ON OPTIMAL RESOURCE ALLOCATION IN PHASED ARRAY RADAR SYSTEMS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

AYHAN IRCI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

SEPTEMBER 2006

Approval of the Graduate School of Natural and Applied Sciences

Prof. Dr. Canan ÖZGEN
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. İsmet ERKMEN
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Prof. Dr. Buyurman BAYKAL
Co-Supervisor

Asst. Prof. Dr. Afşar SARANLI
Supervisor

Examining Committee Members

Prof. Dr. Kemal LEBLEBİCİOĞLU	(METU, EE)	_______________
Asst. Prof. Dr. Afşar SARANLI	(METU, EE)	_______________
Prof. Dr. Buyurman BAYKAL	(METU, EE)	_______________
Assoc. Prof. Dr. T. Engin TUNCER	(METU, EE)	_______________
Asst. Prof. Dr. Yakup S. ÖZKAZANÇ	(Hacettepe University, EE)	_______________

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: Ayhan IRCI

Signature :

ABSTRACT

ON OPTIMAL RESOURCE ALLOCATION IN PHASED ARRAY RADAR SYSTEMS

Ircı, Ayhan
M.Sc., Department of Electrical and Electronics Engineering
Supervisor: Asst. Prof. Dr. Afşar Saranlı
Co-Supervisor: Prof. Dr. Buyurman Baykal

September 2006, 106 pages

In this thesis, the problem of optimal resource allocation in real-time systems is studied. A recently proposed resource allocation approach called Q-RAM (Quality of Service based Resource Allocation Model) is investigated in detail. The goal of the Q-RAM based approaches is to minimize the execution speed in real-time systems while meeting resource constraints and maximizing total utility. Phased array radar system is an example of a system in which multiple tasks contend for multiple resources in order to satisfy their requirements. In this system, multiple targets are tracked (each a separate task) by the radar system simultaneously requiring processor and energy resources of the radar system. Phased array radar system is considered as an illustrative application area in order to comparatively evaluate the resource allocation approaches. For the problem of optimal resource allocation with single resource type, the Q-RAM algorithm appears incompletely specified, namely it does not have a termination criteria set that can terminate the algorithm in all possible cases. In the present study, first, the Q-RAM solution approach to the radar resource allocation problem with single resource type is extended to give a global optimal solution in all possible termination cases. For the case of multiple resource types, the Q-RAM approach can only generate near-optimal results. In this thesis, for the formulated radar resource allocation problem with multiple resource types, the Methods of Feasible Directions are considered as an alternative solution approach. For the multiple resource type case, the performances of both the Q-RAM approach and the Methods of Feasible Directions are investigated in terms of optimality and convergence speed with the help of Monte-Carlo simulations. It is observed from the results of the simulation experiments that the Gradient Projection Method produce results outperforming the Q-RAM approach in closeness to optimality with comparable execution times.

Keywords: Optimal Resource Allocation, Real-time Systems, Phased Array Radar, Q-RAM, Methods of Feasible Directions

ÖZ

FAZ DİZİLİ RADAR SİSTEMİNDE OPTİMAL KAYNAK PAYLAŞIMI

Ircı, Ayhan
Yüksek Lisans, Elektrik-Elektronik Mühendisliği
Tez Danışmanı: Yrd. Doç. Dr. Afşar Saranlı
Ortak Tez Danışmanı: Prof. Dr. Buyurman Baykal

Eylül 2006, 106 sayfa

Bu çalışmada, gerçek-zamanlı sistemlerde optimal kaynak paylaşımı problemi üzerinde durulmuştur. Yakın bir dönemde önerilmiş bir kaynak paylaştırma modeli olan Q-RAM (Quality of Service based Resource Allocation Model) yaklaşımı incelenmiştir. Q-RAM yaklaşımında amaç, gerçek-zamanlı sistemlerde, kaynak kısıtlarına uygun ve toplam kaliteyi maksimize edecek şekilde çalışma zamanını minimize etmektir. Faz dizili radar sistemi, birden fazla uygulamanın sistem kaynaklarına gereksinim duyduğu bir sisteme örnektir. Bu sistemde birden fazla hedef, radar sistemi tarafından aynı anda takip edilmekte ve takip görevleri radar sisteminin işlemci ve enerji kaynaklarını kullanmaktadır. Bu çalışmada, kaynak paylaşımı yaklaşımlarını değerlendirmek amacıyla faz dizili radar sisteminde kaynak paylaşımı problemi incelenmiştir. Sadece bir kaynak tipinin değişken olarak incelendiği kaynak paylaşımı problemlerinde, Q-RAM yaklaşımında bazı olası durumlar için belirli bir bitiş kriteri önerilmemektedir. Bu çalışmada, öncelikle, tek değişkenli radar kaynak paylaşımı probleminde Q-RAM çözüm yaklaşımı bütün olası durumlar için optimal çüzümü verecek şekilde geliştirilmiştir. Birden fazla kaynak tipinin değişken olarak incelendiği kaynak paylaşımı problemleri için Q-RAM çözümü yaklaşık optimal sonuçlar üretmektedir. Bu çalışmada, birden fazla kaynak tipinin değişken olarak incelendiği radar kaynak paylaşımı problemi için Q-RAM yaklaşımının yanısıra verimli yönler yöntemleri çözüm yaklaşımı olarak incelenmiştir. Q-RAM ve verimli yönler yöntemlerinin performansları, Monte-Carlo simülasyon tekniğinden yararlanarak çözüme ulaşma hızı ve çözümün optimale yakınlığı bakımından karşılaştırılmıştır. Yapılan simülasyonlar sonucunda verimli yönler yöntemlerinin, Q-RAM yaklaşımı ile yaklaşık aynı çalışma zamanlarında daha iyi sonuçlar verdiği gözlenmiştir.
Anahtar Kelimeler: Optimal Kaynak Paylaşımı, Gerçek-zamanlı Sistemler, Faz Dizili Radar, Q-RAM

ACKNOWLEDGMENTS

I would like to thank Asst. Prof. Dr. Afşar Saranlı and Prof. Dr. Buyurman Baykal for their valuable supervision, advice and criticism throughout the development and improvement of this thesis.

I would also like to express my gratitude to Havelsan Inc. for facilities provided for the completion of this thesis.

I would like to extend my special appreciation to my family for their encouragement they have given me not only throughout my thesis study but also throughout my life.

TABLE OF CONTENTS

PLAGIARISM……………………………………………………………………………iii

ABSTRACT……………………………………………………………………………iv

ÖZ………………………………………………………………………………………v

ACKNOWLEDGMENTS…………………………………………………………...…vi

TABLE OF CONTENTS……………………………………………………...….…vii

LIST OF FIGURES……………………………………………………………….......ix

LIST OF TABLES……………………………………………………………………....x

NOTATIONS AND ABBREVIATIONS…………………………………….....................xi

CHAPTERS

1. INTRODUCTION 1

   1.1 OVERVIEW 1
   1.2 CONTRIBUTION OF THE THESIS 3
   1.3 OUTLINE OF THE THESIS 4

2. RESOURCE ALLOCATION IN PHASED ARRAY RADAR SYSTEM....................... 6

   2.1 RADAR RESOURCE MODEL 9
      2.1.1 Radar Timing Constraints and Schedulability 9
      2.1.2 Radar Power Constraints 10
      2.1.3 Minimum Resource Requirements 11
   2.2 THE OBJECTIVE FUNCTION 12
   2.3 FORMULATION OF THE RADAR RESOURCE ALLOCATION PROBLEM 16

3. RESOURCE ALLOCATION WITH Q-RAM BASED METHODS 18

   3.1 A LITERATURE SURVEY OF Q-RAM APPROACH 18
   3.2 THE DEFINITION AND OBJECTIVE OF Q-RAM APPROACH 20
      3.2.1 The Definition of the Model 20
      3.2.2 The Assumptions of the Model 21
      3.2.3 The Objective of the Model 22
   3.3 APPROACH FOR THE CASE WITH SINGLE RESOURCE TYPE 22
      3.3.1 Application of the Approach to the Radar Resource Allocation Problem..... 26
      3.3.2 Simulations for Run Time Measurement 31
   3.4 APPROACH FOR THE CASE WITH MULTIPLE RESOURCE TYPE 34
      3.4.1 Extension to the Specific Two Resource Type Case 34
      3.4.2 Extension to the General Multiple Resource Type Case 38
   3.5 DRAWBACKS OF THE APPROACHES FOR MULTIPLE RESOURCE TYPE CASE..... 42

4. RESOURCE ALLOCATION WITH METHODS OF FEASIBLE DIRECTIONS 45

4.1 ZOUTENDIJK ALGORITHM 45
4.1.1 Theory 45
4.1.2 Algorithmic Approach 47
4.2 GRADIENT PROJECTION ALGORITHM 49
4.2.1 Theory 49
4.2.2 Algorithmic Approach 50
4.3 CONVEX-SIMPLEX ALGORITHM 52
4.3.1 Theory 52
4.3.2 Algorithmic Approach 52
4.4 APPLICATION OF THE METHODS TO THE RADAR RESOURCE ALLOCATION PROBLEM 54
4.5 APPROACH FOR THE CASE WITH DISCRETE OBJECTIVE FUNCTIONS 55

5. EXPERIMENTAL EVALUATION 57

5.1 OUTLINE OF THE CHAPTER 58
5.2 A SIMULATED RADAR TARGET TRACKING SCENARIO 59
5.2.1 Target Spectrum 59
5.2.2 Target Sub-selection 61
5.2.3 Discrete Profile Generation 63
5.2.4 Monte-Carlo Simulations 65
5.3 PERFORMANCE MEASURES 66
5.3.1 Closeness to the Optimal 66
5.3.2 Speed of Reaching a Solution 67
5.3.2.1 Measurement of the Execution Time 68
5.3.2.2 Termination Criterion of the Algorithms 69
5.4 COMPARATIVE SIMULATIONS WITH CONTINUOUS OBJECTIVE FUNCTIONS 78
5.4.1 Results of the Zoutendijk Algorithm with Topkis-Veinott's Modification 78
5.4.2 Results of the Gradient Projection Algorithm 80
5.4.3 Results of the Convex-Simplex Algorithm 83
5.4.4 Results of the Q-RAM Based Approach 86
5.5 COMPARATIVE SIMULATIONS WITH DISCRETE OBJECTIVE FUNCTIONS 90
5.6 SUMMARY 95

6. CONCLUSION AND FUTURE WORK 97

6.1 CONTRIBUTIONS 97
6.2 FUTURE WORK 99

REFERENCES 101

APPENDIX 104

LIST OF FIGURES

FIGURES

Figure 1.3.1 Radar System.. 6
Figure 1.3.2 Phased Array Radar 7
Figure 1.3.3 Tracking Operation 8
Figure 2.1.1 Radar Dwell. 9
Figure 2.2.1 Tracking Quality Variation with the Sampling Frequency Variation 13
Figure 2.2.2 Division of the Surveillance Space into Regions 14
Figure 2.2.3 Tracking Quality Functions. 15
Figure 3.4.1 Discrete Tracking Quality Functions Depending on Sampling Frequency (f_k) and Defined for Different Tracking Algorithms. 35
Figure 3.4.2 Merged Tracking Quality Function. 36
Figure 3.4.3 Convex Hull of the Merged Tracking Quality Function 36
Figure 3.4.4 Discrete Tracking Quality Functions for kth Tracking Task (Q_k) for the Case of two (P_k, C_k) Options. 39
Figure 3.4.5 Tracking Quality Function Depending on Resource Vector ($\mathbf{O}_k$) 40
Figure 3.4.6 Discrete Tracking Quality Function. 41
Figure 3.5.1 Obtaining Convex Hull of a Discrete Function. 43
Figure 5.2.1 Continuous Tracking Quality Function. 63
Figure 5.2.2 Discrete Tracking Quality Function. 64
Figure 5.3.1 Making MATLAB a Real-Time Application in Windows OS 68
Figure 5.3.2 Variation of the norm of $\mathbf{N}_k$ vector with iterations of the Zoutendijk Algorithm with Topkis-Veinott's Modification, in Logarithmic Scale. 72
Figure 5.3.3 Variation of the norm of $\mathbf{N}_k$ vector with iterations of the Gradient Projection Algorithm presented in Logarithmic Scale. 73
Figure 5.3.4 Variation of the norm of $\mathbf{N}_k$ vector with iterations of the Convex-Simplex Algorithm, in Logarithmic Scale. 74
Figure 5.3.5 Variation of Optimization Variables of Task 01 for the Zoutendijk Algorithm with Topkis-Veinott's Modification. 75
Figure 5.3.6 Variation of Optimization Variables of Task 01 for the Gradient Projection Algorithm. 76
Figure 5.3.7 Variation of Optimization Variables of Task 01 for the Convex-Simplex Algorithm. 77
Figure 5.4.1 Run Time of Topkis-Veinott's Modification Algorithm 79
Figure 5.4.2 Run Time of Gradient Projection Algorithm 82
Figure 5.4.3 Run Time of Convex-Simplex Algorithm 84
Figure 5.4.4 Comparison of Run Times of the Gradient Projection Algorithm and the Convex-Simplex Algorithm. 85
Figure 5.4.5 Run Time of Q-RAM Algorithm 88
Figure 5.4.6 Comparison of Run Times of the Q-RAM Approach and the Gradient Projection Algorithm. 90
Figure 5.5.1 Generation of Discrete Tracking Quality Functions. 91
Figure 5.5.2 Change of Optimality Percentage with Increase of Number of Targets for Different Resource Allocation Approaches. 93

LIST OF TABLES

TABLES

Table 3.3.1 Input Data for Run Time Measurement of Modified Q-RAM for Single Resource Type Case. 32
Table 3.3.2 Mean Run Time of Modified Q-RAM for Single Resource Type Case for Different Number of Tracking Tasks Included in the Simulation Scenario. 32
Table 3.3.3 Run Time of the Modified Q-RAM Approach. 33
Table 5.2.1 Input Data for the Simulations Performed 60
Table 5.2.2 Numbers of Total Combinations for Different Number of Tasks 61
Table 5.2.3 Number of Discrete Operating Points Depending on Number of Tasks 65
Table 5.3.1 Parameters of the Simulation Scenario for Determining the Termination Criterion. 73
Table 5.3.2 Convergence Iterations Optimization Variables of Task 02, Task 03, Task 04 and Task 05 of Table 5.2.1. 77
Table 5.4.1 Run Time of Topkis-Veinott's Modification Algorithm 79
Table 5.4.2 Number of Iterations of Topkis-Veinott's Modification Algorithm 80
Table 5.4.3 Run Time of Gradient Projection Algorithm 81
Table 5.4.4 Number of Iterations of Gradient Projection Algorithm 82
Table 5.4.5 Run Time of Convex-Simplex Algorithm 83
Table 5.4.6 Number of Iterations of Convex-Simplex Algorithm 85
Table 5.4.7 Run Time of Q-RAM Algorithm 87
Table 5.4.8 Number of Iterations of Q-RAM Algorithm 89
Table 5.4.9 Optimality Percentage of the Result of the Q-RAM Algorithm 89
Table 5.5.1 Run Time of FDRA-D and Q-RAM Approaches. 94
Table 5.5.2 Variation of Optimality Percentage with Variation of RMS of Noise. 94

NOTATIONS and ABBREVIATIONS

α_k	Sensitivity of the tracking quality to the sampling frequency change.
A_k	Average power of a pulse transmitted to *k*th target.
β_k	Sensitivity of the tracking quality to the average power of the transmitted radar signal change.
C_k	Computation time of the tracking algorithm of *k*th tracking task.
f_k	Sampling frequency of *k*th tracking task.
f_{ki}	*i*th discrete sampling frequency level for *k*th tracking task.
$f_{k,min}$	Minimum sampling frequency requirement for *k*th tracking task.
FDRA-D	Feasible directions based resource allocation approach for discrete objective function case.
K_i	*i*th application in the system in which Q-RAM is introduced.
KKT	Karesh-Kuhn-Tucker.
m_k	Control value of minimum achievable performance.
M_k	Marginal return of *k*th tracking task.
$\mathbf{N}_k$	Normalized difference of the operating point vector between successive iterations.
$\mathbf{O}_k$	Operating point vector (Resource vector) of *k*th tracking task in which f_k, C_k and P_k are scalar components.
P_k	Average power of the transmitted radar signal for *k*th tracking task.
P_{ki}	*i*th discrete average power of the transmitted radar signal level for *k*th tracking task.
$P_{k,min}$	Minimum average power of the transmitted radar signal requirement for *k*th tracking task.
$P_{\max}$	Maximum average power that can be supplied by the radar system.
Q-RAM	Quality of service based resource allocation model.
q_{kij}	Tracking quality obtained from the *k*'th tracking task when the sampling frequency and average power of the transmitted radar signal of the task are f_{ki} and P_{kj}, respectively.
$Q_k(f_k)$	Discrete tracking quality function of *k*th tracking task depending on sampling frequency.

$Q_k(f_k, C_k, P_k)$	Discrete tracking quality function of *k*th tracking task depending on sampling frequency, computation time and average power of the transmitted radar signal.
$Q(\mathbf{O}_k)$	Discrete tracking quality function of *k*th tracking task depending on $\mathbf{O}_k$.
$Q_k(U_k)$	Discrete tracking quality function of *k*th tracking task depending on utilization of the radar processor.
OP	Optimality percentage.
QoS	Quality of service.
$\mathbf{R}_i$	*i*th type of resource in the system in which Q-RAM is introduced.
$\mathbf{R}^i$	Allocated resource to application K_i in the system in which Q-RAM is introduced.
R_i^{min}	Minimum resource requirement of application K_i in the system in which Q-RAM is introduced.
$R_{i,j}$	The portion of resource $\mathbf{R}_j$ allocated to application K_i in which Q-RAM is introduced.
S	Sum of the weighted application utility of the applications in which Q-RAM is introduced.
S_i	Application utility of application K_i in the system in which Q-RAM is introduced.
TQ	Total tracking quality obtained from the result of a resource allocation algorithm relative to the minimum resource requirement point.
TQ^{opt}	Total tracking quality obtained from the global optimum operating point relative to the minimum resource requirement point.
t_{wk}	Wait time of a pulse transmitted to *k*th target.
t_{rk}	Receive time of a pulse transmitted to *k*th target.
T_k	Sampling period of *k*th tracking task.
t_{xk}	Transmission time of a pulse transmitted to *k*th target.
U_k	Utilization of the radar processor for *k*th tracking task.
U_k^P	Power utilization of the radar system for *k*th tracking task.
w_i	Weight (relative importance) of application K_i in the system in which Q-RAM is introduced.
V_k	Tracking quality function of *k*th tracking task.

CHAPTER 1

INTRODUCTION

1.1 Overview

The problem of resource allocation in real-time systems, in which there exists multiple applications contending for the same resources, has been recently an active research topic area. In general, it is accepted that service qualities of the applications, where service quality is the degree of satisfaction of the end-user, increase with the increase of the amount of resource allocated to that application. During the resource allocation process, the intention is to allocate resources to applications such that the overall service quality obtained from the collection of these applications (i.e., the overall performance of the system for the intended task domain) is maximized. It should be clear that it is not possible to allocate arbitrary amount of resources to applications without any constraint. There exist some limitations on the resources of the system and the problem can be formulated under the formalism of constrained optimization.

A phased array radar system can be considered as a system with multiple applications and multiple resources. In this system multiple targets can be concurrently tracked. In order for the radar system to track all the targets successfully, processor and energy resources of the system should be allocated to individual tracking tasks in an appropriate manner. Recently in a series of work including [16], [18] [19], [22] and [24], this problem is investigated and a family of solution approaches based on a model called Quality of Service based Resource Allocation Model (Q-RAM), are presented. In this thesis, the solution approaches proposed therein for the radar resource allocation problem are further studied and proposals are made to improve the solution approaches in terms of optimality and real-time performance. These proposals are validated by means of simulation studies complying with a radar resource model from the literature [16], [18] and [19].

The Q-RAM approach is first presented by Rajkumar et al. [6]. In [6], a system with multiple concurrent applications is considered. Rajkumar et al. [6] proposed to maximize the total service quality and presented a resource allocation algorithm for the case of single resource type and single QoS (Quality of service) dimension. In [8], they make an attempt to improve the solution approach in [6] to handle the case with multiple resource type with the assumption that the utility functions are min-linear-max (i.e. The min-linear-max function is a function which is linear from the minimum resource requirement to the maximum resource requirement beyond which it becomes flat). In [7], Lee et al. proposed to support discrete QoS operating points and in order to measure the QoS quantitatively a QoS management system is developed in which a numerical mapping is developed for the quality dimensions that are non-numeric. Also, no assumptions about the concavity of the utility functions are made in [7] and the problem of maximizing system utility by allocating a single finite resource to satisfy the QoS requirements of multiple applications is investigated.

In [10], an improvement is proposed to the approach of [7] by extending to the problem of apportioning multiple finite resources to satisfy the QoS needs of multiple applications and the optimization problem for the case of discrete QoS operating points is considered. An algorithm that yields near-optimal results but can execute at potentially much higher speeds is presented. The approach of the presented algorithm is similar to the approach of Lee et al. [7]. The so called 'resource vector approach' is proposed as a new approach in order to handle the discrete QoS and multiple resource case.

In [16], Lee et al. proposes to solve a radar resource allocation problem by using the Q-RAM approach developed in the previous studies [6] [8] [7] [10]. In [16], radar resource allocation problem for the case of two resource types (computation time (C_k) and sampling frequency (f_k)) is considered and a Q-RAM based near-optimal algorithm is presented. In subsequent work of [18], [19], [22] and [24], the Q-RAM based solution of [16] is proposed to be extended to the radar resource management problem for the case of general multiple resource type. In [18], [19] and [22], radar heat constraints on radar antennas and the global energy resource and computational resource from the radar processor are investigated as resources in the radar system. The resource vector approach presented by Lee et al. [10] is used in [18], [19] and [22]. Each scalar element of the resource vector represents the demand on a particular resource. Therefore, each resource vector defines a discrete operating point for each task in the radar system.

Similar to [18], [19] and [22], computational and energy resources of the radar system are considered in [24]. The sampling frequency (f_k), computation time (C_k) and average power of the transmitted radar signal (P_k) of a radar tracking task (which are explained in detail in subsection 2.1) are investigated as optimization variables of the radar resource allocation problem. A Q-RAM based resource allocation approach similar to the approaches of [18] and [19] is proposed as a solution to the formulated radar resource allocation problem in [24].

In the present work, we identify, as a result of the analysis of the Q-RAM based resource allocation approaches; that the theoretical background of the Q-RAM approach for the case of both single resource type and multiple resource type has deficiencies and propose to alleviate some of these deficiencies. In the next subsection, the contribution of the thesis along these lines is presented.

1.2 Contribution of the Thesis

The contributions of the thesis can be stated as follows:

- As a theoretical contribution, the Q-RAM algorithmic approach for the case of single resource type [6] is improved in order to generate optimal results in all of the possible termination cases.
- Performance of the Q-RAM approach for the multiple resource type case is compared with the Methods of Feasible Directions in terms of closeness to optimal and speed of reaching a solution by means of systematic simulation experiments.
- It is shown, through experimental study, that for the case of multiple resource type, the considered constrained optimization methods belonging to the Methods of Feasible Directions category from the well established optimization literature result in optimal solution with convergence speed matching the Q-RAM approach while the latter is a non-optimal optimization approach and does not provide a well founded mathematical background.

As it is explained in Chapter 3, for both of the single and multiple resource type cases, the Q-RAM based approaches do not have a well formulated theoretical background. For the case of resource allocation problem with single resource type, the algorithm is incompletely specified. In some of the possible cases encountered during the execution of the algorithm, the Q-RAM algorithm does not specify the required steps for the continuation of the optimization algorithm [6]. If the algorithm is directly terminated in these cases, this leads to non-optimal solutions. In the present thesis, for the radar resource allocation problem formulated in subsection 3.3; Q-RAM based single resource type resource allocation algorithm of is modified and an optimal resource allocation approach to the radar resource allocation problem is proposed.

The goal of the Q-RAM based approaches is to minimize the execution speed in real-time systems while meeting resource constraints and maximizing total utility [18], [19], [22] and [24]. In the Q-RAM algorithmic approach for the case of multiple resource type, sufficient conditions for optimality are not considered and near-optimal algorithms are presented for the resource allocation. In order to obtain a theoretically sound and optimal resource allocation approach for the radar resource allocation problem, a family of algorithms called Methods of Feasible Directions, which propose optimization algorithms for the constrained optimization problems with non-linear objective functions, are considered in this thesis. It is shown that by using the Methods of Feasible Directions, it is possible to reach an optimal solution with convergence speed closely matching the Q-RAM approach while guaranteeing an optimal solution. It is also observed that the sub-optimal solution of Q-RAM degrades significantly with the growth of the problem size (number of targets being tracked by the radar system) leading to a quantifiable advantage of the proposed approaches based on Methods of Feasible Directions for the considered task domain.

1.3 Outline of the Thesis

Chapter 2 introduces the phased array radar system in which the resource allocation problem is considered. The resources of the radar system as well as the constraints on these resources are explained. The radar resource allocation problem, for which the Q-RAM based approach and the considered algorithms from the Methods of Feasible Directions category are applied, is formulated and the objective function of the optimization problem is presented.

In Chapter 3, a literature survey on Q-RAM approach is presented and the limited theoretical background of the model is introduced. In this chapter, first, the resource allocation approach for the case of single resource type is investigated and the improvement on the approach for the radar resource allocation problem is explained. Second, the resource allocation approach for the multiple resource type case is presented and drawbacks of the Q-RAM approach are described

In Chapter 4, the algorithms in the literature of the Methods of Feasible Directions [4], which are investigated as alternate solutions to the radar resource allocation problem in this thesis, are presented. Based mainly on [4], the theoretical background of these algorithms are briefly described. The optimization algorithms investigated in this chapter are:

- Zoutendijk Algorithm
- Gradient Projection Algorithm
- Convex-Simplex Algorithm

In Chapter 5, the experimental methodology that is used in order to make systematic comparative simulations on Q-RAM and the Methods of Feasible directions for the radar resource allocation problem is explained. The simulated radar target tracking scenario, which can be considered as an input database for the simulations, is presented and the performance measures related to

- Closeness to the Optimal Solution
- Speed of Reaching a Solution

are explained. In the same chapter, performance results of the optimization approaches are also presented.

Finally in Chapter 6, the conclusions of the thesis and proposals for the improvements on the present study, which can be investigated as a future work, are presented.

CHAPTER 2

RESOURCE ALLOCATION IN PHASED ARRAY RADAR SYSTEM

A phased array radar system is composed of two parts. These are the radar processor and the radar antenna. Radar commands are generated in the radar processor. According to these commands antenna part transmits energy at assigned angles and with assigned waveforms. Based on the results of the processing operations on the echo signals; radar processor declares new detections, initiates tracks and maintains tracks on assigned targets [16].

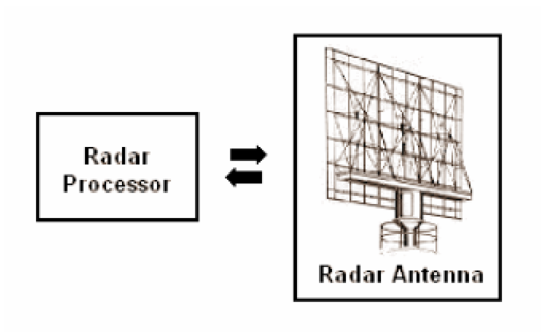

Figure 1.3.1 Radar System. A radar system is composed of two parts; radar antenna transmits the radar signals to the targets with command of the radar processor. Radar processors schedules radar tasks, processes echo signals, decides detections, initiates tracks and maintains tracks on assigned targets.

The antenna in a phased-array radar system can have multiple beams and electronically steer the beams in desired directions. By this way, the phased array radar system can simultaneously track multiple targets depending on distance, acceleration, and other characteristics of targets such as speed, acceleration etc. [16]. The main tasks of the radar system are search and tracking of targets. Usually, there are multiple search tasks that cover the entire angular range of the radar. There is one tracking task corresponding to each target of interest.

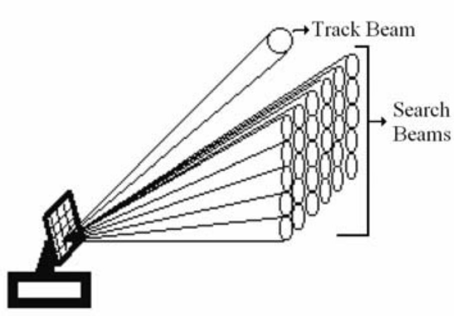

Figure 1.3.2 Phased Array Radar

The search task periodically scans the entire surveillance space to detect the appearance of new targets. Once a new target is detected, a confirmation task is created to identify the type of target. When it is identified, a track task is created and starts tracking the target until it leaves the field of view of the radar system or is destroyed [16]. After the creation of the track task to track a target object, radar system periodically samples the target location and estimates the next location with a particular sampling frequency. Figure 1.3.3 illustrates the periodic transmission and return processing of radar pulses for a tracking task.

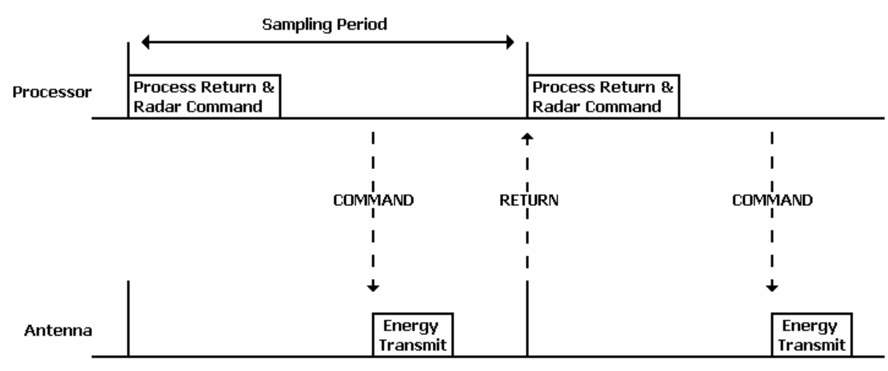

Figure 1.3.3 Tracking Operation. Radar signals are transmitted to the target that is being tracked and estimation algorithms are applied on the echo signals to estimate the next location of the target.

A single period of the process of tracking a particular target consists of sending a radar signal consisting of a series of pulses and receiving the echo of those pulses. This period is known as a *dwell* [16]. In order to appropriately track a target, the dwell needs to have a sufficient number of pulses with a sufficient amount of power on the pulses to traverse through the air, illuminate the target and return back after reflection. The power output of the radar system is limited depending on the power output capability of the energy source.

Based on the received pulses and type of the target that is being tracked, an appropriate estimation algorithm must be used in order to properly estimate the next position of the target. There are many tracking algorithms used in radar systems. Different estimation algorithms result in different tracking performances for different types of targets. Some of the estimation algorithms provide better results than other algorithms in noisy environments and some of the estimation algorithms generate more accurate tracking performance than other algorithms for maneuverable targets [16]. They also have different computational requirements. The execution times of the estimation algorithms on the radar processor vary depending on the computational requirements of the algorithms.

Since a target can maneuver to avoid being tracked, the estimates are valid only for a particular period of time. Based on the processing operations on the echo signals, the time-instant of the next dwell for the tracking task must be determined. Therefore, the tracking task needs to be repeated periodically with a smaller period providing better estimates. For a large sampling period, the estimation error can be so large that the dwell may miss the target. On the other hand, a small sampling period will require higher resource utilization.

As it is explained in the previous paragraphs, radar system requires computational and RF-energy resources in order to maintain an image of selected parts of the air, sea and land activities. Without sufficient amount of computational and RF-energy resources, radar system can not create detections and tracks and can not present the tactical image to the user on a monitor accurately. In the following subsections, the resources of the radar system and constraints on the resources are elaborated. In subsection 2.2 the objective function defined for the resource allocation optimization approach is presented. Then in subsection 2.3, the radar resource allocation problem is formulated based on the explained resources, constraints and the objective function.

2.1 Radar Resource Model

In this thesis, energy and computational resources of the radar system are investigated. In the following subsections, first the constraints on the aforementioned parameters are investigated. Section 2.1.1 considers the computational resources of the radar system and the schedulability condition for the tasks in order to derive the constraint on computational resources of the radar system. Later, the subsequent subsection explains the radar power constraint.

2.1.1 Radar Timing Constraints and Schedulability

A particular target tracking task is accomplished by sending a radar signal consisting of a series of high frequency pulses, receiving the echoes of those pulses and applying appropriate signal-processing algorithms in order to properly estimate the next position of the target. This process is repeated periodically until the tracked target leaves the field of view of the radar system or is destroyed.

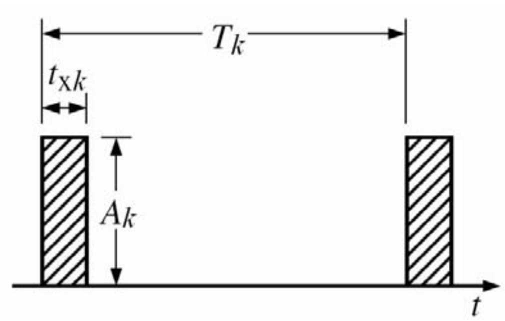

Figure 2.1.1 Radar Dwell. Sending radar signals to a particular target [18].

Assume a tactical environment consisting of N targets which are being tracked by the radar system. In Figure 2.1.1 illustrates the radar dwell for the ith tracking task. The radar dwell is characterized in terms of a transmit power A_k, a transmission time t_{xk}, a wait time t_{wk} and a receive time t_{rk}. T_k is the sampling period. We define f_k as the sampling frequency which is equal to $1/T_k$.

In the sampling period interval, estimation algorithms are applied on the echo signals by the radar processor. Let C_k denote the total execution time of the estimation algorithm for the kth tracking task in a particular sampling period. In the sampling period interval (T_k), the ratio of the computation time (C_k) of a tracking task to the sampling period interval gives the utilization of the radar processor for that particular tracking task (i.e. for the kth task). The utilization of the radar processor for kth tracking task (U_k) ($U_k \in [0,1]$) can be written as follows:

$$U_k = C_k \times f_k \tag{2.1.1}$$

In order for the radar tasks to be scheduled in the radar processor the total utilization of the radar processor should not exceed %100 [16]. Radar timing and schedulability constraint regarding the utilization of the radar processor is:

$$\sum_{k=1}^{N} U_k \leq 1 \tag{2.1.2}$$

which can be written as

$$\sum_{k=1}^{N} C_k f_k \leq 1 \tag{2.1.3}$$

where N is the number of tracking tasks processed by the radar processor.

2.1.2 Radar Power Constraints

In addition to timing constraints, radar system also has power constraints. Power of the transmitted radar signal is limited with the power output capability of the energy source. In

[18], the power utilization of the radar system for a set of N tasks in a particular sampling period interval (T_k) is defined as follows:

$$U^P = \frac{1}{P_{\max}} \sum_{k=1}^{N} A_k \frac{t_{xk}}{T_k} \quad (2.1.4)$$

where $P_{\max}$ is the maximum average power that can be supplied by the radar system without leading any overheating and damage condition. Here, it is considered that the average power is given by the fraction of time each task is transmitting, multiplied by the transmit power for that task. In the expression above, $A_k(t_{xk}/T_k)$ is the average power of the transmitted radar signal in the sampling period T_k for kth tracking task. Let's denote the average power of the transmitted radar signal in the sampling period with P_k for the kth tracking task.

$$P_k = A_k (t_{xk} / T_k) \quad (2.1.5)$$

The total power utilization value of the radar system can not exceed %100 in order for the radar system to operate safely [18]. The power constraint of the radar system can be expressed as follows:

$$\sum_{k=1}^{N} P_k \leq P_{\max} \quad (2.1.6)$$

2.1.3 Minimum Resource Requirements

Position, heading, speed records of the targets which are being tracked are updated at each sampling period. In order for the records of the tracks to be accurate, the sampling period interval should be sufficiently short, i.e. the sampling frequency should be sufficiently high. Minimum sampling frequency requirement of each target changes according to maneuverability, speed, position of the target relative to the ownship where ownship is the platform on which the considered phased array radar system is mounted [16]. Minimum sampling frequency constraint of k^{th} tracking task can be expressed as follows:

$$f_k \geq f_{k,min} \quad (2.1.7)$$

where $f_{k,min}$ is the minimum sampling frequency requirement for the kth tracking task.

In order to appropriately track a target, the transmitted radar signal should have a sufficient amount of power on the pulses to traverse through the air, illuminate the target and return back after reflection. Larger P_k provides better tracking information. The value of P_k required to adequately track a target is proportional to the 4th power of distance between the target and the radar [13]. For each target type and different position and speed of a specific target relative to the ownship, there exist minimum average transmitted power requirements [18]. Minimum requirement on the average power of the transmitted signal of kth tracking task is:

$$P_k \geq P_{k,min} \tag{2.1.8}$$

where $P_{k,min}$ is the minimum requirement on the average power of the transmitted radar signal for the kth tracking task.

2.2 The Objective Function

In [5] and [16] the control system performance variation due to the sampling frequency variation is modeled as an exponential function; Lee et al. [16] use this approach to model the tracking performance variation of the radar system with the sampling frequency variation as an exponential function. As it is explained in [16], in a radar system, the system keeps a record (track) of each target and remembers its current position, heading, speed, etc. as the target moves. The records are updated periodically at sufficiently high frequencies in order to maintain a specified level of confidence in their accuracy. In the exponential model, the accuracy of the record increases with increase of the update frequency. The accuracy improvement by increasing the update frequency is significant at the beginning but becomes only marginal once the accuracy is saturated with a high enough frequency. This exponential behavior is illustrated in Figure 2.2.1.

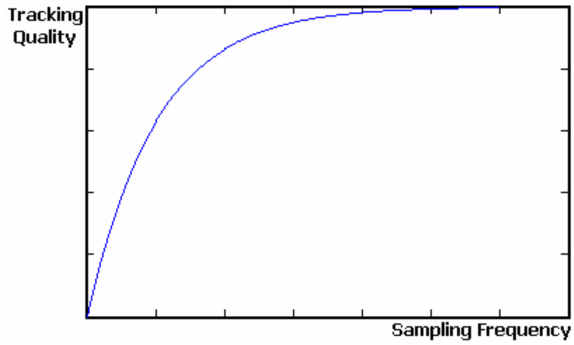

Figure 2.2.1 Tracking Quality Variation with the Sampling Frequency Variation.

The probability of detection can be written as an exponential function of transmission power when noise power and probability of false alarm are in some pre-defined ranges (Refer to the appendix of this thesis for further analysis). Tracking quality is a measure of estimating the next location of the target correctly and we have investigated tracking quality as linearly proportional with the probability of detection. In this thesis, similar to the handling method of change of performance of the system with sampling frequency in [16], [5] and [18]; radar tracking performance depending on the average power of the transmitted radar signal is also investigated as an exponential function. The strength of the echo signal increases with increase of amount of power of the transmitted radar signal. Increase in power of the echo signal provides better signal-to-noise ratio and hence probability of detection of targets increases [13]. Performance increase due to average power of the transmitted radar signal increase is expected to exhibit a saturation characteristic, i.e., the tracking performance increase will gradually saturate after a certain amount of power resource is allocated to that tracking task [19]. The tracking performance, also called tracking quality, for *k*th tracking task can be defined with the following form of exponential function:

$$V_k = (1 - m_k e^{-\alpha_k f_k - \beta_k P_k}) \tag{2.2.1}$$

where V_k is the tracking quality function of the *k*th tracking task depending on sampling frequency (f_k) average power of the transmitted radar signal (P_k). This function is illustrated in Figure 2.2.3. In this formulation, α_k and β_k are the sensitivity of the tracking quality to the sampling frequency and average power of the transmitted radar signal change respectively. The parameters α_k and β_k ideally take different values depending on the speed, maneuverability, distance of the target that is being tracked. The parameter m_k, specify the control value of minimum achievable performance ($m_k \leq 1$) [16]. These parameters can be approximated by observing the average behavior of the radar system in practice runs. Look-up tables of this data can be created and later used.

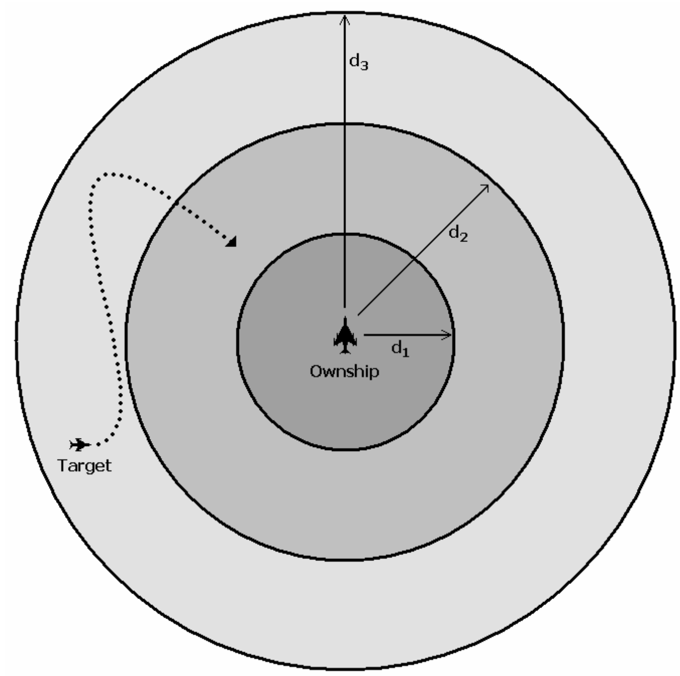

Figure 2.2.2 Division of the Surveillance Space into Regions (Similar to the Figure in [16]). As the shade of the region becomes darker, the region becomes more critical. The distance of the target affects the required sampling frequency and average power of the transmitted radar signal in order to achieve a certain tracking quality.

As shown in the Figure 2.2.2, for a particular target, different α_k and β_k parameter values can be selected in the tracking quality function of the tracking task, according to the region of the target relative to the ownship. For example in the Figure 2.2.2, if the target leaves the region with radius r_3 where $d_3 > r_3 > d_2$ and passes to the region with radius r_2 where $d_2 > r_2 > d_1$, the tracking quality function of the target should be adapted to the new relative position of the target, i.e. the tracking task in the region with radius r_3 should be removed from the task list of the radar system and a new tracking task in the region with radius r_2 should be added to the task list of the radar system with new α_k and β_k parameter values. Intuitively we expect that as the distance of the target to the ownship decreases, the sensitivity of the tracking quality to the change of average power of the transmitted radar signal (β_k) decreases while sensitivity of the tracking quality to the change of sampling frequency (α_k) increases because the change of position of the target relative to the ownship in unit time increases.

Along with the position of the target relative to the ownship, speed and maneuverability plays important role in determination of the α_k and β_k parameter values in the tracking quality function of the target. A target with high speed or maneuvering capability requires higher sampling frequency in order to be tracked accurately; this condition leads to the sensitivity of the tracking quality to the sampling frequency change (α_k) to be higher.

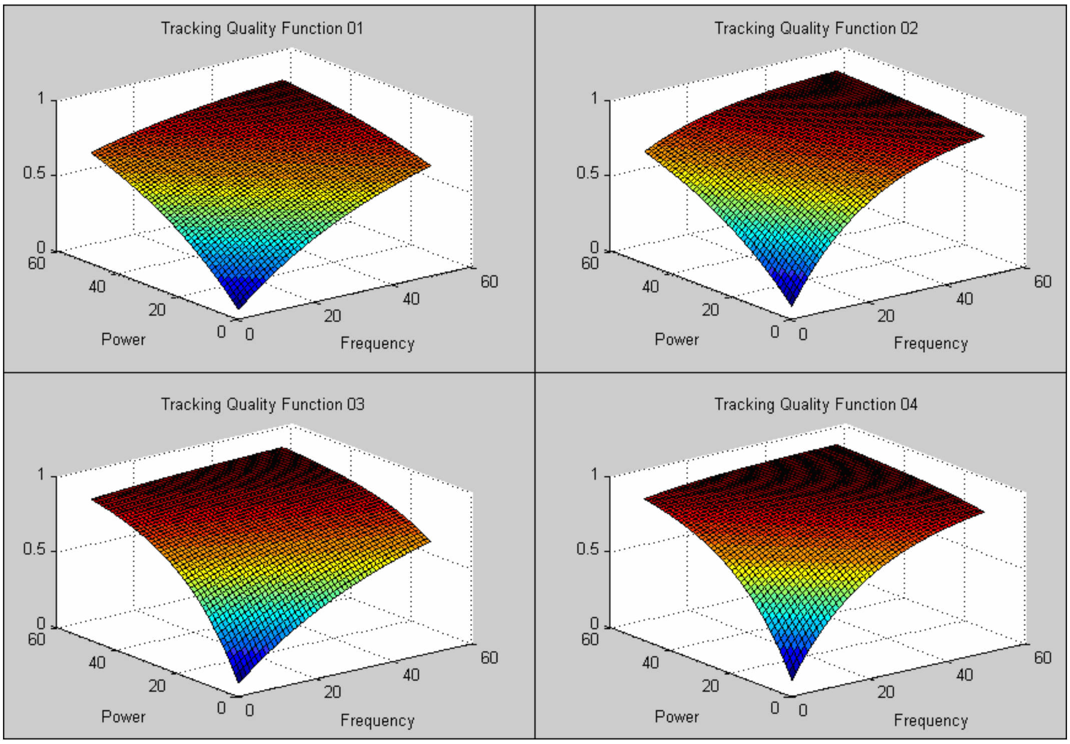

Figure 2.2.3 Tracking Quality Functions. Functions 02, 03 and 04 are obtained with relative changes to 01. In tracking quality function 02, α_k is higher than in 01 with all other parameters the same. In tracking quality function 03, β_k is higher than in 01 with all other parameters the same. And in tracking quality function 04, both of the sensitivity parameter values are higher than in 01 with all other parameters the same.

In Figure 2.2.3, different tracking quality functions are shown for different types of tracking tasks. In the second figure, tracking quality function of a tracking task having higher

α_k parameter value; in the third figure, tracking quality function of a tracking task having higher β_k parameter value and in the fourth figure, tracking quality function of a tracking task having both higher α_k and β_k parameter values compared to the tracking quality function of the first figure are shown. These figures are not plotted using real parameter values, the parameter values in these figures are selected for illustrative purposes.

In [16] the tracking quality functions for different tracking tasks are also considered as discrete functions and Q-RAM is proposed to be applied to the radar resource allocation problem with discrete tracking quality functions. In the present study, for the experimental evaluation of the Q-RAM approach, discrete tracking quality functions are also considered.

2.3 Formulation of the Radar Resource Allocation Problem

The goal of the radar system is to utilize its finite energy and time resources to maximize the quality of tracking. A radar system must make two sets of decisions. First, it must decide what fraction of resources (energy and time) to spend on each target. It must then schedule the radar antenna(s) to allocate the beams and transmit the selected amount of energy through each beam and receive the return echoes in a non-preemptive fashion. Since targets in the sky are continually moving, resource allocation and scheduling decisions must be made on a frequent basis. The radar resource allocation problem which is studied in this thesis can thus be formulated as follows,

Maximize

$$J(f_1, P_1, f_2, P_2, ..., f_N, P_N) = \sum_{k=1}^{N} V_k = \sum_{k=1}^{N} (1 - m_k e^{-\alpha_k f_k - \beta_k P_k}) \qquad (2.3.1)$$

Subject to

$$\sum_{k=1}^{N} C_k f_k \leq 1 \qquad (2.3.2)$$

$$\sum_{k=1}^{N} P_k \leq P_{\max} \qquad (2.3.3)$$

$$f_k \geq f_{k,min}, \qquad k = 1, 2, \ldots, N. \tag{2.3.4}$$

$$P_k \geq P_{k,min}, \qquad k = 1, 2, \ldots, N. \tag{2.3.5}$$

where N is the number of targets that are being tracked, C_k denote the total execution time of the estimation algorithm for the kth tracking task, $P_{\max}$ is the maximum average power that can be supplied by the radar system, $f_{k,min}$ is the minimum sampling frequency requirement for the kth tracking task, $P_{k,min}$ is the minimum average power of the transmitted radar signal requirement for the kth tracking task.

The objective function of the radar resource allocation problem formulated above is continuous, differentiable and concave. The constraints are linear. These properties enable us to use methods of feasible directions, which will be briefly explained in the subsequent chapters, in solving the formulated constrained optimization problem.

CHAPTER 3

RESOURCE ALLOCATION WITH Q-RAM BASED METHODS

In this chapter, QoS-based Resource Allocation Model (Q-RAM) based solutions to the radar resource allocation problem are presented and the model is investigated in detail. Q-RAM assumes a system with multiple concurrent applications, each of which can operate at different levels of quality based on the system resources available to it. The goal of the model is to be able to allocate resources to the various applications such that the overall system utility is maximized under the constraint that each application can meet its minimum needs. In the first subsection, a literature survey of Q-RAM approach is presented and then in the subsequent subsections, the definition and objective of Q-RAM are explained and Q-RAM based algorithmic solution approaches to the radar resource allocation problem and drawbacks of the model are introduced.

3.1 A Literature Survey of Q-RAM Approach

Q-RAM is first presented by Rajkumar et al. [6]. In [6], a system with multiple concurrent applications is assumed and two main constraints are considered: resource consumption can not exceed an upper bound and each application can meet its minimum needs. Based on these constraints, the total system utility is proposed to be maximized and a resource allocation algorithm is presented for the case of single resource type and single QoS (Quality of service) dimension. Rajkumar et al. assumed that the utility functions of each application are nondecreasing, concave and have two continuous derivatives. In [8], they considered the problem of apportioning multiple resources to satisfy a single QoS dimension different from their previous work in [6]. In [8], the utility functions are assumed to be min-linear-max. The optimization problem is defined in a way such that the cost function and constraints becomes linear. It is proposed to apply standard optimization techniques for mixed integer programming in order to obtain optimal solution.

In [6] and [8], Rajkumar et al. assumed continuous QoS dimensions and the utility gained by improvements along a QoS dimension are representable by concave functions. In [7], Lee et al. relax both assumptions. They support discrete QoS operating points. In order to measure the QoS quantitatively a QoS management system is developed. In this structure, a numerical mapping is developed for the quality dimensions that are non-numeric. Therefore the Quality Index is introduced, which maps qualities to indices in order of increasing quality. By analytically planning and allocating resources to multiple applications, it is proposed to maximize the net utility acquired by the end-users. They also make no assumptions about the concavity of the utility functions. Using these as the basis, they tackle the problem of maximizing system utility by allocating a single finite resource to satisfy the QoS requirements of multiple applications.

In [7] Lee et al. studied the problem of maximizing system utility by allocating a single finite resource to satisfy discrete QoS requirements of multiple applications. This study is proposed to be improved in [10]. In [10], Lee et al. focus on the problem of apportioning multiple finite resources to satisfy the QoS needs of multiple applications and deal with the optimization problem for the case of discrete QoS settings. An algorithm that yields near-optimal results but can execute at potentially much higher speeds is presented. The approach of the presented algorithm is similar to the approach of Lee et al. [7]. Resource vector approach is newly proposed in order to handle the discrete QoS and multiple resource case.

In [16], Lee et al. proposed to solve a radar resource allocation problem, where computation time (C_k) and sampling frequency (f_k) are variables of the resource allocation problem, by using the Q-RAM approach [6] [8] [7] [10]. In [16], first the computation time (C_k) is assumed to be fixed and only the sampling frequency (f_k) is assumed to be adjustable. Lee et al. [16] also assumed that the tracking quality function is defined as a continuous convex function and proposed to solve the radar resource allocation problem with these assumptions by using the approach in [5]. After in [16], the computation time (C_k) is also considered as adjustable and assumption about continuity and convexity of the tracking quality function is relaxed; and a near-optimal algorithm based on Q-RAM approach is presented in order to solve the radar resource allocation problem.

In [19], an optimization algorithm for a radar tracking application, based on Q-RAM is presented. Radar heat constraints on radar antennas and global energy source and computational resource from the radar processor are investigated as resources in the radar system. Resource vector approach presented by Lee et al. [10] is used in [19]. Each scalar element of the resource vector represents the demand on a particular resource. Therefore, each resource vector defines a discrete operating point. A resource vector of a task is mapped to a value representing the overall demand on the system for a particular set of resource requirements and tracking quality is investigated as QoS dimension in [19].

In [24] and [20], the Q-RAM based solution of [16], which solves the radar resource allocation with two variables for each tracking task (sampling frequency and computation time), is proposed to be improved and it is proposed to consider not only sampling frequency and computation time but also the average power of the transmitted signal of the radar system for each tracking task as an adjustable parameter. A near-optimal Q-RAM based resource allocation approach is presented in [24] and [20].

3.2 The Definition and Objective of Q-RAM Approach

In this section the mathematical formulation, assumptions and objective of the Quality of Service based Resource Allocation Model (Q-RAM) is presented. Based on the works in [6], [8], [7] and [10], the definition of Q-RAM is presented in the next subsection.

3.2.1 The Definition of the Model

Q-RAM is based on a system in which multiple applications may require access to multiple resource types in order to satisfy requirements. In this system, also an application requires a certain minimum resource allocation to perform acceptably. An application may also improve its performance with larger resource allocations. This improvement in performance is measured by a utility function in Q-RAM. ‘Q-RAM is a model in which resources can be allocated to individual applications with the goal of maximizing a global objective’ [6]. In Q-RAM, it is proposed to satisfy the simultaneous requirements of multiple applications and allow applications access to multiple resources [8]. The characteristics of the considered applications and system in Q-RAM are as follows [8]:

- Each application may have a minimum and/or a maximum need along each QoS dimension.
- An application may require access to multiple resource types.
- Each resource allocation adds some utility to the application and the system, with utility monotonically increasing with resource allocation.

- System resources are limited so that the maximal demands of all applications often cannot be satisfied simultaneously.

Q-RAM is defined as follows. The system consists of n applications $\{K_1, K_2, \ldots, K_n\}$, $n \geq 1$, and m resources $\{\mathbf{R}_1, \mathbf{R}_2, \ldots, \mathbf{R}_m\}$, $m \geq 1$. Each resource $\mathbf{R}_j$ has a finite capacity and can be shared. The portion of resource $\mathbf{R}_j$ allocated to application K_i be denoted by $R_{i,j}$. It is enforced that $\sum_{i=1}^{n} R_{i,j} \leq \mathbf{R}_j$ [6].

The following definitions are introduced:

- The *application utility*, S_i, of an application K_i is defined to be the value that is accrued by the system when K_i is allocated $\mathbf{R}^i = (R_{i,1}, R_{i,j}, \ldots, R_{i,m})$. In other words, $S_i = S_i(\mathbf{R}^i)$. S_i is referred to as the *utility function* of K_i. This utility function defines a surface along which the application can operate based on the resources allocated to it.
- Each application K_i has a relative importance specified by a weight w_i, $1 \leq i \leq n$.
- The *total system utility* $\mathbf{S}(\mathbf{R}^1, \ldots, \mathbf{R}^n)$ is defined to be the sum of the weighted application utility of the applications, i.e. $\mathbf{S}(\mathbf{R}^1, ..., \mathbf{R}^n) = \sum_{i=1}^{n} w_i S_i(\mathbf{R}^i)$.
- Each application K_i needs to satisfy requirements along d QoS dimensions, $d \geq 1$.
- An application, K_i, has minimal resource requirements. These minimal requirements are denoted by $R_i^{min} = \{R_{i,1}^{min}, R_{i,2}^{min}, \ldots, R_{i,m}^{min}\}$ where $R_{i,j}^{min} \geq 0$, $0 \leq j \leq m$. An application, K_i, is said to be *feasible* if it is allocated a minimum set of resources.

In this thesis, we assume that $d = 1$, i.e. only a single QoS dimension, which is tracking quality, is considered. In the following subsections assumptions and objective of the Q-RAM are provided.

3.2.2 The Assumptions of the Model

The assumptions of Q-RAM are as follows [6]:

- The applications are independent of one another.
- The available system resources are sufficient to meet the minimal resource requirements of each application, $\mathbf{R}_i^{min}$, $1 \leq i \leq n$.

- The utility functions S_i are nondecreasing in each of their arguments. And it is assumed that these functions are concave and have two continuous derivatives.
- Each application, K_i, has a weight w_i denoting its relative importance.

If the second assumption does not hold, then the minimal resource requirements cannot be met. If these requirements are not met, then some of the applications must be dropped. Different techniques can be used in order to determine which of the applications should be dropped, or some applications could be allowed to have less than their minimal resource allocations [6]. Although this is a very important issue, it is beyond the scope of this thesis.

In view of the 4th assumption, a weighted utility function for an application as w_iS_i can be defined and then the resource allocation problem for those weighted utility functions can be solved. Thus, the weights can be removed from the allocation problem. In this study, these weighted utilities are used and the weights are dropped.

3.2.3 The Objective of the Model

Based on the definitions and assumptions given in the subsections, the objective of Q-RAM is to make resource allocations to each application such that the total system utility is maximized under the constraint that every application is feasible. In other words, $\{R_{i,j}, 1 \leq i \leq n, 1 \leq j \leq m\}$ should be determined such that $R_{i,j} \geq R_{i,j}^{min}$, amount of allocated resources to the applications are not greater than the upper limit value of the system resources and **S** is maximum [6].

As it is explained in section 3.1, first a resource allocation problem with single resource type is investigated [6] and then it is proposed to extend the solution to the resource allocation problems with multiple resource types in Q-RAM literature [8], [7], [10]. In the following subsections first the case with single resource type is examined and then the case with multiple resource types is investigated subsequently.

3.3 Approach for the Case with Single Resource Type

The case of making resource allocation decisions when there is only a single resource type and a single QoS dimension is considered first in Q-RAM approach [6]. Since there is a single resource, the subscripts associated with the resource types are dropped. For this case, the utility functions of the applications become $S_i = S_i(R_i)$, $1 \leq i \leq n$, where R_i is the amount of resource allocated to the application K_i. The minimum resource allocation needed to satisfy K_i is R_i^{min}. As it is indicated in section 3.2.2, all minimal application resource requests can be met; Rajkumar et al. [6] focus on the allocation of the excess resources available.

In the analysis conducted in [6], it is assumed that $R_i^{min} = 0$, $\forall i = 1$to n and the quantity of available resources is reduced by that amount. The goal is to determine the values of $R_1, R_2, \ldots, R_n$ such that the total system utility, $\sum_{i=1}^{n} S_i(R_i)$, is maximized subject to the constraint $\sum_{i=1}^{n} R_i \leq R$. In [6], the following theorem, which provides a necessary condition for an allocation to be optimal, is presented.

Theorem A necessary condition for a resource allocation to be optimal is $\forall i$, $1 \leq i \leq n$, $R_i = 0$ or for any $\{i, j\}$ with $R_i > 0$ and $R_j > 0$, $S_i'(R_i) = S_j'(R_j)$ [6].

$S'(R)$ is the derivative of S with respect to R. The proof this theorem, which is provided in [6], is as follows:

'***Proof*** The result is a standard conclusion of the Kuhn Tucker theorem [4]. To understand the intuition behind the results, suppose that for some $i \neq j$, let $R_i > 0$, $R_j > 0$ and $S_i'(R_i) = S_j'(R_j)$. Since $R_j > 0$, an infinitesimal amount of R can be subtracted from application K_j and added to application K_i. Since $S_i'(R_i) > S_j'(R_j)$, the total system utility will increase. This contradicts the assumption that the allocation was optimal.' [6].

Rajkumar et al. [6] proposed the following algorithm to determine the optimal resource allocation R_i for each application to obtain maximum utilization. It is assumed that each application has already been allocated its minimum resource requirement. By the assumptions in section 3.2.2; sufficient resources should be available for this allocation. The optimal additional allocation to each application, $R_i \geq 0$; $1 \leq i \leq n$, subject to $\sum_{i=1}^{n} R_i \leq R$ is proposed to be determined as follows:

Q-RAM Procedure for Single Resource Type:

1. Let the current allocation of the resource to K_i be R_i, $1 \le i \le n$. Let the unallocated quantity of the available resource be R^l. Compute $(S_1'(R_1), ..., S_n'(R_n))$.
2. Identify
   i. the subcollection of applications with largest value of $S_i'(R_i)$,
   ii. the number of applications in that subcollection (denoted by p),
   iii. the application (denoted by j) with the second largest value of this quantity if any such application exists.
3. If the largest value of $S_i'(R_i)$ is 0, then stop. No further allocation will increase system utility and spare resources are available.
4. Otherwise, increase R_i for each of the members of the subcollection so that their values of $S_i'(R_i)$ decrease but continue to be equal until one of the following is satisfied,
   i. this value becomes equal to the second largest value or,
   ii. the additional resources added to this subcollection equal R^l.
5. If (ii) is satisfied, stop as all resources have been optimally allocated.
6. If (i) is satisfied, one or more new applications should be added to the subcollection. Return to step 1.

For the considered maximization problem which is 'Maximize $\sum_{i=1}^{n} S_i(R_i)$ such that $\sum_{i=1}^{n} R_i \le R$ and $R_i \ge R_i^{min}$, $i = 1, 2, \ldots, n$', the Karesh-Kuhn-Tucker (KKT) conditions are as follows,

$$\begin{bmatrix} -S_1'(R_1) \\ -S_2'(R_2) \\ \cdot \\ \cdot \\ \cdot \\ -S_n'(R_n) \end{bmatrix} + \lambda \begin{bmatrix} 1 \\ 1 \\ \cdot \\ \cdot \\ \cdot \\ 1 \end{bmatrix} - \mu_1 \begin{bmatrix} 1 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix} - \mu_2 \begin{bmatrix} 0 \\ 1 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix} - \ldots - \mu_n \begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 1 \end{bmatrix} = 0 \qquad (3.3.1)$$

$$\sum_{i=1}^{n} R_i \leq R \qquad (3.3.2)$$

$$R_i \geq R_i^{\min}, \quad i = 1, 2, ..., n \qquad (3.3.3)$$

$$\lambda(\sum_{i=1}^{n} R_i - R) = 0 \qquad (3.3.4)$$

$$\mu_i (R_i^{\min} - R_i) = 0, \quad i = 1, 2, ..., n \qquad (3.3.5)$$

$$\lambda \geq 0 \qquad (3.3.6)$$

$$\mu_i \geq 0, \quad i = 1, 2, ..., n \qquad (3.3.7)$$

where λ, μ_1, μ_2, …, μ_n are Lagrangian multipliers [4]. If the objective function and the constraints are convex, the KKT optimality conditions are sufficient conditions for optimality of a solution [4]. In the optimization problem considered for illustration of the Q-RAM approach, both the objective function and the constraints are convex. Therefore, for the resource allocation problem above, a solution satisfying the KKT optimality conditions is the optimal solution.

The described Q-RAM algorithm, as reported in [6], does not have a termination criteria set that can terminate the algorithm in all possible cases. Certain possible cases exist for which the algorithm does not have a termination criterion at all. To terminate the algorithm for these cases, lead to a sub-optimal solution. These cases are summarized below and are routinely possible during the execution of the algorithm.

- **Case 1:** Suppose that resource allocation is done to all of the applications, slopes of the utility functions of the applications ($S_i'(R_i)$, i = 1, 2, …, n) are all equal to λ, $\lambda > 0$, and there is still excess resources i.e. $\sum_{i=1}^{n} R_i < R$.

- **Case 2:** Suppose that resource allocation is done to k applications out of n ($0 < k < n$) and amount of unallocated resources is not enough to make the slopes of the utility functions of the applications, which are allocated resources, to make equal to the second largest slope value, slopes of the utility functions of the applications ($S_i'(R_i)$, i = 1, 2, …, n) are all equal to λ, $\lambda > 0$, and there is still excess resources i.e. $\sum_{i=1}^{n} R_i < R$.
- **Case 3:** Remember that it is assumed that amount of resources is enough to make minimum resource requirement allocation to all of the applications. Suppose that amount of unallocated resources is not enough to make the slopes of the utility functions of the applications, slopes of the utility functions of which have the highest value, to make equal to the second largest slope value and there is still excess resources i.e. $\sum_{i=1}^{n} R_i < R$.

The cases listed above are possible cases which can be encountered during the execution of the algorithm and no suggestions are proposed in the occurrence of these cases in the Q-RAM approach for single resource type case. In these cases if the algorithm is terminated this leads to non-optimal solutions as the condition 3.3.4 is not satisfied. As it is mentioned previously, because the objective function and constraints are convex the KKT conditions are sufficient for optimality; therefore terminating the algorithm in the cases listed above as all of the KKT conditions are not fulfilled.

In order to terminate the Q-RAM procedure in the case of Case 1, Case 2 and Case 3 while satisfying all of the KKT conditions, a modification should be done on the algorithm. By this way, optimal results can be obtained by applying the algorithm. In [23], radar resource allocation problem that is described in Chapter 2 is considered; single resource type resource allocation algorithm of Q-RAM is modified and an optimal resource allocation approach to the radar resource allocation problem that is explained in the next paragraph is presented.

3.3.1 Application of the Approach to the Radar Resource Allocation Problem

In Chapter 2, radar resource allocation problem with two resource types, which are sampling frequency (computational resource) and average power of the transmitted radar signal (energy resource), is explained. In [23], for the illustrative purposes of the modification algorithm on Q-RAM approach for the single resource case, radar resource allocation problem with single resource type, which is sampling frequency, is considered. The radar resource allocation problem in [23] is as follows:

Maximize

$$\sum_{k=1}^{N}(1-m_k e^{-\alpha_k f_k})$$

Subject to

$$\sum_{k=1}^{N} C_k f_k \leq 1$$

$$f_k \geq f_{k,min}, \qquad k = 1, 2, \ldots, N.$$

As it is explained in section 2.2, the tracking performance of the radar system depending on the sampling frequency (f_k) can be formulated with an exponential function. In [5], Seto et al. modeled the control performance of a system depending on sampling frequency with an exponential function and in [16] Lee et al. used this approach and formulated the tracking quality of the radar system with the exponential function which appears in the problem formulation above. In this formulation, N is the number of tracking tasks, α_k is the sensitivity to the sampling frequency change, C_k is the computation time and $f_{k,min}$ is the minimum sampling frequency requirement of the kth tracking task as explained in Chapter 2.

The KKT optimality conditions for the radar resource allocation problem are as follows:

$$\begin{bmatrix} -m_1\alpha_1 e^{-\alpha_1 f_1} \\ -m_2\alpha_2 e^{-\alpha_2 f_2} \\ . \\ . \\ . \\ -m_N\alpha_N e^{-\alpha_N f_N} \end{bmatrix} + \lambda \begin{bmatrix} C_1 \\ C_2 \\ . \\ . \\ . \\ C_N \end{bmatrix} - \mu_1 \begin{bmatrix} 1 \\ 0 \\ . \\ . \\ . \\ 0 \end{bmatrix} - \mu_2 \begin{bmatrix} 0 \\ 1 \\ . \\ . \\ . \\ 0 \end{bmatrix} - \ldots - \mu_n \begin{bmatrix} 0 \\ 0 \\ . \\ . \\ . \\ 1 \end{bmatrix} = \mathbf{0} \qquad (3.3.8)$$

$$\sum_{k=1}^{N} C_k f_k \le 1 \quad (3.3.9)$$

$$f_k \ge f_{k,\min}, \quad k = 1, 2, ..., N \quad (3.3.10)$$

$$\lambda(\sum_{k=1}^{N} C_k f_k - 1) = 0 \quad (3.3.11)$$

$$\mu_k (f_{k,\min} - f_k) = 0, \quad k = 1, 2, ..., N \quad (3.3.12)$$

$$\lambda \ge 0 \quad (3.3.13)$$

$$\mu_k \ge 0, \quad k = 1, 2, ..., N \quad (3.3.14)$$

Because the sampling frequencies (f_k) have C_k's as multipliers in the constraint $\sum_{k=1}^{N} C_k f_k \le 1$, the expression $\frac{m_k \alpha_k}{C_k} e^{-\alpha_k f_k}$, which is obtained by dividing the minus derivative of tracking quality function of the kth tracking task to the computation time C_k, is used instead of slopes. Let's call $\frac{m_k \alpha_k}{C_k} e^{-\alpha_k f_k}$ as M_k. M_k's are defined as *marginal return* in [12]. The condition in 3.3.8 requires the M_k values of the tracking tasks, whose sampling frequencies are increased, to be equal to each other. When the Q-RAM approach is applied; as a result of the algorithm, suppose that sampling frequency of p out of N applications ($0 \le p \le N$) are increased from the minimum sampling frequency values ($f_{k,min}$, $k = 1, 2, \dots, N$) and the following condition is satisfied:

$$\frac{m_1 \alpha_1}{C_1} e^{-\alpha_1 f_1} = \frac{m_2 \alpha_2}{C_2} e^{-\alpha_2 f_2} = \dots = \frac{m_p \alpha_p}{C_p} e^{-\alpha_p f_p} = \lambda \quad (3.3.15)$$

In this arrangement of the tracking tasks, tasks are arranged in decreasing marginal return order as it is indicated in the algorithm above, i.e. the tasks from 1 to p have the highest marginal utility value. As a result of the algorithm, suppose that the utilization of the radar processor does not reach to % 100, i.e. $\sum_{k=1}^{N} C_k f_k < 1$, but remaining utilization is not enough to make the marginal returns of the tracking tasks with the highest marginal return value, to be equal to the second largest marginal return value. Or p is equal to N and the utilization of the radar processor is not % 100. In both of the cases the KKT condition in 3.3.11 is not satisfied. In [23], to satisfy the considered condition along with the other KKT conditions, the following modification is done on the Q-RAM procedure,

From 3.3.15, the sampling frequency value of kth tracking task can be written as

$$f_k = (\frac{-1}{\alpha_k})\ln(\frac{\lambda C_k}{m_k \alpha_k}). \tag{3.3.16}$$

One can now substitute expression 3.3.16 in place of f_k in 3.3.11 in order to obtain

$$\sum_{k=1}^{p} \frac{-C_k}{\alpha_k}\ln(\frac{\lambda C_k}{m_k \alpha_k}) + \sum_{k=p+1}^{N} C_k f_{k,\min} = 1. \tag{3.3.17}$$

By solving the equation in 3.3.17, ln(λ) can be obtained as

$$\ln(\lambda) = \frac{1 - \sum_{k=p+1}^{N} C_k f_{k,\min} + \sum_{k=1}^{p} \frac{C_k}{\alpha_k}\ln(\frac{C_k}{m_k \alpha_k})}{\sum_{k=1}^{p} \frac{-C_k}{\alpha_k}}. \tag{3.3.18}$$

Again by substituting ln(λ) expression in 3.3.18 into the expression 3.3.16, the required sampling frequency can finally be obtained as

$$f_k = (\frac{1}{\alpha_k})\left[\ln(\frac{m_k\alpha_k}{C_k}) + \frac{1 - \sum_{i=p+1}^{N} C_i f_{i,\min} + \sum_{i=1}^{p} \frac{C_i}{\alpha_i}\ln(\frac{C_i}{m_k\alpha_i})}{\sum_{i=1}^{p}\frac{C_i}{\alpha_i}}\right], \quad k = 1, 2, ..., p \qquad (3.3.19)$$

$$f_k = f_{k,\min}, \quad k = p+1, \ ..., N \qquad (3.3.20)$$

If the utilization of the radar processor did not reach %100 after applying the Q-RAM procedure, sampling frequencies of the tracking tasks, whose sampling frequency values are increased in the Q-RAM procedure, can be found from the formula in 3.3.19. Here p can be found in Q-RAM procedure by iteratively making the marginal returns of the tasks equal to each other beginning from the task(s) with the highest marginal return and increasing the sampling frequency of the task(s) until the marginal utility becomes equal to the second largest marginal return value as described in the algorithmic procedure above.

Therefore, for the single resource case, one can define proposed optimal Q-RAM by the following pseudo-code:

1. Let the current sampling frequency of the kth tracking task be f_k, $1 \leq k \leq N$. Compute $(M_1, M_2, ..., M_N)$.
2. Identify
    i. the subcollection of tasks with largest value of M_k,
    ii. the number of tasks in that subcollection (denoted by p),
    iii. the task (denoted by j) with the second largest value of this quantity if any such task exists, else find the sampling frequencies of all of the tasks by using the expression 3.3.19 and terminate the algorithm.
3. If the largest value of M_k is 0, then stop.
4. Otherwise, increase f_k for each of the members of the subcollection so that their values of M_k decrease but continue to be equal until one of the following is satisfied,
    i. this value becomes equal to the second largest marginal return value, M_j, or,

ii. the utility of the radar processor reaches %100 utilization when the marginal returns, M_k k = 1, 2, …, p, of the tasks in the subcollection become equal to the second largest marginal return value, M_j,

iii. the utility of the radar processor reaches %100 utilization before the marginal returns of the tasks in the subcollection, M_k k = 1, 2, …, p, become equal to the second largest marginal return value, M_j.

5. If (ii) is satisfied, stop as all resources have been optimally allocated.
6. If (ii) is satisfied, find the sampling frequencies of all of the tasks in the subcollection (k = 1, 2, …, p) by using the expression 3.3.19 and the sampling frequencies of the other tasks (k = p+1, 2, …, N) by using the expression 3.3.20 and terminate the algorithm.
7. If (i) is satisfied, one or more new tasks should be added to the subcollection. Return to step 1.

With the modification on the Q-RAM procedure for single resource type, optimal operating points can be obtained. As it is indicated in [16], [6], [19], [8], [7] and [10], main objective of Q-RAM based approaches are to reach to a solution point, which is closest to the optimal point, in real-time systems. As it is shown in [23] and Table 3.3.2, Q-RAM approach with the modification described above reaches optimal point in below one millisecond.

Since a continuous quality function in single resource dimension can not be defined for each application for the multiple resource type case, the modified Q-RAM procedure can not be applied in multiple resource type. Therefore, near-optimal Q-RAM based approaches and methods of feasible direction are investigated in order to handle multiple resource type case. Near-optimal Q-RAM base approach is explained in following subsections of this chapter.

3.3.2 Simulations for Run Time Measurement

In order to measure the run time of the modified Q-RAM approach for single resource type case the input data shown in Table 3.3.1 is used. Detailed explanations regarding the simulation technique and selection of simulation scenarios are provided in Chapter 5 for multiple resource type case; for single resource type case also the same simulation technique is used. The parameters m_k, C_k, α_k and $f_{k,min}$ is selected in order to provide various simulation conditions which enables performing simulations that are unbiased from the specific scenario conditions.

Table 3.3.1 Input Data for Run Time Measurement of Modified Q-RAM for Single Resource Type Case.

Task	T01	T02	T03	T04	T05	T06	T07	T08	T09	T10
m_k	0.8	0.9	0.75	0.85	0.95	0.7	0.9	0.8	0.95	0.85
C_k (ms.)	5	6	7	8	7	6	5	6	7	8
α_k	0.060	0.065	0.065	0.060	0.055	0.050	0.050	0.055	0.060	0.065
$f_{k,min}$	70	50	55	50	60	65	60	75	50	70

N is the number of tracking tasks included in the simulation scenario. *N* tasks are selected from the task list of Table 3.3.1 for each simulation scenario. Multiple simulations are performed for each *N* and execution time of the optimization algorithm is measured for each of the simulations. In Table 3.3.2 average of the run time values measured in different simulation scenarios for different *N*'s are presented.

Table 3.3.2 Mean Run Time of Modified Q-RAM for Single Resource Type Case for Different Number of Tracking Tasks Included in the Simulation Scenario.

Number of Tasks (*N*)	Mean Run Time (sec.)
1	0.00006406
2	0.00010438
3	0.00011549
4	0.00011868
5	0.00012171
6	0.00012500
7	0.00012839
8	0.00013194
9	0.00013438

As it is presented in Table 3.3.2, the execution time of the optimal Q-RAM based solution approach to the radar resource allocation approach is in the order of 1e-4 second, when the number of tracking tasks included in the scenario changes from 1 to 9. In Table 3.3.3, the number of tracking tasks included in the scenario is increased and the numbers of tasks are varied from 20 to 200 with a step of 20.

Table 3.3.3 Run Time of the Modified Q-RAM Approach. Number of tracking tasks included in the scenario increases from 20 to 200 with a step of 20. For each number of tracking task level, 100 simulations are performed and averages of the 100 simulations for each different *N* are presented in this table.

Number of Tasks (*N*)	Mean Run Time (sec.)
20	0.001625
40	0.015625
60	0.025
80	0.046875
100	0.078125
120	0.12188
140	0.17188
160	0.225
180	0.24531
200	0.25469

The optimal solution approach provides global optimum results with a convergence time below one second even if the number of tasks included in the scenario is increased to 200 which can be considered as a dense scenario environment. In real-time systems, it is important to improve the performance by re-allocating the resources adapting to dynamic situations [16]. Therefore, changing task parameters for resource re-allocation with negligible overhead is important for these systems. Hence, based on the results presented above, it can be concluded that the solution approach can be considered for the real-time applications.

3.4 Approach for the Case with Multiple Resource Type

The solution obtained in the previous section can be applied to the resource allocation problems including single resource type and whose objective functions and constraints are suitable with the assumptions in subsection 3.2.2. And as it can be observed from the simulation results and [23], for the specific radar resource allocation case, real-time performance of the solution approach of section 3.3 is favorable. This solution approach is attempted to be extended to the multiple resource type case but an appropriate result can not be obtained. There exist near-optimal Q-RAM based resource allocation approaches in literature [16], [18], [19] and [10] for the multiple resource type case. In this section these approaches are investigated in detail.

In [16], Lee et al. extended the Q-RAM approach for single resource type and proposed a near-optimal for the radar resource allocation algorithms in which sampling frequency (f_k) and computation time (C_k) are considered as resources. In the next subsection, this approach is investigated.

3.4.1 Extension to the Specific Two Resource Type Case

Lee et al. [16] assumed that each tracking task, k (k = 1, 2, …, N), have a discrete tracking quality function, $Q_k(f_k)$, depending on sampling frequency and defined for different tracking filter algorithms (algorithm1, algorithm 2, etc.) as shown in Figure 3.4.1 and considered the radar resource allocation problem formulated to below:

Maximize

$$\sum_{k=1}^{N} Q_k$$

Subject to

$$\sum_{k=1}^{N} C_k f_k \leq 1$$

$$f_k \geq f_{k,min}, \qquad k = 1, 2, \ldots, N.$$

As it is explained in the previous subsection, Q-RAM approach for the single resource type case processes single utility function depending on single variable for each application. In order to handle two variable case (sampling frequency and computation time depending on tracking filter algorithm complexity), Lee et al. merged the tracking quality functions defined for different tracking filter algorithms as shown Figure 3.4.2 which shows the two algorithm case.

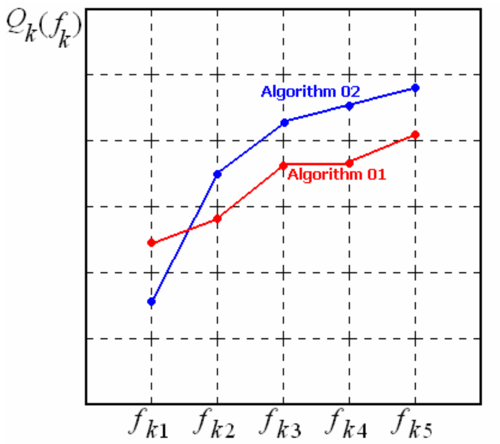

Figure 3.4.1 Discrete Tracking Quality Functions Depending on Sampling Frequency (f_k) and Defined for Different Tracking Algorithms. Lee et al. [16] assumed computation time (C_k) of the first algorithm is 2 ms and that of the second algorithm is 3 ms for illustrative purposes.

In order to obtain a single tracking quality function, discrete tracking quality functions are obtained for each tracking task, depending on utility of the radar processor U_k, by using the relation $U_k = f_k \times C_k$. After this variable transformation, the tracking quality function depending on the utilization of the radar processor, $Q_k(U_k)$, can be obtained by taking the maximum of the two functions (the thick line in Figure 3.4.2). And after this point Lee et al. investigated U_k as resource. Associated with each U_k there exists (f_k, C_k) pairs.

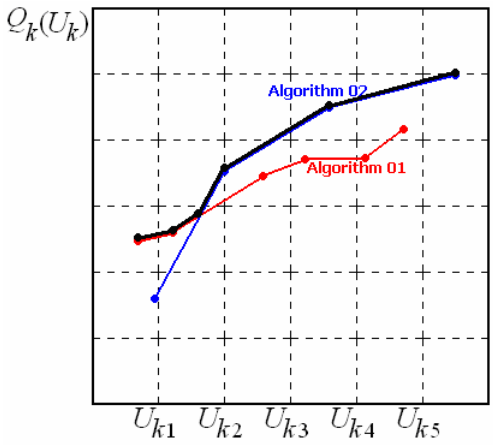

Figure 3.4.2 Merged Tracking Quality Function. The tracking quality functions in the previous figure are re-plotted depending on the utilization of the radar processor which is $U_k = f_k \times C_k$. The maximum of the figures on the same utilization points are taken and one tracking quality function is obtained for each tracking task.

Lee et al. [16] constructed convex hulls of the merged tracking quality function of each tracking task as shown in Figure 3.4.3 and obtained the final tracking quality functions, $Q_k(U_k)$, $k = 1, 2, \ldots, N$.

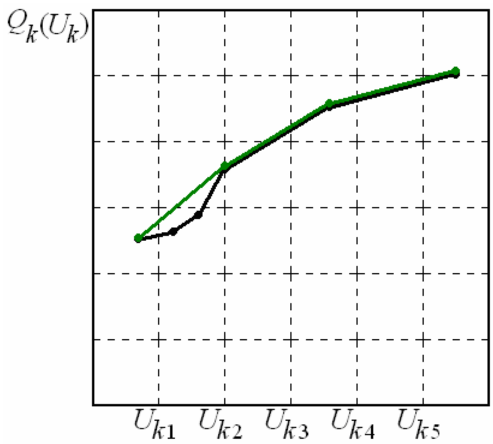

Figure 3.4.3 Convex Hull of the Merged Tracking Quality Function. The green line in this figure shows the convex hull of the merged tracking quality function.

On the final tracking quality functions, Lee et al. applied the following algorithmic approach, which is based on Q-RAM procedure, in order to obtain a resource allocation resulting in a total tracking quality which is closest to the optimal. In this approach, $Q_k'(U_k)$ is the unit tracking quality difference between the current operating point and the next operating point.

1. Calculate $(Q_1'(U_1), Q_2'(U_2), ..., Q_N'(U_N))$ and sort the tracking quality tasks in decreasing $Q_k'(U_k)$ order at current operating points.
2. Evaluate the subcollection of tasks having the highest $Q_k'(U_k)$ value, let's call this set *H*.
3. Test whether the utilization of the radar processor exceed %100 or not if the next operating point following the current operating point on the convex hull curves is selected for each task in the set *H*.
    a. If full utilization is not exceeded, pass to the next operating point following the current operating point on the convex hull curves for each task in the set *H*
    b. If full utilization is exceeded, terminate the procedure.
4. Return to step 1.

In this approach, tracking quality functions of the tracking tasks are taken as discrete functions, Q_k; in these functions tracking quality changes with sampling frequency and computation time pairs. In the algorithmic approach above, *operating point* represents the sampling frequency and computation time pair, (f_k, C_k), for each tracking task as the goal of the algorithm is to find the best operating point, i.e. sampling frequency (f_k) and computation time (C_k) value for each task. In Q-RAM procedure in subsection 3.3 searches are conducted in one dimension for each application because there is single resource type. When resources more that one type are considered, in order to conduct the Q-RAM approach the search dimension is demoted to one dimension by considering the sampling frequency and computation time in the one axis in the approach of [16] which is described above.

When compared with the Q-RAM procedure described in section 3.3, the logic behind the approach of Lee et al. [16] is same with the Q-RAM procedure. In [16] the considered objective functions are discrete functions and some approximations are made in order to obtain convex hull of the discrete functions. The near-optimal algorithm of [16] considers only the computational resources of the radar system. In order to consider the energy resource of the radar system along with the computational resources, Q-RAM based approaches presented in [18], [19], [10] and [24], which considers the general case of multiple resource case, can be examined. In the next subsection general extension of the Q-RAM to the multiple resource type case is investigated.

3.4.2 Extension to the General Multiple Resource Type Case

In [10] Lee et al. proposed resource vectors, which include the different types of resources as scalar components, to handle the multiple resource type case and enable the search in one dimension for each application as the Q-RAM procedure makes search in one dimension for each application. The objective functions of the applications are assumed to be discrete in [10]. In [18], [19] and [24], the approach of [10] is used and Q-RAM based resource allocation approaches are presented for specific problems. In [19] and [24], radar resource allocation problem is considered; in [18], network applications and resource management in phased array radar systems are investigated. In the following paragraphs, Q-RAM based resource allocation procedure of [18], [19], [10] and [24] for general multiple resource type case is explained on the radar resource allocation problem.

The radar resource allocation problem formulated in subsection 2.3 is restated below. But in this case the tracking quality functions are assumed to be discrete different than the previous formulation; Q_k is the tracking quality function of the kth tracking task ($k = 1, 2, \ldots, N$). And, the tracking quality functions are dependent on sampling frequency (f_k), average power of the transmitted radar signal (P_k) and also computation time (C_k). Computation time is also considered as an optimization variable besides sampling frequency and power. In [16] it is assumed that tracking a target with more sophisticated algorithms will require more processor resource but produce better tracking quality. Therefore, for the radar resource allocation problem formulated below, it is assumed that tracking quality increases with increase of the computation time (C_k).

Maximize

$$\sum_{k=1}^{N} Q_k(f_k, C_k, P_k)$$

Subject to

$$\sum_{k=1}^{N} C_k f_k \leq 1$$

$$\sum_{k=1}^{N} P_k \leq P_{\max}$$

$$f_k \geq f_{k,min}, \qquad k = 1, 2, \ldots, N.$$

$$P_k \geq P_{k,min}, \qquad k = 1, 2, \ldots, N.$$

The discrete tracking quality functions are shown in Figure 3.4.4.

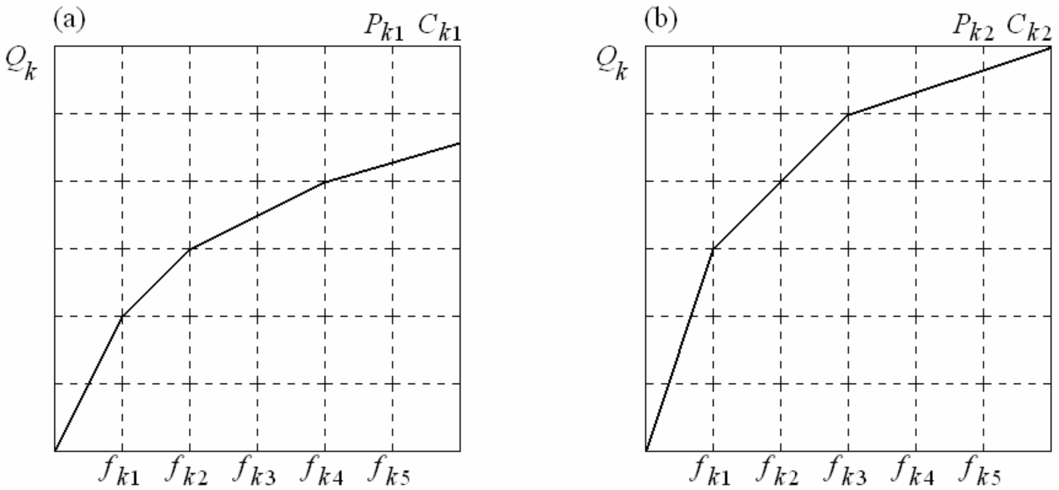

Figure 3.4.4 Discrete Tracking Quality Functions for *k*th Tracking Task (Q_k) for the Case of two (P_k, C_k) Options. The first curve is valid for the average power of the transmitted radar signal level (P_k) of P_{k1} and computation time C_{k1} and the second curve is valid for the average power of the transmitted radar signal level (P_k) of P_{k2} and computation time C_{k1}. These two curves define the discrete tracking quality function (Q_k). If there exist some other P_k and C_k options, the number of curves can be increase in order to define the Q_k function completely.

The Figure 3.4.4(a) is valid for the average power of the transmitted radar signal level (P_k) of P_{k1} and computation time C_{k1} and the Figure 3.4.4(b) is valid for the average power of the transmitted radar signal level (P_k) of P_{k2} and computation time C_{k1}. These two curves define the discrete tracking quality function (Q_k). If there exist some other P_k and C_k options, the number of curves can be increase in order to define the Q_k function completely.

In order to apply Q-RAM approach in multiple resource type case, it is intended to obtain a tracking quality function in one dimension in [18], [19], [10] and [24]. In order to obtain a single tracking quality function for each tracking task sampling frequency (f_k), computation time (C_k) and average transmitted power (P_k) parameters are merged in a resource vector, $\mathbf{O}_k = [f_k \;\; C_k \;\; P_k]^T$ (k = 1, 2, …, N), where each resource vector represents an operating point as shown in Figure 3.4.6 and for each vector there exists a corresponding quality value in the tracking quality curves. Hence, a resource vector-tracking quality function can be obtained for each tracking task by arranging the quality vectors in increasing quality value order. For the two curves in Figure 3.4.4(a) and Figure 3.4.4(b), the resultant resource vector-tracking quality function is shown in Figure 3.4.5.

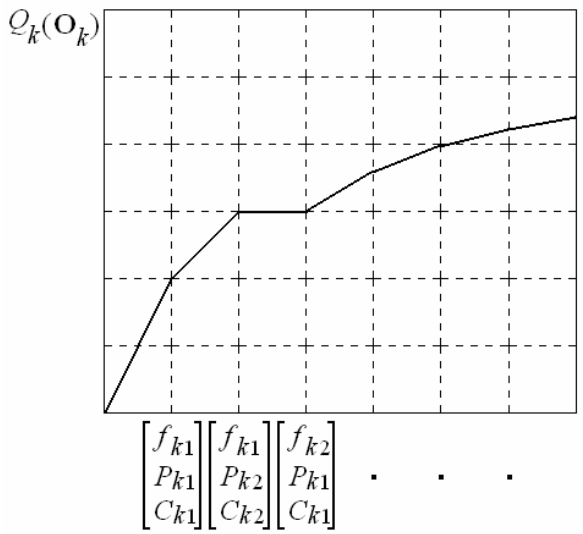

Figure 3.4.5 Tracking Quality Function Depending on Resource Vector ($\mathbf{O}_k$). The discrete operating points in Figure 3.4.4(a) and Figure 3.4.4(b) are arranged in the increasing tracking quality order. As the discrete operating points define the resource vectors, the tracking quality function depending on resource vector is obtained.

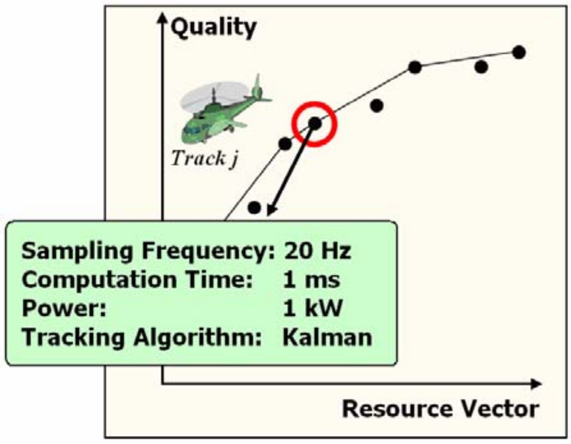

Figure 3.4.6 Discrete Tracking Quality Function. As shown in the figure each discrete point on the function defines an operating point for a tracking task. In this figure for the *j*th tracking task numerical examples for an operating point is given. In this example tracking algorithm for the specified operating point is shown as *Kalman* whose computation time is assumed to be 1 ms for this specific track. Also, the convex hull of the discrete quality function is shown in this figure.

By this way, a single resource-tracking quality function is obtained for each task. These functions are discrete and may not be concave. At this point, similar to the approach of [16], convex hull of each tracking quality function is obtained, by using the approach shown in Figure 3.4.3. Based on the optimization procedure of the Q-RAM approach, the following iterative approach can be applied on the resultant resource vector-tracking quality functions, Q_k, (k = 1, 2, …, N). In this algorithmic approach, Q_k' denotes the unit tracking quality difference between the next operating point and the current operating point divided; i.e. $Q_k' = (Q_k(\mathbf{O}_{k,c+r}) - Q_k(\mathbf{O}_{k,c}))/r$, $\mathbf{O}_{k,c}$ is the current operating point and $\mathbf{O}_{k,c+r}$ is the next operating point on the convex hull of the tracking quality function of the kth task.

1. Let the current allocated resource to kth tracking task be $\mathbf{O}_k$, $1 \leq k \leq N$. Compute and sort $(Q_1'(\mathbf{O}_1), ..., Q_N'(\mathbf{O}_N))$.
2. Identify the first task in the sorted list of step 1. Pass to the next resource vector (discrete operating point) on the convex hull of the resource vector-tracking quality curve obtained prior to the step 1 of the identified task. If resources are not sufficient for the specified allocation, then stop. Else, make the allocation.

3. After allocating resources to the specified task; if there are unallocated resources, return to step 1. If there are no unallocated resources, then stop.

As it can be observed from the Q-RAM based algorithm for the case of multiple resource type, the approach is similar to the case with single resource type case except the objective functions in multiple resource type case are discrete and some approximations are made in order to obtain a resource-quality function in single dimension. In the next subsection the approximations and drawbacks of the Q-RAM based resource allocation approach for the multiple resource type case are explained.

3.5 Drawbacks of the Approaches for Multiple Resource Type Case

As it is mentioned previously, the goal of the Q-RAM based approaches for the multiple resource type case is to reach a solution, which is closest to the optimal solution, in real-time systems. In this algorithmic approach, the convex hulls of the objective functions of the tasks are fed as input to the algorithm. The reason for consideration of the convex hull functions for each task is to obtain the highest quality increase per resource increase for each task at each iteration. In the iterations of the algorithm, the operating point following the current operating point on the tracking quality curve and giving the highest quality increase per resource increase is selected as the next operating point for each task. This operating point selection procedure results in convex hull functions for each task.

In the approach described in previous subsection, in order to obtain the convex hull of the objective functions of the tasks some operating points are not taken into consideration in the search of the solution point process as shown in Figure 3.5.1.

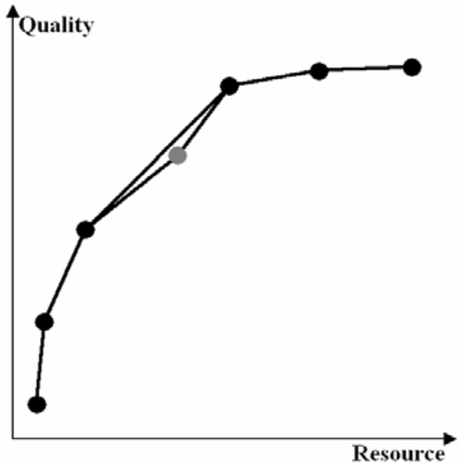

Figure 3.5.1 Obtaining Convex Hull of a Discrete Function. The discrete point having gray color is discarded in order to obtain the convex hull function.

Neglecting some of the operating points, as shown in Figure 3.5.1, in the search process may lead to non-optimal results as the optimal result may contain the neglected operating points for some of the tasks.

In the case of single resource type, it is assumed that the objective functions of the applications are twice differentiable and constraints are convex in Q-RAM approach and an algorithm is presented in order to obtain results satisfying the KKT optimality conditions. Along with the assumptions obtaining a solution satisfying KKT conditions enables to reach optimal results. But in the case of multiple resource type, the Q-RAM resource allocation approach does not have a convincing theoretical background in the view of optimality. The approach for the multiple resource type is similar to the approach the single resource type case as in both of the procedures; the resources of the tasks having the highest marginal returns are increased at each iteration. But, the Q-RAM procedure for the multiple resource type case does not provide necessary and sufficient conditions for the optimality.

In order to obtain a theoretically convincing and optimal resource allocation approach for the radar resource allocation problem, the Methods of Feasible Directions, which propose optimization algorithms for the constrained optimization problems with non-linear objective functions, are considered. The considered algorithms in the literature of the Methods of Feasible Directions are Zoutendijk Algorithm with Topkis-Veinott's Modification, Gradient Projection Algorithm and Convex-Simplex Algorithm. The outputs of these algorithms satisfy the KKT optimality conditions in the case of convergence of the algorithms. When the objective function and constraints are convex, the outputs of the considered algorithms are optimal as KKT conditions are sufficient for optimality. As the constraint in Eq. 2.3.2 is not convex when both f_k and C_k are considered as optimization variables, computation time (C_k) is not considered as an optimization variable and the radar resource allocation problem defined in subsection 2.3 is considered when the Methods of Feasible Directions are used for resource allocation. The Q-RAM based approach for multiple resource type case is also implemented for resource allocation problem of subsection 2.3 and Q-RAM and Methods of Feasible direction are compared in terms of optimality and execution time in Chapter 5.

CHAPTER 4

RESOURCE ALLOCATION WITH METHODS OF FEASIBLE DIRECTIONS

As it is mentioned in the previous chapter, Q-RAM approach generates near-optimal results to the resource allocation problem and the theoretical background of the model does not consider the sufficiency conditions for optimality as a whole. In this thesis, it is proposed to obtain a solution, which provides a well founded mathematical background and generates optimal results as fast as the Q-RAM approach, to the resource allocation problem with multiple resource type. In order to achieve this, the Methods of Feasible Directions, which have been applied to the resource allocation problem in network applications [21] and [26], is first proposed to be applied to the radar resource allocation problem. The radar resource allocation problem formulated in subsection 2.3 is a constrained optimization problem with linear constraints. In this section, the algorithms considered in the Methods of Feasible Directions literature and that can generate optimal results to the optimization problems with convex objective function and linear constraints, are investigated. The performances of the algorithms are compared with Q-RAM on the radar resource allocation problem of subsection 2.3.

4.1 Zoutendijk Algorithm

In this section, the theory of the Method of Zoutendijk and the modification of Topkis and Veinott [1967] on the Method of Zoutendijk is introduced.

4.1.1 Theory

In the Method of Zoutendijk, an improving feasible direction is generated and a search is conducted on the generated direction at each iteration [4]. The definition of improving feasible direction is:

'Consider the problem to minimize $f(\mathbf{x})$ subject to $\mathbf{x} \in s$, where $f{:}E_n \rightarrow E_1$ and S is a nonempty set in E_n. A nonzero vector d is called a *feasible direction* at $\mathbf{x} \in s$ if there exists a

$\delta > 0$ such that $\mathbf{x} + \lambda\mathbf{d} \in s$ for all $\lambda \in (0, \delta)$. Furthermore, **d** is called an *improving feasible direction* at $\mathbf{x} \in s$ if there exists $\delta > 0$ such that $f(\mathbf{x} + \lambda\mathbf{d}) < f(\mathbf{x})$ and $\mathbf{x} + \lambda\mathbf{d} \in s$ for all $\lambda \in (0, \delta)$' [4].

In [4], the optimization problem with linear constraints provided below is considered in order to describe the theory of the algorithm.

$$\begin{aligned} &\text{Minimize} \quad && f(\mathbf{x}) \\ &\text{subject to} \quad && \mathbf{Ax} \le \mathbf{b} \\ & && \mathbf{Ex} = \mathbf{e} \end{aligned}$$

And the following lemma is presented:

Lemma 4.1.1.1

'Consider the problem to minimize $f(\mathbf{x})$ subject to $\mathbf{Ax} \le \mathbf{b}$ and $\mathbf{Ex} = \mathbf{e}$. Let **x** be a feasible solution, and suppose that $\mathbf{A}_1\mathbf{x} = \mathbf{b}_1$ and $\mathbf{A}_2\mathbf{x} < \mathbf{b}_2$, where $\mathbf{A}^t$ is decomposed into $(\mathbf{A}_1^t, \mathbf{A}_2^t)$ and $\mathbf{b}^t$ is decomposed into $(\mathbf{b}_1^t, \mathbf{b}_2^t)$. Then, a nonzero vector **d** is a feasible direction at **x** if and only if $\mathbf{A}_1\mathbf{d} \le \mathbf{0}$ and $\mathbf{Ed} = \mathbf{0}$. If $\nabla f(\mathbf{x})^t\mathbf{d} < 0$, then d is an improving direction' [4].

Based on the lemma above, generating an improving feasible direction is explained as below;

'Given a feasible point **x**, as shown in Lemma 4.1.1.1, a nonzero vector d is an improving feasible direction if $\nabla f(\mathbf{x})^t\mathbf{d} < 0$, $\mathbf{A}_1\mathbf{d} \le \mathbf{0}$ and $\mathbf{Ed} = \mathbf{0}$. A natural method for generating such a direction is to minimize $\nabla f(\mathbf{x})^t\mathbf{d} < 0$ subject to the constraints $\mathbf{A}_1\mathbf{d} \le \mathbf{0}$ and $\mathbf{Ed} = \mathbf{0}$. Note, however, if a **d** such that $\nabla f(\mathbf{x})^t\mathbf{d} < 0$, $\mathbf{A}_1\mathbf{d} \le \mathbf{0}$ and $\mathbf{Ed} = \mathbf{0}$ exists, then the optimal objective value of the foregoing problem is $-\infty$ by considering $\lambda\mathbf{d}$, where λ is arbitrarily large. Thus, a constraint that bounds the vector **d** or the objective function must be presented' [4].

In [4], three problems, each problem using a different normalization constraint explained above, for generating an improving feasible direction is presented. In this thesis the following problem is used in order to generate an improving feasible direction:

Problem D:

$$\begin{aligned} &\text{Minimize} \quad && \nabla f(\mathbf{x})^t\mathbf{d} \\ &\text{subject to} \quad && \mathbf{A}_1\mathbf{x} \le \mathbf{b} \\ & && \mathbf{Ex} = \mathbf{e} \\ & && -1 \le d_j \le 1 \qquad \text{for } j = 1, \ldots, n \end{aligned}$$

This problem can be solved by the simplex method. Since $\mathbf{d} = \mathbf{0}$ is a feasible to the above problem, and since its objective value is zero, the optimal objective value of the above problem can not be positive. If the minimum objective function value of the problem above is negative, then, by the Lemma 4.1.1.1, an improving feasible direction is generated. It is proved in [4] that if the minimal objective function value of the above problem is zero, then $\mathbf{x}$ is a KKT point.

It is proposed to solve the following line search problem after determination of the improving feasible direction, which is stated below, in Zoutendijk Algorithm [4].

Problem M:

$$\begin{aligned} &\text{Minimize} && \nabla f(\mathbf{x}_k + \lambda \mathbf{d}_k) \\ &\text{subject to} && 0 \le \lambda \le \lambda_{max} \end{aligned}$$

where $\mathbf{x}_k$ is the current vector, $\mathbf{d}_k$ is the improving feasible direction and $(0, \lambda_{max})$ is the interval of uncertainty. Determination of λ_{max} is explained in detail in [4], refer to [4] for determination of λ_{max}.

4.1.2 Algorithmic Approach

The Method of Zoutendijk for minimizing a differentiable function f in the presence of linear constraints of the form $\mathbf{Ax} \le \mathbf{b}$ and $\mathbf{Ex} = \mathbf{e}$ is provided below [4].

Initialization: Find a feasible solution $\mathbf{x}_1$. Let $k = 1$ and go to the step 1.

1. $\mathbf{A}^t$ and $\mathbf{b}^t$ are decomposed into $(\mathbf{A}_1{}^t, \mathbf{A}_2{}^t)$ and $(\mathbf{b}_1{}^t, \mathbf{b}_2{}^t)$ such that $\mathbf{A}_1\mathbf{x}_k = \mathbf{b}_1$ and $\mathbf{A}_2\mathbf{x}_k < \mathbf{b}_2$. Let $\mathbf{d}_k$ be an optimal solution to the problem D. If $\nabla f(\mathbf{x}_k)^t\mathbf{d} = 0$, stop; $\mathbf{x}_k$ is a KKT point. Else, go to step 2.
2. Let λ_k is an optimal solution to the problem M. Let $\mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k\mathbf{d}_k$. Replace k by $k+1$ and repeat step 1.

In [4], it is shown that the algorithmic map of Zoutendijk's method is not closed and convergence is not generally guaranteed. A modification of Zoutendijk's method is proposed by Topkis and Veinott which guarantees a solution to a KKT point [4]. The considered problem is:

$$\text{Minimize} \quad f(\mathbf{x})$$

subject to $g_i(\mathbf{x}) \le 0$ for $i = 1, \ldots, m$

in Topkis-Veinott's Modification Algorithm. In order to generate a feasible direction the following direction finding problem is considered instead of problem ***P***.

Problem DF:

$$\begin{aligned} &\text{Minimize} && z \\ &\text{subject to} && \nabla f(\mathbf{x})^t\mathbf{d} - z \le 0 \\ & && \nabla g_i(\mathbf{x})^t\mathbf{d} - z \le - g_i(\mathbf{x}) && \text{for } i = 1, \ldots, m \\ & && -1 \le d_j \le 1 && \text{for } j = 1, \ldots, n \end{aligned}$$

Here, both binding and nonbinding constraints play a role in determining the feasible direction [4]. The line search problem is same as problem M in Zoutendijk's method except the determination of uncertainty interval is different. For determination of the uncertainty interval refer to [4].
The algorithmic procedure of Topkis-Veinott's Modification Algorithm is as follows.

Initialization: Find a feasible solution $\mathbf{x}_1$. Let $k = 1$ and go to the step 1.

1. Let $(z_k, \mathbf{d}_k)$ be an optimal solution to the problem DF. If $z_k = 0$, stop; $\mathbf{x}_k$ is a KKT point. Else, $z_k < 0$ and go to step 2.
2. Let λ_k is an optimal solution to the problem M. Let $\mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k\mathbf{d}_k$. Replace k by k+1 and repeat step 1.

The convergence of Topkis-Veinott's Algorithm is proved in [4]. Refer to [4] for proof and further details of the algorithm.

As it can be observed from the simulation results of the algorithm presented in Chapter 5, the execution time of the algorithm is higher than the other possible alternatives when the algorithm is considered in real-time applications. Alternative algorithms are also investigated in the Methods of Feasible Directions literature in order obtain favorable results in terms of real-time performance. In the next subsection, the theoretical background of the Gradient Projection Algorithm, which is one of the alternatives, is briefly presented.

4.2 Gradient Projection Algorithm

When minimizing a function without constraints, the direction of steepest descent is that of the negative gradient. However, in the case of constrained minimization problems, moving along the steepest descent direction may lead violation of the constraints. In the Gradient Projection Method of Rosen the aim is to project the negative gradient in such a way that the direction is feasible and the objective function is improved [4]. The direction of steepest descent is multiplied by a projection matrix **P** in this method. The definition of projection matrix is 'An $n \times n$ matrix **P** is called a *projection matrix* if $\mathbf{P} = \mathbf{P}^t$ and $\mathbf{PP} = \mathbf{P}$' [4].

4.2.1 Theory

In this section the theoretical background of the Gradient Projection Algorithm is presented on the following optimization problem,

$$\begin{array}{ll} \text{Minimize} & f(\mathbf{x}) \\ \text{subject to} & \mathbf{Ax} \leq \mathbf{b} \\ & \mathbf{Ex} = \mathbf{e} \end{array}$$

where **A** is an $m \times n$ matrix, **E** is an $l \times n$ matrix, **b** is an m vector, **e** is an l vector, and f: $E_n \rightarrow E_l$ is a differentiable function. Assume that **x** is a feasible point. Moving along $-\nabla f(\mathbf{x})$ (the direction of steepest descent) may destroy feasibility. In Gradient Projection Method, in order to maintain feasibility, $-\nabla f(\mathbf{x})$ is multiplied with a suitable projection matrix, **P**, and a feasible direction, $\mathbf{d} = -\mathbf{P}\nabla f(\mathbf{x})$, is obtained [4]. The following lemma, which provides the form of a suitable projection matrix **P**, is presented in [4]

Lemma 4.2.1.1

'Consider the problem to minimize $f(\mathbf{x})$ subject to $\mathbf{Ax} \leq \mathbf{b}$ and $\mathbf{Ex} = \mathbf{e}$. Let **x** be a feasible point such that $\mathbf{A}_1\mathbf{x} = \mathbf{b}_1$ and $\mathbf{A}_2\mathbf{x} < \mathbf{b}_2$, where $\mathbf{A}^t = (\mathbf{A}_1^t, \mathbf{A}_2^t)$ and $\mathbf{b}^t = (\mathbf{b}_1^t, \mathbf{b}_2^t)$. Furthermore, suppose that f is differentiable at **x**. If **P** is a projection matrix such that $\mathbf{P}\nabla f(\mathbf{x}) \neq \mathbf{0}$, and then $\mathbf{d} = -\mathbf{P}\nabla f(\mathbf{x})$ is an improving direction of f at **x**. Furthermore, if $\mathbf{M}^t = (\mathbf{A}_1^t, \mathbf{E}^t)$ has full rank, and if **P** is of the form $\mathbf{P} = \mathbf{I} - \mathbf{M}^t(\mathbf{MM}^t)^{-1}\mathbf{M}$, then **d** is an improving feasible direction' [4].

For proof of the lemma refer to [4].

As it is shown in the Lemma 4.2.1.1 if $\mathbf{P}\nabla f(\mathbf{x}) \neq 0$, then $\mathbf{d} = -\mathbf{P}\nabla f(\mathbf{x})$ is an improving feasible direction. Suppose that $\mathbf{P}\nabla f(\mathbf{x}) = 0$. Then,

$$\mathbf{0} = \mathbf{P}\nabla f(\mathbf{x}) = [\mathbf{I} - \mathbf{M}^t(\mathbf{M}\mathbf{M}^t)^{-1}\mathbf{M}]\nabla f(\mathbf{x}) = \nabla f(\mathbf{x}) + \mathbf{M}^t\mathbf{w} = \nabla f(\mathbf{x}) + \mathbf{A}_1{}^t\mathbf{u} + \mathbf{E}^t\mathbf{v}$$

where $\mathbf{w} = -(\mathbf{M}\mathbf{M}^t)^{-1}\mathbf{M}\nabla f(\mathbf{x})$ and $\mathbf{w}^t = (\mathbf{u}^t, \mathbf{v}^t)$. If $\mathbf{u} \geq \mathbf{0}$, then the point $\mathbf{x}$ satisfies the KKT conditions [4]. If $\mathbf{u} \ngeq \mathbf{0}$, the following projection matrix, $\hat{\mathbf{P}}$, provides an improving feasible direction [4].

$$\hat{\mathbf{P}} = \mathbf{I} - \hat{\mathbf{M}}^t(\hat{\mathbf{M}}\hat{\mathbf{M}}^t)\hat{\mathbf{M}} \qquad (4.2.1)$$

0

where, if $\mathbf{u} \ngeq \mathbf{0}$, let u_j be a negative component of $\mathbf{u}$, $\hat{\mathbf{M}}^t = (\hat{\mathbf{A}}_1^t, \mathbf{E}^t)$, $\hat{\mathbf{A}}_1$ is obtained from $\mathbf{A}_1$ by deleting the row of $\mathbf{A}_1$ corresponding to u_j [4].

4.2.2 Algorithmic Approach

The algorithmic approach of Gradient Projection Method is presented below.

Initialization: A point $\mathbf{x}_1$ with $\mathbf{A}\mathbf{x} \leq \mathbf{b}$ and $\mathbf{E}\mathbf{x} = \mathbf{e}$ is selected. $\mathbf{A}^t$ and $\mathbf{b}^t$ are decomposed into $(\mathbf{A}_1{}^t, \mathbf{A}_2{}^t)$ and $(\mathbf{b}_1{}^t, \mathbf{b}_2{}^t)$ such that $\mathbf{A}_1\mathbf{x} = \mathbf{b}_1$ and $\mathbf{A}_2\mathbf{x} < \mathbf{b}_2$. Let $k = 1$ and go to step 1.

1. Evaluate $\mathbf{M}^t = (\mathbf{A}_1{}^t, \mathbf{E}^t)$. If $\mathbf{M}$ is vacuous, stop if $\nabla f(\mathbf{x}_k) = \mathbf{0}$, let $\mathbf{d}_k = -\nabla f(\mathbf{x}_k)$, and proceed to step 2. Else, let $\mathbf{P} = \mathbf{I} - \mathbf{M}^t(\mathbf{M}\mathbf{M}^t)^{-1}\mathbf{M}$ and set $\mathbf{d}_k = -\mathbf{P}\nabla f(\mathbf{x}_k)$. If $\mathbf{d}_k \neq 0$, go to step 2. If $\mathbf{d}_k = 0$, compute $\mathbf{w} = -(\mathbf{M}\mathbf{M}^t)^{-1}\mathbf{M}\nabla f(\mathbf{x})$ and let $\mathbf{w}^t = (\mathbf{u}^t, \mathbf{v}^t)$. If $\mathbf{u} \geq \mathbf{0}$, stop; $\mathbf{x}_k$ is a KKT point. If $\mathbf{u} \ngeq \mathbf{0}$, choose a negative component of $\mathbf{u}$, say, u_j. Update $\mathbf{A}_1$ by deleting the row corresponding to u_j and repeat step 1.

2. Find the optimal solution, λ_k, to the following line search problem:

   Minimize $\quad \nabla f(\mathbf{x}_k + \lambda\mathbf{d}_k)$

   subject to $\quad 0 \leq \lambda \leq \lambda_{max}$

where λ_{max} is obtained same as it is evaluated in the Method of Zoutendijk. Let $\mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k \mathbf{d}_k$. Replace k by k+1 and repeat step 1.

It is shown that the direction finding map of Gradient Projection Algorithm is not closed, which causes the algorithm not to convergence, and a direction finding routine for a convergent variant of the Gradient Projection Method, which is provided below, is presented in [4].

Step 1 of the algorithmic procedure presented above is proposed to be modified as follows in order to obtain direction finding routine for a convergent algorithm in [4],

1. 'Let $\mathbf{M}^t = (\mathbf{A}_1{}^t, \mathbf{E}^t)$. If M is vacuous, then stop if $\nabla f(\mathbf{x}_k) = \mathbf{0}$, let $\mathbf{d}_k = -\nabla f(\mathbf{x}_k)$, and proceed to step 2. Otherwise, let $\mathbf{P} = \mathbf{I} - \mathbf{M}^t(\mathbf{M}\mathbf{M}^t)^{-1}\mathbf{M}$ and set $\mathbf{d}_k^{\mathrm{I}} = -\mathbf{P}\nabla f(\mathbf{x}_k)$. Also, compute $\mathbf{w} = -(\mathbf{M}\mathbf{M}^t)^{-1}\mathbf{M}\nabla f(\mathbf{x})$ and let $\mathbf{w}^t = (\mathbf{u}^t, \mathbf{v}^t)$. If $\mathbf{u} \geq \mathbf{0}$, then stop if $\mathbf{d}_k^{\mathrm{I}} = \mathbf{0}$; otherwise, put $\mathbf{d}_k = \mathbf{d}_k^{\mathrm{I}} \neq 0$ and proceed to step 2. On the other hand, if $\mathbf{u} \not\geq \mathbf{0}$, let $u_h = \text{minimum}_j \{u_j\} < 0$, let $\hat{\mathbf{M}}^t = (\hat{\mathbf{A}}_1^t, \mathbf{E}^t)$, where $\hat{\mathbf{A}}_1$ is obtained from $\mathbf{A}_1$ by deleting the row of $\mathbf{A}_1$ corresponding to u_h, construct the projection matrix $\hat{\mathbf{P}} = \mathbf{I} - \hat{\mathbf{M}}^t(\hat{\mathbf{M}}\hat{\mathbf{M}}^t)\hat{\mathbf{M}}$, and define $\mathbf{d}_k^{\mathrm{II}} = -\mathbf{P}\nabla f(\mathbf{x}_k)$. Now, based on some scalar constant $c > 0$, let

$$\mathbf{d}_k = \begin{cases} \mathbf{d}_k^{\mathrm{I}} & \text{if } \left\|\mathbf{d}_k^{\mathrm{I}}\right\| > |u_h| c \\ \mathbf{d}_k^{\mathrm{II}} & \text{otherwise} \end{cases} \tag{4.2.2}$$

and proceed to step 2' [4].

With the modification above the Gradient Projection Algorithm either terminates with a KKT solution, or else, generates an improving feasible direction [4]. The proof that the algorithm with the above modification is convergent is provided in [4], for further details of the algorithm refer to [4].

The Gradient Projection Algorithm is applied to the radar resource allocation problem of subsection 2.3 as in the case of the Zoutendijk Algorithm with Topkis-Veinott's Modification that is described in previous subsection. Execution time performance of the Gradient Projection Algorithm is better than that of the Zoutendijk Algorithm with Topkis-

Veinott's Modification as it can be observed from the simulation results presented in Chapter 5. In the next subsection, the Convex-Simplex Algorithm, which is also considered in the Methods of Feasible Direction literature, is briefly explained.

4.3 Convex-Simplex Algorithm

The Convex-Simplex Method is proposed to minimize a convex objective function subject to linear constraints. The method is proposed by Zangwill [1].

4.3.1 Theory

The following optimization problem is considered in Convex-Simplex Algorithm,

$$\begin{aligned} \text{Minimize} \quad & f(\mathbf{x}) \\ \text{subject to} \quad & \mathbf{Ax} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0}. \end{aligned}$$

In this algorithm, the basic variables are modified while maintaining feasibility, therefore the method is similar to the Simplex Method for problems with linear objective function and constraints [4]. For the theoretical background of the algorithm refer to [1] and [4]. In the next subsection, the algorithmic approach of the method is presented.

4.3.2 Algorithmic Approach

Based on [4], the algorithmic procedure of the convex-simplex algorithm can be introduced as follows:

Initialization: Begin with a point $\mathbf{x}_1$ satisfying the constraints $\mathbf{Ax}_1 = \mathbf{b}$ and $\mathbf{x}_1 \geq 0$. Let $k = 1$ and go to the step 1.

1. Compute I_k, $\mathbf{B}$, $\mathbf{N}$ and $\mathbf{r}$ as follows:

   where $\mathbf{A}$ is decomposed as [$\mathbf{B}$, $\mathbf{N}$] ($\mathbf{B}$ and $\mathbf{N}$ are given in the expression 4.3.2) and $\mathbf{d}^t$ is decomposed as $[\mathbf{d}_B^t, \mathbf{d}_N^t]$ such that $\mathbf{Ad} = \mathbf{Bd}_B + \mathbf{Nd}_N$.

$$I_k = \text{index set of the } m \text{ largest components of } \mathbf{x}_k \tag{4.3.1}$$

$$\mathbf{B} = \{\mathbf{a}_j : j \in I_k\} \quad \mathbf{N} = \{\mathbf{a}_j : j \notin I_k\} \tag{4.3.2}$$

$$\mathbf{r}^t = \nabla f(\mathbf{x}_k)^t - \nabla_B f(\mathbf{x}_k)^t \mathbf{B}^{-1}\mathbf{A} \tag{4.3.3}$$

consider the expressions provided below. If $\alpha = \beta = 0$, stop; $\mathbf{x}_k$ is a KKT point [4]. If $\alpha > \beta$, compute $\mathbf{d}_N$ from 4.3.6 and 4.3.8. If $\alpha < \beta$, compute $\mathbf{d}_N$ from 4.3.7 and 4.3.9. If $\alpha = \beta \neq 0$, compute $\mathbf{d}_N$ either from 4.3.6 and 4.3.8 or else from 4.3.7 and 4.3.9. In all cases, $\mathbf{d}_B$ is computed from 4.3.10. And then, go to step 2.

$$\alpha = \text{maximum } \{-r_j: r_j \leq 0\} \tag{4.3.4}$$

$$\beta = \text{maximum } \{x_j r_j: r_j \geq 0\} \tag{4.3.5}$$

$$v = \text{an index such that } \alpha = -r_v \tag{4.3.6}$$

$$v = \text{an index such that } \beta = -x_j r_v \tag{4.3.7}$$

$$d_j = \begin{cases} 0 & \text{if } j \notin I_k, j \neq v \\ 1 & \text{if } j \notin I_k, j = v \end{cases} \tag{4.3.8}$$

$$d_j = \begin{cases} 0 & \text{if } j \notin I_k, j \neq v \\ -1 & \text{if } j \notin I_k, j = v \end{cases} \tag{4.3.9}$$

$$\mathbf{d}_B = -\mathbf{B}^{-1}\mathbf{N}\mathbf{d}_N \tag{4.3.10}$$

2. Solve the following line search problem:

Minimize $\quad f(\mathbf{x}_k + \lambda \mathbf{d}_k)$
subject to $\quad 0 \leq \lambda \leq \lambda_{\max}$

where

$$\lambda_{\max} = \begin{cases} \infty & \text{if } \mathbf{d}_k \geq \mathbf{0} \\ \text{minimum} \left\{ \dfrac{-x_{jk}}{d_{jk}} < 0 \right\} & \text{otherwise} \end{cases} \tag{4.3.11}$$

x_{jk}, d_{jk} are the jth components of $\mathbf{x}_k$ and $\mathbf{d}_k$, respectively. Let λ_k be an optimal solution, and let $\mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k \mathbf{d}_k$. Replace k by k+1 and go to step 1.

Refer to [4] for verification of the convergence of the Convex-Simplex Method.

4.4 Application of the Methods to the Radar Resource Allocation Problem

To the best of our knowledge, the application of this well established family of optimization algorithms, also termed as primal methods, to the problem of resource allocation have been limited to the studies [21] and [26]. In [21], Gradient Projection Algorithm is investigated in the problem of allocation of network resources. Similarly in [26], the Gradient Projection Algorithm is studied for optimized bandwidth allocation in ad hoc networks under overload situations and the convergence properties and performance measured in terms of accumulated utility are investigated. In both of the studies, the research domain is network applications. In this thesis, we apply the Methods of Feasible directions to the radar resource allocation problem and collectively abbreviate our **F**easible **D**irections based solutions to the radar **R**esource **A**llocation problem as FDRA. Later in the following section, a discrete version of this approach for the case with discrete operating points will be presented and termed as FDRA-D with the suffix for **D**iscrete.

The methods described in the previous subsections are for the optimization problems with twice differentiable objective functions and linear constraints [4]. As the objective function is twice differentiable and the constraints are linear in the radar resource allocation problem, which is formulated in the subsection 2.3, the problem can be solved by employing one of the methods described in the previous subsections. The problem can be written in the following form,

$$\begin{aligned} \text{Minimize} \quad & f(\mathbf{x}) \\ \text{subject to} \quad & \mathbf{Ax} \le \mathbf{b} \\ & \mathbf{Ex} = \mathbf{e} \end{aligned}$$

for the Zoutendijk Algorithm with Topkis-Veinott's Modification and the Gradient Projection Algorithm. And the considered problem can also be formulated in the following form,

$$\begin{aligned} \text{Minimize} \quad & f(\mathbf{x}) \\ \text{subject to} \quad & \mathbf{Ax} = \mathbf{b} \\ & \mathbf{x} \ge \mathbf{0}. \end{aligned}$$

for the Convex-Simplex Algorithm. In the last formulation, the variables f_k and P_k (k = 1, 2, …, N) can be changed to f_k - $f_{k,min}$ and P_k - $P_{k,min}$ (k = 1, 2, …, N) and the minimum resource requirement constraints can be written in the form $\mathbf{x} \ge \mathbf{0}$. And as the objective function in the radar resource allocation problem is convex, the optimal result to the problem exist in the constraint boundary of the timing and energy constraints formulated in the subsections 2.1.1 and 2.1.2, therefore the timing and energy constraints can be written in the form Ax = b for the last formulation.

4.5 Approach for the Case with Discrete Objective Functions

As it is explained in the Chapter 3, Q-RAM approach is extended to the multiple resource type case by considering the objective functions for each of the applications of the resource allocation problem as discrete functions. In Chapter 5, the performance of the Q-RAM approach for the multiple resource type case is compared with performance of the algorithms, which are briefly explained in this chapter, on the radar resource allocation problem formulated in subsection 2.3. The continuous objective functions defined in the subsection 2.2 are considered for the Methods of Feasible Directions and discrete objective functions, which are obtained by sampling the continuous objective functions of subsection 2.2 on discrete points, are considered for the Q-RAM approach. In order to obtain an approach with better performance and theoretical background relative to the Q-RAM approach for the multiple resource type case with discrete objective functions option, the methods of feasible directions are proposed to be applied to the continuous objective functions, which are obtained by applying a curve fitting approach to the discrete objective functions.

In this subsection, the proposed approach is explained on the radar resource allocation problem of subsection 2.3 in which sampling frequency (f_k) and average power of the transmitted radar signal (P_k) of the tasks are investigated as variables of the resource allocation problem except the tracking quality functions of the tasks are assumed to be discrete functions. It is proposed to obtain exponential functions of the form of Eq. 2.2.1 best fitting to the discrete tracking quality functions of the tasks by using the least squares method [25].

Assume that there exists s discrete sampling frequency (f_{ki}, $i = 1, \ldots, s$) and t average power of the transmitted radar signal (P_{kj}, $j = 1, \ldots, t$) level options for k'th task. q_{kij} denotes the tracking quality obtained from the k'th tracking task when the sampling frequency and average power of the transmitted radar signal of the task are f_{ki} and P_{kj}, respectively, for the discrete tracking quality function of the kth task. After application of the least squares method, the α_k, β_k and m_k values of the exponential function (Eq. 2.2.1) best fitting to the discrete tracking quality function of the k'th task can be written as

$$\begin{bmatrix}\alpha_k\\ \beta_k\\ m_k\end{bmatrix}=\begin{bmatrix}\sum_{i=1}^{s}\sum_{j=1}^{t}2f_{ki}^2 & \sum_{i=1}^{s}\sum_{j=1}^{t}2f_{ki}P_{kj} & -\sum_{i=1}^{s}\sum_{j=1}^{t}2f_{ki}\\ \sum_{i=1}^{s}\sum_{j=1}^{t}2f_{ki}P_{kj} & \sum_{i=1}^{s}\sum_{j=1}^{t}2P_{kj}^2 & -\sum_{i=1}^{s}\sum_{j=1}^{t}2P_{kj}\\ -\sum_{i=1}^{s}\sum_{j=1}^{t}2f_{ki} & -\sum_{i=1}^{s}\sum_{j=1}^{t}2P_{kj} & \sum_{i=1}^{s}\sum_{j=1}^{t}2\end{bmatrix}^{-1}\begin{bmatrix}-\sum_{i=1}^{s}\sum_{j=1}^{t}2\ln(1-q_{kij})f_{ki}\\ -\sum_{i=1}^{s}\sum_{j=1}^{t}2\ln(1-q_{kij})P_{kj}\\ \sum_{i=1}^{s}\sum_{j=1}^{t}2\ln(1-q_{kij})\end{bmatrix}. \quad (4.5.1)$$

After finding the α_k, β_k and m_k parameters of the continuous tracking quality functions, which are obtained by using Eq. 4.4.1, it is proposed to apply the algorithms presented in the previous subsections. Since, the value, derived from the algorithm, drops into a range which is defined within discrete operating points (f_{ki}, P_{kj}; i = 1, …, s , j = 1, …, t), the nearest point supposed to be selected. The chosen nearest operating point should be the lowest discrete operating point within the range. As an example, assume that the sampling frequency value of a task, which is generated by one of the Methods of Feasible Directions that is applied on the best fitting exponential curves to the discrete tracking quality functions, is 76.7 Hz and the discrete sampling frequency options are 10, 20, 30, …, 100 Hz, in this case the discrete sampling frequency value to be chosen is 70 Hz. The described resource allocation approach is called as FDRA-D (**F**easible **D**irections based **R**esource **A**llocation approach for **D**iscrete objective function case) in rest of the thesis.

CHAPTER 5

EXPERIMENTAL EVALUATION

In this chapter, the resource allocation problem with multiple resource type is considered and the evaluation and comparison of the considered approaches are presented. The simulation environment in this thesis is MATLAB 7.0.1. Our experimental evaluation is intended to quantify the performance of resource allocation algorithms as applied to the radar target tracking problem. We also focus on comparing the performance of the contribution in the present study with those evaluated from the literature. As discussed in a theoretical framework in the preceding chapters; the methods we have considered are Q-RAM based methods and the Methods of Feasible Directions, in particular the Zoutendijk Algorithm with Topkis-Veinott's Modification, the Gradient Projection Algorithm and the Convex-Simplex Algorithm. We evaluate these algorithms in this chapter in terms of closeness to optimality and total execution time. We focus on measuring two main performance metrics for each algorithm:

- The global tracking quality obtained by the resource optimization and its closeness to the global optimum solution.
- The total execution time.

A proper evaluation of the algorithms considered require the selection of a reasonable simulation scenario, acceptably realistic simulation conditions and parameters as well as a formal definition for the performance measures used to compare and contrast these algorithms. We do not claim to present a fully realistic radar tracking scenario, however, we believe that the scenario considered is sufficiently illustrative and useful such that when uniformly applied to all algorithms considered, gives us a good indication of relative performance differences.

5.1 Outline of the Chapter

In this chapter, first the simulation scenario which is used in quantifying and comparing the performances of the resource allocation approaches is presented. In subsection 5.2, the parameter values of the tracking tasks in the simulation scenario and the uniform selection procedure of tasks from the task set are explained in detail. As it is explained in Chapter 3, the Q-RAM approach for the multiple resource type case requires discrete objective functions. In subsection 5.2, the generation of discrete objective functions for the simulations as well as the Monte-Carlo Simulation technique utilized is also explained.

In subsection 5.3, the performance measures for the considered resource allocation approaches are explained and the necessary definitions are presented. Some practical difficulties are encountered for the termination of the Methods of Feasible Directions in implementation. Since the speed of reaching a solution, which is one of the performance measures for the resource allocation approaches, is dependent on the termination criterion of the algorithms, the determination of the termination criterion is deemed important and its selection is discussed and explained in subsection 5.3.

The performances of the considered resource allocation approaches are evaluated first by using continuous objective functions. In subsection 5.4, the simulations with continuous objective functions are described and simulation results are presented.

As previously discussed in Chapter 3, the Q-RAM approach for the multiple resource type case is a discrete optimization approach. The objective functions of the Q-RAM approach are discrete functions for the case of multiple resource type. These consist of performance measure samples taken from the radar system operation at different operating points. With the motivation of obtaining an improvement over the Q-RAM approach in terms of closeness to the global optimal and speed of convergence, the Methods of Feasible Direction are proposed to be applied to the resource allocation problem with multiple resource type and discrete objective function.

In order to achieve this, the following approach is proposed: first, one obtains continuous objective functions from the discrete function samples by curve fitting. Then, a chosen continuous optimization method is applied to obtain a continuous globally optimal solution. The optimization algorithm chosen for our experiments is the Gradient Projection Algorithm, since it has the best convergence speed according to the simulation results presented in subsection 5.4. The last step is to discretize back the solution point. The performance of the proposed alternative approach is compared with the Q-RAM approach. In subsection 5.5 of this chapter, the simulations performed for the case of discrete objective functions are described in detail.

5.2 A Simulated Radar Target Tracking Scenario

It is important to note that we are interested in an average behavior of the system over a range of possible target-radar interactions. The first step in setting up simulation environment is the definition of a target tracking scenario. In this scenario, a given number of targets are in the field of view of the radar system. This visible target profile is a selected combination from a spectrum of targets. As it is explained in section 2.2, the parameters α_k and β_k in the objective function take different values for different tracking tasks. The maneuverability, speed, distance to the radar system of the targets being tracked lead to different α_k and β_k parameter values in the objective function of the radar target tracking problem. The initial condition parameter values $f_{k,min}$, $P_{k,min}$ and the control value of minimum achievable performance (m_k) also play role in the characterization of the tracking tasks. Based on the selection of the values of the parameters α_k, β_k, $f_{k,min}$, $P_{k,min}$ and m_k in the numerical ranges of interest; different target-radar interaction conditions can be obtained. In the following subsections, the construction of the total target spectrum and the visible sub-set are described.

5.2.1 Target Spectrum

In order to generate a statistically meaningful performance estimate for the algorithms, simulations are conducted over randomly constructed target scenarios. These scenarios are extracted from a pre-generated 15 target spectrum which is given in Table 5.2.1. We generated this table with the primary aim of uniformly sampling all parameter ranges of interest affecting the target-radar interaction. Based on the literature, we observe the following: Lee et al. selected minimum sampling frequency (f_k) and sensitivity to the sampling frequency (α_k) values in the order of 10-20 Hz and 0.01-0.1 respectively in evaluating their optimization approach to the radar problem they have examined [16]. Computation time parameter (C_k) is also selected in the order of 2-3 ms by Lee et al. in the experimental evaluation part of [16]. In this study, the minimum sampling frequency ($f_{k,min}$), sensitivity to the sampling frequency (α_k) and computation time (C_k) values are selected consistent with the experimental parameter values of [16] as shown in Table 5.2.1.

Table 5.2.1 Input Data for the Simulations Performed

Task	m_k	C_k (ms)	α_k	β_k	$f_{k,min}$ (Hz)	$P_{k,min}$ (W)
Task 01	0.8	2	0.080	0.075	15	20
Task 02	0.75	1.9	0.085	0.070	16	21
Task 03	0.9	1.8	0.090	0.065	17	22
Task 04	0.85	1.7	0.085	0.060	18	24
Task 05	0.7	1.8	0.080	0.055	19	25
Task 06	0.95	1.9	0.075	0.060	20	26
Task 07	0.9	2	0.070	0.065	21	25
Task 08	0.95	2.1	0.065	0.070	22	24
Task 09	0.8	2.2	0.060	0.075	24	22
Task 10	0.85	2.4	0.055	0.080	22	21
Task 11	0.75	2.5	0.060	0.085	21	20
Task 12	0.7	2.6	0.065	0.090	20	19
Task 13	0.9	2.5	0.070	0.085	19	18
Task 14	0.8	2.4	0.075	0.080	18	17
Task 15	0.7	2.2	0.080	0.075	17	18

Minimum average transmitted power values ($P_{k,min}$) and sensitivity to the average transmitted power (β_k) are selected in the order of 15-25 W and 0.05-0.1 respectively for experimental evaluation of the performance of the optimization approaches described previously to the radar resource allocation problem defined in this study. Sensitivity to the sampling frequency (α_k) and sensitivity to the average power of the transmitted signal (β_k) values of the tracking tasks in Table 5.2.1 is varied in order to obtain different cases for the simulation scenarios. The variation of α_k and β_k parameter values provides different scenario conditions. In the simulation scenarios, different initial conditions that are obtained by selecting different $f_{k,min}$ and $P_{k,min}$ parameter values from the task list of Table 5.2.1 also provides various simulation conditions which enables performing simulations that are unbiased from the specific scenario conditions.

The control value of minimum achievable performance (m_k) is selected in the range 0.5-1.The range 0.5-1 for the control value of minimum achievable performance parameter is selected in order to provide numerical examples for the comparative simulations.

It can be argued that a set of 15 alternative targets is too small to span the entire parameter space created by the set of parameters considered. However, the aim of the study is not to provide an estimate of absolute performance but a comparative evaluation of different alternatives against each other. Therefore it is believed that the target spectrum chosen is reasonably diverse and hence provides a promising ground for comparison of performance.

5.2.2 Target Sub-selection

The primary aim of generating different scenarios having the same number of tasks is to obtain an averaged performance figure for a given number of tasks on-scene. For this purpose, we obtain performance figures for each such scenario and obtain estimates for the first order statistics, namely the sample mean and sample variance figures computed over K such experiments. We wish to pick K large enough to have statistically meaningful performance while dealing with realizable experimental (simulation) time. Hence, we have chosen K in the following manner.

For the comparative simulations, scenarios containing different number of tasks are considered. The number of tasks contained in the scenarios is varied from 1 to 11 while performance measures considered are evaluated for each algorithm considered. These tasks are chosen from the set of M=15 tasks given in Table 5.2.1. The spectrum of 15 different targets allows us to construct different combinations of targets, each with the same number of targets on scene. For example, one can construct a total number of 3003 target scenarios, each with 5 targets on scene. The total numbers of possible scenarios are given in Table 5.2.2 as a function of number of targets on scene. N denotes the number of tracking tasks in the scenario and M_N denotes the number of total combination of simulation cases that can be obtained from the task set of Table 5.2.1 when there are N tasks in the scenario.

Table 5.2.2 Numbers of Total Combinations for Different Number of Tasks

Number of Tasks in the Scenario (N)	Number of Total Combination of Different Simulation Cases (M_N)
1	15
2	105
3	455
4	1365
5	3003
6	5005
7	6435
8	6435
9	5005
10	3003
11	1365

M_N can be obtained using the following equation:

$$M_N = \frac{M!}{(M-N)!N!} \tag{5.2.1}$$

As it can be observed from Table 5.2.2, the number of different simulation cases (M_N) is not equal for different number of tasks included in the simulation scenario (N). Equal numbers of different simulation cases for different number of tasks in the simulation scenario are considered in order to make comparative simulations on the behaviors of the algorithms when the number of tracking tasks in the simulation scenario changes. 400 simulation cases are selected out of M_N simulation cases for $N \geq 3$; for $N \leq 2$ there is not enough different simulation cases for obtaining 400 different simulation scenarios. 15 and 105 different scenarios that are obtained from the task set of Table 5.2.1 is used for $N = 1$ and $N = 2$ cases respectively. We expect that this will introduce some deteriorated variance for these two cases.

400 simulation cases are selected randomly out of M_N simulation cases for each N ($N = 3, 4, \ldots, 11$). Uniformly distributed random selection is accomplished by applying the *rand* function in MATLAB. The *rand* function generates arrays of random numbers whose elements are uniformly distributed in the interval (0, 1). 400 simulation cases are selected by using the following algorithmic approach:

```
Scenario_Set = Generate_Scenario_Set(M_N_Scenario_Set)
1.  Scenario_Set = {};
2.  Divide (0, 1) interval into equal M_N slices;
3.  Generate a random number in the interval (0, 1) → x=rand;
4.  Find in which slice x is located (say k^th slice);
5.  If k^th scenario of M_N scenarios is in Scenario_Set
      a.  Then return to step 3,
      b.  Else add k^th scenario of M_N scenarios to the
          Scenario_Set,
6.  If number of scenarios in the Scenario_Set is equal to 400
      a.  Then stop,
      b.  Else return to step 3.
```

In this algorithmic approach, *M_N_Scenario_Set* is the input scenario set where scenarios are obtained from a task set of M tasks and each scenario contains N tracking tasks.

There are M_N different scenarios in the *M_N_Scenario_Set*. *Scenario_Set* is the set containing 400 different scenarios selected from the *M_N_Scenario_Set*.

5.2.3 Discrete Profile Generation

As discussed in Chapter 3, Q-RAM based algorithms are discrete in nature for the case of multiple resource type and have need for discrete variables to operate. In order to have a basis for comparison, we need the same tracking quality functions to be used for evaluation of the considered approaches. Therefore, discrete tracking quality functions are generated by taking samples of the continuous tracking quality functions described in section 2.2. For a continuous tracking quality function for which frequency changes in the range 25 - 165 Hz, power changes in the range 25 - 165 W, m_k is equal to 1, α_k is equal to 0.01, β_k is equal to 0.015 and that is shown in Figure 5.2.1, discrete tracking quality curve is obtained as shown in Figure 5.2.2.

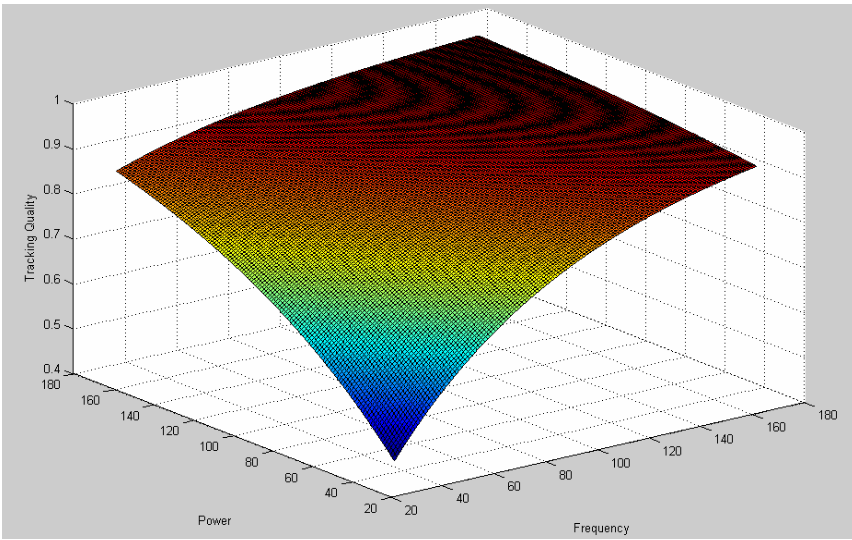

Figure 5.2.1 Continuous Tracking Quality Function. For this plot, the parameters m_k is equal to 1, α_k is equal to 0.01, β_k is equal to 0.015 in the tracking quality function.

In the discrete tracking quality function shown in Figure 5.2.2, frequency increases from 25 Hz to 165 Hz with step of 10 Hz, power increases from 25 W to 165 W with step of 10 W. 15 different frequency levels are considered for the frequency axis and for the power axis also 15 different power levels are considered. From these selections, for a tracking task there are 15 × 15 = 225 discrete operating points that can be selected. If the number of tasks included in the scenario increases, the number of discrete operating point to be searched also increases. If N is the number of tasks included in the scenario and D is the total number of discrete points to be searched for a task, the total number of discrete points to be searched for the global optimal solution (D_{TOTAL}) is:

$$D_{TOTAL} = D^{N} \tag{5.2.2}$$

For all of the scenario combinations obtained from Table 5.2.1, 15 discrete frequency levels and 15 discrete power levels are considered for the discrete tracking quality functions that will be fed to the Q-RAM Algorithm. Discrete sampling frequency and average power of the transmitted signal levels are selected beginning from the minimum power requirements of the tracking tasks. Each discrete operating point is feasible when minimum resource requirements of the tasks are considered. For a scenario containing N tracking tasks there are 225^{N} discrete operating points that can be selected.

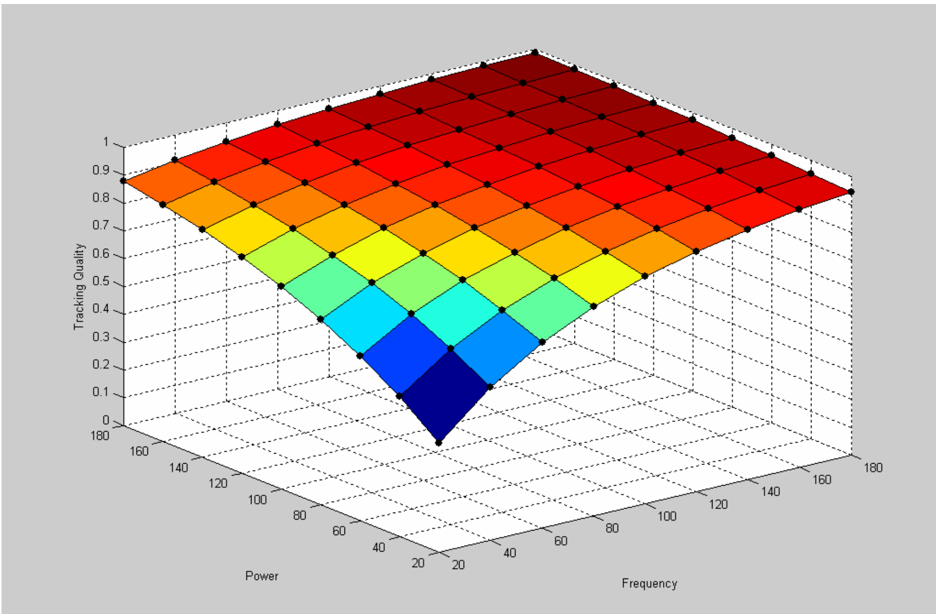

Figure 5.2.2 Discrete Tracking Quality Function. The discrete tracking quality function is generated by taking samples on the continuous tracking quality function of Figure 5.2.1.

The number of discrete operating points depending on the number of tasks included in the scenario is presented in Table 5.2.3 when there exists 225 discrete operating points for one tracking task.

Table 5.2.3 Number of Discrete Operating Points Depending on Number of Tasks

Number of Tasks	Total Number of Discrete Operating Points
1	225
2	50625
3	11390625
4	2.5629e+009
5	5.7665e+011
6	1.2975e+014
7	2.9193e+016
8	6.5684e+018
9	1.4779e+021
10	3.3253e+023
11	7.4818e+025

5.2.4 Monte-Carlo Simulations

As partly discussed in the previous section, in our performance evaluation, the aim is to illustrate the expected behavior for a scenario with known number of visible targets. It is also known that the particular set of visible targets can be selected from the total available target spectrum in more than one way since there are M_N combinations of N targets out of 15. The expected or average behavior for a given number of targets can be estimated by performing simulations with a large number of possible selections from the available spectrum where each selection having the same fixed number of visible targets N. This estimate can be derived by averaging the performance data over this set of multiple simulations and the standard deviation can be used as a basic measure of the confidence to the results. The sample average and the sample standard deviation are computed according to the following definitions.

Standard deviation s and mean $\overline{x}$ of data vector $\mathbf{x}$ is [3]:

$$\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{5.2.3}$$

and

$$s = \left(\frac{1}{n}\sum_{i=1}^{n} (x_i - \overline{x})^2 \right)^{\frac{1}{2}} \tag{5.2.4}$$

n is the number of samples in the result set.

5.3 Performance Measures

As it is indicated at the beginning of the chapter, the performance measures for the observed optimization approaches are closeness of the solution to global the optimal and the speed of reaching a solution. In the following subsections the performance measures are elaborated.

5.3.1 Closeness to the Optimal

In order to evaluate the performance of the algorithms in terms of the closeness to the optimal, the measure called ‘optimality percentage’ (OP) is introduced as discussed below. Optimality is assessed over the objective function which is the total tracking quality. In the definition below, *TQ* denotes the total tracking quality obtained from the result of the algorithm relative to the minimum resource requirement point and TQ^{opt} denotes the tracking quality obtained from the global optimum operating point relative to the minimum resource requirement point. Because the cost function and constraints defined in the optimization problem in subsection 2.3 are continuous and convex, the results of the Zoutendijk Algorithm with Topkis-Veinott’s Modification, Gradient Projection and Convex-Simplex Algorithms ensure the necessary and sufficient conditions for global optimality [4]. So the optimality percentage of Zoutendijk Algorithm with Topkis-Veinott’s Modification, Gradient Projection and Convex-Simplex Algorithms are %100; that is, the ratio of the tracking quality obtained by using these algorithms to the global optimal tracking quality is one.

$$\text{OP} = (\ TQ\ /\ TQ^{opt}\ ) \times 100 \tag{5.3.1}$$

However in the case of the implementation of the Q-RAM based approach, it not guaranteed that the total tracking quality obtained from the resultant point of the algorithm is equal to the global optimum value. As it is mentioned in Chapter 3, the main objective of the Q-RAM model is to reach a solution that is closest to the optimal with a high convergence speed which can enable the model to be applied in real-time applications. The distance to the global optimal value of the performance measure makes it possible to evaluate the performance of the Q-RAM based approach experimentally in terms of optimality. Our primary aim here is thus to see what is the compromise between speed and optimality for all these algorithms considered.

5.3.2 Speed of Reaching a Solution

For the resource allocation approaches to be applied in real-time applications, it is important for the algorithms to reach the final solution in the shortest amount of time. For radar resource allocation problem, targets arrive and leave the field of view of the radar system dynamically. The tracking task list of the radar system changes with arrival and departure of the targets in the environment of the radar system. Radar system should also reconfigure the operating parameters considered in the radar resource allocation problem (f_k, P_k) of a target when the speed and distance of the target changes. The reconfiguration of the f_k and P_k parameters of the tracking tasks in the real-time environments requires the resource allocation algorithms to be applied under the real-time timing constraints. Although the absolute requirements for the speed will vary between systems, a faster algorithm will usually mean better applicability.

In the present study, we do not have a computational complexity analysis of the formal algorithms considered. Furthermore, iterative algorithms rely on a termination criterion to reach a solution. Instead, we opted for measuring execution time with the rationale that this will give an objective ground for comparison when all algorithms are executed on the same computational platform. In the following subsection, the measurement of the execution times of the optimization approaches in MATLAB is presented. When the optimization algorithms, which are presented in Chapter 4, are implemented in MATLAB, some practical difficulties are encountered on the termination phase of the algorithms. Termination criterion of an algorithm is important in order to measure the execution time of the algorithm exactly. The encountered difficulties regarding the termination of the algorithms and the proposed termination approach are explained in subsection 5.3.2.2.

5.3.2.1 Measurement of the Execution Time

In order to compare the performances of the optimization methods intended to solve the previously defined radar target tracking problem, MATLAB is used as the simulation tool. Simulations are performed on MATLAB 7.0.1 that is running on a computer with AMD Athlon 64 3200 processor, 1 GB RAM and Windows XP Operating System. For CPU time calculation, the execution priority of the MATLAB is made ‘Real-Time’ to avoid CPU being used by other applications.

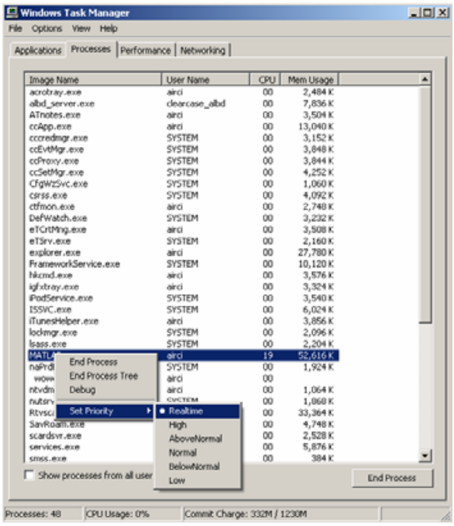

Figure 5.3.1 Making MATLAB a Real-Time Application in Windows OS

MATLAB function *cputime* is used to measure the total execution time of an algorithm. The function *cputime* returns the total CPU time (in seconds) used by MATLAB from the time it was started. For example, the following code is used to measure the total execution time of the Gradient Projection Algorithm for a specific input scenario *Input_Scenario* and power limit *Pmax*.

```
CurrentTime=cputime;
[OP,No_of_Iterations]=GradientProjection(Input_Scenario,Pmax);
ElapsedTime=cputime-CurrentTime;
```

In this code *OP* is the output resultant operating point, *No_of_Iterations* is the total number of iterations performed by the algorithm in order to reach the final resultant operating point and *ElapsedTime* is the total execution time of the algorithm.

5.3.2.2 Termination Criterion of the Algorithms

Some difficulties regarding termination of the algorithms are encountered in implementation of the Zoutendijk Algorithm with Topkis-Veinott's Modification, Gradient Projection and Convex-Simplex Algorithms in MATLAB. In the following paragraphs the encountered difficulties are explained and solutions utilized are discussed:

Zoutendijk Algorithm with Topkis-Veinott's Modification:

In Zoutendijk Algorithm Topkis-Veinott's Modification, the algorithm is terminated when the parameter 'z' becomes zero since the operating point, to which the algorithm is converged, is a KKT point as it is described in section 4.1.2 and in [4]. However, in most of the cases the parameter 'z' never becomes zero; there is no uniform convergence to zero but to a value near zero with a bias in practical implementations. For example, in some cases it converges to values in the neighborhood of -1e-10, in some other cases it converges to values in neighborhood of -1e-15, etc. The value to which the 'z' parameter converges, changes according to the different scenarios used. Any threshold value of 'z' parameter that can be used for the termination of the algorithm can not be determined because in some scenarios the operating points converge at a specified 'z' value, in some other scenarios the operating points converge at another specified 'z' value. As a result, some other termination criterion should be determined and applied in the practical implementations of the algorithm.

Gradient Projection Algorithm:

Similar situation occurs in the case of implementation of the Gradient Projection algorithm. When the direction vector (**d**) becomes zero and also the vector $(\mathbf{MM}^T)^{-1}\mathbf{M}\nabla f(\mathbf{x}_k)$ is greater than or equal to zero, the algorithm is terminated as the point to which the algorithm converged is a KKT point as it is explained in section 4.2.2 and in [4]. But, similar to the situation explained in the previous paragraph, in most of the cases the magnitude of the direction vector never becomes zero when the algorithm is implemented. The magnitude of the direction vector converges to values very near to zero; in some scenarios the value considered converges to a value in the neighborhood of 1e-15, in some other scenarios it converges to a value in the neighborhood of 1e-20, etc. The value, to which the magnitude of the direction vector converges, changes according to the different scenarios. Similar to the

Zoutendijk Algorithm with Topkis-Veinott's Modification, some other termination criterion should be determined and applied in the practical implementations also for the Gradient Projection algorithm.

Convex-Simplex Algorithm:

From [4] and section 4.3.2, the Convex-Simplex Algorithm is terminated when the case $\alpha = \beta = 0$ occurs. But, similar to the cases encountered in Zoutendijk Algorithm with Topkis-Veinott's Modification and Gradient Projection Algorithms explained in previous two paragraphs, the parameters α and β of the Convex-Simplex Algorithm never become zero in practical implementation of the algorithm. At some of the iterations of the algorithm in which the specified parameters (α, β) are very near to zero, the operating point vector (remember that the scalar components of the operating point vector are parameters of the optimization problem) changes considerably. Therefore, closeness to zero of α and β can not be used as a termination criterion for the algorithm in none of the cases.

The following approach [4] is used as termination criterion for the algorithms of the method of feasible directions, which are presented in Chapter 4, in this thesis.

Terminating the Algorithms:

In order to overcome the explained practical difficulties encountered in the implementation of the algorithms, one can use uniformly the difference between successive operating points as the algorithms converges instead of the different specific convergence criteria of the three algorithms[4]. This will be illustrated in the following paragraphs.

Let $\mathbf{O}_k$ denote the operating point vector in the k^{th} iteration, which is defined as follows: $\mathbf{O}_k = \left[f_{1,k}\ P_{1,k}\ f_{2,k}\ P_{2,k} \ldots f_{N,k}\ P_{N,k} \right]^T$ for N targets, where $f_{n,k}$ and $P_{n,k}$ are the sampling frequency and the average power of transmitted signal allocated to n^{th} target (n = 1, 2, …, N) at k^{th} iteration, respectively. The normalized difference vector ($\mathbf{N}_k$) between the successive iterations is defined as follows:

$$
\mathbf{N}_k = \begin{bmatrix} \left|(f_{1,k} - f_{1,k-1}) / f_{1,k-1}\right| \\ \left|(P_{1,k} - P_{1,k-1}) / P_{1,k-1}\right| \\ \left|(f_{2,k} - f_{2,k-1}) / f_{2,k-1}\right| \\ \left|(P_{2,k} - P_{2,k-1}) / P_{2,k-1}\right| \\ . \\ . \\ . \\ \left|(f_{N,k} - f_{N,k-1}) / f_{N,k-1}\right| \\ \left|(P_{N,k} - P_{N,k-1}) / P_{N,k-1}\right| \end{bmatrix} \tag{5.3.2}
$$

The Euclidian norm and infinity norm (maximum scalar component of the vector) of the $\mathbf{N}_k$ vector can be used as a termination criterion for the considered algorithms. The objective function is a function of the operating point vector by definition. Provided that the objective function obeys a certain smoothness (which is the case of our exponentially constructed objective function), one would expect that the objective function value reaches a steady state as the norm of the change in the operating point vector approaches zero. The 'z' parameter in the Zoutendijk Algorithm with Topkis-Veinott's Modification, the direction vector **d** in the Gradient Projection algorithm and the parameters α and β in Convex-Simplex Algorithm are directly calculated from the operating point vector ($\mathbf{O}_k$). Similar to the objective function, when the change in the $\mathbf{O}_k$ between successive iterations is very small, the change in the considered parameters is also negligible.

In order to determine a threshold value on the norm of the $\mathbf{N}_k$ vector, some examples are considered and amount of change in the optimization variables with the amount of change in the norm of the $\mathbf{N}_k$ vector is investigated. In the following paragraphs, an example illustrating the determination of the termination criterion of the algorithms is described.

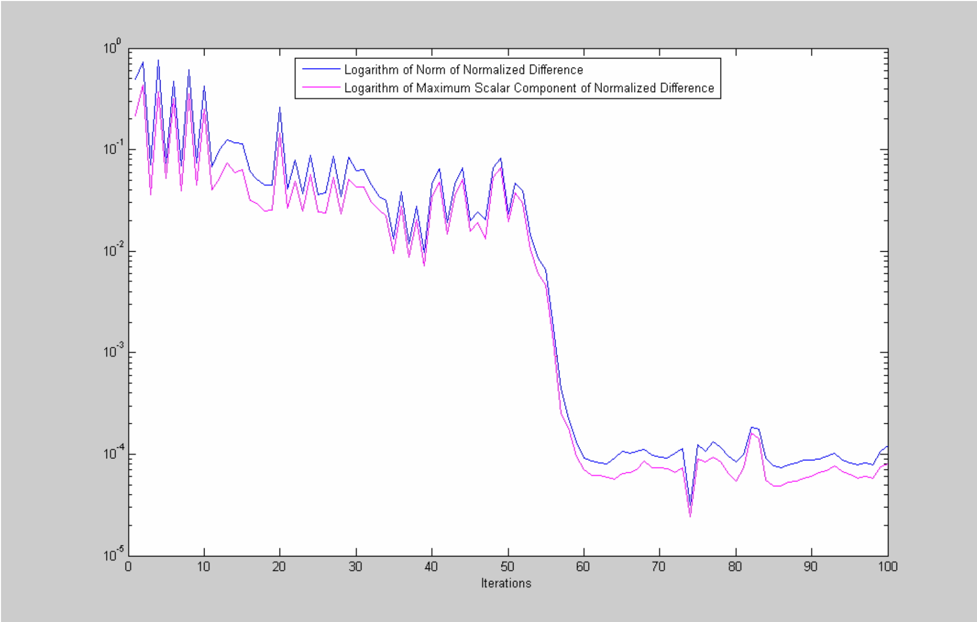

Figure 5.3.2 Variation of the norm of $\mathbf{N}_k$ vector with iterations of the Zoutendijk Algorithm with Topkis-Veinott's Modification, in Logarithmic Scale. The numerical values presented in Table 5.3.1 are used for the optimization problem parameters for this example.

The scenario used in the simulations contains five targets and tracking quality function parameters of the targets are provided in Table 5.3.1. The m_k parameters of all of the tracking functions are selected as $m_k = 1$ in these simulations. As it is shown in Table 5.3.1, five tracking tasks are considered in the simulation scenario. The variations of the optimization variables (f_k, P_k) of Task 01 with algorithm iterations are presented in Figure 5.3.5, Figure 5.3.6 and Figure 5.3.7 respectively for the Zoutendijk Algorithm with Topkis-Veinott's Modification, the Gradient Projection Algorithm and the Convex-Simplex Algorithm. It can be observed from these figures that the sampling frequency parameter of Task 01 converges at around 60^{th} iteration for the Zoutendijk Algorithm with Topkis-Veinott's Modification, 55^{th} iteration for the Gradient Projection Algorithm and 150^{th} iteration for the Convex-Simplex Algorithm. The average power of the transmitted radar signal parameter of Task 01 converges respectively at around 60^{th}, 60^{th} and 160^{th} iterations for the same three algorithms considered. Similar convergence figures can also be observed for other tracking tasks in the simulation scenario. Since the presented figures of Task 01 are considered adequate to provide an opinion about the convergence behaviors of the algorithms, the variation of optimization variables for other tasks are not presented on individual figures and rather, only the convergence iterations for these cases are presented in Table 5.3.2.

Table 5.3.1 Parameters of the Simulation Scenario for Determining the Termination Criterion.

Task	C_k (ms)	α_k	β_k	$f_{k,min}$ (Hz)	$P_{k,min}$ (W)
Task 01	5	0.08	0,06	25	45
Task 02	6	0.07	0,05	41	25
Task 03	5	0.06	0,07	29	25
Task 04	6	0.09	0,05	20	50
Task 05	5	0.07	0,06	25	41

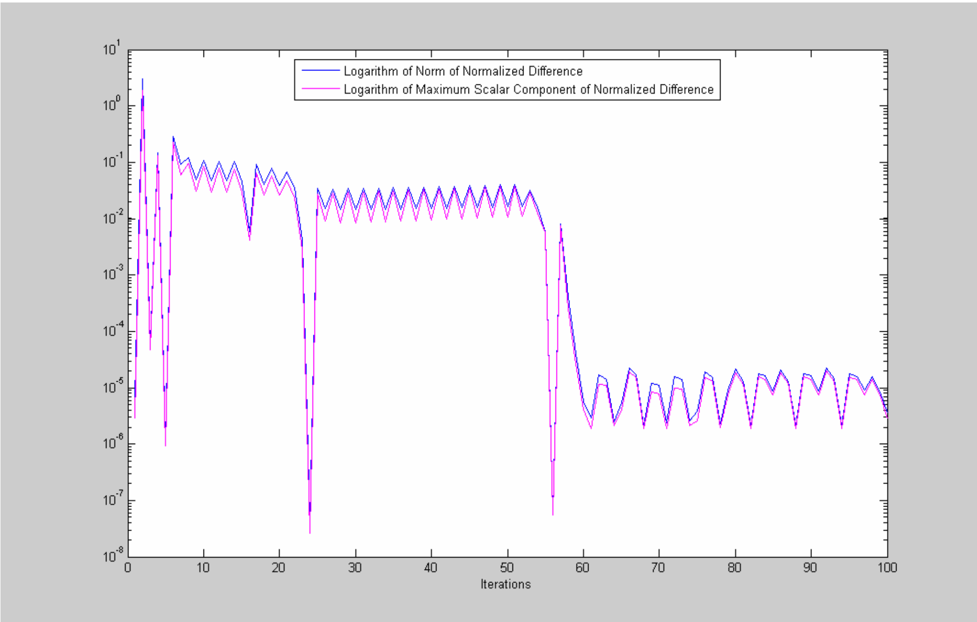

Figure 5.3.3 Variation of the norm of $\mathbf{N}_k$ vector with iterations of the Gradient Projection Algorithm presented in Logarithmic Scale. The numerical values presented in Table 5.3.1 are used for the optimization problem parameters for this example.

When the variation of optimization variables of Task 01 for the Zoutendijk Algorithm with Topkis-Veinott's Modification and Table 5.3.2 are observed, it can be concluded that the algorithm converges at around 60th iteration; this is also true for the Gradient Projection algorithm after observation of the Figure 5.3.6 and Table 5.3.2. As it is shown in Figure 5.3.7 the optimization variables of Task 01 reaches steady state at around 160th iteration. Table

5.3.2 also illustrates that 160 is also the highest convergence iteration value of the optimization variables for all the other tasks; therefore it can be concluded that the Convex-Simplex Algorithm converges at 160th iteration.

At around 60th iteration, both Euclidian norm and infinity norm of the normalized difference vector ($\mathbf{N}_k$) decreases to values in the neighborhood of 1e-4. This fact can be observed in Figure 5.3.2 and Figure 5.3.3 respectively for the Zoutendijk Algorithm with Topkis-Veinott's Modification and Gradient Projection Algorithms. For the Convex-Simplex Algorithm, both Euclidian norm and infinity norm of the normalized difference vector ($\mathbf{N}_k$) decreases to values in the neighborhood of 1e-4 at around 160th iteration as shown in the Figure 5.3.4.

It appears this change can reliably be used as a termination criterion for the algorithms. When the magnitude of $\mathbf{N}_k$ takes values in the neighborhood of 1e-4 or below in the successive iterations, the change in the cost function and operating parameters of the targets are very close to zero.

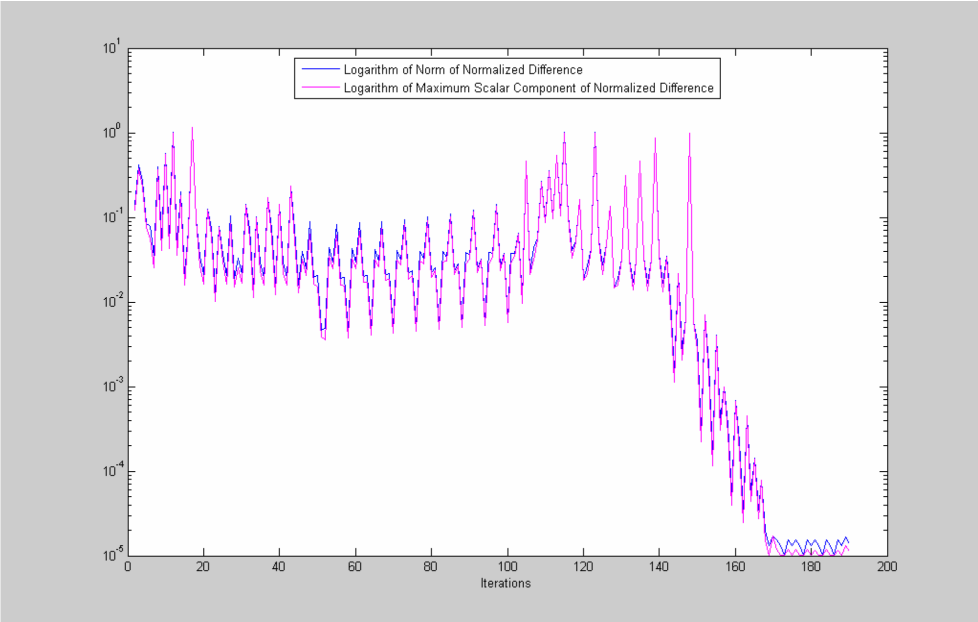

Figure 5.3.4 Variation of the norm of $\mathbf{N}_k$ vector with iterations of the Convex-Simplex Algorithm, in Logarithmic Scale. The numerical values presented in Table 5.3.1 are used for the optimization problem parameters for this example.

In the following figure, variations of sampling frequency (f_k) and average power of the transmitted signal (P_k) values of Task 01 with iterations of the Zoutendijk Algorithm with Topkis-Veinott's Modification for the tracking tasks of Table 5.3.1 are presented. Both the power and the frequency of the Task 01 converge at around 60th iteration when the Zoutendijk Algorithm with Topkis-Veinott's Modification is employed for the resource allocation.

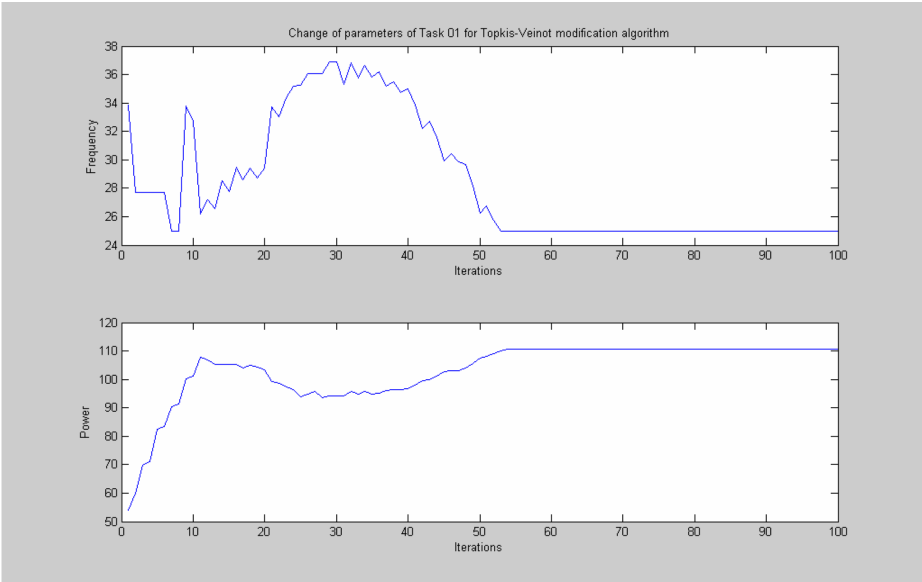

Figure 5.3.5 Variation of Optimization Variables of Task 01 for the Zoutendijk Algorithm with Topkis-Veinott's Modification. Both the power and the frequency of the Task 01 converge at around 60th iteration when the Topkis-Veinott's Modification Algorithm is employed for the resource allocation.

Figure 5.3.6 shows variations of sampling frequency (f_k) and average power of the transmitted signal (P_k) values of Task 01 with iterations of the Gradient Projection Algorithm for the tracking tasks of Table 5.3.1.

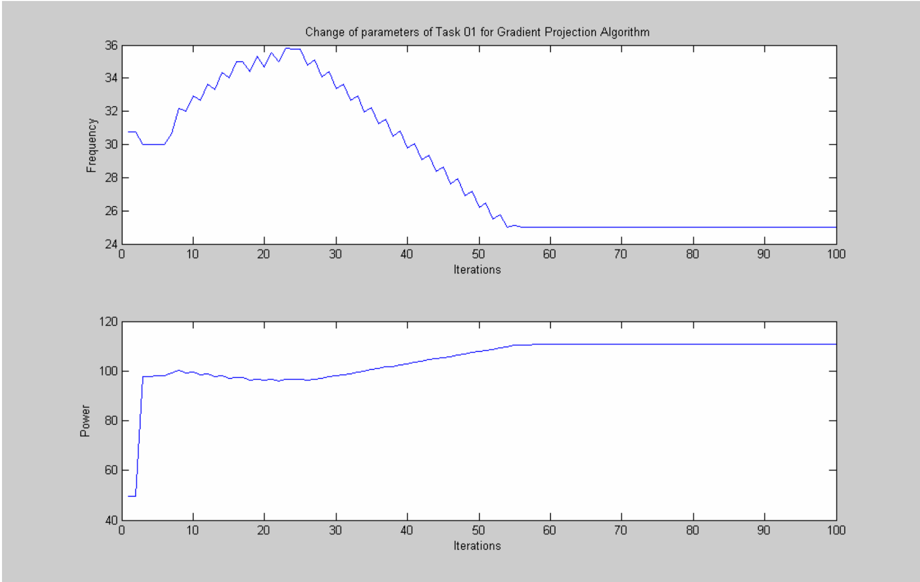

Figure 5.3.6 Variation of Optimization Variables of Task 01 for the Gradient Projection Algorithm. Both the frequency of the Task 01 converge at around 55th iteration and the power of Task 01 converge at around 60th iteration when the Gradient Projection Algorithm is employed for the resource allocation.

Variations of sampling frequency (f_k) and average power of the transmitted signal (P_k) values of Task 01 with iterations of the Convex-Simplex Algorithm for the tracking tasks of Table 5.3.1 are presented in Figure 5.3.7.

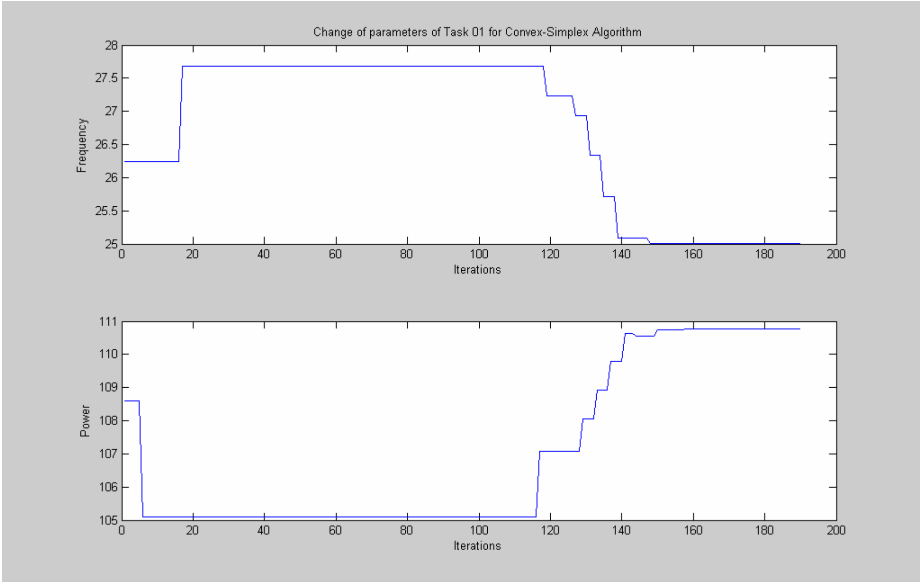

Figure 5.3.7 Variation of Optimization Variables of Task 01 for the Convex-Simplex Algorithm. Both the power and the frequency of the Task 01 converge at around 150th iteration when the Convex-Simplex Algorithm is employed for the resource allocation.

Table 5.3.2 Convergence Iterations Optimization Variables of Task 02, Task 03, Task 04 and Task 05 of Table 5.2.1. The Zoutendijk Algorithm with Topkis-Veinott's Modification and the Gradient Projection Method converge at around 60th iteration and the Convex-Simplex Method converges at around 160th iteration. At around these iterations, the norm of $\mathbf{N}_k$ falls under 1e-4 for the considered algorithms.

Optimization Variable	Convergence Iteration		
	Zoutendijk Algorithm Topkis-Veinott's Modification	Gradient Projection Algorithm	Convex-Simplex Algorithm
f_1	60	55	150
P_1	60	60	150
f_2	34	22	115
P_2	60	28	150
f_3	18	5	15
P_3	55	20	150
f_4	55	60	140
P_4	55	60	150
f_5	18	15	125
P_5	60	28	160

One can conclude from these results that the norm values of $\mathbf{N}_k$ can be used as a reliable termination criterion for the considered algorithms. The considered norm avoids the practical difficulties encountered in the termination of these algorithms in practical implementations. As it is shown in the figures presented above, along with the decrease of values of the vector $\mathbf{N}_k$ to the order of 1e-4, all the optimization parameters appear to have converged. Simulations with different scenarios which contain different number of targets are also performed and also support this observation. Therefore, we pick in this study, the decrease of the norm of $\mathbf{N}_k$ vector to below 1e-4 as a termination criterion for the investigated Methods of Feasible Directions.

5.4 Comparative Simulations with Continuous Objective Functions

In the following subsections, the performances of the optimization approaches proposed to the radar resource allocation problem are presented based on the simulated radar target tracking scenario and the performance measures explained in the previous subsections. In these simulations, continuous tracking quality functions of the form Eq. 2.2.1 for each tracking task are used and discrete functions are generated from the continuous tracking quality functions for the simulations performed using Q-RAM.

5.4.1 Results of the Zoutendijk Algorithm with Topkis-Veinott's Modification

The experimental results for the Topkis-Veinott's Modification Algorithm are presented in Table 5.4.1 and Figure 5.4.1. In the results shown in Table 5.4.1, different scenarios are used as inputs for each simulation and minimum, mean and maximum run times of these simulations are presented. As it can be observed from Table 5.4.1, all run time values increases with increase of number of tasks as expected. For each task, two optimization parameters (sampling frequency (f_k), average power of the transmitted radar signal (P_k)) are found as explained in the previous sections. The number of such optimization parameters is $2N$ for each tracking radar problem containing N tracking tasks. It is expected for run time of the algorithm to increase with increase of the tracking tasks in the scenario.

Table 5.4.1 Run Time of Topkis-Veinott's Modification Algorithm

Number of Targets	Best Run Time Value (sec.)	Mean Run Time Value (sec.)	Worst Run Time Value (sec.)
1	0.03125	0.065625	0.39063
2	0.015625	0.038437	0.046875
3	0.03125	0.044375	0.046875
4	0.046875	0.3675	0.54688
5	0.1875	0.93387	1.4219
6	0.32813	1.293	1.625
7	0.34375	1.0851	1.3438
8	0.60938	2.3752	3.5938
9	1.1094	4.1363	6.9844
10	2.2656	13.677	22.625
11	3.2344	23.564	28.844

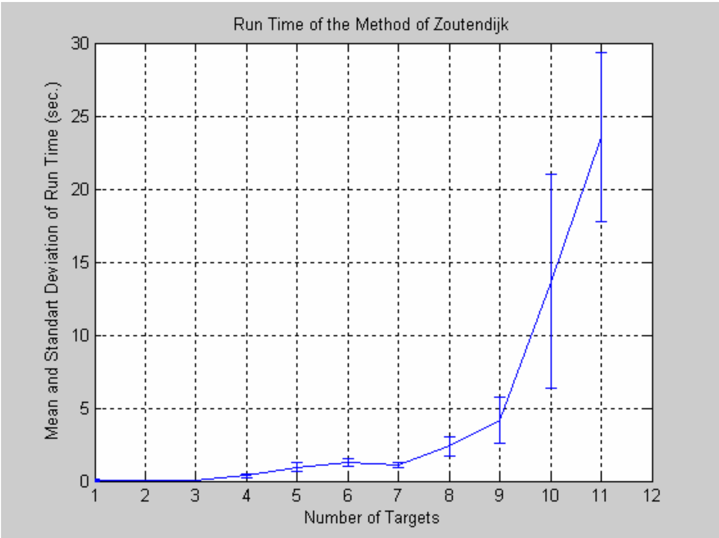

Figure 5.4.1 Run Time of Topkis-Veinott's Modification Algorithm

Figure 5.4.1 shows the variation of mean run time with number of tasks included in the simulation scenario. Bars at mean run time values show standard deviation of the run time from the experiments. The figure makes it clear that the mean run time increases with exponential characteristic with increase of number of tasks in the scenario.

Table 5.4.2 Number of Iterations of Topkis-Veinott's Modification Algorithm

Number of Targets	Number of Iterations (Average)	Run Time Per Iteration (sec.)
1	4.2	0,015625
2	4.1	0,0093749
3	4	0,011094
4	29.7	0,012374
5	70.5	0,013246
6	81.1	0,015943
7	80.6	0,013463
8	141.6	0,016774
9	227.7	0,018166
10	619.4	0,022081
11	1011.9	0,023287

When Table 3.3.1 is observed, it can be concluded that number of iterations and run time of the algorithm per iteration increases as the number of tracking tasks increases. With increase of the number of tracking tasks in the simulation scenario, the number of optimization parameters that are processed at each iteration of the algorithm increases. A linear programming problem is solved at each step of the algorithm in order to determine the best improving feasible direction. With increase of the number of tasks in the simulation scenario, variables of the linear programming problem increases. More computation time is required for solving the direction finding problem with more parameters. Therefore, run time per iteration of the algorithm increases with increase of the number of tasks.

5.4.2 Results of the Gradient Projection Algorithm

Simulation results of the Gradient Projection Algorithm are presented in Table 5.4.3, Figure 5.4.2 and Table 5.4.4. Similar to the results of the Topkis-Veinott's Modification algorithm, run time of the Gradient Projection Algorithm increases with increase of number of tasks included in the scenarios as shown in Table 5.4.3.

Table 5.4.3 Run Time of Gradient Projection Algorithm

Number of Targets	Best Run Time Value (sec.)	Mean Run Time Value (sec.)	Worst Run Time Value (sec.)
1	0.0045312	0.0046146	0.005
2	0.015625	0.014687	0.015625
3	0.03125	0.055	0.0625
4	0.046875	0.070703	0.078125
5	0.046875	0.069141	0.078125
6	0.03125	0.071484	0.078125
7	0.015625	0.068633	0.078125
8	0.015625	0.072734	0.078125
9	0.046875	0.080352	0.09375
10	0.015625	0.086836	0.10938
11	0.03125	0.10492	0.125

Figure 5.4.2 shows variation of mean run time for $N = 1$ to $N = 14$. The variation of run time of the Gradient Projection Method exhibits an exponential characteristic in the interval $4 \leq N \leq 11$. In order to see whether the exponential increase characteristic of run time, the number of tasks included in the scenario is increased and averaged performance of the Gradient Projection Method is evaluated for the cases of $N = 12$, 13 14. Although the behavior is not as clear as the previous case, it can be concluded within the standard deviations that the mean run time of the Gradient Projection Algorithm increases almost linearly with increase of number of tasks included in the scenario. The Gradient Projection Algorithm has better execution times than the Method of Zoutendijk with Topkis-Veinott's modification. Because of the exponential versus linear behavior, the difference between run times of the Gradient Projection and Topkis-Veinott's Modification Algorithms becomes more distinct with the increase in the number of tasks.

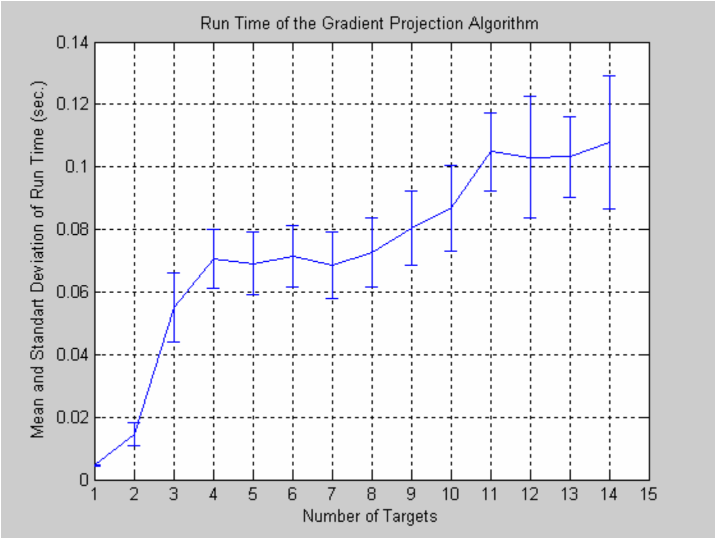

Figure 5.4.2 Run Time of Gradient Projection Algorithm

Table 5.4.4 shows average number of iterations and average run time for each iteration of Gradient Projection Algorithm. Increase of number of tasks included in the scenario leads to increase of number of calculations at each iteration. As shown in Table 5.4.4 average run time per iteration increases. Run time per iteration in Table 5.4.4 is obtained by dividing mean run time value in Table 5.4.3 to the mean iteration value in Table 5.4.4.

Table 5.4.4 Number of Iterations of Gradient Projection Algorithm

Number of Targets	Number of Iterations (Average)	Run Time Per Iteration (sec.)
1	6	0.0007691
2	13.76	0.0010674
3	46.48	0.0011833
4	56.145	0.0012593
5	53.72	0.0012871
6	53.915	0.0013259
7	52.102	0.0013173
8	52.915	0.0013745
9	56.7	0.0014171
10	59.138	0.0014684
11	68.028	0.0015423

5.4.3 Results of the Convex-Simplex Algorithm

Table 5.4.5, Figure 5.4.3 and Table 5.4.6 shows the simulation results of the Convex-Simplex Algorithm. It observed from the presented results of Table 5.4.5 that similar to the simulation results of the algorithms previously presented, run time of the Convex-Simplex algorithm also increases with increase of the number of tasks as expected.

Table 5.4.5 Run Time of Convex-Simplex Algorithm

Number of Targets	Best Run Time Value (sec.)	Mean Run Time Value (sec.)	Worst Run Time Value (sec.)
1	0.00015625	0.00025	0.0003125
2	0.00015625	0.00016875	0.0003125
3	0.00015625	0.00015625	0.00015625
4	0.0625	0.098828	0.10938
5	0.34375	0.55297	0.65625
6	0.64063	0.91027	1.0313
7	0.64063	0.94781	1.0781
8	0.875	1.2624	1.4063
9	1	1.3557	1.5
10	1.2031	1.7225	1.9063
11	1.1875	1.8709	2.1406

Figure 5.4.3 provides a plot of the mean run time and its standard deviation as error bars for the case of Convex-Simplex Algorithm for $N = 1$ to $N = 11$. Run time of Convex-Simplex Algorithm shows almost a linearly increasing characteristic with increasing number of tasks included in the scenario as shown in Figure 5.4.3.

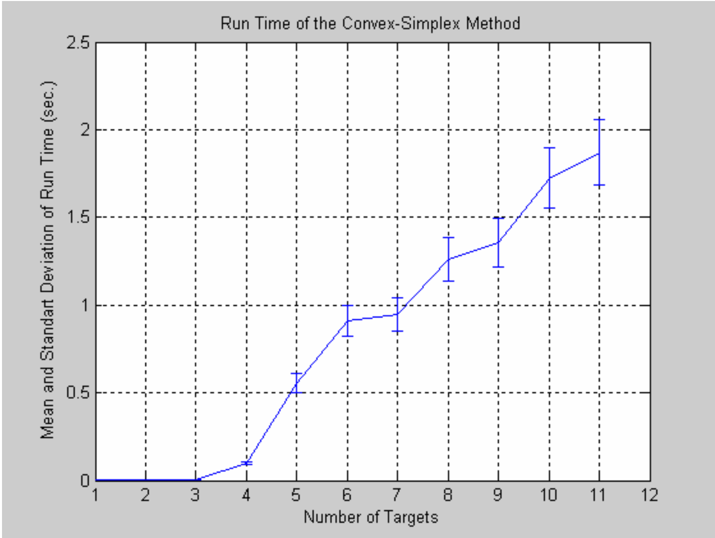

Figure 5.4.3 Run Time of Convex-Simplex Algorithm

The results suggest that the Convex-Simples algorithm is very advantageous over Topkis-Veinott's Modification and Gradient Projection Algorithms for the case of $N \leq 3$. This relatively better performance of Convex-Simplex Algorithm is not however observed for the case $N \geq 4$. Comparing Table 5.4.3 with Table 5.4.5, the Gradient Projection Algorithm appears more favorable as compared with the Convex-Simplex Algorithm for the scenarios with $N \geq 4$. The comparative plot in Figure 5.4.4 better illustrates the explained behavior.

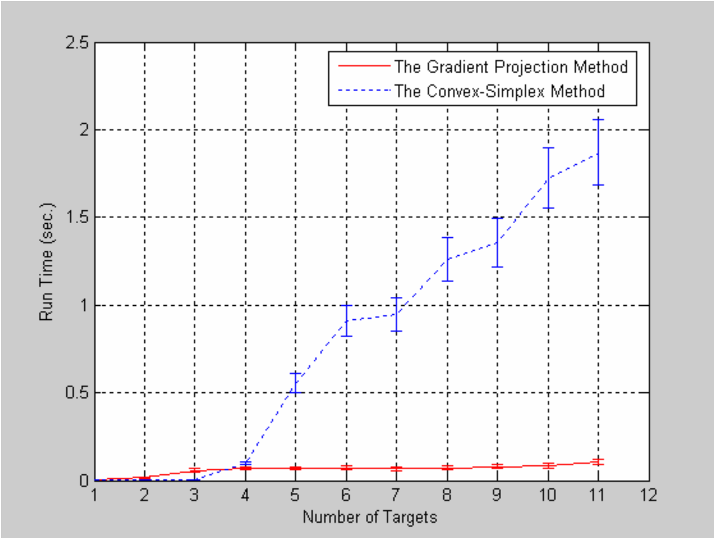

Figure 5.4.4 Comparison of Run Times of the Gradient Projection Algorithm and the Convex-Simplex Algorithm.

In the Table 5.4.6 below, average number of iterations and average run time per iteration values of the Convex-Simplex Algorithm is presented. Run time per iteration of Convex-Simplex Algorithm, which is increasing with increasing N as expected, is higher than that of the Gradient Projection Algorithm. Convex-Simplex Algorithm converges to the final operating point in one iteration and run time for the convergence of the algorithm is better than the Gradient Projection Algorithm when compared with results presented in Table 5.4.4 for the case of $N \leq 3$ as shown in the table below. But with increase of the number of tasks, number of iterations and computations at each iteration of the Convex-Simplex Algorithm becomes much higher.

Table 5.4.6 Number of Iterations of Convex-Simplex Algorithm

Number of Targets	Number of Iterations (Average)	Run Time Per Iteration (sec.)
1	1	0.00025
2	1	0.00016875
3	1	0.00015625
4	36.97	0.0026732
5	197.27	0.0028031
6	299.66	0.0030377

Table 5.4.6 Continued		
7	289.13	0.0032781
8	360.37	0.0035031
9	362.26	0.0037423
10	434.48	0.0039645
11	444.94	0.0042048

The Gradient Projection Method has better execution time among the other alternative algorithms considered in the Methods of Feasible Directions. In the Method of Zoutendijk with Topkis-Veinott's modification, a linear programming sub-problem is solved at each step in order to find the best improving feasible direction. However in the Gradient Projection Method, the improving feasible direction is found by projecting the negative gradient of the cost function onto the nullspace of the binding constraints. The evaluation of the direction vector is simpler in the Gradient Projection Method.

In the Convex-Simplex Method, after evaluation procedure of the reduced gradient vector, which is similar to the direction finding phase of the Gradient Projection Method, comparisons are performed on the scalar components of the nonbasic part of the reduced gradient vector in order to determine the nonbasic component of the direction vector that best improves the objective function. The basic component of the direction vector is computed from the nonbasic component by multiplying it with nonbasic and inverse of the basic components of the constraint matrix. Hence, there are more comparative computations in the evaluation of the direction vector phase of the Convex-Simplex Method than that of the Gradient-Projection Method.

After determination of the direction vector, the line search phase is same in the considered algorithms of the Methods of Feasible Directions. Therefore, the Gradient Projection Algorithm has better execution time when compared with the other alternatives in the Methods of Feasible Directions.

5.4.4 Results of the Q-RAM Based Approach

Simulation results of the Q-RAM Algorithm are provided in Table 5.4.7, Figure 5.4.5, Table 5.4.8 and Table 5.4.9. Mean run time of Q-RAM Algorithm increases with increase of N and it is comparable with mean run time of Gradient projection Algorithm as it can be observed from Table 5.4.3 and Table 5.4.7.

Table 5.4.7 Run Time of Q-RAM Algorithm

Number of Targets	Best Run Time Value (sec.)	Mean Run Time Value (sec.)	Worst Run Time Value (sec.)
1	0.0079688	0.0081563	0.0084375
2	0.015625	0.018304	0.046875
3	0.015625	0.028228	0.046875
4	0.015625	0.036756	0.0625
5	0.03125	0.046989	0.078125
6	0.046875	0.057084	0.078125
7	0.046875	0.065873	0.09375
8	0.046875	0.077195	0.40625
9	0.0625	0.089376	0.10938
10	0.078125	0.098183	0.125
11	0.09375	0.11176	0.14063

Figure 5.4.5 shows change of mean run time of Q-RAM algorithm for N increasing from 1 to 11. We verify that in agreement with the behavior specified in [6] run time of the Q-RAM Algorithm increases linearly with increase of tasks included in the simulation scenario. When compared with the simulation results of the Zoutendijk with Topkis-Veinott's Modification, Gradient Projection and Convex-Simplex Algorithms, Q-RAM Algorithm together with the Gradient Projection Algorithm seems favorable to the remaining two alternatives from the run time point of view.

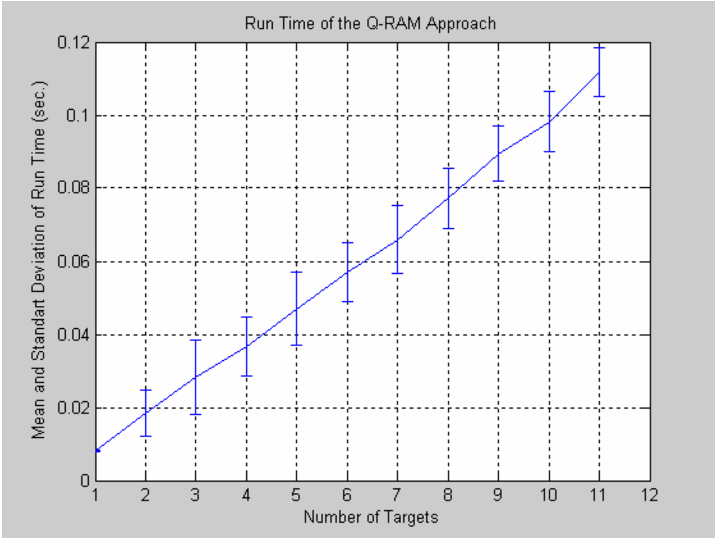

Figure 5.4.5 Run Time of Q-RAM Algorithm

Number of iterations and average run time of one iteration of Q-RAM Algorithm is shown in Table 5.4.8. Number of iterations decreases with increase of tasks as shown in the table. On the other hand, the table also shows the run time per iteration to increase. The former behavior can be explained by the execution of the algorithm. When the number of tasks in the scenario increases, unallocated resource amount that will be allocated to the tasks by using Q-RAM approach in order to obtain an optimal resource allocation will decrease with assignment of minimum resource requirements to the tasks. This leads to the distance between the current operating point and constraint boundary to decrease. This in turn leads to the decrease of the number of iterations of the Q-RAM Algorithm with increase of the number of tasks included in the simulation scenario.

Increase of N requires Q-RAM Algorithm to search more tasks for the best resource allocation in every iteration. Number of calculations at every iteration increases for this reason and run time per iteration increase with increase of N.

Table 5.4.8 Number of Iterations of Q-RAM Algorithm

Number of Targets	Number of Iterations (Average)	Run Time Per Iteration (sec.)
1	21	0.0003884
2	40	0.0004576
3	28.193	0.0010012
4	19.077	0.0019267
5	17.8	0.0026398
6	15.802	0.0036125
7	13.935	0.0047272
8	12.791	0.0060351
9	12.594	0.0070967
10	12.325	0.0079662
11	11.986	0.0093242

In the Table 5.4.9, the optimality percentage of the Q-RAM Algorithm is presented. As shown in the table, the distance of the result of the Q-RAM Algorithm to the global optimal resource allocation increases with increase of the number of tasks included in the scenario. This decrease is not significant however. As will be illustrated later, the loss of performance becomes much worse when the number of tasks becomes significantly large.

Table 5.4.9 Optimality Percentage of the Result of the Q-RAM Algorithm

Number of Targets	Tracking Quality Ratio (%)
1	100
2	100
3	99.939
4	99.513
5	99.129
6	98.772
7	98.516
8	98.342
9	98.265
10	98.219
11	98.183

The comparative plot presented in Figure 5.4.6 shows that the Q-RAM approach and the Gradient Projection Method have similar execution times. The Gradient Projection Method results in optimal solutions, however the Q-RAM approach has near-optimal results. Therefore, the Gradient Projection Method appears to be advantageous over Q-RAM in terms of closeness to optimal. However, when the results presented in Table 5.4.9 are observed, the distance of the results of the Q-RAM approach to the global optimum solution can be ignored for the case $N = 1$ to $N = 11$. In the next subsection, a detailed analysis for the cases, in which N is much more higher than 11, is presented.

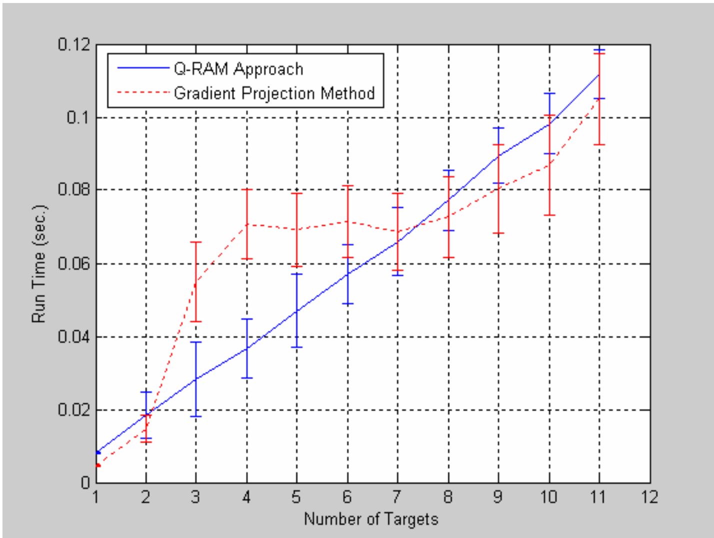

Figure 5.4.6 Comparison of Run Times of the Q-RAM Approach and the Gradient Projection Algorithm.

5.5 Comparative Simulations with Discrete Objective Functions

The Q-RAM algorithm can directly be applied in this case since it is inherently a discrete algorithm. However, Feasible Directions based algorithms are continuous in nature and their application in the discrete objective function case requires some adaptation to be incorporated. This adaptation is named in this study as FDRA-D. Due to its favorable performance, we have selected the Gradient Projection method from this group to be compared with the Q-RAM approach for the discrete objective function case. As discussed in more detail in subsection 4.5, this algorithm can be applied to the discrete case after finding the best fitting exponential curves ($1-m_k e^{-\alpha_k f_k - \beta_k P_k}$) to the discrete tracking quality functions of tracking tasks. The result of the continuous optimization thus obtained is discretized back by finding the discrete operating point closest to the continuous result. FDRA-D searches for the closest discrete value towards to the decreasing resource direction in order to guarantee the feasibility of the operating point. It should be noted that the described method forms a complete discrete algorithm where an inner loop is based on continuous optimization.

In these simulations, the tracking quality functions of tracking tasks are taken as discrete functions and for a simulation scenario containing N targets, there are 225^N discrete operating points to be searched for the optimal operating point. As it can be observed from Table 5.4.9 that the optimality percentage of the Q-RAM approach is in the neighborhood of % 98 - % 100 for the scenarios with N ranging from 1 to 11; the aim of the simulations performed in this subsection is to observe the performance of the Q-RAM approach when the number of targets in the scenario (N) is increased to numbers considerably larger than 11 and to see whether the FDRA-D can generate better results with the execution time comparable with that of the Q-RAM approach to the resource allocation problem whose objective function is discrete.

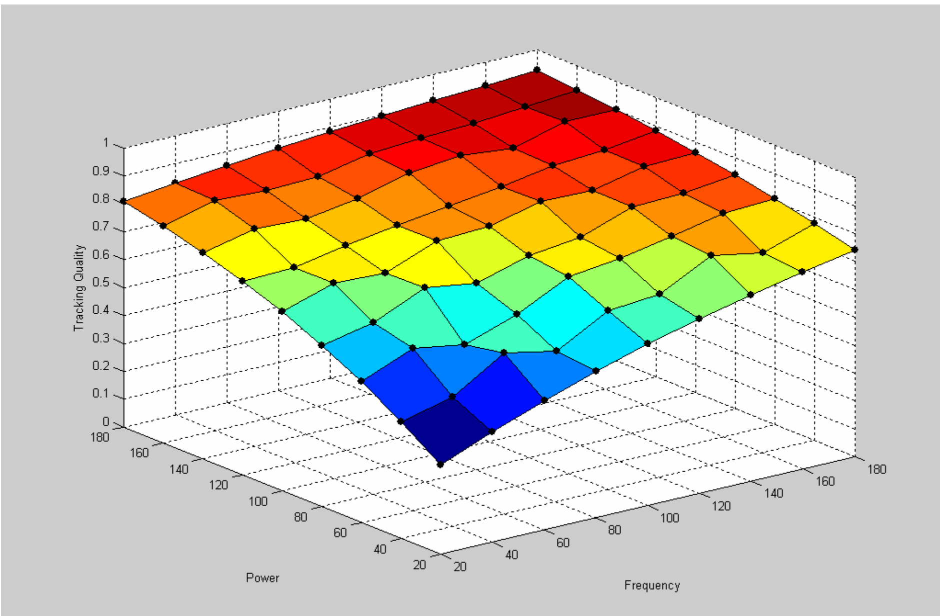

Figure 5.5.1 Generation of Discrete Tracking Quality Functions. For the simulations performed for comparative performance evaluation for the case of discrete objective functions, the input discrete functions are generated by

adding noise to the continuous functions. In this figure, noise is added to the function $1-m_k e^{-\alpha_k f_k - \beta_k P_k}$ for which $\alpha_k = 0.008$, $\beta_k = 0.006$ and $m_k = 0.9$.

In these simulations, discrete quality functions are first generated by sampling the continuous quality functions. On the discrete quality functions, quality differences of each point with their four neighbor points are evaluated separately. A random number is generated in the interval [0, 1] by using the *rand* function in MATLAB. The interval [0, 1] is divided into four equal intervals (i.e. [0, 0.25), [0.25, 0.5), [0.5, 0.75) and [0.75, 1]). Depending on the interval in which the generated number lies in, the quality value of the discrete point is increased or decreased by the quality difference between one of the four neighbor points multiplied by a re-generated random number with the function *rand*. In order to decide whether to increase or decrease the quality at the considered point an other random number is generated with *rand* function and if the generated number is greater than 0.5 the quality value is increased else the quality value is decreased.

In this subsection, Gradient Projection Algorithm is used in the Methods of Feasible Directions phase of the FDRA-D and the performance of the FDRA-D is compared with the Q-RAM on the radar resource allocation problem. The reason for using the Gradient Projection Algorithm can be seen from the simulation results presented in the previous subsection. In these results, it can be observed that the execution time of the Gradient Projection Algorithm is much better than the other two algorithms (Zoutendijk Algorithm with Topkis-Veinott's Modification and Convex-Simplex Algorithm) and comparable with the execution time of the Q-RAM approach.

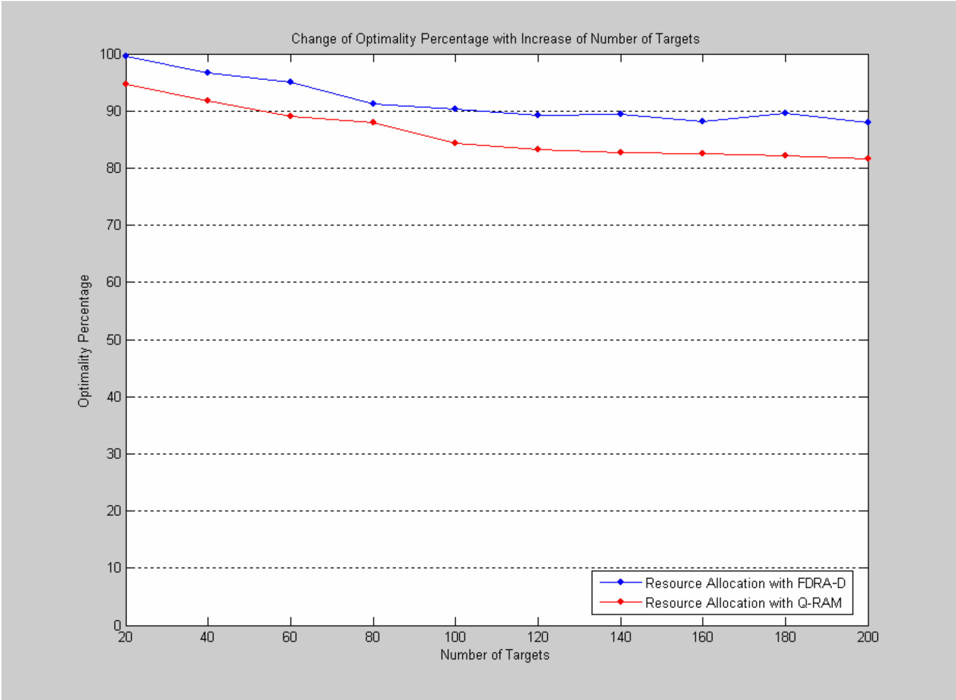

Figure 5.5.2 Change of Optimality Percentage with Increase of Number of Targets for Different Resource Allocation Approaches. Number of targets included in the simulation scenario is increased from 20 to 200 for this simulation. The results of the FDRA-D approach is closer to the optimal than the results of the Q-RAM approach.

As it can be observed from Figure 5.5.2 the results of the FDRA-D approach are closer to the optimal results than the results of the Q-RAM approach. Note that the discretized version of the feasible directions based methods (in this case the Gradient Projection method) is no longer globally optimal when compared to the continuous global optimum point. Unfortunately, it is not feasible, even with off-line exhaustive search to determine the exact global optimum point of the discrete problem as a reference, hence our use of the continuous global optimum (%100 point).

The table below presents the average run time of the FDRA-D and Q-RAM approaches with change of the number of tracking tasks included in the scenario. For the comparative simulations, a number of scenarios are considered. In all of these scenarios, the number of tracking tasks (*N*) ranges from 20 to 200 with an increment of 20. These tasks are chosen from a set of 400 tasks. Total combination of 10 different simulation cases are considered for different number of targets included in the scenario. Table 5.5.1 provides average of the run time values for 10 different simulation scenarios for each *N* value.

Table 5.5.1 Run Time of FDRA-D and Q-RAM Approaches. As the number of targets included in the scenario is increased, the run times of both of the approaches increase; run time of the FDRA-D approach is comparable with that of the Q-RAM approach.

Number of Tasks (N)	Run Time of FDRA-D (sec.)	Run Time of Q-RAM (sec.)
20	0.35822	0.30938
40	1.2689	1.0297
60	2.7755	2.5485
80	5.0018	4.7391
100	8.222	7.1219
120	9.2	9.9969
140	13.917	12.87
160	18.494	16.172
180	21.737	19.355
200	27.038	22.824

As it is explained in subsection 4.5, the result of the feasible direction method (scalar components of the operating point vector) is rounded to the nearest lower discrete operating point in order to guarantee the feasibility. Due to this rounding process the optimality percentage of the FDRA-D approach decreases with increase of the number of tracking tasks included in the simulation scenario as it is shown in Figure 5.5.2.

Root-mean-square (rms) of the noise added to the discrete function that is obtained by sampling the continuous function at discrete points is 0.017 for the simulations results shown in Figure 5.5.2 and Table 5.5.1. When rms of the noise added to the input function is increased the optimality percentage of the FDRA-D approach decreases as it is shown in Table 5.5.2. In Table 5.5.2, simulations results for 200 tracking task case are shown.

RMS of the Noise	Optimality Percentage (%)
0.0085	89
0.0171	87.9
0.0342	87.1
0.0685	85.2

Table 5.5.2 Variation of Optimality Percentage with Variation of RMS of Noise. In these simulations there exist 200 tasks in the simulation scenario. 10 different

simulations are performed for each different rms value and averages of these simulations are presented in this table.

When the simulation results presented in subsection 5.4 are observed, it can be concluded that when the tracking quality functions of the tracking tasks are modeled as continuous functions, the Methods of Feasible Directions can be used in order to solve the radar resource allocation problem, which can be formulated as a constrained optimization problem. The execution time of the Gradient Projection Algorithm is comparable with the Q-RAM approach, which was [6] proposed to be applied in the real-time applications. This implies that we are effectively proposing a theoretically well founded and optimal resource allocation approach to resource allocation problems with multiple resource type and continuous objective functions with results comparing favorably to those of the Q-RAM approach. The Gradient Projection Method, which have comparable execution time with the Q-RAM approach, appear to be a better alternative than the Q-RAM approach for resource allocation problems with multiple resource type and continuous objective functions.

For the resource allocation problems with multiple resource type and discrete objective functions, the presented study proposes to find best fitting continuous curve to the objective function and apply optimal continuous optimization techniques. For the radar resource allocation problem, in which f_k and P_k are investigated as optimization variables and the tracking quality functions of the tracking tasks are discrete functions, we have obtained best fitting exponential curves to the tracking quality functions and applied the Methods of Feasible Directions. The proposed FDRA-D approach provides results outperforming the Q-RAM approach to the radar resource allocation problem with comparable run times. This can be observed from Figure 5.5.2 and Table 5.5.1. Since the theoretical background of the FDRA-D approach is well founded and the approach is advantageous over the Q-RAM approach in terms of convergence and closeness to the global optimal solution, it is our belief that it is a better alternative than the Q-RAM approach for the resource allocation problems with multiple resource type and discrete objective functions.

5.6 Summary

In this chapter, the performances of the Methods of Feasible Directions and the Q-RAM approach are evaluated and compared on the radar resource allocation problem. The sampling frequency (f_k) and average power of the transmitted radar signal (P_k) are investigated as optimization variables. In subsection 5.4, the case with continuous objective functions is considered and the objective functions formulated in subsection 2.2 are used. Subsection 2.2 also discusses that the exponential tracking quality functions can be obtained for different tracking quality tasks with different speed, distance to the radar system and maneuverability properties in practical applications. Hence such exponential functions are utilized in our simulation experiments. In these simulations, the Q-RAM approach is applied based on discrete objective functions obtained by sampling the continuous objective functions on discrete operating points. Our simulation results show that the Gradient Projection Method proves to be advantageous over the Q-RAM approach in terms of closeness to optimal with comparable execution times with the Q-RAM approach.

As it is mentioned in subsection 2.2, the tracking quality functions for different tracking tasks are also considered as discrete functions and Q-RAM is proposed to be applied to the radar resource allocation problem [16]. In the present study, an alternative resource allocation approach to the Q-RAM called FDRA-D is proposed to be applied to the radar resource allocation problem for discrete case of the resource allocation problem. The simulation results of subsection 5.5 show that the proposed FDRA-D approach is advantageous over the Q-RAM approach in terms of closeness to optimal while maintaining comparable speed. This approach is also solidly founded on optimization theory as opposed to the Q-RAM approach.

CHAPTER 6

CONCLUSION AND FUTURE WORK

In this thesis, the resource allocation problem in real-time systems is investigated and a phased array radar system is considered as an illustrative area in order to comparatively evaluate the resource allocation approaches. A detailed investigation of a recently proposed resource allocation approach called Q-RAM is presented in two different cases: single resource type case and multiple resource type case. For the multiple resource type case, we propose to apply the Methods of Feasible Directions to the radar resource allocation problem. The performances of both the Q-RAM and the Methods of Feasible Directions based approaches are investigated in terms of optimality and convergence speed with the help of Monte-Carlo simulations. In the following subsections, first, the contributions of the thesis are outlined and second, the future work that can improve the present study is presented.

6.1 Contributions

The contributions of the present study can be outlined as follows:

- The Q-RAM approach, when applied to the radar resource allocation problem, is evaluated and shortcomings of the approach are identified.
- The Q-RAM approach to the radar resource allocation problem with single resource type is extended to give a global optimal solution.
- Algorithms from the well established Methods of Feasible Directions are proposed and applied to the radar resource allocation problem with multiple resource type with promising results.
- A comparative evaluation of algorithms investigated for the cases with continuous objective function and discrete objective, is presented.

The following paragraphs will briefly elaborate on these contributions:

For the case of single resource type, the Q-RAM algorithmic approach is improved in order to generate optimal results in all of the possible termination cases when the radar resource allocation problem formulated in subsection 3.3.1 is considered. In this solution approach, since the objective function in the minimization problem is twice differentiable and convex together with the convex constraints, the KKT optimality conditions are proposed to be satisfied completely as a result of the algorithm. As it is shown in the simulation results presented in 3.3.2, the proposed optimal resource allocation have minimal execution times, hence is still suitable for real-time applications.

As it is explained in subsection 3.4, the goal of the Q-RAM based approaches for the multiple resource type case is to reach a solution, which is closest to the optimal solution, in real-time systems. The emphasis is on a fast approximate solution. For this case, the Q-RAM based solution approach is a near-optimal optimization approach and does not provide a well founded mathematical background. With the motivation of using a theoretically well founded method, we propose algorithms based on Methods of Feasible Directions. To the best of our knowledge, we present the first application of this family of methods to the radar resource allocation problem. The results obtained reveal, in particular for the Gradient Projection Method that globally optimum solutions are possible with comparable computational speed. In order to overcome the practical difficulties encountered in determination of termination of the algorithms of the Methods of Feasible Directions in practical implementations, the norm values of $\mathbf{N}_k$ is used in this thesis as it is explained in the subsection 5.3.2.2.

It is proved in [4]; the Methods of Feasible Directions generate optimal results to the minimization problems with twice differentiable and convex objective functions and convex constraints. The objective function in the radar resource allocation problem formulated in subsection 2.3 has twice differentiable and convex objective function and convex constraints when it is re-formulated in the form of minimization problem. Therefore, the results generated by the Methods of Feasible Directions to the radar resource allocation problem are optimal. As it can be observed from the simulation results presented in subsection 5.4, the convergence speed of the Gradient Projection Algorithm is comparable with that of the Q-RAM approach. Hence, an optimal solution, which is as fast as the Q-RAM approach and has a comparable mathematical background, for the radar resource allocation problem in this thesis.

As it is explained in Chapter 3, the Q-RAM approach is applied on the discrete objective functions for the resource allocation problem with multiple resource type. For the comparative simulations whose results are presented in subsection 5.4, discrete samples are taken on the continuous objective functions in order to generate simulation inputs to the Q-RAM approach. In this thesis, an approach called FDRA-D is proposed in order to obtain an improved method over the Q-RAM approach executing directly on the discrete objective functions for the resource allocation problem. In this approach, it is proposed to fit exponential curves to the discrete tracking quality functions of the tracking tasks for the radar resource allocation problem and applying one of the Methods of Feasible Directions on the continuous exponential curves. The resultant operating point, derived from the algorithm, drops into a range defined within discrete operating points. The nearest point, which is the lowest discrete operating point within the range, is selected. As it can be observed from the simulation results presented in subsection 5.5, the FDRA-D approach generates favorable results than the Q-RAM approach with execution times comparable with the Q-RAM.

6.2 Future Work

In this thesis, sampling frequency (f_k) is investigated as computational resource while average power of the transmitted radar signal (P_k) is investigated as the energy resource of the radar system. By considering these parameters as optimization variables, an optimization problem with convex objective function and convex constraints can be formulated. When the computation time (C_k) of the tracking algorithms is also considered as an optimization parameter along with the sampling frequency and average power of the transmitted radar signal, the constraints of the optimization problem become non-convex. In this case, solving the formulated optimization problem with the Methods of Feasible Directions does not provide sufficient conditions for the global optimality of the results. As an ongoing work, it is proposed to solve the radar resource allocation problem by considering also the computation time as an optimization variable in order to obtain global optimum results with maximal convergence speed.

After determination of the tracking parameters for the tracking tasks, the tracking tasks should be scheduled in the radar processor. In dense tactical environments in which all of the targets in the environment can not be tracked simultaneously by the radar system, the determination of which tracking tasks to drop from the task list and how to schedule the tasks in the radar processor is important. In the future studies, it is proposed to consider the problem of optimal scheduling of the radar tasks in the radar processor.

In the present study, resource allocation in one radar system is considered. Resource allocation in a system containing multiple radar systems tracking a set of targets is planned to be considered in the future studies. In this system, each target is tracked by exactly one radar system and responsibility for targets from one radar system can be transferred to another as the targets move. A resource manager allocates resources to the tasks of the radar systems in order to achieve a globally optimal tracking quality.

REFERENCES

[1] Willard I. Zangwill, “The Convex-Simplex Method”, Management Science, Vol. 14, No: 3, 1967.

[2] David G. Luenberger, “Introduction to Linear and Nonlinear Programming”, Addison-Wesley, 1973.

[3] Papoulis, A., “Probability, Random Variables, and Stochastic Processes”, Third Edition, McGraw Hill Inc., New York, 1991.

[4] M.S. Bazaraa, H.D. Sherali, C.M. Shetty, “Nonlinear Programming: Theory and Algorithms”, 2nd. ed., Wiley, New York, 1993.

[5] Danbing Seto, John P. Lehoczky, Lui Sha, and Kang G. Shin, “On Task Schedulability in Real-Time Control Systems”, In Proceedings of the 17th Real-Time Systems Symposium, 1996.

[6] Ragunathan Rajkumar, Chen Lee, John Lehoczky, Dan Siewiorek, “A Resource Allocation Model for QoS Management”, 18th IEEE Real-Time Systems Symposium, 1997.

[7] Chen Lee, John Lehoczky, Ragunathan (Raj) Rajkumar, Dan Siewiorek, “On Quality of Service Optimization with Discrete QoS Options”, 19th IEEE Real-time Technology and Applications Symposium, 1998.

[8] Ragunathan (Raj) Rajkumar, Chen Lee, John P. Lehoczky, Daniel P. Siewiorek, “Practical Solutions for QoS-based Resource Allocation Problems”, 19th IEEE Real-Time Systems Symposium, 1998.

[9] S. Blackman and R. Popoli, “Design and Analysis of Modern Tracking Systems”, Artech House, Nordwood, 1999.

[10] Chen Lee, John Lehoczky, Dan Siewiorek, Ragunathan Rajkumar and Jeff Hansen, “A Scalable Solution to the Multi-Resource QoS Problem”, 20th Real-Time Systems Symposium, May 1999.

[11] Caccamo, M., Buttazzo, G., and Sha, L., “Elastic feedback control”, In Proceedings of the 12th Euromicro Conference on Real-Time Systems, 2000.

[12] Hakan Aydin, Rami Melhem, Daniel Mossé, and Pedro Mejia-Alvarez, "Optimal Reward-Based Scheduling for Periodic Real-Time Tasks", IEEE Transactions on Computers, Vol. 50, No. 2, February 2001.

[13] Merrill I. Skolnik, "Introduction to Radar Systems", 3rd. ed., McGraw-Hill, 2001.

[14] Shirley L. Coetzee, Karl Woodbridge, Christopher .I. Baker, "Multifunction Radar Resource Management using Tracking Optimisation", Proceedings of the International Radar Conference, 2003.

[15] Young-Hun Jung, Sun-Mog Hong, "Modeling and Parameter Optimization of Agile Beam Radar Tracking", IEEE Trans. on Aerospace and Electronic Systems, vol. 38, no. 1, January 2003.

[16] Chang-Gun Lee, Chi-Sheng Shih, Lui Sha, "Online QoS Optimization Using Service Classes in Surveillance Radar Systems", Publication in Journal of Real-Time Systems, 2003.

[17] Johannes Wintenby, "Resource Allocation in Airborne Surveillance Radar", Thesis for the degree of Doctor of Philosophy, Department of Signals and Systems, School of Electrical Engineering, Chalmers University of Technology, Göteborg, Sweden, 2003.

[18] Sourav Ghosh, "Scalable QoS-based Resource Allocation", A Dissertation Submitted to the Graduate Education Committee at the Department of Electrical and Computer Engineering Carnegie Mellon University in Partial Fulfillment of the Requirements for the degree of Doctor of Philosophy in Electrical and Computer Engineering, 2004.

[19] Sourav Ghosh, Jeffery Hansen, Ragunathan (Raj) Rajkumar and John Lehoczky, "Adaptive QoS Optimizations with applications to Radar Tracking", RTCSA - Real-Time and Embedded Computing Systems and Applications Conference, 2004.

[20] Ayhan Ircı, Buyurman Baykal, Renan Mert Özel, "Resource Management in a Multifunction Radar System", SİU IEEE 13th Signal Processing and Communication Applications Conference, Kayseri, 2005.

[21] Nan Jin, Gayathri Venkitachalam, and Scott Jordan, "Dynamic Congestion-Based Pricing of Bandwidth and Buffer", IEEE/ACM Transactions on Networking, Vol. 13, No. 6, December 2005.

[22] Jeffery Hansen, Ragunathan (Raj) Rajkumar, John Lehoczky, Sourav Ghosh, "Resource Management for Radar Tracking", IEEE International Radar Conference, Verona NY, April 2006.

[23] Ayhan Ircı, Afşar Saranlı, Buyurman Baykal, "A Real-Time Optimal Resource Allocation Approach in a Radar System", SİU IEEE 14th Signal Processing and Communication Applications Conference, Antalya, 2006.

[24] Ayhan Ircı, Afşar Saranlı and Buyurman Baykal, "On Optimal Resource Allocation in Multifunction Radar Systems", IEEE International Radar Conference, Verona NY, April 2006.

[25] http://mathworld.wolfram.com/LeastSquaresFittingExponential.html

[26] Marcel Lüthi, Simin Nadjm-Tehrani, Calin Curescu, "Comparative Study of Price-based Resource Allocation Algorithms for Ad Hoc Networks", 20th International Parallel and Distributed Processing Symposium, 2006.

APPENDIX

A.1 Convexity

A set S in E_n is said to be *convex* if, for each $\mathbf{x}_1, \mathbf{x}_2 \in S$, the *line segment* $\lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2$ for $\lambda \in [0, 1]$ belongs to S.

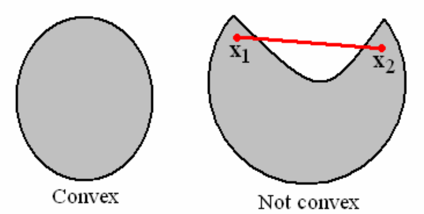

Figure A.1.1 Illustration of convexity

Let S be nonempty convex set in E_n. The function $f : S \rightarrow E_1$ is said to be *convex* on S if

$$f[\lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2] \leq \lambda f(\mathbf{x}_1) + (1-\lambda)f(\mathbf{x}_2)$$

for each $\mathbf{x}_1, \mathbf{x}_2 \in S$ and for each $\lambda \in [0, 1]$. The function f is said to be *concave* if $-f$ is convex.

A.2 KKT Optimality Conditions

Consider the following problem:

Problem P:	Minimize	$f(\mathbf{x})$	
	subject to	$g_i(\mathbf{x}) \leq 0$	for $i = 1, \ldots, m$
		$h_i(\mathrm{x}) = 0$	for $i = 1, \ldots, l$
		$\mathrm{x} \in X$	

where $f, g_i, h_i : E_n \rightarrow E_1$ and X is a nonempty open set in E_n. The KKT necessary optimality conditions are as follows. If x is a local optimum solution to Problem P, and under a suitable constraint qualification, there exists a vector $(\mathbf{u}, \mathbf{v})$ such that

$$\nabla f(\mathbf{x}) + \sum_{i=1}^{m} u_i \nabla g_i(\mathbf{x}) + \sum_{i=1}^{l} v_i \nabla h_i(\mathbf{x}) = \mathbf{0}$$
$$u_i g_i(\mathbf{x}) = 0 \qquad \text{for } i = 1, ..., m$$
$$u_i \geq 0 \qquad \text{for } i = 1, ..., m$$

u_i and v_i are the *Lagrange multipliers* associated with the constraints $g_i(\mathbf{x}) \leq 0$ and $h_i(\mathbf{x}) = 0$, respectively. When objective function and constraints of the minimization problem are convex, the KKT conditions are *sufficient* for optimality.

A.3 Relation between Probability of Detection and Transmission Power

The probability of detection equation in radar systems is provided below [13]:

$$S/N = A + 0.12AB + 1.7B \tag{A.3.1}$$

In this equation, S is the power of transmitted radar signal and N is noise power and

$$A = \ln(0.62/P_{fa}) \tag{A.3.2}$$

$$B = \ln\left[P_d/(1-P_d)\right] \tag{A.3.3}$$

In the equations A.3.1 and A.3.2, P_d is the probability of detection and P_{fa} is probability of false alarm. When the probability of detection equation is re-arranged, it can be written in the following form.

$$P_d = 1 - \frac{1}{K_1 e^{K_2 S} + 1} \tag{A.3.4}$$

K_1 and K_2 in the above equation are provided in the below equations,

$$K_1 = e^{-\frac{\ln(\frac{0.62}{P_{fa}})}{0.12\ln(\frac{0.62}{P_{fa}})+1.7}} \tag{A.3.5}$$

$$K_2 = \frac{\frac{1}{N}}{0.12\ln(\frac{0.62}{P_{fa}}) + 1.7} \qquad \text{(A.3.6)}$$

When the probability of false alarm and noise power are selected in the shaded region of the figure shown below, the probability of detection can be written as an exponential function of transmission power from equation A.3.1. In this region, 1 can be ignored in the denominator of the second term and probability of detection can be written as in A.3.6.

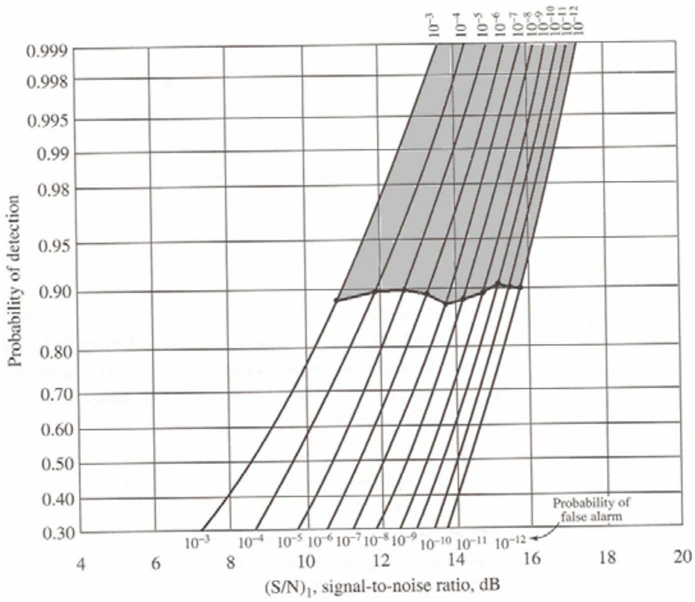

Figure A.3.1 Probability of detection for a sinewave in noise as a function of the signal-to-noise (power) ratio and the probability of false alarm [13]