Norwegian Defence Research Establishment

23/00688

FFI-RAPPORT

Using genetic algorithms with variable-length chromosomes in radar detector scan schedule optimization

Lars Sandnes

Using genetic algorithms with variable-length chromosomes in radar detector scan schedule optimization

Lars Sandnes

Norwegian Defence Research Establishment (FFI) 10 March 2023

Keywords
Radar
Maskinlæring
Genetiske algoritmer
Elektronisk krigføring

FFI report
23/00688

Project number
1557

Electronic ISBN
978-82-464-3472-8

Approvers
Tore Ulversøy, *Research Manager*
Tor-Odd Høydal, *Director of Research*

The document is electronically approved and therefore has no handwritten signature.

Summary

The process of designing a radar detector scan schedule is inherently one of compromise and problem specific prioritization. Tuning such a schedule to the problem at hand tends to happen iteratively. This is characterized by continuous testing and evaluation against some acceptable performance benchmark. This project explored whether genetic algorithms with variable-length chromosomes (GA-VLC) could be used for generating such schedules. This optimization scheme is inspired by the process of evolution by natural selection. Given a database of known radar systems, the aim was to generate scan schedules capable of detecting as many of these systems as possible when placed in a radar environment described by the database in question. Achieving this required the design of a fitness function capable of assigning a figure of merit to a given schedule. This figure would reflect the degree to which the schedule was capable of capturing the underlying structure of the solution space. These figures of merit, i.e. their fitness value, were then used to guide the schedules towards optimality. After the optimization process, the schedules were tested using Monte Carlo simulations, the results of which shoved that GA-VLC is capable of generating scan schedules of high quality, optimized for radar databases of different sizes.

Sammendrag

Å designe en søkestrategi for en radardetektor krever problemspesifikke kompromisser og prioriteringer. Kalibreringen av slike strategier er typisk iterative prosesser. Framgangsmåtene kjennetegnes av kontinuerlig testing og evaluering, opp mot et satt kvalitetskriterium. I dette prosjektet undersøktes det hvorvidt genetiske algoritmer med kromosomer av variabel lengde kunne brukes til å generere slike strategier. Dette er en optimaliseringsmetode som er inspirert av den naturlige evolusjonsprosessen. Målet var å lage strategier som, gitt en database med kjente radarsystemer, kunne detektere så mange av disse systemene som mulig når søkestrategien ble brukt i et radarmiljø som svarer til databasen. For at dette skulle være mulig måtte det defineres en fitness-funksjon. Funksjonen måtte være i stand til bedømme en gitt strategis evne til å kapre det underliggende løsningsrommet. Det vil si at jo bedre en løsning var, dess høyere ble dens fitness-verdi. Disse verdiene var nødvendige for å guide optimaliseringsprosessen. Etter optimaliseringen ble strategiene testet med Monte Carlo-simuleringer av radarmiljøene. Resultatene viste at de beskrevne genetiske algoritmene er i stand til å generere søkestrategier av høy kvalitet for radardatabaser av ulik størrelse.

Contents

1 Introduction

When designing a scan schedule for a radar detector, a common procedure is to make an initial guess based on some knowledge of the radar environment or problem parameters, followed by subsequent testing and evaluation, leading to a continuous, iterative tuning of the schedule, with the aim being an increase in performance in relation to one or more objectives. Essentially, one first generates a schedule using a random search with a set of “sensible” parameters, before the schedule in question becomes subject to a subsequent tuning process, one in which various objectives are introduced and attempted met. While relatively straightforward for detection systems in simple radar environments, or systems whose task is comparatively achievable, the design of a scan schedule becomes increasingly difficult as the radar environment or task increases in complexity. The aim of this paper is to explore the feasibility of using genetic algorithms with variable-length chromosomes in the design of scan schedules for radar detectors. Specifically, whether such an optimization scheme might be able to handle the tuning process of a schedule initialized using random search automatically.

2 Optimization and the Scan Schedule Problem

2.1 Optimization

Not all problems have a single, optimal and analytically discoverable solution. For a given problem, there may exist a wide range of possible solutions, or indeed an infinite amount, depending on the criteria by which a problem would be considered solved. While the entire set of possible solutions might not be of interest, for instance due to the low quality of a given solution, the subset in which only the suitable ones are contained might still be quite large. Searching the entirety of this space for a so-called *optimal* solution might be difficult, be it due to the sheer size of the solution space, uncertainty as to what constitutes an optimal solution or merely the fact that there might not exist one single optimal solution, but indeed several, with each differing from the others in some way. This type of problem is considered an *optimization problem* and can be solved using optimization algorithms, i.e. algorithms that iteratively search for an ideal solution based on some pre-defined criteria of quality for the problem in question [1]. A general assumption concerning such problems is that there is an underlying structure in the solution space, although its precise nature is not, or cannot be, known analytically. The aim of optimization algorithms is to find, or at least approximate, these solutions. A typical optimization problem will consist of several parameters that are to be tuned in relation to each other, with the value of one often affecting which values the others might take. Together, these values then correspond to some solution. The manner in which the solution parameters are tuned is governed by some *objective function*, which is an expression of the desired goals and constraints the algorithm should optimize towards and operate within, respectively. The objective function is a mathematical formulation of the criteria an algorithm must consider when searching for a solution to a given problem. Examples of typical optimization problems, which tend to be characterized by there not being one single ideal way of solving the problem in question, include tasks related to scheduling or resource allocation. In other words, tasks in which different solutions reflect different considerations, prioritizations and ideas of optimality. The nature of these kinds of problems makes optimization an exercise in compromise. One such problem concerns the scanning pattern of a radar detection system.

2.2 The Scan Schedule Problem

The radar bands, i.e. the frequency range in which most radars operate, span from about 3 MHz to 300 GHz [2]. If one were to sample signals from this entire range at once, disregarding the technical challenges and inconveniences related to designing a system capable of doing so, the noise level alone would render the results gathered rather useless for most applications. Now, generally and luckily, one is usually not interested in listening to the entire radar spectrum at once, but rather a subset of it, with the precise frequency range of interest being dependent upon the task at hand. In electromagnetic warfare (EW), the research area with which this report concerns itself, a general example of a frequency range of interest is 2-18 GHz. While definitely an improvement, this is still, however, quite a large frequency span to sample from at once, particularly considering that one of the major challenges EW faces is trying to detect signals from sources which would very much prefer to, and might indeed have been specifically designed to, go undetected. The fact that these sources pose a potentially lethal threat further incentivizes having a sensitive detection system. These considerations, in addition to technical and noise related challenges, result in it being common to design radar detection systems that only operate within a limited part of the frequency range of interest at any one time, sequentially switching between bandlimited channels. While this significantly increases the sensitivity of the system, it does bring with it some problems, mainly stemming from the fact that most radars are not constantly illuminating their target, meaning the signals will not always be there for the detection system to detect. The consequence of this is that, although a radar signal might be present at some point in time, the detection system might not be sampling at the relevant frequency at that time, meaning the radar signal goes undetected. The problem, therefore, is this: if a given radar detector is to be useful, it must operate by sequentially switching between frequency channels, but doing so will cause constant blind spots that potentially contain signals that ought to be detected. This is the scan schedule problem for radar detection systems. How should the system divide its time between the various channels in order to maximize the signal detections?

2.3 Problem Description

This paper examines the feasibility of using genetic algorithms with variable-length chromosomes[1] (GA-VLC) in scan schedule optimization for radar detection systems. Given a radar database, i.e. a list of potentially detectable radar systems, the goal was to maximize the mean detection rates achieved by a schedule in a corresponding radar environment. Every radar system was assigned a priority of detection (1, 2 or 3) and greater emphasis was placed on the detection of high priority systems. Additionally, a subset of the top priority systems would have a time restriction associated with them. These were systems that, instead of being present at all times, might suddenly appear in the radar environment during the schedule run. The time restriction would then signify how quickly this system ought to be detected once present. This latter point was of key importance, and a major goal of the project was to examine whether it was possible to maximize the detections of time-restricted systems without otherwise compromising the overall performance of a schedule. Each scan schedule throughout the project was to last the same amount of time, $T_{schedule}$, somewhat arbitrarily set equal to 5 seconds. After optimization, the robustness of the scan schedules was tested using Monte Carlo (MC) simulations of the radar environments. This was done for radar databases of varying size, as the scalability of the optimization method was of interest. The optimized results were compared with results achieved by schedules that were generated using random search.

2.3.1 Radar Databases

As the purpose of the project was not to explore a specific situation, but rather to examine the general feasibility of using GA-VLC for solving the scan schedule problem, the radars used for optimization were not required to be overly complex, or real, systems. Instead, the radar

[1] Explained in chapter 3.

databases were generated by sampling pre-defined intervals of common radar parameters, generating randomized systems, while remaining subject to certain restrictions. In order to avoid a uniform distribution of systems across all channels, which is an overly unrealistic and rather uninteresting scenario, the radar systems were distributed across smaller frequency intervals, each of which had restrictions as to the type of system that might be present. This was done in order to better emulate a real radar environment, in which certain systems only operate within a small subset of the total frequency range available. Generally, the lower frequencies might be populated by lower priority systems, while the medium frequency channels might contain systems of any priority. At the highest end of the frequency range, there might only be time-restricted systems of top priority. Additionally, depending on the frequency range in which a system operated, some arbitrary restrictions were placed on the possible range of scan times the system in question be sampled from. This was done to assure non-uniformity in the radar parameters across different channels. Furthermore, the systems were allowed a maximum duty cycle of 10%, they were set to operate within only one frequency channel, and the parameters, once assigned, were held constant. In Table 2.1 is presented the value ranges from which the system parameters were sampled, while Table 2.2 contains information about how the systems were distributed across radar channels. The radar channel distribution is also illustrated in Figure 2.1.

Table 2.1 The parameters used for describing a radar system along with the possible values they might take.

Parameter	Value
Pulse width (PW)	0.1 – 100 μs
Pulse repetition interval (PRI)	10 – 1000 μs
Dwell time	5 – 20 PRI
Time restriction	0.2 – 0.5 $T_{schedule}$, None
Scan time (systems with time restriction)	0, 2 – 10 dwell times
Scan time (priority 1 systems without time restriction)	10 dwell times – 5 s
Scan time (priority 2 & 3 systems without time restriction)	1 – 5 s

Table 2.2 Description of how radar systems of varying priorities are distributed across frequency channels and the potential constraints that consequently might be placed on the scan times.

Radar system distribution				
# of systems [% of database size]	Frequency intervals	Priority distribution [1 restricted / 1 unrestricted / 2 / 3]	Min. scan time	Max. scan time
10	2.5 GHz - 3.5 GHz	0 % / 0 % / 50 % / 50 %	4 s	-
10	5 GHz - 6 GHz	0 % / 20 % / 50 % / 30 %	-	2 s
70	6.5 GHz - 12.5 GHz	10 % / 10 % / 40 % / 40 %	-	-
10	13.5 GHz – 17 GHz	100 % / 0 % / 0 % / 0 %	-	-

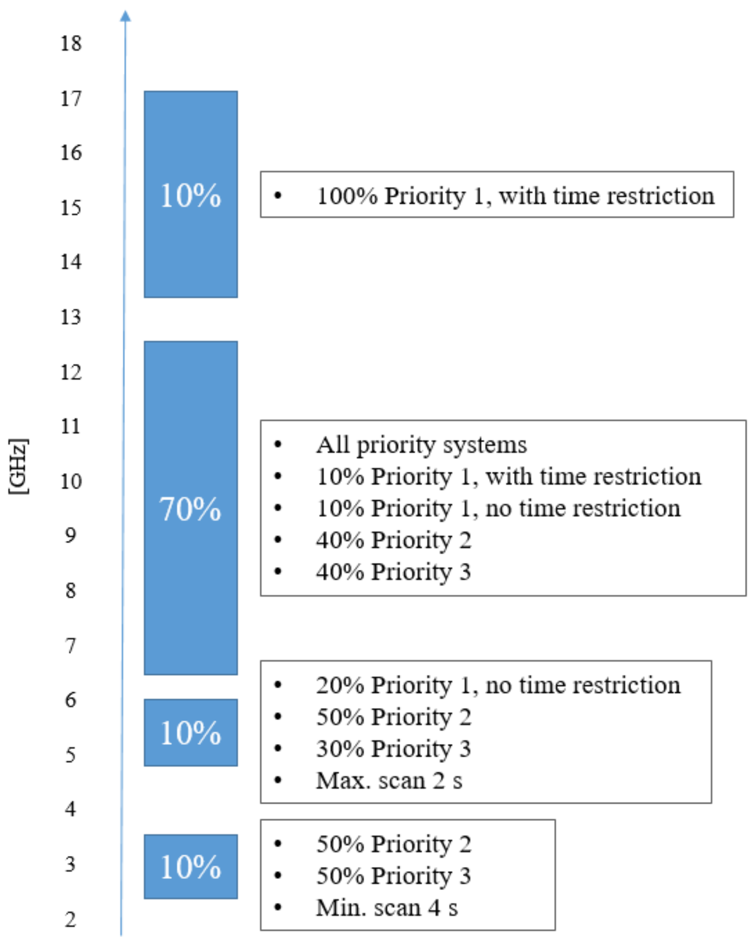


Figure 2.1 Radar system distribution across frequencies.

3 Genetic Algorithms

Genetic algorithms (GA), a family of algorithms that again belongs to the larger family of evolutionary computation algorithms, are search algorithms whose behavior is inspired by the principle of evolution by natural selection [1, 3]. Robust and versatile, such algorithms have been used in a wide range of complex optimization problems. The basic principle is to generate a group of candidate solutions, also referred to as a population of individuals[2], and then continuously improving upon the quality of these using biologically inspired mechanisms, such as crossover, mutation and selection.

3.1 Biological Overview for Intuition

While an in-depth understanding of evolutionary biology is certainly not necessary in order to understand this paper, a brief and quite simplified overview of the process of evolution might be helpful for intuition. Consider the lobster. Consider every lobster. Consider them solutions, a realization of a blueprint written by their very DNA, and consider this blueprint to be contained within a box called a "chromosome". This chromosome would then contain within it a complete description of a lobster[3], with separate sections of the DNA – the "genes" – corresponding to different features of the organism, such as size or color. Now, the theory of evolution suggests that the lobster chromosome did not start out this way, but rather, that its variety of features were developed gradually from generation to generation, with slight improvements over time gradually leading to what is now considered a lobster. The idea is that there are small variations within the chromosomes of individuals in a given species, meaning some individuals have certain features that others might lack. If such features prove to be useful for survival, this might then lead to an increased likelihood of reproduction, yielding these features a greater chance of crossing over into the next generation. Following this principle of crossover from generation to generation, one can see how, over time, the most useful genes might survive. However, this does not fully account for the variation one encounters in nature, as organisms occasionally develop entirely new features not already available in the mating pool. This happens through mutation, i.e. a random change that occurs in the DNA sequence of a living organism. If such a change proves useful for the organism, it will consequently have a greater chance of being passed on to the next generation.

[2] A quick note on the terminology: while "population" and "individual" are commonly used terms in the discussion of genetic algorithms, even a light perusal of the literature will reveal a somewhat divided community concerning the language one should use. Some opt for a naming convention in line with the biology from which it stems, while others prefer more computer scientifically sounding terms.
[3] In reality, an organism tends to be made up of several chromosomes, but for simplicity, assume only one is needed.

3.2 Theory of Genetic Algorithms

Designed to emulate the process of evolution by natural selection, genetic algorithms work by continuously evolving a population of candidate solutions, i.e. generating increasingly suitable individuals, in order to solve some optimization problem. Initially, once such a problem is defined, a population P_T consisting of N random individuals is generated, with each individual, i, being coded to represent a possible solution. In GA, solutions are commonly represented as an array of fixed length, with each element of the array corresponding to a part of the solution. These arrays are directly analogous to the structure of genes within DNA and may be coded in a variety of ways, depending on the problem at hand. Once initialized, each individual, i, is then evaluated based on some problem specific objective function, referred to as a *fitness function*, which yields a numerical score reflecting the quality of the solution to which the individual corresponds. After the fitness evaluation, the population will undergo genetic operations in order to create the next generation of the population, P_{T+1}. When generating a child population, the first step might be to clone the N_{top} individuals of the existing population, as this will make sure that the quality of the next generation remain at least at the same level as the current one. The population P_T will then undergo a selection process, which involves a pairwise statistical selection of parent individuals, with their selection probability depending on their fitness score. These parent individuals will then be used to generate one or more children through a crossover operation. The specifics of the crossover operation will vary depending on the nature of the problem and its implementation, but generally involves the exchange of features between the parents. Following this operation, the resulting children might then undergo a mutation operation, depending on some predefined mutation probability, during which random changes to its features may occur. This allows for the full exploration of the solution space. The children might also be generated through mutation alone, without crossover, if so desired. Regardless of method, the child individuals are then added to the population P_{T+1}. This cycle of selection-crossover-mutation repeats until the population P_{T+1} contains N individuals, at which point the new generation is evaluated and the cycle starts over. This process repeats for a desired number of generations or until some condition for termination is reached. The cycle is illustrated in Figure 3.1, as well as in the pseudocode below.

1. Initialize a population P_T to consist of N random individuals
2. Evaluate each individual i and assign them a fitness value
3. Clone the N_{top} most fit individuals into population P_{T+1}.
4. Select a pair of parent individuals from P_T
5. Generate children using crossover operations on the parents
6. Mutate children according to probability
7. Add children to P_{T+1}
8. Repeat step 4-7 until P_{T+1} contains N individuals
9. Return to step 2 and then either terminate or keep going

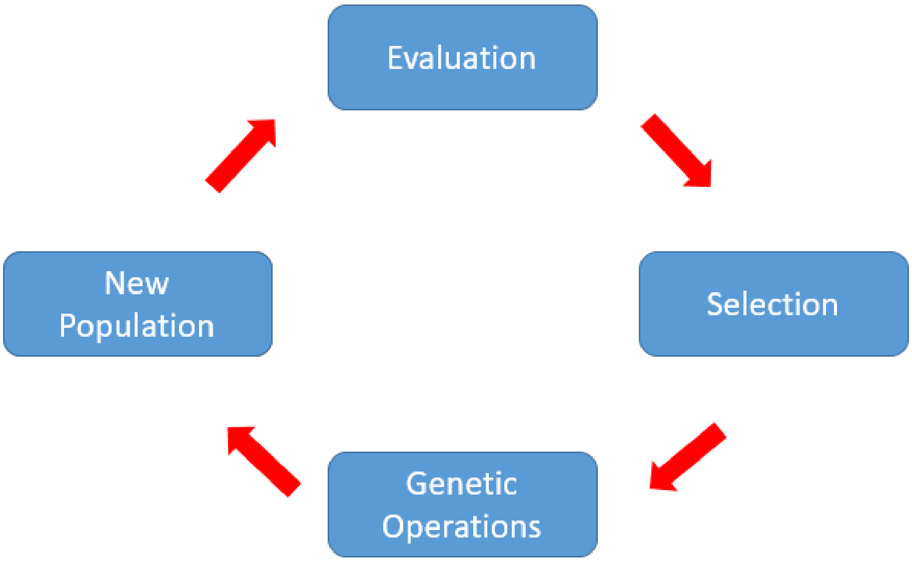


Figure 3.1 Cycle of genetic algorithms.

3.3 Implementing a Genetic Algorithm

Although every implementation of genetic algorithms will generally follow the abovementioned structure, each implementation will differ in several regards, as the design of a given operation will depend upon problem specific knowledge, albeit to a varying degree.

3.3.1 Selection Method

The selection process, i.e. the process through which elite individuals to be cloned or used for reproduction are chosen, need not necessarily depend on any knowledge of the problem, but rather on the fitness quality assigned during the evaluation process. There are several selection methods available in the literature, each with its own benefits and drawbacks [4]. In this project, *rank* selection was used, which works by ranking each solution according to their fitness score, a value which would then correspond to the probability of that individual being chosen as a parent.

3.3.2 Crossover and Mutation

The crossover and mutation operations tend to be comparatively problem specific, as these need to ensure the possible exploration of the entire solution space and the ways in which features are exchanged between individuals. The structure of the solutions and the values the variables might take will therefore affect the implementation of these operations, and their specifics will depend heavily on how the solutions are coded (for instance, crossover might work differently for binary strings as opposed to real-coded chromosomes – the same being the case for mutation). Simply put, a certain level of understanding relating to how features are allowed to link together and which values the parameters might take is beneficial during implementation.

3.3.3 The Fitness Function

The fitness function, i.e. the governing force of the entire algorithm, will depend heavily on knowledge (or assumptions) about what constitutes a high quality solution. During this process, a specially designed fitness function is used to assign a measure of quality to each individual in a population, reflecting how suited they are for solving the problem in question. The specifics of how such a fitness function is implemented will determine the quality of the solutions generated, and its design tend to be iterative rather than straightforward, as a lack of knowledge about what constitutes a so-called "good solution" might be the exact reason one opts for using an optimization algorithm in the first place. Finding a good fitness function is arguably the greatest challenge when it comes to GA and is itself, in a sense, an optimization problem.

3.3.4 Fixed- and Variable-Length Chromosomes

A typical implementation of GA will use fixed-length chromosomes [3, 5], which means that every solution will consist of exactly the same number of parameters. This is not ideal when it comes to the optimization of a scan schedule, as such an implementation would require either a priori knowledge or some rather major assumptions concerning the optimal amount of scan windows needed in a given situation. Instead, an implementation using variable-length chromosomes (VLC) was of interest, as this allowed for a more dynamic exploration of solutions.

4 Method

This section contains the specifics of how GA-VLC was implemented, as well as a description of how the MC simulations were used for schedule testing.

4.1 Structure of Variable-Length Chromosomes for Radar Scan Scheduling

A radar detector scan schedule is, essentially, a description of how a radar scanner should divide its time between frequency channels, that is, when and for how long should the system switch to a given channel before moving on to the next. This can be written as an array of time-frequency intervals, each of which describes a specific *time window*. A scan schedule, S, can be formulated as

$$S = \left[[t_1, f_1], [t_2, f_2], \ldots, [t_n, f_n]\right] \tag{4.1}$$

Where t_i and f_i correspond to a duration and a center frequency, respectively. While both values are real-coded, the frequency values may only take on a range of discrete values, each corresponding to a specific channel. In this project, the channel size was set as 250 MHz, while the entire frequency span ranged from 2-18 GHz, which translates into 64 possible frequency channels. The time variable, on the other hand, was allowed to take on a continuous range of values spanning from the shortest pulse width to the max scan time in the radar environment in question. Upon initialization, however, the time variables were sampled from the interval $[PRI_{min}, 3PRI_{max} + PW_{max}]$, the rationale for which follows from section 4.3.1. All schedules were initialized to last approximately $T_{schedule}$.

4.2 Genetic Operations

4.2.1 Crossover

The crossover operation in GA mixes features from two (or more) parent individuals, with the aim being a thorough exploration of how the solutions are affected by the various combinations of features. While there are several ways to implement this operation, a common method is k-point crossover [6]. With a fixed-length chromosome, this might involve randomly choosing k points along the chromosomes, splitting both chromosomes at said points, and then combining the cut-off parts in some manner – for instance, equally or by randomly choosing which parent a given section is to come from. K-point crossover is illustrated in Figure 4.1.

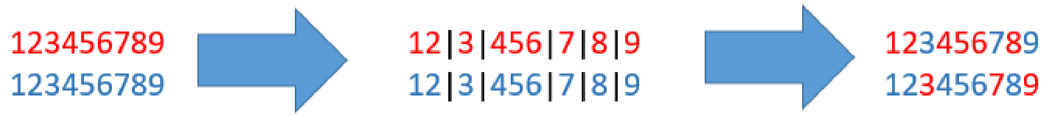


Figure 4.1 K-point crossover for two fixed-length chromosomes (red and blue) with k=5.

While straightforward for fixed-length chromosomes, a slight modification is necessary in order for this to be useful for VLC as well. In order to maintain the “variable-length” element of the algorithm, k-point crossover can instead be performed temporally. That is, by making use of the fact that the scan schedules are to last a fixed amount of time, the chromosomes can each be split into k (relatively equal) time intervals, combining these to create children, despite the sections might be of unequal length. This was the crossover operation used in this project, with the number of splits being randomly sampled between 1% and 10% of $T_{schedule}$.

4.2.2 Mutation

In order to allow the full exploration of the solution space, each time-frequency interval, i.e. scan window, of a mutating individual would undergo one of the following mutation operations:

- Copy: the scan window remains unaltered
- Random channel switch: the scan window switches frequency channel
- Remove: the scan window is removed from the chromosome
- Time change: the scan window gets a new duration, sampled using a normal distribution with mean equal to the original duration and standard deviation equal to 10% of the duration.

The probability of an interval undergoing a given operation was set to the mutation rates shown in Table 4.1, in which the variable $mutation\_scale$ was initialized to 100 and would increase by one as each generation passed. These values were arrived upon through hyperparameter tuning.

Table 4.1 Mutation operations and the corresponding mutation rates.

Mutation Operation	Probability of Operation
Copy	$\frac{95}{mutation\_scale}$
Channel switch	$\frac{1}{mutation\_scale}$
Remove window	$\frac{1}{mutation\_scale}$
Time change	$\frac{3}{mutation\_scale}$

4.3 The Fitness Function

The purpose of the fitness function is to assess the individuals in a population, assigning them a measure of quality corresponding to how good they are at solving the problem in question. The best individuals then have a greater chance of being selected, ideally leading to an overall increase in quality of the next generation. The job of the fitness function is to guide the solutions towards optimality. The function must therefore be precisely defined in order to ensure an appropriate convergence, and its implementation will require a degree of problem knowledge. A key element is the difference in how systems with and without time-restrictions are handled. In the derivation below, first is assumed the latter case, as a variant of this was used to derive the former.

4.3.1 Probability of Detection

The quality of a given scan window should, for a given radar, reflect how likely it is to intercept the radar signal, as well as how likely it is to classify it. It was assumed that the probability of classification would depend on the amount of pulses the window were potentially able to contain, with classification being guaranteed with 3+ pulses. The window would therefore be given a quality measure reflecting whether it would be able to contain 0, 1, 2 or 3+ pulses. The quality Q_w^R of a given window *w* in relation to a given radar *R* could be calculated using the following equation:

$$Q_w^R = q_w^R \begin{cases} 0, & t_w \leq t_{PW}^R \\ \dfrac{t_w - t_{PW}^R}{3t_{PRI}^R}, & t_{PW}^R < t_w < 3t_{PRI}^R + t_{PW}^R \\ 1, & t_w \geq 3t_{PRI}^R + t_{PW}^R \end{cases} \tag{4.2}$$

Where t_w is the duration of the scan window, t_{PRI}^R and t_{PW}^R are the PRI and PW of radar R, respectively (the explanations of these terms can be found in appendix A), and q_w^R is

$$q_w^R = \begin{cases} 0, & t_w \leq t_{PW}^R \\ 0.1, & t_{PW}^R < t_w \leq t_{PRI}^R + t_{PW}^R \\ 0.5, & t_{PRI}^R + t_{pw}^R < t_w \leq 2t_{PRI}^R + t_{PW}^R \\ 1, & 2t_{PRI}^R + t_{PW}^R < t_w \end{cases} \tag{4.3}$$

Furthermore, a scan window should be judged by how likely it is to intercept a given radar signal, a probability which will depend on the size of the scan window. Assuming a window is able to contain $i \in [1, 3]$ pulses, the probability of it doing so will depend on the relationship between the window size and the radar parameters. This can be expressed[4] by the effective scan window size $t_{w_{eff}}^R$, which is a measure of the time interval a scan window effectively spans, when considering the size of the dwell of a given radar, *R*. That is, because a radar dwell might contain more than *i* pulses, the probability of the scan window intercepting *i* pulses should depend in some way on both the size of the radar dwell and the size of the scan window. Think of it as a range of potential scan window starting positions. This time interval can be calculated using the equation below.

$$t_{w_{eff}}^R = \begin{cases} \left(t_w - (i-1)t_{PRI}^R - t_{pw}^R\right)\left(\frac{t_{dwell}^R}{t_{PRI}^R} - (i-1)\right), & (i-1)t_{PRI}^R + t_{PW}^R < t_w < it_{PRI}^R + t_{PW}^R \\ t_w + t_{dwell}^R - (2i-1)t_{PRI}^R - t_{pw}^R, & t_{w\,effmax}^R \geq t_w \geq it_{PRI}^R + t_{PW}^R \end{cases} \tag{4.4}$$

Where t_{dwell}^R is the dwell time of radar *R*. Once found, $t_{w_{eff}}^R$ can be divided by the maximum effective window size, $t_{w\,effmax}^R$, i.e. the effective window size corresponding to the guaranteed detection of the radar system in question, in order to gain the probability of interception for the given scan window. $t_{w\,effmax}^R$ is given by the equation below.

$$t_{w\,effmax}^R = t_{scan}^R - t_{dwell}^R + it_{PRI}^R + t_{PW}^R \tag{4.5}$$

Where t_{scan}^R is the scan time of radar *R*. One last coefficient needed to describe the quality of a scan window is a measure of scan window repetition. Due to the cyclical nature of many radar systems, the scan windows stand a chance of covering the same relative period of a radar cycle, essentially yielding no extra potential for detecting the radar system in question. This can be accounted for by applying an overlap reduction factor, calculated by the equation below.

$$o_w^R = \frac{t_w - t_{overlap}^R}{t_w} \tag{4.6}$$

Where $t_{overlap}^R$ corresponds to how much of t_w has already been covered by previous windows, i.e. how much of the window *w* is essentially "wasted" when trying to detect radar system *R*. $t_{overlap}^R$ can be found by shifting all scan windows down into the interval $[0, t_{scan}^R]$ and measuring the overlap between a given scan window and all previous scan windows.

[4] A more in-depth explanation of Equation 4.4 and 4.5 can be found in appendix B.

Combining the above equations yields, for all windows w in a given frequency channel, an expression for the probability of detecting radar *R*:

$$\Phi^{D_R} = \sum_{w} Q_w^R o_w^R \min\left(\frac{t_{w_{eff}}^R}{t_{w_{effmax}}^R}, 1\right) \tag{4.7}$$

Φ^{D_R} then corresponds to the probability of detecting a radar *R* with the given schedule. As the aim was to maximize all detection probabilities, the *minimum* value of all detection probabilities were be added to the overall fitness value of the schedule. This was handled separately for each priority, *p*, as shown below.

$$\Phi_1^{\mathrm{D}} = \sum_{p} p\Phi_{min_p}^{D} \tag{4.8}$$

With $\Phi_{min_p}^{D}$ being the minimum detection probability of all radar systems with priority *p*. This also made it simple to leave out low priority systems, if so desired, as these might fall under the category of "nice to detect" rather than being of critical importance.

Lastly, as the complexity of the problem was quite large, a further method of guiding the optimization process was included, specifically the weighted rewarding of each detection probability in itself:

$$\Phi_2^{\mathrm{D}} = \sum_{p} \frac{\log(1 + p \sum_{R^p} \Phi^{D_{R^p}})}{N_p} \tag{4.9}$$

Where N_p is the number of systems with priority p (and without time-restriction). The log-term is included to make sure Φ_2^{D} is unable to dominate Φ_1^{D}. Finally, this yields:

$$\Phi^{\mathrm{D}} = \Phi_1^{\mathrm{D}} + \Phi_2^{\mathrm{D}} \tag{4.10}$$

4.3.2 Necessary Tuning for Time-Restricted Systems

Although the above derivation will mostly hold for time-restricted systems as well, certain changes are necessary to make it applicable. As these systems might show up whenever, and would then require quick detection, several visits to the channel needed to be encouraged. Furthermore, these visits had to be spread out in time, and the scan windows that perform these visits should be of as high a quality as possible. By considering every such radar system as a series of subsystems, this can be achieved by first dividing the schedule time $T_{schedule}$ into smaller time windows and then calculating a separate detection probability for each of these. The number of desired sections N_{ds}^R required for a time-restricted radar system with time-restriction T_{res}^R is:

$$N_{ds}^R = int\left(2 \times round\left(\frac{T_{schedule}}{T_{res}^R}\right) + round\left(\frac{T_{res}^R}{T_{schedule}}\right)\right) \tag{4.11}$$

The above formula guarantees that scan windows contained within subsequent detection windows are always separated by less than T^R_{res}. Within each of these sections, the probability of detection would have to be calculated as described in equation 4.10, although with certain modifications. Firstly, Φ^{D}_2 was omitted completely. Further, the remaining elements needed some minor altering, as certain time-restricted systems would have a scan time equal to zero. For systems with $t^R_{scan} > 0$, however, no further alteration needed to be made, and the probability of detecting a time-restricted radar R within a section *n* is given by:

$$\Phi_{\mathrm{n}}^{D_R} = \sum_w Q_w^R \min\left(\frac{t^R_{w_{eff}}}{t^R_{w\ effmax}}, 1\right) \tag{4.12}$$

While for systems with $t^R_{scan} = 0$, $\Phi_{\mathrm{n}}^{D_R}$ would reduce to:

$$\Phi_{\mathrm{n}}^{D_R} = \sum_w Q_w^R \min\left(\frac{t_w}{it_{PRI} + t_{pw}}, 1\right) \tag{4.13}$$

The probabilities could then be summed and normalized. Additionally, in order to further incentivize the detection of time-restricted systems, the normalized probability sum, being in the interval [0, 1], was used as the argument in the exponential function, yielding:

$$\Phi^D_{\mathrm{res}} = \exp\left(\frac{\sum_R \frac{\sum_n \Phi_n^{D_R}}{N^R_{ds}}}{N^{res}}\right) \tag{4.14}$$

4.3.3 Time Constraint

Finally, the fitness function needed to adhere to constraints regarding duration. A time constraint was added to the total fitness using the following equation:

$$\Phi^T = abs\left(T_{schedule} - \sum_i t_{w_i} \right) \tag{4.15}$$

Where $\sum_i t_{w_i}$ is the total duration of a given scan schedule.

4.3.4 The Complete Fitness Function

The total fitness value for a given schedule is then given by:

$$\Phi = \Phi^{\mathrm{D}} + \Phi^D_{\mathrm{res}} + \Phi^T \tag{4.16}$$

4.4 Monte Carlo

The radars within a given radar environment tend to work independently of one another, i.e. they are not synchronized in any way. Therefore, when testing the robustness of a scan schedule, any simulation of a radar environment should be able to represent its inherently stochastic nature. Monte Carlo (MC) simulations are a useful way of achieving this. In MC, the relevant parameters for describing the relative state of a radar system in relation to the scan schedule can be (pseudo-)randomly sampled, and testing the scan schedule against a wide range of states yields useful statistical data. The simulation was designed to take a database of radars and a scan schedule for radar detectors as input. It would randomly initialize every radar system according to their parameters, before measuring whether or not the scan schedule was able to detect 0, 1, 2 or 3 pulses within the schedule time (or, in the case of the time restricted systems, within the given time restriction). This would be performed for the desired number of simulation runs, ultimately yielding the detection rates for each schedule. The non-restricted systems, which were assumed always present in the radar environment, would be assigned a random dwell start time t_{start}^{R} in a way that ensured that the entire dwell was guaranteed to appear within the first scan time of the radar system. It would then repeat cyclically at intervals equal to the size of the scan time until $T_{schedule}$. These dwells were then compared with the scan schedule windows, yielding detection statistics by calculating the overlaps between them. The time-restricted systems were similar in that they repeated cyclically once present. These systems, however, were assigned a random start time t_{start}^{R} within the entire schedule time, i.e. a time of sudden appearance, from which point they repeated cyclically at intervals equal to the scan time. This would repeat, not until schedule end, but from time of appearance until it had been present for a duration equal to the time restriction, i.e. from t_{start}^{R} to $t_{start}^{R} + T_{res}^{R}$. If the duration from appearance until schedule end was less than this, the schedule – which was assumed to repeat once it reached its end – was assumed to start over.

5 Results

The optimization algorithm was performed for five randomly initialized radar databases of five different sizes: 50, 100, 250, 500 and 1000. These were initialized as described in section 2.3. Each radar database was used in a GA-VLC optimization process, as well as a random search (RS) one. In the interest of fairness, and for making sure optimization is actually taking place when using GA, the RS optimization process generated the same amount of schedules as were generated in total by GA-VLC. These were evaluated using the fitness function and the best were used for comparison with the best ones generated by GA-VLC.

As the goal of this project was to explore the feasibility of using GA-VLC for scan schedule optimization, i.e. examine it as a method for guiding scan schedule design, convergence was not a necessary criterion in order for the feasibility to be demonstrated. A relatively low number of generations was therefore used. Each optimization process ran for 500 generations for all databases, with a population size equal to the radar database in every case. Three elite individuals were copied from one generation to the next. Mutation- and crossover rates were both set equal to 50%, meaning a child individual were equally likely to be produced through either crossover or mutation. Rank selection was used, as was temporal k-point crossover. After optimization, every schedule underwent 10^4 runs of an MC simulation. Below is presented the mean detection rates for 3+ pulses for time-restricted systems, top priority systems, and all systems. Each table and figure show the results for schedules generated by both RS and GA-VLC.

5.1 Time-Restricted Systems

Table 5.1 shows the mean detection percentages for all the time-restricted radar systems, obtained through 10^4 runs in an MC simulation. Detections statistics are shown for the best schedules achieved by both RS and GA-VLC. Upon comparison, the rates achieved by the GA-VLC optimized schedules are greater than those that are achieved by those achieved through RS. This is demonstrated graphically in Figure 5.1.

Table 5.1 Mean detection rates for time-restricted systems achieved by RS and GA-VLC generated schedules.

Mean Detection Rates [%] for Time-Restricted Systems		
# Radars	RS	GA-VLC
50	94.40	98.67
100	91.80	97.29
250	92.59	96.83
500	89.20	94.10
1000	89.83	95.39

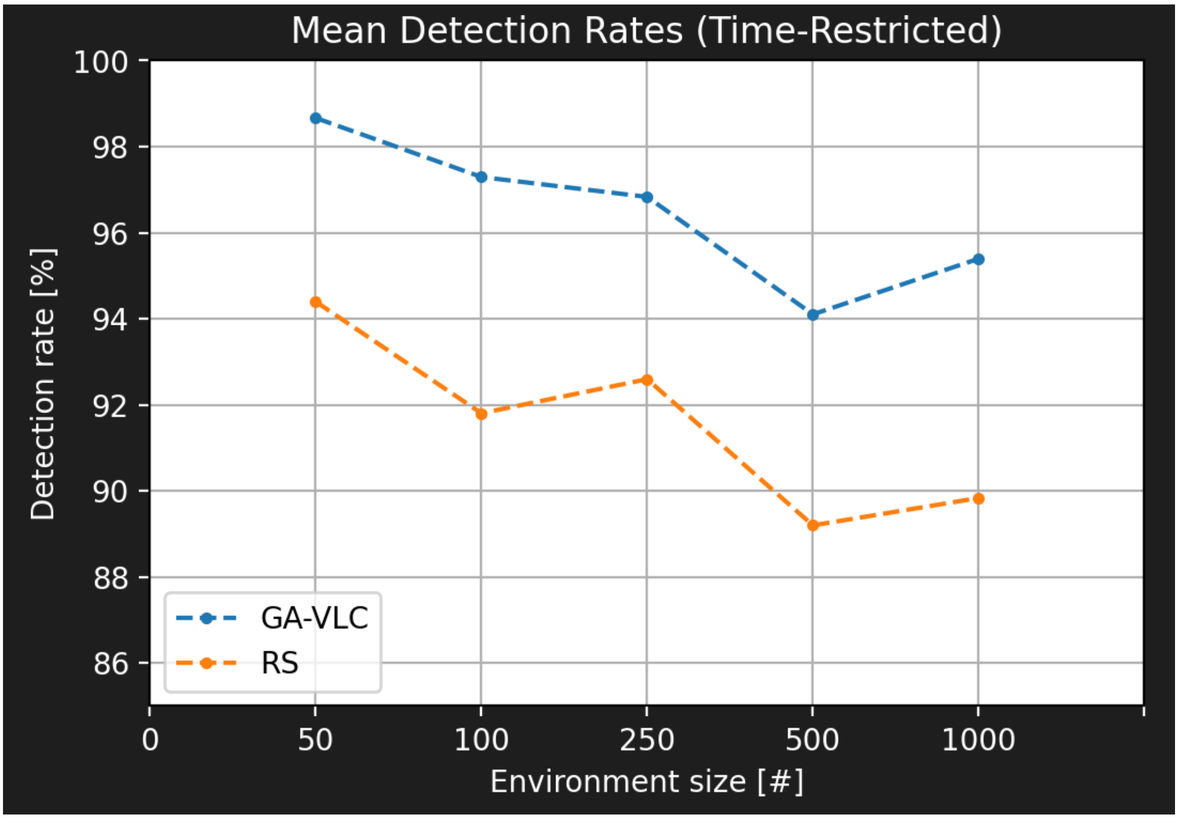


Figure 5.1 Comparisons of the mean detection rates for time-restricted radar systems achieved by RS and GA-VLC generated scan schedules.

5.2 Top Priority Systems

The mean detection percentages for all the top priority radar systems, i.e. radar systems with priority equal to 1, including the time-restricted ones, are presented in Table 5.2. As with the time-restricted results shown in the previous section, the GA-VLC rates are, for every radar database, greater than those found through RS. This is illustrated in Figure 5.2.

Table 5.2 Mean detection rates for top priority systems achieved by RS and GA-VLC generated schedules.

Mean Detection Rates [%] for Top-Priority Systems		
# Radars	RS	GA-VLC
50	70.81	76.69
100	64.53	70.26
250	67.09	70.49
500	62.44	66.78
1000	65.00	68.49

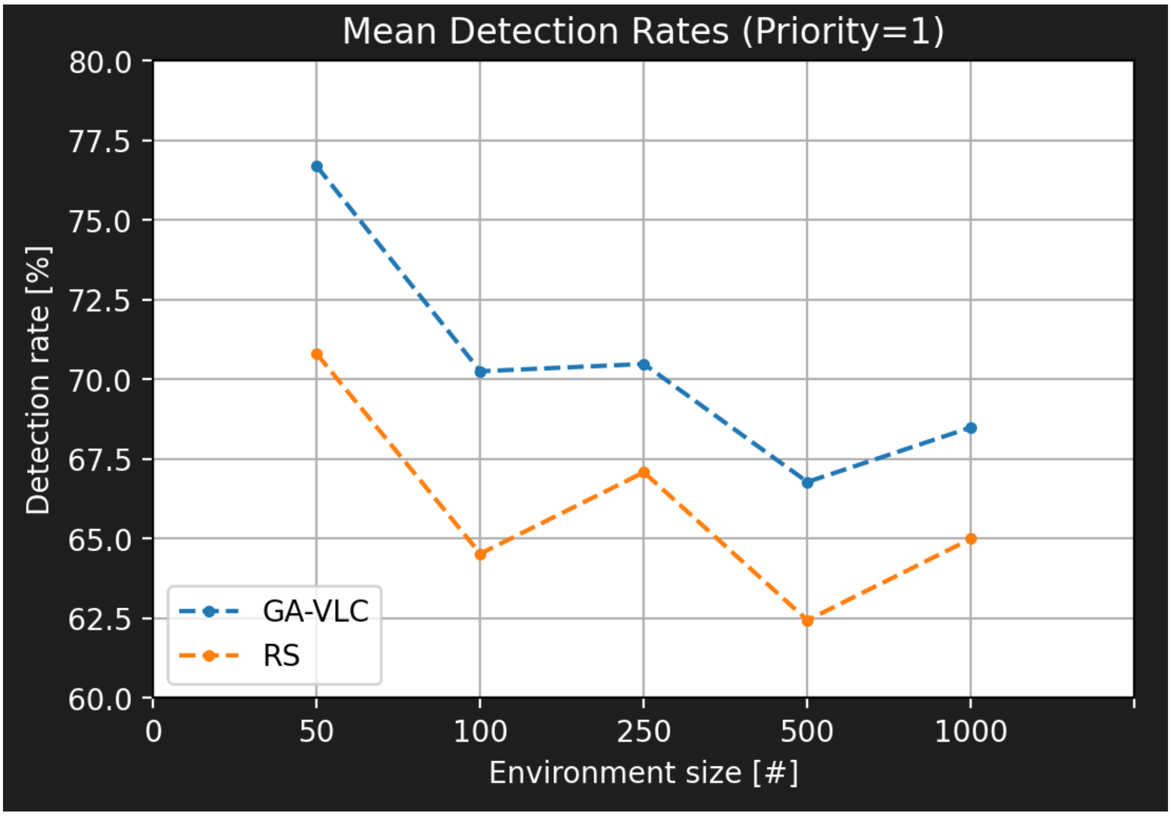


Figure 5.2 Comparisons of the mean detection rates for top-priority radar systems achieved by RS and GA-VLC generated scan schedules.

5.3 All Systems

Table 5.3 shows the mean detection percentages for all radar systems, regardless of priority and time-restriction. As with the time-restricted and top priority systems, the detection rates achieved by the GA-VLC generated schedules are greater than those achieved by RS. The data is plotted in Figure 5.3.

Table 5.3 Mean detection rates for all radar systems achieved by RS and GA-VLC generated schedules.

Mean Detection Rates [%] for All Systems		
# Radars	RS	GA-VLC
50	29.32	30.33
100	23.67	25.02
250	23.75	24.61
500	22.33	23.84
1000	22.73	23.45

Figure 5.3 Comparisons of the mean detection rates for all radar systems achieved by RS and GA-VLC generated scan schedules.

6 Discussion and conclusion

The aim of this project was not to generate a perfect schedule for a specific situation, but rather to examine whether GA-VLC could be used to generate schedules subject to a certain set of restrictions and considerations. In other words, the key elements of interest are not the solutions themselves, as the test cases in this report are generic and simplified, but rather whether or not the algorithm is capable of sufficiently capturing the underlying solution space in order to guide the scan schedule design process automatically. The algorithm was therefore not required to converge upon any final solution, but instead, it needed to be able to structure the solutions according to the target objectives. The degree to which this was achieved could be assessed by comparing the results of the GA-VLC optimized schedules with the results of the RS schedules, as the results of such a comparison would yield an indication as to whether the fitness functions had the ability to guide the GA algorithm towards optimality.

As seen in Table 5.1, 5.2 and 5.3, and the respective figures to which they correspond, the optimized schedules were, for every radar database, capable of producing greater detection rates than were achieved by RS. This suggests that the fitness function is capable of guiding the optimization process, given the problem parameters. Of special note, then, is the manner in which the time-restricted cases are considered, i.e. as a sum of subsystems, as this method does not necessarily require the radar parameters to remain constant. This indicates that, as the radar systems take on aspects of real systems, such as a varying frequency, scan pattern or any other parameter variation, the optimization algorithm should be able to be expanded in order to accommodate these changes.

In conclusion, GA-VLC can be used as a tool for capturing the underlying structure of the solution space of radar detector scan schedules and seems to be capable of guiding candidate solutions towards optimality.

Appendix

A Overview of Radar Theory

A radar (radio detection and ranging) is a detection system that transmits radiofrequency (RF) waves, i.e. electromagnetic (EM) waves with frequencies below 300 GHz, toward a region of interest, using the reflections of these to gather information about the region in question [2]. The system will generate an electrical signal that can be transmitted into space using an antenna, with which the return signal can later be received as well (assuming a monostatic system, i.e. the transmitter and receiver is located at the same place). The time delay, relative strength, direction and potential frequency- and phase shift of the return signal can then be used to gain information about the surrounding environment and the objects and entities within it. A radar antenna can be either omnidirectional or directional, and the RF waves can be transmitted continuously or as a series of pulses. In the latter case, the *pulse width* then describes the duration of the pulse, i.e. how long the radar was transmitting, and the time between pulse transmissions is described by the *pulse repetition interval* (PRI). This is visualized in Figure A.1.

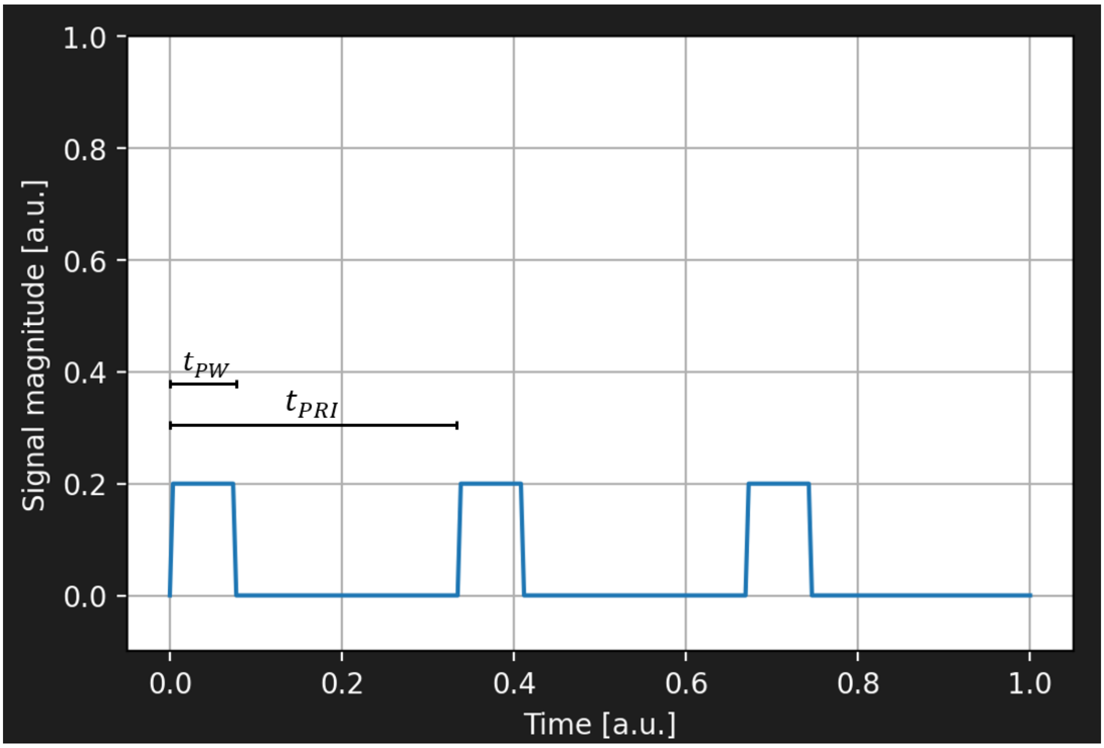


Figure A.1 Plot of three subsequent pulses of pulse width T_{PW} and pulse repetition interval (PRI)

While this report considers pulsed, directional radars, it is worth noting, in the interest of thoroughness, that although the radars considered are thought to be directional, the nature of EM waves will cause some energy to be transmitted in other directions as well. The energy transmitted in such secondary directions are contained in the *side lobes* of the *antenna radiation pattern*, while the main part of the energy is contained in the *main lobe*. An example of a typical antenna radiation pattern is presented in Figure A.2. Although relevant at shorter ranges, the side lobes become harder to detect at long ranges, as they tend to contain significantly less energy than the main lobe. Consequently, if a large distance between a radar and a radar detection system can be assumed, as it is in this project, only the main lobe is detectable and of interest.

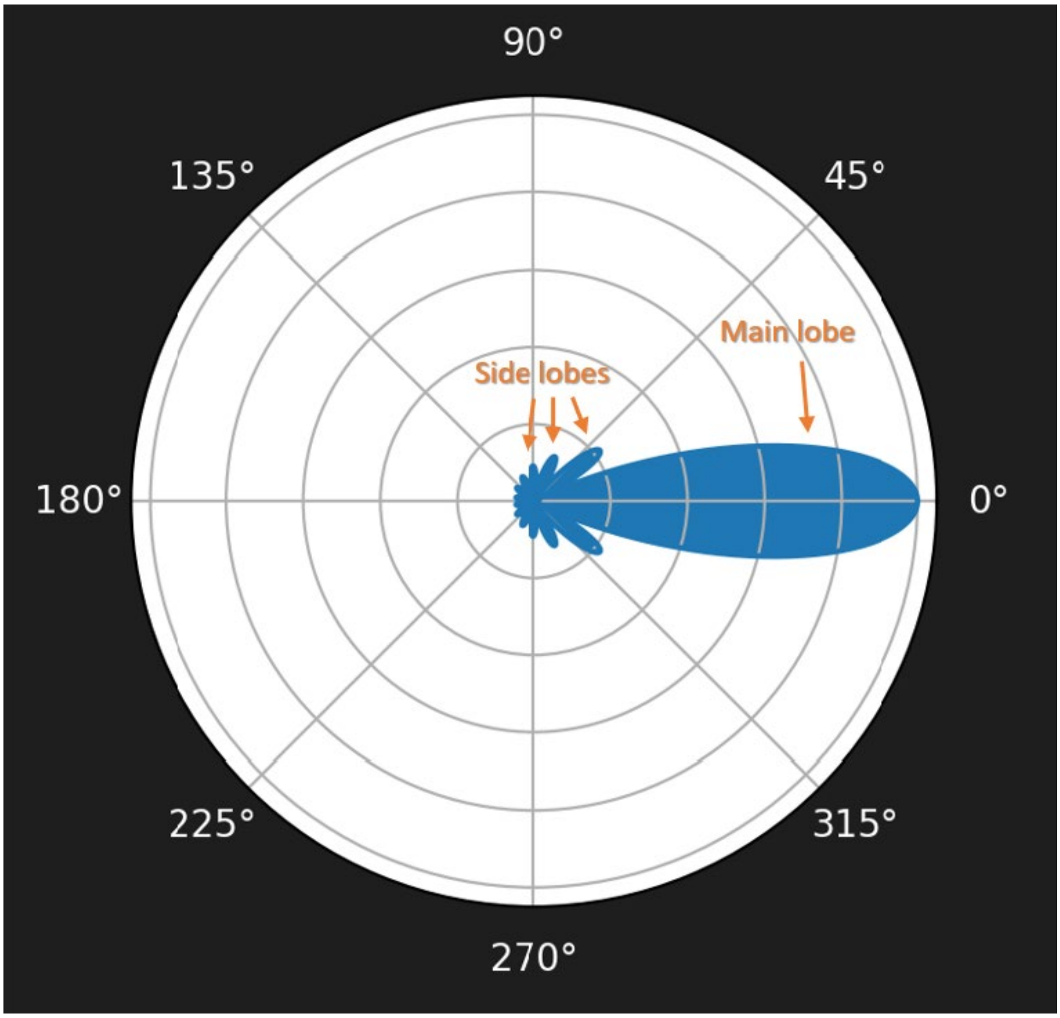


Figure A.2 Antenna radiation pattern with marked main- and side lobes.

Although dependent upon its task and properties, a directional radar system will often *scan* its main lobe around in some way rather than having it point in one fixed direction at all times. While the exact scan pattern and mode of operation of a radar will vary from system to system, the consequence of scanning is that the various regions of interest will be illuminated sequentially, as opposed to constantly. The time a given region is illuminated by the main lobe is called the *dwell time* and the time interval between the start of subsequent dwells is called the *scan time*. This is illustrated in Figure A.3.

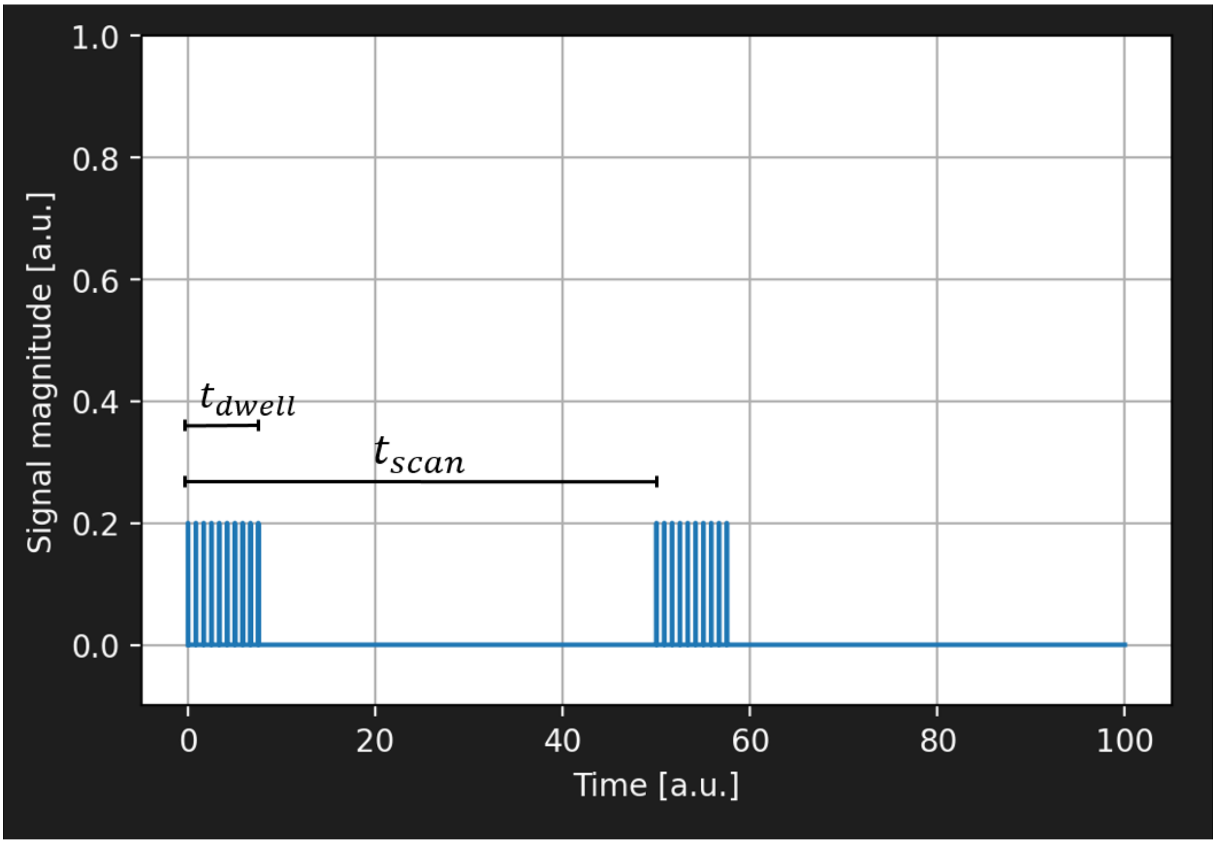


Figure A.3 Plot of two subsequent scan times of duration T_{scan} containing dwells of duration T_{dwell}.

During a pulsed radar dwell, which can be said to consist of a series of PRIs, the target region will be illuminated by the number of pulses that corresponds to the dwell time in question. Scan- and dwell times, PRI and pulse width are all among the characteristic features of a given radar system and the measurements of these parameters can therefore aid in radar classification.

B Explaining the Equations for Effective Scan Window Sizes

This appendix contains an explanation of the reasoning behind Equations 4.4 and 4.5, which together can yield the probability that a given scan schedule window of size t_w intercepts *i* pulses from radar *R*.

A convenient starting point is this general assumption: there might be several useful placements of a given, eligible scan schedule window relative to a radar dwell, each of which would be able to contain the desired number of pulses. How many such placements there are would depend on the radar parameters and the scan schedule window size.

Now, assume that the range of possible placements can be expressed as one or more time intervals. Furthermore, assume that, at some point, a scan schedule window will be so large that it is guaranteed to contain the desired number of pulses, that is, there is a maximum (or, rather, a "sufficient") range of placements. From this follows the need for Equations 4.4 and 4.5, the former of which yields the size of the time interval(s) corresponding to possible scan window placements ($t^R_{w_{eff}}$), while the latter yields the size of the maximum/sufficient time interval ($t^R_{w\ eff_{max}}$).

B.1 Effective Scan Window Size

Equation 4.4 is presented below.

$$t^R_{w_{eff}} = \begin{cases} \left(t_w - \left(i-1\right)t^R_{PRI} - t^R_{pw}\right)\left(\dfrac{t^R_{dwell}}{t^R_{PRI}} - (i-1)\right), & (i-1)t^R_{PRI} + t^R_{PW} < t_w < it^R_{PRI} + t^R_{PW} \\ t_w + t^R_{dwell} - (2i-1)t^R_{PRI} - t^R_{pw}, & t^R_{w\ eff_{max}} \geq t_w \geq it^R_{PRI} + t^R_{PW} \end{cases} \tag{4.4}$$

As the equation shows, there are two possible situations one might encounter, depending on whether or not t_w is smaller than $it^R_{PRI} + t^R_{PW}$. Simply put, these two cases reflect whether the possible placements of the scan schedule windows can be found in a series of discrete intervals (henceforth referred to as "Case 1") or along a continuous interval ("Case 2"). Case 1 refers to the first line in Equation 4.4, while Case 2 refers to the second.

B.1.1 Case 1

So consider Case 1, where $t^R_{w_{eff}}$ is the product of the two terms contained within the outermost parentheses. The first of these terms corresponds to the size of a single time interval of possible scan window placements within a radar dwell, while the second term is an integer value corresponding to how many such time intervals there are for the dwell in question. Deriving the second term is relatively straightforward (it is simply the number of groups containing *i* consecutive pulses that can be formed within a dwell). The first term is slightly less straightforward, but the basic idea is this: place a scan window so that it ends *just* behind the edge of a pulse (for simplicity, assume *i*=1 – the principle remains the same regardless) and then just *slide* the window forwards until the start of the window is *just* before the start of the pulse. The length the window has been slid is then the interval of possible start positions.

This can be understood through Figure B.1. Here, the red intervals (equal in size, as they refer to the same scan window) correspond to the first and last possible placements of a given scan window in order for it to be capable of containing the first pulse in the figure. $t_{w_{last}}$ is basically just $t_{w_{first}}$ after sliding it towards the right. The green interval can then be thought of as the interval of possible start positions for the scan schedule window in question, if the window is to contain the first pulse. The green interval, which is described by the first term in Case 1 (in this case with *i*=1), would then be multiplied by the number calculated by the second term, which in this case would be equal to 2, as there are only two pulses present in this particular dwell.

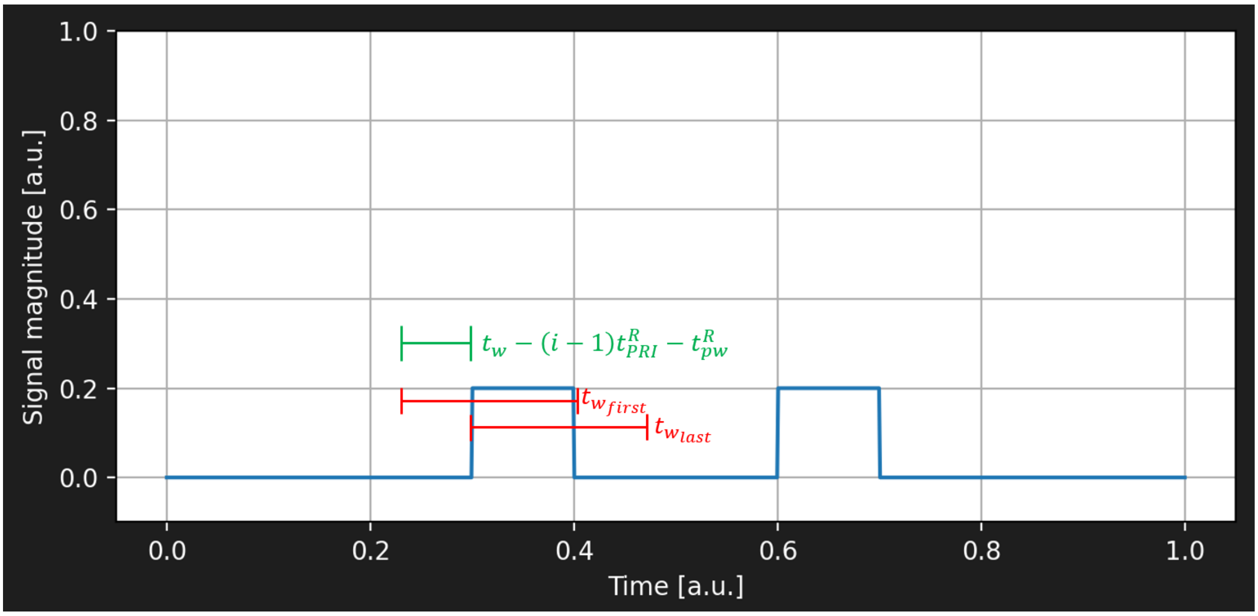


Figure B.1 *How discrete window placement intervals are calculated for a scan schedule window* t_w *and i=1. The two red intervals correspond to the first and last possible placements of the scan schedule window if it is to detect the first pulse. The green interval is the range of all possible start positions for the window.*

B.1.2 Case 2

Case 2 is very much the same as Case 1, the only difference being that there is a continuous interval of positions from which the scan schedule windows might start rather than several smaller ones, i.e. there are no jumps between discrete intervals within the dwell and so no interval multiplication is needed. The scan window is large enough to always contain *i* pulses as it is slid across the dwell. This is illustrated in Figure B.2. Here, *i* is still equal to 1, but the scan schedule window is now large enough to always contain a pulse, no matter where it is set to start within the green interval. The green interval is given by the second line in Equation 4.4.

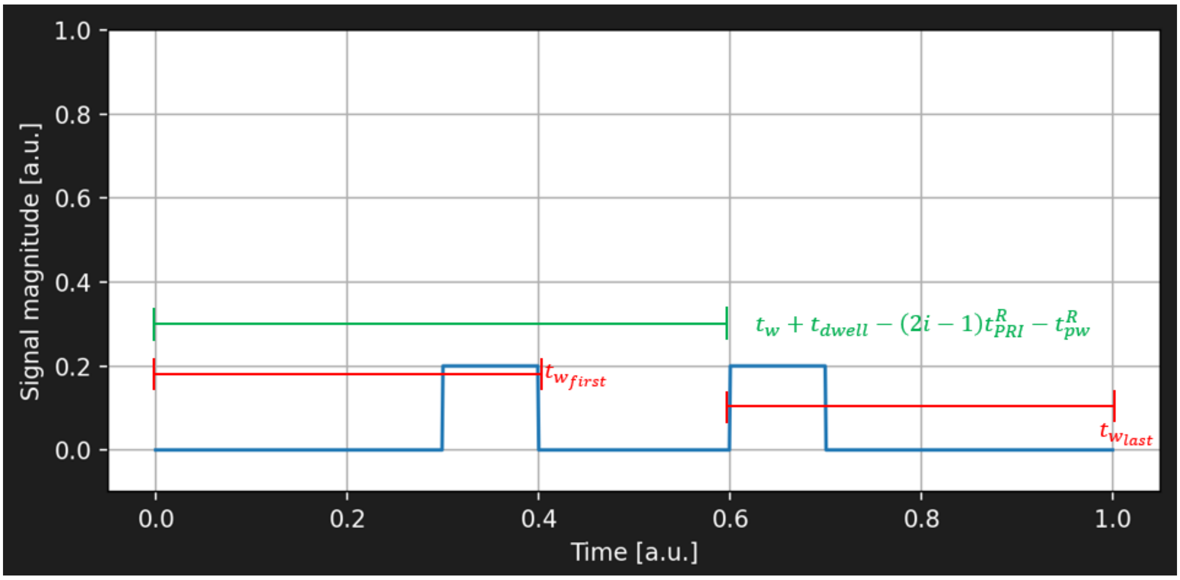


Figure B.2 How a continuous window placement interval is calculated for a scan schedule window t_w and i=1. The two red intervals correspond to the first and last possible placements of the scan schedule window if it is to detect the i=1 pulse. The green interval is the range of all possible start positions for the window.

B.2 Max Effective Window Size

Equation 4.5 is presented below.

$$t^R_{w\,effmax} = t^R_{scan} - t^R_{dwell} + it^R_{PRI} + t^R_{PW} \qquad (4.5)$$

This expression yields the maximum effective window size, i.e. the largest window needed in order to guarantee detection of *i* pulses. The derivation goes like so:

Imagine placing a large scan window that *just* misses a radar dwell. That is, imagine it starting at the end of a radar dwell and ending just before the beginning of the next one. It currently has size $t_w = t^R_{scan} - t^R_{dwell}$ and there are no pulses contained within it.

Then, if one wants to expand the window so that it always contained the desired number of pulses (in this case, say *i*=1), one can imagine stretching the window slightly on both sides, letting each side expand until a whole pulse is contained on either side. The size of the expansion required at the right side of the window is equal to one pulse width, while the left side should be expanded by a PRI. There are now 2 (or i+1) pulses contained within the window, from different dwells, at the edges of the scan window, which now has size equal to Equation 4.5. If the window now is slid slightly to the right, the pulse at the leftmost edge will no longer be contained, but the pulse on the right side will be. Meaning, there are now i pulses in the scan window. The same would be true if the window was slid towards the left instead. No matter how the window is shifted, there will always be (at least) i pulses within the scan window.

References

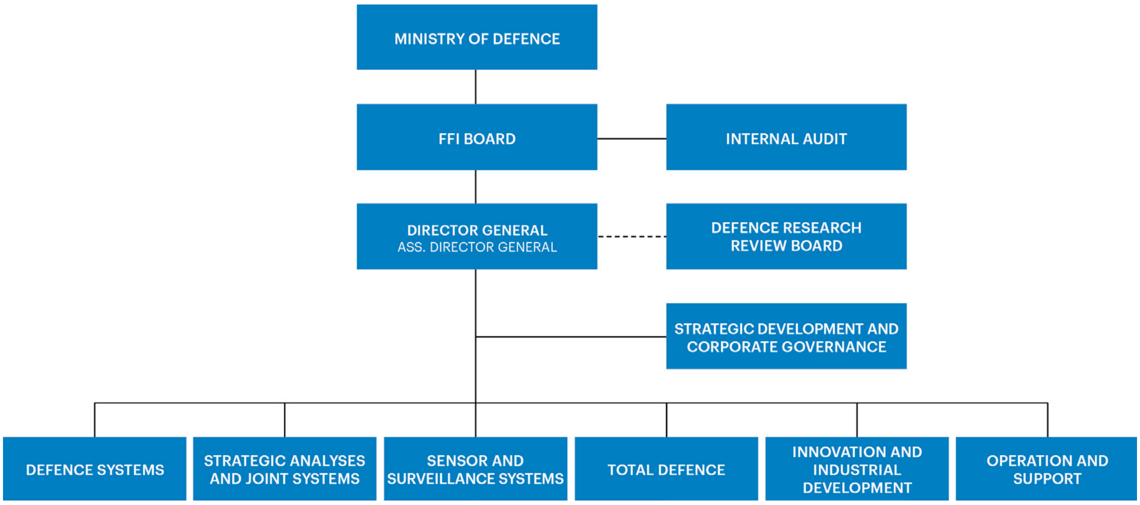
1. Eiben, A.E. and J.E. Smith, *Introduction to Evolutionary Computing*, in *Natural Computing Series,*. 2015, Springer Berlin Heidelberg : Imprint: Springer,: Berlin, Heidelberg.
2. Richards, M.A., et al., *Principles of modern radar*. 2010, SciTech Pub.: Raleigh, NC.
3. Goldberg, D.E., *Genetic algorithms in search, optimization, and machine learning*. 1989, Reading, Mass.: Addison-Wesley Pub. Co.
4. Shukla, A., H.M. Pandey, and D. Mehrotra, *Comparative Review of Selection Techniques in Genetic Algorithm.* 2015 1st International Conference on Futuristic Trends on Computational Analysis and Knowledge Management (Ablaze), 2015: p. 515–519.
5. Kim, I.Y. and O.L. de Weck, *Variable chromosome length genetic algorithm for progressive refinement in topology optimization.* Structural and Multidisciplinary Optimization, 2005. 29(6): p. 445–456.
6. Umbarkar, D.A. and P. Sheth, *Crossover operators in genetic algorithms: a review.* ICTACT Journal on Soft Computing (Volume: 6 , Issue: 1), 2015.

About FFI
The Norwegian Defence Research Establishment (FFI) was founded 11th of April 1946. It is organised as an administrative agency subordinate to the Ministry of Defence.

FFI's mission
FFI is the prime institution responsible for defence related research in Norway. Its principal mission is to carry out research and development to meet the requirements of the Armed Forces. FFI has the role of chief adviser to the political and military leadership. In particular, the institute shall focus on aspects of the development in science and technology that can influence our security policy or defence planning.

FFI's vision
FFI turns knowledge and ideas into an efficient defence.

FFI's characteristics
Creative, daring, broad-minded and responsible.



Forsvarets forskningsinstitutt (FFI)
Postboks 25
2027 Kjeller

Besøksadresse:
Kjeller: Instituttveien 20, Kjeller
Horten: Nedre vei 16, Karljohansvern, Horten

Telefon: 91 50 30 03
E-post: post@ffi.no
ffi.no

Norwegian Defence Research Establishment (FFI)
PO box 25
NO-2027 Kjeller
NORWAY

Visitor address:
Kjeller: Instituttveien 20, Kjeller
Horten: Nedre vei 16, Karljohansvern, Horten

Telephone: +47 91 50 30 03
E-mail: post@ffi.no
ffi.no/en