T.R.
ONDOKUZ MAYIS UNIVERSITY
INSTITUTE OF GRADUATE STUDIES
DEPARTMENT OF COMPUTER ENGINEERING

DEPLOYMENT OF SENSOR NETWORKS WITH IMMUNE PLASMA ALGORITHM BASED TECHNIQUES

Master's Thesis

Adnan TAŞDEMİR

Supervisor
Asst. Prof. Dr. Sercan DEMİRCİ

II. Supervisor
Assoc. Prof. Dr. Selçuk ASLAN

SAMSUN
2023

ACCEPTANCE AND APPROVAL OF THE THESIS

The study entitled **“DEPLOYMENT OF SENSOR NETWORKS WITH IMMUNE PLASMA ALGORITHM BASED TECHNIQUES"** prepared by **Adnan TAŞDEMİR** and supervised by **Asst. Prof. Dr. Sercan DEMİRCİ** and **Assoc. Prof. Dr. Selçuk ASLAN** was found successful and unanimously accepted by committee members as Master thesis, following the examination on the date 05/04/2023.

	Title Name SURNAME **University** **Department/Art**	**Final Decision**
Chairman	Assoc. Prof. Dr. Tansel DÖKEROĞLU Çankaya University Department of Sofware Engineering	⊠ Accept ☐ Reject
Member	Asst. Prof. Dr. Sercan DEMİRCİ Ondokuz Mayıs University Department of Computer Engineering	⊠ Accept ☐ Reject
Member	Asst. Prof. Dr. Gökhan KAYHAN Ondokuz Mayıs University Department of Computer Engineering	⊠ Accept ☐ Reject

This thesis has been approved by the committee members that already stated above and determined by the Institute Executive Board.

Prof. Dr. Ahmet TABAK
Head of Institute of Graduate Studies

DECLARATION OF COMPLIANCE WITH SCIENTIFIC ETHIC

I hereby declare and undertake that I complied with scientific ethics and academic rules in all stages of my M.S. thesis, that I have referred to each quotation that I used directly or indirectly in the study, and that the works I have used consist of those shown in the references, that it was written in accordance with the institute writing guide and that the stated cases in article 3, section 9 of the Regulation for TÜBİTAK Research and Publication Ethics Board were not violated.

Is an Ethics Committee Necessary?

☐ Yes

☒ No

02/05/2023
Adnan TAŞDEMİR

DECLARATION OF THE THESIS STUDY ORIGINALITY REPORT

Thesis Title: DEPLOYMENT OF SENSOR NETWORKS WITH IMMUNE PLASMA ALGORITHM BASED TECHNIQUES

As a result of the originality report taken by me from the plagiarism detection program on 02/05/2023 for the thesis titled above:

Similarity ratio : %20

Single source rate : %1 has been released.

02/05/2023
Asst. Prof. Dr. Sercan DEMİRCİ

ÖZET

İMMÜN PLAZMA ALGORİTMASI TEMELLİ YÖNTEMLER
İLE DUYARGA AĞLARININ YERLEŞİMİ

Adnan TAŞDEMİR
Ondokuz Mayıs Üniversitesi
Lisansüstü Eğitim Enstitüsü
Bilgisayar Mühendisliği Ana Bilim Dalı
Yüksek Lisans, Nisan/2023
Danışman: Dr. Öğr. Üyesi Sercan DEMİRCİ
II. Danışman: Doç. Dr. Selçuk ASLAN

Kablosuz sensör ağlarının (KSA'ler) sayısız uygulama için önemi, son yıllarda araştırmanın temel ve sıcak konularından biri olarak kabul edilmiştir. Sensör düğümlerinin, gücün, bant genişliğinin, belleğin ve diğer sınırlı kaynakların küçük boyutu ve ucuz üretim maliyeti nedeniyle, WSNs son on yılda bu alandaki araştırma faaliyetlerini ve endüstriyel yatırımları motive etti.

KSA'LER, yaşam tarzımızı önemli ölçüde değiştirme potansiyellerini gösteren çeşitli uygulama alanlarında başarıyla kullanılmıştır. Sensör düğümlerinin doğru şekilde konumlandırılması, kablosuz sensör ağları tasarımcıları için önemli bir zorluktur. KSA'lerdeki en büyük zorluklardan biri olarak, çıkarılan verilerin farklı konumlara işlenmesi ve iletilmesi için birçok düğümün etkin bir şekilde dağıtılması esastır. Bu nedenle, kablosuz sensör ağlarının çeşitli uygulama alanlarında en iyi performansı göstermesi için sensör düğümlerinin kapsama alanı en üst düzeye çıkarılmalıdır.

Verimli düğüm dağıtım algoritmalarının kullanılması, izleme bölgelerindeki sensör düğümlerinin kapsamını iyileştirebilir, ağın daha az enerji tüketmesine, ağın ömrünü en üst düzeye çıkarmasına ve sensörleri etkin ve verimli bir şekilde çalıştırmasına olanak tanır.

Bu çalışma, plazma veya antikorların bir hastadan diğerine transferine dayanan meta-sezgisel bir algoritma olan immün plazma algoritmasının (IP) iki versiyonunu kullanarak kapsama alanı sorununu ele almaktadır. Deneysel sonuçlar, önerilen algoritmanın KSA dağıtım sorununu iyi çözdüğünü göstermektedir.

Anahtar Sözcükler: Kapsama Problemi, Kablosuz Sensör Ağı, Simülasyon, Optimizasyon, İmmün Plazma Algoritması, Yönlendirme Protokolleri

ABSTRACT

DEPLOYMENT OF SENSOR NETWORKS WITH IMMUNE PLASMA ALGORITHM BASED TECHNIQUES

Adnan TAŞDEMİR
Ondokuz Mayıs University
Institute of Graduate Studies
Department of Computer Engineering
Master, April/2023
Supervisor: Asst. Prof. Dr. Sercan DEMİRCİ
II. Supervisor: Assoc. Prof. Dr. Selçuk ASLAN

The importance of wireless sensor networks (WSNs) for numerous applications has been considered one of the fundamental and hot topics of research in recent years. Due to the small size and cheap manufacturing cost of sensor nodes, power, bandwidth, memory, and other limited resources, WSNs motivated research activities and industrial investments in this field in the last decade.

WSNs have successfully been used in a variety of application fields, demonstrating their potential to significantly alter our way of life. Correctly locating the sensor nodes is a significant difficulty for designers of wireless sensor networks. As one of the major challenges in WSNs, effectively deploying many nodes is essential to processing and transmitting the extracted data to different locations. Thus the coverage area of sensor nodes must be maximized for wireless sensor networks to perform optimally in various areas of applications.

Using efficient node deployment algorithms can improve the coverage of sensor nodes in monitoring regions, allows the network to consume less energy, maximize the life of the network, as well as effectively and efficiently operate the sensors.

This study addresses the coverage area problem by using two versions of the immune plasma algorithm (IP), A meta-heuristic algorithm based on the transfer of plasma or antibodies from one patient to another. Experimental results show that the proposed algorithm solves the WSN deployment problem well.

Keywords: Coverage Problem, Wireless Sensor Network, Simulation, Optimization, Immune Plasma Algorithm, Routing Protocols

ACKNOWLEDGEMENT

In the beginning, I must first thank God Almighty who enabled me to reach this high scientific stage and paved the way for me to be among you today to discuss my master's thesis. I also extend my thanks and gratitude to my generous mother, my dear father, and my family to support me in reaching what I have achieved. I would like to thank each of Asst. Prof. Dr. Sercan DEMİRCİ, and Assoc. Prof. Dr. Selçuk ASLAN for their supervision and for giving them a lot of time firsthand in the publication of this scientific thesis in the form in which it appeared also their guidance and advice played an essential role in completing my academic studies. Thanks to all my friends who also supported me to finish this work. Thanks to everyone who helped me directly or indirectly to complete this thesis.

Adnan TAŞDEMİR

CONTENTS

SYMBOLS AND ABBREVIATIONS

2D	:	Two-dimensional
ABC	:	Artificial Bee Colony Algorithm
AMO	:	Animal Migration Optimizasyon
AODV	:	Ad-hoc On-demand Distance Vector
CR	:	Coverage Ratio
DSDV	:	Destination-sequenced Distance Vector
DSR	:	Dynamic Source Routing
FFA	:	Firefly Algorithm
IP	:	Immune Plasma Algorithm
MOICA	:	Multi-objective Imperialist competitive Algorithm
MSE	:	Mean Square Error
NAM	:	Network Animator
NoD	:	Number of Donors
NoE	:	Number of Evaluations
NoR	:	Number of Receivers
NoS	:	Number of Sensors
PDR	:	Packet Delivery Ratio
PS	:	Population Size
PSO	:	Particle Swarm Algorithm
R	:	Sensor Sensing Range
RREP	:	Route Reply
RREQ	:	Route Request
RRESP	:	Route Response Request
SEQ NO	:	Sequence Number
VFC	:	Virtual Force-directed Coevolutionary
WSN	:	Wireless Sensor Network
mIP	:	Modified Immune Plasma Algorithm

FIGURES LEGENDS

TABLES LEGENDS

1. INTRODUCTION

In many fields, wireless sensor networks (WSNs) are envisioned as promising futures due to the advancement of sensor technologies. A WSN is a network of nodes capable of detecting the environment and transmitting the collected data from the monitored area via wireless connections. It can be described as a system of nodes that can provide, receive and control the connections between people or computers with the environment (Enayatifar et al., 2014). In recent years, WSNs have attracted significant attention from the community and industry research field, providing various new directions for applications (Bhat & KV, 2022). The reason for the recent interest in WSNs is their potential applications in a wide range of contexts, including soldierly operations, environmental monitoring, observation systems, healthcare, general safety, disaster intervention, status identification for production equipment, disaster crisis search-and-rescue, and medical care, etc. (Aljubori et al., 2022) (L. Wang et al., 2018).

The most important factors that directly affect the power consumption of the WSNs are the number of nodes, their size and bandwidth, their storage and computing capabilities, the power and coverage required by detection, and storage and computing units (Njoya et al., 2020) (J. Wu et al., 2022) (Li et al., 2022) (Huynh Thi Thanh et al., n.d.). Additionally the coverage problem becomes more difficult to solve, especially in the WSNs where the sensors are able to cover just a particular area, causing additional power consumption caused by the power needed by the coverage sensors to stay active (Al-Mousawi, 2020).

In node deployment, the main requirement is enough coverage and connectivity properties. In general, coverage issues can be divided into three categories: (a) area coverage, which entails covering an entire area; (b) point coverage, which entails covering a collection of stationary or moving points; and (c) barrier coverage, which entails reducing the likelihood that undetected penetration is possible through a barrier (a sensor network) (Enayatifar et al., 2014) (Khoufi et al., 2017).

Area coverage ensures that at least one node and connectivity monitor each detection area, ensuring that nodes can communicate among themselves. It is obvious that the characteristics of WSN that should be optimized directly correlate with positional data or wireless sensor installations. A deterministic or random deployment technique is used to place sensor nodes in the areas of interest to create a WSN. A random deployment can be utilized for the initial settling of a network with sensor nodes if there is insufficient time for a thorough surface or environmental study or if the region being monitored cannot be explored due to geographic constraints. However, following initial deployment, the placements of the sensor nodes should be modified while taking the network's goals and the optimization targets into account (Aslan et al., 2018). It is suboptimal to deploy sensors randomly since it results in uneven coverage and requires more sensors than planned. In addition, coverage holes and disconnected networks can exist in this deployment type (W. Liu et al., 2018) (Huang et al., 2022). This reduces the reliability of the networks because some critical events may get unnoticed. In other words, it is simple to create WSNs at random, but this sensor distribution at random may result in a network with low energy efficiency and sensor coverage. Therefore, the minimum number of nodes must be used to achieve maximum network coverage (L. Wang et al., 2018).

Figure 1.1 shows the difference between deploying sensors randomly and optimally. The drone chooses where to place sensors in (a). Random deployment leaves gaps in some areas, as shown in (c), where some places have numerous sensors and others have none at all. For the coverage area to be as big as possible and for the sensors to work better in the network, they need to be in a place where they can be used to their fullest. As shown in (b), after optimization, only a small number of sensors are spread across the whole area, allowing for the most coverage.

It also seems sensible that this deployment would affect how well or poorly protocols can be routed. When designing a deployment for a WSN, it is important to consider the shortest route between sender and receiver, the number of packets delivered and received, as well as the congestion and energy used by each node of the WSN. That is the end goal of simulation software. They provide a real-world scenario using this deployment and provide details on the routing protocol's performance (Kuriakose et al., 2014).

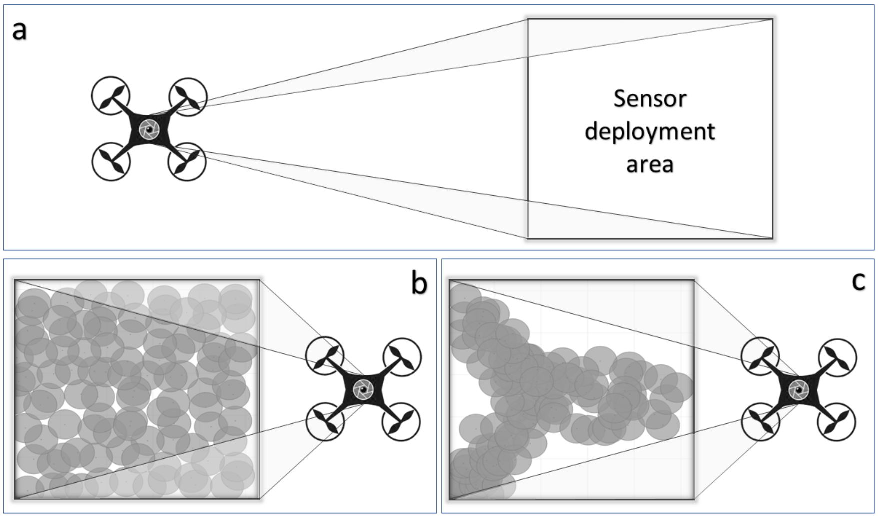


Figure 1.1. Deployed sensors randomly and optimally

1.1. Research Objectives

The aim of this thesis is to completely address the issue of sensor deployment and coverage in WSNs. The IP algorithm is suggested as a solution to the coverage area issue, and its performance is evaluated against that of other algorithms using a variety of metrics across a number of routing methods. The findings demonstrate that the IP and modified IP (mIP) algorithms have the capability and effectiveness to enhance the deployment of sensors to expand the coverage of WSNs and they can challenge other algorithms in improving coverage while having no negative impact on wireless sensor network simulation performance. The IP algorithm's performance, nevertheless, was sometimes on par with or even superior to that of other algorithms.

1.2. Thesis Structure

This thesis consists of five chapters. The second chapter "Literature Review", which follows the introductory chapter, starts with a broad description of the coverage issue of WSNs. Additionally, it provides a clear and accurate description of the popular techniques for determining WSNs coverage. The specification of the simulation software used to simulate the sensor distribution is then given, along with an explanation of the different kinds of routing protocols that were employed and the performance metrics used to compare those protocols. At last, the related works are given.

The third chapter "Materials and Methods" discusses the origins of the IP algorithm and the notion of plasma therapy. Followed by an outline of the key phases in both the IP and mIP algorithms. Additionally, a short description of the other algorithms—animal migration optimization (AMO), firefly algorithm (FFA), and artificial bee colony (ABC)—as well as their parameters utilized for comparison.

In the fourth chapter, the IP algorithm's parameter settings are explained. Where a variety of scenarios are created to produce various outcomes in order to compare them with those of the other algorithms mentioned above. Additionally, in this chapter, the simulation program's settings are also reviewed, and graphs are used to compare the results in many metrics.

The last chapter examines and rates the performance of the IP and mIP in the aforementioned tests. It provides a summary of this study's findings and makes suggestions for how to enhance the used methodologies in further studies.

2. LITERATURE REVIEW

Before discussing the primary responsibilities of this thesis, it is necessary to provide a quick review of the coverage issue and its solutions models, routing protocols, and simulation tools.

2.1. Definition of Sensor Coverage Problem

Suppose A is a two-dimensional (2D) field divided into equally sized grids, S number of sensors are located in A, and P is the set of grids in A, as shown in Figure 2.1. The distance between S_i and P_j refers to the euclidean distance as given in Eq. 2.1 (Ben Amor et al., 2022) (Saheb et al., n.d.) (Z. Wang et al., 2019).

$$(x_1 - x_2)^2 - (y_1 - y_2)^2 = \gamma^2 \tag{2.1}$$

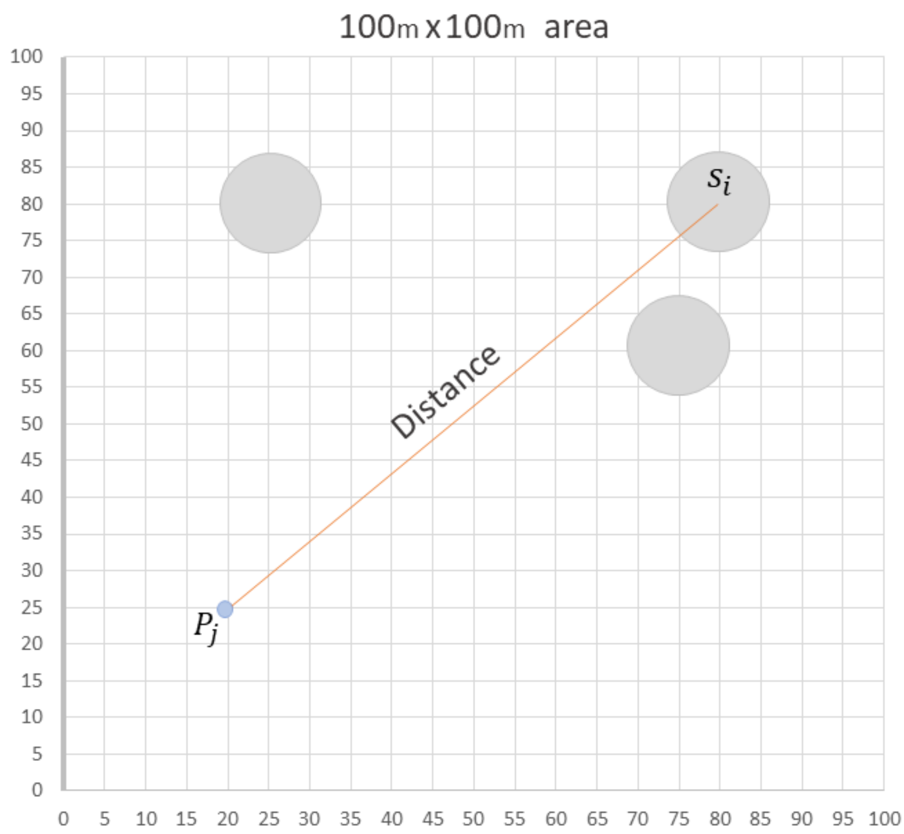


Figure 2.1. Euclidean distance between S_i and P_j

The two most significant approaches for calculating sensor coverage in WSNs are coverage area by a probabilistic model and coverage area by a binary detection model. The primary difference between (b) probabilistic and (a) binary models are seen in Figure 2.2.

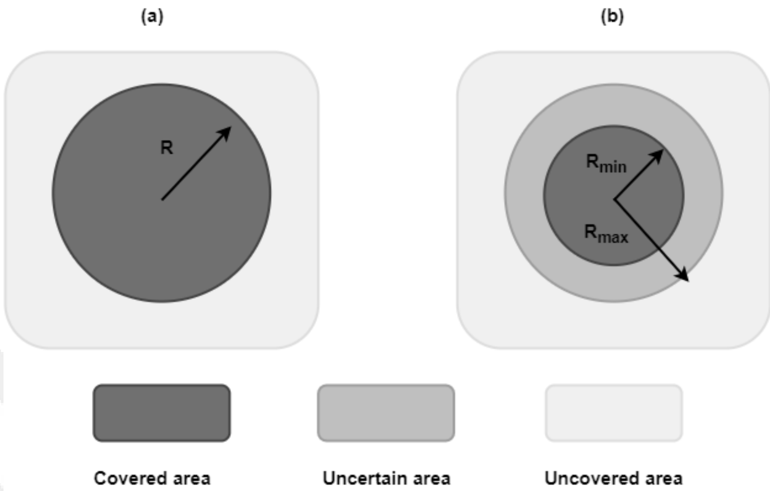


Figure 2.2. Node sensing range

2.1.1. Coverage Area by Binary Detection Model

Finding reference grids in the needed region and evaluating whether they are covered is one of the most used methods for assessing a sensor node's coverage. Each corner grid may act as a benchmark for calculating the coverage ratio with the area divided into equally sized grid sub-areas. In this model, any grid in A is covered by a sensor node if it is in that sensor node's sensing range (Amutha et al., 2020) (J. Liu & Sun, 2022). Otherwise, it is claimed that the grid has a coverage hole (Juneja et al., 2022). The pseudo-code of this model is given in Alg. 1. In this model, the coverage objective function is given in Eq. 2.2:

$$C(S,P) = \begin{cases} 1, & \text{if } d(S,P) \leqslant R \\ 0, & \text{otherwise} \end{cases} \tag{2.2}$$

Where $d(S,P)$ is the Euclidean distance between sensor node S and point P, and R is the node's sensing area. The deployment's coverage ratio (CR) is calculated as shown in Eq. 2.3. Where C_i relates to the coverage of the i^{th} sensor.

$$CR = \frac{U_{c_i}}{A}, i \in S \tag{2.3}$$

Algorithm 1 Binary Coverage Ratio

```
Let X, Y, NoS = 100
Let NoCF, isC, x, y, n, dis = 0
Let R = 7
Let  N = [[x_1, y_1], [x_2, y_2], ..., [x_i, y_i]]
for x = 0 to X do
    for y = 0 to Y do
        isC = 0
        for n = 0 to NoS do
            dis = distance(N[n], [x, y])
            if isC = 1 then
                continue
            else
                if dis < R then
                    NoCF++
                    isC = 1
                end if
            end if
        end for
    end for
end for
return NoCF/X*Y
```

2.1.2. Coverage Area by Probabilistic Model

The authors in (Zou & Chakrabarty, 2003) presented the probabilistic sensing model as a more realistic extension of the boolean sensing model. One of the most widely used approaches for calculating a sensor node's coverage is finding reference grids in the required area and determining whether they are covered (Si et al., 2020). As a result, $C(S, P)$, the objective function, must be written in probabilistic terms. The pseudo-code of this model is given in Alg. 2. This model defines two sensing radii, R_{min} and R_{max}, with R_{min} defining the start of the sensor detection uncertainty. If a point P in the network field is within sensor node S_i sensing range R_{min}, it is believed that point P is certainly detected by S. Point P is not detected by S if it is positioned outside of the sensing range R_{max}. Otherwise, S detects point P with probability P. In a probabilistic sensor detection model, given that $R_{\min} = (r - r_e)$ and $R_{\max} = (r + r_e)$, the coverage of the sensor S_i for point P may be determined, as shown in Eq. 2.4 (Aslan et al., 2018).

$$c_i = \begin{cases} 0, & d\left(S_i, P\right) \geqslant b \\ e^{-((\lambda_1 \alpha_1 \beta_1)/(\alpha_2 \beta_2) + \lambda_2)}, & a < d\left(S_i, P\right) < b \\ 1, & d\left(S_i, P\right) \leqslant a \end{cases} \tag{2.4}$$

Algorithm 2 Probabilistic Coverage Ratio

$Let\ X, Y, NoS = 100$
$Let\ TC, CP, x, y, n, dis = 0$
$Let\ R = 7$
$Let\ OP = 1$
$Let\ R_{min} = 3.5$ and $R_{max} = 10.5$
$Let \quad N = [[x_1, y_1], [x_2, y_2], ..., [x_i, y_i]]$
for $x = 0$ to X **do**
    for $x = 0$ to Y **do**
        $OP = 1$
        for $n = 0$ to NoS **do**
            $dis = distance(N[n], [x, y])$
            if $dis < R_{min}$ **then**
                $CP = 1$
            else if $dis >= R_{max}$ **then**
                $CP = 0$
            else
                $CP = e^{-(R_{min}+dist)/\sqrt{R_{max}-dist}}$
            end if
            $OP* = 1 - CP$
        end for
        $TC+ = 1 - OP$
    end for
end for
return $TC/X*Y$

where β_1, β_2, and λ_1 are measurement conditions parameters, re is the detection uncertainty of the detection range constant, and λ_2 is the distribution factor. Finally, the α_1 and α_2 factors are calculated as $re - d(S_i, P)$ and $re + d(S_i, P)$ respectively (Ozturk et al., 2011).

The point coverage value in the detection regions of numerous sensor nodes may be greater than the reference points covered by a single sensor node in the probabilistic sensor detection model. Taking into consideration any overlays, the average coverage value of point P may be determined as shown in Eq. 2.5 (Ozturk et al., 2011).

$$C_p = 1 - \prod_{s_1 \in S} \left(1 - C_p\left(s_i\right)\right) \tag{2.5}$$

2.2. Routing Protocols

The routing protocols specify how nodes will talk to one another and how data will go across the network. The routing protocols used by WSNs may be categorized in a variety of ways. Figure 2.3 depicts the fundamental categorization of routing protocols (Mahdi et al., 2021).

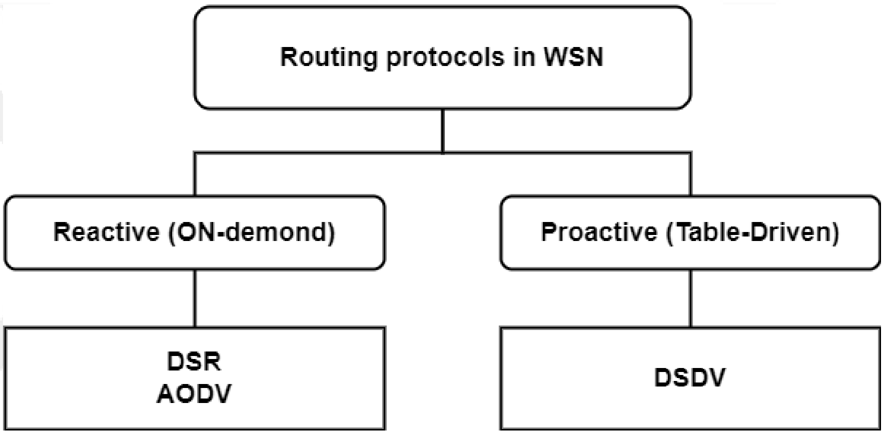


Figure 2.3. Fundamental categorization of routing protocols

2.2.1. Reactive (on-demand) Routing Protocol

Reactive routing techniques require that a node only start a route discovery when it needs to transmit a packet to a certain location. They don't regularly update or maintain their route databases with the most recent route topology. Because of the on-demand route establishment procedure, the communication overhead is therefore decreased but the latency is raised (Younis et al., 2021). The reactive routing protocols used in this study are the ad-hoc on-demand distance vector routing system (AODV) and the dynamic source routing (DSR).

- *AODV*: The AODV is used to get around the shortcomings of the DSDV and the DSR. Sequence number (SEQ NO) and broadcast ID are two counters that are kept along with each node's routing tables in AODV. The IP address at which data will be sent from the source is already known. Thus, the destination Sequence Number (SEQ NO) helps to determine an updated path from source to destination. Route request (RREQ) and path response (RRESP) packets are used in conjunction with these counters, with RREQ overseeing figuring out the route from source to destination and RRESP returning the route information response to its source (Perkins et al., 2003). Each node includes a routing table in addition to the Route Request (RREQ) packet in the format shown in Figure 2.4.

Destination IP	Destination	Sequence Number	Source IP	Source Sequence Number	Hop Count

Figure 2.4. Format of RREQ

As shown in Figure 2.5, consider a network with five nodes, S, A, B, C, and D, all of which are situated a unit distance apart from one another. Where S serves as the network's source node and D serves as the network's destination node. Also, the source node S's and destination node D's IP addresses are previously known.

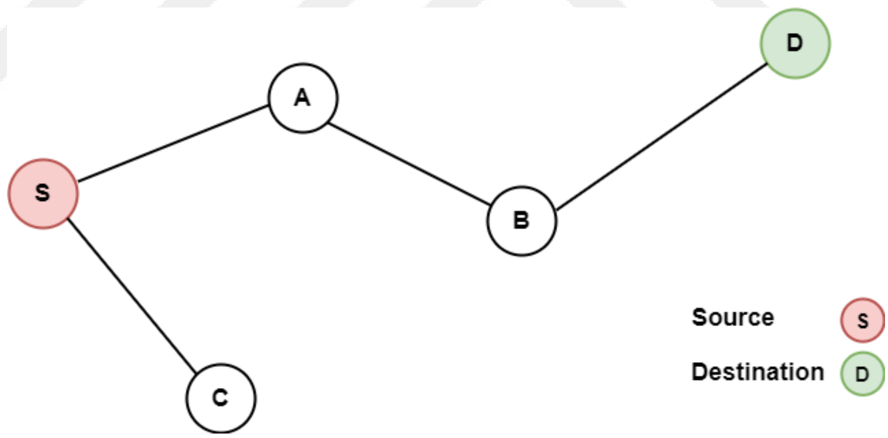


Figure 2.5. AODV network example

In the first stage, Source node S will send Route Request i.e., RREQ packet to its neighbors A and C. Nodes A and C will check for a route in the second stage and will reply by sending an RRESP packet back to source S. C is the final node in this instance, but it is not the destination so it will notify S in the RREQ packet that the route was not found. However, node A will broadcast the RRESP to node B and transmit the RRESP packet with the message "Route Found". In the third stage, the net hop field in the RREQ format will now be changed, and node B will send node

A the message "Route Found" in return while also updating the next hop field. The fourth phase involves node B broadcasting the RREQ packet to destination node D and further updating the next hop field. The RRES packet will then be delivered to B, where it will be forwarded to source node S through nodes B and A to create an ideal route. The network would be modified as shown in Figure 2.6.

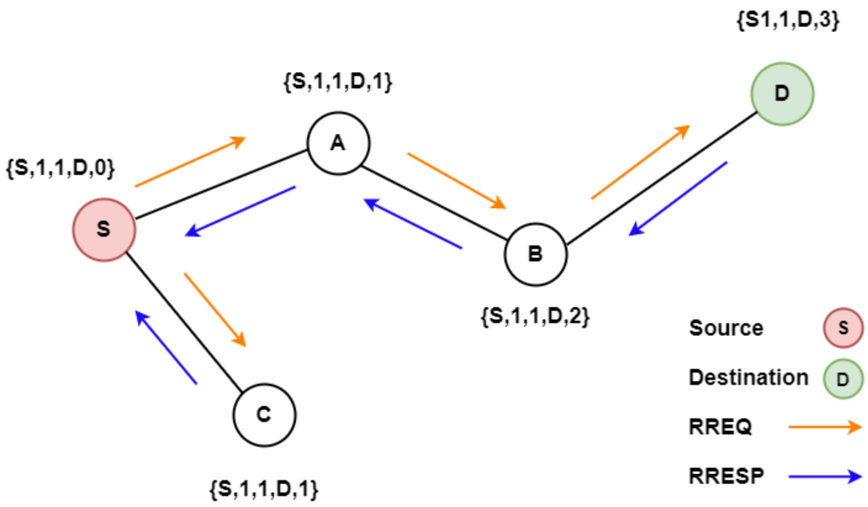


Figure 2.6. Last stage of AODV

- *DSR*: Wireless sensor networks that may be reactive or on-demand employ the DSR routing protocol. It employs source routing rather than routing tables, as its name implies. Route discovery and route maintenance are the two components of DSR routing. Route request and route reply (RREP) messages make up the route discovery phase, which the source node will start. Unlike AODV, where every intermediate node would respond with a route reply message RREP, in DSR only the destination node will respond with a route reply RREP message to the source node. The following stage of route maintenance is utilized to decrease the distance between the source and destination nodes and prevent RREP message flooding (Abdullah & Aziz, 2014). As shown in Figure 2.7, Consider a network of 10 nodes, with node S acting as the source and node D as the destination node.

  The DSR begins broadcasting information from source node S to neighbor nodes in the first stage. As a result of the direct connection between nodes S and A, the route information in this example is [S]. In the second stage, transmitting prior route information to A's neighbor nodes i.e., B, C, and E. In every situation, the new route will stay the same as [S, A]. The preceding route [S, A] is broadcast by node B to the

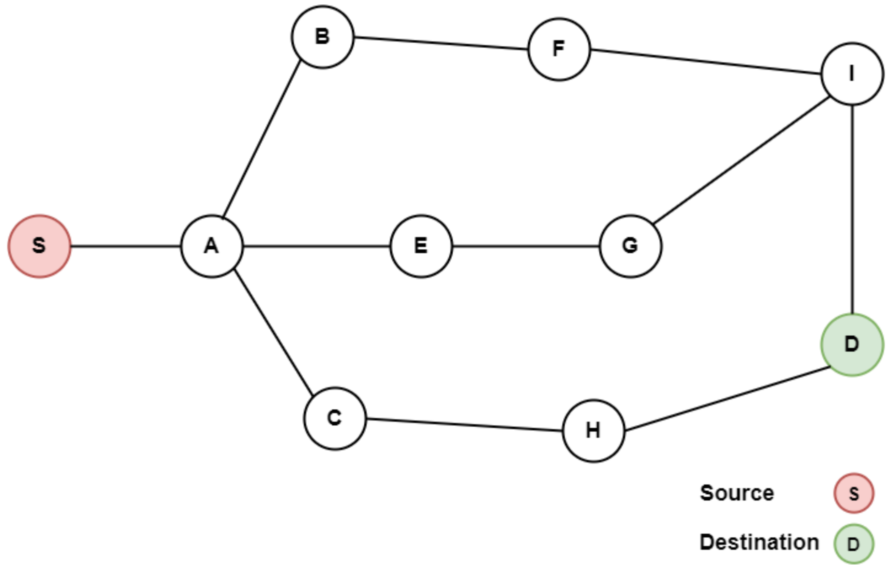


Figure 2.7. DSR network example

next neighboring nodes i.e., to node F, at the third stage. [S, A, B] will be the new route up to node F, and the same procedure may be used for other nodes i.e., E and C. In the fourth stage, Further, broadcasting the new routes i.e., [S, A, B, F], [S, A, E], and [S, A, C] to nodes G, H, and I respectively. In the fourth stage, new routes [S, A, B, F], [S, A, E], and [S, A, C] are broadcast to nodes G, H, and I, respectively. The previous stages will be repeated in the final stage until all possible pathways lead to the target node. The routes will be modified as shown in Figure 2.8.

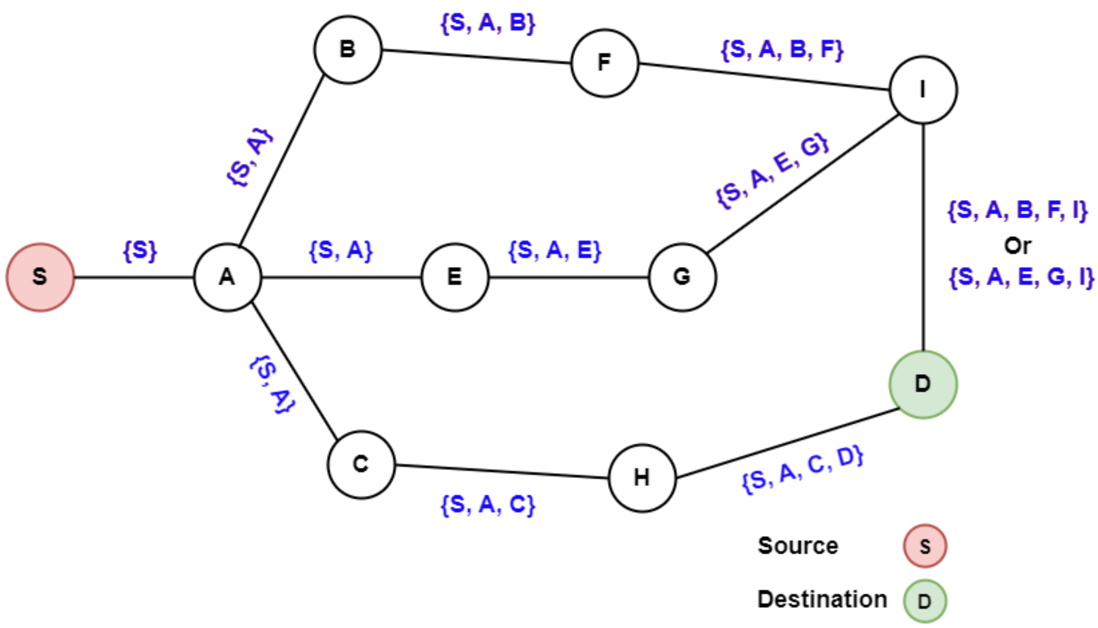


Figure 2.8. Modified routes in DSR

Following that, the "Re-Request" packet will be forwarded from destination node D to source node S in reverse. Finally, it will calculate the shortest path by counting the nodes along the path that was found in the earlier phases. Since [S, A, C, H] has the fewest nodes and is thus certain to be the shortest route as shown in Figure 2.9, it will be selected so that data may be transported appropriately.

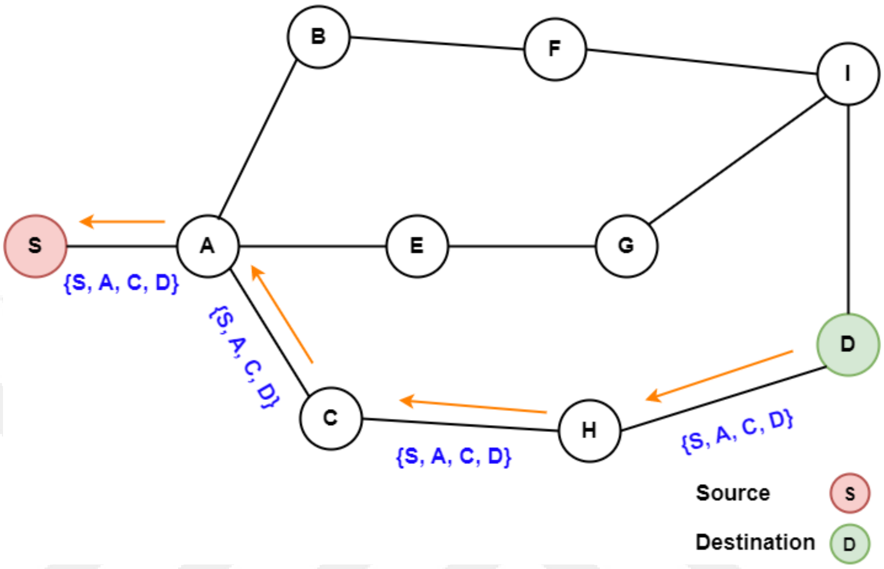


Figure 2.9. Last stage of DSR

2.2.2. Proactive (table-driven) Routing Protocols

Because they maintain the routing tables for the whole network by transmitting network information from node to node, and because the routes are pre-defined before usage and even when there is no traffic flow, they are sometimes referred to as table-driven routing protocols (Soomro et al., 2022). The proactive routing protocol used in this study is Destination-sequenced distance vector routing (DSDV).

- *DSDV*: Based on the concept of the traditional Bellman-Ford Routing algorithm, the DSDV is a hop-by-hop distance vector routing protocol that mandates each node to broadcast periodic routing updates. Each node maintains a routing table listing the next hop for each reachable destination, number of hops to reach the destination, and the sequence number assigned by destination node. The sequence number is used to differentiate between old and new routes and prevent loop formation. The stations provide their close neighbors with their routing tables on a regular basis. If there has been a major change since the previous update transmitted, a station additionally sends its routing table. The update is thus triggered by events and by time. One of two methods—a full dump or an incremental update—can be used to provide routing table changes (Mohapatra & Kanungo, 2012). As above, the DSDV protocol uses a single table for every node individually. The DSDV table contains attributes as shown in Figure 2.10. These tables updating time is indicated by SEQ No.

Node	Destination	Next Hop	Distance	SEQ No

Figure 2.10. Included attributes in the DSDV table.

As shown in Figure 2.11, consider a network of three nodes having distances of 1 on each of the edges respectively.

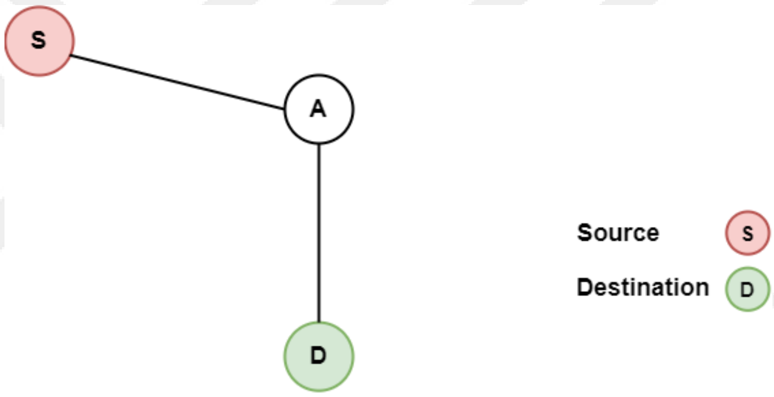


Figure 2.11. DSDV network example

Tables for all the nodes S, A, and D along with the distance and sequence number are determined. Node A may disseminate the routing table if desired. Then, all the network's revised routing tables will appear as shown in the tables below.

Table 2.1. Routing table of node S in DSDV

Destination	Next Hop	Cost
S	S	0
A	A	1
D	A	2

Table 2.2. Routing table of node A in DSDV

Destination	Next Hop	Cost
S	S	1
A	A	0
D	D	1

Table 2.3. Routing table of node D in DSDV

Destination	Next Hop	Cost
S	A	2
A	A	1
D	D	0

2.3. Performance Metrics of Routing Protocol

The consumed energy, throughput, and packet delivery ratio (PDR) are the three key performance indicators used to compare the three protocols' performances in this thesis. These three metrics are explained below.

2.3.1. Packet Delivery Ratio (PDR)

PDR is the percentage of data packets that are successfully delivered to the destination to those that are sent by the source. The PDR formula is as in Eq. 2.6 (Khan et al., 2013) :

$$PDR = \frac{\text{Total no. of packets received}}{\text{Total no. of packets sente}} \tag{2.6}$$

2.3.2. Throughput

The amount of data that may be transferred across a network over time is known as throughput. This data or information may be sent across a logical or physical connection or from a particular network node. Eq. 2.7 shall be utilized for throughput calculation. Throughput is usually measured in bits per second (bit/s or bps), and sometimes in data packets per second (p/s or pps) or data packets per time slot.

$$Throughput = \frac{\mu}{\text{t}} \tag{2.7}$$

Where μ is the number of bytes received per unit of time (t). It can be the average throughput of all nodes for the whole network.

2.3.3. Energy Consumption

Energy consumption is equal to the total energy used by the entire network by considering each node's energy consumption during packet transmission (Barati et al., 2012).

2.4. Simulation Tools

Systems research needs performance evaluation because it enables the assessment of novel concepts, the detection of issues and bottlenecks, and the optimization of current systems. There are three general methods of performance assessment which are prototyping, analytical modeling, and simulation. Particularly for large-scale systems, prototyping is often impractical or time-consuming, and it also offers only a limited level of control and observability. Analytical modeling is also unable to represent very complex systems. As a result, simulation has become a popular alternative that is widely utilized in assessing the performance of computer systems (Siraj et al., 2012). Many simulation tools, including NS-2, TOSSIM, EmStar, OMNeT++, J-Sim, ATEMU, and Avrora, are employed in WSNs. In this thesis, networks will be simulated using Ns-2.

2.4.1. Ns-2 Simulation Tool

The open-source discrete event simulation software Ns-2 (Network Simulator-2) was created in 1989 using C++ and OTcl. OTcl is A high-level defining language used for developing scripts is OTcl. The OTcl codes operate by invoking low-level C++ modules. Local area networks, wide area networks, and personal area networks may all be simulated using Ns-2. Figure 2.12 displays the Ns-2 package's contents.

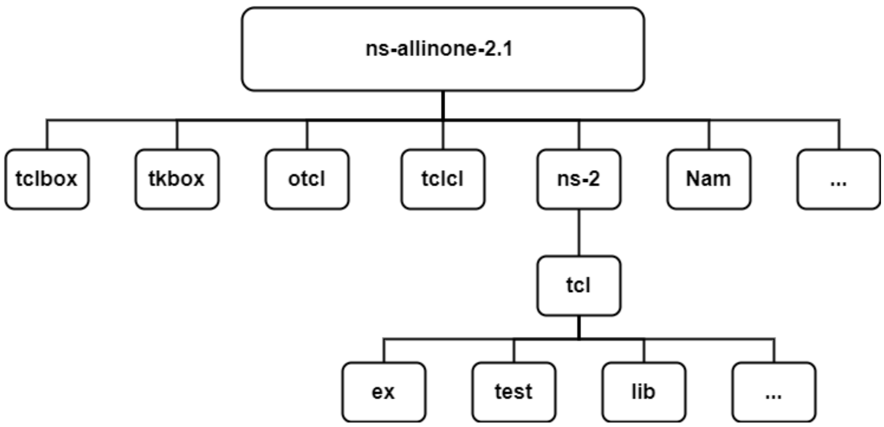


Figure 2.12. Package's contents of the Ns-2

Ns-2 provides two distinct outputs. The "trace files" history file with the ".tr" extension is the first output. This file contains information that explains when a sensor got the information, when it delivered information, when it lost information, and when it accumulated information. The second output is a file called a Network Animator (NAM) file that lets us view the motions of the sensors and package data graphically (Estrin et al., 2000).

2.4.2. NAM Animator

NAM is an animation-based program for monitoring network simulation traces and real-world data packets, it supports topology layout, packet-level animation, and various data exploration tools. NAM reads the trace file, creates the topology, a screen pops up, and shows the simulation, and provides multiple views of the animation as shwon in Figure 2.13 (Issariyakul & Hossain, 2009).

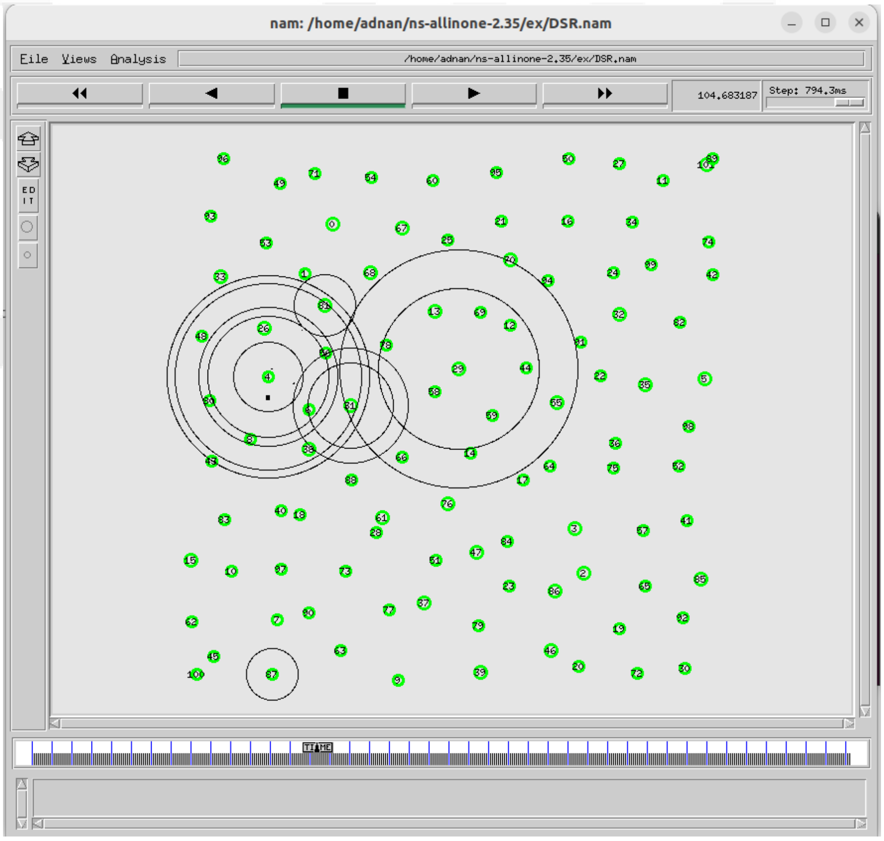


Figure 2.13. The output of NAM program for monitoring network simulation traces

2.4.3. Ns-2 Trace File

The Ns-2 trace file contains the necessary data for a network and protocol performance analysis, such as the number of packets transported, delay, and packet loss. The Figure 2.14 shows an example of the trace file exported after the simulation process.

```
s -t 1.010000000 -Hs 2 -Hd -2 -Ni 2 -Nx 0.00 -Ny 0.00 -Nz 0.00 -Ne -1.000000 -Nl AGT -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 \
 -Is 4194305.0 -Id 0.1 -It cbr -Il 80 -If 0 -Ii 5 -Iv 32 -Pn cbr -Pi 1 -Pf 0 -Po 0
r -t 1.010000000 -Hs 2 -Hd -2 -Ni 2 -Nx 0.00 -Ny 0.00 -Nz 0.00 -Ne -1.000000 -Nl RTR -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 \
 -Is 4194305.0 -Id 0.1 -It cbr -Il 80 -If 0 -Ii 5 -Iv 32 -Pn cbr -Pi 1 -Pf 0 -Po 0
f -t 1.010000000 -Hs 2 -Hd 4194304 -Ni 2 -Nx 0.00 -Ny 0.00 -Nz 0.00 -Ne -1.000000 -Nl RTR -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 \
 -Is 4194305.0 -Id 0.1 -It cbr -Il 100 -If 0 -Ii 5 -Iv 32 -Pn cbr -Pi 1 -Pf 0 -Po 0
s -t 1.010295000 -Hs 2 -Hd 4194304 -Ni 2 -Nx 0.00 -Ny 0.00 -Nz 0.00 -Ne -1.000000 -Nl MAC -Nw --- -Ma 13a -Md 0 -Ms 1 -Mt 800 \
 -Is 4194305.0 -Id 0.1 -It cbr -Il 158 -If 0 -Ii 5 -Iv 32 -Pn cbr -Pi 1 -Pf 0 -Po 0
r -t 1.010584455 -Hs 1 -Hd 4194304 -Ni 1 -Nx 0.00 -Ny 0.00 -Nz 0.00 -Ne -1.000000 -Nl MAC -Nw --- -Ma 13a -Md 0 -Ms 1 -Mt 800 \
 -Is 4194305.0 -Id 0.1 -It cbr -Il 100 -If 0 -Ii 5 -Iv 32 -Pn cbr -Pi 1 -Pf 1 -Po 0
s -t 1.010594455 -Hs 1 -Hd -2 -Ni 1 -Nx 0.00 -Ny 0.00 -Nz 0.00 -Ne -1.000000 -Nl MAC -Nw --- -Ma 0 -Md 1 -Ms 0 -Mt 0
+ 1.010609 1 0 cbr 100 ------- 0 1.0.1.0 0.0.0.1 1 5
- 1.010609 1 0 cbr 100 ------- 0 1.0.1.0 0.0.0.1 1 5
```

Figure 2.14. An example of the trace file

Where the send (s), receive (r), drop (d), and forward (f) determine the type of the event. The second field starting with (-t) may stand for time. The ID of the node (-Hs) and ID of the next hop (-Hd) give hop information. Node ID (-Ni), coordinates of the node (-Nx,y,z), node energy (-Ne), trace level (-NI), and flags (-Nw) in the third group give node information. Duration (-Ma), MAC address destination (-Md), MAC address source (-Ms), and ethernet type (-Mt) in the fourth group give the packet information at the MAC level. Source(-Is), destination (-Id), packet type (-It), packet size (-Il), flow ID (-If), packet ID (-Ii), and TTL value (-Iv) in the fifth group give the packet information at the IP level. The rest of the values give application-level information (Bouras et al., 2012).

Figure 2.14 can be explained as follows: The first event at time t = 1.010000000, is that the agent (AGT) has sent (s) a CBR packet from node 4194305 to node 0, with a payload size of 80 bytes. The second event shows this packet (first event) has been received (r) by the routing agent. The third event is a forwarding (f) event of this packet (ID 5). The trace then continues and shows that packet 5 has been given to the MAC later, and the MAC layer has sent (s) the packet. Following the packet, it is shown that the packet has been received by the MAC layer of the AP (-Hs 1), as this is a wireless network with AP. The last two lines show that the packet has been queued and dequeued in the wired part of the network.

2.4.4. AWK Scripting Language

AWK scripting language, A useful language for processing and collecting data from trace files output from simulation applications. These data may be used to evaluate how well the simulation procedure was performed.

2.5. Related Works

The related works for this thesis is divided into three sections. Research on the sensor deployment issue in WSNs is presented in the first part, while studies on Ns-2 simulation and comparison of routing protocols are presented in the second. In the last section research on the usage of metaheuristic algorithms in routing problems is given.

2.5.1. Research On The Sensor Deployment Issue

WSN sensor deployment has been addressed in various research over the past few years. Here are some of the research that is most relevant to this study. According to Soumya et al. (Bhat & KV, 2022), a completely connected network can be achieved by using the arithmetic optimization algorithm. The algorithm has been tested in a variety of fields. In addition to being able to locate nodes accurately, the algorithm is also capable of identifying them. Lei Wang et al. (L. Wang et al., 2018) developed the swarm optimization algorithm for optimizing the coverage area of WSNs. He turned the problem into a multi-objective optimization problem and proved that the algorithm can effectively improve network coverage. In Rasul et al. study (Enayatifar et al., 2014), MOICA (novel multi-objective imperialist competitive algorithm) is proposed for handling sensor deployment. Based on the results of a set of experiments, MOICA can produce more accurate results for a given number of sensors. The performances of the conventional parallel and cooperative model-based parallel artificial bee colony (ABC) algorithms were investigated in Selcuk study (Aslan, 2019a). The results showed that the cooperative model parallelized the ABC algorithm, achieving similar or better coverage. It was also faster than the parallelized implementation of its serial counterpart. To improve the coverage effect of wireless sensor networks, a network coverage algorithm based on evidence theory has been proposed by Qiangyi et al. (Li & Liu, 2021). This study shows that the monitoring area can be effectively covered, and the nodes can move closer together.

In another study (Kukunuru et al., 2010), the particle swarm algorithm (PSO) was used to find the optimal positions of the sensors to determine the best coverage area. The results show that the PSO approach is effective and robust for the adeguate coverage problem of sensor deployment and is considered to provide almost the optimal solution in WSNs. The algorithm used was called virtual force-directed coevolutionary particle swarm optimization (VFCPSO) (X. Wang & Wang, 2010). The VFCPSO was expanded to include distributed-VFCPSO, hetero-H-VFCPSO, and homo-H-VFCPSO for jointly deploying sensor nodes. The results demonstrate that Homo-H-VFCPSO can deploy sensor nodes in WSNs, particularly large-scale WSNs, quickly and efficiently. Liao et al. (Aslan, 2019a), based on the ant colony optimization algorithm, proposed a distribution scheme to extend network life while providing a solution to the problem of full coverage. Simulations show that our algorithm can extend the life of the network. Simulation results in (Yarinezhad & Hashemi, 2020) demonstrate that these two improved versions of the particle swarm optimization (PSO) algorithms have a more extended network lifetime than similar algorithms in solving the target coverage problem.

(Xie & Zhang, 2022) introduces an improved sparrow search algorithm to deal with deployment issues. In this study, five different swarm intelligence approaches are tested. Results from simulation experiments indicate that both coverage rate and connectivity measurements can be measured with a good performance by the proposed algorithm. In (Carter & Ragade, 2009) explains the issue of using a limited number of sensors to cover a set of target points. It is suggested to create a probabilistic model that takes into account the sensing devices' detection probabilities, which may decrease with distance, the environment, and hardware configurations. The deployment is optimized using a genetic algorithm based on a probabilistic coverage matrix. In comparison to other deployment methods, wth fewer sensors, the proposed probabilistic sensor deployment model identifies more effective solutions.

The authors of (Ni et al., 2017) suggest the VFCPSO algorithm, an enhanced coevolutionary particle swarm optimization algorithm, as a workable strategy for wireless sensor networks with dynamic deployments. Simulating results demonstrate that the proposed VFCPSO algorithm provides good global searching and regional convergence capabilities during optimization and that it is capable of deploying hybrid WSNs with mobile sensor nodes and stationary sensor nodes dynamically, effectively, and robustly. The majority of the studies examined in (Zaimen et al., 2020) research used evolutionary algorithms to improve node location. The authors spoke about their conceptual strategy for deploying WSN in BIM-enabled intelligent buildings. They evaluated the current projects connected to WSN deployment, from classical approaches to evolutionary algorithms and computational geometry techniques. Additionally, the proposed framework was integrated with BIM tools so that database information could be used to automatically create and display deployment solutions.

To solve the WSN coverage issue, the authors of (Liang et al., 2022), developed a user position-based WSN deployment method. To specifically explain the coverage rate of WSN for dynamic users, they created a coverage model. They suggested an enhanced Particle Swarm Optimization (PSO) technique based on the coverage model to find the best solution for dynamic users. The experimental findings show that, in comparison to existing representative methods, the suggested approach can increase the coverage rate for dynamic users. In (G. P. Gupta & Jha, 2019), a BBO-based method for determining the ideal number of carefully chosen locations for the deployment of sensor nodes that meet the network's k-coverage and m-connectivity criteria is provided. Along with the migration and mutation operators from the BBO, the proposed BBO-based approach formulates an objective function and offers an effective encoding technique for habitat representation. The effectiveness of the suggested method for determining the approximate optimum number of eligible places for various combinations of k and m is thoroughly addressed. In addition, a comparative study of the GA-based and the GSA-based methods has also been done, and its analysis confirms the superiority of the proposed BBO-based method over them. The issue of sensor deployment to enhance the lifespan of WSNs with linear topology was investigated by the authors in (Domga et al., 2019). They divided the network into virtual nodes to make deployment easier. They proposed an analytical model for calculating the number of operations per virtual node by taking message transmission and/or reception into account as energy-consuming operations. According to the performance assessment, the suggested simple virtual node-based deployment may extend the network lifespan by up to 40% when compared to the uniform deployment, depending on the number of sensors.

2.5.2. Research On Ns-2 Simulation and Comparison of Routing Protocols

The authors in (MHMOOD & Zengin, 2021), compare DSDV and AODV routing protocols that are proposed for ad-hoc mobile networks. Mobile nodes periodically broadcast their route data to their neighbors in the DSDV routing protocol. Each node must keep up with the upkeep of its routing table. In the AODV protocol, the route is found as necessary by utilizing the route request packet. The characteristics of packet delivery ratio, throughput, end-to-end delay, packet loss rate, and energy spent are used to compare these protocols. According to the simulation findings, the DSDV performs better than the AODV in terms of packet loss and energy consumption, while the AODV performs better than the DSDV in terms of packet delivery ratio and end-to-end latency. The performance of the AODV and the DSDV in the throughput metric was combined. As a result, the DSDV performs better for small networks whereas the AODV is ideal for bigger networks.

The performance of the MANETs' the AODV, the DSR, and the DSDV routing protocols in high mobility cases at various densities—low, medium, and high—is examined in the study (Aggarwal et al., 2014). They changed the number of nodes in a set topography of 1000 x 1000 meters from 25 (low density) to 200 (high density). Additionally, as this study's random waypoint mobility model was utilized to produce node mobility, they averaged out 10 randomly generated situations to conduct a thorough performance analysis. The simulation results show that the AODV performance is the best when examining its capacity to sustain connection via periodic information exchange. And even with a huge number of nodes in the network, the AODV and the DSR perform better than the DSDV in terms of throughput. Generally, the study demonstrates that the DSR performs better in networks with fewer nodes whereas the AODV performs better in networks with more nodes. And with the DSDV, the average End-to-End Delay is the smallest and remains unchanged as the number of nodes rises. They conclude that the AODV is a good option for MANETs, however, when the number of network nodes rises, NRL for the AODV grows more rapidly than it does for the DSDV, and the DSR.

By using the NS-2 Simulator, the authors in (Rao et al., 2015), assessed how well the DSDV, the DSR, the AODV, and the AOMDV protocols performed according to measurements for packet delivery ratio, packet loss, throughput, overhead, and delay. The obtained results were the AOMDV protocol performs better than the AODV and the DSR in terms of packet delivery percentage, throughput, and delay. Additionally, the throughput value of the DSDV was higher, the DSR protocol has less overhead, the AOMDV incurs greater routing cost than the AODV, the AODV has higher packet loss, and the AODV has lower latency in relation to pause duration. The key finding of this research is that the network's characteristics determine which protocol should be used. In (P. Gupta & Verma, n.d.), the simulation results reveal that none of the AODV, the DSR, and the DSDV protocols performs well in all the scenarios proposed by the authors. The AODV and the DSR are preferred in big networks, but the DSDV is only appropriate in small, low-dynamic networks, and in situations involving great movement, AODV is often preferable.

A performance evaluation of three different routing protocols the DSR, the AODV, and the DSDV was proposed in (Kong & Cui, 2017). Using measures for packet delivery ratio, average end-to-end data packet delay, and normalized routing overhead, the authors examined the mentioned protocols. The DSR and the AODV achieve a similar higher packet delivery ratio than the DSDV. Compared to the DSR and the DSDV, the AODV performs better in terms of average delay. In comparison to the AODV and the DSDV, the DSR has the least routing overhead. Also, a performance comparison has been done between the DSR, the AODV, and the DSDV to precisely determine which protocol is more effective in (AL-Dhief et al., 2018). According to the findings, the DSDV routing protocol performs better than the DSR and AODV in terms of packet delivery ratio and packet loss, whereas the AODV performs better than the DSDV and DSR in terms of average throughput. When it comes to end-to-end data packet delay, the DSR is superior to other protocols. As a result, the DSDV is the best protocol when compared to the AODV and the DSR protocols due to the significance of the packet delivery ratio.

2.5.3. Research On The usage of Metaheuristic Algorithms In Routing Protocols

The authors introduced a novel on-demand routing strategy for mobile multi-hop ad hoc networks in (Thottempudi & Umar, 2013). Swarm intelligence, namely the ant colony optimization meta-heuristic, is the foundation of this strategy. They examined the method's adaption to mobile multi-hop ad hoc networks and demonstrated via simulations how effective it is in these networks. The performance for the simulation situations under consideration is quite comparable to that of DSR, but it produces less overhead. To enhance the qualitative characteristics of the routing process, the authors in (J. Jain et al., 2011), examined the use of the fundamental ant colony algorithm in the context of mobile ad hoc networks. The flexibility of the ant colony algorithm provides numerous approaches to modify mobile ad-hoc networks. Each node's local data is utilized as a heuristic parameter. In the event of a broken connection, this algorithm may be used. Reactive routing will be used to find the route, and all nodes in the connection will regularly send out "HELLO" messages to keep it alive. In the case of a link, the failure node will find an alternate route with the help of information available in the routing table about the next active node in the link. Therefore, the time required to build a route from a source to a destination was significantly reduced, which is crucial for a reactive routing algorithm.

In (Singh et al., 2019), many directing conventions have been examined using a variety of criteria. A relative investigation has been done between the proactive and responsive kinds of directing conventions. After that, AODV directing convention enhancement is suggested using bio-roused techniques. To find the optimal route for guiding packages, the AODV is combined with the bio-motivated tactics of Cuckoo search and honeybee settling. The structured half-and-half convention is thought to optimize plan execution and provide better results in terms of bundle loss, throughput, and delay. The authors in (Horane et al., 2013), propose a useful routing system that relies on swarm intelligence, in particular the ACO. The difficult issue of routing in MANET is resolved by ACO using a group of mobile agents (forward, backward, and update ants). They showed via the comparison of ACORA, AODV, and DSR that ACO performs better in terms of routing performance in terms of energy usage than more traditional routing techniques like AODV and DSR.

In (Saba et al., 2022), the authors evaluated the three meta-heuristic algorithms GA, PSO, and GSO and they introduced four Dos, D-dos, Wormhole, and Sinkhole attacks. The effectiveness of these algorithms was then assessed using the four metrics which are delay, throughput, energy use, and degree of warnings. According to the results of our investigation, the PSO algorithm has the best throughput, and the GA method has the maximum energy. As a result, the two algorithms PSO and GA may be combined to create a new algorithm that has high energy and high throughput with fewer false warnings and extremely good network efficiency. To reduce the needed energy consumption, the authors (Abdali et al., 2020), improved conventional PSO and incorporated it into the LAR procedure. The assessment via simulation results demonstrates that the proposed OPSO-LAR can achieve high performance in a network environment compared to the relevant state of the art. Particularly, the PSO technique was used to optimize the parameter in the computing function. Additionally, it was utilized to choose two or more values for the FR and CZR parameters, which regulate network flooding and node coverage.

In WSNs, energy efficiency is a crucial problem. Energy is perhaps the most important resource restriction since sensor nodes are often battery-powered and the number of hours the network can operate depends on the battery's lifespan. The authors in (Golzari & Hosseinkhani, 2017), have presented a new protocol for WSN routing operations. The protocol is implemented by optimizing routing pathways using the ACO algorithm, which offers a reliable multi-path data transmission technique to achieve communications in the event of node failures. In (Chandren Muniyandi et al., 2021), the authors investigated the evaluation performance of the OLSR routing protocol to auto-tune the parameters of this routing protocol and find the best configurations suited for improving the performance of routing protocols for vehicular ad hoc networks, which are known for their unique and challenging features. Additionally, they investigated employing a new meta-heuristic technique called Harmony Search Optimization, to optimize the OLSR parameters for the VANET. The outcomes of the simulation, which were used in a highway scenario, demonstrated that the IHS could identify more optimal OLSR parameter configurations. Authors in (A. Jain & Dixit, n.d.), evaluated the security and congestion implications of several mobile ad hoc network routing strategies. They examined the pressure against ad hoc routing and provided the requirements that must be met for secure routing. To send data in a way that is free of congestion and vulnerable to attack, they have designed a PSO superior safe routing weave. For regulating a secure communication system in mobile ad hoc networks, the Secure Compromising route Algorithm offers a solid basis.

3. MATERIALS AND METHODS

Optimization term refers to the process of determining the most optimal solution from a set of alternatives for a given issue (Lange, 2013). The use of optimization methods to solve the maximum coverage problem is a great approach. By applying these algorithms to the given issue, we may increase the coverage area that is covered by the sensors. The IP algorithm will be used in this thesis to get optimal results. It is essential to know about the IP algorithm and a little bit about other algorithms used for comparisons.

3.1. Immune Plasma (IP) Algorithm

In 2020, Selcuk and Sercan invented a novel meta-heuristic algorithm known as the IP algorithm, which is based on giving the antibody-rich portion of the blood collected from patients who have already recovered to critical persons (Aslan & Demirci, 2020). Let us first look at the immune system to explain the general steps of the IP algorithm. The immune system begins different defense procedures to search for and destroy antigens when they trigger an infection with the help of lymphoid organs that produce lymphocytes such as T and B lymphocytes produced in the bone marrow and play an essential role in the adaptive immune response (Hung et al., 2011) (Parkin & Cohen, 2001) (Delves & Roitt, 2000). When B lymphocyte receptors attach to a specific antigen, helper T cells are activated and produce substances that allow B lymphocytes to grow and develop into plasma cells. Interleukins and interferons are the chemical names for these substances. Plasma cells produce antibodies, which are cells that produce proteins (Hung et al., 2011) (Parkin & Cohen, 2001) (Delves & Roitt, 2000). These antibodies are made up of Y-shaped polypeptide chains. Aid in the binding of antigens for which antibodies are produced.

Antibody response begins a few weeks after an infection is triggered, develops gradually to a peak around ten days, and then declines over time. The issue here is that certain people with weakened immune systems or immune system illnesses are incapable of making enough antibodies (Hung et al., 2011) (Parkin & Cohen, 2001) (Delves & Roitt, 2000). A therapeutic procedure known as a convalescent plasma or immune plasma has been developed as a remedy to this problem and has been successfully tested for various ailments. The blood, or the antigen-rich component of it known as plasma, from patients who have recovered from the same infection can be utilized as a helpful supply for sick people in this treatment (Hung et al., 2011) (Parkin & Cohen, 2001) (Delves & Roitt, 2000). The notion of immune plasma therapy is what led to the creation of the IP algorithm, a brand-new meta-heuristic. Identifying the person or persons who recovered quickly from infection, identifying the actions taken by the immune system at the start of the infection, and transferring plasma to the crucial person affected by the infection are the fundamental operations that contribute to the exploration of the IP algorithm. The immune response of an individual in the IP algorithm represents the quality of the corresponding solution, even though each member of the population in the population indicates a potential solution to the problem being optimized (Aslan & Demirci, 2020). The general concept of the IP algorithm is listed below, the flowchart is given in Figure 3.1, and the pseudo-code is shown in Alg. 3. The first three lines describe the parameters of the algorithm. The infection phase is described in lines 5 to 14, while the plasma transferring phase is covered in lines 19 to 48. Finally, the donor update phase is discussed in lines 49 to 60.

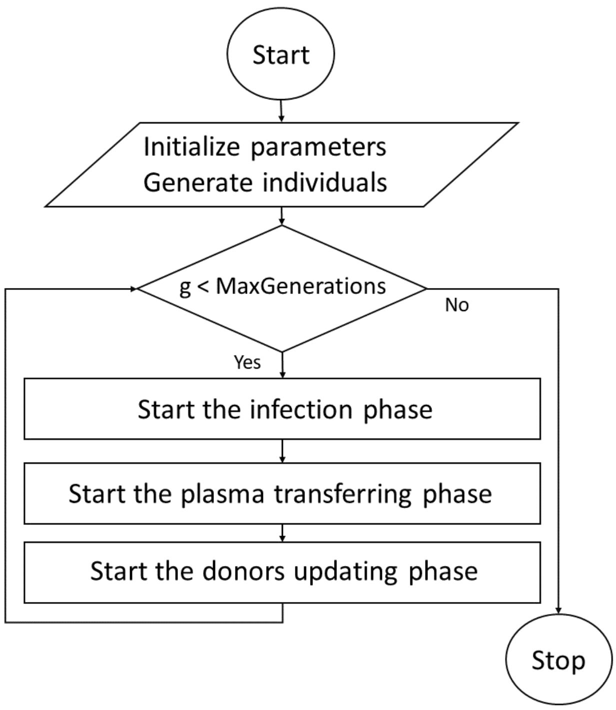


Figure 3.1. Flowchart of the general concept of the IP algorithm

Algorithm 3 Pseudo-code of IP algorithm

Assign values to PS, N, NoD and NoR control parameters
Set x_{best} as the best individual of PS individuals
Select t_{max} and set t_{cr} to PS.
while $t_{cr} < t_{max}$ **do** do
  for $k = 1$ to PS **do** do
    if $t_{cr} < t_{max}$ **then** then
      $t_{cr} = t_{cr} + 1$
      x_k^{inf} = infect x_k with x_m using Eq. 3.2.
      if $f(x_k^{inf}) < f(x_k)$ **then** then
        update x_k with x_k^{inf} as described in Eq. 3.3.
        update x_{best} with x_k if $f(x_k) < f(x_{best})$
      end if
    end if
  end for
  $doseControl[1...NoR]$ = set each element to 1.
  $treatmentControl[1...NoR]$ = set each element to 1.
  $dIndexes[1...NoD]$ = get the indexes of the donors.
  $rIndexes[1...NoR]$ = get the indexes of the receivers.
  for $i = 1$ to NoR **do** do
    update x_{best} with x_k if $f(x_k) < f(x_{best})$
    if $t_{cr} < t_{max}$ **then** then
      $k = rIndexes[i]$, and $m = a$ random element from $dIndexes$.
      x_k^{rcv} = get the kth individual from population.
      x_k^{dnr} = get the mth individual from population.
      while $treatmentControl[i] == 1$ **do** do
        if $t_{cr} < t_{max}$ **then** then
          $t_{cr} = t_{cr} + 1$
          x_k^{rcv-p} = plasma treatment to x_k^{rcv} with x_k^{dnr} using Eq. 3.4.
          if $doseControl[I] == 1$ **then** then
            if $f(x_k^{rcv-p}) < f(x_k^{dnr})$ **then** then
              $doseControl[i] = doseControl[i] + 1$
              update x_k^{rcv} with x_k^{rcv-p}
            else
              update x_k^{rcv} with x_m^{dnr}
              set $treatmentControl[i] = 0$ for completing transfer.
            end if
          else
            if $f(x_k^{rcv-p}) < f(x_k^{rcv})$ **then** then
              update x_k^{rcv} with x_k^{rcv-p}
            else
              set $treatmentControl[i] = 0$ for completing transfer.
            end if
          end if
          update x_{best} with x_k^{rcv} if $f(x_k^{rcv}) < f(x_{best})$
        end if
      end while
    end if
  end for
  for $k = 1$ to NoD **do** do
    if $t_{cr} < t_{max}$ **then** then
      $t_{cr} = t_{cr} + 1$
      x_{dnr} = get the mth individual from the population.
      if $t_{cr}/t_{max} < rand(0, 1)$ **then** then
        update x_m^{dnr} using Eq. 3.5.
      else
        update x_m^{dnr} using Eq. 3.1.
      end if
      update x_{best} with x_m^{dnr} if $f(x_m^{dnr}) < f(x_{best})$
    end if
  end for
end while

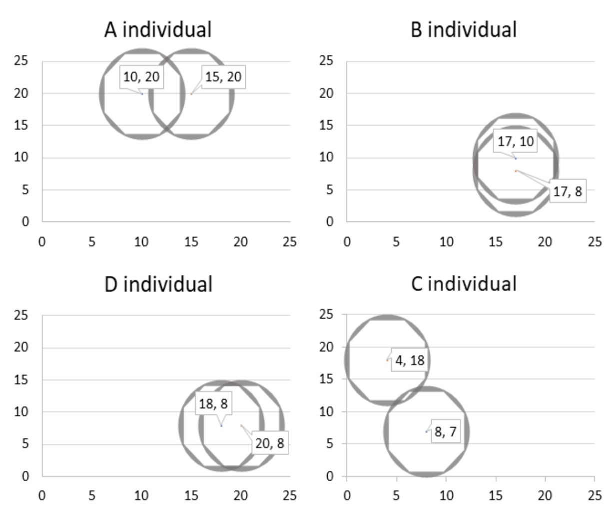
3.1.1. Generating Initial Individuals

Every individual of the IP algorithm is connected to a possible solution to the issue at hand. Assuming the optimization problem at hand has D distinct choice parameters, the j^{th} decision parameter of the x_k individual may be used to construct the x_{kj} individual as given in Eq. 3.1 (Aslan & Demirci, 2020).

$$x_{kj} = x_j^{low} + \text{rand}(0,1)\left(x_j^{high} - x_j^{low}\right) \qquad \forall k \in \{1, 2, \ldots, PS\} \quad , \quad \forall j \in \{1, 2, \ldots, D\} \tag{3.1}$$

where x_j^{low} and x_j^{high} correspond to the lower and higher limits of the j^{th} decision parameter, and $\text{rand}(0,1)$ is randomly chosen value between 0 and +1. An example of this phase is given in Figure 3.2.

Sensors	A	B	C	D		Population	Donor	Receiver
1. x,y	A_1(10,20)	B_1(17,10)	C_1(4,18)	D_1(18,8)	Number	4	1	1
2. x,y	A_2(15,20)	B_2(17,8)	C_2(8,7)	D_2(20,8)	Individual	A,B,C,D	?	?



- **Generating initial individual by using** Eq. 3.1

x^{low}	x^{high}	$Rand$	k
0	100	[0,1]	$individual$

$A_{1x} = 0 + 0.1 * (100 - 0) = 10$
$A_{1y} = 0 + 0.2 * (100 - 0) = 20$

$A_{1(x,y)} = (10,20)$

$A_{2x} = 0 + 0.15 * (100 - 0) = 15$
$A_{2y} = 0 + 0.2 * (100 - 0) = 20$

$A_{2(x,y)} = (15,20)$

By following the same operations other individuals can be generated.

Figure 3.2. Example of generating initial individuals

3.1.2. Infection Spreading and Immune System Reaction

A small number of sick people can have a significant effect on the entire population. An infection can distribute easily among individuals of a population. In the IP algorithm, Eq. 3.2 is used to simulate how an individual's secretions affect another individual and trigger the immune system to generate a specific response.

Both of m and k indexs are selected form PS but m must be different than the k.

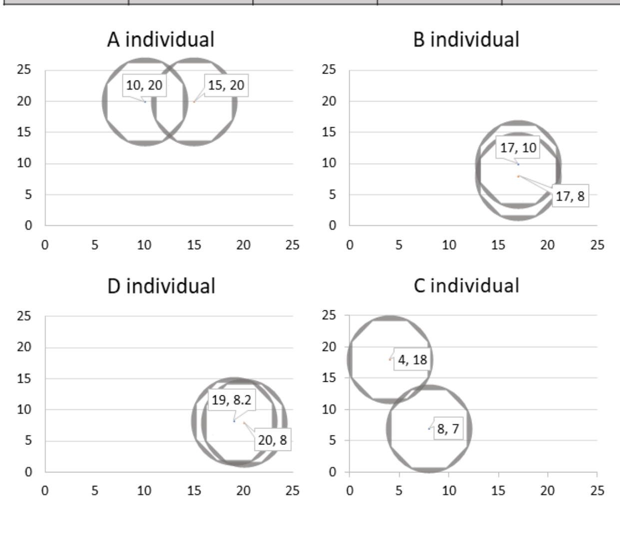
$$x_{kj}^{inf} = x_{kj} + \text{rand}(-1, 1)\left(x_{kj} - x_{mj}\right)$$
$$\forall k \in \{1, 2, \ldots, PS\} \quad , \quad \forall m \in \{1, 2, \ldots, PS\} - k \tag{3.2}$$

where x_{kj} and x_{mj} are matched with j^{th} decision parameter of the and x_k and x_m. x_{kj}^{inf} is the new individual matched with the j^{th} decision parameter of the infectious x_k individual. As time goes by, some individuals recover quickly while others become critical (Aslan & Demirci, 2020). For a minimization problem, after infection if the immune response of the infectious x_k individual is less than the immune response of the infectious x_k individual before infection it is assumed that the x_k individual is can manage infection without additional treatments being used and x_{kj} is updated with x_{kj}^{inf}. Otherwise, x_{kj} it is not updated as shown in Eq. 3.3. An example of this phase is given in Figure 3.3.

$$x_{kj} = \begin{cases} x_{kj}^{\text{inf}}, & \text{if } f\left(x_{kj}^{\text{inf}}\right) < f\left(x_k\right) \\ x_{kj}, & \text{if } f\left(x_{kj}\right) \geqslant f\left(x_k\right) \end{cases} \tag{3.3}$$

Sensors	A	B	C	D
1. x,y	A_1(10,20)	B_1(17,10)	C_1(4,18)	D_1(18,8)
2. x,y	A_2(15,20)	B_2(17,8)	C_2(8,7)	D_2(20,8)

	Population	Donor	Receiver
Number	4	1	1
Individual	A,B,C,D	?	?



- **Infection spreading by using** Eq. 3.2

x_{kj}	x_{mj}	*Rand*
D_1	B_1	1 for x and −0.1 for y axis

x_m is a random individual – here, is B
x_k is a random individual but not m – here, is D
x_k^{inf} newly infected individual – here is newD

$D_{1(x,y)} = (18,8)$, $B_{1(x,y)} = (17,10)$

$newD_{1x} = 18 + 1 * (18 - 17) = 19$
$newD_{1y} = 8 + (-0.1) * (8 - 10) = 8.2$

$D_{1(x,y)} = (19,8.2)$

As shown in the left figure, the coverage area of individual D is now lower than before.

Figure 3.3. Example of infection spreading

3.1.3. Plazma Extraction and Transfer

Some individuals who have recovered may be able to provide plasma for seriously critical patients. The plasma that is antibody-rich is taken from donors' blood and can be used to support the immune system of a patient who needs it. Due to this, the IP Algorithm first identifies the number of donors (NoD) and receivers (NoR) before starting plasma transfer operations. The best NoD individuals are donor individuals, whereas the worst NoR individuals are recipient individuals (Aslan & Demirci, 2020). The flowchart below Figure 3.4 describe the plasma extraction and transfer in the IP algorithm.

For simulating plasma transfer from the x_m^{dnr} individual to the x_k^{rcv} receiver individual, the IP algorithm uses Eq. 3.4.

$$\begin{gathered} x_{kj}^{rcv-p} = x_{kj}^{rcv} + \mathrm{rand}(-1,1)\left(x_{kj}^{rcv} - x_{mj}^{dnr}\right) \\ \forall j \in \{1,2,\ldots,D\} \end{gathered} \tag{3.4}$$

Where x_m^{dnr} is an individual chosen randomly from donors, x_k^{rcv} is the x_k receiver individual and x_k^{rcv-p} individual represents the x_k^{rcv} after plasma treatment (Aslan & Demirci, 2020). After the first plasma dosage, if the immune response of the x_k^{rcv-p} is better than the immune response of the donor individual x_m^{dnr} such as $f\left(x_k^{rcv-p}\right) < f\left(x_m^{dnr}\right)$ the x_k^{rcv} individual is updated with x_k^{rcv-p} and the plasma treatment for x_k^{rcv} continues with the x_m^{dnr} donor individual, otherwise the x_k^{rcv} is updated with x_m^{dnr}. If the first dosage of plasma is effective, x_k^{rcv} is already better than x_m^{dnr}, therefore comparing $f\left(x_k^{rcv-p}\right)$ and $f\left(x_k^{rcv}\right)$ values can help determine whether plasma therapy should continue with subsequent doses (Aslan & Demirci, 2020). An example of this phase is given in Figure 3.5.

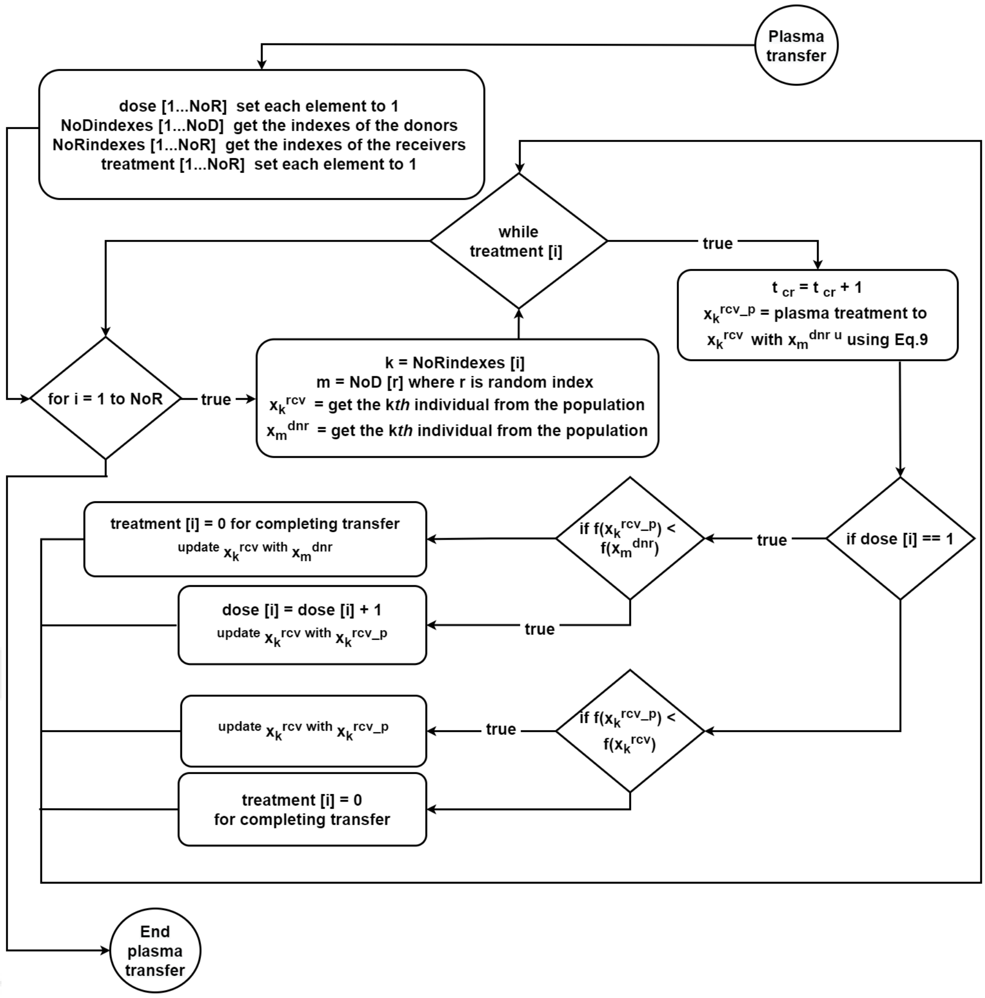
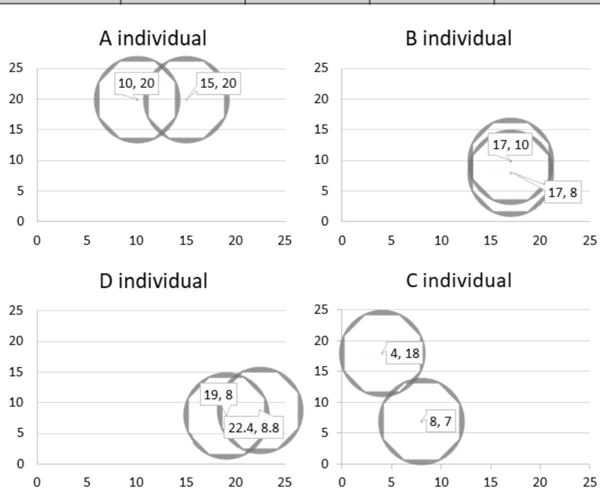


Figure 3.4. Plazma extraction and transfer in the IP algorithm

Sensors	A	B	C	D
1. x,y	$A_1(10{,}20)$	$B_1(17{,}10)$	$C_1(4{,}18)$	$D_1(19{,}8.2)$
2. x,y	$A_2(15{,}20)$	$B_2(17{,}8)$	$C_2(8{,}7)$	$D_2(20{,}8)$

	Population	Donor	Receiver
Number	4	1	1
Individual	A,B,C,D	C	D



- **Plazma transferring by using** Eq. 3.4

x_{kj}^{rcv}	x_{mj}^{dnr}	$Rand$
D_2	C_1	0.2 for x and 0.8 for y axis

x_{mj}^{dnr} is a donor – here, is C
x_k^{rcv} is a receiver – here, is D
x_k^{rcv-p} is a treated receiver – here is newD

$D_{2(x,y)} = (20{,}8)$, $\quad C_{2(x,y)} = (8{,}7)$

$newD_{2x} = 20 + 0.2 * (20 - 8) = 22.4$
$newD_{2y} = 8 + 0.8 * (8 - 7) = 8.8$

$D_{2(x,y)} = (22.4{,}8.8)$

As shown in the left figure, the coverage area of individual D is now better than the last phase.

Figure 3.5. Example of plazma extraction and transfer

3.1.4. Donors Update

The IP algorithm employs a controlled-randomized strategy to replace the previously selected donor persons when the plasma therapy is finished.If the ratio between the present fitness evaluation x_{tr} and the declared maximum fitness evaluation x_{max} is less than or equal to the random integer generated between 0 and +1,By using the previously provided values as a guide, the x_m^{dnr} individual is modified as indicated in Eq. 3.1. Otherwise, it is thought that the individual's immune system reaction benefited individuals in need. However, not sufficient to produce stationary antibody memory, so its parameters are reset with Eq. 3.5 (Aslan & Demirci, 2020).

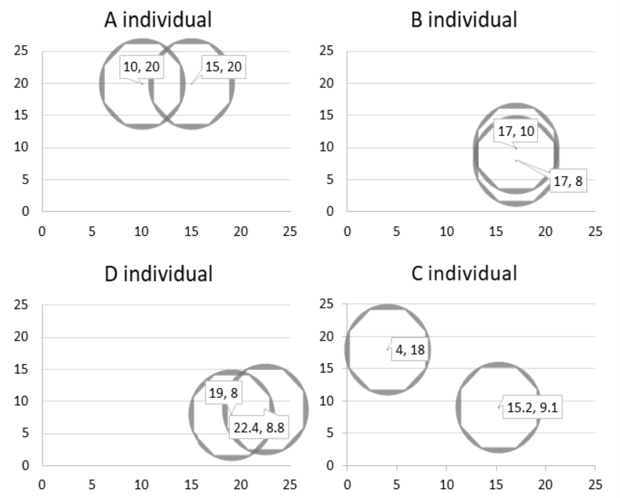
$$x_{mj}^{dnr} = x_{mj}^{dnr} + \text{rand}(-1,1)\left(x_{mj}^{dnr}\right) \tag{3.5}$$

It's safe to argue that IP algorithms include phrases for exploration and exploitation. Exploration ensures the algorithm reaches different promising regions of the search space, whereas exploitation ensures the search for optimal solutions within the given region. Finding the donor individual or individuals during the IP algorithm's plasma extraction phase is referred to as exploration, whilst updating the donors at the IP algorithm's final stage to have more effective donors is referred to as exploitation (X. Wu, Zhang, Xiao, & Yin, 2019). An example of this phase is given in Figure 3.6.

3.2. Modified Immune Plasma (mIP) Algorithm

Plazma extraction and transfer in the mIP algorithm are the same as in the IP algorithm but when updating the individual if the immune response of the x_k^{rcv-p} is better than the immune response of the donor individual x_m^{dnr} such as $f\left(x_k^{rcv-p}\right) < f\left(x_m^{dnr}\right)$ continue the treatment, otherwise stop the treatment. When the treatment is stopped if the immune response of the x_k^{rcv-p} is better than the immune response of the donor individual x_k^{rcv} such as $f\left(x_k^{rcv-p}\right) < f\left(x_k^{rcv}\right)$ the x_k^{rcv} is updated with x_k^{rcv}. The flowchart below Figure 3.7 describe the plasma extraction and transfer in the mIP algorithm.

Sensors	A	B	C	D		Population	Donor	Receiver
1. x,y	$A_1(10,20)$	$B_1(17,10)$	$C_1(4,18)$	$D_1(19,8.2)$	Number	4	1	1
2. x,y	$A_2(15,20)$	$B_2(17,8)$	$C_2(8,7)$	$D_2(22.4,8.8)$	Individual	A,B,C,D	C	D



- **Donor update by using** Eq. 3.5

x_{mj}^{dnr}	*Rand*
C_2	0.9 for x and 0.3 for y axis

x_{mj}^{dnr} is a donor – here, is C

$C_{2(x,y)} = (8,7)$

$newC_{2x} = 8 + 0.9 * 8 = 15.2$
$newC_{2y} = 7 + 0.3 * 7 = 9.1$

$C_{2(x,y)} = (15.2,9.1)$

The coverage area of individual C has changed, as illustrated in the left figure. As a result, the algorithm is preparing to produce new results.

Figure 3.6. Example of donors update

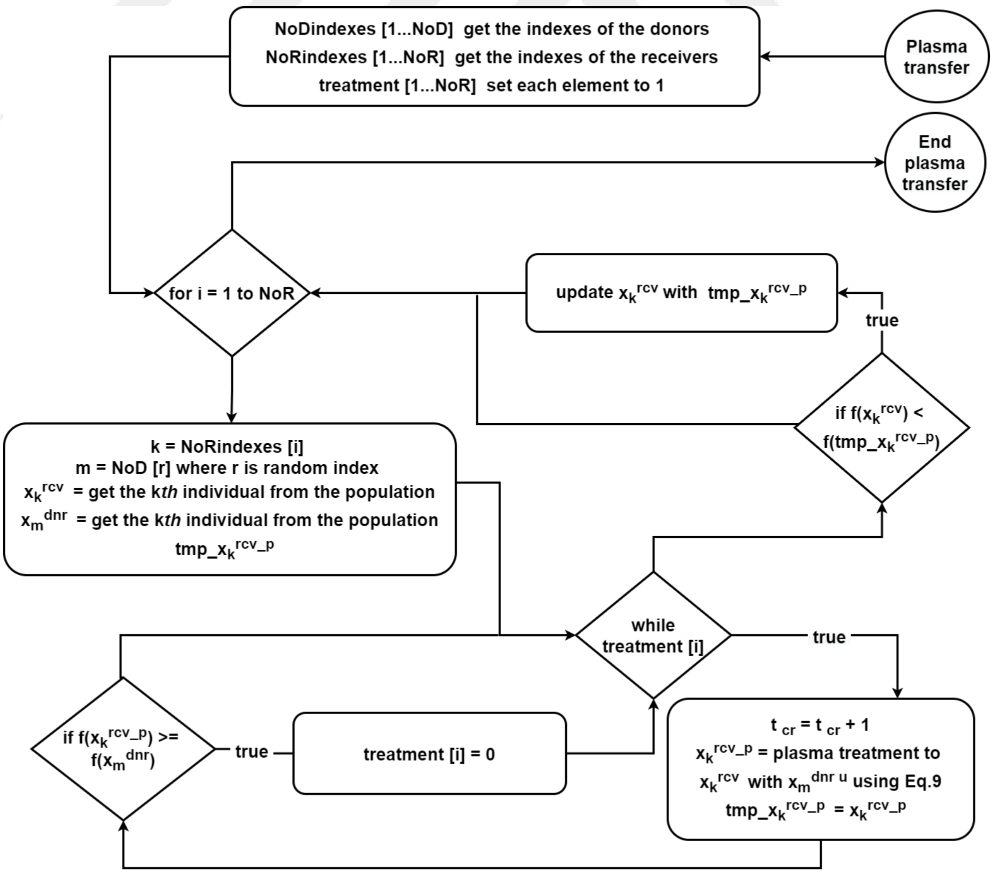


Figure 3.7. Plazma extraction and transfer in the mIP algorithm

3.3. Artificial Bee Colony (ABC) Algorithm

Dervis Karaboga defined the artificial bee colony (ABC) algorithm, which was inspired by the clever behavior of honeybees. Employed bees linked to certain food sources, onlooker bees watching employed bees dance within the hive to choose a food source, and scout bees looking for food sources at random are the three types of bees in the ABC algorithm. Scouts and onlookers are sometimes referred to as jobless bees. Scout bees first locate every location of a food supply. After that, employed bees and onlooker bees use the nectar of food sources, and their continued exploitation exhausts these sources. The employed bee, which was using the exhausted food source, then resumes its role as a scout bee in search of other food sources. In ABC, the location of a food supply symbolizes a potential solution for the issue, and a food source's nectar content reflects the caliber (fitness) of the corresponding solution. Since each employed bee is linked to one and only one food source, the number of employed bees equals the number of food sources (solutions). The initialization phase, employed bees phase, onlooker bees phase, and scout bees phase are the general steps of the ABC algorithm (Aslan & Demirci, 2019).

3.4. Firefly Algorithm (FFA)

An insect that emits subtle lights is called a firefly. The firefly's body contains several chemicals on which the flashlight relies. firefly inhabits hot, tropical climates, fireflies prefer to only be active at night, when the temperature drops below the peak daytime temperature. fireflies emit light to entice and mate with their mates, to signal to one another the whereabouts of food supplies, or even to trick and capture their victims. After studying these behaviors, Xin-She Yang created this algorithm, which is categorized as a swarm-based meta-heuristic algorithm and performed well when used to solve optimization issues. Because they are unisex, fireflies are drawn to all other brighter ones. The brightness directly affects the attraction of fireflies. As a result, the less brilliant bright ones are drawn to the brighter ones. If there isn't another firefly nearby that is brighter, the firefly will update its position at random. To decide how to calculate the firefly's brightness (fitness), an objective function must be determined. Those rules are the major rules for the firefly algorithm (Aliwi, Aslan, & Demirci, 2020).

3.5. Animal Migration Optimization (AMO) Algorithm

Animal behavior ecologists have been studying animal migration extensively. They described it as an often-seasonal, somewhat long-distance relocation of animals to new environments. Animal migration is often triggered by things like a shortage of food supplies, local temperature changes, the time of year, or mating. Each bird modifies its location in relation to its six or seven nearest neighbors, according to recent studies of starling flocks. Animals must adhere to three key principles while migrating: they follow their neighbors' movements, stay close to them, and steer clear of collisions. Animal migration optimization (AMO) is a novel swarm intelligent algorithm that was suggested by X. Li and colleagues in accordance with the principles given before. This algorithm follows these assumptions the animal with the best fitness—also referred to as a group leader—will survive and reproduce and the population size is constant, and with a certain degree of probability, some animals will depart the group and other animals will replace them. The initialization, migration, population updates, and fitness comparison steps make up the AMO algorithm's four main steps (Duraki, Aslan, & Demirci, 2021).

4. EXPERIMENTAL RESULTS

In this chapter, the experimental settings for each of the used algorithms will be determined first, as well as the simulation program (Ns-2) settings. Then several experiments involving various circumstances will be described.

However, it's a good idea to examine the system as a whole before delving into the details. The overall description of the system is as shown in Figure 4.1.

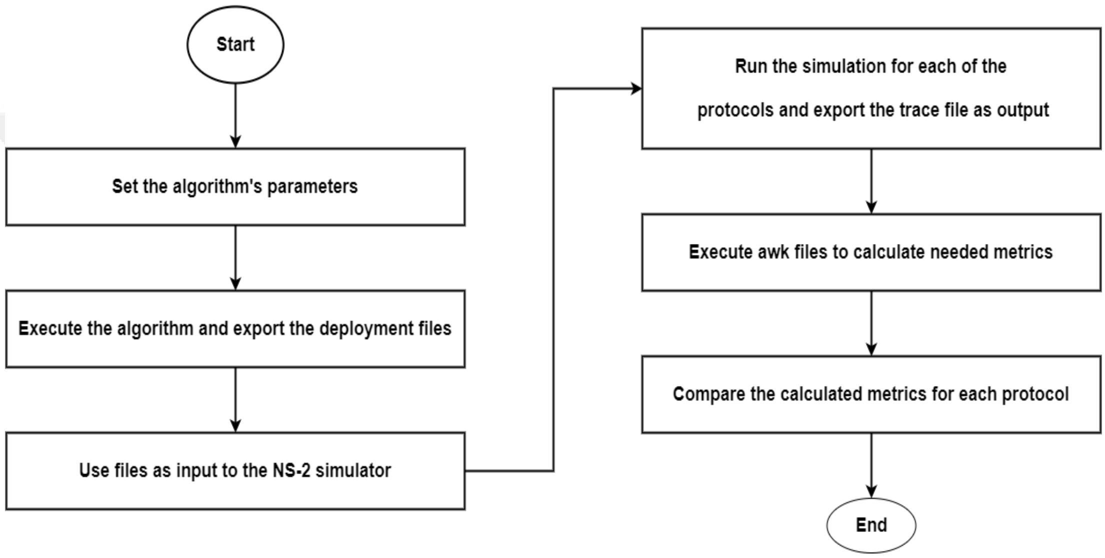


Figure 4.1. The overall description of the system

4.1. Experimental Settings

In this section, each of the IP and mIP algorithms and the other algorithms (ABC, FFA, and AMO) used for comparison will be determined.

4.1.1. Settings of The IP and mIP Algorithms

In order to find the optimal solution to maximum coverage, we ran numerous simulations based on two versions of IP algorithm adapted for different numbers of donors, receivers, evaluations, populations, and detection models. Parameter settings and scenarios are explained in Table 4.1. The number of sensors placed on an area of 100x100m^2 is 100, and the detection distance of each sensor is $7m$. Two detection models were used for each version of the IP algorithm. Each model was analyzed by running the algorithm 30 times with different population sizes and NoD/NoR.

Table 4.1. Settings of the IP and mIP Algorithms

Parameters	Value
Population size (PS)	10, 20, 30, 40, 50
No. of donors and receivers (Nor),(NoD)	(1,1), (1,2), (1,3), (1,4), (2,1), (2,2), (3,1), (3,3), (4,1), (4,4)
No. of runs	30
No. of evaluations (NoE)	10000
Objective function	Probabilistic and binary detection models
Field size (2D)	100x100 m2
No. of sensors (NoS)	100
Sensor sensing range (R)	7m

4.1.2. Settings of The ABC, FFA, and AMO Algorithms

The parameters used for each ABC, FFA, and AMO algorithm are shown in Table 4.2, Table 4.3 and Table 4.4 respectively.

Table 4.2. Settings of the ABC Algorithm

Parameters	Value
Population size (PS)	50
No. of runs	30
No. of evaluations (NoE)	10000
Objective functions	Probabilistic and binary detection models
Field size (2D)	100x100 m2
No. of sensors (NoS)	100
Sensor sensing range (R)	7m

4.1.3. Settings of Ns-2 Simulation

In this subsection simulation parameters and nodes' parameters will be mentioned in Table 4.5 and Table 4.6 tables respectively. The endpoints where network layer packets are created and consumed are represented by agents. In Ns-2, the agent's responsibility is to provide the destination address and perform packet generation. There are many agent kinds and different role types in Ns-2 such as TCP, UDP, TCPSink, and

Table 4.3. Settings of the FFA Algorithm

Parameters	Value
PS	10
NoE	10000
Objective function	Binary detection models
Field size (2D)	100x100 m2
NoS	100
R	7m
α	0.5
β	0.1
γ	1.0

Table 4.4. Settings of the AMO Algorithm

Parameters	Value
PS	10
NoE	10000
Objective function	Binary detection models
Field size (2D)	100x100 m2
NoS	100
R	7m

NULL agents. The TCP agent is for emitting TCP traffic, the UDP agent is for emitting UDP traffic, the TCPSink agent is for the receipt of TCP traffic and the NULL Agent is for receiving UDP packets (NAGAR et al., 2014). In this thesis, TCP and TCPSink agents are used and the packet size is 1500 bytes.

4.2. Optimization and Simulation Results

The results of both the optimization and simulation processes are presented in this section.

4.2.1. Optimization Results

A set of experimental studies has been conducted to analyze the performance of the IP and the mIP algorithms in solving the sensor deployment problem. The IP algorithm simulation results for the probabilistic and the binary detection model are shown in Table 4.7 to Table 4.16, whereas the mIP algorithm simulation results are

Table 4.5. Simulation parameters

Parameters	Value
No. of sensors (nn)	100
Channel type (chan)	Channel/WirelessChannel
Radio-propagation model (prop)	Propagation/TwoRayGround
Network interface type (netif)	Phy/WirelessPhy
MAC type (mac)	Mac/802-11
Interface queue type (ifq)	Queue/DropTail/PriQueue and CMUPriQueue
Link layer type (ll)	LL
Antenna model (ant)	Antenna/OmniAntenna
Max packet in ifq (ifqlen)	50
Routing protocol (rp)	AODV, DSR, and DSDV
X dimension of topography (x)	100
Y dimension of topography (y)	100
Time of simulation (stop)	300.0

Table 4.6. Nodes parameters

Parameters	Value
adhocRouting	Value of rp
llType	Value of ll
macType	Value of mac
ifqType	Value of ifq
ifqLen	Value of ifqlen
antType	Value of ant
propType	Value of prop
phyType	Value of netif
channel	Value of chan
topoInstance	new Topography
energyModel	EnergyModel
initialEnergy	250.0
txPower	0.9
rxPower	0.7
idlePower	0.6
sleepPwer	0.1
agentTrace	ON
routerTrace	ON
macTrace	ON
movementTrace	ON

shown in Table 4.17 to Table 4.26.

The results in Table 4.7 to Table 4.11 indicate a clear improvement in the general coverage of the sensors when using the IP algorithm, as the average is often 10 to 16 percent higher than the primitive coverage for some scenarios. Similarly, the results of the probabilistic model from Table 4.12 to Table 4.16 indicate an increase in the general coverage of the sensors by around 8 to 12 percent.

By analyzing the results from Table 4.17 to Table 4.21, it is clear that the general coverage of the sensors improved when using the mIP algorithm. In some scenarios, improvements in results were higher than initial coverage, by 13 to 17 percent. According to the results of the probabilistic model mentioned in Table 4.22 to Table 4.26, some scenarios improved by 10 to 14 percent.

Considering the nature of the probabilistic model, it is natural that the binary model's results are better than the probabilistic model. Additionally, when applying the binary model for IP and mIP, the algorithms give good results when NoD is higher than NoR, but not for the probabilistic model because the algorithms are parametrically designed.

Table 4.27 shows a comparison of the results found in the binary model with other studies. It is clear that the results of IP and mIP are better or somewhat similar to the results of other algorithms.

Furthermore, as shown in Figure 4.2, after running the IP and the mIP algorithms in binary and probabilistic models sensors are located at good positions and distributed throughout the region. Also as seen in Figure 4.3 and Figure 4.4, each of the IP and mIP in binary and probabilistic results models has more coverage ratio than the other algorithms where each algorithm's two samples, algorithm-1 and algorithm-2, were employed for comparison. According to the "No Free Lunch" theory which states that when averaged across all optimization problems without resampling, all optimization strategies perform equally well. That does not imply that the other algorithms are ineffective or do not provide satisfactory results but according to the "no free lunch" theorem, which states for certain optimization issues, some algorithms may provide high performance or results; for other optimization problems, they may provide unsatisfactory performance or results (Adam et al., 2019). The reason the grid model has a better solution in binary and probabilistic than the results of the others is the grid model sensors are evenly distributed across the region. Despite being a decent option for coverage, grid deployment is not ideal since it is a part of the grid topology, which has certain drawbacks due to the limited distance between mesh nodes. As a result, you may need to include nodes in your network that are not necessarily essential only to maintain your mesh's connectivity. Additionally, the interconnected nature of a mesh network means that if one node fails at a choke point, the whole network may collapse.

Table 4.7. IP's optimization results in binary model when PS is 10

NoR,NoD	Mean	Std Dev	Best	Worst	Initial mean
1,1	0.95914	0.00273	0.00423	0.00645	0.79344
1,2	0.95849	0.00477	0.00838	0.01132	0.79332
1,3	0.96274	0.00534	0.00882	0.01138	0.79388
1,4	0.96150	0.00435	0.00708	0.01076	0.79069
2,1	0.82816	0.00650	0.01079	0.01538	0.79584
2,2	0.95465	0.00776	0.02513	0.01094	0.79326
3,1	0.83034	0.00693	0.01160	0.01379	0.79088
3,3	0.93248	0.00824	0.02110	0.01517	0.79121
4,1	0.82833	0.00640	0.01312	0.01296	0.79176
4,4	0.92462	0.00965	0.01853	0.01685	0.79491

Table 4.8. IP's optimization results in binary model when PS is 20

NoR,NoD	Mean	Std Dev	Best	Worst	Initial mean
1,1	0.94271	0.00331	0.00702	0.00602	0.79717
1,2	0.93266	0.00579	0.01099	0.01107	0.79740
1,3	0.94121	0.00832	0.01778	0.01800	0.79605
1,4	0.94498	0.00806	0.02712	0.01336	0.79770
2,1	0.82334	0.00629	0.01009	0.01844	0.79668
2,2	0.93578	0.00399	0.00539	0.01059	0.79909
3,1	0.82713	0.00829	0.01408	0.02170	0.79544
3,3	0.93065	0.00656	0.01280	0.01102	0.79965
4,1	0.82824	0.00864	0.01538	0.02256	0.79944
4,4	0.92631	0.00665	0.01483	0.01399	0.80161

Table 4.9. IP's optimization results in binary model when PS is 30

NoR,NoD	Mean	Std Dev	Best	Worst	Initial mean
1,1	0.91655	0.00822	0.01517	0.01552	0.80452
1,2	0.91700	0.00735	0.01434	0.01546	0.80156
1,3	0.92410	0.00750	0.01164	0.01689	0.80416
1,4	0.92736	0.00766	0.01608	0.01088	0.80339
2,1	0.82694	0.00856	0.01143	0.02719	0.80426
2,2	0.92767	0.00403	0.00767	0.00812	0.80472
3,1	0.82458	0.00654	0.01338	0.01554	0.80590
3,3	0.92171	0.00370	0.00651	0.01036	0.79973
4,1	0.82628	0.00671	0.01185	0.01766	0.80486
4,4	0.92019	0.00524	0.01136	0.01531	0.80389

Table 4.10. IP's optimization results in binary model when PS is 40

NoR,NoD	Mean	Std Dev	Best	Worst	Initial mean
1,1	0.92129	0.00392	0.00863	0.00842	0.80619
1,2	0.91380	0.01007	0.02486	0.02180	0.80507
1,3	0.90882	0.00645	0.01126	0.01364	0.80556
1,4	0.91197	0.00823	0.01333	0.01647	0.80667
2,1	0.82656	0.00513	0.00801	0.01297	0.80496
2,2	0.91728	0.00417	0.00806	0.00792	0.80537
3,1	0.82066	0.00786	0.01437	0.02171	0.80422
3,3	0.91427	0.00513	0.00848	0.01231	0.80702
4,1	0.82476	0.00764	0.01278	0.01653	0.80571
4,4	0.91417	0.00479	0.00749	0.01094	0.80303

Table 4.11. IP's optimization results in binary model when PS is 50

NoR,NoD	Mean	Std Dev	Best	Worst	Initial mean
1,1	0.91235	0.00448	0.00783	0.01501	0.80503
1,2	0.91048	0.00930	0.02410	0.01904	0.80478
1,3	0.90006	0.00982	0.01397	0.02887	0.81050
1,4	0.90474	0.00804	0.01454	0.01703	0.80977
2,1	0.83208	0.00427	0.00804	0.00882	0.80359
2,2	0.91057	0.00501	0.00684	0.01286	0.81006
3,1	0.82269	0.00738	0.01189	0.01419	0.80617
3,3	0.90571	0.00475	0.00796	0.01126	0.80633
4,1	0.82416	0.00854	0.01316	0.01821	0.80754
4,4	0.90508	0.00524	0.00782	0.01316	0.80776

Table 4.12. IP's optimization results in probabilistic model when PS is 10

NoR,NoD	Mean	Std Dev	Best	Worst	Initial mean
1,1	0.75213	0.00296	0.00463	0.00638	0.62361
1,2	0.75679	0.00410	0.00768	0.00870	0.62313
1,3	0.75808	0.00428	0.00933	0.00907	0.62510
1,4	0.75857	0.00437	0.00788	0.01041	0.62535
2,2	0.74201	0.00704	0.02002	0.01222	0.62144
2,3	0.75818	0.00425	0.00982	0.01038	0.62132
2,4	0.76316	0.00461	0.00731	0.00868	0.62329
3,3	0.72876	0.00403	0.01008	0.00886	0.62795
3,4	0.75333	0.00379	0.01256	0.00683	0.62265
4,4	0.72445	0.00664	0.01463	0.01445	0.62244

Table 4.13. IP's optimization results in probabilistic model when PS is 20

NoR,NoD	Mean	Std Dev	Best	Worst	Initial mean
1,1	0.73966	0.00329	0.00516	0.00561	0.62816
1,2	0.73352	0.00668	0.01562	0.01185	0.62736
1,3	0.73757	0.00526	0.01460	0.00974	0.62725
1,4	0.74497	0.00586	0.01245	0.01006	0.62923
2,2	0.73299	0.00391	0.01093	0.00679	0.62662
2,3	0.73576	0.00531	0.00996	0.01213	0.62518
2,4	0.74706	0.00583	0.01537	0.00896	0.62804
3,3	0.72821	0.00557	0.01048	0.01295	0.62908
3,4	0.73560	0.00482	0.01116	0.00879	0.62682
4,4	0.72256	0.00553	0.01138	0.00910	0.62875

Table 4.14. IP's optimization results in probabilistic model when PS is 30

NoR,NoD	Mean	Std Dev	Best	Worst	Initial mean
1,1	0.72858	0.00316	0.00562	0.00913	0.63054
1,2	0.71762	0.00616	0.00871	0.01713	0.62943
1,3	0.72434	0.00406	0.01056	0.00776	0.63231
1,4	0.72519	0.00619	0.01343	0.01412	0.63091
2,2	0.72513	0.00301	0.00620	0.00684	0.63325
2,3	0.72064	0.00749	0.01607	0.01468	0.63118
2,4	0.73258	0.00434	0.00837	0.00839	0.63258
3,3	0.72097	0.00483	0.01135	0.01311	0.63054
3,4	0.72189	0.00465	0.00867	0.00786	0.62938
4,4	0.71868	0.00383	0.00758	0.00970	0.63084

Table 4.15. IP's optimization results in probabilistic model when PS is 40

NoR,NoD	Mean	Std Dev	Best	Worst	Initial mean
1,1	0.72098	0.00359	0.00660	0.00609	0.63273
1,2	0.71322	0.00783	0.01194	0.02110	0.63066
1,3	0.71221	0.00491	0.01082	0.01246	0.63298
1,4	0.71558	0.00484	0.00994	0.01105	0.63208
2,2	0.71769	0.00368	0.00884	0.00857	0.63260
2,3	0.71403	0.00706	0.01157	0.01689	0.63268
2,4	0.71760	0.00630	0.00975	0.01498	0.63239
3,3	0.71532	0.00387	0.00691	0.00965	0.63448
3,4	0.71342	0.00687	0.02178	0.01238	0.63392
4,4	0.71288	0.00351	0.00618	0.01020	0.63207

Table 4.16. IP's optimization results in probabilistic model when PS is 50

NoR,NoD	Mean	Std Dev	Best	Worst	Initial mean
1,1	0.71349	0.00327	0.00572	0.00940	0.63623
1,2	0.70849	0.00723	0.01104	0.02140	0.63056
1,3	0.70633	0.00673	0.01726	0.01273	0.63530
1,4	0.70478	0.00667	0.01351	0.01329	0.63651
2,2	0.71173	0.00402	0.00666	0.01220	0.63414
2,3	0.71289	0.00665	0.01250	0.01488	0.63587
2,4	0.71180	0.00603	0.01052	0.01603	0.63380
3,3	0.70964	0.00402	0.00631	0.00868	0.63232
3,4	0.71065	0.00694	0.01225	0.01170	0.63512
4,4	0.70750	0.00462	0.00710	0.01097	0.63464

Table 4.17. mIP's optimization results in binary model when PS is 10

NoR,NoD	Mean	Std Dev	Best	Worst	Initial mean
1,1	0.96822	0.00258	0.00586	0.00473	0.79695
1,2	0.96562	0.00451	0.00787	0.00938	0.78923
1,3	0.96561	0.00355	0.00639	0.00635	0.79318
1,4	0.96486	0.00407	0.00819	0.00936	0.79265
2,1	0.83297	0.00732	0.01149	0.01498	0.78821
2,2	0.96417	0.00390	0.00858	0.00907	0.79128
3,1	0.83216	0.00643	0.01205	0.00913	0.78940
3,3	0.96328	0.00466	0.00641	0.01064	0.79083
4,1	0.83109	0.00672	0.01284	0.01275	0.79416
4,4	0.95866	0.00477	0.00689	0.01419	0.79235

Table 4.18. mIP's optimization results in binary model when PS is 20

NoR,NoD	Mean	Std Dev	Best	Worst	Initial mean
1,1	0.95565	0.00453	0.01153	0.01003	0.79928
1,2	0.95331	0.00435	0.01301	0.00826	0.79937
1,3	0.95464	0.00432	0.00836	0.01173	0.80055
1,4	0.95193	0.00419	0.00565	0.01376	0.80218
2,1	0.82926	0.00728	0.01267	0.02458	0.79967
2,2	0.95206	0.00443	0.00980	0.00785	0.79898
3,1	0.82927	0.00584	0.01180	0.01172	0.79950
3,3	0.95160	0.00531	0.01139	0.00919	0.79933
4,1	0.82628	0.00637	0.01028	0.01393	0.79937
4,4	0.94923	0.00612	0.01589	0.01107	0.80118

Table 4.19. mIP's optimization results in binary model when PS is 30

NoR,NoD	Mean	Std Dev	Best	Worst	Initial mean
1,1	0.94498	0.00516	0.01036	0.01248	0.80203
1,2	0.94335	0.00633	0.02011	0.01077	0.80479
1,3	0.94476	0.00514	0.01230	0.00937	0.80216
1,4	0.94321	0.00605	0.00879	0.01091	0.80410
2,1	0.83277	0.00678	0.00873	0.01617	0.80445
2,2	0.94523	0.00563	0.00963	0.01233	0.80066
3,1	0.82846	0.00508	0.01002	0.01136	0.80373
3,3	0.94250	0.00503	0.01004	0.00927	0.80234
4,1	0.82418	0.00572	0.01386	0.01035	0.80477
4,4	0.94252	0.00587	0.01506	0.01445	0.80552

Table 4.20. mIP's optimization results in binary model when PS is 40

NoR,NoD	Mean	Std Dev	Best	Worst	Initial mean
1,1	0.93827	0.00638	0.01091	0.01282	0.80788
1,2	0.93740	0.00688	0.01749	0.01163	0.80303
1,3	0.93797	0.00669	0.01669	0.01360	0.80545
1,4	0.93748	0.00644	0.01394	0.01380	0.80325
2,1	0.83487	0.00607	0.00906	0.01711	0.80491
2,2	0.93694	0.00563	0.01360	0.01061	0.80649
3,1	0.82300	0.00755	0.01063	0.02142	0.80598
3,3	0.93707	0.00475	0.01128	0.00950	0.80423
4,1	0.82593	0.00764	0.01973	0.01615	0.80503
4,4	0.93514	0.00518	0.01033	0.01163	0.80402

Table 4.21. mIP's optimization results in binary model when PS is 50

NoR,NoD	Mean	Std Dev	Best	Worst	Initial mean
1,1	0.92971	0.00670	0.01235	0.01735	0.80709
1,2	0.93565	0.00539	0.01006	0.01269	0.80506
1,3	0.93478	0.00698	0.01301	0.01551	0.80846
1,4	0.93129	0.00701	0.01373	0.01774	0.80648
2,2	0.93296	0.00616	0.01266	0.01136	0.80758
3,3	0.92945	0.00760	0.01669	0.01497	0.80739
4,1	0.82514	0.00681	0.01228	0.01831	0.80921
4,2	0.83103	0.00729	0.01435	0.01604	0.80632
4,3	0.86041	0.00702	0.01196	0.01676	0.80842
4,4	0.92850	0.00584	0.01908	0.00935	0.80758

Table 4.22. mIP's optimization results in probabilistic model when PS is 10

NoR,NoD	Mean	Std Dev	Best	Worst	Initial mean
1,1	0.76319	0.00342	0.00717	0.00705	0.62686
1,2	0.76231	0.00504	0.01163	0.01389	0.62245
1,3	0.75882	0.00476	0.00991	0.00914	0.62518
1,4	0.75836	0.00453	0.00760	0.00881	0.62441
2,2	0.75985	0.00457	0.01061	0.00883	0.62212
2,3	0.76422	0.00335	0.00755	0.00602	0.62178
2,4	0.76516	0.00372	0.00769	0.00733	0.62280
3,3	0.75381	0.00475	0.01013	0.01006	0.62400
3,4	0.76396	0.00366	0.00877	0.00858	0.62299
4,4	0.75257	0.00493	0.00867	0.00999	0.62454

Table 4.23. mIP's optimization results in probabilistic model when PS is 20

NoR,NoD	Mean	Std Dev	Best	Worst	Initial mean
1,1	0.74939	0.00491	0.00938	0.00883	0.62569
1,2	0.74526	0.00414	0.00896	0.00979	0.62833
1,3	0.74760	0.00526	0.01447	0.00915	0.62833
1,4	0.74705	0.00487	0.01202	0.01002	0.62773
2,2	0.74662	0.00446	0.00944	0.00851	0.62807
2,3	0.75281	0.00507	0.00839	0.00940	0.62773
2,4	0.75509	0.00428	0.00797	0.00740	0.62782
3,3	0.74563	0.00447	0.00827	0.00731	0.63373
3,4	0.75220	0.00340	0.00802	0.00586	0.63213
4,4	0.74373	0.00485	0.01538	0.00993	0.62706

Table 4.24. mIP's optimization results in probabilistic model when PS is 30

NoR,NoD	Mean	Std Dev	Best	Worst	Initial mean
1,1	0.74055	0.00541	0.01122	0.01466	0.63053
1,2	0.74004	0.00484	0.00918	0.00783	0.63245
1,3	0.73961	0.00439	0.00869	0.01032	0.62864
1,4	0.73846	0.00537	0.01429	0.01081	0.63210
2,2	0.73956	0.00499	0.00832	0.01075	0.63014
2,3	0.74415	0.00501	0.00897	0.01115	0.63234
2,4	0.74795	0.00478	0.00912	0.01317	0.63040
3,3	0.73930	0.00444	0.00903	0.00765	0.62964
3,4	0.74349	0.00584	0.01355	0.01231	0.63277
4,4	0.73511	0.00430	0.00733	0.00859	0.63106

Table 4.25. mIP's optimization results in probabilistic model when PS is 40

NoR,NoD	Mean	Std Dev	Best	Worst	Initial mean
1,1	0.73640	0.00468	0.00899	0.00891	0.63511
1,2	0.73357	0.00659	0.01468	0.01252	0.63335
1,3	0.73504	0.00581	0.01368	0.01404	0.63224
1,4	0.73334	0.00651	0.01134	0.01405	0.63415
2,2	0.73367	0.00575	0.00941	0.01266	0.63149
2,3	0.73869	0.00549	0.01276	0.01155	0.63346
2,4	0.74111	0.00567	0.01337	0.01349	0.63414
3,3	0.73192	0.00463	0.00977	0.01102	0.63322
3,4	0.73966	0.00458	0.01215	0.00839	0.63295
4,4	0.73249	0.00535	0.00947	0.01412	0.63049

Table 4.26. mIP's optimization results in probabilistic model when PS is 50

NoR,NoD	Mean	Std Dev	Best	Worst	Initial mean
1,1	0.72955	0.00578	0.01100	0.01388	0.63310
1,2	0.73213	0.00462	0.00835	0.00937	0.63246
1,3	0.73352	0.00608	0.01305	0.00994	0.63379
1,4	0.72808	0.00589	0.01225	0.01084	0.63314
2,2	0.72940	0.00525	0.00808	0.01025	0.63469
2,3	0.73516	0.00498	0.01051	0.00833	0.63611
2,4	0.73522	0.00499	0.01418	0.00735	0.63451
3,3	0.72561	0.00605	0.00941	0.01109	0.63435
3,4	0.73378	0.00479	0.01016	0.00966	0.63447
4,4	0.72649	0.00489	0.01223	0.01209	0.63304

Table 4.27. Comparing the results in the binary model with other studies

	Mean		
algorithms	NoE=1000	NoE=2000	NoE=10000
ABC (Aslan, 2019a)	0.88257	0.91207	0.96755
pABC (Aslan, 2019a)	0.87507	0.90887	0.96904
tlABC (Aslan, 2019b)	-	0.92330	0.96932
p-tlABC (Aslan, 2019b)	-	0.91632	0.95702
IP	0.86637	0.89810	0.96274
mIP	0.90289	0.92902	0.96822

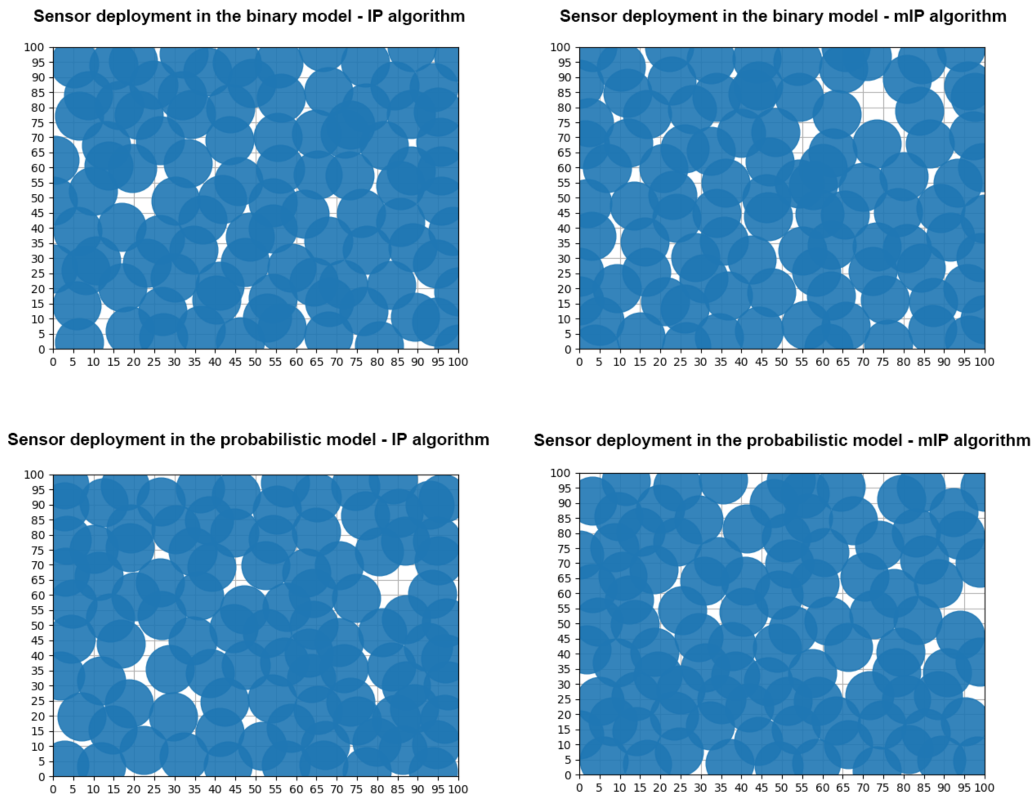


Figure 4.2. Optimizing sensors deployment with IP and mIP algorithms

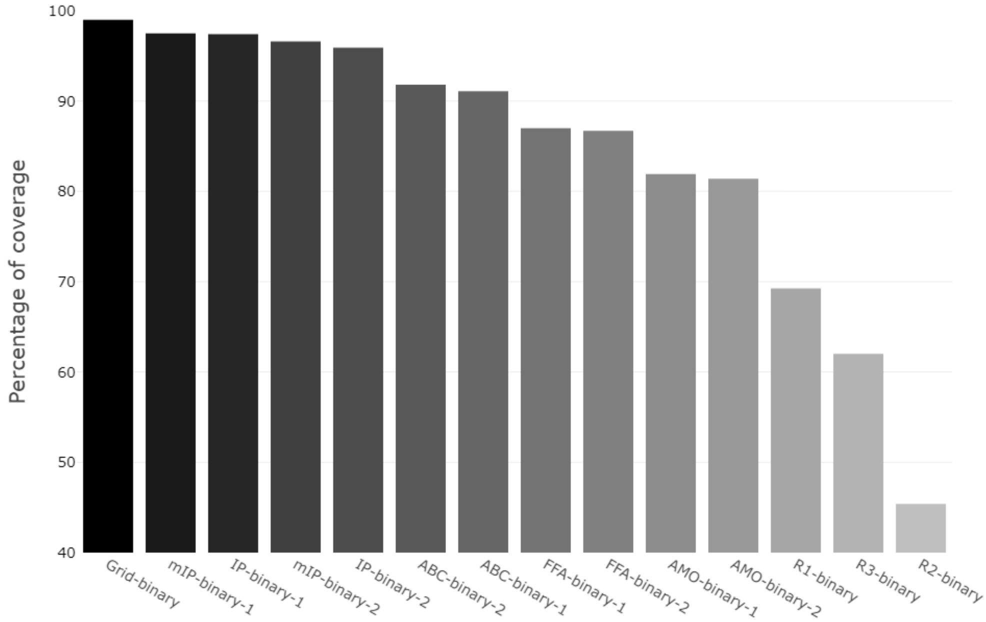


Figure 4.3. Percentage of coverage in all sensor scenarios in binary model

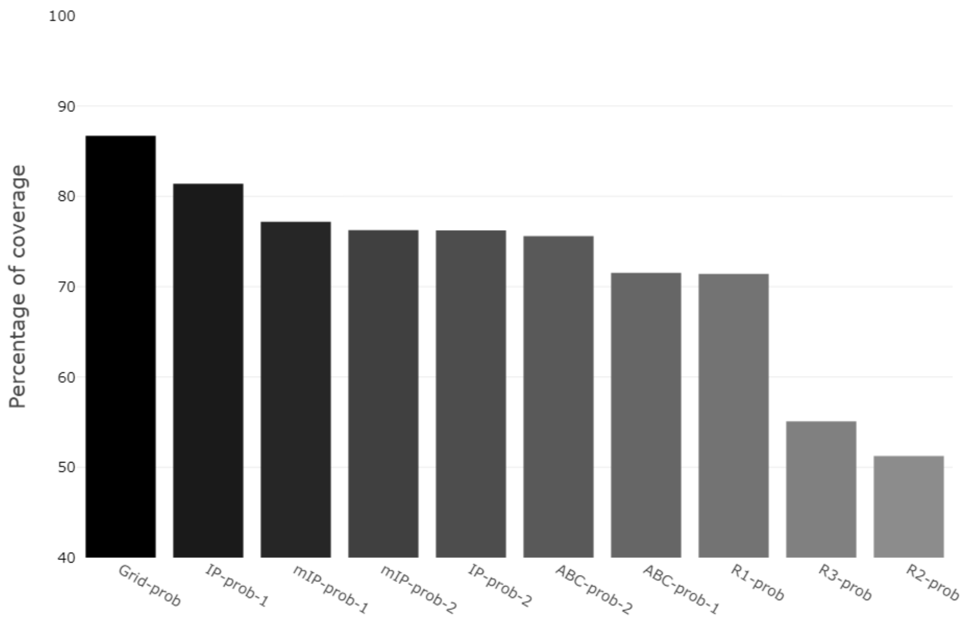


Figure 4.4. Percentage of coverage in all sensor scenarios in probabilistic model

4.2.2. Simulation Results

The impact of the IP and mIP algorithm on WSN routing protocols will be explored in this section, along with a comparison of its outcomes with those of the other algorithms listed above.

4.2.2.1. Comparison of the PDR ratio

The efficiency of a wireless network can be measured in terms of its PDR. The PDR is defined as the ratio of the number of packets successfully delivered to the number of packets sent. A high PDR indicates better network performance. Various deployments and algorithms were tested to evaluate their PDR values.

The PDR values of grid-based, R-1, and R-2 deployments obtained at random were found to be practically identical, as shown in Figure 4.5. However, R-1 and R-2 did not transmit as many packets as the grid-based deployment. In fact, the PDR values of R-1 and R-2 were similar to the grid-based deployment despite transmitting fewer packets. On the other hand, R-3 did not transmit any packets, and hence, its PDR value is 0.

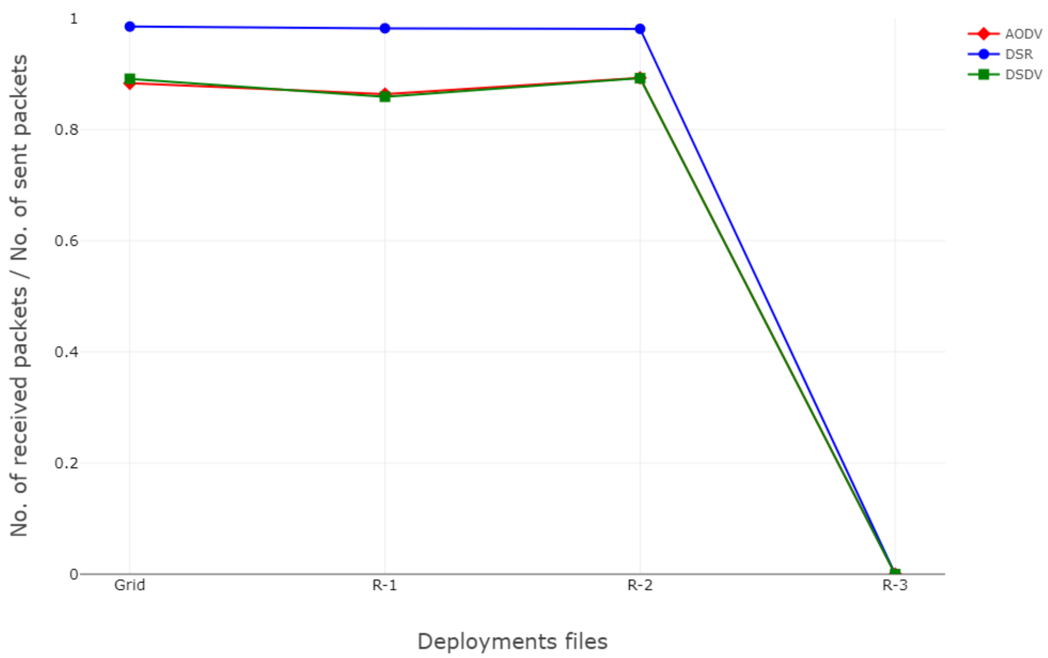


Figure 4.5. Packet delivery ratio of random deployment

In binary deployment, the PDR ratio of all algorithms was nearly identical for each routing protocol, as indicated in Figure 4.6. However, the mIP-1 algorithm outperformed other algorithms for the DSR and DSDV routing protocols. In AODV, the PDR ratio of mIP-1 was better than in some other algorithms. This implies that mIP-1 is more efficient in terms of packet delivery when compared to other algorithms for DSR, DSDV, and AODV routing protocols.

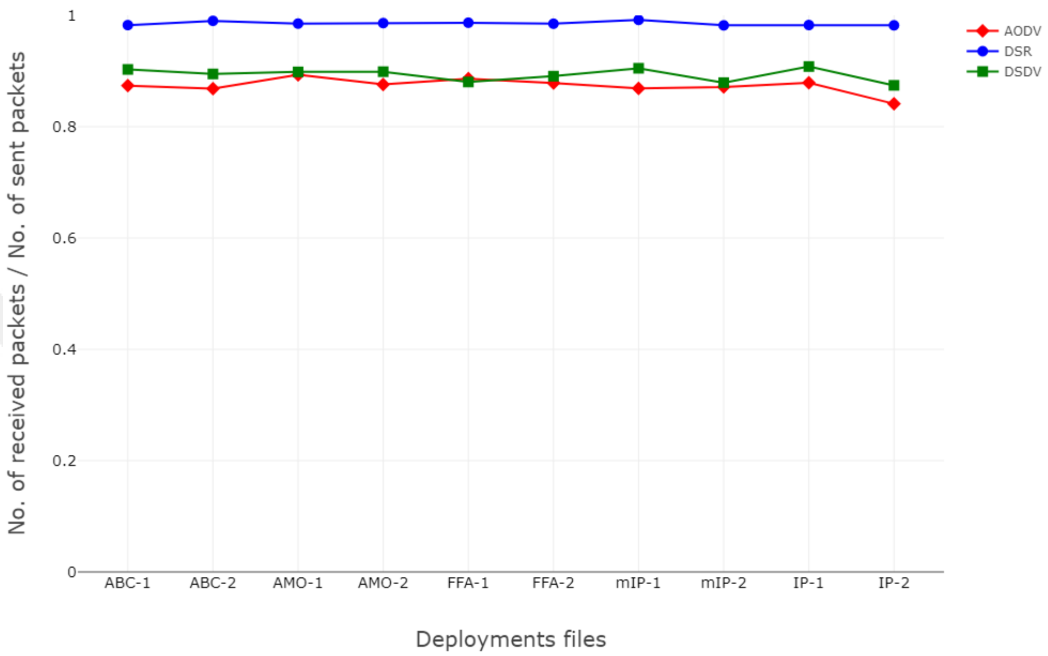


Figure 4.6. Packet delivery ratio in binary model

As shown in Figure 4.7, the PDR ratio of all algorithms was also identical in probabilistic deployment. In fact, each of the IP versions performed better than the ABC algorithm for the DSR and DSDV routing protocols, respectively. When compared to the ABC algorithm, the PDR ratio in AODV of IP had a somewhat difference of less than 0.1%. This suggests that IP algorithms, specifically IPv4 and IPv6, are generally more efficient in terms of packet delivery than the ABC algorithm.

Overall, the results indicate that IP algorithms outperform random or other algorithm deployments in terms of PDR ratio. This is because random deployments do not consider the network topology and hence are less efficient in terms of packet delivery. Similarly, other algorithm deployments may not be optimized for specific routing protocols and hence may not be the most efficient. On the other hand, IP algorithms are specifically designed for network communication and are optimized for packet delivery in various routing protocols. Hence, they are more efficient in terms

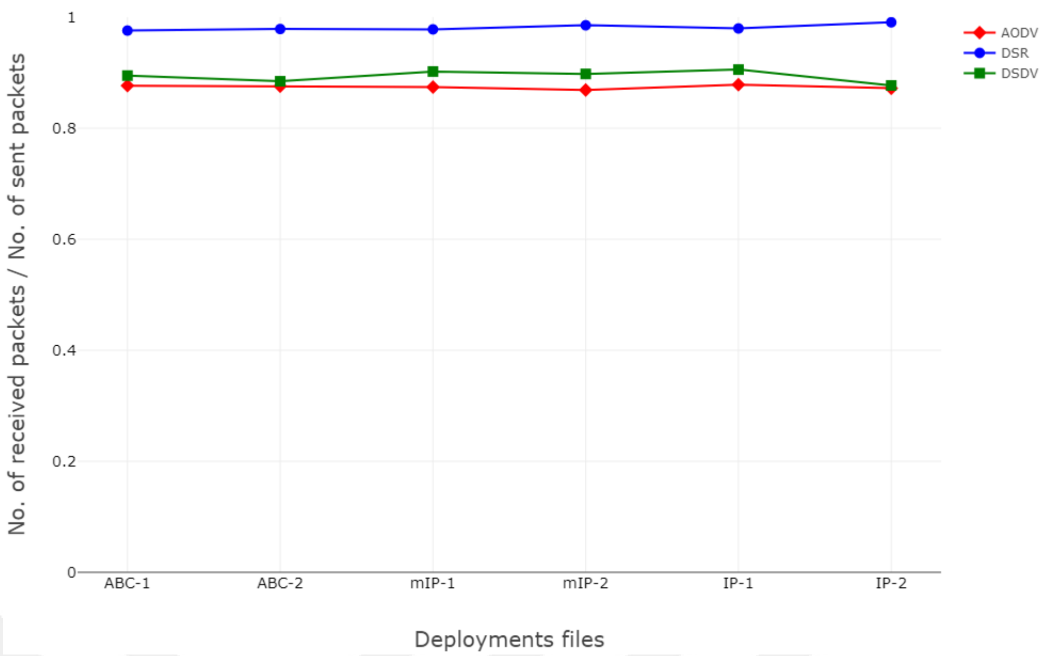


Figure 4.7. Packet delivery ratio in probabilistic model

of the PDR ratio.

It is worth noting that while the PDR ratio is an important metric to measure network efficiency, it is not the only metric. Other metrics such as latency, throughput, and energy consumption may also be important depending on the specific application.

In conclusion, the study shows that IP algorithms are more efficient in terms of packet delivery when compared to random or other algorithm deployments. The mIP-1 algorithm outperforms other algorithms for the DSR and DSDV routing protocols, and in AODV, the PDR ratio of mIP-1 is better than some other algorithms. These findings can be helpful in designing and optimizing wireless networks for various applications.

4.2.2.2. Comparison of the consumed energy

In WSNs, energy consumption is a crucial factor to consider as sensors are usually battery-powered and require a limited energy supply. The term "consumed energy" refers to the amount of energy that each sensor node expends to perform a certain task, such as transmitting or receiving data.

The consumed energy of all deployments using the mentioned algorithms in binary and probabilistic models is generally better than that of random deployments. This is shown in Figure 4.8, Figure 4.9, and Figure 4.10.. In binary deployment, the consumed energy of all algorithms in the DSR is very similar. In the DSDV, the IP algorithms consume less energy compared to other algorithms. On the other hand, in the AODV, the IP algorithms consume slightly more energy than others.

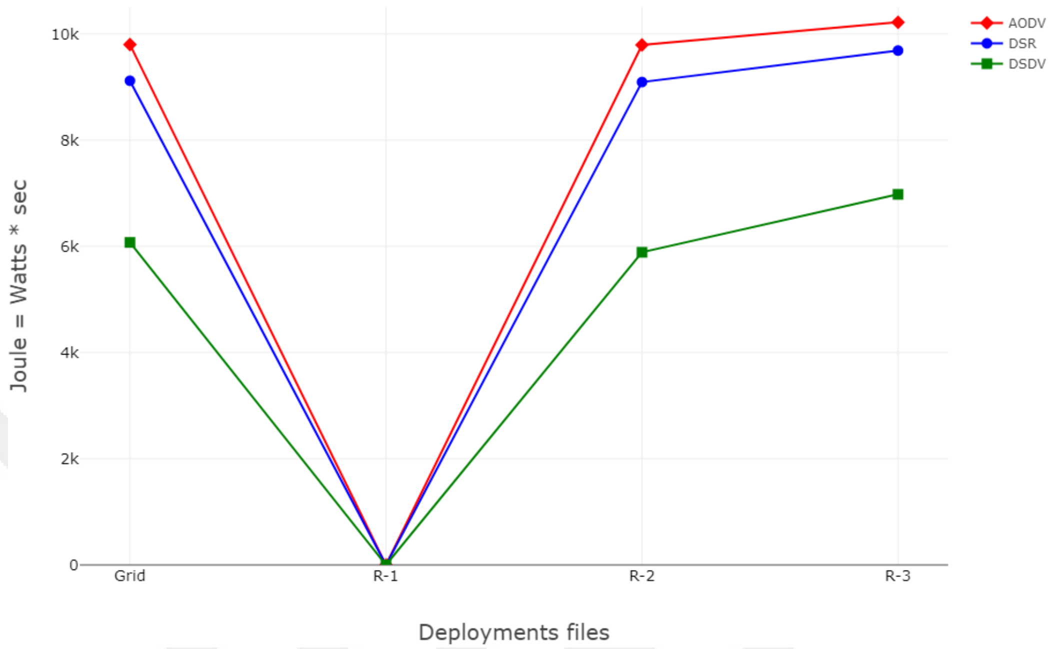


Figure 4.8. Consumed energy of random deployment

In probabilistic deployment, the consumed energy of the ABC and IP algorithms in the AODV and DSR is almost the same. However, in the DSDV, the IP algorithms consume less energy in one scenario. Therefore, the comparison of energy consumption among algorithms in different scenarios depends on the specific deployment model and routing protocol used.

Overall, minimizing energy consumption in WSNs is essential to extend the network's lifetime and ensure its proper functioning. It can be achieved through various techniques, such as optimizing routing protocols, reducing communication overhead, and adapting transmission power based on the distance between nodes.

4.2.2.3. Comparison of the average throughput

The average throughput of a WSN refers to the amount of data transmitted through the network over a certain period of time. It is a measure of the network's ability to effectively transmit data packets.

The performance of various algorithms for WSNs was evaluated in terms of their

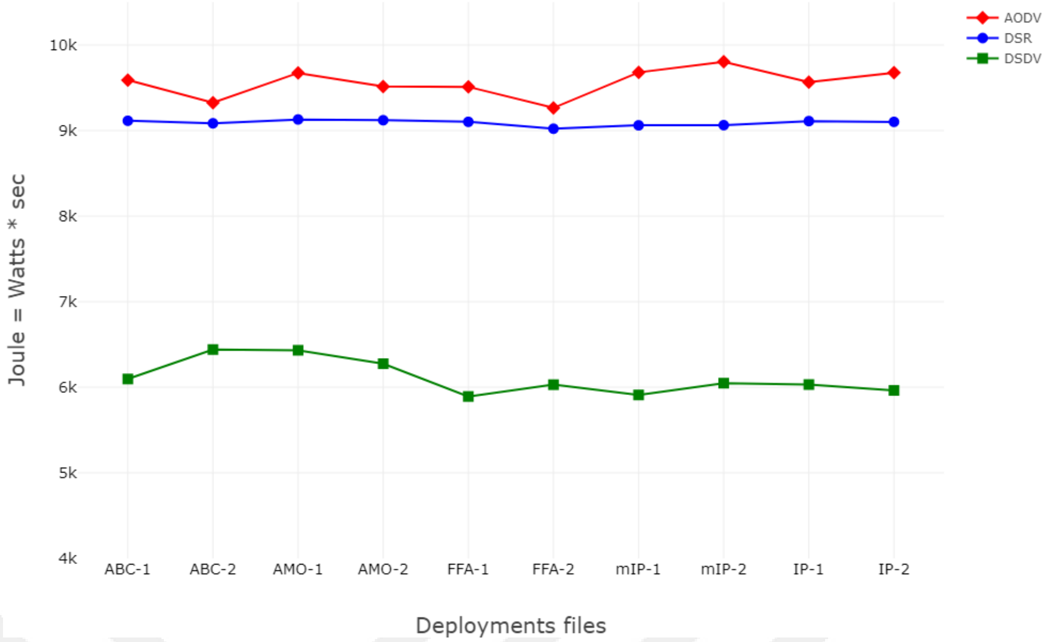


Figure 4.9. Consumed energy in binary model

average throughput. The results, presented in Figure 4.11, Figure 4.12, and Figure 4.13, show that IP algorithms generally perform as well as or better than other algorithms and random deployments in both DSDV and DSR protocols. However, in the case of AODV, the performance of IP algorithms is sometimes better than random deployment, but sometimes inferior to other algorithm deployments.

Overall, the results suggest that IP algorithms can be a reliable and efficient choice for WSNs, particularly in DSDV and DSR protocols. However, in the case of AODV, the choice of algorithm deployment may need to be carefully considered based on the specific requirements of the network.

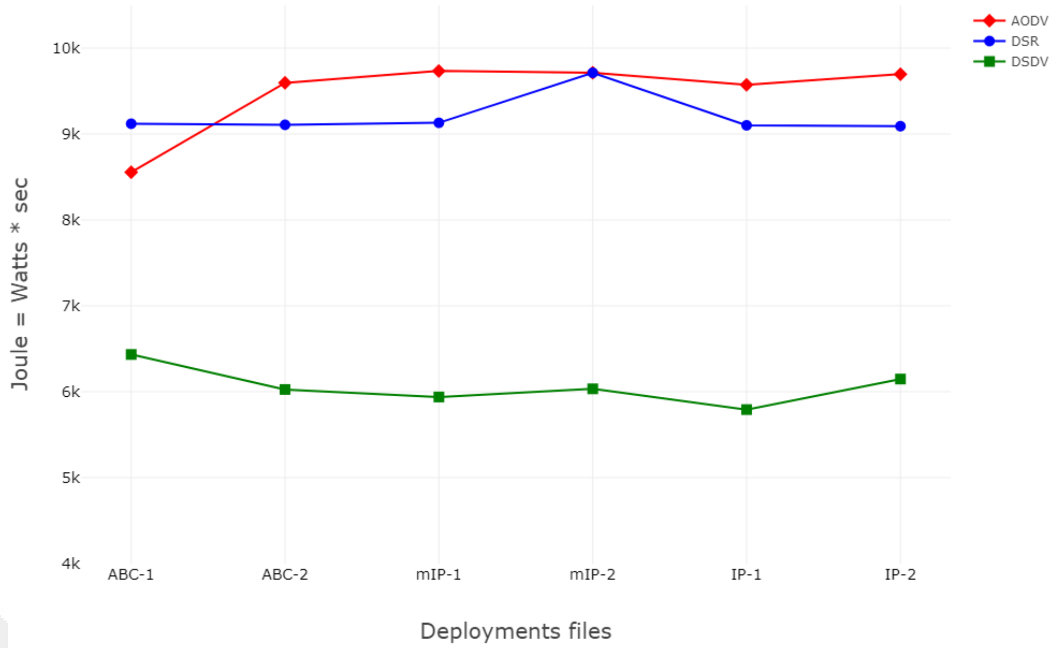


Figure 4.10. Consumed energy in probabilistic model

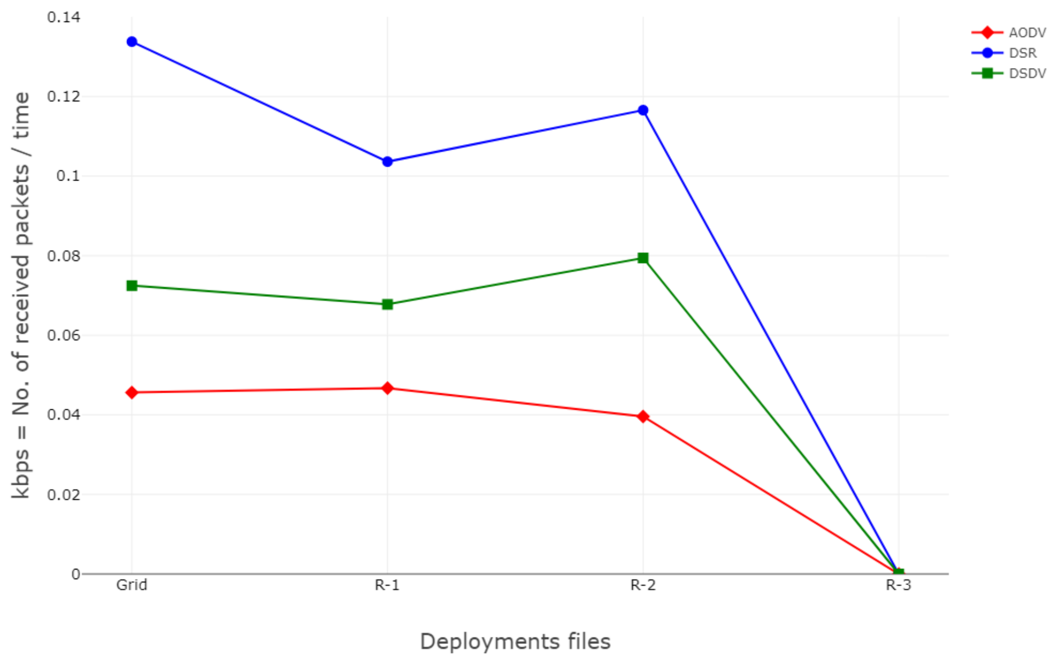


Figure 4.11. Average throughput of random deployment

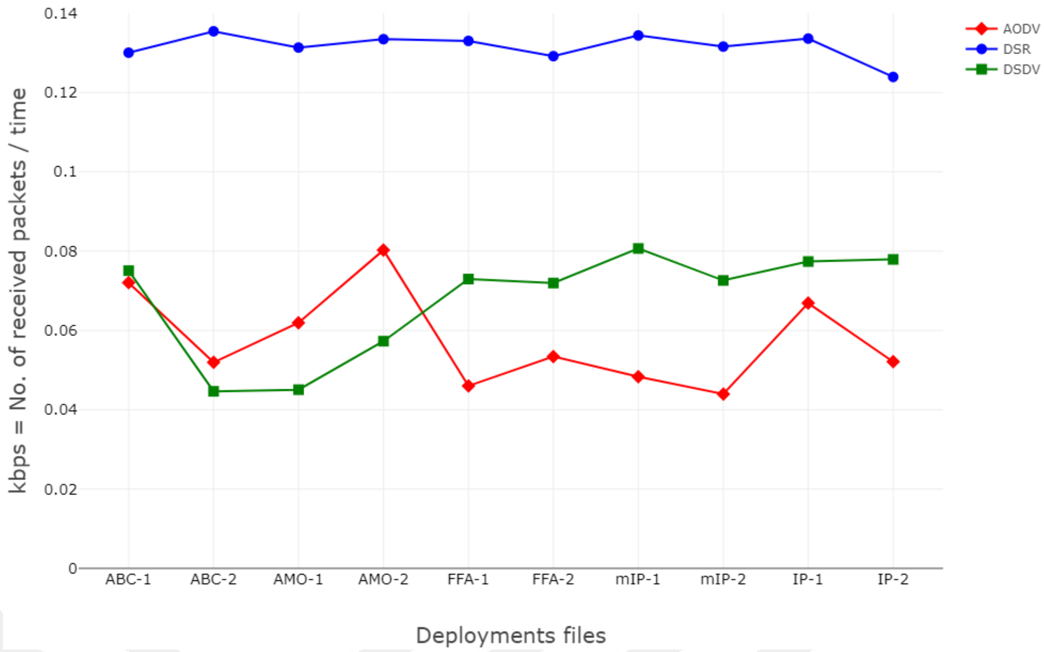


Figure 4.12. Average throughput in binary model

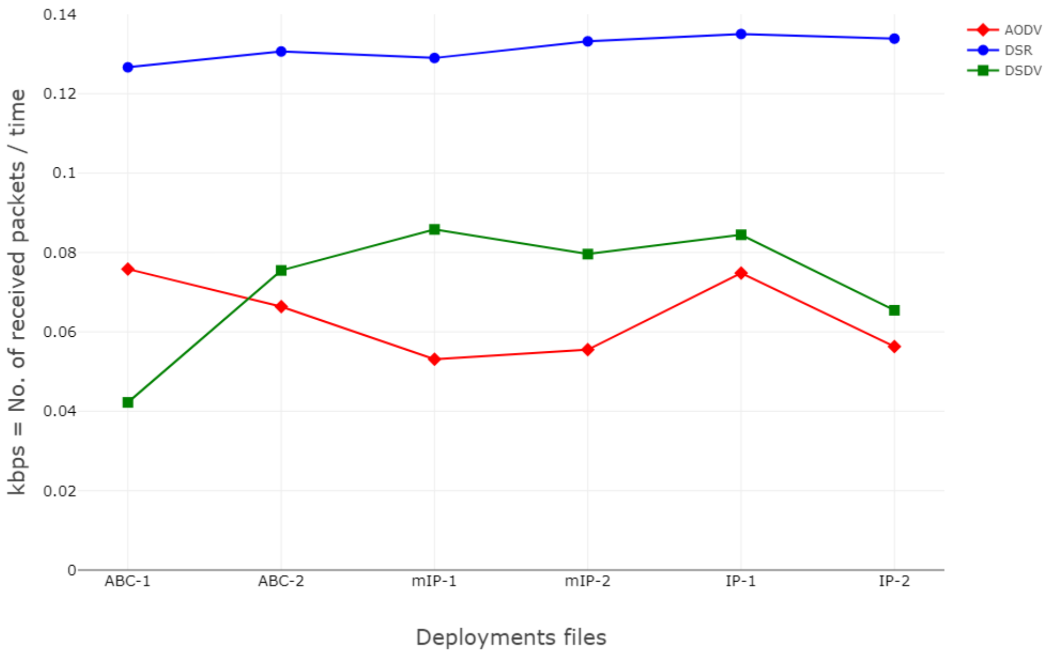


Figure 4.13. Average throughput in probabilistic model

5. CONCLUSION AND RECOMMENDATIONS

5.1. Conclusion

This thesis highlights the crucial role of WSNs in a variety of applications and the challenge of effectively deploying sensor nodes for optimal performance. The thesis focuses on addressing the coverage area problem in WSNs using two versions of the immune plasma algorithm (IP), a meta-heuristic algorithm that transfers plasma or antibodies from one patient to another.

The thesis discusses binary and probabilistic detection models and explores several scenarios using IP, mIP, and other algorithms. The results obtained after optimization were incorporated into the NS-2 simulation tool, and the IP algorithm's performance was compared to other algorithms using various metrics in multiple routing algorithms.

The study reveals that the IP algorithms outperformed all other algorithms in enhancing coverage without causing any detrimental effects on the simulation performance of WSNs. Although the IP algorithm's performance was sometimes equal to or better than other algorithms, it consistently provided better coverage area results.

The research methodology included simulations of several sensor deployment files on three routing protocols, and various metrics were compared. The thesis also highlights the importance of correctly deploying sensor nodes for optimal performance and addresses the challenge of coverage area in WSNs.

The thesis analyzed the performance of the IP and mIP algorithms in solving the sensor deployment problem. The results showed that using the IP algorithm resulted in a clear improvement in sensor coverage, with an average increase of 10 to 16 percent higher than primitive coverage for some scenarios. Similarly, the results of the probabilistic model indicated an increase in the general coverage of sensors by 8 to 12 percent.

Also, results showed that the mIP algorithm also improved general coverage in the binary model, showing improvements of 13 to 17 percent higher than initial coverage. In the probabilistic model, some scenarios improved by 10 to 14 percent. In addition, the simulation results for both IP and mIP were not only non-negative but also exhibited better performance in terms of PDR, consumed energy, and average throughput when compared to other algorithms in the DSDV, AODV, and DSR routing protocols.

In conclusion, the thesis provides valuable insights into the use of the IP algorithm for addressing coverage area problems in WSNs. The results obtained from the study can assist in the development of more efficient algorithms for improving coverage in WSNs. The study emphasizes the need for proper deployment of sensor nodes to achieve optimal performance and encourages further research in this area to improve WSNs' overall performance.

5.2. Recommendations

In future works, more experiments can be done using various parameters, different models can be used to calculate sensor coverage area and apply it to irregular or three-dimensional spaces, and improve the IP algorithm to get better results than the results mentioned above. Additionally, other routing protocols other than the ones now in use may be employed. These protocols may also be compared using additional metrics. Also, there are several simulation tools available than the one being utilized which can be used.

REFERENCES

Abdali, T.-A. N., Hassan, R., Muniyandi, R. C., Mohd Aman, A. H., Nguyen, Q. N., & Al-Khaleefa, A. S. (2020). Optimized particle swarm optimization algorithm for the realization of an enhanced energy-aware location-aided routing protocol in manet. *Information*, *11*(11), 529.

Abdullah, A. M., & Aziz, R. H. H. (2014). The impact of reactive routing protocols for transferring multimedia data over manet. *Journal of Zankoy Sulaimani-Part A*, *16*(4), 9–24.

Adam, S. P., Alexandropoulos, S.-A. N., Pardalos, P. M., & Vrahatis, M. N. (2019). No free lunch theorem: A review. *Approximation and optimization*, 57–82.

Aggarwal, A., Gandhi, S., & Chaubey, N. (2014). Performance analysis of aodv, dsdv and dsr in manets. *arXiv preprint arXiv:1402.2217*.

AL-Dhief, F. T., Sabri, N., Salim, M., Fouad, S., & Aljunid, S. (2018). Manet routing protocols evaluation: Aodv, dsr and dsdv perspective. In *Matec web of conferences* (Vol. 150, p. 06024).

Aliwi, M., Aslan, S., & Demirci, S. (2020). Solving uav localization problem with firefly algorithm. In *2020 28th signal processing and communications applications conference (siu)* (pp. 1–4).

Aljubori, M. H. H., Khalilpour Akram, V., & Challenger, M. (2022). Improving the deployment of wsns by localized detection of covered redundant nodes in industry 4.0 applications. *Sensors*, *22*(3). Retrieved from `https://www.mdpi.com/1424-8220/22/3/942` doi: 10.3390/s22030942

Al-Mousawi, A. J. (2020). Evolutionary intelligence in wireless sensor network: routing, clustering, localization and coverage. *Wireless Networks*, *26*(8), 5595–5621.

Amutha, J., Sharma, S., & Nagar, J. (2020). Wsn strategies based on sensors, deployment, sensing models, coverage and energy efficiency: Review, approaches and open issues. *Wireless Personal Communications*, *111*(2), 1089–1115.

Aslan, S. (2019a). Deployment in wireless sensor networks by parallel and cooperative parallel artificial bee colony algorithms. *An International Journal of Optimization and Control: Theories & Applications (IJOCTA)*, *9*(1), 1–10.

Aslan, S. (2019b). A transition control mechanism for artificial bee colony (abc) algorithm. *Computational Intelligence and Neuroscience*, *2019*.

Aslan, S., Aksoy, A., & GUNAY, M. (2018). Performance of parallel artificial bee colony algorithm on solving probabilistic sensor deployment problem. In *2018 international conference on artificial intelligence and data processing (idap)* (pp. 1–5).

Aslan, S., & Demirci, S. (2019). Solving uav localization problem with artificial bee colony (abc) algorithm. In *2019 4th international conference on computer science and engineering (ubmk)* (pp. 735–738).

Aslan, S., & Demirci, S. (2020). Immune plasma algorithm: a novel meta-heuristic for optimization problems. *Ieee Access*, *8*, 220227–220245.

Barati, M., Atefi, K., Khosravi, F., & Daftari, Y. A. (2012). Performance evaluation of energy consumption for aodv and dsr routing protocols in manet. In *2012 international conference on computer & information science (iccis)* (Vol. 2, pp. 636–642).

Ben Amor, O., Chelly Dagdia, Z., Bechikh, S., & Ben Said, L. (2022). Many-objective optimization of wireless sensor network deployment. *Evolutionary Intelligence*, 1–17.

Bhat, S. J., & KV, S. (2022). A localization and deployment model for wireless sensor networks using arithmetic optimization algorithm. *Peer-to-Peer Networking and Applications*, *15*(3), 1473–1485.

Bouras, C., Charalambides, S., Kioumourtzis, G., & Stamos, K. (2012). A tool for enhancing simulation trace files processing.

Carter, B., & Ragade, R. (2009). A probabilistic model for the deployment of sensors. In *2009 ieee sensors applications symposium* (pp. 7–12).

Chandren Muniyandi, R., Hasan, M. K., Hammoodi, M. R., & Maroosi, A. (2021). An improved harmony search algorithm for proactive routing protocol in vanet. *Journal of Advanced Transportation*, *2021*.

Delves, P. J., & Roitt, I. M. (2000). The immune system. *New England journal of medicine*, *343*(1), 37–49.

Domga, R. K., Stanica, R., Tchuente, M., & Valois, F. (2019). Sensor deployment in wireless sensor networks with linear topology using virtual node concept. *Wireless Networks*, *25*(8), 4947–4962.

Duraki, S., Aslan, S., & Demirci, S. (2021). Performance analysis of immune plasma algorithm with different donor-receiver configurations. *Avrupa Bilim ve Teknoloji Dergisi*(29), 259–263.

Enayatifar, R., Yousefi, M., Abdullah, A. H., & Darus, A. N. (2014). A novel sensor deployment approach using multi-objective imperialist competitive algorithm in wireless sensor networks. *Arabian Journal for Science and Engineering*, *39*(6), 4637–4650.

Estrin, D., Handley, M., Heidemann, J., McCanne, S., Xu, Y., & Yu, H. (2000). Network visualization with nam, the vint network animator. *Computer*, *33*(11), 63–68.

Golzari, M., & Hosseinkhani, J. (2017). Providing an effective strategy in order to proportional distribution nodes meanwhile energy saving network and appropriate coverage in sensor network. *IJCSNS*, *10*(12), 1.

Gupta, G. P., & Jha, S. (2019). Biogeography-based optimization scheme for solving the coverage and connected node placement problem for wireless sensor networks. *Wireless Networks*, *25*(6), 3167–3177.

Gupta, P., & Verma, S. (n.d.). A comparitive performance study of dsdv, aodv and dsr manet protocols using ns2. 34.

Horane, A., Patkar, S., & Jaykumar, A. (2013). Energy optimised routing protocol for manet. *nternational Journal of Emerging Trends in Electrical and Electronics*, *5*, 33–36.

Huang, Y., Zhang, J., Wei, W., Qin, T., Fan, Y., Luo, X., & Yang, J. (2022). Research on coverage optimization in a wsn based on an improved coot bird algorithm. *Sensors*, *22*(9). Retrieved from https://www.mdpi.com/1424-8220/22/9/3383 doi: 10.3390/s22093383

Hung, I. F., To, K. K., Lee, C.-K., Lee, K.-L., Chan, K., Yan, W.-W., ... others (2011). Convalescent plasma treatment reduced mortality in patients with severe pandemic influenza a (h1n1) 2009 virus infection. *Clinical Infectious Diseases*, *52*(4), 447–456.

Huynh Thi Thanh, B., Hanh, N. T., Van Quan, L., & Uyen, T. T. (n.d.). Lifetime maximization for target coverage problem in mobile wireless sensor networks. *Nguyen Thi and Van Quan, La and Uyen, Tran Thi, Lifetime Maximization for Target Coverage Problem in Mobile Wireless Sensor Networks*.

Issariyakul, T., & Hossain, E. (2009). Introduction to network simulator 2 (ns2). In *Introduction to network simulator ns2* (pp. 1–18). Springer.

Jain, A., & Dixit, P. (n.d.). Route optimization to make the ad-hoc network efficient using pso algorithm.

Jain, J., Gupta, R., & Bandhopadhyay, T. (2011). Ant colony algorithm in manet-local link repairing of aodv. In *2011 3rd international conference on electronics computer technology* (Vol. 6, pp. 270–273).

Juneja, S., Kaur, K., & Singh, H. (2022). An intelligent coverage optimization and link-stability routing for energy efficient wireless sensor network. *Wireless Networks*, *28*(2), 705–719.

Khan, M. F., Felemban, E. A., Qaisar, S., & Ali, S. (2013). Performance analysis on packet delivery ratio and end-to-end delay of different network topologies in wireless sensor networks (wsns). In *2013 ieee 9th international conference on mobile ad-hoc and sensor networks* (pp. 324–329).

Khoufi, I., Minet, P., Laouiti, A., & Mahfoudh, S. (2017). Survey of Deployment Algorithms in Wireless Sensor Networks: Coverage and Connectivity Issues and Challenges. *International journal of autonomous and adaptive communications systems*, *10*(4), 341-390. Retrieved from https://hal.inria.fr/hal-01095749 doi: 10.1504/IJAACS.2017.10009671

Kong, F.-S., & Cui, B.-B. (2017). Performance evaluation of aodv, dsr and dsdv in mobile ad-hoc network using ns-2. In *Itm web of conferences* (Vol. 12, p. 04007).

Kukunuru, N., Thella, B. R., & Davuluri, R. L. (2010). Sensor deployment using particle swarm optimization. *International Journal of Engineering Science and Technology*, *2*(10), 5395–5401.

Kuriakose, J., Joshi, S., Vikram Raju, R., & Kilaru, A. (2014). A review on localization in wireless sensor networks. *Advances in signal processing and intelligent recognition systems*, 599–610.

Lange, K. (2013). *Optimization* (Vol. 95). Springer Science & Business Media.

Li, Q., & Liu, N. (2021). Nodes deployment algorithm based on data fusion and evidence theory in wireless sensor networks. *Wireless Personal Communications*, *116*(2), 1481–1492.

Li, Q., Liu, N., et al. (2022). Monitoring area coverage based on control multimedia nodes position in mixed underwater mobile wireless sensor networks. *Advances in Multimedia*, *2022*.

Liang, F., Yuan, F., & Liu, X. (2022). User position-based wireless sensor network deployment algorithm. In *2022 ieee international conferences on internet of things (ithings) and ieee green computing & communications (greencom) and ieee cyber, physical & social computing (cpscom) and ieee smart data (smartdata) and ieee congress on cybermatics (cybermatics)* (pp. 70–76).

Liu, J., & Sun, G. (2022). A deployment strategy of nodes in wsn based on "x" partition. *Journal of Sensors*, *2022*.

Liu, W., Yang, S., Sun, S., & Wei, S. (2018). A node deployment optimization method of wsn based on ant-lion optimization algorithm. In *2018 ieee 4th international symposium on wireless systems within the international conferences on intelligent data acquisition and advanced computing systems (idaacs-sws)* (p. 88-92). doi: 10.1109/IDAACS-SWS.2018.8525824

Mahdi, H. F., Abood, M. S., & Hamdi, M. M. (2021). Performance evaluation for vehicular ad-hoc networks based routing protocols. *Bulletin of Electrical Engineering and Informatics*, *10*(2), 1080–1091.

MHMOOD, A., & Zengin, A. (2021). Performance evaluation of manet routing protocols aodv and dsdv using ns2 simulator. *Sakarya University Journal of Computer and Information Sciences*, *4*(1), 1–10.

Mohapatra, S., & Kanungo, P. (2012). Performance analysis of aodv, dsr, olsr and dsdv routing protocols using ns2 simulator. *Procedia Engineering*, *30*, 69–76.

NAGAR, S. A., NOR, S. M., & BOBA, M. S. (2014). Generation of real world traffic using ns2 traffic agents. *Journal of Theoretical & Applied Information Technology*, *67*(3).

Ni, Q., Du, H., Pan, Q., Cao, C., & Zhai, Y. (2017). An improved dynamic deployment method for wireless sensor network based on multi-swarm particle swarm optimization. *Natural Computing*, *16*(1), 5–13.

Njoya, A. N., Ari, A. A. A., Nana Awa, M., Titouna, C., Labraoui, N., Effa, J. Y., . . . Gueroui, A. (2020). Hybrid wireless sensors deployment scheme with connectivity and coverage maintaining in wireless sensor networks. *Wireless Personal Communications*, *112*(3), 1893–1917.

Ozturk, C., Karaboga, D., & Gorkemli, B. (2011). Probabilistic dynamic deployment of wireless sensor networks by artificial bee colony algorithm. *sensors*, *11*(6), 6056–6065.

Parkin, J., & Cohen, B. (2001). An overview of the immune system. *The Lancet*, *357*(9270), 1777–1789.

Perkins, C., Belding-Royer, E., & Das, S. (2003). *Ad hoc on-demand distance vector (aodv) routing* (Tech. Rep.).

Rao, G. S., Jagadeeswararao, E., Priyanka, U. J., & Darsini, T. I. P. (2015). Performance analysis of manet routing protocols-dsdv, dsr, aodv, aomdv using ns-2. *Global Journal of Computer Science and Technology*.

Saba, A., Michelle Liu, X., & Mansourvar, M. (2022). Application of meta-heuristic algorithms in intrusion detection system. *International Journal of Nonlinear Analysis and Applications*, *13*(2), 2517–2540.

Saheb, S. I., Khan, K. U. R., & Bindu, C. S. (n.d.). A multi-objective sensor deployment using bat algorithm to improve coverage, lifetime and energy consumption in wireless sensor networks.

Si, P., Ma, J., Tao, F., Fu, Z., & Shu, L. (2020). Energy-efficient barrier coverage with probabilistic sensors in wireless sensor networks. *IEEE Sensors Journal*, *20*(10), 5624–5633.

Singh, R., Singh, A., Singh, B., & Singla, V. (2019). Comparative analysis and enhancement in aodv protocol for path establishment using swarm intelligence in manets. *Think India Journal*, *22*(16), 2194–2200.

Siraj, S., Gupta, A., & Badgujar, R. (2012). Network simulation tools survey. *International Journal of Advanced Research in Computer and Communication Engineering*, *1*(4), 199–206.

Soomro, A. M., Fudzee, M., Hussain, M., Saim, H. M., Zaman, G., Atta-ur Rahman, H., & Nabil, M. (2022). Comparative review of routing protocols in manet for future research in disaster management. *Journal of Communications*, *17*(9).

Thottempudi, S. P., & Umar, S. (2013). A review of ant colony based routing algorithm in wireless ad-hoc networks. *International Journal of Science, Engineering and Computer Technology*, *3*(11), 396.

Wang, L., Wu, W., Qi, J., & Jia, Z. (2018). Wireless sensor network coverage optimization based on whale group algorithm. *Computer Science and Information Systems*, *15*(3), 569–583.

Wang, X., & Wang, S. (2010). Hierarchical deployment optimization for wireless sensor networks. *IEEE Transactions on Mobile Computing*, *10*(7), 1028–1041.

Wang, Z., Xie, H., He, D., & Chan, S. (2019). Wireless sensor network deployment optimization based on two flower pollination algorithms. *IEEE Access*, *7*, 180590–180608.

Wu, J., Li, H., Luo, L., Ou, J., & Zhang, Y. (2022). Multiobjective optimization strategy of wsn coverage based on ipso-ircd. *Journal of Sensors*, *2022*.

Wu, X., Zhang, S., Xiao, W., & Yin, Y. (2019). The exploration/exploitation tradeoff in whale optimization algorithm. *IEEE Access*, *7*, 125919–125928.

Xie, J., & Zhang, C. (2022). An improved sparrow search algorithm for deployment problem in wireless sensor network.

Yarinezhad, R., & Hashemi, S. N. (2020). A sensor deployment approach for target coverage problem in wireless sensor networks. *Journal of Ambient Intelligence and Humanized Computing*, 1–16.

Younis, Z. A., Abdulazeez, A. M., Zeebaree, S. R., Zebari, R. R., & Zeebaree, D. Q.

(2021). Mobile ad hoc network in disaster area network scenario: A review on routing protocols. *International Journal of Online & Biomedical Engineering*, *17*(3).

Zaimen, K., Dollinger, J.-F., Moalic, L., Abouaissa, A., Idoumghar, L., et al. (2020). A overview on wsn deployment and a novel conceptual bim-based approach in smart buildings. In *2020 7th international conference on internet of things: Systems, management and security (iotsms)* (pp. 1–6).

Zou, Y., & Chakrabarty, K. (2003). Sensor deployment and target localization based on virtual forces. In *Ieee infocom 2003. twenty-second annual joint conference of the ieee computer and communications societies (ieee cat. no. 03ch37428)* (Vol. 2, pp. 1293–1303).

APPENDICES

Appendix A. Routing Protocols Metrics Results For All Deployment Scenarios

Table A.1. Routing protocols metrics results for all deployment scenarios

Deployment	*Consumed energy*	*Throughput*	*Packet delivery ratio*
ABC-binary-1 AODV	9590.114859	0.072047	0.873678
ABC-binary-1 DSR	9114.083297	0.130045	0.982443
ABC-binary-1 DSR	6096.278633	0.075083	0.902815
ABC-binary-2 AODV	9325.41818	0.051947	0.868277
ABC-binary-2 DSR	9084.595029	0.135439	0.990103
ABC-binary-2 DSR	6440.487732	0.044645	0.894669
AMO-binary-1 AODV	9672.924346	0.061921	0.893156
AMO-binary-1 DSR	9129.438256	0.131335	0.985145
AMO-binary-1 DSR	6431.812745	0.045059	0.898338
AMO-binary-2 AODV	9514.367723	0.080266	0.875793
AMO-binary-2 DSR	9120.802537	0.13345	0.986061
AMO-binary-2 DSR	6274.897134	0.057289	0.898741
FFA-binary-1 AODV	9510.937276	0.046012	0.886282
FFA-binary-1 DSR	9101.599912	0.13302	0.986724
FFA-binary-1 DSR	5891.981376	0.072959	0.880279
FFA-binary-2 AODV	9264.712817	0.05342	0.87844
FFA-binary-2 DSR	9021.218006	0.129165	0.984915
FFA-binary-2 DSR	6029.716073	0.071951	0.8907
mIP-binary-1 AODV	9679.847806	0.048315	0.868852
mIP-binary-1 DSR	9061.005453	0.134395	0.991818
mIP-binary-1 DSR	5909.85512	0.080667	0.904899
mIP-binary-2 AODV	9803.607627	0.04395	0.871278
mIP-binary-2 DSR	9062.721486	0.131579	0.981975
mIP-binary-2 DSR	6047.413777	0.072603	0.879082
IP-binary-1 AODV	9566.809364	0.066882	0.878961
IP-binary-1 DSR	9110.324976	0.133598	0.982556
IP-binary-1 DSR	6031.291539	0.07738	0.908029
IP-binary-2 AODV	9676.203113	0.052109	0.841121
IP-binary-2 DSR	9101.432526	0.123928	0.982034
IP-binary-2 DSR	5963.006005	0.077915	0.874234
ABC-prob-1 AODV	8554.802311	0.07583	0.876663
ABC-prob-1 DSR	9119.183446	0.12665	0.9758
ABC-prob-1 DSR	6433.668674	0.042234	0.894701
ABC-prob-2 AODV	9595.1147	0.066366	0.875717
ABC-prob-2 DSR	9106.363578	0.130637	0.978608
ABC-prob-2 DSR	6026.165528	0.07548	0.8845
mIP-prob-1 AODV	9733.007526	0.053129	0.874003
mIP-prob-1 DSR	9129.83612	0.128996	0.97799

Table A.1. Routing protocols metrics results for all deployment scenarios

Deployment	*Consumed energy*	*Throughput*	*Packet delivery ratio*
mIP-prob-1 DSR	5938.797643	0.085809	0.90206
mIP-prob-2 AODV	9711.627301	0.055516	0.86884
mIP-prob-2 DSR	9100.509982	0.13319	0.985347
mIP-prob-2 DSR	6033.717211	0.079599	0.897723
IP-prob-1 AODV	9571.587995	0.074831	0.878409
IP-prob-1 DSR	9090.406069	0.135038	0.979642
IP-prob-1 DSR	5792.184896	0.08446	0.905563
IP-prob-2 AODV	9695.696407	0.056313	0.87191
IP-prob-2 DSR	9069.217195	0.133873	0.990681
IP-prob-2 DSR	6147.779651	0.065439	0.876989
Grid-binary AODV	9801.235871	0.045632	0.883068
Grid-binary DSR	9118.626719	0.133783	0.985027
Grid-binary DSDV	6074.163988	0.072465	0.891038
R1-binary AODV	0.046701	0.046702	0.863393
R1-binary DSR	0.10363	0.10363	0.981917
R1-binary DSDV	0.067752	0.067752	0.858828
R2-binary AODV	9793.876972	0.039577	0.892896
R2-binary DSR	9094.482797	0.11657	0.98091
R2-binary DSDV	5888.219829	0.079425	0.892136
R3-binary AODV	10221.0824	0	0
R3-binary DSR	9687.710534	0	0
R3-binary DSDV	6979.465696	0	0

CURRICULUM VITAE

Adnan Taşdemir graduated in 2019 as a computer engineer from Engineering Faculty / Ondokuz Mayıs University. Started master program in 2019 at the Institute of Graduate Studies / Ondokuz Mayıs University. Speaks Arabic (native), English and Turkish. Working as a senior developer at FLO Magazacılık. (August 2021).

Contact Information :

ORCID NO :0000-0002-5256-1120

Publications :

1. Duraki, S., Mehrat, A., and Demirci, S. (2019, November). A Mobile Application for Wireless Attendance System. In 2019 1st international informatics and software engineering conference (UBMYK) (pp. 1-6). IEEE.

2. Taşdemir, A., Demirci, S., and Aslan, S. (2022, March). Performance Investigation of Immune Plasma Algorithm on Solving Wireless Sensor Deployment Problem. In 2022 9th International Conference on Electrical and Electronics Engineering (ICEEE) (pp. 296-300). IEEE.