

© 2006 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Replanning of Multiple Autonomous Vehicles in Material Handling

G. Paul, D.K. Liu

ARC Centre of Excellence for Autonomous Systems (CAS)
University of Technology, Sydney
NSW, Australia

Gavin.Paul@eng.uts.edu.au, dkliu@eng.uts.edu.au

Abstract—The fully automated docks in Australia present opportunities for applications of autonomous vehicles and engineering innovation. When planning tasks to be done by multi-autonomous vehicles in an enclosed area with a known dynamic map (i.e. bi-directional path network), there are many issues that have not yet been comprehensively solved. The real world presents more complexity than the initial algorithms addressed. There are problems that occur due to interaction with the real-world. This means autonomous vehicles can stop, are affected, or face problems, and hence tasks and vehicles' paths and motion need to be replanned. In order to replan, a greater understanding of the state of vehicles, the state of the map, and importantly the importance of tasks and vehicles is definitely needed. This paper explores the improvements made to replanning by gaining a thorough understanding of the map and then utilising map information to make the best, most efficient replanning decision. Five replanning Methods are investigated and four Options which combine the Methods in different ways, are tested in this research. A map analysis Method is also presented. Simulation studies show that map information based replanning is the most efficient Method out of those tested.

Keywords—replanning, multi-autonomous vehicles, path planning

I. INTRODUCTION

Applications of autonomous vehicles in various areas such as automated container terminals, manufacturing, etc. have been increasing significantly since 1990's, which resulted in a considerable increase in complexity of autonomous vehicle motion coordination, planning and scheduling with the goal to operate autonomous vehicles efficiently and safely. Planning and scheduling of autonomous vehicles, including path and motion planning and collision avoidance, play an important role to improve the productivity, and have, therefore, been recently recognised as key research issues in autonomous vehicle systems by both academics and industry [1][2][3].

Path planning algorithms have been widely studied for applications in environments that are mapped to path networks. In unknown or partially known environments, for example, D* [4] and Delayed D* [5], E* algorithm [6], etc. were proposed. Meta-based optimization algorithms have been studied for applications in path planning, e.g. particle swarm optimization (PSO) [7] and genetic algorithm [8]. For applications in known

static environments such as road networks, A* and Dijkstra algorithms and their variations are widely used.

The real world, however, presents more complexity than these theoretical algorithms addressed. There are problems that occur due to interaction with the real-world. This means autonomous vehicles can stop and delay, are affected, or face problems, and hence tasks and vehicle paths and motion need to be replanned. Real-time replanning of multi-autonomous vehicles in dynamic environments such as container terminals is essential to productivity improvement. This, however, has so far not attracted a lot of attention.

There are many issues faced while replanning that need to be considered. For example the number, state, status, position of all vehicles in the bound environment; the state of the map, traffic conditions, bottleneck conditions; the number, importance of tasks and probability of arrival of new tasks; any contingency plans for the vehicles, map and tasks; most importantly the safety, etc.

Current system methods involve taking the highly inefficient, but safe option, of stopping of all vehicles in the environment. This means that path and task data is not lost. However, due to the hindrance that an insignificant vehicle can cause when the system reacts to its deviation from initial plan, this proves to be unnecessarily inefficient. Work has been done on graph-based replanning algorithm able to produce bounded suboptimal solutions, by tuning the solution quality on available search time [9].

In order to generate an intelligent replanning algorithm, initially the state of all vehicles is determined, and vehicles are ranked in order of importance. In the real world, time for deliberation is usually very limited; vehicles need to make decisions and act upon these quickly. Many researchers have worked on coping with imperfect information and dynamic environments, efficient replanning algorithms have been developed that correct previous solutions based on updated information [10][11][12][13]. These algorithms maintain optimal solutions for a fraction of the computation required to generate such solutions from scratch. In general these approaches are limited to a single robot operating in a partially known/unknown environment. They have difficulty being applied to multiple robot situations and when solving bottleneck problems in bi-directional path networks.

This Paper develops five Methods, all of which could essentially be applied independently, but are most effective when combined. A map analysis Method is presented to extract map information, i.e. node importance in the path network map, for the purpose of replanning vehicles' paths and speeds when a vehicle is stopped or delayed. Four testing Options, namely stop all, automatic, map information only and no map information, are tested by simulation studies. The result is an improved replanning methodology for multiple robots including bottleneck avoidance in bi-directional path networks.

II. REPLANNING METHODS

A. An Example Environment

Fig.1 shows an example environment which is represented by a bi-directional path network map. The numbers in the map represent the nodes the autonomous vehicles can access. The lines are the connections among nodes and represent possible paths the vehicles have to follow. Each autonomous vehicle is supposed to move from node to node following the link between nodes. Planning, scheduling and collision avoidance for multi-autonomous vehicles in the bi-directional path network are much more difficult than that in unidirectional path networks. Although the map is not vast in size (about 20mx30m), nor in node numbers (186), it is in fact quite complex due to the large number of bottleneck areas. B1, B2, B3, B4 and B5 are five examples of the bottleneck areas. Broadly speaking the map is broken up into 4 areas with corridors between them. These corridors are highly important, and must be kept clear as much as possible since they are used to move between the nodes in the different areas.

A simultaneous path and motion planning (SiPaMoP) method [14] is applied to coordinate the motions of multi-autonomous vehicles in this environment. This method integrates path and speed planning and collision avoidance to coordinate efficiently a team of vehicles. The SiPaMoP approach is applied in this research to generate collision free paths and speeds for all vehicles in the path network shown in Fig.1. Replanning Methods studied in this paper will call the SiPaMoP function to replan vehicles' paths and speeds when a vehicle is stopped or delayed.

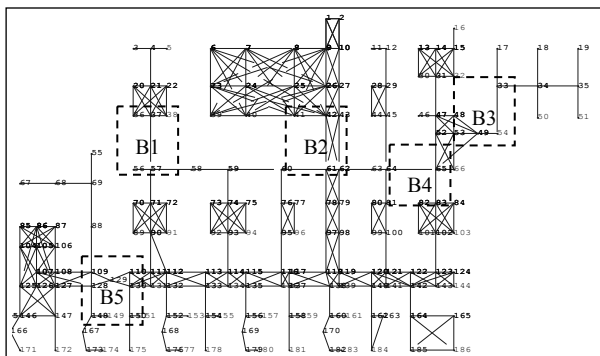


Figure 1. The example path network

B. Replanning Methods

The first step of the replanning process is to get the state of the vehicles. This must be done when a vehicle changes state

since its relative state and all other vehicles' relative states are initially undetermined according to the main controller. It is not possible to say without this check whether the stopped vehicle will impede other vehicles, or whether by having the vehicle stopped, another possibly more efficient path is opened up for other vehicles. The state of vehicles is determined by working out which vehicles are affected by other stopped ones, and then consequently which vehicles are affected by the affected vehicles which are stopped.

Five Methods are proposed to assist in making improved efficiency decisions. Each method results in an order of importance M_i where i is for method numbers 1 to 5 for vehicles V_x where $x=\{1,2,\dots,n\}$

1) Vehicle State Based Method

The first Method is the simplest one and bases the levels of importance upon the state S_{V_x} of vehicles V_x . The rules for this Method are as follows:

- Finish vehicles that can be without planning
- Replan and finish vehicles that can be replanned
- Remaining vehicles in 4 states: Completely finished; Directly stopped; Awaiting new or altered task; Affected by stopped vehicles in states 1 or 2
- Only possible to act upon vehicles which are unaffected and awaiting instructions.

$$M_1 = f(S_{V_1}, S_{V_2} \dots S_{V_n}) \quad (1)$$

2) Task Priority/Vehicle Position Based Method

The second Method uses the importance of the tasks in order to determine the most important vehicles. When vehicles are allocated globally numbered tasks T_n priority is given to the lowest task number, so for V_x priority P_{V_x} is a relative importance out of currently allocated tasks. This Method of replanning utilises this predetermined importance to give a rank to each vehicle. The advantages of this Method are its simplicity, and also its ability to use pre-calculated relative importance in order to make the best decision. It is advantageous in terms of attempting to finish higher priority tasks. Logically it is still not possible to plan or finish vehicles which are in states of "Stopped" or "Completely Blocked". The disadvantage of this Method is that it is not based precisely on actual present conditions since the decision making was made by initial task allocation logic.

$$M_2 = f(P_{V_1}, P_{V_2} \dots P_{V_n}) \quad (2)$$

3) Task Node Importance Based Method

This Method draws from the work performed in Section III on analysing maps to find the relative node and link importance. This analysis led to an understanding of the bottleneck areas of a map. Basically, the pre-work for this Method is in three steps, ordered in increasing levels of computational complexity.

- All nodes that only have 1 link connected to them are not bottlenecks, these are the least important
- Plan paths from every node to every other node and calculate the data to find the usage of each node
- Lock each node individually then for each locked node, plan from every node to every other node. Count the number of impossible paths created for each locked node. For example, if a link is removed or the in-between node is locked, can a vehicle get between the two nodes (A->B)? If not the node is a bottleneck which has higher priority (Fig.2).

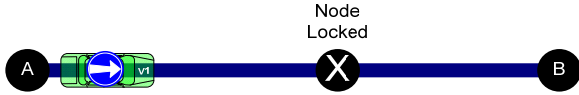


Figure 2. Impossible Path

The results from these three steps give a clear idea about where the most important nodes are in terms of efficiency and bottlenecks, and are stored in a data file which need only be calculated once for a basically static map. Then the rankings of the nodes are used to determine the relative importance $I_{N_{Vx}}$ of a vehicle based upon its positional node N_{Vx} .

$$M_3 = f(N_{Vx}, I_{N_{Vx}}, K_{Vx}), x = \{1, 2, \dots, n\} \quad (3)$$

With node importance calculations where the relative importance of the node is much higher than the others, using a greater coefficient for this Method increases the accuracy of decision making in the overall importance. Rather than use a linear coefficient of importance for the Method, a more intuitive coefficient is used. It determines a way to increase the coefficient more rapidly than linearly depending upon the relative importance. For example if this Method works out and ranks 3 vehicles, based upon their node importance of 1st, 50th and 186th then obviously the vehicle with rank 1 is of vital importance compared to the others and this would not be sufficiently exemplified by the linear coefficient. This Method's inherent coefficient K_{M3} to multiply the resultant ranks by is set to:

$$K_{Vx} = \sqrt[4]{(\text{NodeRank} / \text{TotalNodes})} \quad (4)$$

4) Remaining Weight and % Method

Method 4 is a product of the remaining weight Rw_{Vx} and the relative remaining weight, i.e. remaining weight as percentage (%) of overall plan RRw_{Vx} . This will ultimately result in the order of importance for each vehicle.

$$M_4 = f(Rw_{Vx}, RRw_{Vx}), x = \{1, 2, \dots, n\} \quad (5)$$

5) Remaining Nodes and % Method

Similarly to the previous Method, Method 5 is evaluated based upon the remaining nodes Rn_{Vx} and the relative remaining nodes as percentage (%) of overall RRn_{Vx} .

6) Overall Rank Calculation

Each of the above five Methods results in a different ranking for each vehicle and allocates points based upon coefficients of importance for each Method (lower scores translate to higher importance and visa-versa). The final culminated resultant order of importance for n vehicles is calculated by (6) and used to replan more intelligently and efficiently while maintaining a consistent safety standard.

$$OR_{Vx} = \text{AscendingSort} \left\{ \sum_{i=1}^5 M_i \times C_i \right\} \quad (6)$$

where, OR is overall rank of V_x for $x = \{1, 2, \dots, n\}$, M_i is the ranks obtained and C_i is the set coefficient from each Method.

III. MAP ANALYSIS AND EFFECT ON REPLANNING

A bottleneck is a node or a link in a path network which vehicles need to use more often than most other nodes. An example bottleneck is shown in Fig.3. If it is assumed that the link between node 6 and 7 be bi-directional and can only carry one vehicle in any position at a time, and that there is the same probability of having tasks between any two nodes, then there will be a bottleneck condition on the link between node 6 and 7. Any vehicle going to the other side will have a 6/11 chance of having to cross the link j.

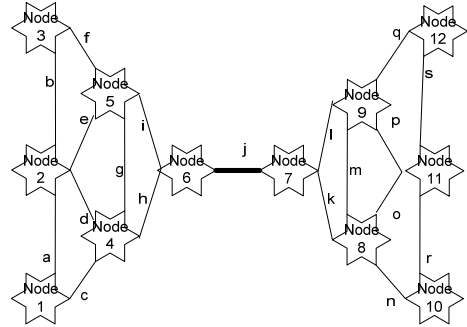


Figure 3. Bottleneck Demonstration

A. Calculating Most Efficient Paths and Node Usage

In order to discover bottlenecks, iterative exploratory program was created which basically went through each possible task that can be allocated and worked out the quickest path according to the SiPaMoP planning method used. Plans are generated for all possible hypothetical task case - from every one point to every other point on the map. The next step is to go through each path and culminate the usage of nodes and the link in between. For example a path between nodes 1->12 is 1->4->6->7->9->12 which means the links "c", "h", "j", "l", "q" are all used once. The usage (times used) of nodes and the usage of links for the simple map (Fig.3) are shown in Fig.4.

Every time the link j is being used then the nodes 6 and 7 are both part of the path. This is intuitive but is very helpful in calculating the relative importance of the nodes. The more important a node is, the greater the probability of affecting the system. This concept is central to replanning.

B. Impossible Paths: Discovering Potential Bottlenecks

An additional method for working out the importance of nodes is to lock each node individually and then go through and attempt to plan all paths. When there is no path for this vehicle at this time, then this locked node's impossible-path-created variable increases. After going through each node, there is a total value of the impossible-path-created variable for each node that was locked. The results for the simple map (Fig.3), as expected, show that the only nodes that when locked create impossible paths for other vehicles, are nodes 6 and 7.

The same analysis is performed on the complex map shown in Fig.1. Due to the relationship between nodes and links, the process has been restricted to an analysis of the nodes themselves which are able to be locked hence affectively shutting down links. By studying the graphic result of the usages of the nodes it is clear that there are sections of nodes that are passages to other sections of the map. These sections are highly important for the most effective plans of vehicles and, in many cases, the only plan for vehicles. This analysis shows that nodes at the corridors 129, 62 and 64 are of vital importance for many paths. There are other nodes like 65 and 57 which are also of high importance in terms of both efficiency and bottlenecks. In some cases there are high importance nodes in terms of the most efficient route that may not be bottlenecks, but are purely highly utilised nodes.

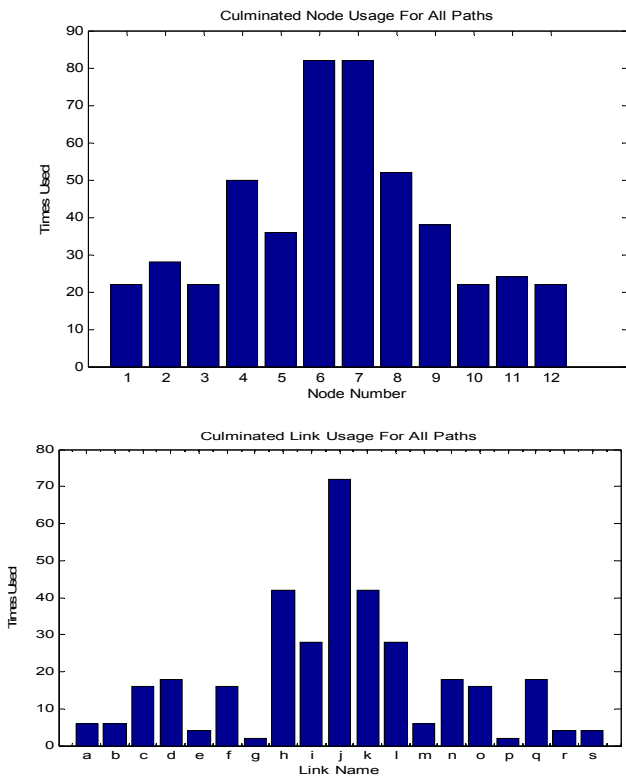


Figure 4. Node and link usage in the map shown in Fig.3

IV. SIMULATION STUDIES AND DISCUSSION

The simulation engine that was developed for testing of the interaction with the vehicles and the replanning actions on the map is shown in Fig.5. It is possible to stop and start vehicles

individually or multiply. Replanning data evaluated based upon previously explained replanning Methods is displayed as part of the program in order to interface with the user and provide valuable information (left to right: Vehicle Number; Current Task Number; Current Status of the Vehicle; Beginning Node; End Node; Remaining; Importance: shows vehicle's 5 Methods and overall importance).

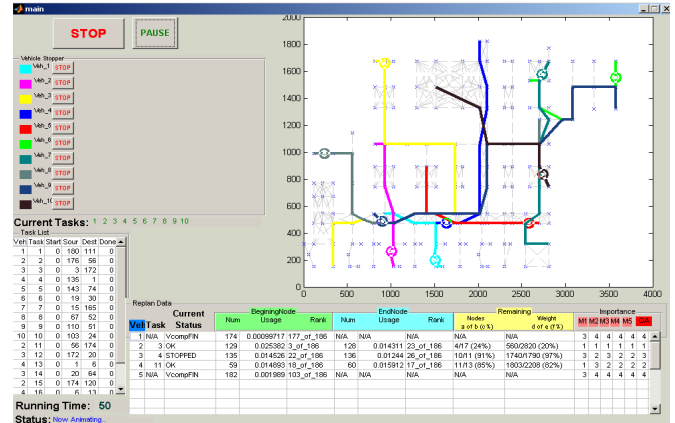


Figure 5. Simulation GUI

A. Testing overview

The testing process involved using randomly created task databases, then randomly stopping vehicles for an amount of time. The stopping time is increased after each minor test to a specified maximum. Each minor test uses one of four Options which are based upon different combinations of the replanning Methods. The 4 Options are:

- Option 2** – Stop all (current system method): when one vehicle is stopped then all other vehicles must stop which means they are all hindered for the amount of time a vehicle is stopped. Assuming best case: where the map remains static during stopped periods, otherwise it would be even more inefficient.
- Option 3** – Automatic: it uses all the 5 replanning Methods information to determine the best planning order for vehicles.
- Option 4** – Map information only: it utilises information from Method 3 only to make replanning decisions.
- Option 5** – NO map information: Uses all other Methods except Method 3.

Note that an Option 1 exists, but performs no replanning and does not react to vehicles stopping. This Option is of no value during testing for this Paper Also note all Options, except Option 2, collect all information for all replanning Methods, then the Option-specific final decisions utilise the information in different ways. Option 2 simply holds all path and motion plans, and subsequently when vehicles are restarted they are able to continue their movements from where they were without replanning.

B. Testing procedure

The testing procedure involves an automatic stopping feature to stop and start vehicles for a specified time period (eg 60sec). For different numbers of vehicles, stopping orders are created. The order does not vary between different Options

doing the same task database, nor when stopping periods changed.

Task time data t_{Ty} was collected for each task T_y , where y is the task number. For a single Option and a certain stopping time, the average time Av for n tasks is: $Av = (\sum_{y=1}^n t_{Ty}) / n$

The Total Time is the makespan (the total time until the last job is done) of the entire task database.

C. Testing results

Two case studies are conducted and the results are discussed below.

1) Case 1: 5 Vehicles and 50 tasks, stopping time changes from 30 to 180sec with 15sec increments

In this case study, 50 tasks were allocated evenly to 5 vehicles. Various vehicle speeds were tested, one is displayed, and the stopping time for each test went from 30sec to 180sec by increments of 15sec. All 4 Options were tested and the results, i.e. total task time and average task time, are shown in Fig.6. The graph shows that overall Option 3 exceeds the efficiency of all other Options. Intermittently Option 4 and 5 performed well, but as expected, the map information available to Option 3 resulted in better replanning decisions and hence more efficient average task times. Option 2 performed well only for the total task time when the stopping period was less (and relative to replanning time was greater).

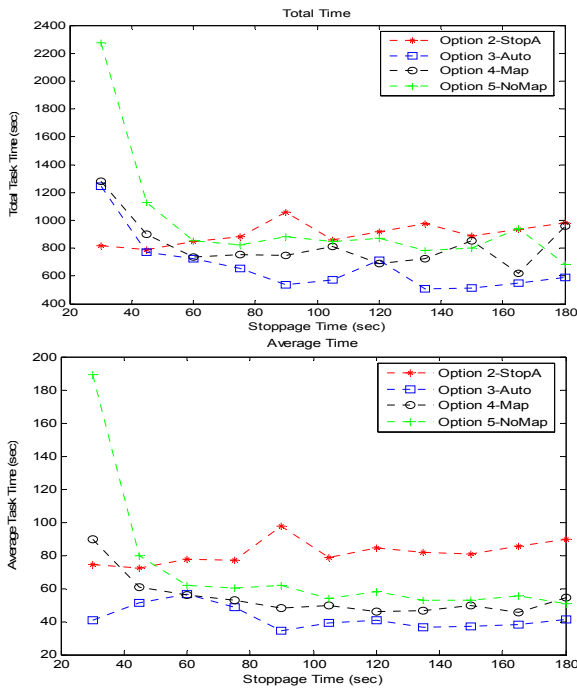


Figure 6. Case 1 Testing Results: (a) Total Time; (b) Average Task Time

2) Case 2: 4 Vehicles and 40 tasks, stopping time changes from 30 to 180sec with 15sec increments

In this case, 40 tasks are allocated to 4 vehicles. It can be seen from the results shown in Fig.7 that Options 3, 4 and 5 decisively outperform Option 2 in both total task time (Fig.7a) and average task time (Fig.7b). Among Options 3, 4 and 5,

Option 3 generally makes better decisions and performs more efficiently than the other Options (Fig.7c). These results imply that map information plays an important role in replanning. Note that for stopping time = 30 and 45, Option 4 is quicker than others. However for the stopping time = 90sec, Option 4 has made less efficient judgments.

D. Discussion on map information importance

Due to the ordering of the nodes in the map, there are several nodes around the map border which are not impeding any vehicles when they are locked and also they are only used when going to and from themselves. However, they need to have some sort of rank hence node numbers were used. This means one node may be given more preference over another even though it is actually the same importance. Although the coefficient will minimize this error, if only Method 3 (Option 4) is used, it can make the minor difference between making the best decision and making not such an efficient one. The utilisation of the other Methods as in Option 3 mitigates this while still utilising map information.

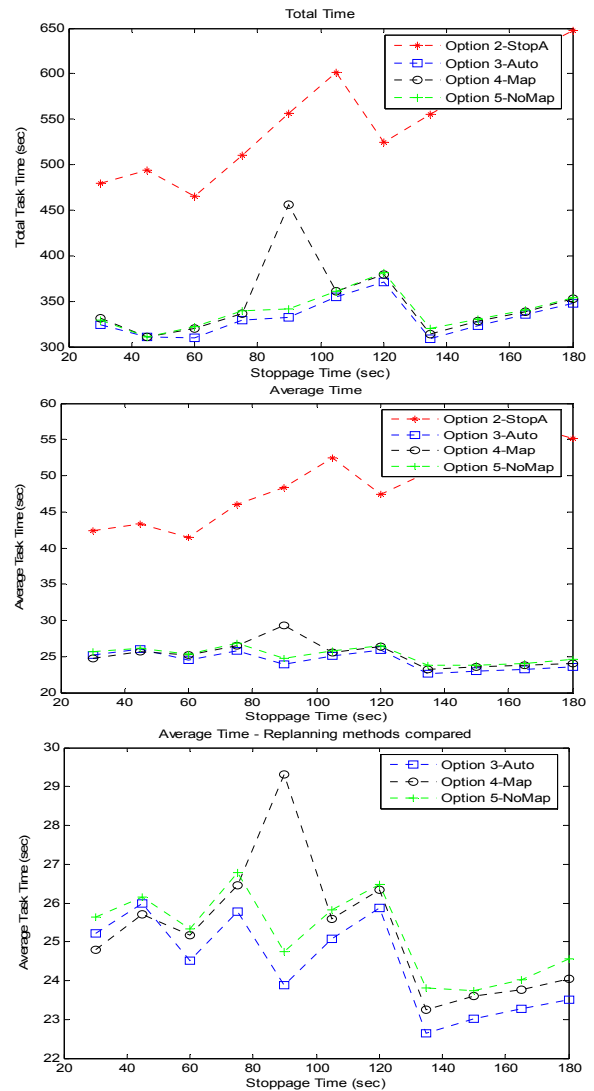


Figure 7. Case 2 Testing Results: (a) Total Time; (b) Average Task Time; (c) Average Task Time - among Options 3, 4 and 5

Overall, the map information is an invaluable inclusion into the replanning in regularly occurring situations whenever vehicles are stopped near nodes with high importance such as bottleneck nodes. The example below shows a simple example of the replanning decisions with and without map information. There are four vehicles and tasks in this example, each vehicle performs one task as listed in Table 1. Vehicle 4 is going to be stopped on node 129 (Fig. 8). This node is the most important node overall in regards to the probabilistic efficiency and bottlenecks created. While vehicle 4 is stopped no vehicles can move due to its impeding all paths. Subsequently, upon restart, all vehicles will experience replanning.

TABLE I. A SIMPLE EXAMPLE

Vehicle	Task	Source	Destination
1	1	113	125
2	2	133	127
3	3	90	148
4	4	109	137

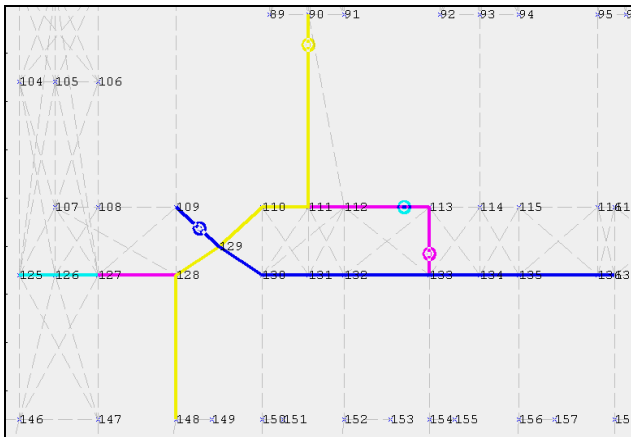


Figure 8. Vehicle 4 to Impede at node 129

Since vehicle 4 (performing task 4) still has many nodes, and much weight in its remaining path, once it is restarted, it is ranked low according to most replanning Methods. This means that if map information was not used then it would be replanned last, and hence no other vehicles would be able to plan until vehicle 4 was finally planned. Without the use of map information, it is significantly time wasting to try and plan for vehicles when one vehicle has a high probability of being in a position to block other vehicles. If Option 3 or 4 was used then due to the coefficient of importance of node 129, vehicle 4 would have its priority significantly elevated to a position where it would plan first, hence allowing other vehicle to replan their paths efficiently.

V. CONCLUSION AND REMARKS

Path and motion replanning for autonomous vehicle has been done in correlation with the map analysis and other Methods. Simulation studies show that the presented five Methods improve efficiency for replanning in particular when culminated (as in Option 3) as it automatically considers the five replanning Methods for an application and utilises map

information. Those Methods and Options have been built into working modules and simulated using the developed environment. It has allowed for the ability to create and demonstrate a more intelligent reactive replanning algorithm.

ACKNOWLEDGMENT

This work is supported in part by the ARC Centre of Excellence programme, funded by the Australian Research Council (ARC) and the New South Wales State Government, Australia.

REFERENCES

- [1] I. F.A. Vis, "Survey of research in the design and control of automated guided vehicle systems", *European Journal of Operational Research* 170 677-709, 2006
- [2] L. Qiu, W.J. Hsu, S.Y. Huang and H. Wang, "Scheduling and Routing Algorithms for AGVs: a survey", *International Journal of Production research*, Vol. 40, No. 3 745-760, 2002
- [3] M. S. Boddy, B. H. Bennett, B. A. Isle, R. A. Isle, "NASA Planning and Scheduling Applications: Emerging Technologies and Mission Trends", Final Report, 28/03/2004, Adventium Labs, USA, 2004
- [4] A. Stentz, "Optimal and efficient path planning for partially-known environments", *Proceedings of the IEEE International Conference on Robotics and Automation*, May 1994
- [5] D. Ferguson and A. Stentz, "The Delayed D* algorithm for efficient path replanning", *Proceedings of the 2005 IEEE International Conference on Robotics and Automation Barcelona*, Spain, (2005) 2057-2062
- [6] R. Philippsen, and R. Siegwart, "An interpolated dynamic navigation function", *Proceedings of the 2005 IEEE International Conference on Robotics and Automation Barcelona*, Spain, April 2005, 3793-3800
- [7] Y. Qin, D. Sun, N. Li, Y. Cen, "Path planning for mobile robot using the particle swarm optimization with mutation Operator", *Proceedings of 2004 International Conference on Machine Learning and Cybernetics*, 2004, Volume: 4, 26-29 Aug. 2004, 2473 - 2478
- [8] R. Chiba, J. Ota and T. Arai, "Integrated design for AGV systems using cooperative co-evolution", *Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, September 28-October 2, 2004, Sendai, Japan, 3791-3796
- [9] A. Stentz and M. Hebert, "A complete navigation system for goal acquisition in unknown environments". *Autonomous Robots* 2(2):127-145, 1995
- [10] M. Likhachev, D. Ferguson, G. Gordon, A. Stentz, S. Thrun, "Anytime Dynamic A*: An Anytime, Replanning Algorithm", *American Association for Artificial Intelligence*, 2005
- [11] A. Stentz, "The Focussed D* Algorithm for Real-Time Replanning". *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 1995
- [12] S. Koenig, D. Furcy and C. Bauer, "Heuristic search-based replanning" *Proceedings of the International Conference on Artificial Intelligence Planning and Scheduling*, 294-301, 2002
- [13] S. Koenig and M. Likhachev, "Improved fast replanning for robot navigation in unknown terrain" *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2002
- [14] D.K. Liu, X. Wu, A.K. Kulatunga, G. Dissanayake, "Motion coordination of multi-autonomous vehicles in dynamic and strictly constrained environments", *IEEE Intern. Conf. on Cybernetics and Intelligent Systems*, June 2006, Thailand.