Proceedings of the 36th Chinese Control Conference
July 26-28, 2017, Dalian, China

Path Planning and Intelligent Scheduling of Multi-AGV Systems in Workshop

Chengbao Liu[1, 2], Jie Tan[1], Hongsheng Zhao[1], Yaning Li[1, 2], Xiwei Bai[1, 2]

1. Institute of Automation, Chinese Academy of Sciences, Beijing 100190

2. University of Chinese Academy of Sciences, Beijing 100049

E-mail: liuchb15@163.com

Abstract: With the continuous development of intelligent logistics, the application of Automated Guided Vehicles (AGVs) increasingly becomes popular in many industrial fields. However, there are a series of problems in multi-AGV systems, such as resource allocation, conflict and deadlock. It is difficult to plan the shortest path for each AGV without conflict and collision in multi-AGV systems. In this paper, a multi-AGV scheduling system in workshop is established by using the unidirectional directed graph method and the A* algorithm for path planning of AGVs. In addition, the system is realized by programming and a simulation experiment of 20 AGVs is set up. Finally, the simulation results show that the system can effectively solve the conflict problem of AGVs, and is stable and high real-time. The system is easily extended to their similar multi-AGV scheduling systems, and has a great application value.

Key Words: Automated Guided Vehicle, Path Planning, Scheduling

1 Introduction

Automated Guided Vehicle (AGV) as a means of transport has been widely used in automated warehouses and flexible manufacture systems (FMS). However, planning a conflict-free and collision-free path for each AGV is still a key for multi-AGV systems. In recent years, multi-AGV systems have been gradually used in express sorting systems, which hundreds of AGVs work simultaneously in an area and hundreds of thousands of pieces of express are classified in an hour. This will improve well sorting efficiency of express and save manpower cost. Therefore, it has great theoretical significance and application value to establish an efficient multi-AGV scheduling system.

Path planning and scheduling are two basic problems in AGV systems. At present, researchers have done many studies and applications. Fazlollahtabar and Saidi-Mehrabad [1] reviewed in detail for scheduling and path planning problems of AGV, and the relevant methods were compared. Nishi et al. [2] addressed a bilevel decomposition algorithm for solving well the simultaneous scheduling and conflict-free path planning problems for multi-AGV systems. Rashidi and Tsang [3] established a Minimum Cost Flow model and put forward a novel algorithm, NSA+, which extended the standard Network Simplex Algorithm (NSA) and can be used to solve very large problems with polynomial time complexity. In addition, path planning falls into two categories: static and dynamic algorithms. Static algorithms include Dijkstra algorithm [4], A* algorithm [5-6], ant colony algorithm, genetic algorithm, artificial potential field [7], fuzzy path planning algorithm [8-9] and so forth. For dynamic algorithms, a banker algorithm [10-11] is improved to avoid deadlock and to achieve better utilization of vehicles, but most studies also solve conflict-free path planning problems by time windows method [12-14]. In [4], Sun and Liu et al. improved Dijkstra algorithm by adding the turning number and angle of AGV as the evaluation index, and established a multi-AGV scheduling system by combining the time windows method. Wang et al. [5] and Shen et al. [6] improved A* algorithm respectively and detailedly researched AGV path planning problems. For multi-AGV scheduling systems, static algorithms are often inefficient, or unable to solve the collision problem of AGV, so Smolic-Rocak and Bogdan et al. [14] proposed a time windows method, which can effectively solve the collision problem. However, when there are a large number of AGVs in a scheduling system, the time complexity of the time windows algorithm increases and it is limited its wider applications.

In this paper, a multi-AGV scheduling system is established by using the unidirectional directed graph method and combining the A* algorithm. It can effectively solve the conflict and collision problems of multi AGV systems. Besides, a 20 AGVs simulation system is set up and the results show that the system can run continuously and well, and has a good robustness.

2 System Architecture

Multi-robot architecture [15] is generally divided into centralized control, distributed control and hybrid control. Centralized control centralizes planning the strategies of robots by using global informations and has no communication between robots. The centralized control system can guarantee the optimal control performance, but the upper control system needs a lot of resources and has a higher request to the server. In distributed control, each robot is an agent, can independently make their own decisions, and has communication with other robots. Then the system has flexibility, scalability and fault tolerance, but cannot guarantee the optimal control performance. Hybrid

*This work is supported by National Natural Science Foundation (NNSF) of China under Grant U1201251 and the National Science and Technology Support Program of China under Grant 2015BAF09B01.

control has advantages of the former two and can not only perform tasks independently, but also be scheduled by the upper control system. The hybrid control system can complete tasks cooperatively and adapt to the complex environment. According to the actual demand, a multi-AGV system only completes simple handling tasks and needs more AGVs. Therefore, in order to reduce costs, the multi-AGV system in this work chooses the centralized control strategy.

Figure 1 shows the control framework of multi-AGV system, which is divided into the upper system and the lower system. The upper system comprises four modules: AGV module, task module, scheduling module, and interface monitoring module. AGV module includes adding, management and charging functions of AGV and updates status informations of AGV. Task module includes adding and management functions of tasks, and the design of external data interface, such as receiving task informations from MES. Scheduling module includes path planning, scheduling and intelligent obstacle avoidance of AGV, which AGVs are allocated tasks according to some intelligent scheduling rules and collision detection algorithms are designed to realize intelligent obstacle avoidance. Interface monitoring module maintains work log of the system and saves alarm informations to model and analyze the system by using the big data technology in future, in addition to managing the environment map and establishing a real-time monitoring interface of the system. However, the lower system consists of many AGVs, which each AGV as a single client can sense location, speed, obstacle, power and load to detect the status of AGV in real time. The upper system communicates with the lower system by using WiFi or other wireless networks, which the server sends path informations to every client and every client sends their location, power and other informations to the server, respectively.

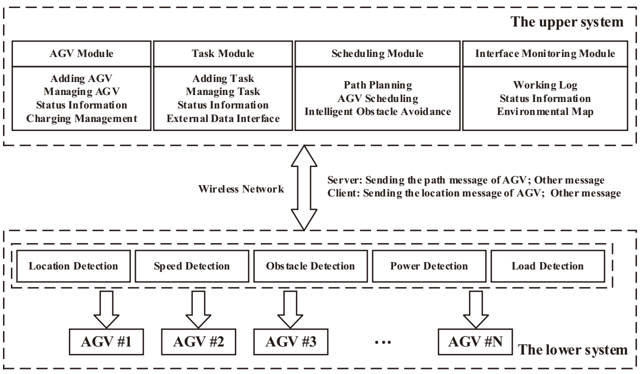

Fig.1: The control framework of multi-AGV system

3 Path Planning and Intelligent Scheduling

3.1 Environment Modelling

According to the actual environment in workshop, it is a basis of path planning to establish a suitable environmental model. There are many methods for environment modeling of the workshop, and the most common method is the topological map by using the graph theory to realize the mathematical representation of the environment. More specifically, the graph is denoted as $G = (V, E)$, where V is a set of vertexes, E is a set of edges, and each edge in E connects two vertices in V. The weights of edges in E are denoted as w_{ij}, where $i \neq j$ *and* $i, j = 1,2, \cdots, n$, and n is the number of vertices. According to the workshop environment of a factory, a topological map of the environment is established in this work, as shown in Figure 2. There are 70 nodes and 99 arcs in the graph and all nodes are numbered from 1 to 70. Each arc represents a bidirectional path or a unidirectional path and the weight of the arc represents the length of the path. In addition, the topological map includes two loading areas, six warehouse areas, a charging area and a parking area. The loading areas consist of six loading platforms and their nodes are 7, 8, 9, 12, 13, 14; the warehouse areas consist of eighteen unloading platforms and their nodes are 33, 35, 38, 40, 43, 45, 48, 49, 52, 53, 56, 57, 60, 61, 64, 65, 68, 69; the charging area consists of an entrance and an exit, 31 and 21 respectively; the parking area also consists of an entrance and an exit, 19 and 17 respectively. So the path planning problem is transformed into a graph search problem by giving a start node and a goal node and using a search algorithm to plan an optimal path for AGV.

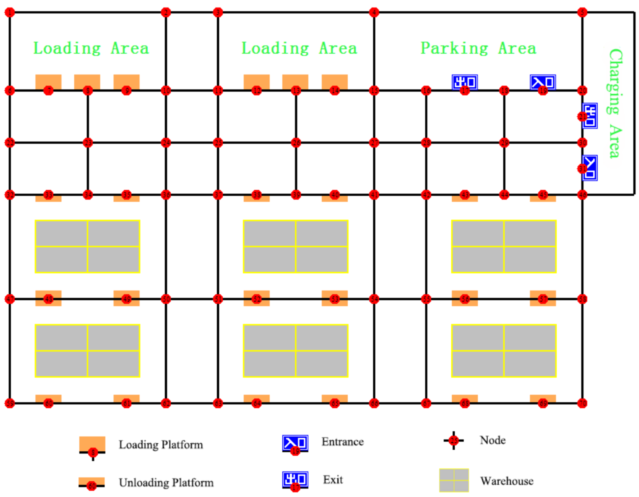

Fig.2: The environmental map of multi-AGV system

3.2 Path Planning

Based on the environmental map in section 3.1, the A* algorithm is used to plan paths of AGVs. The A* algorithm is a heuristic search algorithm, which adds an evaluation function on basic of breadth first search algorithm, then evaluates all extended nodes of a parent node through using the evaluation function to find an optimal extended node and again search an optimal node from the optimal extended node until the goal node. The evaluation function of A* algorithm is expressed as:

$$f(n) = g(n) + h(n) \quad (1)$$

Wherein $0 \leq h(n) \leq h^*(n)$.

$f(n)$ represents the estimated cost from the start node through the node n to the goal node; $g(n)$ represents the real cost from the start node to the node n; $h(n)$ represents the estimated cost from the node n to the goal node; $h^*(n)$ represents the real cost from the node n to the goal node.

According to the environmental map of AGV, the Euclidean distance between two nodes is chosen as an estimating function, which is shown as formula (2).

$$h(n) = \sqrt{(x_n - x_g)^2 + (y_n - y_g)^2} \quad (2)$$

Wherein (x_n, y_n) represents the coordinate of the current node n; (x_g, y_g) represents the coordinate of the goal node.

Obviously, $h(n)$ in formula (2) satisfies its conditions, $0 \le h(n) \le h^*(n)$, so the A* algorithm is complete and always to find the optimal path in the environmental map. The description of path planning algorithm of AGV is as follows.

Algorithm 1: Path planning of AGV using A* algorithm.

Inputs: The adjacency list of the graph $G = (V, E)$, which is the environmental map of AGV system, is denoted by $List_{graph}$. The start node of AGV is denoted by $Node_{start}$. The goal node of AGV is denoted by $Node_{goal}$.
Outputs: The optimal path list of AGV is denoted by $List_{\text{optimal-path}}$.
Initialization: Create an OPEN list, which is denoted by $List_{OPEN}$, and Create a CLOSED list, which is denoted by $List_{\text{CLOSED}}$. Set the extended nodes (or adjacent nodes) of a node i, which is denoted by Ex$Node_i$. Set $List_{OPEN} = \{Node_{start}\}$, $List_{\text{CLOSED}} = \{\emptyset\}$.

foreach $node$ **in** $List_{graph}$ **do**
    if $Node_{start} = node$ **then**
        if Ex$Node_{start}$ contain $Node_{goal}$ **then**
            return $List_{\text{optimal-path}} \leftarrow \{Node_{start}, Node_{goal}\}$;
        else
            $List_{OPEN} \leftarrow \emptyset$; $List_{OPEN} \leftarrow$ Ex$Node_{start}$;
            $List_{\text{CLOSED}} \leftarrow Node_{start}$;
        end if
    end if
end foreach
while $List_{OPEN} \ne \emptyset$ **do**
    traverse all nodes in $List_{OPEN}$;
    if $node \in List_{OPEN}$ contain $Node_{goal}$ **then**
        $List_{\text{CLOSED}} \leftarrow node$;
        return $List_{\text{optimal-path}} \leftarrow List_{\text{CLOSED}}$;
    else
        find a node $Node_{temp}$ in $List_{OPEN}$, which the value of $f(n)$ is minimized;
        $List_{\text{CLOSED}} \leftarrow Node_{temp}$; $List_{OPEN} \leftarrow \emptyset$;
        $List_{OPEN} \leftarrow$ Ex$Node_{temp}$;
    end if
end while
return $List_{\text{optimal-path}} \leftarrow List_{\text{CLOSED}}$;

In the above algorithm, the environmental graph model of AGV, $G = (V, E)$, is represented by an adjacency list. The optimal path of AGV can be obtained by the Algorithm 1 when inputting the adjacency list, the start node and the goal node.

3.3 Intelligent Scheduling

The scheduling of multi-AGV systems is designed a conflict-free and collision-free path for each AGV and an evaluation index of the system is minimized, such as minimizing the total number of AGV or minimizing the total running time of AGV. However, for a multi-AGV system, there are three types of conflict: intersection conflict, catching up with conflict and facing conflict, as shown in Figure 3.

The three conflicts are essentially a time and space overlap problem. Only by solving the conflicts can the system run safely and smoothly, otherwise the system will bring about the deadlock without continuing to run. This paper assumes that the running speed of AGV is constant, and the catching up with conflict in Figure 3 (b) can be avoided. Then the intersection conflict in Figure 3 (a) can be solved by waiting strategy. In addition, if two AGVs run in the same direction and one in front stops (e.g. loading or unloading) or breaks down, then the conflicts can also be solved by waiting strategy. However, the facing conflict in Figure 3 (c) can be solved only through reconsidering new strategies. At present, the time windows method cannot be applied to a system with a large number of AGVs. So in this paper, a scheduling system of multi-AGV is established by using the unidirectional directed graph method, which each arc is unidirectional and then the conflicts in the system can be solved by waiting strategy.

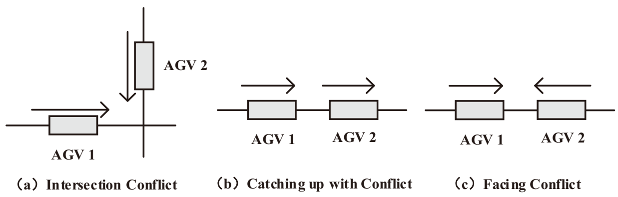

Fig.3: Three conflict types of multi-AGV system

Figure 4 shows the running process of multi-AGV system, which the upper system is started firstly, adds tasks and AGVs, and connects the clients of AGVs to keep their normal communication. The system consists of four threads: the main thread, the scheduling thread, the collision detection thread, and the interface display thread. The main thread is responsible for adding new AGVs, new tasks, and the system exit. The scheduling thread realizes the intelligent scheduling of tasks, the collision detection thread detects the conflicts of AGVs and processes them, and the interface display thread realizes the real-time display of AGVs running.

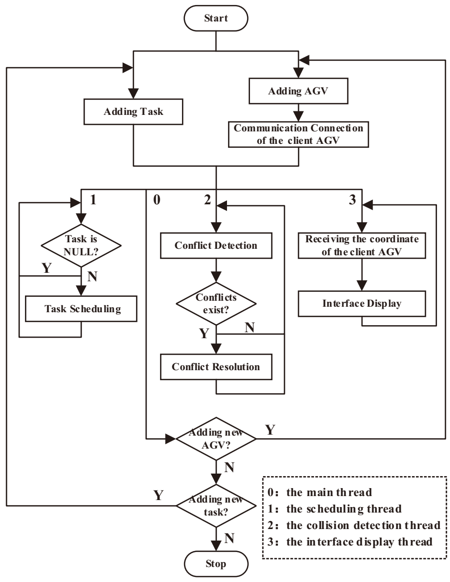

Fig.4: The running process of multi-AGV system

4 Simulation and Result

4.1 Design of Simulation System

In this paper, an intelligent scheduling, management and simulation system of multi-AGV in workshop is established, which is based on the Studio Visual 2013 platform, and is used C# as the development language and Microsoft SQL Server 2014 as the system database.

A. The upper system

The upper system consists of menu, electronic map, and system status information, as shown in Figure 5. The menu includes a system menu, a task menu, an AGV menu, a communication menu and a help menu, which the system menu comprises three sub menus: starting system, system setting and exit, the task menu comprises two sub menus: adding task and task management, the AGV menu comprises two sub menus: adding AGV and AGV management, the communication menu comprises a sub menu: communication setting, and the help menu comprises two menus: operating instruction and about. The electronic map is the environmental map in section 3.1. The system status informations are updated in real time, such as system status and status information of AGVs and tasks.

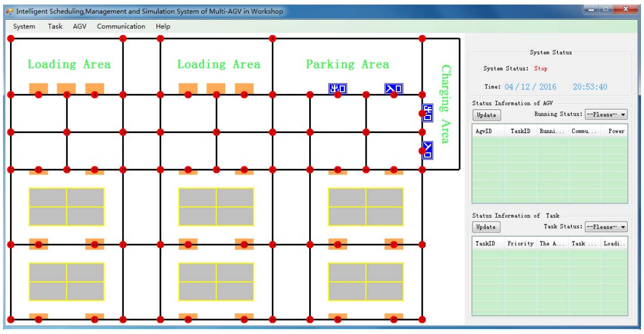

Fig.5: The upper system

Fig.6 The simulation client of AGV

B. The lower system

Due to the limited conditions, the simulation client is used instead of AGV, as shown in Figure 6. Assume that AGVs are running at a constant speed, $v = 1m/s$, the time necessary is $1s$ when an AGV turns 90°, and the time necessary is $3s$ when an AGV loads in the loading platform or unload in the unloading platform. Then a path-time mode is established, which simulates the real running process of AGV. The simulation client sends the coordinates of AGV to the upper system to display its running status in real time. Figure 6 contains two parts: server setting and AGV information. The server setting realizes the connection with the upper system. The AGV information shows the current location, the power, and the path of AGV.

4.2 Result

The simulation system is set up by 20 AGVs and 1000 tasks, and the simulation running process of multi-AGV is shown in Figure 7, where a small yellow rectangular block represents an AGV, each rectangular block is numbered to correspond to the number of the simulation client and the node 1 is set to the coordinate origin (0,0). Each AGV comprises four running directions, which 1 represents the east, 2 represents the north, 3 represents the west and 4 represents the south. All conflicts can be solved through waiting strategy. For example, in Figure 7, the AGV 9 moves to the east, the AGV 15 moves to the west, and there will be a conflict in the node 15. Then if the distance between the AGV 9 and the AGV 15 is less than the safety distance, the AGV 15 will stop until the AGV 9 passes through the node 15. The simulation results show that the simulation system can run reliably and efficiently, the conflicts of AGVs can effectively be solved, the algorithm has lower complexity in computation, and the system has a high real-time performance and can be applied to a scheduling system with a large number of AGV.

Fig.7: The process of simulation running of multi-AGV

The statistics of the simulation that 20 AGVs completed 1000 tasks is shown in Table 1, where $n_{total\ tasks}$ denotes the total number of tasks completed by each AGV; t_{run} denotes the total time for each AGV to run and is the sum of t_{task} and $t_{no-task\ run}$; t_{task} denotes the total time for each AGV to perform its tasks; $t_{no-task\ run}$ denotes the total time for each AGV to run without tasks; $r_{utilization}$ denotes the average utilization rate of each AGV, as shown in formula (3).

$$r_{utilization} = \frac{t_{task}}{t_{task}+t_{no-task\ run}} \times 100\% = \frac{t_{task}}{t_{run}} \times 100\% \quad (3)$$

The statistical result shows that the total number of tasks completed by each AGV is very close and about 50.So the system is able to uniformly assign tasks for each AGV according to the scheduling algorithm and avoid excessive use of an AGV. But the utilization rate of each AGV is lower and about 30%. Therefore, there should be more studies to enhance the utilization rate of AGV in future.

Table 1: The statistics of the simulation that 20 AGVs completed 1000 tasks

AGV number	$n_{total\ tasks}$	t_{task}/s	t_{run}/s	$r_{utilization}$
1	52	1543	5060	30.49%
2	52	1536	5067	30.31%
3	51	1493	5029	29.67%
4	49	1537	4987	30.82%
5	50	1608	5031	31.96%
6	50	1515	5052	29.99%
7	49	1527	5033	30.34%
8	51	1558	5008	31.71%
9	50	1514	4989	30.35%
10	48	1629	5038	32.33%
11	52	1519	5063	30.00%
12	50	1477	5049	29.25%
13	49	1487	4997	29.76%
14	48	1555	5047	30.81%
15	50	1479	4976	29.72%
16	49	1452	4942	29.38%
17	49	1573	4980	31.59%
18	50	1514	4965	30.49%
19	49	1543	5029	30.68%
20	52	1507	5034	29.94%

5 Conclusion

The continuous development of intelligent logistics has brought about a wide range of application prospects for mobile robots, and meanwhile puts forward new challenges. The robot express sorting system with AGVs has low cost and high efficiency, and provides a new direction for the express industry. However, the system needs a large number of AGVs and a deadlock-free scheduling system of multi-AGV becomes a difficult. In this paper, a scheduling system of multi-AGV is established by using the unidirectional directed graph method and the A* algorithm for path planning of AGV, which effectively solves the conflict of AGV and has high reliability, small calculation and real-time performance, and the number of AGV in the system less affects the complexity of algorithm. But the unidirectional directed graph method is at the expense of the operating efficiency of the system, and the optimal path may not be the shortest path. In the future work, more researches should be carried on dynamic scheduling algorithms of multi-AGV to solve the conflict of AGV and as far as possible to plan shortest path for each AGV to improve the efficiency of the system.

References

[1] H. Fazlollahtabar and M. Saidi-Mehrabad, Methodologies to Optimize Automated Guided Vehicle Scheduling and Routing Problems: A Review Study, *Journal of Intelligent & Robotic Systems*, 77(3-4): 525-545, 2015.

[2] T. Nishi, Y. Hiranaka, I.E. Grossmann, A Bilevel Decomposition Algorithm for Simultaneous Production Scheduling and Conflict-free Routing for Automated Guided Vehicles, *Computers & Operations Research*, 38(5): 876–888, 2011.

[3] H. Rashidi and EPK. Tsang, A Complete and an Incomplete Algorithm for Automated Guided Vehicle Scheduling in Container Terminals, *Computers & Mathematics with Applications*, 61(3): 630-641, 2011.

[4] Q. Sun, H. Liu, et al., On the Design for AGVs: Modeling, Path Planning and Localization, *in Proceedings of the 2011 IEEE International Conference on Mechatronics and Automation*, 2011: 1515-1520.

[5] C. B. Wang, et al., Path Planning of Automated Guided Vehicles Based on Improved A-Star Algorithm, *in Proceeding of the 2015 IEEE International Conference on Information and Automation*, 2015: 2071-2076.

[6] B. W. Shen, N. B. Yu, and J. T. Liu. Intelligent Scheduling and Path Planning of Warehouse Mobile Robots, *CAAI Transactions on Intelligent Systems*, 9(6): 659-664, 2014.

[7] Z. Z. Yu, J. H. Yan, et al., Mobile Robot Path Planning Based on Improved Artificial Potential Field Method, *Journal of Harbin Institute of Technology*, 43(1): 50-55, 2011.

[8] Z. B. Sui and J. L. Lu, A Study on the Path Planning of Mobile Robot with the Fuzzy Logic Method, *Transactions of Beijing Institute of Technology*, 23(3): 157-162, 2003.

[9] R. Zhou, Z. H. Lu, and C. L. Chen, Navigation for Mobile Robots Using Reinforcement Learning and Fuzzy Logic, *Computer Simulation*, 22(8): 157-162, 2005.

[10] L. Kalinovcic, T. Petrovic, et al., Modified Banker's Algorithm for Scheduling in Multi-AGV Systems, *in Proceedings of 2011 IEEE International Conference on Automation Science and Engineering*, 2011: 351-356.

[11] V. Bobanac and S. Bogdan, Routing and Scheduling in Multi-AGV Systems Based on Dynamic Banker Algorithm, *in Proceedings of 2008 16th Mediterranean Conference on Control and Automation*, 2008: 1168 – 1173.

[12] X. B. Hu; M. K. Zhang, et al., An Effective Method to Find the K Shortest Paths in A Generalized Time-Window Network, *in Proceedings of 2016 IEEE Congress on Evolutionary Computation*, 2016: 2345 – 2351.

[13] L. L. Liao, X. S. Cai, Vehicle Routing with Time Windows Based on Two-Stage Optimization Algorithm, *in Proceedings of 2016 Chinese Control and Decision Conference*, 2016: 4741 – 4745.

[14] N. Smolic-Rocak, S. Bogdan, et al., Time Windows Based Dynamic Routing in Multi-AGV Systems, *IEEE TRANS on Automation Science And Engineering*, 7(1): 151-155, 2010.

[15] Z. X. Cai, B. F. Chen, et al., Synergy Principles and Technologies of Multi-mobile Robots, Beijing: *National Defence Industry Press*, 2011.