DOI:10.24996/ijs.2018.59.2C.17

ISSN: 0067-2904

Proposed Method of Air Traffic Routing in Dynamic Environment Usiing Modifed rrt With Collision Avoid Nace

Alia Karim Abdul Hassan , Sheelan Waad Adwaan*
Department of Computer Science, University of Technology, Baghdad, Iraq

Abstract

In the current Airlines Air Traffic Management (ATM), the Air Traffic Control Operators (ATCO), with the Air Traffic Control systems (ATC), operate air traffic paths with a small number of fixed routes. Problems of fixed routes appear such that even when they reduce the chance for conflict, they also produce flight paths plans that do not reduce flight time or fuel usage. In nowadays Airlines are heavily overloaded and anxious to minimize the aircrafts fuel usage costs, increase airplanes, and optimize the paths or the flight routes in order to find the most optimal suitable flight paths. This paper presents a new path planning method that deal with such problem effectively, the idea was to produce flights routes by allowing the ATCO, or the pilot to select better fuel-efficient routes with reducing flight time. This work has been done by using modified 'Rapidly Exploring Random Tree' (RRT) path planning algorithm in dynamic cluttered environment with collision avoidance. The experiment of the developed algorithm simulation experiment shows promising result for future research.

Keywords - Air Traffic ,ATM, ATOC, Robot, Agent, RRT Algorithm.

مقترح طريقه لأداره الزحمة الجوية في بيئة متحركة مع تجنب التصادم باستخدام RRT المعدلة

علياء كريم عبد الحسن، شيلان وعد عدوان*
قسم علوم الحاسوب، الجامعة التكنولوجية، بغداد، العراق

الخلاصة

في ادارة الزحمة الجوية للخطوط الجوية الحالية، فأن العاملين على سيطرة الزحمة الجوية، يشغلون مسارات الزحمة الجوية مع عدد صغير من الطرق الثابتة. مشاكل الطرق الثابتة على الرغم من انها تقلل من احتمالية التصادم، فأنها تنتج مسارات رحلات لا تقلل من كمية الوقت او الوقود المستهلك . في هذه الايام الخطوط الجوية مكتظة ،مثقله ومتعبة من محاولات التقليل من كلفة استهلاك الوقود، زيادة استخدام المطارات، تقليل مسار الرحلة وايجاد افضل اقصر مسار. هذا البحث يقدم طريقة تخطيط مسار جديده والتي تتعامل مع هكذا مشاكل، الفكرة هي التلاعب بمسار الطائرة بطريقة تسمح للعاملين على إدارة سيطرة الزحمة الجوية او الطيار من من اختيار المسار بأقل كلفة و وقود مع التقليل من وقت الطيران. هذا العمل تم بواسطة توظيف خوارزمية (شجرة الاكتشاف العشوائية السريعة RRT) المعدلة لتخطيط المسار في بيئة متحركة ومضببة مع تجنب العوائق. النتائج المختبرية للنظام المحاكي للطريقة المطورة اظهرت نتائج واعدة للأبحاث المستقبلية.

*Email: waad.sheelan@yahoo.com

1. Introduction

The free flight concept is related to the family of the proposed ATC schemes, in which ATCO in airline companies, a crow or pilots would be allowed to choose path route with a great flexibility from one station (airport) to another. Air traffic is highly crowded and increase daily, in fact, the aviation industry planned for widely growth worldwide[1].

In nowadays, ATCO is overloaded with responsibilities and trying to work with the limits of their abilities, therefor the expected increasing of the air traffic intensity will eventually exceed the ability of the existing airlines ATC systems. In such situations, due to the very dynamic , unpredictable nature of airspace environment, ATCO might not have the ability to observe and control the situation [2].
Free flight is also the solution to the process of decision making for choosing route to be allowed not only by the ATCO, but also for the pilot cockpit by making flight decisions routing based on the dynamic information obtained [3]. The basic idea is based on employing Artificial Intelligent (IA) [4] approaches of robots motion path planning methods with an agents robots concept to optimize the flight routes of finding the most optimal suitable path .i.e. the complete path is generated depending on a group of dynamic[5] information.

As the existing of different robot motion planning algorithms that provides dynamic inputs the ‘Rapidly-exploring Random Tree’ (RRT) algorithm is one of the well-known algorithm in such aspects, and by using RRT, it will provide an end-to-end routing, and collision avoidance from the very take-off (source) to the landing (destination) [6]. However, RRT is utilized for solving problems of motion robot navigation. It has been successful at finding feasible solutions for many types of problems.

The purpose of agent free flight concept procedure is to ensure safety and also to improve the abilities of ATM system processing in the airspace, and the procedure of flight route designing is scientifically planning and searching, and reliable framework for the landing and takeoff air routes [7]. This paper, will firstly introduce the ATM, ATC and ATCO (section1), traditional RRT (section3) and the modified RRT algorithm (section4). Secondly describes the proposed method of enhancements that have been explored (section5), and then provides this work by experimental result (section6). The paper concludes with some insights and promising future research directions in (ATC) systems discovered through this work (section7).

2. Related work

In the recent years the problems of ATM have been studying, and searching to find new enhanced ATM problem solving methods . A lot of methods are acquired over the years, and below will illustrate few of them.

‘LaValle’ 1998 was the first one who introduce the basic RRT algorithm and use it in path planning. In this paper, the author efficiently covered the whole space of RRT algorithm and how it’s probabilistic worked [6].

‘Karaman’ and ‘Frazzoli’ in 2011 enhanced the RRT algorithm, by introducing a RRT* that changed the length of the generated path and reduced it by allowing algorithm rewiring of the connections path of the tree . RRT* is asymptotically optimal, but its convergence is slow especially in large environments [7].

In Lagrangian method, it proposed to take into accounts the dynamics of each one of the airplanes, in order to avoid crashing during navigation, also, in Eulerian work, which suggest that only the total flow of patterns of the airplanes being modeled [8]. Agogino method, also adapts free flight and agent concept, where [9] it uses an agent flight algorithm to minimize the overcrowding. Also [10] Jardin start to work with optimal airplanes by solving the problem of ATM by using an indirect method and conceder it an optimal control problem. Sethian and Valdirminsky developed a new methodological for free flight path planning, that call the ‘Order Upwind’ [11]. As well as, Alton employs the methodological approach along with Semi-Lagrangian approach for ensuring complete trajectories for planning paths [12].

Sislak’s work suggested that an individual agent does not handle one airplane, an individual is cooperative with a specific (rigid) location to minimize the crowding around that specific rigid location, it also studied the decentralized [13] ATM planning , and ATC in order to offer much capability utilization of the available environment of airspace, in additional to improve supporting of re-planning and collision preventing.

In all cases, the generation of optimization from the resultant models is sophisticated work, and much complexity minimization is required being into account. These models are useful for a wide range of issues, from collision avoidance to traffic minimization[14]. Although, the majorly objective in this paper focuses on how to manage the air traffic. The method proposed in this work can go much further by including airports. Eventually, the results are generated by the RRT algorithm modification. Besides, searching for, an unfixed route for AT path planning, it has benefited in alleviating the effect of the airport failing ability of capacities, and it helps to make the entire performance more responsive to unpredicted changes in system abilities.

3. Traditional RRT and Modified RRT

The RRT is a path planning and data structure algorithm which it designed to search efficiently for paths in a high-dimensional workspaces [6]. It is an incremental searching algorithm which providing benefits over the conventional roadmap[15] planners because of its inherent feasibility for the solutions generating.

RRTs construct incrementally via expanding the specific tree to a randomly-sampling of point in the workspace, while it is satisfying the given constraints, such as including obstacles or a dynamic constraints.

The RRT in Figure-1 starts with tree Ʈ which contain at first the root beginning point (q_{init}) only. After that, it frequently began adding points to the tree Ʈ by an expanding operation EXTEND(T, q_{rand}) , RRT generates new points by a randomly sampling generator and detect the nearest closer neighbor with tree Ʈ (q_{near}), then RRT executes the expanding procedure to the selected nearest neighbor and selects a new random point(q_{new}) , then it continues until it reaches to a specific point counting (N) the problem has been solved, or it was connected (reached) a specific goal point , based on the difference [6] .

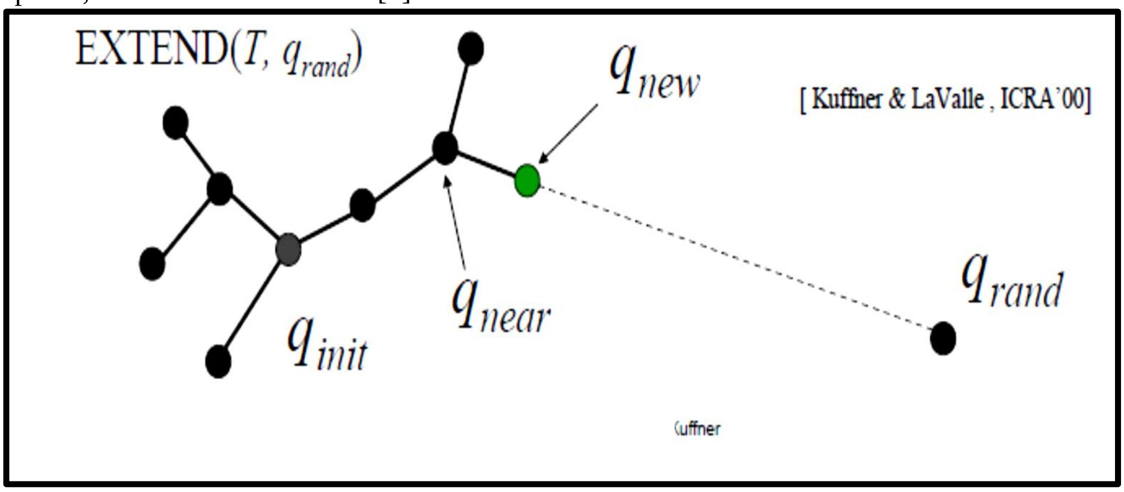

Figure1-RRT expansion tree.

The traditional RRT is able to cover the whole workspace, (which is 2D dynamic ATM airspace) very efficiently and quickly. It provides an efficient method for solving planning problems and effectively find a feasible path, thus to be a probabilistically complete, algorithm (1) depicts the basic RRT [1].

Algorithm 1: The Basic RRT Algorithm

Input: an W workspace environment, a root position R_{root}, number *H* nodes, a step size α_R, metric of distance *DM*.

Output: tree *Ʈ* having *N* points rooted at R_{root},

Step 1: *Ʈ* =r_{root},

Step 2: While number of nodes (*Ʈ*) < *H* do // till stopping condition reached

Step3: R_{rand} = random CFG() // picking points randomly from the configuration space graph (CFG)

Step 4: R_{near}=nearest neighbor (R_{rand}, *Ʈ*, *DM*) // choosing the nearest neighbors points to the picked random point

Step 5: R_{new}=expand(R_{near}, R_{rand}, α_R) // picking the most nearest point from q nearest

Step 6: If is valid (R_{new}) then // if the new point is accepted
1. $\mathcal{T}$. Add node(R_{new}) // add it to the path
2. $\mathcal{T}$. add edge(: R_{near}, R_{new})
Step 7: return $\mathcal{T}$

The modified RRT algorithm changes the basic RRT expansion directions by making it expand only toward the goal by using a heuristic function. It calculates the heuristic value for both the current with the goal points, and for each produced new random point with the goal point , if the heuristic value of the new point is less than the current, it will rejects and searches for new random point, otherwise accepts and adds the new generated random point, this is done by the following :

1- Calculating the Euclidean distance between the current point and the goal point by using equ.1 :

$$D\ current = \sqrt{(x_{goal} - x_{currentl})^2 + (y_{goal} - y_{urrent})^2} \quad \ldots\ldots (1)$$

2- Calculating the Euclidean distance between each new generated random point and the goal point by using equ.2 :

$$D\ new = \sqrt{(x_{goal} - x_{new})^2 + (y_{goal} - y_{new})^2} \quad \ldots\ldots\ldots\ldots\ldots\ldots (2)$$

3- For each modified RRT iteration $D\ current$ is updated ,and $D\ new$, then checks if $D\ new$ is less than $D\ current$, hence rejects and searches for new generated random point otherwise accepts and adds to path, the directed RRT algorithm is shown, as in the following:

Algorithm 2: Modified RRT Algorithm

Input: an W workspace environment, a root position R_{root}, number *H* nodes, a step size α_R, metric of distance *DM*.

Output: tree $\mathcal{T}$ having *N* points rooted at R_{root},

Step 1: $\mathcal{T}$ =r_{root},
Step 2: $D_{current}$=distance(R_{root}, R_{goal}) // calculating distance from start point to the current
Step 3: While number of nodes ($\mathcal{T}$) < H do // till terminal condition is reached
Step 4: R_{rand} = random CFG( ) // picking points randomly from the configuration space graph (CFG)

Step 5: R_{near}=nearest neighbor (R_{rand}, $\mathcal{T}$, *DM*) // choosing the nearest neighbors points to the picked random point

Step 6: R_{new}=expand($R_{near}, R_{rand}, \alpha_R$) // picking the most nearest point from q nearest
Step 7: D_{new} =distance(R_{new}, R_{goal}) // calculating distance from the new random point to the goal point
Step 8: If ($D_{new} \geq D_{current}$) then // if the new distance is less than the current distance then add to path
1. $\mathcal{T}$. add node(R_{new})
2. $\mathcal{T}$. add edge(R_{near}, R_{new})
Step 9: Otherwise return to step 4
Step 10: Update $D_{current}$= D_{new}
Step 11: return $\mathcal{T}$

4. The Proposed Method

This paper introduces a new path planning method for ATM systems routing and control (flight routing). the idea is about dealing with an individual flight (aircraft) as single robot (Agent), and carry the task of finding path for each agent from the source airport (as a start point) to the destination airport (as a goal point) in dynamic and cluttered environment (free flight space). To accomplish this task, the dynamic modified RRT has been employed to find nearest optimal path along with collision avoidance technique, and obtains the shortest flight route that is unfixed, more suitable, and less costing time and fuel while keeping air traffic safety matters into consideration, the dynamic modified RRT algorithm pseudo code, is shown in algorithm(3).

Algorithm 3: Dynamic Modified RRT Method

Input: an W workspace environment, a root position R_{root}, number *H* nodes, a step size α_R, metric of distance *DM*.

Output: tree $\mathcal{T}$ having N points rooted at R_{root},
Step 1: $\mathcal{T}$ =r_{root},
Step 2: $D_{current}$=distance(R_{root}, R_{goal}) // calculating distance from start point to the current
Step 3: While number of nodes ($\mathcal{T}$) < H do //till terminal condition is reached
Step 4:check value =false
Step 5: while(check value ≠true)
Step 6: R_{rand}= random CFG( ) //picking points randomly from the configuration space graph (CFG)
Step 7: check collision ($object_{OB}$, R_{rand}) //function for checking and avoidance obstacle
Step 8: if valid (check collision)
1: check value=true and (end while) otherwise
.2: return to step 6
Step 9: R_{near}=nearest neighbor (R_{rand},Ţ,DM) //choosing the nearest neighbors points to the picked random point

Step 10: R_{new}=expand(R_{near}, R_{rand}, α_R) // picking the most nearest point from q nearest
Step 11: D_{new} =distance(R_{new}, R_{goal}) // calculating distance from the new random point to the goal point
Step 12: If ($D_{new} \geq D_{current}$) then // if the new distance is less than the current distance then add to path
1.Ţ. add node(R_{new})
2.Ţ. add edge(R_{near}, R_{new})
Step 13: Otherwise return to step 4
Step 14: Update $D_{current}$ = D_{new}
Step 15: return $\mathcal{T}$

5. The Simulation Approach

The proposed dynamic modified RRT algorithm was simulated using MATLAB programming language in a 2D workspace and run on a laptop with inter® Pentium® CPU P6200 @ 2.13GHz 2.13GHz,4.00GB memory, and windows 7 ultimate.

The execution of the proposed method is a simulation to the dynamic free flying air traffic environment of 2D in W of (X,Y) with coordination (0,0),(400,400), and initial state at (200,200) which it represents the plane take off point (source), for example (Baghdad international airport), the goal state at (400,400) which it represents the plane landing point (destination), which is the point when it passes over the air managing coverage range for that country and heading to access the destination coverage range . The task of the dynamic modified RRT is to find the nearest most optimal path from the source (start state) to the destination (goal state), the best free flying air traffic route from takeoff point till landing as shown in Figure-2 .

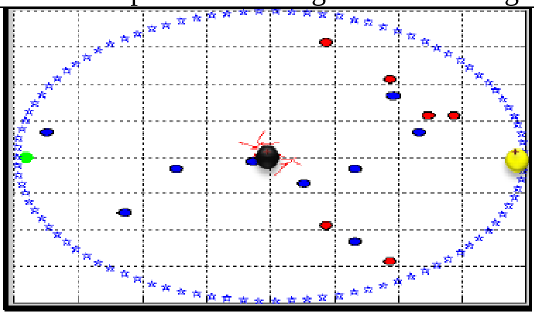
Figure-2(a)

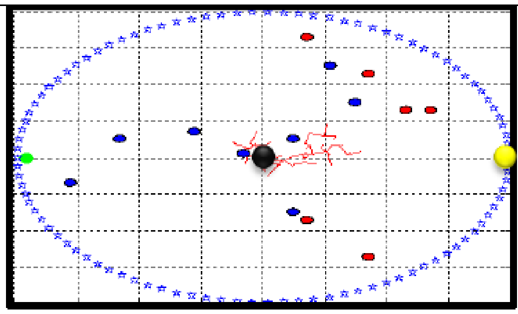
Figure-2(b)

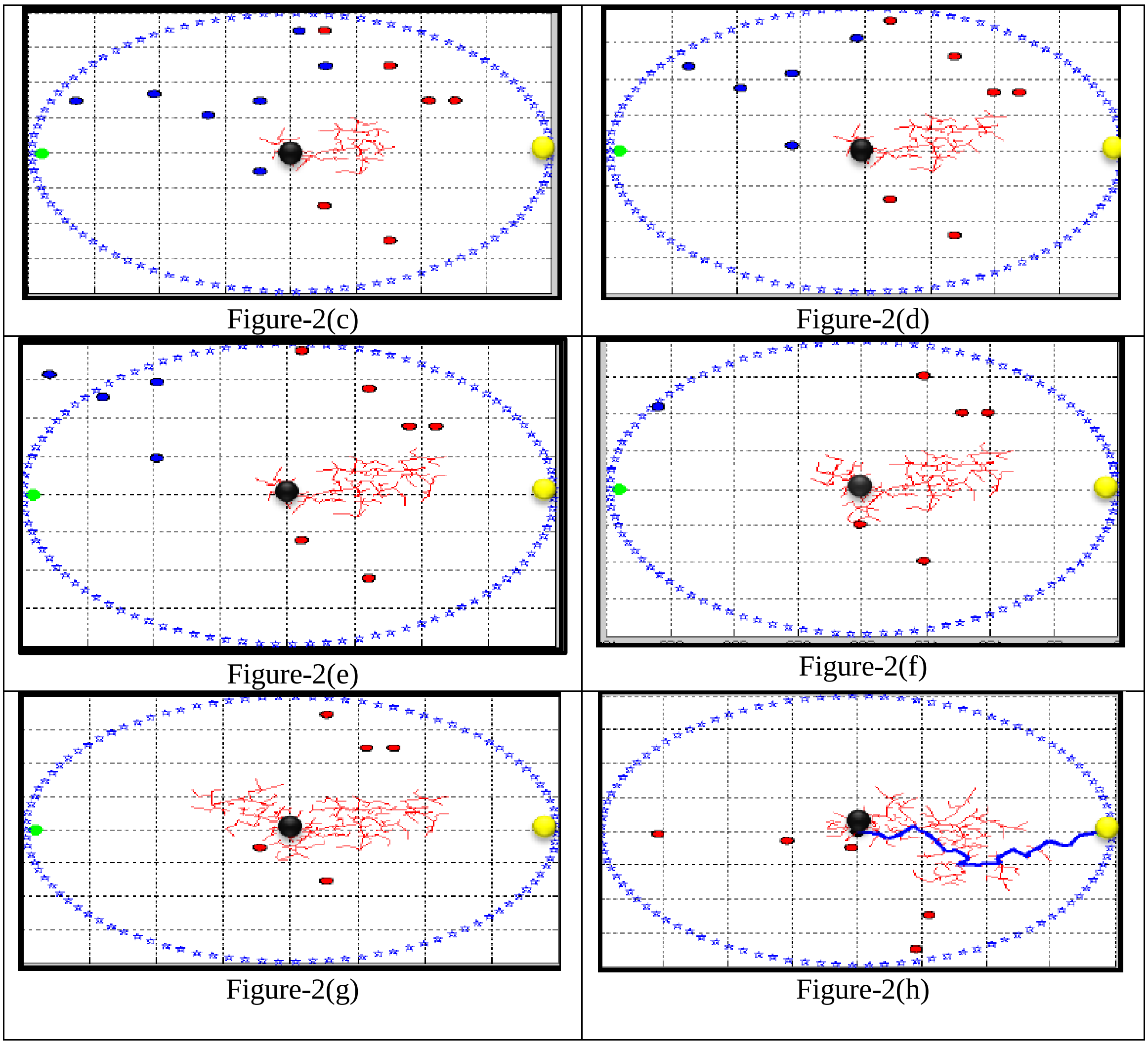
Figure-2(c) Figure-2(d)
Figure-2(e) Figure-2(f)
Figure-2(g) Figure-2(h)

Figure 2-execution of path planning in (AT) dynamic environment using modified RRT with collision avoidance.

Figure-2 shows the dynamic air traffic W where the dark dots represents the start point (which is the airport that the aircraft "flying agent" takeoff from), and the yellow dots (which is the point it passes the coverage range of (ATC) systems that is noted by the stars blue cycle, or the airport that the aircraft landing on it), the collared dots represents dynamic moving aircraft (obstacles) that passes through the coverage range of (ATC) systems, as it show in Figure-2(a) to Figure-2(g) The modified RRT is searching in directed technique with dynamic moving environment for the most suitable path for the aircraft "flying agent" to fly and pass the air traffic, and reach the desired destination as it show in Figure-2(h).

6. The Experimental Result

This experiment projects a new highly promising result for finding most optimal path routs of free flying agent (an aircraft) in dynamic air traffic environment and by using modified RRT algorithm for searching, and avoiding moving obstacle which is (the aircraft that passes through ATC coverage range) with taking into consideration two execution measuring units time(seconds), and length (units). In this experiment the path length from the source point till the desired destination is (428.9023 – in length units), and the execution costing time is (352.9393- in second), the resulted path shown to be short and near optimal which is what the (ATCO) highly required to reduce their fuel costs and increases aircraft utilization and optimizes the flight paths in the matters of finding the most optimal suitable path, and also it seemed to be effectively succeeded to avoid moving object without collision, which is very promising improvement in ATM systems.

7. Conclusions

The proposed modified RRT intended to reduce the number of expanding nodes, thus reducing the data structure memory storage, also reduces the amount of searching time till it finds a goal, it shows a promising result in dynamic environment, and worked effectively in collision avoidance with moving obstacle. The major problem of ATM systems was to obtain flight route that is safe, more suitable, and less consuming of time and fuel with taking into consideration passing through the crowding dynamic environment. However, this was possible to be accomplished via projected modified RRT algorithm that carried out the task of searching, and finding the nearest most optimal path (flight route) for "flying agent" with avoiding crashing during moving obstacle till reaching to the destination successfully. The proposed method seems to have very effective results, and preamble the way to new generation of ATM system control and path planning.

References

1. Vijayan, V. P., John, D., Thomas, M., Maliackal, N. V. and Vargheese, S. S. **2009.** Multi Agent Path Planning Approach to Dynamic Free Flight Environment. Rajagiri School of Engineering and Technology, Cochin, India, *International Journal of Recent Trends in Engineering*. **1**(1).
2. Pechoucek, M., Sislak, D. and Payne, T. R. *Agent-Based Approach to Free-Flight Planning, Control, and Simulation*. Czech Technical University, University of South hampton .
3. Vijayan, V. P., John, D., Thomas, M., Maliackal, N. and VRajagiri, S. **2009.** *School of Engineering and Technology,* Cochin, India. *International Journal of Recent Trends in Engineering*. **1**(1).
4. Abdul Hassan, A.K. **2014.** Path Planning Method for Single Mobile Robot in Dynamic Environment Based on Artificial Fish Swarm Algorithm. Computer Science Depart, University of Technology/ Baghdad . *Eng. & tech. Journal* . **32**, part (B) (2).
5. Jaleel, T. A. and Abdul Hassan, A.K. **2016.** Collision Avoidance Using Cat Swarm Algorithm for Multi Mobile Robot Path Planning in Dynamic Environment. *Iraqi Journal of Science*.
6. LaValle, L. and Steven, M. **1998.** Rapidly-Exploring Random Trees A New Tool for Path Planning.
7. Agogino, A. K. and Tumer, K. **2012.** A Multi agent Approach to Managing Air Traffic Flow, Appears in Autonomous Agents and Multi Agent Systems.
8. Bayen, A. M., . Raffard, R. L and Tomlin, C. J. **2006.** Adjoint-based control of a new Eulerian network model of air traffic flow. *IEEE Transactions on Control Systems Technology*.
9. Anonymous, A**. 2005.** Terminal area forecast summary, fiscal year 2004-2020, Technical Report. FAAAPO-05-1, U.S. Department of Transportation, Federal Aviation Administration.
10. Alpaydin, E. **2004.** *Introduction to Machine Learning.* The MIT Press. Cambridge.
11. Balakrishnan, H. December **2007.** Techniques for reallocating airport resources during adverse weather. In Proceedings of the 46th IEEE Conference on Decision and Control. pages 2949–2956. New Orleans, LA..
12. Bastin, G. and Guffen, V. **2006**. *Congestion control in compartmental network systems*. Systems and Control Letters.
13. Abdul Hassan, A.. K., Murad, M. S. and Alani, M. S. **2007.** Motion Planning Method Using Approximate Cell decomposition with Petri Nets for Single Mobile Robot. University of Technology, Baghdad, Iraq.
14. Kuwata, Y., Fiore, G. A., Teo, J. and Frazzoli, E. **2008.** How Motion Planning for Urban Driving using RRT. IEEE/RSJ International Conference on Intelligent Robots and Systems.
15. Shwail, S. H. and Abdul Hassan, A.. K. **2014.** Probabilistic Roadmap, ‘A*, and GA for Proposed Decoupled Multi-Robot Path Planning. *Iraqi journal of applied physics,* **10**(2).