A

Dissertation Report

On

Development of Multi-Agent System for Air Runway Sequencing and Scheduling Problem

Dissertation submitted in fulfillment of the requirements for the Degree of

MASTER OF TECHNOLOGY

by

VIRENDRA KUMAR VERMA

(2013PIE5327)

Supervisor

Dr. M. L. MITTAL

DEPARTMENT OF MECHANICAL ENGINEERING

MALAVIYA NATIONAL INSTITUTE OF TECHNOLOGY, JAIPUR

June, 2015

MALAVIYA NATIONAL INSTITUTE OF TECHNOLOGY JAIPUR

JAIPUR – 302017 (RAJASTHAN), INDIA

CERTIFICATE

This is to certify that the dissertation entitled **"Development of Multi-Agent System for Air Runway Sequencing and Scheduling Problem"** being submitted by **Virendra Kumar Verma (2013PIE5327)** is a bonafide work carried out by him under my supervision and guidance, and hence approved for submission to the **Department of Mechanical Engineering, Malaviya National Institute of Technology Jaipur** in partial fulfillment of the requirements for the award of the degree of **Master of Technology (M.Tech) in Industrial Engineering**. The matter embodied in this dissertation report has not been submitted anywhere else for award of any other degree or diploma.

Dr. M. L. Mittal
Associate Professor,
Department of Mechanical Engineering,
MNIT Jaipur

Place: Jaipur
Dated: 14 June 2015

MALAVIYA NATIONAL INSTITUTE OF TECHNOLOGY JAIPUR

JAIPUR – 302017 (RAJASTHAN), INDIA

CANDIDATE'S DECLARATION

I hereby declare that the work which is being presented in this dissertation entitled **"Development of Multi-Agent System for Air Runway Sequencing and Scheduling Problem"** in partial fulfillment of the requirements for the award of the degree of **Master of Technology (M.Tech)** in **Industrial Engineering**, and submitted to the **Department of Mechanical Engineering, Malaviya National Institute of Technology Jaipur** is an authentic record of my own work carried out by me during a period of one year from July 2014 to June 2015 under the guidance and supervision of **Dr. M. L. Mittal** of the Department of Mechanical Engineering, Malaviya National Institute of Technology Jaipur.

The matter presented in this dissertation embodies the results of my own work and has not been submitted anywhere else for award of any other degree or diploma.

Virendra Kumar Verma

(2013PIE5327)

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Dr. M. L. Mittal

Supervisor

Place: Jaipur

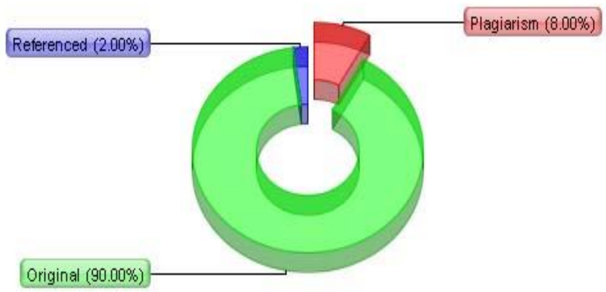
Dated: 14 June 2015

PLAGIARISM DETECTION CHART

Analyzed document:

"Virendra_2013PIE5327.docx"

Relation chart:



Core version:	874
Size:	121253 words
Registered to:	Originality report generated by unregistered Demo version!
Generated:	6/25/2015 12:20:59 AM
License type:	Plagiarism Detector

ACKNOWLEDGEMENT

With great delight, I acknowledge my indebted thanks to my guide and mentor **Dr. M. L. Mittal** who has always been a source of inspiration and encouragement for me. His stimulated guidance and unwavering support always motivated me to reach out for, and achieve higher levels of excellence. This dissertation could not have attained its present form, both in content and presentation without his active interest, direction and help. I am grateful to him for keeping trust in me in all circumstances. I thank him for being big-hearted with any amateurish mistakes of mine.

I extend my deep sense of gratitude to **Prof. I. K. Bhatt,** Director, MNIT Jaipur for strengthening the research environment of institute by providing all necessary facilities to the students.

I express my sincere gratitude to **Prof. G. S. Dangayach** (HOD) and **Prof. Rakesh Jain** (Ex. HOD) for their cooperation in work and help throughout the course of study.

I express my sincere gratitude to **Prof. Awadhesh Bhardwaj** (PG Coordinator IE)**,** and **Dr. Gunjan Soni** (Assistant Professor ME Deptt.) for their support and guidance throughout the course of study at MNIT Jaipur.

I am extremely thankful to **Mr. Andy Park,** Sector Manager - Aerospace & Defence, AOS Group for providing me the JACK^{TM} agent Platform software for completing the research work. I am also thankful to **Matteo Pedrotti,** Software Developer, AOS Group for helping me in the installation of software.

I sincerely thank **Vahit Kaplanoglu,** Assistant Chair, University of Gaziantep, Department of Industrial Engineering**,** who, despite of their busy schedule, cheerfully responded to my queries and made this research task possible.

I am indebted to my parents **Ram Shankar Verma** and **Sanvari Devi** for their hearty support, and patience in accomplishing this work. I am also thankful to my elder brother, sister and all family member and relatives from the bottom of my heart, for their prayers and confidence in me.

I highly acknowledge and duly appreciate the support extended by my seniors, colleagues-friends and juniors for their help & support in accomplishment of this work during my stay at MNIT Jaipur.

Virendra Kumar Verma

Abstract

Due to an anticipated increase in air traffic during the next decade, air traffic control in busy airports is one of the main challenging tasks to the controllers in the near future. Since the runway is often a bottleneck in an airport system, there is great interest in optimizing usage of the runway by applying the algorithms and various techniques. Sequencing and scheduling of aircraft landing at airports is a major issue in daily air traffic control (ATC) operations. Airports, especially busy airports, proved to be the bottleneck resources in the air traffic control system. The effective and efficient arrival scheduling and sequencing is one of main concerns to improve the safety, capacity, and efficiency of the airports.

Arrival runways are a critical resource in the air traffic system. Arrival delays have a great influence on airline operations and cost. Therefore, intentional arrival planning is becoming increasingly important. Air runway sequencing and scheduling is a well-known and important problem for optimizing daily air traffic control operations. A large variety of solution techniques such as dynamic programming, branch and bound algorithms and meta-heuristics have been applied to solve the problem. The existing methods, however, have significant drawback. One of the major drawbacks is their incapability to handle dynamic situation. This demands the development of intelligent scheduling techniques, which can handle the dynamic situation. Multi-agent System (MAS), a branch of artificial intelligence, has been applied to the air runway sequencing and scheduling problem.

MAS have gained a tremendous popularity in the field of manufacturing system for machine scheduling but in the air landing field it’s limited. In this work we developed a Multi-agent System (MAS) for air runway sequencing and scheduling that handles the dynamic situations. The algorithm is based on the negotiation/ bidding mechanism. The sequencing procedure is a bidding protocol where the arrival airport manager first queries all runway managers for their respective "best landing time" for an aircraft, and thereafter it chooses one. The "best landing time" for a runway allows faster aircraft, that arrive later to the airport control area, to "push out" earlier, already assigned slower aircraft. In those cases, the slower aircraft will be re-assigned when the "push out" occurs. In this thesis we try to optimize the runway throughput subject to the various real life constraints such as separation between landing of aircrafts, respecting the time window for landing. The multi-agent system framework is developed on BDI (belief, desire and intension) architecture based java agent platform software JACK™.

Table of Contents

List of Figures

List of Table

Abbreviations

ALP- Air Landing Problem

MAS- Multi-Agent System

IATA- The International Air Transport Association

ATCs- Air Traffic Controllers

ATM- Air Traffic Management

DP- Dynamic Programming

B & B- Branch and Bound

GA- Genetic Algorithms

SA- Simulated Annealing

ACO- Ant Colony Optimization

ALS- Air Landing Scheduling

FCFS- First Come First Serve

ICAO- International Civil Aviation Organization

KQML- Knowledge Query and Manipulation Language

KIF- Knowledge Interchange Format

CPS- Constraint Position Shifting

FIPA - Foundation for Intelligent Physical Agents

ACL- Agent Communication Language

ETA- Expected Time for Arrival

ALT- Actual Landing Time

ELT- Estimated Landing Time

CNP- Contract Net Protocol

BDI- Belief Desire Intention

Chapter 1: Introduction

1.1 Background

Over the past few decades, air traffic has experienced a tremendous growth. The en-route traffic capacity improvement has increased significantly. As a result, the air traffic congestion is shifting from the en-route segment to the airports. The throughout world traffic is increasing significantly it is well known. Due to increase in traffic there is an increasing demand for decision support tool to make the efficient use of the limited capacity (runway, airspace, taxi route etc.). Airport arrival and departure management is a very complex problem that plays a crucial role for airports. The number of runway and air-traffic controller capacity are limited in the airport, the traffic has to be planned carefully to optimizing the limited resources.

According to projections, air transportation demand is expected to grow annually at high rates. The International Air Transport Association (IATA) announced global passenger traffic results for the full year of 2014 showing demand (revenue passenger kilometers or RPKs) rose 5.9% compared to the full year of 2013. This 2014 performance was above the 10-year average growth rate of 5.6% and the 5.2% annual growth experienced in 2013 compared to 2012. Due to this increase of traffic demand causes congestion in the airport control area, holding area and taxi route. This congestion level causes a great challenges to the traffic controller to accommodating the flights. Airport runway capacity is often a limiting factor when creating plans to offer additional flights at an airport. Although airport capacity can be increased by building a new runway, making the best usage of the existing runway(s) through careful scheduling may reduce the need to improve the infrastructure.

In addition to issues of safety, which is the prime responsibility of air traffic controllers (ATCs), there are other stakeholders with an interest in how aircraft landings and take-offs are scheduled. Punctuality is a priority for airlines and airports. Airport operations such as gate assignment and baggage handling require careful planning in advance, and delays to an aircraft landing and take-off may have a detrimental effect on similar operations for the subsequent aircraft. Airlines also prefer schedules that minimize the cost of fuel, and governments typically have targets for reducing CO_2 emissions. Long queues and additional manoeuvres by aircraft to create a landing and take-off sequence may increase emissions. ATCs organize the landing and take-off of aircraft to meet safety requirements and maximize throughput. Ideally, the aims of all of the various stakeholders would also be taken into account when scheduling the landings and take-off of aircraft.

Today, Air Traffic Management (ATM) is highly concerned about traffic optimization at the airport and in terminal manoeuvring areas for economic, environmental and capacity reasons. In this situation, air traffic controllers have to meet various challenges such as: avoiding long air and ground queues; considering the best usage of available airspace, runways, taxiways and gates; taking into account fuel efficiency; reducing noise disturbance and environmental impact; minimizing delays; and accounting for safety issues.

1.2 Air Runway Sequencing and Scheduling

Sequencing and scheduling is the prime concern to increase the runway capacity. Considering its importance a significant amount of research has been devoted to a wide variety of air landing problems, but the heuristics have attracted maximum attention of researcher for optimization. Various other methods are available for optimization which provides the very good results at short time interval like meta-heuristics but the application of these methods is limited in the field of ALP. Separation between the aircraft is necessity of ATC to avoid the vortex for safety considerations. Wake vortices are rotating masses of air that are generated by aircraft as a consequence of their lift. Without sufficient separation, wake vortices provide a hazard for the following aircraft.

"Air runway scheduling is the process of assigning runway and time slot to the aircraft." The optimization process by which limited resources is done over the sequential and parallel activities. With air traffic globalization, the optimization process is highly important for air traffic management to increase the capacity of runway.

Considering the complexity of airport runway scheduling, it is difficult to find the optimal solution to the problem in most cases. Thus, it draws significant attention from different scientific communities with various research studies carried out on modeling and developing algorithms to increase capacity at an airport. The best sequencing and scheduling reduces the separation times between the aircrafts and improve the efficiency of the airports.

Solving air landing problem (ALP) can be accomplished by exact procedures, meta-heuristics, or simple heuristics. For a real world problem with a large number of aircraft, optimizing the runway and time slot using exact procedure such as dynamic programming, integer programming are computationally difficult, time consuming and impractical. Instead companies mostly use simple heuristics to solve the problem.

High levels of uncertainty are presents in the air landing problem so several issues need to consider for effective planning and scheduling of ALP. Some of them are shown in Figure 1.1. One of the main challenging issues of researchers in this field is to allocate the time window and runway that call for resolving the conflicts that arises in air traffic controller arriving many aircraft at a time in the dynamic environment. This demands the development of intelligent planning and scheduling techniques, which are able to take decision, flexible, efficient and reliable solution qualities in software development for such types of system. In recent years Multi-agent System

(MAS), a branch of artificial intelligence have gained tremendous popularity in providing solution to the distributed decision making problems in different domain of industrial life.

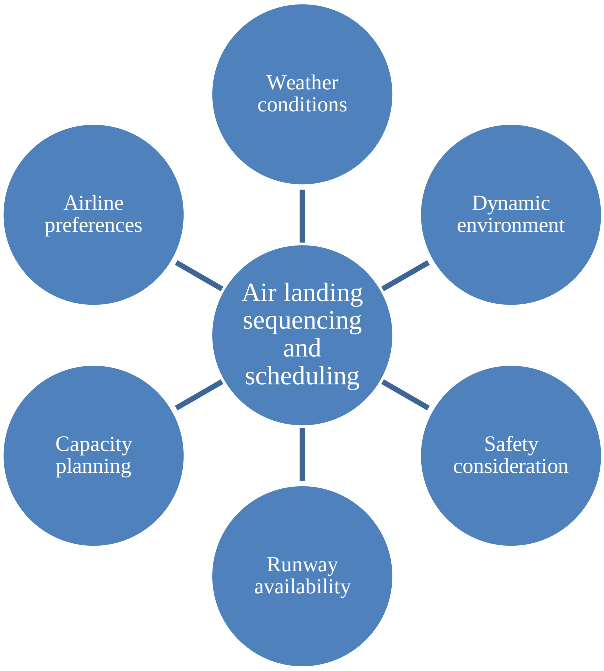


Figure 1.1: Major issues in air runway sequencing and scheduling Problem

It is a distributed system consisting of a set of self-interested, interacting problem solving entities, called agents (Michael Wooldridge & Jennings, 1995).

1.3 Practical Motivation of MAS Air Runway Sequencing and Scheduling Problem

A comprehensive survey reveals that all algorithm for determining the optimal solution as well as heuristics approaches based on centralized planning and control. A centralized approach is not suitable in real life situations. In centralized approaches all the information relevant to the problem is passed through the project manager and makes a centralized decision for all the plans and schedules in order to optimize the single objectives. If a small changes occurs in the system due to dynamic nature than it requires to change the whole system. In air traffic management every party have its own objectives like airlines wants to maximize profit, government wants to ensure safety of passenger, airport wants to maximize the throughput etc. so centralized approaches is not sufficient to handle all these objectives.

For solving air landing problem, only few methods are available in the literature. However most of the researchers totally ignore the uncertainty and dynamic nature. Researcher takes the consideration of only theoretical constraints to solve the problem but they not concern about the practical constraints. Some researcher compare the machine scheduling problem to the air landing problem, in the machine scheduling recently MAS gain tremendous growth but in air landing problem research still limited up to heuristics and meta-heuristics. Since scheduling in real life tends to the dynamic, and having a lots of uncertainty in the system like runway unavailability, safety issues, weather condition etc. are rarely consider in the existing state of art algorithms. Owning to these gaps, there is an urgent need to develop a new solution approaches that are free from these restrictive assumptions and provide better results. The brief outline of this research as given below:

The review of literature presents that until now very little work done towards application of MAS to the air runway sequencing and scheduling. The currently available approaches are not suitable in real world application due to either the time consuming or static nature. The present research considers the air landing situation where the arrival of aircraft is dynamic in nature. This study proposes a multi-agent based system for air runway sequencing and scheduling for maximize the throughput. The MAS are developed on BDI (belief, desire and intention) architecture. To support this BDI architecture, JACK^{TM} a java based agent platform is used in the research. A negotiation protocols are used in the system for agent bidding.

1.4 Research Objectives and Scope of Present Work

The proposed research work aims at developing a Multi-agent System (MAS) for air runway sequencing and scheduling. In order to fulfill the research objectives, the proposed approach divide the work into following different stages outlined as below:

- Development of a MAS framework for air runway sequencing and scheduling problem.
- To develop a negotiation mechanism among the agents for optimizing runway sequencing and scheduling problem.

The scope of present work is limited to

- Number of runways is limited to only one i.e. single runway scheduling problem is considered.
- Only landing of aircrafts is considered.

1.5 Organization of the Thesis

The thesis organized into six chapters. A brief description about the content in each chapter as provided below.

Chapter one, the current chapter, provides the brief overview of background of air transportation, air traffic management, air runway sequencing and scheduling, application of MAS in air runway scheduling, practical motivation, and objective and scope of present work. The need and application of multi-agent system (MAS) are introduced.

Chapter two, review the literature review on the air landing problem and application of Multi-agent system (MAS) in the scheduling field. The air landing problem classified based on the algorithm and methods used for optimization, like dynamic programming, branch and bound, heuristics, genetic algorithm, simulated annealing and ant colony optimization. The applications of Multi-agent System (MAS) review are generally classified in the machine scheduling problem in manufacturing and air traffic management.

Chapter three provides the basic concepts related to the air landing problem. The general term associated with air runway sequencing and scheduling problem are briefly described in this chapter.

Chapter four presents some issues and complexity of the air landing problem. The basics of Multi-agent System (MAS), MAS architecture and the interaction protocol are discussed in this chapter. The overview approach of the Multi-agent System (MAS) is design and types of agents are discussed. The scope of this chapter is limited to the MAS overview and design of approach diagram of agents.

Chapter five, deals with the multi-agent system for air runway sequencing and scheduling problem. The multi-agent architecture designed and developed using BDI (belief, desire and intention) architecture. In this define the air landing problem and mathematical modeling of the problem. A MAS is designed and simulated using JACK^{TM} agent development software. A new negotiation protocols between the agents is proposed.

Chapter six is the concluding chapter that summarizes the present research. Significant contributions and limitations of the research also highlight including the suggestion for the future research.

In this chapter, we review the main algorithmic contributions for scheduling aircraft landings and application of multi-agent system in scheduling environment. The subsections are organized according to the main methodology used in the study. There are many variants of air landing problem and many approaches to solving it. Some takes single runway scheduling problem some multiple runway. Different solving approaches are used for optimization like dynamic programming, branch and bound, simulated annealing, genetic algorithm, ant colony optimization etc.

2.1 The Air Landing Problem

In this section, various sequencing and scheduling solution methods like dynamic programming, branch and bound, heuristics, genetic algorithm, simulated annealing and ant colony optimization are reviewed.

2.1.1 Dynamic Programming

Dynamic Programming (DP) is a general optimization technique used for taking decision of sequential activities. Almost all ALPs can be usefully demonstrated as DP problems because the algorithms can evaluate current partial solutions independently of the exact sequencing decisions used to form these solutions.(Bianco et al., 1999) takes a ALP on a single runway as a single machine scheduling problem with release dates and sequence-dependent processing times to maximize runway throughput. In his work he shows the problem is equivalent to the cumulative travelling salesman problem. For this latter problem, he give a dynamic programming formulation from which lower bounds are derived. For real time problem also proposed two heuristics methods and compare the results. The results on the aircraft sequencing problem show that a significant reduction in the aircraft mean delay, with respect to that occurred when aircraft are sequenced by FCFS discipline, can be obtained in real-time conditions using the proposed heuristic algorithms.

Bayen & Tomlin, (2004) consider an approximation algorithm for scheduling aircraft with holding time. The aim of this paper is to minimize the two objective one is the sum of the starting times of all jobs and other is the make span. In his paper he takes two problem one is solved by dynamic programming and other is solved by linear programming.

Balakrishnan, (2006) propose a dynamic programming for scheduling aircraft landing under constraint position shifting problem. The objective is to maximize the runway throughput (equivalently, minimizing the landing time of the last aircraft or make span). He uses the dynamic programing based approach in the problem and test the model on real world data for Denver International Airport.

Chandran & Balakrishnan, (2007) extended his previous work presented an approach for determining the tradeoff between robustness and throughput, while scheduling single runway operations under Constrained Position Shifting. The proposed algorithm compute a tradeoff curve between runway throughput and probability of deviation of aircraft from the schedule time. He proposed dynamic programing that handles the several source of uncertainty and that able to handle the real time application.

More recently (Lee et al., 2008) further extended the previous framework proposed by (Balakrishnan, 2006) and (Chandran & Balakrishnan, 2007) by presenting a dynamic programming algorithm for minimizing the sum of landing costs of an arrival schedule. They use this approach firstly for minimizing total delay, which is equivalent to minimizing the sum of landing times, and secondly for minimizing fuel cost, where the strategy of speeding up some aircraft at the expense of burning extra fuel is explored. He generate a 1000 problem instances of 30 aircraft sequence problem and doing the tradeoff between average delay and throughput. In the result its shows the average delay is minimized by decreasing the small amount of throughput.

2.2 Branch and Bound

Brinton, (1992) introduces one of the first branch-and-bound approaches for the ALP and the runway assignment problem. In his objective function he uses the combination of various costs and proposed a implicit enumeration algorithm. The implicit enumeration algorithm is coded to optimize the aircraft arrival sequence and schedule. This algorithm is adopted by the foundation of the Traffic Management Advisor which is part of the Center/Terminal Radar Approach Control (TRACON) Automation System (CTAS). The result found that the significant improvement are possible by combining the runway assignment and sequencing with scheduling process.

Beasley et al., (2000) design branch-and-bound algorithms by employing linear programming (LP)-based tree search approaches for both single and multiple-runway problems. The objective function is to minimize the cost of deviation from the scheduled time and proposed a mixed integer programming to solve the problem. He tested the algorithm with number of problem up to 50 aircraft and four runway.

Sölveling & Clarke, (2014) presented the stochastic branch and bound algorithm to find the optimal solution of the air runway landing scheduling problem. The bjective function is to minimize the total makespan and consider the uncertain availability of runway. The algorithm is tested and result shows the proposed algorithm is better then the deterministic sequencing model.

2.3 Heuristics

Beasleyet al., (2001) discusses how a modem heuristic technique (a population heuristic) can be used in a decision support tool to enable more effective decisions to be made with respect to one aspect of the air traffic control system, the scheduling of aircraft waiting to land. The developed heuristic algorithm is tested on using actual data related to aircraft landings at London Heathrow. The results indicate that the air traffic control decision improved by 2-5 % in terms of makespan.

Xiao-Bing Hu, (2005) introduces the concept of receding horizon control (RHC) to the problem of arrival scheduling and sequencing in a dynamic environment. The performance of the proposed new real-time arrival scheduling and sequencing method is investigated under different degrees of congestion, different levels of uncertainties, and using different receding horizons for optimization.

Soomer & Franx, (2008) consider the single runway arrival problem and focus on the cost function related to arrival delay for flights. (J. E. Beasley, M. Krishnamoorthy, Y. M. Sharaiha, 2000) show that the considered decision problem is NP-complete. The problem is not solved in zero instant by any polynomial solution algorithms the MIP problem provide optimal solution but not efficient. Therefore, he developed the heuristics that solve the problem very fast. The developed heuristics is tested on a large number of instances created using the data from a major European hub These experiments show large cost savings for the airlines compared to current practice. The heuristic is able to solve instances with over 100 flights in a few minutes.

Mahmoudian et al., (2013) proposed a time segmental heuristics method to land an aircraft on runway, assign the proper runway and calculating the scheduled landing time for each aircraft. The objective function of the problems is to minimize the deviation of target time for each aircraft. The algorithm is tested and compare with the results obtained by the (J. E. Beasley, M. Krishnamoorthy, Y. M. Sharaiha, 2000). Result shows there is no improvement occurs with existing solution due to adding a large number of additional constraints.

Farhadi et al., (2014) examines a three-faceted approach for runway capacity management, on the basis of runway configuration, scheduling approach and separation standards. These factors provide quick alternative runway settings that are encapsulated using a classical mixed-integer formulation. A heuristics is developed for optimization and applied to the Doha International Airport for testing the results. The empirical study based on historical data shows that the proposed heuristic consistently yields optimal or near-optimal schedules, with considerable savings in fuel cost and reductions in delays, while preserving the spirit of an FCFS sequencing policy.

2.3.1 Genetic Algorithm

The use of GAs is widely spread in operations research in various field of optimization like shop scheduling transportation etc. in sequencing and scheduling problems GA presents some specific features.

Glen Stevens, (1995) proposed a approach to scheduling aircraft landing times Using genetic algorithms. He checks the applicability of GA in the real time aircraft scheduling problems and objectives is to minimize the earliness and lateness of landing. The presented GA approach is tested on ten data set of 30-40 aircraft and compare with the heuristics solution but currently it not perform well.

Ciesielski & Scerri, (1998) presents the genetic algorithm and investigate the applicability of genetic algorithms to the problem of real time scheduling of aircraft arrival times at airport. For the simulation the data set is to be taken Easter at the Sydney airport. Two data set are generated, one containing 28 plane arriving in 37 minutes and other 29 plane in 38 minutes. In the results the schedule should be optimal or near to be optimal.

Cheng et al.,(1999) presents the application of GA for runway assignment, sequencing and scheduling of arrival aircraft having multiple runways. Four different genetic-search formulations have been developed by changing the chromosomal representation and applied to the runway assignment, sequencing, and scheduling problem for multiple flights arriving at an airport with multiple runways. Due to change in chromosomal representation of runway assignment and scheduling the convergence of solution is to be changed.

Capri & Ignaccolo, (2004) has been adopted the same four class subdivisions (only arrivals), time separations and objective function as proposed in the work by (Bianco et al., 1999). Two GAs are used to solve these problems, the first, called 'GAs', which objective function as optimize fuel consumption while the second, called 'GAsII', introduces the maximum landing time constraint in order to obtain more balanced solutions. The results is compared with the CIH algorithms and obtain the GAs algorithms always provide the better results. The extension of this work is to increase the number of runway.

Xiangui et al., (2010) design an efficient GA based on aircraft categories to tackle the aircraft arrival sequencing and scheduling problem. In this paper design an efficient GA whose chromosomes are constructed as the permutation of the categories of the arriving aircraft. The advantage of the resulting GA is the reduced encoding space so that the searching efficiency is promoted, which is very important for its real time application. The result is compared with (Bianco et al., 1999) CIH model.

Xiao-rong et al., (2014) present a heuristic genetic algorithm for multi-runway flights landing scheduling problem. The algorithm is based on a single chromosome coding and dynamic way flights runway allocation. The fitness function of the problem is minimum delay time. The experimental results compared with the traditional genetic algorithm, new algorithm is able to improve the operation efficiency effectively, prevent local optimum, while reducing the total delay in the process of landing. Future scope, it can be combined with ALS and takes a further research on flight departure and arrival flight cooperatively.

2.3.2 Simulated Annealing

Jungai & Hongjun, (2012) presents a simulated annealing algorithm model for optimizing arrival flight delays to reduce serious air traffic flight delays. He developed a dynamic model for optimization of the flight delay with the objective function of delay cost. The model is verify by using the data from an airport in China and compared with the traditional flight delay sequence method. The data shows the

SA model is effective and better to implement. The cost and delay are also reduce significantly by applying the developed SA model.

2.4 Ant Colony Optimization

Randall, (2002) propose the ant colony optimization model for scheduling aircraft landings problem. His algorithms objective is to minimize the difference between a estimated landing time and the actual landing time for each aircraft subject to a specified time window and the separation criteria. Bencheikh et al., (2011) consider the aircraft landing problem on single and multiple runways. In the first part a mathematical formulation of the problem and second part, he presents a new heuristic for scheduling aircraft landing on one runway. He tested both algorithms on instances involving 10 to 50 aircraft on one runway.

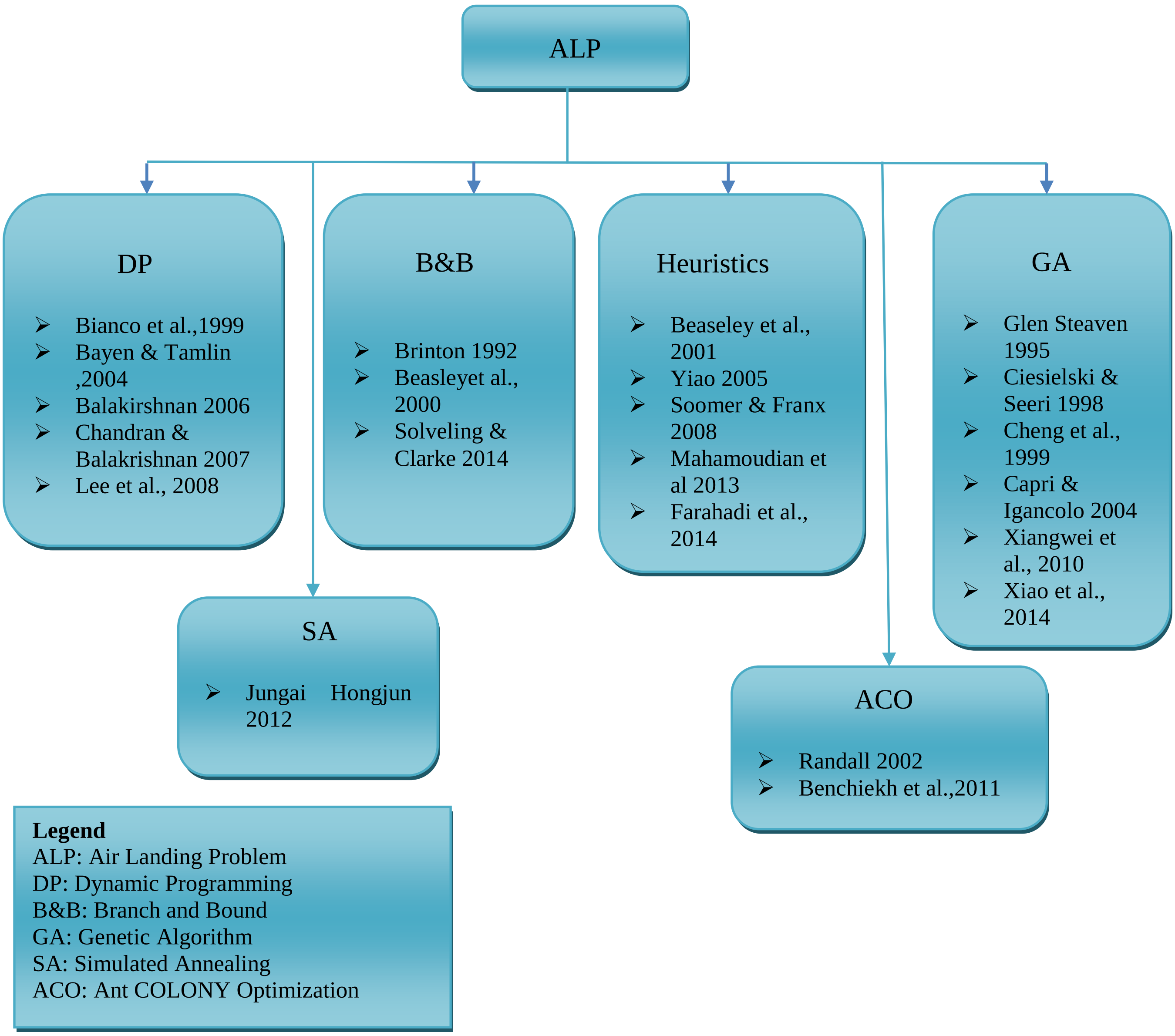


Figure 2.1: Overview of air landing problem literature

2.5 Multi-Agent System in Scheduling

The application of multi-agent system has been found in solving wide variety of complex scheduling problems such as supply chain scheduling, airport ground scheduling, transportation problem, traffic control etc. Different methodologies and techniques have been proposed, developed, and used in the literature for agent based scheduling. In this section review the application of multi-agent system in field of manufacturing shop scheduling and air traffic management.

2.5.1 MAS in Manufacturing Shop Scheduling

Cheng et al., (2006) consider a two-agent single-machine scheduling problem with truncated sum-of-processing-times-based learning considerations. He proposed Branch-and-bound and three simulated annealing algorithms to solve the problem and apply the learning effect.

Leitao, (2009) introduces a holonic disturbance management architecture in which the disturbance handling functions, including detection, diagnosis, recovery, and rescheduling, are performed by a community of autonomous and co-operative holons.

Le et al., (2011) discussed a single machine scheduling problem involving more than two agents in whom each agent is responsible for his own set of jobs and wishes to minimize the total weighted completion time of his own set of jobs. They provide the approximation algorithms for efficient scheduling considering the learning effect. For searching the optimal solution used the branch and bound and simulated annealing algorithms.

Duan et al., (2011) proposed a negotiation-based optimization method for scheduling of a manufacturing system. they are using two agents one for manufacturer and other is from suppliers. This paper presents the study of applying automated negotiation to self-interested agents each with a local, but linked, combinatorial optimization problem.

In their paper (Mor & Mosheiov, 2010) discussed a single machine scheduling problem, using two agents which are on the use of a single processor. Each of the agents needs to process a set of jobs in order to optimize the flow time. He takes two agent for the job and job is categorized into two category either agent one or other. Each job of both agent categories is processed on a single machine.

Erol et al., (2012) proposed machine scheduling together with the automated guided vehicles (AGVs) in a flexible manufacturing environment. The proposed methods is to be developed on line scheduling system based on an MAS framework for both AGVs and machines. The proposed system is implemented and tested on JACK^{TM} agent development platform. The proposed multi-agent based approach works under a real-time environment and generates feasible schedules using negotiation/bidding mechanisms between agents.

Lou et al., (2012) presented a multi-agent-based proactive-reactive scheduling for job shop scheduling problem. He uses three types of agent for scheduling task management agent (TMA), scheduling management agent (SMA), and machine agent (MA). The TMA and SMA are responsible for global scheduling and MAs are responsible for local scheduling. Two types of communication protocols are used to transfer information between agents in this architecture: one is for agents to communicate with each other directly; the other is to communicate indirectly through the blackboard.

Cheng et al., (2013) proposed Two-agent singe-machine scheduling with release times to minimize the total weighted completion time. He considers N number of jobs and two agent each agents belongs to any one of the agents. The objective of the system is to minimize the total weighted completion time of the jobs of one agent with that the maximum lateness of the jobs of the other agent does not exceed a given limit. The branch and bound algorithms solve the problem up to 24 jobs in a reasonable time and simulated annealing algorithm performs well with an average percentage error of less than 0.5% for all the tested cases.

Kaplanoglu, (2014) proposed a single machine scheduling problem with sequence-dependent setup times and maintenance constraints. The regular and irregular maintenance activities is consider in the problem and solve the problem under both of the conditions. The propose approach is entirely based on BDI agent types and BDI agent types are implemented on JACK^{TM}. Four types of agent are used in the developed framework to solve the problem.

2.6 MAS in Air Traffic Management

The Aircraft Landing Scheduling (ALS) problem is one of the important problems in Air Traffic Control (ATC). The problem is to determine the landing sequences and landing time for a given set of aircrafts. The application of multi-agent system is found very rarely in the air traffic management.

Hill et al., (2005) consider a cooperative multi-agent approach to free flight. Air traffic control will require automated decision support systems in order to meet safety, reliability, flexibility, and robustness demands in an environment of steadily increasing air traffic density. Automation is most readily implemented in free flight, the segment of flight between airports. The centralized control is impractical in this environment and requires a distributed decision making.

Mors et al., (2007) present two MAS strategies that can effectively handle aircraft deicing incidents. These MAS strategies help improve to prevent and reduce e.g. airplane delays at deicing stations due to changing weather conditions or incidents at the station. In the developed MAS strategy each agent having its own decision making capability to deal with the uncertainty. Each proposed MAS strategy outperforms a first-come first-served coordination strategy by taking both these issues.

In Agent-based decision-making process in airport ground handling management (Ansola et al., 2011) gives an agent-based decision-making process in airport ground handling operation operations involve the sequencing, control, and optimization of the operations related to assist the airplanes at their chosen parking positions. For agent interaction protocol used constant product-driven negotiation based on the Markov parameters. The Ciudad Real Central Airport is being used as a test bench to validate the design and development of MAS.

In this paper (Huang et al., 2012) propose a semantic agent negotiation mechanism for aircraft landing scheduling. The main idea of the SANM is that aircrafts can make the landing sequence through cooperation. SANM solves the problem consider the whole condition and single aircraft, while other algorithms only consider the whole condition. He taken multi runway air landing problem and genetic algorithm is used for optimization. The simulation results show this method is reasonable and it can reduce the total time delay, optimize the approach sequencing.

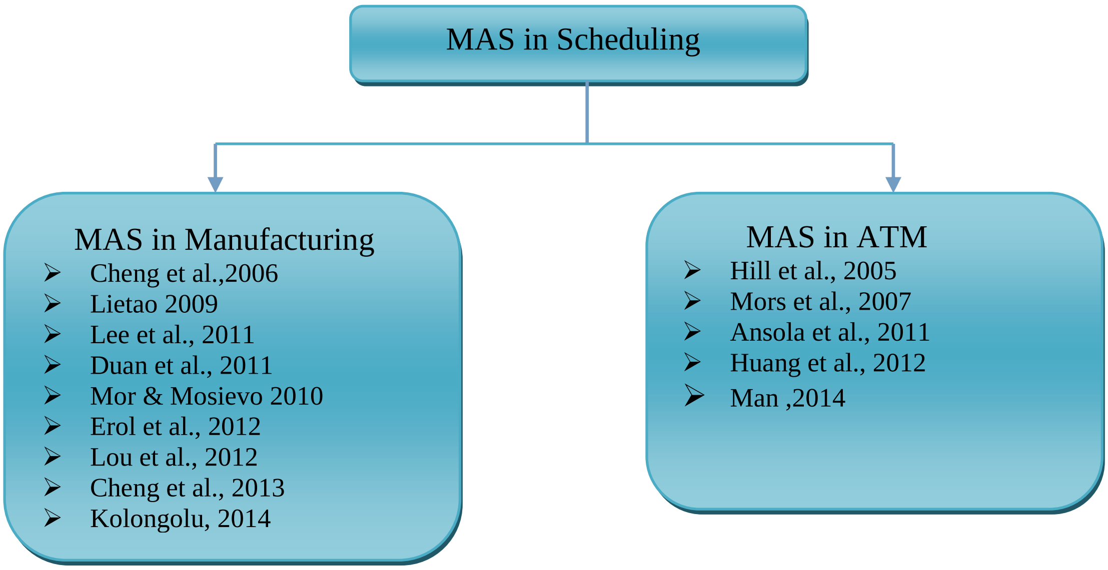


Figure 2.2: Overview of MAS in Scheduling literature

Man, (2014) presents a new approach for automated merging multiple aircraft flows in a busy Terminal Maneuvering Area (TMA). This approach to optimize a set of aircraft planned to land at a given airport based on multi-agents technique, which is automated generating comprehensive sequencing plan and conflicts-free trajectories. He designed the whole system by two main models to manage the process of arriving flows: Sequencing leg and Link. And used, 4 kinds of agents are designed to support the implementation of this automated system: aircraft agent, flow manager agent, conflict detection and resolution agent and 4D trajectory planning agent.

2.7 Summary and Research Gap

This chapter provides a review of literature on air landing problem, categorized based on solution procedure, which has been classified into major categories, dynamic programming, branch and bound, heuristics, genetic algorithm, simulated annealing and ant colony optimization. In figure shows the overview of air landing problem literature

This chapter specially reviewed an application of MAS in the field of sequencing and scheduling problem. The application of MAS found the variety of complex scheduling problems such as supply chain scheduling, airport ground scheduling, transportation problem, traffic control etc. Different methodologies and techniques have been proposed, developed, and used in the literature for agent based scheduling. Now a day multi-agent system is the emerging field of research for the new researcher in the field of sequencing and scheduling.

Following are some research gap in existing literature:

- Drawback in the heuristics and meta-heuristics: Air landing scheduling literature survey results that all algorithm for determining an optimal schedule of the problem as well as all heuristics and meta-heuristics approaches are based on centralized decision making environment, but in practical situation it is not possible.

- Practical vs theoretical models: Most of the existing research of different solving model of heuristics and meta-heuristics limited to the static environment due to exponential nature of complexity associated with such situations. A practical problem involves very large number of factors and the conditions varies dynamically in the changing environment. Since the air landing problem require intelligent techniques which are flexible, autonomous and dynamic.

- Defining the objective functions and constraints: Choosing the appropriate objective function for ALP is controversial and stakeholders (air traffic control, airports, airlines, and government) may have conflicting criteria. Thus, selecting one or more objective that can satisfy the interests of all parties, or provide an acceptable compromise, is an important first step towards the model to be implemented.

- Throughput is the primary objective for ATC: in the literature review different objective function criteria are discussed, whereas in general controllers are only concerned with throughput. In order to satisfy the all criteria controllers need more information and good decision support tools to use this information.

- Lack of communication: In the air traffic management system the communication between all the parties is very tough task. So flexible and strong communication protocols are needed to handle the dynamic situations and uncertainty.

- Robustness and flexibility: In the air landing problem different level of uncertainty are associated due to dynamic environment. these different level of uncertainty caused by weather conditions such as wind direction and invisibility, the precision of equipment's, as well as the uncertainty in pushback times and taxi times for departing aircraft. However, most studies consider a static rather than a more realistic dynamic environment.

- Use of MAS in air landing problem: Application of MAS in the various field is identified through the literature but in the field of air landing problem MAS application are not found. Some of the researcher relates the air landing problem with machine scheduling problem but no one can precede the research in MAS based air landing scheduling.

These research gaps are identified by the literature review and motivated by the above-mentioned facts, the proposed research work aims to developing a multi-agent system (MAS) for air landing sequencing and scheduling problem that provides a comprehensive solution of the problem.

Chapter 3: Basic Concepts of Air Runway Sequencing and Scheduling Problem

3.1 Introduction

The air traffic management main aims to assure safety and efficiency of air traffic flow by establishing a set of services. Mainly three types of facilities control the air traffic flow between two airports. These facilities are airport traffic control tower, terminal airspace control tower and en route control tower.

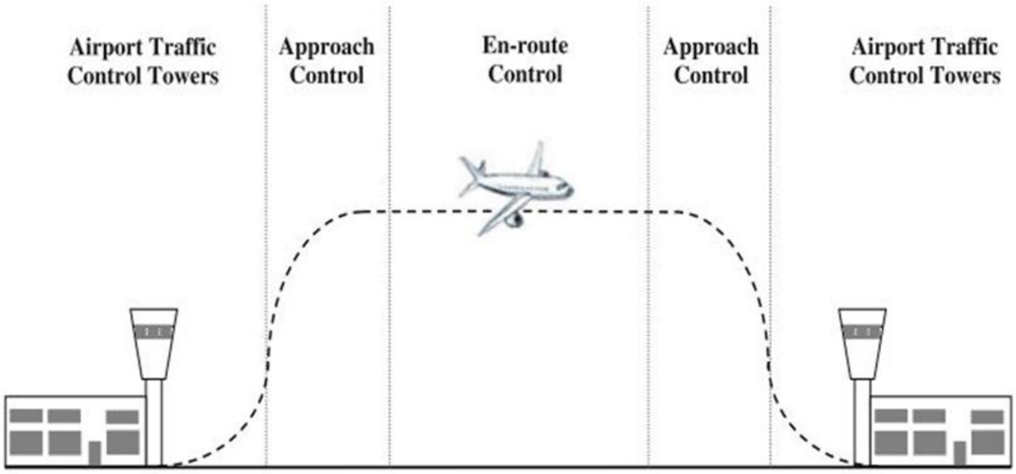


Figure 3.1 : Air traffic controller- Source (de Neufville & Odoni, 2003)

In the Figure 3.1 (de Neufville & Odoni, 2003) air traffic control tower handles the ground level activities landing and takeoff up to 5 nautical mile and 3000 ft above from ground level of airport. The terminal approach control handles up to 40 nm and 10000 ft above from ground level. The en-route control handles the situation outside the terminal maneuvering area.

The airspace is divided into the large number geographical region like sector. This sector is managed by the number of controller. The number of aircraft flying in the sector depends upon the certain limitation like safety, flight geometry, controller workload, weather, experience and expertise of that controller (Filar et al., 2001). Dividing the airspace into smaller and smaller sector to deal with increasing traffic demand but it creating the problem the possibility of mistake by pilot because of changing the radio frequencies more often and it create problem to the controller (de Neufville & Odoni, 2003).

3.2 Decision Problem

Generally air landing problem consist of sequencing, scheduling and runway assignment decision problems. In the sequencing problem determine the sequence aircraft land from the set of feasible sequence. While the aim of scheduling is to provide the schedule landing time to each aircraft as per given sequence to make certain extent of safety (Lucio Bianco & Paolo Dell'Olmo, 1997).When more than one runway is available to the airport for landing than assign a proper runway to the aircraft. During the peak period sometimes the demand of runway is exceeds the runway capacity which make problems to the runway decision (Bianco et al., 1999).

Scheduling of landing aircraft categorized into three stages: creating an initial schedule, modify the schedule and freeze the schedule (Bennell et al., 2011) . For landing aircraft initial schedule is based on first come first serve (FCFS). The aircraft is schedule to the runway for landing without consideration of other aircraft. The schedule order requires updating the schedule when new aircraft enters the radar range before 30-40 min of touch down. In the second stage the schedule is modify when new aircraft enters into the radar range. And finally schedule is frozen before two or three minutes of landing and after that no modification is to be done on the landing time.

3.3 Time Window

A time frame is fixed for each and every aircraft for landing in the initial schedule. Every aircraft have a time window of Earliest landing time and latest landing time. In Figure 3.2 shows the landing time must be lies between the earliest landing time and latest landing time. The landing time is depends on the some technical and operational constraints such as fuel limitation, path follows, runway availability, maximum and minimum airspeed, holding time etc. Time window is taken as hard constraints.

Target landing time is the time at which aircraft land to the allotted runway. As when an aircraft deviates its time window its create problem to the other aircraft.

Figure 3.2: Time Window

3.4 First Come First Serve (FCFS)

Generally sequencing of aircraft on runway is done on the basis of first come first serve. An estimated landing time is calculated on the basis of speed of aircraft, holding time, route of aircraft etc. FCFS assign the scheduled landing time for each aircraft based on the estimated landing time generated for each aircraft.

However, it has been proved that FCFS is rarely the best sequencing policy, neither in terms of airport capacity, aircraft average delay nor, even more so, for average passenger delay (Capri & Ignaccolo, 2004).

3.5 Separation

Every aircraft creates turbulence while landing to the runway, aircraft is categorized into different category on the basis of size of the aircraft like heavy, medium and lighter aircraft.

The turbulence of bigger aircrafts' is more than the smaller ones. Moreover, the time between two sequential landings is named as separation time, since the turbulence can make some serious problems for other aircrafts. Also, bigger aircrafts can resist the bigger turbulence than others (Ciesielski & Scerri, 1998).

Safety is the prime responsibility of air traffic controller (ATC). Horizontal and vertical separation of aircraft is the main ATC safety consideration. By taking aerodynamic consideration into account separation times are bounded. Aircraft generates wake vortices when landing to the airport and as an unwanted product of lift. Lift is generated by the pressure difference between the upper and lower surfaces of the wing, the pressure at the lower surface being greater than the pressure at the upper surface. At the wing tips this difference in pressure between the upper and lower wing surfaces generates a rotating mass of air-an effect known as wake vortex. The wake vortices strength generated by the aircraft is directly proportional to the mass of aircraft. Wake vortices can cause turbulent conditions for an aircraft following too close, with associated passenger discomfort and possible damage to the following aircraft. Indeed a number of aircraft accidents are believed to have been caused by this phenomena (Beasley et al., 2001). The heavier aircraft generates high turbulence as compare to the lighter aircraft. So a light aircraft following the heavy aircraft requires more separation distance.

Air speed is not fixed and it depends on the various factors such as aircraft type, weather condition and flight level. The separation standard must satisfy the triangle inequality:

$$Sij \geq X_j - X_i \,, \qquad \text{for all aircraft classes } i\,,\, j$$

where $S_{i\,j}$ is the WV separation between aircraft classes *i* and *j* (Balakrishnan, 2006)

The simple ICAO's standard international classification of aircraft is based on three weight categories (Heavy, Medium and Light), and using distance separations numbers of nautical miles. However, in the United Kingdom, the original ICAO three-group scheme has been modified to five groups (Heavy, Upper-Medium, Lower-Medium, Small and Light) to provide more appropriate separations for certain aircraft types (Bennell et al., 2011).

3.6 Runway Capacity and Assignment

Runway capacity is the very important constraints for runway assignment. Since demand for air-transportation is predicted to increase, there is a need to realize additional take-off and landing slots through better runway scheduling. The runway capacity can be defined a maximum hourly rate of aircraft landing. The runway capacity mainly depends on the runway occupancy time, availability of taxiway, weather condition, type of aircraft, performance of ATC system (de Neufville & Odoni, 2003). Instead of increasing number of runway ATC aims to use the method and technology to maximize the runway capacity.

As the air traffic develops, the limitation of the runway becomes the bottleneck during the airport operation. For example, London Heathrow airport, one of the busiest airports in the world, has only two runways (Atkin et al., 2008). When the number of approaching flights exceeds the airport capacity, some of these aircraft cannot be landed on its ′ perfect ′ landing time. The assignment of runway is done by the air traffic controller. The flight approaches to the ATC for landing than ATC fixed the time window and assign the runway as per availability.

3.7 Holding

Controller may take an aircraft at hold when landing because of runway unavailability, poor visibility, traffic congestion, weather condition, or missed time slot. The holding of aircraft is very tough task to the controller due to fuel limitation, holding area, dependency on aircraft speed, weather condition etc. (Bennell et al., 2011).

When more than one aircraft, reaches the landing terminal area at the same time, in that situation one of the aircraft has to be on hold on the basis of aircraft preferences. Or aircraft comes under the terminal area but runway is not available for landing due to runway capacity or poor visibility or weather condition than aircraft taken on hold for some time.

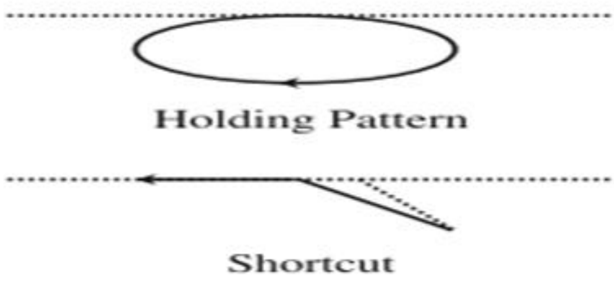


Figure 3.3: Holding pattern

3.8 Position Shifting or Push out

The deviating or advancing of aircraft is not desirable due to operating environment. Re-sequencing is difficult when aircraft is too closer to the landing than position shifting is used in the FCFS. Re-sequencing increase the capacity of runway in the terminal area but it also increase the workload of controller. Re-sequencing is done up to a certain deviation from FCFS. Due to this limited flexibility of re-sequencing motivate the constraint position shifting. CPS is the maximum number of position an aircraft may be moved or position shift (Lee et al., 2008).

Chapter 4: A MAS Framework for Air Runway Sequencing and Scheduling Problem

In this chapter, we present the details of the major issue associated with multi-agent system for air runway scheduling followed by basic framework of multi-agent based air runway scheduling considering dynamic arrival of aircraft. At the beginning, the conceptual definitions of the related term and complexity of the air runway scheduling problem is provided.

4.1 Complexity of Air Runway Sequencing and Scheduling Problem

Air traffic control in busy terminal area is one of the main challenges to the air traffic controller. Since extending the existing airport creates lots of environmental, geographical and economic problems. So the improving and optimizing the existing airport capacity is the best option. In recent years several researcher carried out the study in developing optimization model and algorithms to increase airport capacity.

The complexity of the air runway sequencing and scheduling problem is discussed below:

There is an increase in the complexity and number of aircraft, the traditional scheduling techniques can no longer provides an optimal solution under feasible time and constraints. Scheduling of air runway is the difficult problem to the air controller because of the following:

- Time Constraints: While determining the schedule for runway controller need to compute the time at that airport can utilize the runway. In the case of aircraft landing the time will be the possible arrival time at the runway. There can be a earliest time that can be an aircraft land to the runway and as well as a latest time for landing.

- Runway allocation: Most of the airport having single runway and they have work on mixed mode so the allocation of runway is the big challenge to the controller. Heathrow airport in London is the busiest airport in the word having only two runways and they work on segregated mode i.e. one is used for landing and other is for takeoff. When an aircraft approach to the controller weather runway is available or not which aircraft assign to which runway that is the big task.

- Dynamic situation: the sequencing and scheduling decision in management of multiple aircraft become more complex due to dynamic nature. This can be analyzed with respect to i) random arrival of aircraft ii) changing priorities of sequencing iii) disruption and uncertainty. In such a dynamic environment controller face a lot of decision problem. Typical problem are: analyzing the scheduled landing time weather it lies between given time window or not, deciding the precedence and allocation of runway. All these decision should satisfy the constraints and meet the stated objectives.

- Precedence constraints: Sometimes the freedom of ordering the aircraft is limited by precedence constraints. They hence specify which aircraft land first and which one takeoff after which. These constraints due to airlines preferences from banking constraints, overtaking constraints, or high priority flight.

- Environmental issues: Environmental issues play a significant role in sequencing and scheduling of aircraft. These factors are weather condition, poor visibility, noise restriction, high air speed. In such an improper weather conditions the whole schedule of landing get disrupted.

The complex decision making process require a system and models that take into account the deliberate behavior of individual decision makers. For an individual agent it is not possible to solve these complex problems. Therefore, a multi-agent system is the suitable means for modeling a framework in such environment. The reason for why multi-agent system are suitable for scheduling domain (Kaplanoglu, 2014)

- The domain is highly distributed

- Human intelligence is used for planning operation

- High amount of environmental factor exists (runway unavailability, SLT changes, weather condition, flight delay etc.)

- Need for a cooperation among the all airlines.

Recently the artificial intelligence techniques such as multi-agent technology have applied in various fields like in manufacturing system for job ship scheduling, resources allocation in project, for managing the supply chain and activity scheduling for distributed planning network. Since the multi-agent technology is considered to be a valuable techniques for the aircraft runway scheduling.

4.2 Multi-Agent Technology for Planning and Control

In recent years, multi-agent systems (MAS) have gained tremendous growth. It provides solution to the decision making problem in different domains of industrial problem e.g. electronic commerce, e-business, air traffic control, process control and telecommunications, besides manufacturing. In fact, manufacturing, transport, telecommunications and health-care are seen as the most significant domains for agent technology(Luck, McBurney, Shehory, & Willmott, 2005). The term agent refers to a software entity which is autonomous, proactive, social and having its own execution environment and decision power (Michael Wooldridge, 2009). In MAS, combination of different agent in which each agent is independent and capable of taking decision on its own belief, desire, and intension (Michael Wooldridge & Jennings, 1995).

(Russell & Norvig, 1995) given the definition of agent as follows-

“An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators.”

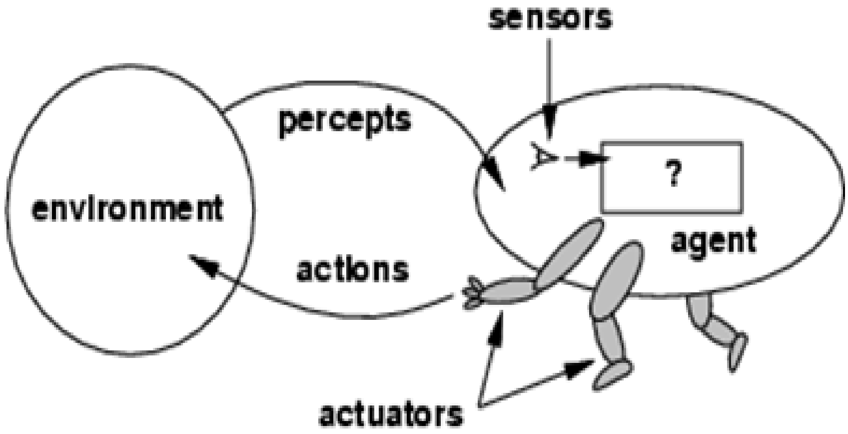


Figure 4.1: Agent and its environment, Source- (Russell & Norvig, 1995)

A MAS is distributed artificial intelligence system composed of a number of agents capable of communicating and collaborating with each other to achieve common goal. An agent and multi-agent system have many definition in the literature, the definition presented here is adopted from (Michael Wooldridge & Jennings, 1995)

“An agent is a computer system that is situated in some environment that is capable of autonomous action in this environment in order to meet its delegated objectives.”

A working definition of agent-

An agent is a computational system that interact to the one or more real world system with the following key features on varying degree:

- Autonomy: Agents operates without intervention of humans or others.
- Social ability: Agents interact with other agent by some common communication language.
- Reactivity: agents are able to perceive their environment, and respond in a timely fashion to changes that occur in it in order to satisfy its design objectives.
- Pro-activeness: intelligent agents are able to exhibit goal-directed behavior by taking the initiative in order to satisfy its design objectives.

There are four classes of agents given by (M. Wooldridge, 1997):

- *Logic-based*- in which decision making realized through Logical deduction or theorem proving.

- *Reactive agents*- in which decision making through situation to action.

- *Belief-Desire-Intention (BDI)*: it involves two processes:
  Deliberation: deciding which goals we want to achieve.
  Means-ends reasoning ("planning"): deciding how we are going to achieve these goals.

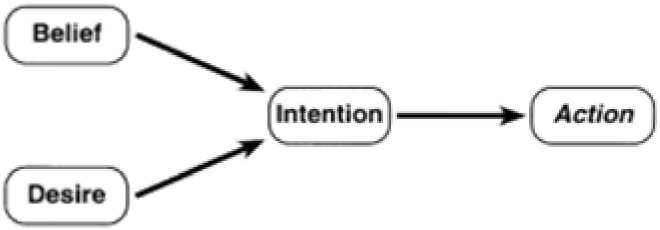


Figure 4.2: BDI architecture-Source(M. Wooldridge, 1997)

  - Beliefs: Beliefs represent the informational state of the agent, in other words its beliefs about the world (including it and other agents).

  - Desires: Desires represent the motivational state of the agent. They represent objectives or situations that the agent would like to accomplish or bring about.

  - Intentions: Intentions represent the deliberative state of the agent – what the agent has chosen to do. Intentions are desires to which the agent has to some extent committed.

- *Layered architectures*: in which decision making is realized via various software layers, each of which has some reasoning about the environment at different level of abstraction.

The study of multi-agent systems (MAS) focuses on systems in which many intelligent agents interact with each other. To interact successfully they require the ability to cooperate, negotiate and coordinate with each other. The main issues to develop MAS are coordination, cooperation and negotiation. The development of standard communication language is the major challenge to the researcher and software developer both. The agents are considered to be autonomous entities, such as software programs or robots. Their interactions can be either cooperative or selfish. That is, the agents can share a common goal (e.g. an ant colony), or they can pursue their own interests (as in the free market economy).

MAS researchers develop communications languages, interaction protocols, and agent architectures that facilitate the development of multi-agent systems. For example, a MAS researcher can tell you how to program each ant in a colony in order to get them all to bring food to the nest in the most efficient manner, or how to set up rules so that a group of selfish agents will work together to accomplish a given task. MAS researchers draw on ideas from many disciplines outside of AI, including biology, sociology, economics, organization and management science, complex systems, and philosophy.

4.3 Multi-Agent System Architecture

Within MAS, different types of agents have different degrees of problem solving capabilities within different problem domains. MAS architectures vary according to the complexity of problem domains (i.e. in number of agents, system design, and the number of variables determining agents' decision-making behavior) (Lee et al., 2008).

Baker (1998) suggest three types of MAS architecture-

- Functional architecture
- Blackboard architecture
- Hetrarchical architecture

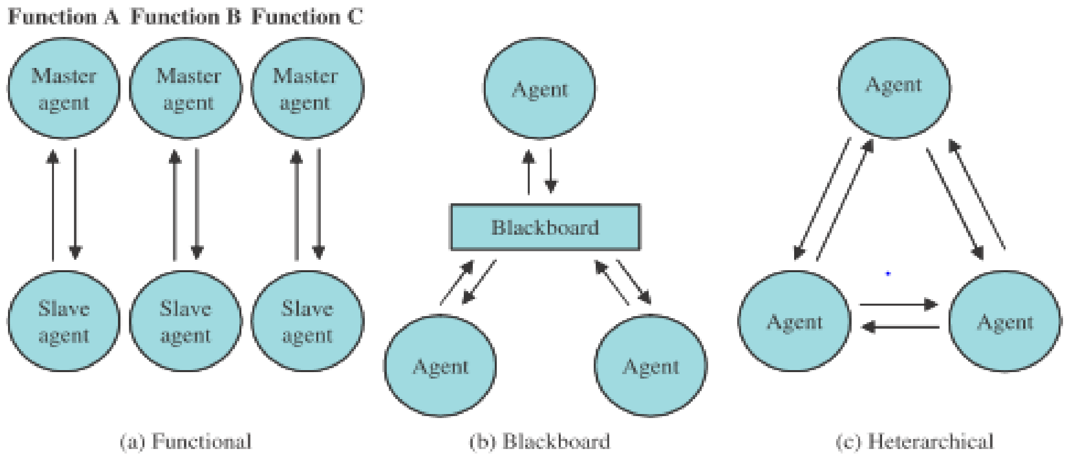


Figure 4.3: MAS Architecture-source (Park, Kim, & Lee, 2000)

The simple, '**functional**' architecture organizes each function of a total process as a single, unique (i.e. non-distributed) agent. Agents communication is done through the fixed interaction protocols (Baker, 1998). Functional architecture used in commercial software due to chief application.

A **'blackboard'** architecture is a more distributed system of decision-making, in this function is shared across the agents because each agents have expertise in certain areas. When solutions to the problem are partial, agents involved in executing functions post outstanding work to the central board hence the term 'blackboard'. Due to distribution of function among the agents the bottleneck is reduced in the system. Blackboard architectures have been most widely used in shop floor control and scheduling and in the design of job rotation schemes and elsewhere in workforce planning (Shen & Norrie, 1999).

The last MAS architecture, the 'heterarchical', combines features of the first two, featuring heterogeneous agents cooperating in hierarchical relationships. These relationships, however, unlike those of the functional arrangement, are not predetermined in advance, but arise out of agents implementing the system's needs at a given time (Baker, 1998). Within heterarchical architectures, a greater variety of flexible process routes are available for any task, with a measure of fault tolerance built in by the availability of alternative routes. While heterarchical architectures have been used on distributed shop floors, in different application settings mixed forms of architecture provide a greater degree of realism in simulating real situations.

4.4 Multi-Agent Communication Protocol

Communication is required in MAS where agent have to cooperate, negotiate with other agents. Communication protocol provides rules that structure message passing and produce meaningful conversation. Communication can enable the agent to coordinate their agents and behavior, resulting in distributed systems that are more coherent. Coordination is a property of system of gents performing some activity in shared environment. Cooperation protocol provides a framework within which agent can coordinate their action to achieve a complex task or solve a difficult problem in a cooperative manner. While negotiation protocol is used in a situation where agent have incompatible goal to enables the parties involved to reach a compromise and resolve the conflicts between agents.

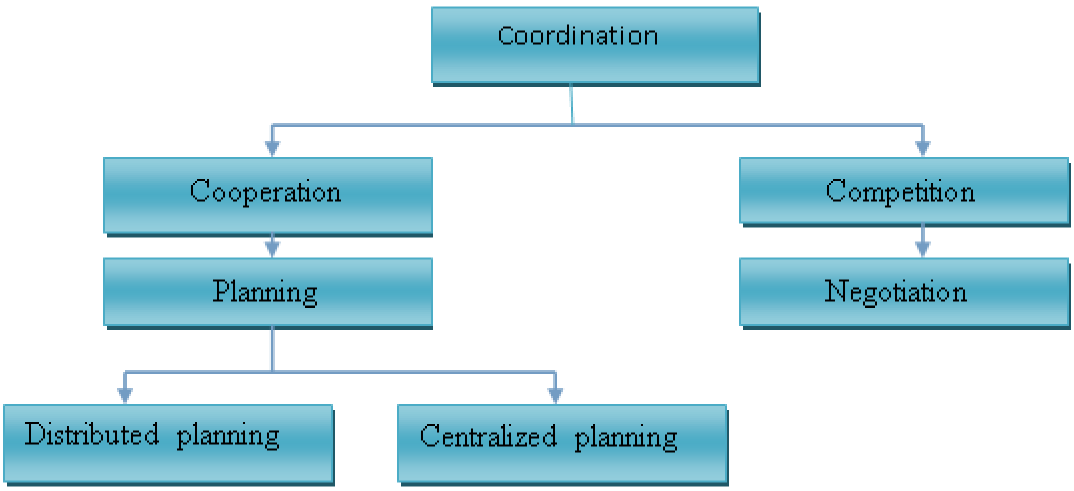


Figure 4.4: Taxonomy of agent coordination

Communication has mainly three aspects-

- Syntax- how the symbol of communication are structured
- Semantics- what the symbol denotes
- Pragmatics- how the symbol are interpreted

Figure 4.4 shows a taxonomy of some of the different ways in which agents can coordinate their behavior and activities.

Typical social communication can use for communication among computational agents. Speech act theory developed by (Searle, 1969) is a popular source for analyzing human communication via human natural language as action, such as request, reactions proposals and promises. This work further helps to develop some more languages like knowledge query, manipulation language (KQML) and the knowledge interchange format (KIF). KQML assumes that each agent maintains a knowledge base described in terms of knowledge (more accurately, belief) assertions. The foundation for intelligent agent (FIPA) developed an agent communication language (ACL). A goal for the FIPA ACL or Agent Communication Language was to specify a definitive syntax through which interoperability among agents created by different developers could be facilitated. In addition, to ensure interoperability, the FIPA ACL also specified the semantics of the primitives. Like KQML's, the FIPA ACL semantics is mentalist, although it has a stronger basis in logic. The FIPA ACL semantics is based on a formalization of the cognitive concepts such as the beliefs and intentions of agents.

Contract net protocol (CNP) (Smith, 1980) is the most familiar and widely applied interaction protocol for cooperative problem solving. CNP is best know and widely applied to distribute tasks. The contract net provides a solution called connection problem: finding an appropriate agent to work on a given task. An agent waiting a task so called as the manager, and the agent that solve the task called potential contractor.

The basic prospective for manager point of view are:

- Announces a task that needs to be performed
- Receives and evaluates bids from potential contractors.
- Award a contract to a suitable contractor.
- Receive and synthesize results

From the contractors prospective, the process is:

- Receive task announcement
- Evaluate my capability to respond
- Respond (decline, bid)
- Perform the task if my bid is accepted
- Report my results.

The structure of task announcement should include the:

- Addressee: Contractor
- Eligibility Specification: Contractors should meet certain criteria to make bids.
- Task Abstraction: A brief description of the task is used by contractors to rank tasks from several task announcements.
- Bid Specification: Tells contractors, what info must be providing with the bids. Manager compares different contractors on basis of bids.
- Expiration Time: Deadline for receiving bids.

The **Negotiation Protocol** (Michael Wooldridge & Jennings, 1995) negotiation protocols are the set of rules that govern the interaction. This covers, the permissible types of participants (e.g., the negotiators and relevant third parties), the negotiation states (e.g., accepting bids, negotiation closed), the events that cause state transitions (e.g., no more bidders, bid accepted), and the valid actions of the participants in particular states (e.g., which can be sent by whom, to whom and at when). In systems composed of multiple autonomous agents, negotiation is a key form of interaction that enables groups of agents to arrive at a mutual agreement regarding some belief, goal or plan. The process of negotiation may be of many different forms, such as auctions, protocols in the style of the contract net, and argumentation, but it is unclear just how sophisticated the agents or the protocols for interaction must be for successful negotiation in different contexts. These are the some attribute of negotiation are efficiency, stability, simplicity, distribution and symmetry. System of negotiation is either environment centered or agent centered. In environment centered agents can interact productively and fairly irrespective of their capabilities or intentions.

Airport use the negotiation mechanism to optimize the scheduling to optimize the problem. The negotiation between aircraft agent, runway agent and airport agent is held to satisfy the airlines objectives and achieve the best schedule.

4.5 Agent Development Environments

Software framework assist the development of intelligent multi-agent system that is used by a various community of user as a tool for supporting research and real world application (Bellifemine et al., 2008). This development environment framework save developers time and support in the standardization of MAS development. Java is the most commonly used language for developing multi-agent system. There are many agent development environment are available for developing the agent system. Table shows the different well known java packages.

Table 4.1: Environment for multi-agent development

Sr. No.	Agent Platform	Developed by	Main Characteristics
1	*3APL*	Utrechet University	A programming language for implementing agent-based systems via programming constructs to deal with agents' beliefs, goals, basic capabilities and integrated with Java and Prolog.
2	Cougaar	Cougaar software Inc.	It is a Java-based framework for large-scale and flexible distributed agent-based applica- tions. It uses a flexible component model to dynamically load components
3	BOND	Purdue University	A java based distributed object system and agent framework
4	FIPA –OS	Nortel	A component based architecture to enable the development of domain specific agent which can utilize services of FIPA platform
5	Hive	MIT Media Lab	A java based platform for distributed application
6	JIAC	Technical University Berlin	A java class library for the development of a universal architecture of a agent oriented architecture
7	*OAA*	SRI AI Center	A framework for integrating hetrogenious software agents in a distributed environments
8	Agent Factory	University College Dublin	A distributed FIPA-complaint Run-Time environment that support development of multi-agent system
9	JATLite	Stanford Univ. Center for Design Research	A package of java class that allow the user to create new agent and communicate via internet
10	Zeus	British Telecom Lab ISR Group	A library that facilitate the design, development and deployment of agent
11	JADE	Telecom Italia	Software Framework fully implemented in the Java language. It simplifies the implementation of multi-agent systems through a middle-ware that complies with the FIPA specifications and through a set of graphical tools that support the debugging and deployment phases
12	JACK	Agent Oriented Software Pty. Ltd.(AOS)	A framework in java for multi-agent system development that uses BDI architecture. JACK is a mature, cross-platform environment for building, running and integrating commercial-grade multi-agent systems.

4.6 A MAS Base Framework for Air Runway Sequencing and Scheduling Problem

The proposed work attempts to develop a multi-agent system framework for solving the air runway sequencing and scheduling problem where the agents in the air runway system must be able

- To find the best sequence of aircraft

- To maximize the runway throughput
- To deal with system dynamics (flight delay, flight cancellation, weather condition, runway unavailability, etc.).

Agent architectures are the fundamental mechanisms underlying the autonomous components that support effective behavior in real-world, dynamic and open environments. In this work BDI architectures used to model the proposed landing problem. This BDI architecture of agents is implemented on java based agent platform JACK™. The JACK™ agent language is based on java programming language and is used for implementing agent oriented software systems. JACK™ has its self-modeling interface which runs under BDI architecture (Kaplanoglu, 2014). In BDI model, JACK™ intelligent agents are autonomous software components that have explicit goals to achieve and events to handle. BDI agents are programmed with a set of plans in order to achieve these desires. In the varying circumstances the goal is achieved through a set of plans. Set to work, the agent pursues its given goals (desires), adopting the appropriate plans (intentions) according to its current set of data (beliefs) about the state of the world. This combination of desires and beliefs initiating context- sensitive intended behavior is a part of what characterizes a BDI agent.

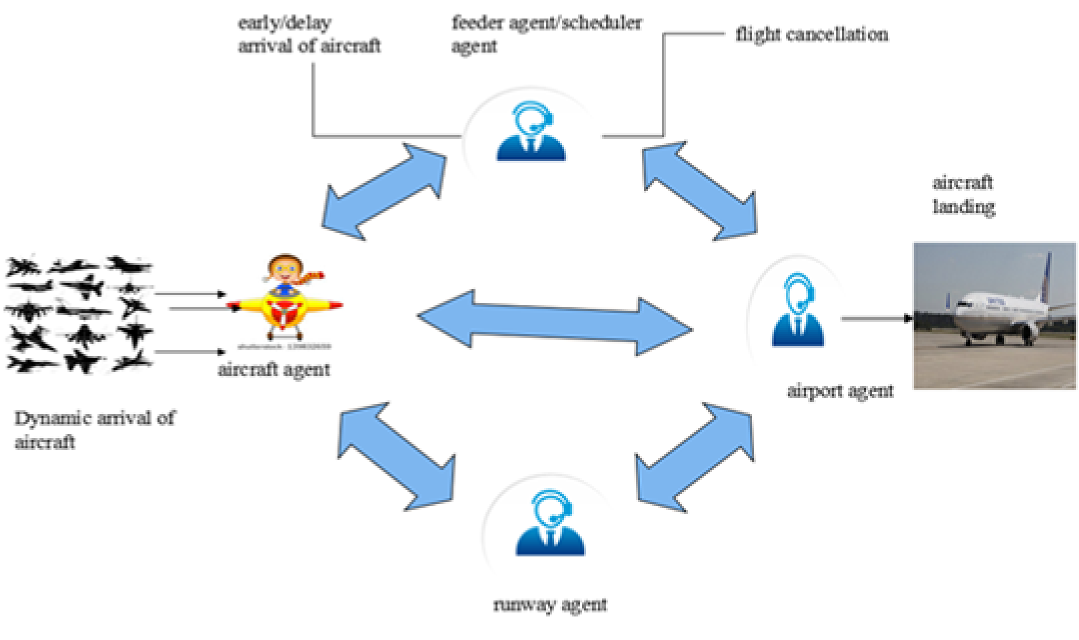


Figure 4.5: Proposed multi-agent based approach overview diagram

The proposed MAS approach using the BDI agent types and this agent types are design and implemented on agent software JACK™. Blocks between sequencing activities, aircraft, job feeder and runway managers are represented as BDI agent types in the proposed approach. Unexpected events are represented as event types. The major goal for JACK™ were to provide developers with a robust, stable, light-weight product, to satisfy a variety of practical application needs, to ease technology transfer from research to industry, and to enable further applied research. It has been designed for extension by properly trained engineers, familiar with agent concepts and with a sound understanding of concurrent object-oriented programming(Paolo et al., 1999). The proposed view of multi-agent based approach is given in **Figure 4.5**, It consists of four types of agents:

- Aircraft agent
- Runway agent
- Feeder agent
- Airport agent

4.6.1 Aircraft Agent

Aircraft agent is an agent type that stands for the dynamic arriving of aircraft. Aircraft agent is created by the feeder agent after entering the aircraft into the control area. Every aircraft have its own agent which communicated to all other agents. The aircraft agent sends the proposal to the aircraft agent when enters into the aircraft control area for requesting the time slot and assign the runway.

4.6.2 Feeder Agent

It creates the job agent based on the aircraft which are arriving to the system dynamically. Feeder agent have its own schedule on the basis of that schedule it takes the traffic decision between the destination airports. The unexpected event are also handled by the feeder agent like delay/earlier of flight, flight cancellation etc.

4.6.3 Runway Agent

When aircraft agents send the proposal to the airport agent for requesting to assign the runway, the airport agent sends the request to the runway agents. Runway agent sends the information to the all runway managers to check the availability of runway and receive the back information from all the runway managers. On the basis of collected information runway agent checks the appropriate runway availability and sends the information to the airport agent.

4.6.4 Airport Agent

The airport agent is the main agent where all the decision and optimization is to be done. When aircraft agent send the request to the airport agent than it send and collect all the information from other agents and on the basis of optimization techniques given to the block agents it allot a time window. It responds to call for proposal of aircraft agent with a minimum completion time. Airport agent makes its own reasoning and decision making capabilities while making the aircraft sequencing decision and scheduling is to be done by the optimization techniques used.

After introducing the agent types, their conceptual designs and implementations on JACK^{TM} are illustrated. Figure 4.6: Flow diagram of multi-agent based approach, after arrival of the new aircraft into control area the negotiation starts between the agents for find the best schedule for landing the aircraft.

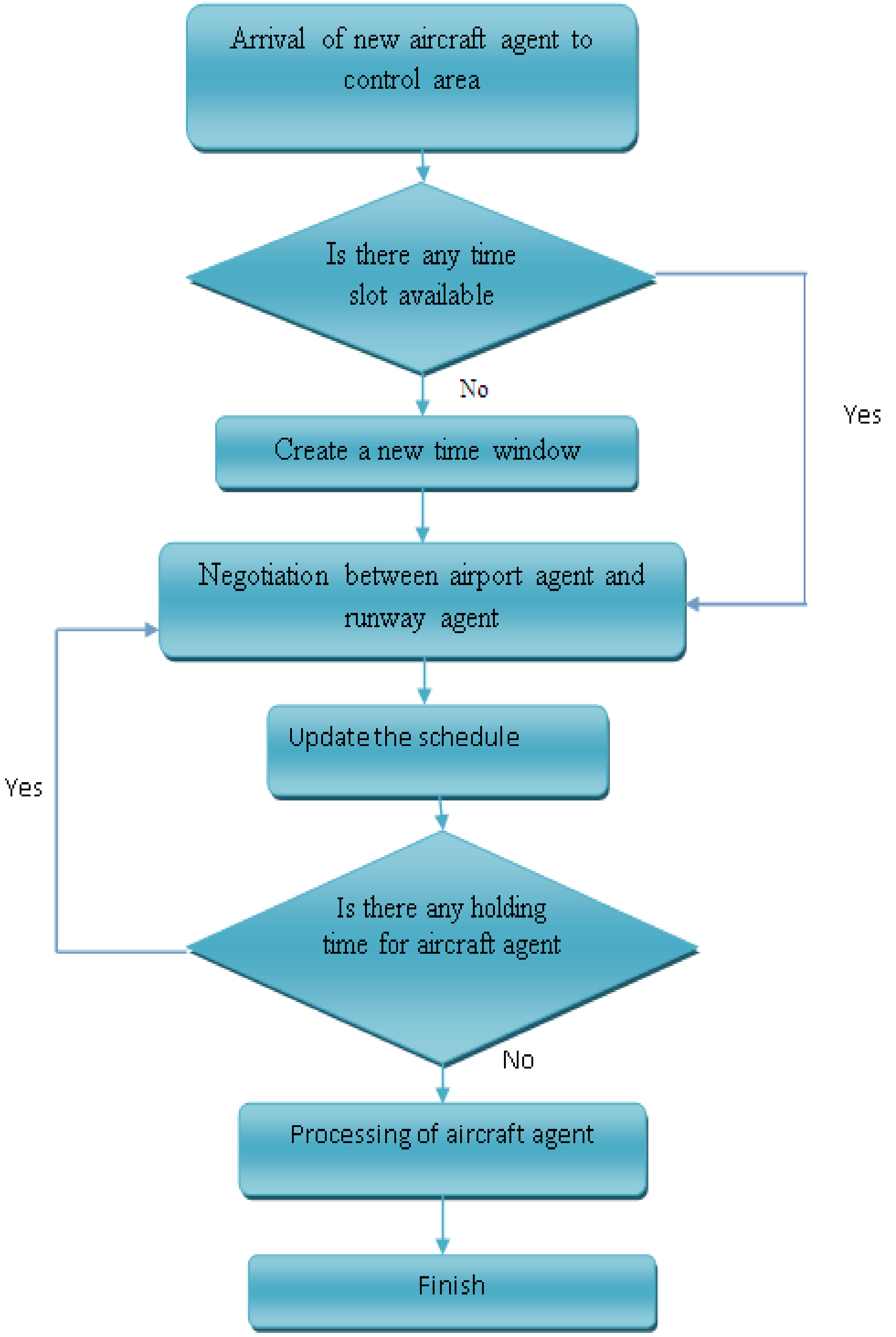


Figure 4.6: Flow diagram of multi-agent based approach

4.7 Summary

This chapter resents some definition, issues and challenges related to air runway scheduling and multi-agent system followed by developed the basic approach for air runway sequencing. The architecture of the multi-agent system is developed and basic of all agents are described briefly. In extension to this multi-agent approach the development of the system on JACK^{TM} is described in next chapter with detail. The next chapter described the documentation and design view of multi-agent system on JACK^{TM} for air runway sequencing and scheduling.

Chapter 5: A MAS for Air Runway Sequencing and Scheduling Problem

In extension to the approach developed in previous chapter, this chapter deals with the design and documentation view of multi-agent system on JACK^{TM} for air runway sequencing and scheduling. The multi-agent architecture designed and developed on BDI architecture based java agent environment JACK^{TM} . The sequencing procedure is a bidding protocol where the arrival airport manager first queries all runway managers for their respective "best landing time" for an aircraft, and thereafter it chooses one. The "best landing time" for a runway allows faster aircraft, that arrive later to the airport control area, to "push out" earlier, already assigned slower aircraft. In those cases, the slower aircraft will be re-assigned when the "push out" occurs.

5.1 Introduction

The air runway scheduling problem is more similar to machine scheduling problem. In machine scheduling job is arrived dynamically and machine as a resource similar in air runway problem aircraft enters to the control area dynamically and runway as a resource.(Brentnall & Cheng, 2009) compare the air landing problem to the machine scheduling problem. The air runway sequencing and scheduling problem can be viewed as a machine scheduling problem with sequence-dependent processing times and setup times for optimizing objective functions such as makespan and tardiness by penalizing early and late jobs in terms of time windows. The following analogy illustrates the relationship between the ALP as a machine scheduling problem. The job in machine scheduling correspond to a arriving aircraft, machine represents the runway, the ready time (release time) of the job corresponds to the Estimated Landing Time (ELT) of the aircraft, the starting time of the job represents the Actual Landing Time (ALT) of the aircraft, the completion time of the job corresponds to the time the aircraft frees the runway, and the sequence-dependent processing time between jobs represents the separation between aircraft. There is a crucial difference between the ALP and common machine scheduling problem, which is the minimum required separations must be respected not only between immediate successive jobs (aircraft) but also between any pairs of jobs.

In recent years, Multi-agent Systems (MAS) have gained tremendous growth in providing solution to the decision making problems in different domain of industrial life. We proposed a Multi-agent System (MAS) for air runway sequencing and scheduling based on negotiation protocol. The uniqueness of this work with respect to the other state-of- art approaches is that our approach is totally new in the field of air runway problem. The application of multi-agent system in manufacturing field is very much but in the field of air runway problem MAS is a novel approach.

5.2 Problem Definition

In this section describe the air runway sequencing and scheduling problems objective function and various constraints that has been considered within the literature. In this research work there is an undefined set of aircraft arriving to an airport environment.

Before a aircraft is landing on a runway a scheduled time window is required. Window time depends on the previously landed aircraft type, and separation time, on the runway. The Federal Aviation Administration (FAA) has fixed minimum separation requirements between aircraft to prevent turbulence (wake vortices) generated by a leading aircraft and to establish the aerodynamic stability. FAA has categorized the aircraft into three weight class small, large and heavy and fixed in trail separation requirements to ensure safety. Later on to increase the capacity of the runway it classified into five categories by the many researchers so that the separation time is to be minimized between the aircraft.

The main objective of the problem is to maximize the throughput of airport. There are three ways to define the optimality (throughput) of a scheduling strategy:

1) Minimization of the sum of Landing Processing Times (LPT)

2) Minimization of the overall Makespan

3) Minimization of the Last Aircraft of the sequence's LST

The air runway problem can be formally defined as follows:

- There are indefinite number of aircraft *(i=1 ….N)* that are to be scheduled.
- Each aircraft have a scheduled time window and the actual target time must be lies between these time windows.
- E_i is the earliest time for landing and L_i is the latest time for landing.
- X_i is the actual time of landing the aircraft on scheduled runway.
- R is the set of runway, only one runway is available in the airport for landing.
- S is the separation time between two aircraft to insure the safety of passenger.
- Each aircraft landing exactly one runway.
- Push out is to be occurs if the scheduled aircraft is slower than the any other aircraft which is enters into the control area with a faster speed. Slower aircraft is re-scheduled after the push-out occurs.

Mathematical model:

Decision variable

X_i: The scheduled landing time of each aircraft i, calculated by trajectory synchronizer equipment after entering the aircraft into the radar range.

Parameters

n: The number of aircraft to be scheduled.

X_{ij}: Defined to be 1 if aircraft *i* land before (not necessarily immediately) aircraft *j* and 0other wise.

E_i: The expected (or target) landing time of aircraft *i*, based on the assigned time slot which is normally specified in flight plan.

T_i: Aircraft type *i* in size category based on three different types of aircraft in small, medium and large.

S_{ij}: Be the minimum time separation between aircraft *i* and *j*, if aircraft *i* land before aircraft *j*.

R: number of runway available *(R=1)*

Objective functions:

Maximizing runway throughput: To maximize the runway through put minimize the total landing time of the arriving aircraft instead of maximizing the number of aircraft landing on the airport.

$$Minimize \sum_{i=1}^{n} Xi$$

Constraints:

- *Time window*:

$$E_i \le X_i \le L_i$$

Actual landing time must be lies between the earliest landing time and latest landing time.

- *Minimum separation distance:*

$$X_j - X_i \ge S_{ij}$$

Aircraft should be in a safe distance from other aircraft to avoid turbulence creating by aircraft ahead.

- *Runway use restrictions :*

$$X_{ij} + X_{ji} = 1$$

Each runway can be used by only one aircraft at the same time. So, the aircraft *i* lands before the aircraft *j* or vice versa.

5.3 Agent Based Mechanism for Air Runway Sequencing and Scheduling Problem

For air runway sequencing and scheduling a MAS is proposed that is designed as BDI (Belief, Desire, Intension) architecture for decision making which has been widely used in shop floor control and job shop scheduling (Kaplanoglu, 2014). The work is similar to the Kaplanoglu, (2014) applied the JACK^{TM} to develop the MAS for machine scheduling problem.

The multi-agent system proposed in this work uses autonomous, self- interested and competitive agents. The agents are reactive which respond to environment changes or other agent messages. As shown in Figure 5.1, this is the design view of agents on JACK^{TM}.

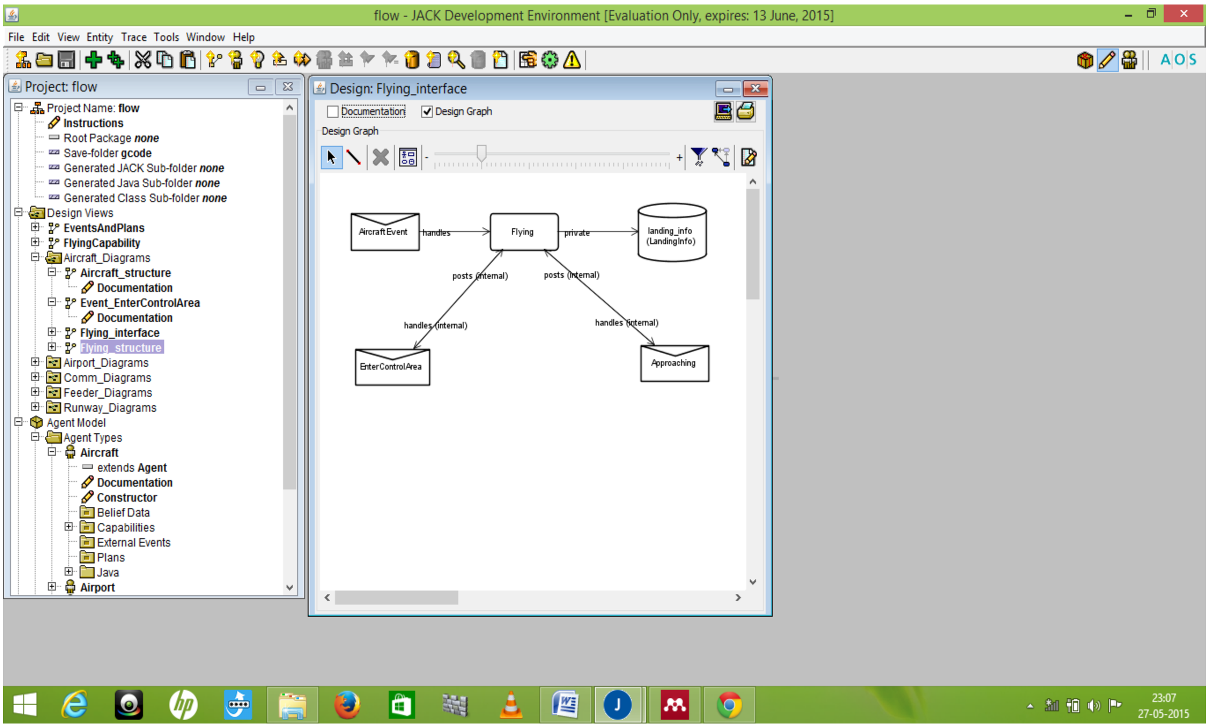


Figure 5.1: Aircraft agent design view in JACK^{TM}

5.3.1 Aircraft agent

Whenever new aircraft enters into the airport control area a new aircraft agent is dynamically created by the feeder agent. Figure 5.2 shows the aircraft structure its capabilities and how to handle the events and plan. Aircraft agent have the flying capability construct post the “EnterControlArea” event which is turn handed by the “MonitorAircraft”, “InitialApproach”, “FollowApproach” and “AssignSlot” plan construct. The aircraft agent have the flying capability contains the tracking of the approach from when the aircraft enters the destination airport control area. A private named data available which reads the information from all the plans.

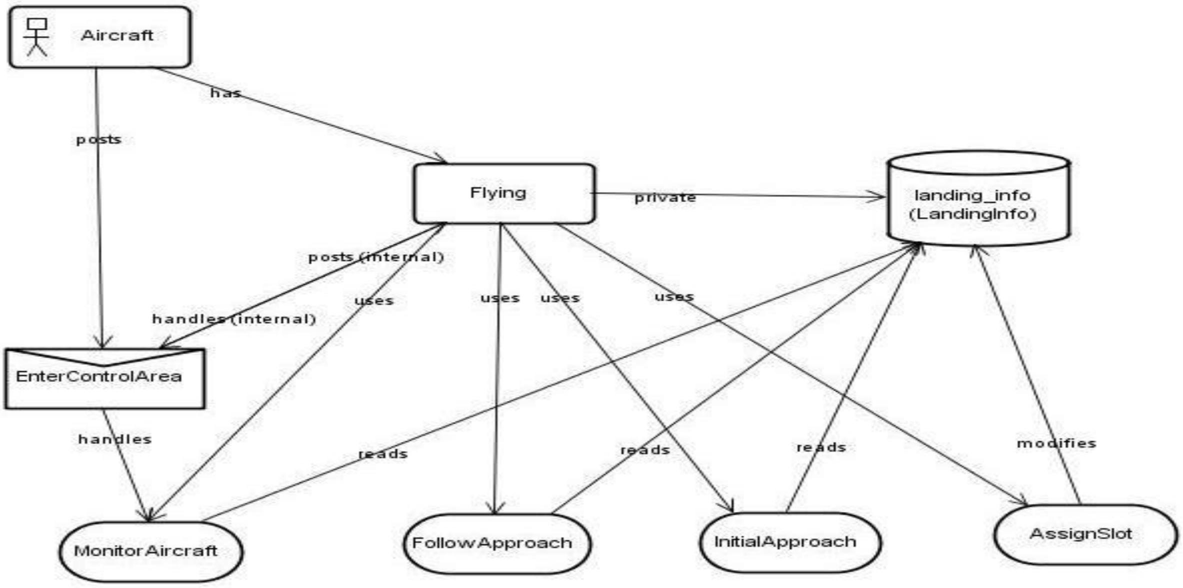


Figure 5.2: Aircraft agent structure diagram

Event "EnterControlArea": In Figure 5.3 shows the overview of the event "EnterControlArea" and the plans and data it uses to guide an aircraft during the process of landing.

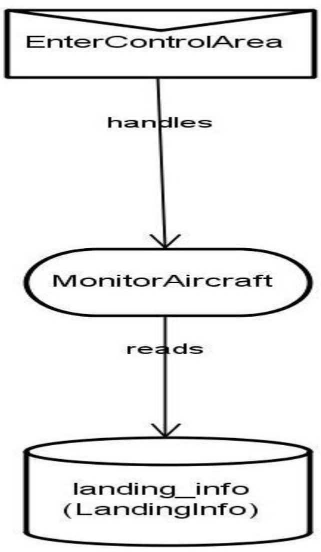


Figure 5.3: Review diagram Event enter control area

The "EnterControlArea" event marks for an aircraft that it enters the control area of the destination airport. The "MonitorAircraft" plan is invoked for monitoring the flight of an arriving aircraft, from entering the control area to its landing and the "MonitorAircraft" reads the landing info from the previous stored information.

An overview of the "Flying" capability interface structure: In the Figure 5.4 shows the "Flying" capability contains the tracking of the approach from when the aircraft enters the destination airport control area. The event "AircraftEvent" is used in the messaging between aircraft and airport.

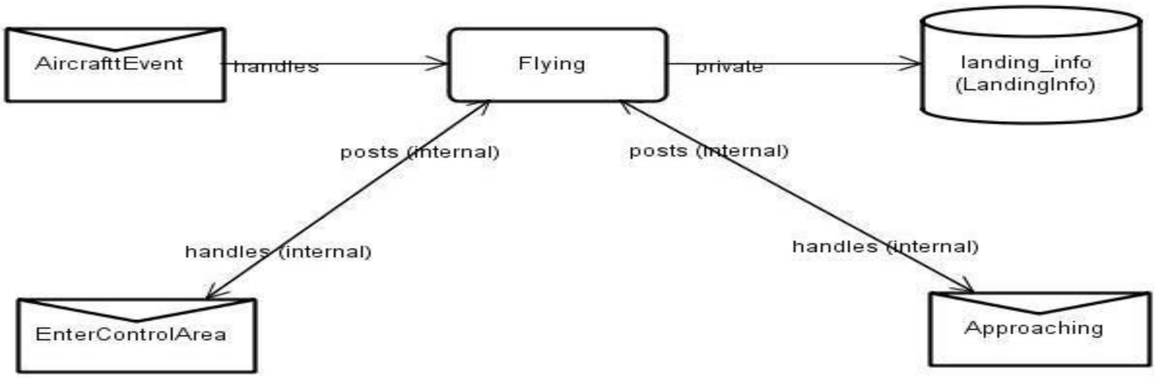


Figure 5.4: Flying interface structure

The "Approaching" event marks for an aircraft the period from entering the control area of the destination airport to the landing of the aircraft. It is posted after that the aircraft has entered the control area, and succeeds when the aircraft has landed. Being an event extended in time, the Approaching event is a BDI Goal Event. It is also "forgiving", allowing a failed plan to re-enter the applicable set immediately (if it again is applicable). This means that the event is insistent and will fail only if there is no applicable plan at all when the event is re-posted.

An overview of the "Flying" capability: The "Flying" capability contains the tracking of the approach from when the aircraft enters the destination airport control area.

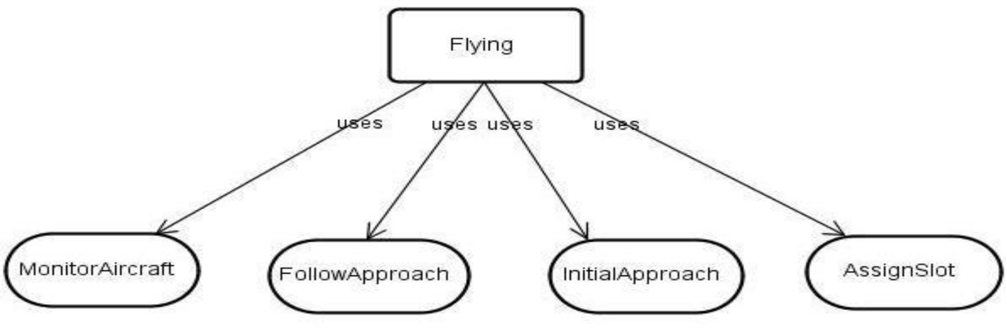


Figure 5.5: Overview diagram of flying capability

In Figure 5.5 sows when aircraft enters the control area the “Flying” capability starts tracking of the information which uses the “MonitorAircraft, FollowApproach, InitialApproach and AssignSlot”. The “MonitorAircraft” plan is invoked for monitoring the flight of an arriving aircraft, from entering the control area to its landing and The “InitialApproach” plan progresses an aircraft to landing during the initial phase when the landing allocation is not yet provided. The “FollowApproach” plan progresses an aircraft to landing when a landing allocation is provided. The plan “AssignSlot” digests an ATL notification from the airport by updating the “landing_ info” belief.

5.3.2 Runway Agent

Runway agent has the capability to assign different runways to different aircrafts waiting to land. It checks the availability of runway when aircraft agent approaches to the airport agent for landing. Send the message to the entire runway manager and receive the status of all runways then send the data to the airport agent. Figure 5.6, shows the overview diagram of runway agent, its capability, events and plan. The “Runway” agent having “RunwayAssigning” capability constructs the “AircraftEvent” event which is the turn handed by the “RunwayAssign” and” RunwayRequest” plan constructed.

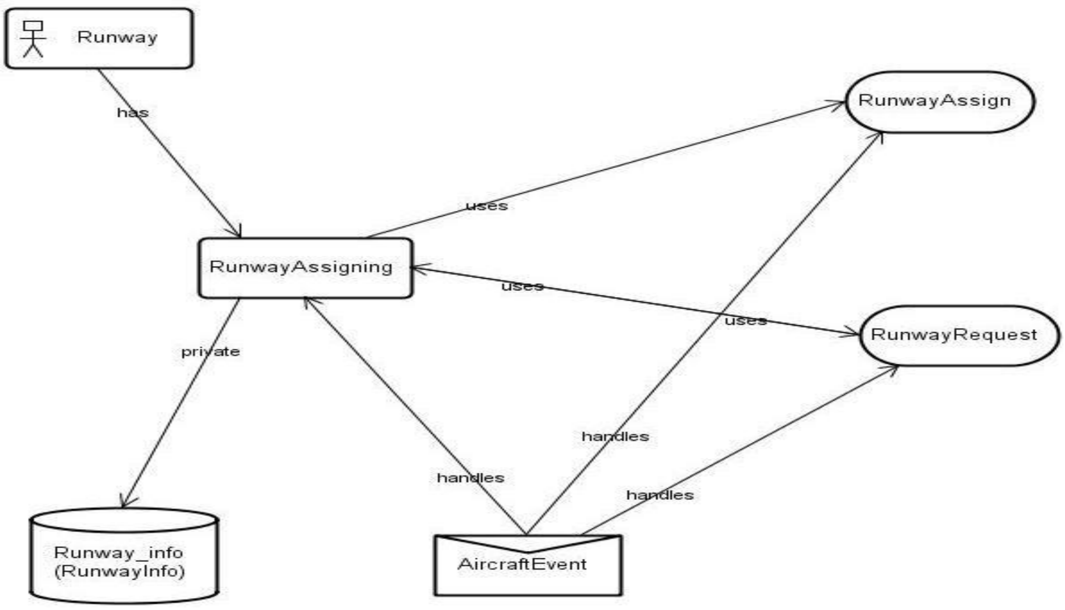


Figure 5.6: Overview diagram of Runway agent structure

An overview of “RunwayAssigning” interface :Figure 5.7: RunwayAssigning interface diagram. The “RunwayAssigning” capability contains the bidding side of the runway assignment negotiation. The event “AircraftEvent” is used in the messaging between aircraft and airport. Capability uses the runway information already available to the runway agent.

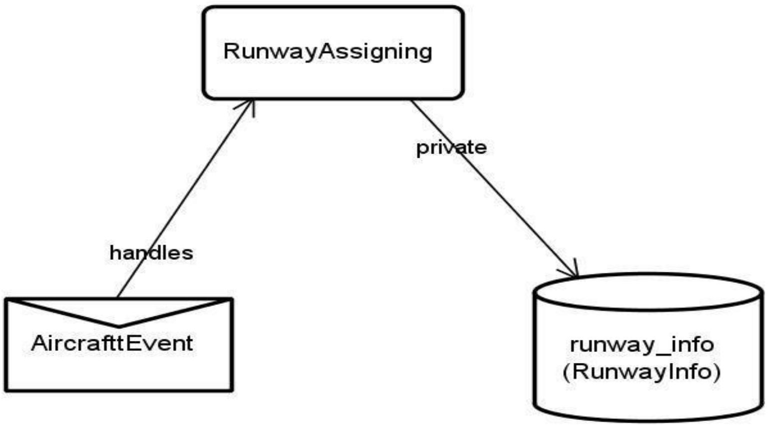


Figure 5.7: RunwayAssigning interface diagram

An overview of the "RunwayAssigning" capability: The "RunwayRequest" plan responds to an "AircraftEvent" request by suggesting an allocation for this runway. The allocation inspects all time slots from the given ETA, to find the first that is unused, or used with an allocation of "lesser importance". If this request is an early booking, then it may push a previous earlier booking if this ETA is prior to the earlier booking's ETA. If this request is the arrival request, then it may push an earlier booking (regardless), or a previous arrival assignment this ETA is prior to the previous assignment's ETA. In Figure 5.8 shows the "RunwayAssign" plan responds to an "AircraftEvent" assign by filling an allocation for this runway. If the allocation slot is already occupied, the occupant is re-scheduled. Also, the aircraft getting an allocation is notified.

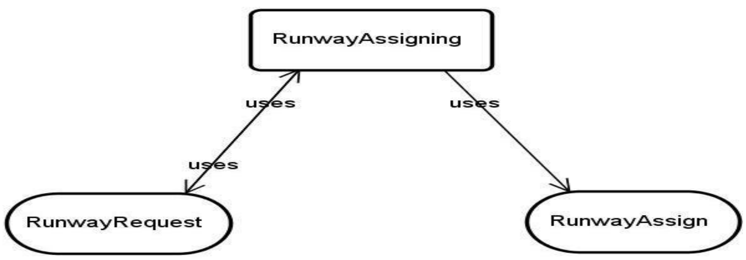


Figure 5.8: Runwayassigning structure diagram

5.3.3 Feeder Agent

The "Feeder" agents model source airports and other "sources of aircraft" and each "Feeder" agent has it's own schedule. The "Feeder" agent generates the "Aircraft" agent when new aircraft enters into the control area, and it updates the traffic schedule which is already having it. Figure 5.9, shows the complete structure of "Feeder" agent. This agent has the "TrafficFeeding" capability which post the "TrafficEvent" that is handle by the agent using "Traffic, Takeoff and TakeoffDiscard" plan.

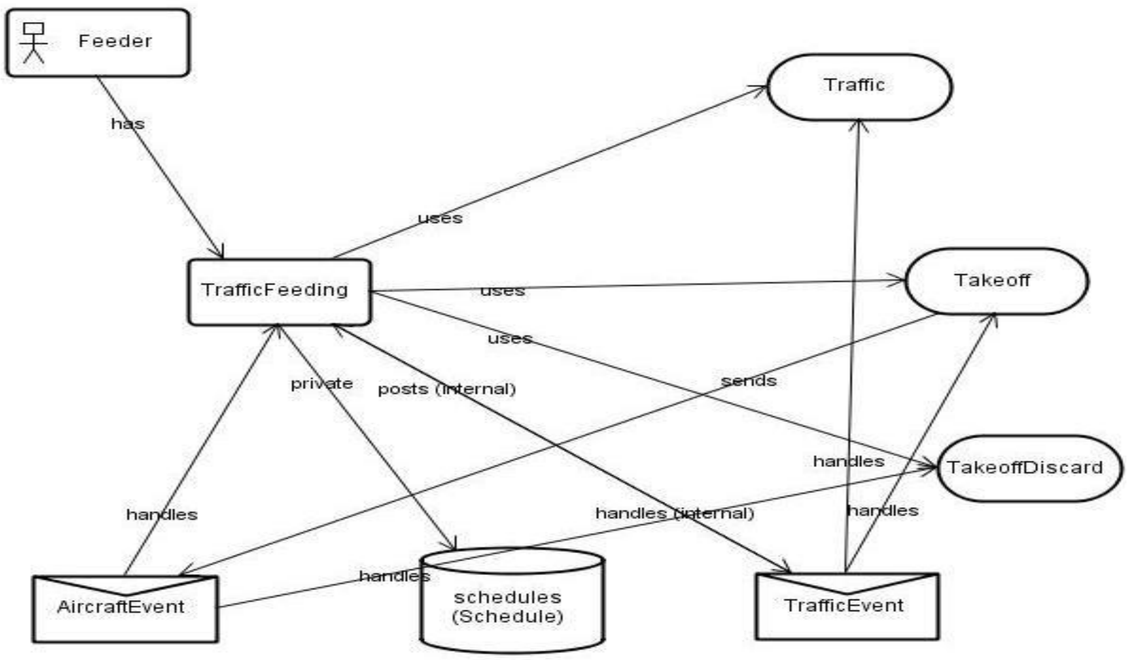


Figure 5.9 : Overview diagram of Feeder agent structure

An overview of the "TrafficFeeding" capability:

The "TrafficFeeding" capability contains the processing of a departure schedule. It uses the "Traffic, Takeoff and TakeoffDiscard" plans which are shown in Figure 5.10.

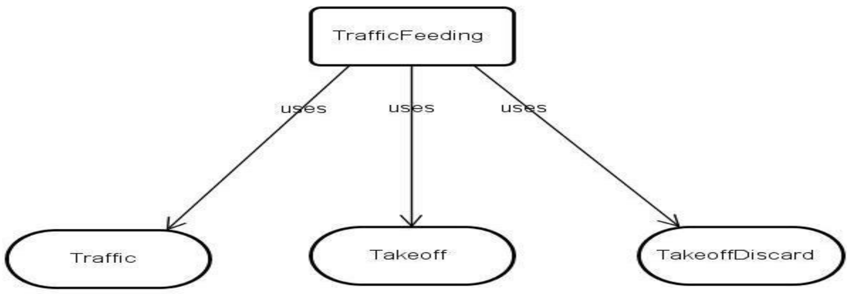


Figure 5.10: Traffic feeding structure diagram

The "Traffic" plan processes the Schedule for a Feeder and "Takeoff" plan handles a Schedule Row for a Feeder. It first issues a booking request for the aircraft concerned. Then it waits until the aircraft is in the destination airport's control area, at which time it constructs an "Aircraft" agent. The "TakeoffDiscard" plan is a handler for the "AircraftEvent" assign message returned from the airport for the booking, though it is never relevant, because we don't care about the on-route behavior.

An overview of Traffic feeding interface: Traffic Feeding capability's to handle events and use schedule for scheduling traffic shown in Figure 5.11.

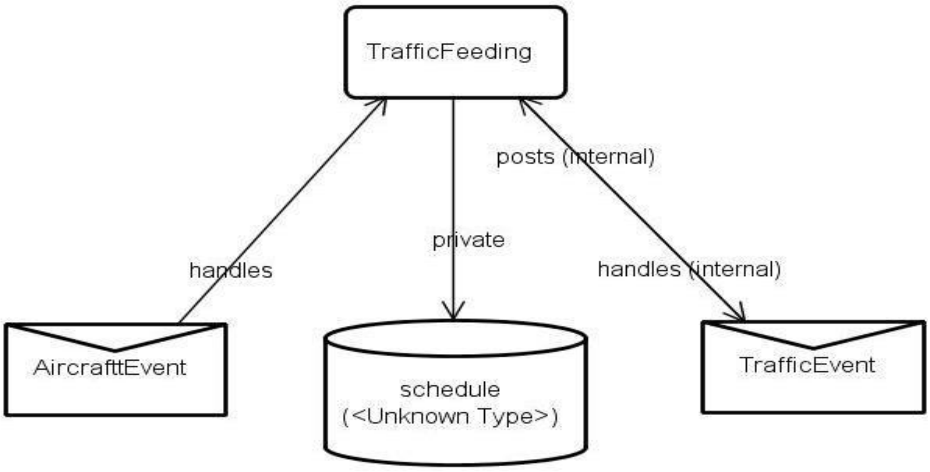


Figure 5.11: Traffic feeding interface

The "TrafficFeeding" capability handles by the "AircraftEvents" and posts the "TrafficEvent" it uses the private schedule. The event Aircraft Event is used in the messaging between aircraft and airport. A "TrafficEvent" marks events in the feeder traffic. An initial "TrafficEvent" schedule is posted at agent construction, and then "TrafficEvent" aircraft events are posted for the schedule rows.

5.3.4 Airport Agent

Airport agent showing the capability of sequencing and scheduling the aircraft arrivals. The "ArrivalSequencing" capability contains the handling of landing requests from aircraft through negotiation/bidding protocols with available runways for an appropriate landing allocation.

Figure 5.12, shows the "Airport" agent has the "ArrivalSequencing" capability which post the "AircraftEvent" that is handled by the agents by using the "RequestSlot" plan. The "ArrivalSequencing" capability contains the handling of landing requests from aircraft through negotiation/bidding with available runways for an appropriate landing allocation and uses the "Requestslot" plan types. "RequestSlot" handles an "AircraftEvent" request by propagating it to all available runways, collecting all their suggestions, choosing the best one, and then notifying the runway and aircraft concerned.

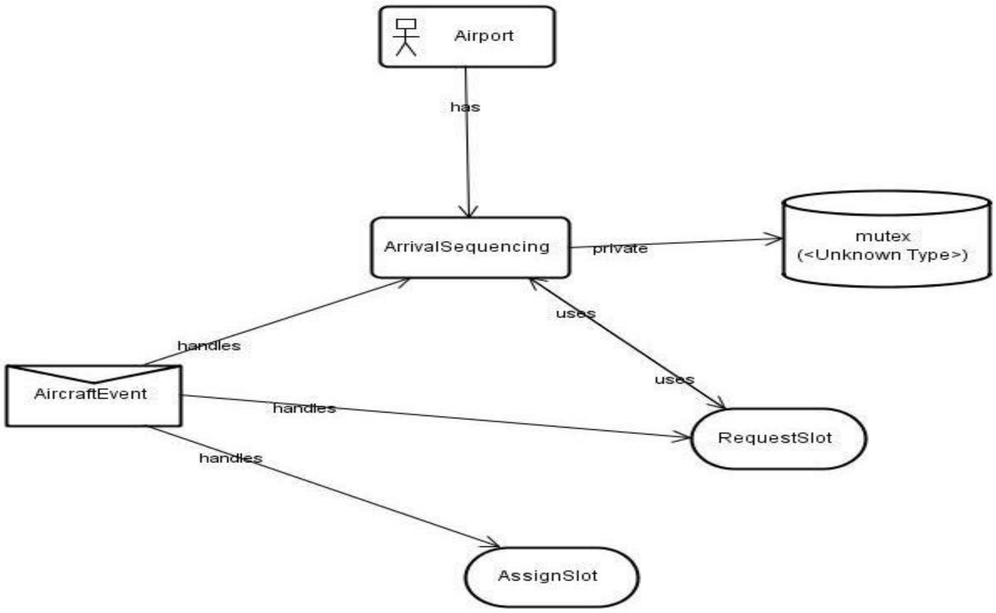


Figure 5.12: Overview diagram of airport agent structure

An overview of "ArrivalSequencing" interface: The "ArrivalSequencing" capability handles by the "AircraftEvenT". In Figure 5.13, shows overview of the capability "ArrivalSequencing" to handle events and use the mutex Semaphore for sequencing (mutual exclusion)

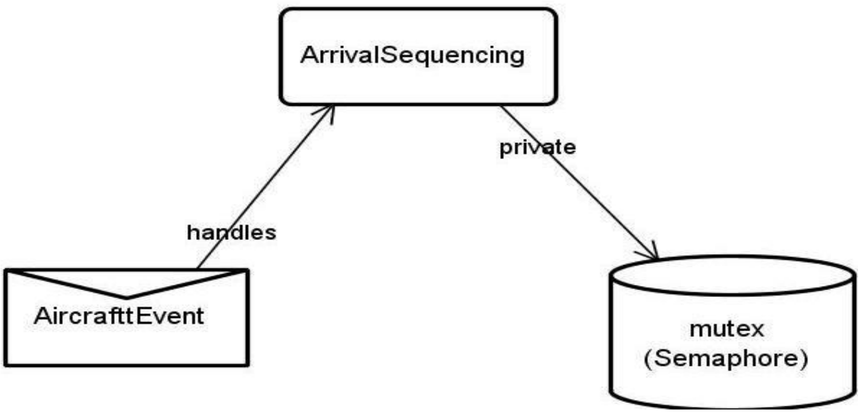


Figure 5.13: Arrival sequencing interface

An overview of "ArrivalSequencing" structure: The "ArrivalSequencing" capability contains the handling of landing requests from aircraft through negotiation/bidding with available runways for an appropriate landing allocation and uses the "Requestslot" plan types. In Figure 5.14, plan "RequestSlot" handles an "AircraftEvent" request by propagating it to all available runways, collecting all their suggestions, choosing the best one, and then notifying the runway and aircraft concerned.

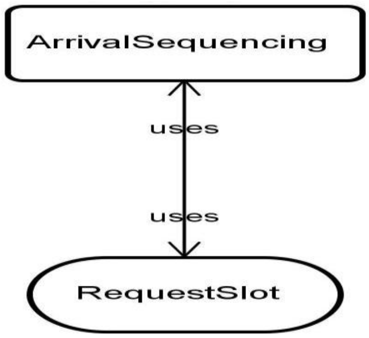


Figure 5.14: Arrival sequencing structure diagram

5.3.5 Communication: Event and Plan Structure

In Figure 5.15, shows the event and plan structure for landing an aircraft. The "AircraftEvent" is happens when aircraft enters into the control area the "Aircraft" agent interact to the other agents. This event is handling by the plans of various agents. The "MonitorAircraft" plan reads the landing information of the aircraft enters into the control area and sends to the "AircraftEvent".

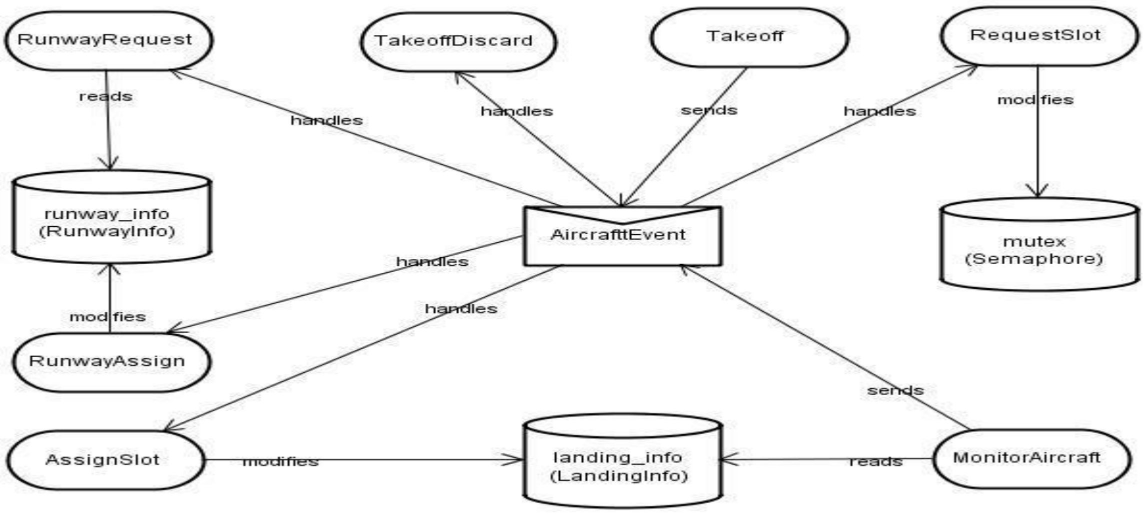


Figure 5.15: Aircraft Event, plans and data uses in the process of landing an aircraft

The "Takeoff" plan first send a booking request of the arrived aircraft to the "AircraftEvent". The "AircraftEvent" message the "Airport" agent to request the slot for landing. At the same time "Airport" agent send the request to the "Runway" agent for runway allocation. The "Runway" agent assign the runway on the basis of available runway information and send to the "Airport" agent. At last the "Airport" agent sends the final schedule to the "Aircraft" agent for landing.

In Figure 5.16, shows the "MonitorAircraft" plan monitor the arriving aircraft into the control area and approaching the "Airport" agent. The "Approaching" events handle by the "InitialApproach" plan (it progress an aircraft for landing when allocation is not provided) and "FollowApproach" plan (it progress an aircraft for landing when allocation is provided). These landing information sends to the "Aircraft" agent and wait until the further information is not received.

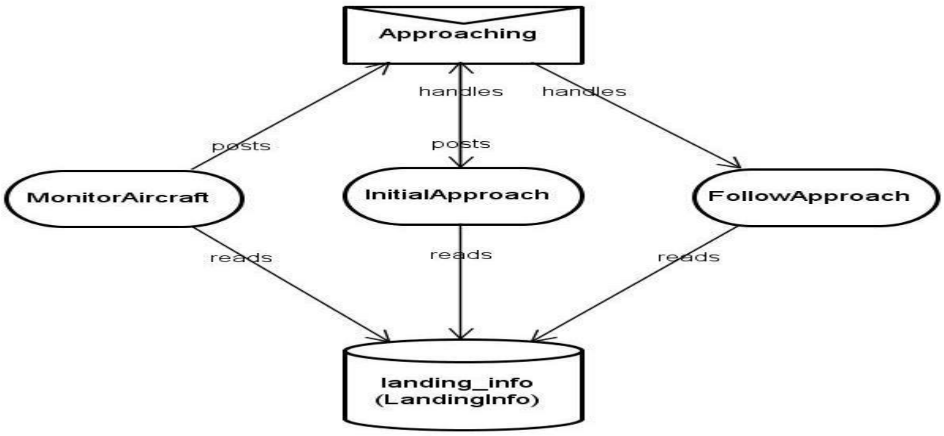


Figure 5.16: Approaching events, plans and data uses to process the action of landing

In Figure 5.17, an initial "TrafficEvent" schedule is posted at agent construction, and then "TrafficEvent" are posted for the schedule rows, this event is handled by the "Traffic and Takeoff" plan. The "Traffic" plan reads the initial schedule provide to the "Feeder" agent and and process the schedule for feeder. The "Takeoff" plan handles the schedule row for feeder, It first issues a booking request for the aircraft concerned, then it waits until the aircraft is in the destination airport's control area, at which time it constructs an "Aircraft" agent.

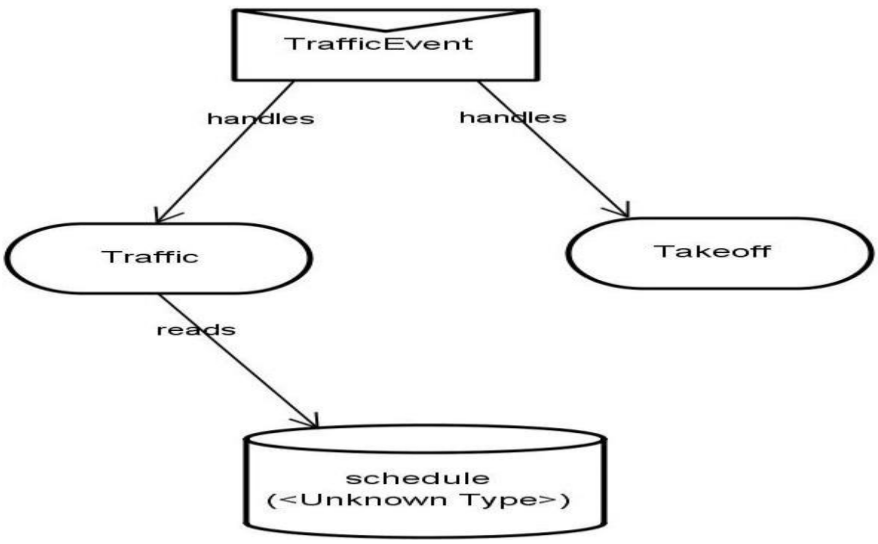


Figure 5.17: TrafficEvent, plans and data uses to schedule traffic

5.3.6 Aircraft Agent and Airport Agent Interaction Protocol

In Figure 5.18 shows the interaction protocol between the aircraft agent and airport agent. After creating the aircraft agent by the feeder agent when aircraft enters into the control area the aircraft agent send the proposal to the aircraft agent requesting the time slot for landing. The aircraft agent communicates the other agents and returns the proposal to the aircraft agent or wait for slot. The aircraft agent accepts the proposal of allotted slot and confirm to the airport agent. Before landing the aircraft airport informs the aircraft agent if any changes in the schedule otherwise inform final schedule and freeze it.

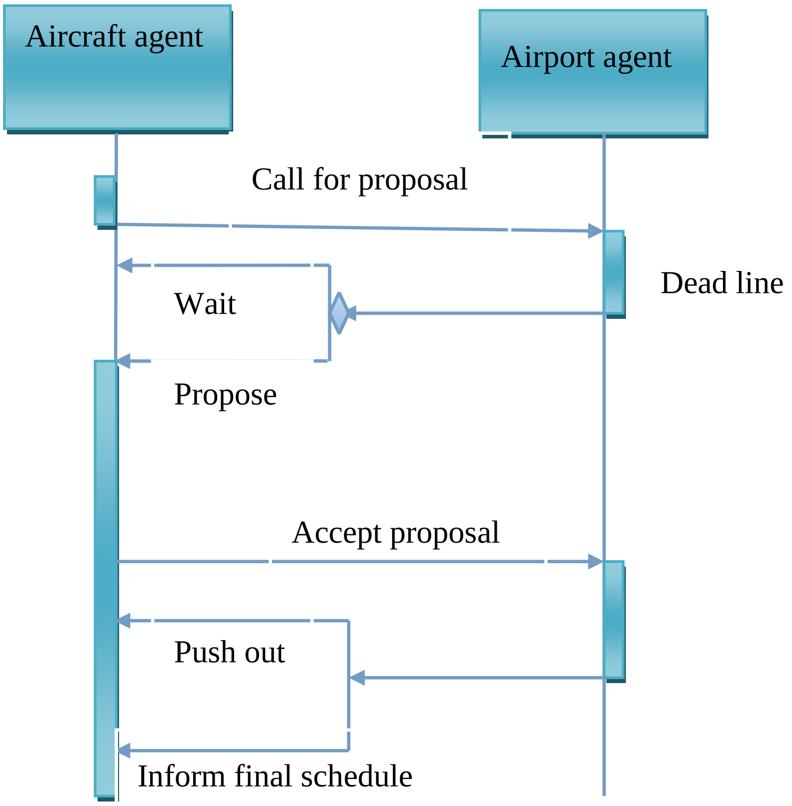


Figure 5.18: Aircraft agent and airport agent interaction protocol

5.3.7 Airport Agent and Runway Agent Interaction Protocol

In Figure 5.19 shows the interaction protocol between airport agent and runway agent. After receiving the aircraft agent proposal for landing the airport agent send the proposal to the runway agent to check the availability of the runway. Runway agent collects the information from the runway managers and checks the available slot to assign the runway. If vacant slot is not available check the least important allotted slot and push it. Otherwise it informs to the airport agent for hold the aircraft and waits the re-schedule slot and inform to the aircraft agent. After sometimes runway agent again check the availability of runway and re-scheduled it.

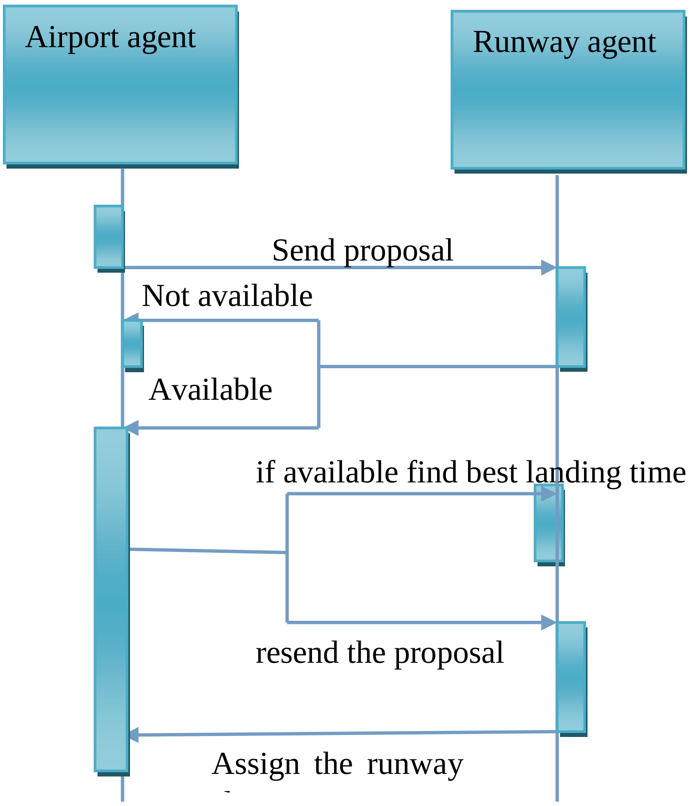


Figure 5.19: Airport agent and runway agent interaction protocol

5.4 Results and Analysis

This thesis has presented an MAS approach and method for aircraft runway sequencing and scheduling which is to be simple and dynamic and handles the complex air landing problems. The developed MAS framework is able to solve the single runway sequencing and scheduling problem under the given real time constraints. This developed MAS framework is tested on the JACK™ run window and it respond positively. The simulation is controlled by the time dilation over the real time clock initially it takes more time to compile and run the program to increase the speed of compilation change the dilation factor.

Chapter 6: Conclusion and Future Research

6.1 Conclusion

With the increase in globalization and increasing the air traffic day by day, increasing the capacity of the airport is very tough task to the air traffic management. The prime need of the air traffic management is to provide safe journey to the passenger. For safety concern, the separation is keeping to avoid the vortex due to the air pressure difference. To increasing the number of runway is not possible due to many constraints. To increase the capacity of the existing runway is the best solution. Most of the optimization techniques like general, heuristics and meta-heuristics are not well-suited in real world situations as they are complex in computation and inflexible. Now a day's multi-agent is the emerging field to handle the dynamic and complex situations. Keeping in this mind, this thesis addresses the research work aim at developing a Multi-agent System an intelligent planning and scheduling techniques, which can yield flexible, scalable, efficient, cost effective and reliable solution qualities in software development for air runway sequencing and scheduling problems. To find out the research gap in the existing literature, initially a detailed literature review on the air landing problems was carried out. Specific review on an application of MAS in the field of manufacturing and air traffic management is given in the chapter 2. The review of literature reveals that till now very little has done towards application of MAS to the air traffic management. Basic concepts regarding air runway problem described in chapter 3.

Motivated by the above mentioned facts, the proposed research work on this thesis aimed at the development of a Multi-Agent System (MAS) for air runway sequencing and scheduling. The complexity and approach overview are present in chapter 4. In the chapter 5, we developed the framework of the multi-agent system architecture on JACK™ java based agent software. Bidding/negotiation approaches are also developed in this chapter for obtaining the best landing time.

6.2 Future Research

The research presented in this thesis suggests a framework for air runway sequencing and scheduling problems. Yet large extensions are available for a more general problems situation indicating several important and promising areas of future research. The some main interesting area are given below for future extension-

- Apply the above MAS framework to solve the real-world data problems and compare the results to other optimizing techniques.

- Consider the takeoff situation along with the landing and both the situation handled parallel.

- Extend the framework for more than one runway problem.

Ansola, P. G., Higuera, A. G., Jose Pastor, M., & Otamendi, F. J. (2011). Agent-based decision-making process in airport ground handling management. *Logist. Res.*, *3*, 133–143.

Atkin, J. A. D., Burke, E. K., Greenwood, J. S., & Reeson, D. (2008). On-line decision support for take-off runway scheduling with uncertain taxi times at London Heathrow airport. *Journal of Scheduling*, *11*, 323–346.

Baker, A. D. (1998). A Survey of Factory Control Algorithms That Can Be Implemented in a Multi-Agent Heterarchy : Dispatching , Scheduling , and Pull. *Journal of Manufacturing System*, *17*(4), 297–320.

Balakrishnan, H. (2006). Scheduling Aircraft Landings under Constrained Position Shifting. *AIAA Guidance, Navigation, and Control Conference and Exhibit 21 - 24 August 2006, Keystone, Colorado*, (August), 1–23.

Bayen, A. M., & Tomlin, C. J. (2004). An approximation algorithm for scheduling aircraft with holding time. In *43rd IEEE Conference on Decision and Control (CDC)* (Vol. 3, pp. 2760–2767). IEEE.

Beasley, J. E., Sonander, J., & Havelock, P. (2001). Scheduling aircraft landings at London Heathrow using a population heuristic. *The Journal of the Operational Research Society*, *52*(5), 483–493.

Bellifemine, F., Caire, G., Poggi, A., & Rimassa, G. (2008). JADE: A software framework for developing multi-agent applications. Lessons learned. *Information and Software Technology*, *50*(1-2), 10–21.

Bencheikh, G., Boukachour, J., & Alaoui, A. (2011). Improved ant colony algorithm to solve the aircraft landing problem. *International Journal of Computer Theor and Engineering*, *3*, 1793–8201.

Bennell, J. a., Mesgarpour, M., & Potts, C. N. (2011). Invited review. Airport runway scheduling. *4OR-Q J Oper Res*, *9*, 115–138.

Bianco, L., Dell, P., & Giordani, S. (1999). Minimizing total completion time subject to release dates and sequence-dependent processing times. *Annals of Operations Research*, *86*, 393–415.

Brentnall, A. R., & Cheng, R. C. H. (2009). Some effects of aircraft arrival sequence algorithms. *Journal of the Operational Research Society*, *60*, 962–972.

Brinton, C. R. (1992). An implicit enumeration algorithm for arrival aircraft. In *[1992] Proceedings IEEE/AIAA 11th Digital Avionics Systems Conference* (pp. 268–274). IEEE.

Capri, S., & Ignaccolo, M. (2004). Genetic algorithms for solving the aircraft-sequencing problem : the introduction of departures into the dynamic model. *Journal of Air Transport Management*, *10*, 345–351.

Chandran, B., & Balakrishnan, H. (2007). A Dynamic Programming Algorithm for Robust Runway Scheduling. *Proceedings of the 2007 American Control Conference Marriott Marquis Hotel at Times Square New York City, USA*, 1161–1166.

Cheng, T. C. E., Chung, Y., Liao, S., & Lee, W. (2013). Two-agent singe-machine scheduling with release times to minimize the total weighted completion time. *Computers and Operation Research*, *40*, 353–361.

Cheng, T. C. E., Ng, C. T., & Yuan, J. J. (2006). Multi-agent scheduling on a single machine to minimize total weighted number of tardy jobs. *Theoretical Computer Science*, *362*, 273–281.

Cheng, V. H. L., Crawford, L. S., & Menon, P. K. (1999). Air traffic control using genetic search techniques. In *Proceedings of the 1999 IEEE International Conference on Control Applications (Cat. No.99CH36328)* (Vol. 1, pp. 249–254). IEEE.

Ciesielski, V., & Scerri, P. (1998). Real Time Genetic Scheduling of Aircraft Landing Times. *IEEE*, *98*(9), 4869–7803.

De Neufville, R., & Odoni, A. (2003). *AIRPORT SYSTEMS. PLANNING, DESIGN AND MANAGEMENT*. *Mac-graw hill*.

Duan, L., Doğru, M. K., Özen, U., & Beck, J. C. (2011). A negotiation framework for linked combinatorial optimization problems. *Autonomous Agents and Multi-Agent Systems*, *25*(1), 158–182.

Erol, R., Sahin, C., Baykasoglu, A., & Kaplanoglu, V. (2012). A multi-agent based approach to dynamic scheduling of machines and automated guided vehicles in manufacturing systems. *Applied Soft Computing*, *12*, 1720–1732.

Farhadi, F., Ghoniem, A., & Al-salem, M. (2014). Runway capacity management – An empirical study with application to Doha International Airport. *Transportation Research Part E*, *68*, 53–63.

Filar, J. A., Manyem, P., & White, K. (2001). How Airlines and Airports Recover from Schedule Perturbations: A Survey. *Annals of Operations Research*, *108*(1-4), 315–333.

Glen Stevens. (1995). *An Approach To Scheduling Aircraft Landing Times Using Genetic Algorithms*.

Hill, J. C., Archibald, J. K., Johnson, F. R., Stirling, W. C., & Frost, R. L. (2005). A Cooperative Multi-Agent Approach to Free Flight. In *AAMAS"05,Utrecht, Netherland* (pp. 1083–1090).

Huang, Z., Song, X., Sun, J., & Li, Z. (2012). Aircraft Landing Scheduling Based on Semantic Agent Negotiation Mechanism. *Springer-Verlag Berlin Heidelberg*, (2), 483–489.

J. E. Beasley, M. Krishnamoorthy, Y. M. Sharaiha, D. A. (2000). Scheduling Aircraft Landings — The Static Case. *Transportation Science*, *34*(2), 180–197.

Jungai, T., & Hongjun, X. (2012). Optimizing Arrival Flight Delay Scheduling Based on Simulated Annealing Algorithm. *Physics Procedia*, *33*, 348–353.

Kaplanoglu, V. (2014). Multi-agent based approach for single machine scheduling with sequence-dependent setup times and machine maintenance. *Applied Soft Computing*, *23*, 165–179.

Lee, H., Balakrishnan, H., & Constraints, A. (2008). Fuel Cost , Delay and Throughput Tradeoffs in Runway Scheduling. *2008 American Control Conference Westin Seattle Hotel, Seattle, Washington, USA June 11-13, 2008*, 2449–2454.

Lee, W., Chen, S., Chen, C., & Ã, C. W. (2011). A two-machine flowshop problem with two agents. *Computers and Operation Research*, *38*, 98–104.

Leitao, P. (2009). Engineering Applications of Artificial Intelligence Agent-based distributed manufacturing control : A state-of-the-art survey. *Engineering Applications of Artificial Intelligence*, *22*, 979–991.

Lou, P., Liu, Q., Zhou, Z., Wang, H., & Sun, S. X. (2012). Multi-agent-based proactive – reactive scheduling for a job shop, 311–324.

Lucio Bianco, Paolo Dell'Olmo, S. G. (1997). Modelling and Simulation in Air Traffic Management. *Modelling and Simulation in Air Traffic Management*, 139–167.

Luck, M., McBurney, P., Shehory, O., & Willmott, S. (2005). Agent technology: computing as interaction (a roadmap for agent based computing). *AgentLink III*, 33–35.

Mahmoudian, M., Aminnayeri, M., & Ph, A. M. (2013). Aircraft landing scheduling based on unavailability of runway constaint through a time segment heuristic method. *International Journal of Conception on Management and Social Sciences*, *1*(2), 31–36.

Man, L. (2014). An agent-based approach to automated merge 4D arrival trajectories in busy terminal maneuvering area. *Procedia Engineering*, *00*, 000–000.

Mor, B., & Mosheiov, G. (2010). Scheduling problems with two competing agents to minimize minmax and minsum earliness measures. *European Journal of Operational Research*, *206*, 540–546.

Mors, A. ter, Mao, X., Roos, N., Witteveen, C., & Salden, A. (2007). Multi-Agent System Support for Scheduling Aircraft De-icing. In *Proceedings of the 4th International ISCRAM Conference Delft, the Netherlands*.

Paolo Busetta, Ralph Rönnquist, Andrew Hodgson, and A. L. (1999). JACK Intelligent Agents - Components for Intelligent Agents in Java. *Agentlink News Letter*, *2*(1), 2–5.

Park, Y., Kim, S., & Lee, Y. (2000). Scheduling jobs on parallel machines applying neural network and heuristic rules. *Computers & Industrial Engineering*, *38*, 189–202.

Randall, M. (2002). Scheduling aircraft landings using ant colony optimisation. In *6th IASTED International Conference Artificial Intelligence and Soft computing Banff,Alberta, Canada* (pp. 129–133).

Russell, S. J., & Norvig, P. (1995). Artificial Intelligence A Modern Approach. In *Prentice Hall, Englewood Cliffs, New Jersey 07632* (Vol. 13, pp. 31–49).

Searle, J. R. (1969). *Speech Acts: An Essay in the Philosophy of Language* (Vol. 0). Cambridge University Press.

Shen, W., & Norrie, D. H. (1999). Agent-Based Systems for Intelligent Manufacturing : A State-of-the-Art Survey. *Knowledge and Information Systems*, *1*, 129–156.

Smith, R. G. (1980). Communication and Control in a Distributed Problem Solver. *IEEE Transactions on Computers*, *C*(12), 1104–1113.

Sölveling, G., & Clarke, J. P. (2014). Scheduling of airport runway operations using stochastic branch and bound methods. *Transportation Research Part C*, *45*, 119–137.

Soomer, M. J., & Franx, G. J. (2008). Scheduling aircraft landings using airlines ' preferences. *European Journal of Operational Research*, *190*, 277–291.

Wooldridge, M. (1997). Agent-based software engineering. *IEE Proceedings - Software Engineering*, *144*(1), 26.

Wooldridge, M. (2009). *An Introduction to MultiAgent Systems*. John Wiley & Sons.

Wooldridge, M., & Jennings, N. R. (1995). Intelligent agents: theory and practice. *The Knowledge Engineering Review*, *10*(02), 115.

Xiangwei, M. X. M., Ping, Z. P. Z., & Chunjin, L. C. L. (2010). Aircraft category based genetic algorithm for aircraft arrival sequencing and scheduling. *Control Conference (CCC), 2010 29th Chinese*, 5188–5193.

Xiao-Bing Hu, W.-H. chen. (2005). Receding Horizon Control for Aircraft Arrival Sequencing and Scheduling. *IEEE Transactions on Intelligent Transportation Systems*, *6*(2), 189–197.

Xiao-rong, F., Xing-jie, F., & Zhao-rui. (2014). Using The Heuristic Genetic Algorithm in Multi-runway Aircraft Landing Scheduling. *Indonesian Journal of Electrical Engineering*, *12*(3), 2203–2211.

Appendix A

Name	: Virendra Kumar Verma
Date and Place of Birth	: 15th September 1990, Ambedkar Nagar (Uttar Pradesh), India

Academics

2015	: Post Graduation in Industrial Engineering, from Malaviya National Institute of technology (MNIT) Jaipur, India
2011	: Bachelor of Engineering in Mechanical Engineering, from ITM Gorakhpur, India

Research Interests:

Multi-agent System, Agent based modeling, Project Management and Scheduling, Artificial Intelligence, Optimization, Supply Chain Management.

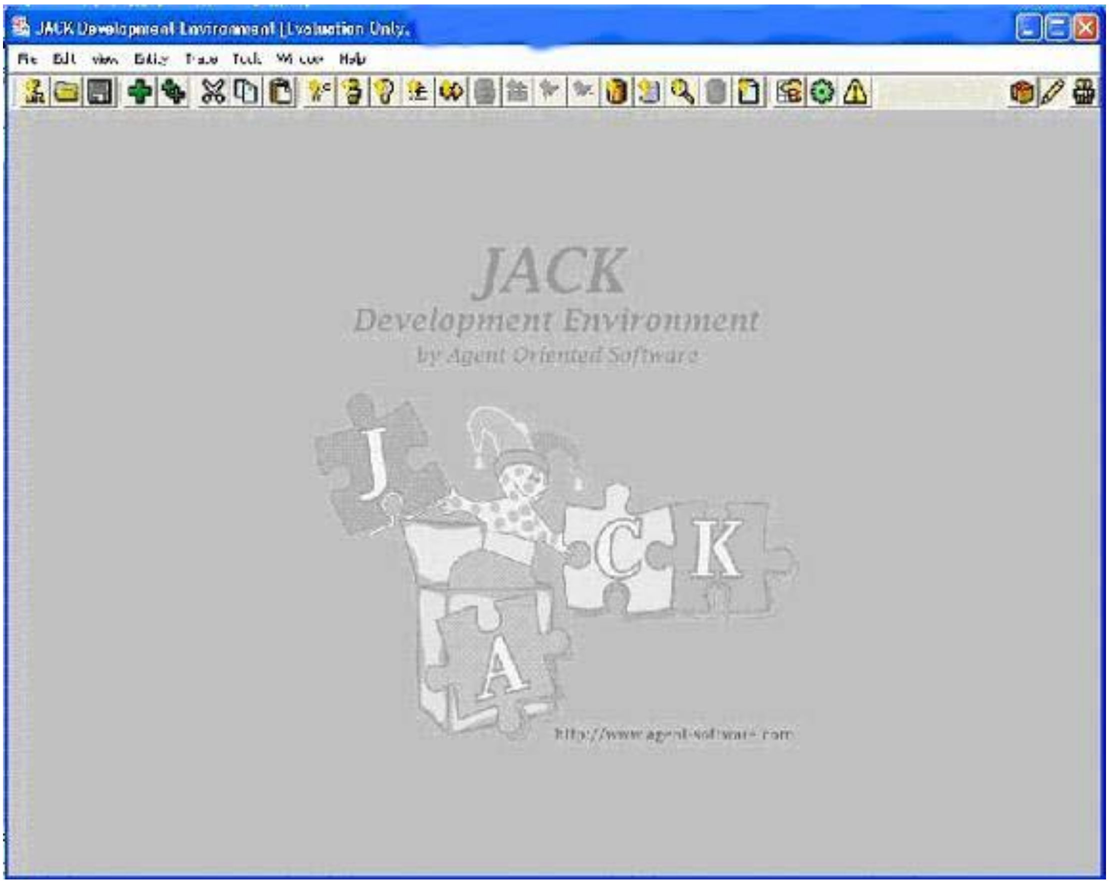


Figure B.1: Snapshot of JACK™ agent software