A NEW EFFICIENT RESCHEDULING METHOD USING A-STAR ALGORITHM FOR TRAIN TRAVEL MANAGEMENT SYSTEM

M.S. Saravanan*, Abhishek Kumar
Professor, Department of CSE & IT, Saveetha School of Engineering, Saveetha University, Chennai, India
IV CSE – A, Department of Computer Science and Engineering, Saveetha School of Engineering, Saveetha University, Chennai, India
Email: saranenadu@gmail.com

Received on 10-08-2016
Accepted on 06-09-2016

Abstract

This paper deal with rescheduling the process of train travel management system. In order to so reduce the delays of train when accidents, natural disasters or technical problems occur on railway lines. An algorithm for automatic train rescheduling with a train rescheduling pattern system. We purpose a rescheduling algorithm of train groups to reschedule trains, instead of previous just waiting. A Petri-net-based conflict resolution algorithm adapted from the A-star algorithm is designed to search for an optimal or a near-optimal realistic schedule. The algorithm takes into account the railway working principles when generating new characterizing, so that the new schedule has less train delays and respects the safety principles. An evaluation principle based on the extent of the interference towards the original timetable and in a particular time is designed to select the optimal path for each train. Show that the rescheduling algorithm is able to reduce the delays of train time and make the parallel processing, make a contact of the real-time response of the train-group rescheduling well.

Keywords: Rescheduling train, Heuristic search, petri-nets, A-star algorithm, Train travel management.

1. Introduction

In railway transportation system, train be on time is a very important attribute. When people choose to commute by train, they think to arrive at their destination at the scheduled time. During the traffic control system they predicted events, such as train delays and blockage of some track, may be effect on the scheduled train time operations. Advanced real-time rescheduling and rerouting train systems should be developed to update train on time and dynamic use of tracks [1].

However, when disruptions happen, the control station masters need to reshuffle train orders, make unplanned stops, reroute the train and even delay or cancel scheduled train services. In practice, train rescheduling activities do not
start from scratch [2]. Rather, the new temporary timetable is obtained based on the existing schedule, of which the information is collected from the time graph. Rescheduling process may require adjusting the train time schedules, adding or cancelling a service trip of a scheduled train, changing the platform [3], and relocating the stops or any other necessary changes. As the process is very complex and also it will consuming a more time, if it can be done manually by human, therefore, mathematical models gives a resolve technique which saves time in the process of searching for the optimal solution. At present expert train dispatchers are sophisticated in train rescheduling [4]. They regularly monitor daily train schedule operations nearly at their post. Once an accident that likely cause interruption of transport services occurs, train dispatchers gather information about disruption and prepare the plan based on their previous train schedule. It is required to introduce an A-star algorithm for automatic train rescheduling. In order to maintain stable transport services, railway traffic management system require a mechanism which protect the quality of the plan [5]. Recently computer systems support in operational services to assist train dispatchers in charge of train rescheduling.

In this paper, we purpose an algorithm for automatic train rescheduling. Our algorithm is artificial heuristic search, which is based on a ‘A-star algorithm’. The rest of paper is designed as follows: In section 2, we introduce the related work on A-star algorithm for reschedule a train. Section 3: How to design a reschedule algorithm to avoid a train delayed or cancelled cause by some natural disaster or railway traffic blockages.

2. Related Work
Rescheduling is a time reallocation of resources for work completion when the initial scheduling cannot meet demands in completing a task for deterministic or theoretic reasons, for example, the railway road damages. Rescheduling is directed towards produced exact objective while maintaining utility of safety and special constraints. Some general scheduling purposes are cost minimization and maximization of customer satisfaction. The most common rescheduling objective is to recover the target initial schedule.

Even though there are many rescheduling scenario on security operation and rescheduling of train, most of them aim at some special situation, for example, natural disaster and railway traffic damages and so on. We take the specific situation of Indian railway into account, and refer to the rescheduling of different types of train groups running on the railway, not only including the transportation trains, but Express trains and local train. Additionally, we implement real time rescheduling train for real-time control of train schedule.

2.1 Train Rescheduling Pattern

Some railway control management system recognizes specifiable parts of the expert’s knowledge as the patterns to restore effectively damaged railway traffic. Each train dispatcher has own strategy in preparation of the plan, which effect variations in the plans. For example, under the same conditions, a dispatcher follow a plan by which disrupted railway traffic capable as possible while another dispatcher selects a plan that maintains sufficient railway track capacity [6]. In order to provide secure and accurate transportation services to public, every train dispatcher needs to own common and appropriate strategies for train rescheduling. Railway systems accordingly have taken a step under which is from every train dispatcher utilizes the scheduled patterns. In figure 1 shows that real optimization of rescheduled trains.

![Diagram](image)

**Fig1: Real-time Schedule.**

3. A-Star Algorithm for Rescheduling Trains

The rescheduling train strategy we design is a kind of A-star algorithm referring to heuristic search. When there is an emergency or any railway damages on the railway traffic, the scheduled trains cannot operate successfully on the railway given to the initial timetable, so different proper path have to be available for each involved train, or they have to wait and delay. So how to get the different proper path is the fully difference between heuristic algorithms and traditional search, for example, the DFS and BFS algorithms. Heuristic search tends to choose the next search node from the current node according to a A-star algorithm function, which we can get the right result in a shorter
time, instead of blindness search of DFS and BFS. A-Star algorithm has been established to be one of the most
efficient A-star algorithms to get the shortest path in fixed railroad network, which has been generally used on
solving shortest path and strategy designing problems. We represent’s’ as the start node and ’t’ as the destination node,
and ‘n’ as current node between s and t[7]. The core part of, given an equation of A-Star algorithm is the following
formula:
\[ f(n) = g(n) + h(n) \]
f (n) is the approximated cost from s to t passing n, g(n) is the absolute values from s to n, h(n) is the approximated
values from n to t. So how to design the h (n) determines whether the heuristic search algorithm is the A-Star
algorithm. And the sufficient conditions are as follows:
1) There is the perfect path from s to t in the search tree.
2) Problem domain is fixed.
3) The search values of all the nodes and child nodes are above zero.
4) h(n) <= h*(n) (h*(n) is the absolute values from n to t.
The ability to search for the optimal shortest path is mostly determined by,
The function h(n). When the g(n) is definite, the less the h(n) is, the less the f(n) will be, which can guarantee the
search direction towards the destination node as fast as possible. A good evaluation of h(n) is that h(n) is below the
lower bound of h*(n), and be close to h*(n) as much as possible[8]. Take the real time schedule operation on the
railway system; the shortest path between nodes generally cannot meet the actual need to be a perfect another path for
involved trains.
So we have to choose the kth shortest path (we tag k=1, 2, 3 …k) instead. Now we determine the h(n) as the length of
shortest path from s to n. Then the heuristic function can be express to the following formula:
\[ g(x) = x\text{.len}; h(x) = lsp(x.n); \]
\[ f(x) = g(x) + h(x) = x\text{.len} + lsp(x.n), \]
x represents the path from s to some node, which we tag x.n. lsp(x.n) represents the length of shortest path from the
node n to t; and the length of the path notes x.len. We can demonstrate the formula is a kind of A-Star algorithms.
According to the above-mentioned sufficient condition of A-Star algorithms, the demonstration is as follows:
1) The best path for each train exists in the actual rail net;
2) The rail net is surely finite;
3) The actual cost for each search path is surely above zero since the weight of line is above zero;

4) \( h(x) \) is one of \( h^*(x) \) in itself, so of course \( h(x) \leq h^*(x) \)

Considering the complexity of implementing and coding, we choose the Dijkstra algorithm and priority queue to obtain the lsp(n).

```plaintext
function A*(start, goal)
  // The set of nodes already evaluated.
  closedSet := {}
  // The set of currently discovered nodes still to be evaluated.
  // Initially, only the start node is known.
  openSet := {start}
  // For each node, which node it can most efficiently be reached from.
  // If a node can be reached from many nodes, cameFrom will eventually
  // most efficient previous step.
  cameFrom := the empty map
  // For each node, the cost of getting from the start node to that node.
  gScore := map with default value of Infinity
  // The cost of going from start to start is zero.
  gScore[start] := 0
  // For each node, the total cost of getting from the start node to the goal
  // by passing by that node. That value is partly known, partly heuristic.
  // For the first node, that value is completely heuristic.
  // Score[start] := heuristic_cost_estimate(start, goal)
  while openSet is not empty
    current := the node in openSet having the lowest fScore[] value
    if current = goal
      return reconstruct_path(cameFrom, current)
    openSet.Remove(current)
    closedSet.Add(current)
    for each neighbor of current
      if neighbor in closedSet
        continue        // Ignore the neighbor which is already evaluated.
      // The distance from start to a neighbor
      tentative_gScore := gScore[current] + dist_between(current, neighbor)
      if neighbor not in openSet     // Discover a new node
        openSet.Add(neighbor)
      else if tentative_gScore >= gScore[neighbor]
        continue        // This is not a better path.
      // This path is the best until now. Record it!
      cameFrom[neighbor] := current
      gScore[neighbor] := tentative_gScore
      fScore[neighbor] := gScore[neighbor] + heuristic_cost_estimate(neighbor, goal)
      return failure
    function reconstruct_path(cameFrom, current)
      total_path := [current]
      while current in cameFrom.Keys:
        current := cameFrom[current]
        total_path.append(current)
      return total_path
```

**Fig-2: Pseudo code of A-star algorithm.**

The initial state, \( x.n = s \); \( x.len = 0 \); every time, we pop the x whose has the minimal \( f_x \) out of the priority queue, and then get the next state according to the edges whose start node is from \( x.n \), push those edges into the queue; then when we meet the \( x.n == t \) for the first time, the shortest path from \( s \) to \( t \) arises, whose length is \( f_x \); similarly, when it is the \( k \)th time to meet \( x.n == t \), we get the \( k \)th shortest path from \( s \) to \( t \).
2.3 Evaluation Criteria

When we are searching the right methods for the trains that are unit concerned within the railway timetable, there typically train unit has many correct methods for the concerned trains to decide on [1]. Therefore the way to choose the best path for every concerned train is usually determined by what the analysis criterion of best path is. It’s typically celebrated that some common programming objectives area unit value reduction and maximization of client satisfaction, which the foremost common rescheduling objective is to recover the target initial schedule. That the analysis criteria of best methods we tend to style area unit as follows:

1) The more matching with the original timetable, the matching factor is identify as Q. 2) The limited time values of the kth shortest path for each train. The general time value is tagged as T. When we schedule the concerned trains, there could also be the conflicts of attempting to fulfil to the above named criterions. To avoid true that the changeable timetable affects an excessive amount of alternative trains’ operation, we have a tendency to take the primary criterion under consideration before the second [2]. The time value of rescheduling T chiefly is that the time spent on the rescheduled railroads for the train. It refers two problems:

1) Get the kth shortest path.
2) The time cost for the competition of trains on the same track.

The kth shortest path problems can be identified using above mentioned A-star algorithm using heuristic search; as to another problem. We can evaluate it, according to the following conflicts resolution strategy. We tag T as the total time cost of rescheduling for each train, and ET as the time cost that the handling of the railroad emergencies spend, Q as the factor of matching with the original timetable, we can simply set the Q as the number of stations of kth shortest path for each train.

4. Conclusion

We proposed A-star rescheduling algorithm of train timetable based on heuristic algorithm to deal with the time loss due to the railway traffic damages or natural disaster. The rescheduling train strategy based on the A-Star algorithm in heuristic search chooses proper paths for the involved trains. An evaluation pattern based on the term of the disturbance towards the given a original timetable and limited time is designed to select the optimal path for each train schedule. And a conflict resolution strategy is designed to deal with the path and time conflicts among trains when rescheduling. In the end, Due to the real-time needs of the railway traffic scheduling, we implement the parallelization for the train-group rescheduling referring to the division of rail net. The rescheduling strategy can be
optimized to improve the efficiency of the rescheduling algorithm and in the process of rescheduling; we ignore some uncontrollable factors that may impact the time cost of rescheduling. In our future work, we will continue to optimize the rescheduling algorithm to get a better efficiency on the process of search proper paths. And it is also necessary to design a new model to take into the above mentioned uncontrollable factors to improve the accuracy of rescheduling of train groups. We will focus a lot on them in the future.

References


