. So the worst case can be calculated,

$$R_{min} = R(\log_2 N) \quad (19)$$

Fig. 8 shows the curve of  $R_{max}$  and  $R_{min}$  as a function of  $\alpha$  when  $N = 25$ . Therefore, in actual deployment, the value of  $R_{min}$  can be used as an indicator. When the boost value of signal strength is greater than  $R_{min}$ , the system realizes signal strength optimization. Taking  $\alpha = 0.2$  as an example, when the signal strength is increased by more than 2.5dB, the SmartShell optimization can be considered successful.

## 4 DESIGN

### 4.1 System Overview

The overall design of the system is shown in Fig. 9. The system consists of five main parts: (1) the array optimization algorithm running in the mobile device. It inputs the RSSI (Received Signal Strength Indicator) value from Wi-Fi module and outputs array control commands. (2) The Wi-Fi RSSI collection thread running

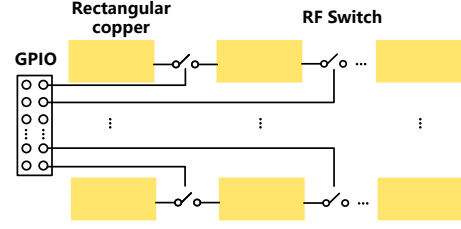


Figure 10: Antenna array design scheme: including GPIO ports controlled by MCU, RF switches, and rectangular copper sheets.

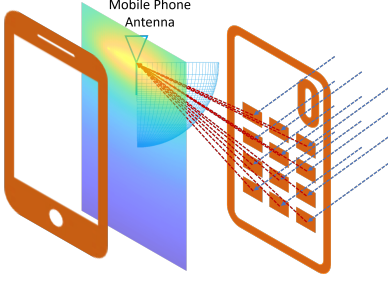
in the mobile device that calls system function to measure real-time RSSI values and put them to the optimization algorithm as the feedback. We use `getRssi()` [26] to obtain RSSI value in the resolution of 1dBm at the sampling period of 100ms. We use the RSSI reported by the AP (Access Point) Beacon frame to ensure it is periodically collected and works as a stable feedback. (3) The BLE (Bluetooth Low Energy) configuration mechanism that conveys the array control commands generated by the mobile device to the SmartShell MCU (Microcontroller Unit). (4) The SmartShell array attached in the device shell which accepts the MCU commands to control each RF switch state. A half-dipole antenna is formed when an RF switch is turned on by joining two rectangular copper sheets together. (5) An energy harvesting circuit that provides energy for the SmartShell hardware independent of the mobile device. Here parts (1) and (2) are integrated into a single Android APP; and we select a low-power wireless MCU in (4), allowing the MCU to regularly enter a low-power state, further reducing the system's power consumption.

The main workflow is as follows: the app obtains the RSSI value from the Wi-Fi module, iterates the algorithm based on the RSSI value, and outputs the configuration information of the array according to the algorithm. This information is then encoded and sent to the event listener of SmartShell via the Bluetooth GATT (Generic Attribute Profile) protocol. Finally, SmartShell decodes the information and configures the array accordingly.

### 4.2 Array Design

As discussed in Sec. 2.2, the way to avoid the internal coupling situation is to make  $H_{internal}$  a diagonal matrix. Therefore, two requirements must be satisfied in the design of the array: (A)  $\xi_1 \approx 0$ ; (B)  $\xi_2 \gg \xi_1$ . Condition A ensures that the  $i_{th}$  ports do not impact the  $j_{th}$  ports, minimizing internal coupling. Condition B ensures that the array can operate effectively without being affected by non-linearity, with a higher value of  $\xi_2$  resulting in better performance.

Based on the design principles discussed above, we design the antenna array for deployment in smartphones in the 5 GHz band, as shown in Fig. 10. The 5 GHz band is more suitable for smartphone-sized arrays than the 2.4 GHz band. The rectangular copper pieces are created by laying copper, which can reflect the antenna characteristics and save on cost. Through High-Frequency Simulator Structure (HFSS) simulations, we achieve the design that satisfies conditions A and B as shown in Table 1, and the performance loss of the array relative to the decoupled array is within an acceptable



**Figure 11: Diagram of how the signal is reflected to the antenna of the user device under the near-field condition.**

range. The size of each rectangular copper piece is approximately  $\lambda/10 \times \lambda/4$ , and the antenna spacing of each row is approximately  $\lambda/10$ . We compare the performance of the array with that of a decoupled array in Sec. 6.1 and show that the impact on performance from nonlinearity is acceptable.

Array	$\xi_1$	$\xi_2$	$\xi_2/\xi_1$
SmartShell	0.0097	0.4736	16.886 dB
Decoupled	0.0078	0.5395	18.399 dB

**Table 1: Performance parameters of different arrays.**

### 4.3 Optimization Algorithm

To control the reflective array more efficiently and in real-time, the algorithm needs to be optimized within  $N$  iterations. Since the system is a nonlinear problem, it is impossible to find the optimal solution by solving the parameters of the system in a short time.  $\mathbf{b}^*$ , as mentioned in Sec. 3.2, can be transformed into a BQP problem to solve, so we use the heuristic algorithm to solve the problem.

We designed an algorithm called quick start (QS) based on simulated annealing (SA) algorithm and genetic algorithm (GA). This combination has two main advantages: (1) The iteration phase of SA can leverage the  $\Delta RSSI$  caused by multiple  $b_i$  changes in a single measurement. (2) The mutation phase of GA is for searching a local optimal in limited trials. When using SmartShell, the user device's antenna is in the near-field where the waves are spherical. In order to focus the incident energy on the antenna of the user device, the optimal scattered waves should converge on the user device's antenna. The QS algorithm function is to control the state of the SmartShell array elements to converge the energy of the incident plane wave in the form of a spherical reflected wave at the user device antenna, as shown in Fig. 11.

In the QS phase, the device will continuously monitor the RSSI value. If it is detected that the signal strength is weakened at  $t_n$ , the optimization is started. Algorithm 1 shows the optimization operation performed at  $t_n$ , which is essentially a heuristic algorithm. First, generate a random 0/1 vector  $\mathbf{b}_1$ , with the length equal to the number of array elements, corresponding to whether the element is in the input or the output state. Configure the array according to the value of  $\mathbf{b}_1$ , and then measure the RSSI. We note that due to the multipath effect, there will be other copies of the signals arriving at the SmartShell array from other paths even in the case the SmartShell array is turned back toward the AP. Therefore, the RSSI

---

#### Algorithm 1: QS phase at time $t_n$

---

**Result:**  $\mathfrak{R}$  : Optimized RSSI  
 $b_{opt} \leftarrow$  The final optimized state ;  
 $Max \leftarrow$  Maximum of RSSI ;  
 $\mathbf{b}_1 \leftarrow$  The vector with random 0 or 1 ;  
 $flag \leftarrow 0$  ;  
 $\Delta RSSI \leftarrow$  Relative to the RSSI at time  $t_{n-1}$  ;  
get  $RSSI_1$  ;  
**for**  $i \leftarrow 0$  **to**  $K$  **do**  
    **if**  $flag == 0$  **then**  
         $\mathbf{b}_2 \leftarrow Distortion(\mathbf{b}_1)$  ;  
        get  $RSSI_2$  ;  
         $flag \leftarrow 0$  ;  
        **if**  $RSSI_2 > RSSI_1$  **then**  
             $\mathbf{b}_1 \leftarrow \mathbf{b}_2$  ;  
            **if**  $RSSI_2 - RSSI_1 > Threshold$  **then**  
                 $\mathbf{b}_2 \leftarrow Mutation(\mathbf{b}_1)$  ;  
                 $flag \leftarrow 1$  ;  
             $RSSI_1 \leftarrow RSSI_2$  ;  
        **if**  $RSSI_2 > Max$  **then**  
             $b_{opt} \leftarrow \mathbf{b}_2$  and  $Max \leftarrow RSSI_2$  ;  
 $\mathfrak{R} \leftarrow RSSI_1$ .

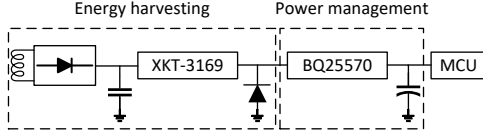
---

changes whenever the value of  $\mathbf{b}_1$  changes. Next, randomly flip half of the elements in  $\mathbf{b}_1$ , corresponding to the function  $Distortion()$  in Algorithm 1, and then measure the RSSI according to the flipping result. If the RSSI increases, the new  $\mathbf{b}_1$  is used as the new iteration starting point. If the increase in RSSI exceeds a certain threshold, it can be considered as a better combination, and we can make minor changes to find a better combination near this point, such as flipping the value of a random element in  $\mathbf{b}_1$ . This operation corresponds to the function  $Mutation()$ . The variable  $flag$  in Algorithm 1 is used to determine whether to execute  $Distortion()$  or  $Mutation()$ . Experiments show that QS, SA and greedy algorithms achieve relatively good optimization results, which are shown in Section 6.5.

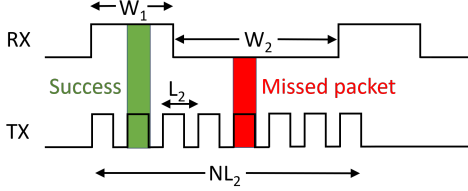
### 4.4 Wireless Configuration

Implementation of a method for mobile devices to transmit configuration information to SmartShell should not impact normal operation of the device. Therefore, a wired communication method cannot be used. Use of sound waves would occupy the device's audio output and use of vibration would affect the user experience.

We use BLE as the wireless technique where Bluetooth's Generic Access Profile (GAP) is used to control device connections and broadcasts[27–29]. The GAP defines several roles for the device, the two main ones being: Peripheral and Central. In this design scenario, the mobile device is used as the central device and the MCU is used as the peripheral device. The central device can be connected to multiple peripheral devices. Therefore, this scheme does not influence the use of other devices of the phone user, like Bluetooth headsets. The low power Bluetooth connection is based on the GATT protocol, which defines the way two Bluetooth low



**Figure 12: The energy supply consists of two main modules: the energy harvesting module and the power management module.**



**Figure 13: Duty cycle design of smartphone-to-MCU BLE command delivery mechanism.**

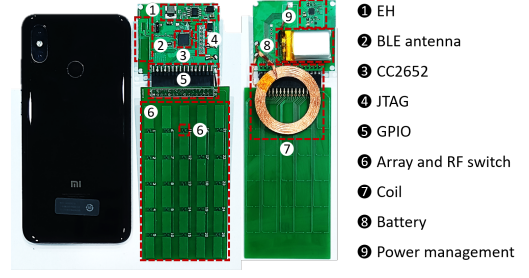
power devices transfer data back and forth using a concept called Service and Characteristic. To send configuration information to the MCU, the data can be written to a Characteristic with write access, thus enabling the transfer of information.

Note that the design solution with Bluetooth means that an antenna needs to be installed to the SmartShell, so the coupling effect of the antenna to the array also has to be considered. Therefore, as described in Sec. 2.2, the antenna should be kept at a distance  $d_E$  from the array during the design process to avoid coupling between the two.

## 4.5 Low Power Design

**4.5.1 EH design.** In order to address the issue of the SmartShell’s energy supply, we utilize energy harvesting technology. Solar or thermal energy is limited by the application scenario[30]: users are unlikely to use the device in sunny or high temperature environments for long periods of time, so it is not suitable. Therefore, the use of RF[31] or coil energy harvesting is a better solution. However, RF energy harvesting would cause frequency interference with antenna array and low charging efficiency. Therefore coil energy harvesting is a better solution and solves these two problems: on the one hand, the mechanism of coil energy acquisition is consistent with the wireless charging of mobile devices, so users can charge the SmartShell while wirelessly charging their mobile devices; on the other hand, the coil works at MHz, which will not interfere with the normal use of the SmartShell.

As shown in the Fig. 12, the energy supply consists of two main modules: the energy harvesting module and the power management module. After the coil receives the signal from the transmitter coil, it is converted into DC by a rectifier circuit and then regulated by chip XKT-3169 to output 5V, which can be used as a regulated power supply module to supply power to the following circuit. The transmission coil contains 150 strands, and is matched at 1.5MHz. We designed the energy management module based on the bq25570



**Figure 14: SmartShell hardware.**

chip[32], whose primary function is to efficiently control the usage of the energy and store the energy collected by the coil in a rechargeable battery.

**4.5.2 Duty Cycle.** To save power, the MCU needs to enter the low-power state at regular intervals.  $W_2$  in Fig. 13 indicates the time when the MCU enters the low-power consumption, and  $W_1$  indicates the time when it enters active which includes listening state time  $t_{listening}$  and array configuration time  $t_{array}$ . In order to ensure that the mobile device successfully transmits the configuration information, it needs to send the same packet repeatedly.  $L_2$  indicates the time occupied by the mobile device each time it sends a packet, and  $N$  is the number of repeated packets. If the packet is to be received by the MCU within the time of  $NL_2$ , it needs to satisfy

$$NL_2 > W_2 + t_{listening} \quad (20)$$

In addition to that, the MCU needs to receive a complete packet in the time period of  $t_{listening}$ , so it also needs to satisfy

$$t_{listening} \geq 2L_2 \quad (21)$$

Satisfying the two conditions will guarantee the successful transmission of each configuration message.

After the mobile device sends the configuration information, it needs to wait for a period of  $t_{array}$  for the MCU to configure the configuration information to the array. Then the period of the mobile device sending configuration information is

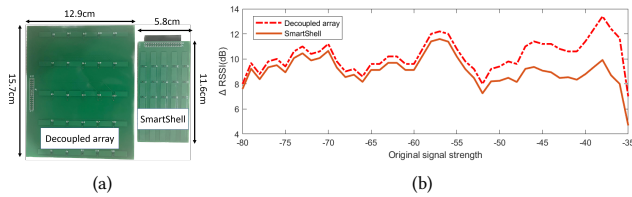
$$T = \frac{1}{NL_2 + t_{array}} \quad (22)$$

## 5 HARDWARE IMPLEMENTATION

We build a SmartShell prototype based on the MCU CC26x2R1 [33–35] and deploy it experimentally on a mobile device, as shown in Fig. 14. CC26x2R1 has a Bluetooth communication module inside the chip and provides 30 GPIO ports, of which we use 25 to control the array. The remaining pins perform the functions needed for data transfer, as well as device debug functions such as UART transfers. We design the antenna array on the printed circuit board by laying copper as described before, and chose sky13585-679LF[36] as the RF switch used to inter-connect the antenna array. Each RF switch only costs 0.6 dollars at a quantity of  $\leq 100$  pieces. The photo of the PCB antenna array is shown in Fig. 14.

On the smartphone side, we write an Android APP that collects RSSI data every 100ms to accommodate the algorithm as mentioned in Sec. 4. Realizing the algorithm on the smartphone can not only





**Figure 15: (a) Decoupled array and SmartShell array. (b) RSS enhancement performance comparison between the decoupled array and SmartShell in static scenarios.**

facilitate the collection of the RSSI data, but also alleviate the computing cost of the smartshell MCU and hence reduce its power consumption. The smartphone APP is developed with Android Studio and physically runs in Honor play5, MI CC 9e, and MI8 smartphone with 3 ~5% CPU utilization rate and 5.35MB RAM consumption. Smartshell poses no constraint over the brand or mode of Wi-Fi AP that serves the smartphone as long as it supports 5G band Wi-Fi. In our evaluations, we use an iPhone 13 as the Wi-Fi AP for convenience.

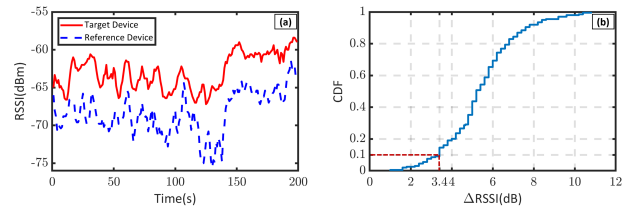
## 6 EVALUATION

### 6.1 Microbenchmarks

**Linearity.** We first evaluate the actual linearity of the antenna array of SmartShell. Recall in Sec. 2.1, our mathematical model is built with the assumption that our antenna array can be seen as a linear array. Now we verify the linearity of our actual array via comparing it to a reference array. The reference array is an individually designed array used only for this comparison. It is designed following the zero-coupling rule (spacing  $\lambda/2$ ). Our results will show that our array behaves identically with this reference array, which means the antenna array in SmartShell is approximately linear.

We conduct a static experimental comparison in a room with low wireless traffic and low device interference. Two arrays are used respectively to optimize the Wi-Fi signal strength of the target device (MI CC 9c) with the same initial RSSIs. To show that the increase in signal strength is not due to environmental changes, we also measure signal strength without the reflective array using a reference device (Honor Play5). The error between the two measurements was within 3 dB, allowing us to use the reference device data as a baseline for comparison. The experimental results are shown in Fig. 15(b). It can be seen that in the same static scenario, although SmartShell’s array slightly underperforms the decoupled array, but the maximum difference is within 2dB when initial RSSI is below -50dBm. Therefore, we can conclude that the SmartShell array performance is close to linear. When the original RSSI is high, the SmartShell’s internal coupling effect further degrades its performance. In contrast, the decoupled array’s internal coupling effect has less impact, allowing for a larger RSSI gain. However, the experiment shows that SmartShell also realizes signal strength improvement even with the presence of coupling at high RSSI.

We note that we do not simply deploy the reference array in SmartShell because the zero-coupling rule causes a large and unacceptable array size as shown in Fig. 15(a). The photos show the



**Figure 16: SmartShell real-time performance.**

reference array is significantly bigger than the SmartShell array under the same number of antenna elements; and therefore it is not applicable for mobile phones.

**Real-time capability.** Can SmartShell quickly respond to time-variability of the wireless channel? To test this, We conduct a 200-second experiment in the corridor with people going back and forth around. We measure the signal strength before and after optimization using the same method as previously mentioned. The target device is still a MI CC 9c smartphone and the reference device is an Honor Play5 smartphone. The experimental results are shown in left graph of Fig. 16. It illustrates that the enhanced signal strength of the target device keeps the pace with with the raw signal strength of the reference device; and their difference maintains no less than 3.4 dB at 90% as shown in the right graph of Fig. 16. It means that the SmartShell can respond very quickly to the fluctuations of dynamic wireless environment.

### 6.2 Effectiveness

We place the AP in fixed locations and conduct experimental deployment in different locations in the indoor environment. Fig. 17(a) shows the experiment field. During the experiments, we allow people to walk around and use other devices, adding to the complexity of the wireless channel. The data is measured  $\approx 500$  times at each location, during which the mobile devices are used to browse webs and listening to music using BLE headphone (AirPods 2), allowing the APP to run the algorithm in the background and communicate with SmartShell. Line-of-sight (LOS), non-line-of-sight (NLOS), and corridor scenarios are evaluated in this experiment. Fig. 17(c) and 17(d) depict the improvement of the RSS for different positions in line-of-sight and non-line-of-sight scenarios respectively. The experimental results show that in different indoor states, the SmartShell can play a significant role in improving the signal strength of mobile devices and the signal strength can be improved by an average of 5 ~ 10dB.

Besides, our results show that in the LOS scenarios, SmartShell can more effectively enhance the Wi-Fi RSSI when AP is more distant from the smartphone. This is due to the fact that an increase in the distance reduces the signal strength at the receiver side which bridges the difference between a multi-path signal ( $|h_i|$ ) and a direct-path signal ( $|H|$ ). As a result, a long distance from AP boosts the effect of signal strength enhancement.

### 6.3 Robustness

**Distance.** We place the AP at a fixed location in the corridor and optimize the signal strength at different distances from SmartShell (together with the smartphone) to the AP as shown in Fig. 18(a). The behavior of system performance with distance is shown in

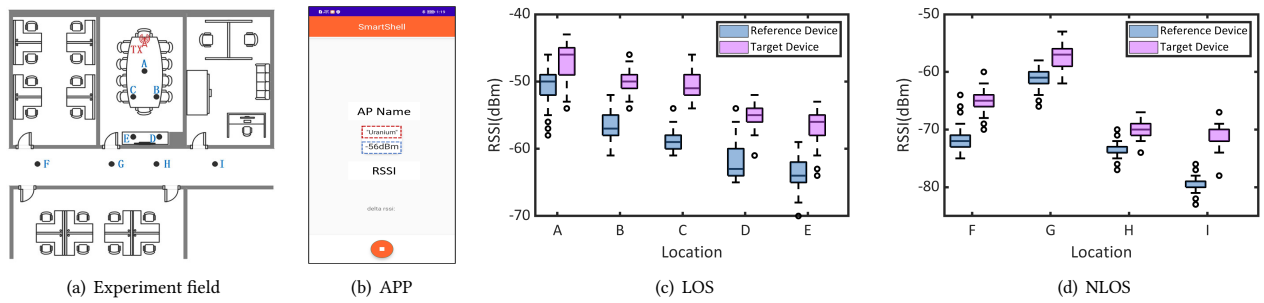


Figure 17: (a) SmartShell experiment field.(b) GUI of SmartShell APP on Android. (c) Performance in LOS scenarios. (d) Performance in NLOS scenarios.

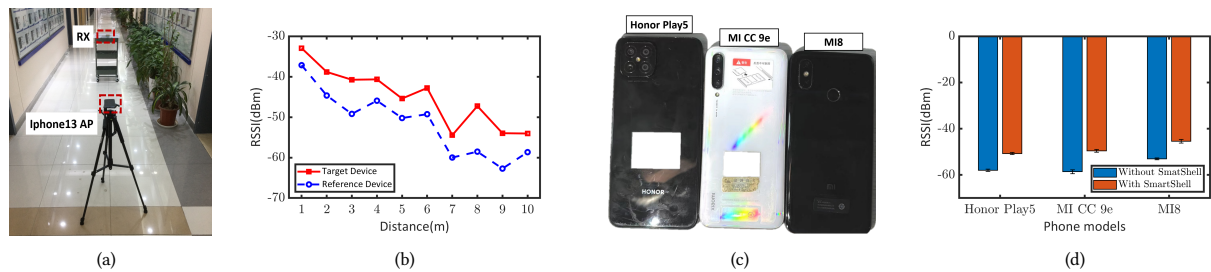


Figure 18: (a) Experiment field.(b) SmartShell performance versus distance to AP. (c) Various phone models. (d) RSSI improvements for various phone models.

Fig. 18(b). RSS overall declines with distance while multipath effect causes some Local fluctuations. When the SmartShell is close to the AP, the higher RSS results in less room for optimization and thus weaker improvements, but as the distance increases, the received signal decreases resulting in more room for optimization resulting in an average improvement of 5 ~ 10 dB. The experiments show that SmartShell is able to optimize the signal strength at different distances.

**Different mobile devices.** On three different smartphones: Honor Play5, MI CC 9e, and MI8, we evaluate the Wi-Fi signal enhancement caused by SmartShell. Photos of the three phone models are shown in Fig. 18(c). At a distance of 3.6 meters from the AP, we evaluate the average improvement in signal strength of three phone models with and without Smartshell. Fig. 18(d) illustrates the results of the experiment, demonstrating that Smartshell has a good improvement on various phone models. Fig. 18(d) further demonstrates that, when tested in the same location, the RSSI error between the Honor Play5 and MI CC 9e phones is smaller than 3dB.

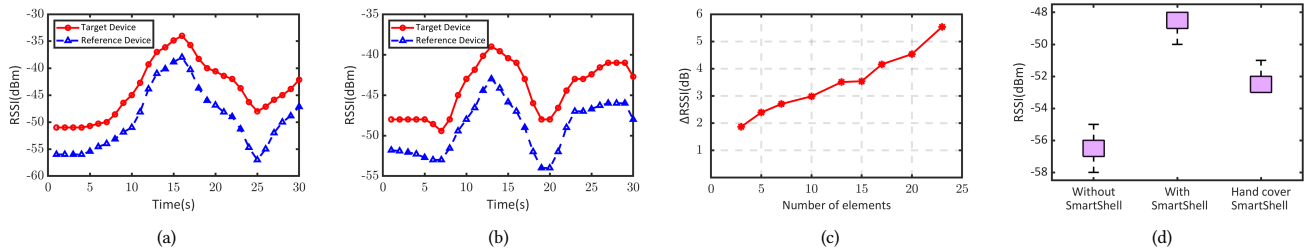
**Mobility.** In this section, we measure the signal enhancement performance of the array in a mobile scenario, where we place the AP in a fixed location and let the smartphone with SmartShell move away from AP. Movement results in a change in the raw RSSI without enhancement. In order to avoid confounding the effects of movement with the effects of array operation, we also set up the same reference device to measure the average signal strength before and after optimization. We measure the impact of SmartShell on the improvement of signal strength at two different

movement speeds: 0.9m/s and 1.2m/s. The experimental results are shown in Fig. 19(a) and Fig. 19(b). When the movement speed is low, SmartShell performs marginally better. Due to the slower moving speed, SmartShell has more time to respond to the channel changes. The experiments show that SmartShell is also able to achieve signal strength optimization in a mobile scenario.

**Interference resistance.** To test the SmartShell’s resistance to interference, we first explore the relationship between the number of enabled elements and SmartShell performance. The experimental results are shown in Fig. 19(c). It shows that the array can still be useful when the number of elements is small; and the performance of the array gradually improves as the number of elements increases. We then evaluate the impact on SmartShell performance when the back of the antenna array is covered by hand. This setup simulates the scenario where a user is holding his/her smartphone. The experimental results are shown in Fig. 19(d). It shows that when the back of the antenna array is covered by hand, the optimised performance drops by approximately 3dB. However, when SmartShell is deployed on a phone, it could be covered by hands, but this risk is significantly lower when it is deployed on a tablet or computer.

## 6.4 End-to-end Metrics

To connect the improvement in RSSI to the positive impact on the device transmission, we test SmartShell’s performance in throughput. At a distance of 5.6 meters from the AP, we evaluate the throughput of three phone models with and without Smartshell. We use the TCP (Transmission Control Protocol) to test throughput and



**Figure 19: (a) Walking speed of 0.9m/s. (b) Walking speed of 1.2m/s. (c) Average signal strength enhancement under different number of enabled elements. (d) Performance dropped by approximately 3dB when the back of the antenna array was covered by hand.**

configure the TCP window size for mobile phones to be 2 MBytes. The experimental results are shown in Fig. 20(a). It shows that the throughput is improved by an average of 20 to 30 Mbps when using SmartShell as compared to when not using it for the three phones.

## 6.5 Algorithm Comparison

We deploy different algorithms on the mobile app and conduct experiments in house. As we describe in Sec. 3.3, we use the results of the random sampling i.e., voting approach as an indicator of SmartShell performance. The experimental results as shown in Fig. 20(b) indicate that the results using QS, SA, and greedy methods are all higher than those of random sampling and achieve a signal strength improvement of approximately 8dB for  $N = 25$ . We design the QS algorithm to achieve faster optimization speed and optimization results. It is worth noting that the QS algorithms all enter the local optimum at less than 25 iterations. Therefore, in order to further reduce the latency of system operation in practical deployment, we can set up a decision mechanism to decide the number of iterations based on the actual performance of the algorithm, which can further reduce the latency with guaranteed performance.

## 6.6 Impact for Target Devices Transmitting

Does the SmartShell have any negative impact on the transmitted Wi-Fi signal from the target device? To test this, we evaluate the target device’s packet loss rate when transmitting UDP (User Datagram Protocol) packets at two different distances with (disabled and enabled) and without SmartShell. The UDP transmission rate is 64 kbps, and each packet contains 80 bytes. Fig. 20(c) illustrates that the packet loss rate is roughly the same no matter with or without SmartShell. The target device’s transmitting signal is almost not adversely affected by SmartShell.

## 6.7 Power Consumption

We measure the power consumption of SmartShell using OTII [37], which is a power consumption measurement instrument. We use one 3.7V, 500mAh rechargeable battery for the experiment. The power consumption of the system is shown in the Table 2.

Experiments show that the SmartShell takes about 15 minutes to charge, which is acceptable based on the consistency of SmartShell’s charging time with the mobile device. With a full charge, consider

Mode	Power consumption
Charging	9.8W
System work	13.6761mW
low power	2.67mW

**Table 2: Power consumption in different modes.**

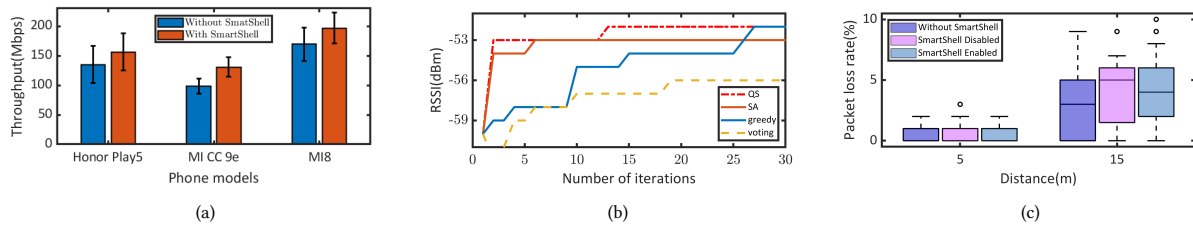
	$f_c$ (GHz)	$D$ (m)	$N$	$A$ (m <sup>2</sup> )	$G$ (dB)
RFcous	2.4	O(10)	3200	6	9.5
LAIA	2.4	O(10)	36	≈4	≈2~10
LLAMA	2.4	O(0.1)	180	0.23	≈10
<b>This work</b>	5.8	O(0.01)	25	0.0098	5~10

**Table 3: Comparison of intelligent surfaces.**

the most extreme case: having the SmartShell in a working state all the time. At this point the SmartShell is working for more than 120 hours, so it is able to ensure the normal use of the user.

## 6.8 Performance Comparison

We compare performance with related works, as shown in table 3. RFcous [11, 12] utilizes large arrays of 2D planes with 3200 elements to improve signal strength. RFcous performs optimization algorithms using RSSI parameters, which require the observed RSSI signal to be fed back from the receiver to the RFcous controller. However, reliable communication over long distances is a challenge. Considering the distance between the RFcous array and the receiving end, a large array is needed to strengthen the signal. As a result, the RFcous area is 6 square meters, leading to higher costs that make large-scale deployment impractical. LAIA [9, 10] enhances the RSS by using phase shifters. A single LAIA element has a significant impact on the channel because LAIA goes through the walls with wires, which enables LAIA to operate with a small number of elements. Although LAIA only has 1.44 times as many elements as SmartShell, its deployment area is almost 408 times larger. This is because LAIA uses phase shifters for each element, which results in a large area for a single element. By rotating the polarization of signals that reflect from the surface, LLAMA [38] lowers polarization mismatch to increase signal strength. Table 3 demonstrates that the SmartShell’s size and element number are substantially smaller than those of other related works, but because



**Figure 20: (a) Throughput for various phone models. (b) Iterative results of different algorithms. (c) Packet loss rate of target device with and without SmartShell.**

to its close proximity to the receiver, considerable improvements are still achieved. In addition, since one reflective surface cannot beam the signal toward multiple directions, the existing works cannot serve multiple users simultaneously. In comparison, SmartShell supports multi-user scenarios since the surfaces are deployed on end devices.

## 7 DISCUSSION

**2.4 GHz operation.** The SmartShell array is designed to enhance 5.8 GHz Wi-Fi instead of 2.4 GHz Wi-Fi for the following reasons: (1) The 2.4 GHz band has a larger wavelength, resulting in a  $2\times$  longer antenna length and a  $4\times$  larger array size to achieve the same signal strength optimization performance. Such a size is too large for small mobile devices like smartphones. (2) The latest Wi-Fi protocols, such as 802.11ac/ax, primarily utilize the high bandwidth of the 5 GHz band to achieve high throughput, making the 5 GHz the dominant band of Wi-Fi data traffic.

**Possibility of supporting other IoT protocols.** The design principles and optimization algorithm of SmartShell are agnostic to IoT protocols, meaning that they can be applied to a variety of protocols. To adapt the current SmartShell design to a different protocol, two steps must be taken: (1) Change the shape of the antenna element to  $\lambda/4 \times \lambda/10$  with the new  $\lambda$  value derived from the new carrier frequency. (2) Collect the RSSI from the radio chip to replace the current Wi-Fi RSSI. These steps will enable SmartShell to support other IoT protocols, providing enhanced signal strength.

**RSS gain achieved on different directions.** The RSS gain achieved on different directions may not be the same for the following reasons: (1) The signal's propagation path will be unevenly distributed around the environment, leading to the varying values of  $\alpha = \frac{|h_i|}{|H|}$  mentioned in Sec. 3.3. The RSS gain is proportional to the value of  $\alpha$  as shown in Fig. 8. (2) It is impossible to guarantee that the RSS gain is the same for each optimization because the QS optimization algorithm stops the optimization as soon as the RSS gain exceeds a certain threshold.

**Multi-user issue.** Large-scale reflective arrays struggle to handle multiple users, as they must focus the reflected signal towards a single user to achieve valid enhancement of the RSS. Focusing beams in multiple directions can weaken each beam, thereby reducing the overall quality of the reflected signal. Additionally, when using a time-division multiple access (TDMA) scheme to control the array, some users may not be able to benefit from the environment when needed most, and coordination among users is challenging.

In contrast, SmartShell avoids the multi-user issue naturally since the surfaces are deployed on end devices.

## 8 RELATED WORK

Recent research works apply metasurface to the new frontier of wireless communication, in order to program radio propagation environment. LALA uses a large-scale antenna array to make favorable wireless channel conditions for single and multi-antenna communication links [9, 10]. The reflecting elements are essentially phase shifters for generating in-phase multipath effects at the receiver, where accurately controlling those elements is non-trivial. RFocus is a 2D plane composed of micro RF switches to realize beamforming for increasing the RSS [11, 12]. The receiver needs to feed back the observed RSSI to the plane controller, which then adjusts the reflecting element's impedance through with those RF switches. RFocus presents a majority voting based optimization algorithm for controlling, which is proved to be near optimal. The LLAMA system utilizes the metasurface to adjust the polarization direction of the signal [38]. The above-mentioned reflecting surfaces incur high manufacturing or deployment cost, and all need the receiver to provide information to the centralized controller, where the feedback communication itself could be a challenge.

## 9 CONCLUSION

This paper presents SmartShell, a near-field array that can program the radio propagation environment around a mobile device. It is a small and low-cost reflecting surface attached to the device's shell, controlled by companion optimization software running on the mobile device for real-time Wi-Fi RSS enhancement. We implement a prototype of SmartShell and conduct experiments on a mobile device. Our experiments show that SmartShell can optimize the signal in a complex and variable channel environment, improving signal strength by an average of 5-10 dB.

## 10 ACKNOWLEDGEMENTS

The work in this paper is supported by the National Key Research and Development Program of China 2020YFB1708700, and National Natural Science Foundation of China (No. 92167205, 62272293, 61922055, 61872233).

## A ARTIFACT APPENDIX

The research artifact accompanying this paper is available via <https://doi.org/10.5281/zenodo.7908213>.

## REFERENCES

- [1] H.-T. Chen, A. J. Taylor, and N. Yu, "A review of metasurfaces: physics and applications," *Reports on Progress in Physics*, vol. 79, no. 7, p. 076401, jun 2016. [Online]. Available: <https://doi.org/10.1088/0034-4885/79/7/076401>
- [2] M. Chen, M. Kim, A. M. Wong, and G. V. Eleftheriades, "Huygens' metasurfaces from microwaves to optics: a review," *Nanophotonics*, vol. 7, no. 6, pp. 1207–1231, 2018. [Online]. Available: <https://doi.org/10.1515/nanoph-2017-0117>
- [3] Q. Wu, S. Zhang, B. Zheng, C. You, and R. Zhang, "Intelligent reflecting surface-aided wireless communications: A tutorial," *IEEE Transactions on Communications*, vol. 69, no. 5, pp. 3313–3351, 2021.
- [4] W. Chen, X. Ma, Z. Li, and N. Kuang, "Sum-rate maximization for intelligent reflecting surface based terahertz communication systems," in *2019 IEEE/CIC International Conference on Communications Workshops in China (ICCC Workshops)*, 2019, pp. 153–157.
- [5] X. Tan, Z. Sun, J. M. Jornet, and D. Pados, "Increasing indoor spectrum sharing capacity using smart reflect-array," in *2016 IEEE International Conference on Communications (ICC)*, 2016, pp. 1–6.
- [6] S. Lin, B. Zheng, G. C. Alexandropoulos, M. Wen, M. D. Renzo, and F. Chen, "Reconfigurable intelligent surfaces with reflection pattern modulation: Beamforming design and performance analysis," *IEEE Transactions on Wireless Communications*, vol. 20, no. 2, pp. 741–754, 2021.
- [7] M. Cui, G. Zhang, and R. Zhang, "Secure wireless communication via intelligent reflecting surface," *IEEE Wireless Communications Letters*, vol. 8, no. 5, pp. 1410–1414, 2019.
- [8] Q. Wu and R. Zhang, "Towards smart and reconfigurable environment: Intelligent reflecting surface aided wireless network," *IEEE Communications Magazine*, vol. 58, no. 1, pp. 106–112, 2019.
- [9] R. Chandra and K. Winstein, "Programmable radio environments for smart spaces-hotnets-xvi dialogue," in *ACM Workshop on Hot Topics in Networks, Palo Alto*, 2017.
- [10] Z. Li, Y. Xie, L. Shangguan, R. I. Zelaya, J. Gummeson, W. Hu, and K. Jamieson, "Towards programming the radio environment with large arrays of inexpensive antennas," in *16th USENIX Symposium on Networked Systems Design and Implementation (NSDI 19)*. Boston, MA: USENIX Association, Feb. 2019, pp. 285–300. [Online]. Available: <https://www.usenix.org/conference/nsdi19/presentation/lizhuqi>
- [11] V. Arun and H. Balakrishnan, "Rfocus: Beamforming using thousands of passive antennas," in *17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20)*. Santa Clara, CA: USENIX Association, Feb. 2020, pp. 1047–1061. [Online]. Available: <https://www.usenix.org/conference/nsdi20/presentation/arun>
- [12] V. Arun and H. Balakrishnan, "Rfocus: Practical beamforming for small devices," *arXiv preprint arXiv:1905.05130*, 2019.
- [13] J. Huang and J. A. Encinar, *Reflectarray antennas*. John Wiley & Sons, 2007, vol. 30.
- [14] K. Zheng, L. Zhao, J. Mei, B. Shao, W. Xiang, and L. Hanzo, "Survey of large-scale mimo systems," *IEEE Communications Surveys Tutorials*, vol. 17, no. 3, pp. 1738–1760, 2015.
- [15] J. Perry, P. A. Iannucci, K. E. Fleming, H. Balakrishnan, and D. Shah, "Spinal codes," *ACM SIGCOMM computer communication review*, vol. 42, no. 4, pp. 49–60, 2012.
- [16] Z. Rafique and B.-C. Seet, "Chapter 15 - energy-efficient mimo-ofdm systems," in *Handbook of Green Information and Communication Systems*, M. S. Obaidat, A. Anpalagan, and I. Woungang, Eds. Academic Press, 2013, pp. 393–422. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B9780124158443000152>
- [17] A. Lipson, S. G. Lipson, and H. Lipson, *Optical physics*. Cambridge University Press, 2010.
- [18] E. Björnson, Ö. T. Demir, and L. Sanguinetti, "A primer on near-field beamforming for arrays and reconfigurable intelligent surfaces," *arXiv preprint arXiv:2110.06661*, 2021.
- [19] Q. Wu and R. Zhang, "Intelligent reflecting surface enhanced wireless network via joint active and passive beamforming," *IEEE Transactions on Wireless Communications*, vol. 18, no. 11, pp. 5394–5409, 2019.
- [20] M. R. Garey and D. S. Johnson, *Computers and intractability*. freeman San Francisco, 1979, vol. 174.
- [21] X. Gan, H. Zhang, G. Hang, Z. Qin, and H. Jin, "Fast-charging station deployment considering elastic demand," *IEEE Transactions on Transportation Electrification*, vol. 6, no. 1, pp. 158–169, 2020.
- [22] S. G. Chen, "Efficiency improvement in explicit enumeration for integer programming problems," in *2013 IEEE International Conference on Industrial Engineering and Engineering Management*, 2013, pp. 98–100.
- [23] P. Merz and B. Freisleben, "Greedy and local search heuristics for unconstrained binary quadratic programming," *Journal of heuristics*, vol. 8, no. 2, pp. 197–213, 2002.
- [24] G. Kochenberger, J.-K. Hao, F. Glover, M. Lewis, Z. Lü, H. Wang, and Y. Wang, "The unconstrained binary quadratic programming problem: a survey," *Journal of combinatorial optimization*, vol. 28, no. 1, pp. 58–81, 2014.
- [25] S. Kim and M. Kojima, "Binary quadratic optimization problems that are difficult to solve by conic relaxations," *Discrete Optimization*, vol. 24, pp. 170–183, 2017.
- [26] *WiFiInfo*, [https://developer.android.com/reference/android/net/wifi/WifiInfo#getRssi\(\)](https://developer.android.com/reference/android/net/wifi/WifiInfo#getRssi()).
- [27] C. Bisdikian, "An overview of the bluetooth wireless technology," *IEEE Communications magazine*, vol. 39, no. 12, pp. 86–94, 2001.
- [28] G. Celosia and M. Cunche, "Fingerprinting bluetooth-low-energy devices based on the generic attribute profile," in *Proceedings of the 2nd International ACM Workshop on Security and Privacy for the Internet-of-Things*, 2019, pp. 24–31.
- [29] "Bluetooth network encapsulation protocol (bnep) specification," <https://www.bluetooth.com/specifications/specs/bluetooth-network-encapsulation-protocol-1-0>.
- [30] A. Saffari, M. Hessar, S. Naderiparizi, and J. R. Smith, "Battery-free wireless video streaming camera system," in *2019 IEEE International Conference on RFID (RFID)*, 2019, pp. 1–8.
- [31] "Highly-efficient, regulated dual-output, ambient energy manager for high-frequency rf input with optional primary battery," [https://e-peas.com/wp-content/uploads/2020/04/E-peas\\_RF\\_Energy\\_Harvesting\\_Datasheet\\_AEM40940.pdf](https://e-peas.com/wp-content/uploads/2020/04/E-peas_RF_Energy_Harvesting_Datasheet_AEM40940.pdf).
- [32] "bq25570 nano power boost charger and buck converter for energy harvester powered applications," [https://www.ti.com/lit/ds/symlink/bq25570.pdf?ts=1651751479251&ref\\_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FBQ25570](https://www.ti.com/lit/ds/symlink/bq25570.pdf?ts=1651751479251&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FBQ25570).
- [33] "Ce2652r simplelink multiprotocol 2.4 ghz wireless mcu," <https://www.ti.com/lit/ds/swrs207h/swrs207h.pdf?ts=1651728146185>.
- [34] "Low energy tree structure network," [https://www.ti.com/lit/an/swra648/swra648.pdf?ts=1651752866149&ref\\_url=https%253A%252F%252Fwww.ti.com%252Ftool%252FLAUNCHXL-CC26X2R1](https://www.ti.com/lit/an/swra648/swra648.pdf?ts=1651752866149&ref_url=https%253A%252F%252Fwww.ti.com%252Ftool%252FLAUNCHXL-CC26X2R1).
- [35] "Launchxl-cc26x2r1 design files (rev. b)," <https://www.ti.com.cn/cn/lit/zip/swrc346>.
- [36] "Sky13585-6791f: 1.0 to 6.0 ghz spdt switch," [https://www.skyworksinc.com/-/media/SkyWorks/Documents/Products/2401-2500/SKY13585\\_6791F\\_203458G.pdf](https://www.skyworksinc.com/-/media/SkyWorks/Documents/Products/2401-2500/SKY13585_6791F_203458G.pdf).
- [37] "Power analyzer, log sync & power supply in one product," <https://www.qoitech.com/>.
- [38] L. Chen, W. Hu, K. Jamieson, X. Chen, D. Fang, and J. Gummeson, "Pushing the physical limits of iot devices with programmable metasurfaces," in *18th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 21)*, 2021, pp. 425–438.