Quickest paths on congested networks: some special cases

Giovanni Andreatta   Luigi De Giovanni   Giorgio Salmaso

Department of Pure and Applied Mathematics, University of Padova
via Trieste, 63 - 35121 Padova (Italy)

Abstract

Determining the shortest path for a vehicle moving on a network is an easy task. If more vehicles share the same network resources and we want vehicles to reach their destinations as soon as possible, moving on shortest paths might not be the optimal solution. Conflicts may arise if more vehicles need the same resource at the same time, requiring some vehicles to stay idle at some point of the shortest path: perhaps, a path different from the shortest one may be quicker. The problem arises in many contexts, like the coordination of automated guided vehicles, the management of airport groundside traffic etc. In this work we consider the problem of finding the quickest paths on congested networks and we will present the results of a first study. Some special cases are derived and related optimal dispatching algorithms are given.

Keywords*: network routing, quickest paths, conflict management, exact dispatching algorithms.*

1 Introduction

We consider a routing network where several vehicles have to be moved from their origin to their destination. The network resources are nodes and arcs. Nodes corresponds to vehicle origins and destinations, route crossing points or points where vehicles are allowed to stop and stay idle. Arcs correspond to directed routes between couples of nodes. For each arc/node a capacity is given, corresponding to the maximum number of vehicle moving/idle on it. For each arc a fixed traversal time is also given, which does not depend on the time we enter the arc nor on the number of vehicles moving on it. We want to determine a route for each vehicle in order to let it reach the desired destination *as quickly as possible*, as we will detail later. The problem arises in many contexts: the routing of automated guided vehicles in container terminals [2], the coordination of ground service vehicles in airport aprons [6], aircraft taxiing operations [4] etc. Our research has been stimulated by the preparatory work for the project "Integrated Airport Apron Safety Fleet Management - AAS", funded by the European Community in the Seventh Framework Programme (grant agreement 213061), aiming at improving the efficiency and the control of airport groundside movements by advanced vehicle and staff management.

A possible solution to the proposed problem is to easily compute shortest paths on the graph underlying the routing network and to route vehicles on those shortest paths. In this case, conflicts may arise: several vehicles may be at the same node or arc at the same time and the capacity constraints might be violated. Vehicles may be idle at some node of the shortest path for some time, waiting for the network resources to be available. It follows that moving on shortest paths may not be the optimal choice. In this work we consider the problem of coordinating by a single centralized dispatcher a set of vehicles on a routing network. The dispatcher has to determine the schedule of the vehicles on the network, that is, to determine, for each vehicle, *what* resources to use (the routing) and *when* to use them.

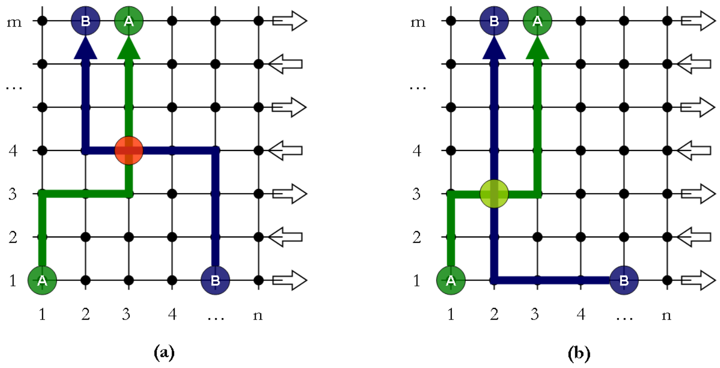

Figure 1: Alterative routing on a grid network.

The problem has been the object of several studies, proposing different approaches. For example, [3] considers a simulation method to determine the schedule of aircraft moving on airport aprons; [5] proposes a metaheuristic approach to the same problem, using genetic algorithms; [1] uses queueing models to analyze the traffic of airport groundside vehicles; [4] and [6] use Mixed Integer Linear Programming to model and solve the problem of coordinating aircraft taxiing operations and airport groundside traffic as a flow problem on time-expanded networks; [2] proposes a heuristic algorithm for routing automated guided vehicles in container terminals, based on dynamically solving a set of shortest path problems with time windows.

In this work we state the Congested network Quickest Path (CQP) problem and we present some preliminary results on some special cases. We start from the CQP problem on grid networks: we present a polynomial dispatching algorithm to solve the problem to optimality and we outline some directions for future research.

2 Problem statement in grid networks

Although the problem could be stated for general networks, in this work we will consider the special case of the *CQP problem on grid networks*. A grid network is defined by a graph $G = (N, A)$ where N is the set of nodes of the routing network and $A \subseteq N \times N$ is the set of directed arcs, such that the network is a planar grid with m rows and n columns (see figure 1). Columns correspond to vertical lanes. Each row corresponds to a horizontal lane and is also called *level*. Note that, for each node $i \in N$ horizontal and vertical coordinates $x(i)$ and $y(i)$ can be defined, corresponding to the horizontal and vertical lane of the node itself: node i can be identified by the pair $\langle x(i), y(i) \rangle$. Traversing a horizontal or vertical arc takes one time unit. Therefore, we consider only discrete times $t = 0, 1, 2 \dots$. Vehicles are allowed to stop at any node in the network; stopping on arcs is not allowed. For each node and arc a capacity is defined: no more than one vehicle can stay simultaneously idle on a node, and an arc can be traversed by one vehicle at a time. Also, we assume that horizontal lanes allow alternate one-way movements. We consider a set of vehicles V moving on the grid network. Given a vehicle $v \in V$ we denote by $s_v \in N$ its origin and by $d_v \in N$ its destination. All vehicles starts at time 0 and are initially placed at the same level, corresponding to the bottom of the grid (row 1). The destination of all vehicles lays on the highest level (row m). No two vehicles have the same origin, nor the same destination. We want to route all vehicles from their origins to their destinations in such a way that node and arc capacity constraints are satisfied and a given performance measure is optimized. The performance measure is related to the time spent to route all the vehicles on the network: for example minimizing the sum of all the vehicles arrival times, or the maximum over all the vehicle arrival times.

In the following, we will concentrate on this simplified CQP problem and we will derive a polynomial algorithm to solve it to optimality.

3 A polynomial dispatching algorithm for grid networks

It is easy to observe that the length of a shortest path between any two nodes i and j on the grid network corresponds to their *Manhattan distance*: $|x(i) - x(j)| + |y(i) - y(j)|$. Every path with length equal to the Manhattan distance is called *Manhattan path*.

Observation 1. *If we are able to route all vehicles on one of the Manhattan paths* without stopping*, we obtain an optimal solution to the CQP problem on grid networks.*

In fact, routes would have a number of vertical and horizontal movements as small as possible and vehicle would never stay idle at any nodes. Note that the solution would be optimal with relation to both the performance indexes: the sum of all the vehicles arrival times and the maximum over all the vehicle arrival times.

Of course, conflicts may arise. For example, if we consider the routes in Figure 1a, both vehicles A and B would arrive at the node $\langle 3, 4\rangle$ at the same time 5. The conflict may be resolved, for example, by stopping vehicle A at node $\langle 3, 3\rangle$ for one time unit, but we may loose the optimality. Or we may route A and B according to the figure 1b, where A and B cross node $\langle 2, 3\rangle$ at time 3 and 5 respectively, so that vehicles do not need to stop and the routing optimality is preserved. The question is thus the following: is it possible to choose Manhattan paths and to avoid any node or arc conflicts?

We observe that every Manhattan path for B contains more horizontal steps than A and, in the conflict-free solution, B is always *below* A. Generalizing this observation, we would give priority to the horizontal movement of the vehicles with a *high enough* number of horizontal steps remaining. For the sake of the definition of an algorithm based on this idea, we introduce the following notation. Given a vehicle $v \in V$ and a time index t:

- $H_0(v)$ is the initial horizontal distance, that is the number of horizontal steps in *any* Manhattan path from s_v to d_v: $H_0(v) = |x(d_v) - x(s_v)|$;
- $p_t(v) \in N$ is the position (node) of vehicle v at time t;
- $H_t(v)$ is the current horizontal distance: $H_t(v) = |x(p_t(v)) - x(d_v)|$;
- $L_t(v)$ is the current level of vehicle v: $L_t(v) = y(p_t(v))$;
- $concordant_t(v)$ (resp. $discordant_i(v)$) is *true* if $H_t(v) > 0$ (horizontal steps remaining) and v is on a horizontal lane allowing (resp. not allowing) the horizontal step towards the desired destination, *false* otherwise.

The following dispatching algorithm determines the position of each vehicle in terms of node crossed at time $0, 1, 2\ldots$; a vehicle v is thus assumed to traverse the arc $(p_t(v), p_{t+1}(v))$ in the time interval $[t, t+1]$ (the traversal time of any arc is one).

Dispatching algorithm for grid networks

```
t := 0; for each v ∈ V, set p_0(v):=s_v
Repeat
    M := max_{v∈V} {L_t(v)}
    K := max {H_t(v) : discordant_t(v) = true and L_t(v) = M}
    for each v ∈ V
        if concordant_t(v) then
            if L_t(v) < M then p_{t+1}(v) := push_HOR(t,v)
            else if H_t(v) ≥ K then p_{t+1}(v) := push_HOR(t,v)
            else p_{t+1}(v) := push_VER(t,v)
        else p_{t+1}(v) := push_VER(t,v)
    t := t + 1
Until p_t(v) = d_v, ∀ v ∈ N
```

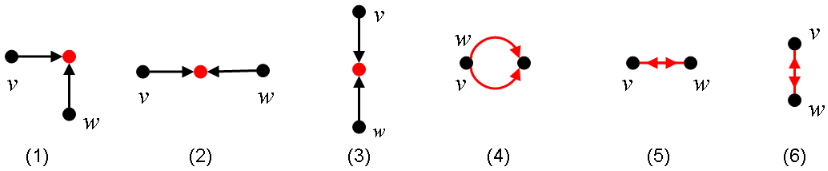

Figure 2: Potential conflicts on a grid network.

The function `push_HOR(t,v)` moves a vehicle v horizontally; `push_VER` moves v vertically, or let v stay idle if it is already at its destination. The algorithm moves horizontally all concordant vehicles, except those on the last occupied level which distance is less than the threshold K. In fact, this threshold allows us to *give priority* to the vehicles with *high enough* horizontal steps remaining, as observed before. All other vehicles are pushed vertically: the horizontal movement is not possible (discordant vehicles) or would cause a non-Manhattan path (vehicles aligned with their destinations).

The Dispatching Algorithm (DA) solves the CQP problem on grid networks to optimality. In fact, we can formally show that no conflicts are generated and that vehicles are routed on one Manhattan path without stopping: in this short paper, we will just sketch the proof.

Figure 2 illustrates all potential types of conflicts arising on a grid network. We observe that: the grid contains one-way horizontal lanes and conflicts of type 2 and 5 are excluded; the DA never pushes vehicles down and conflicts of type 3 and 6 are avoided; conflicts of type 4 are excluded if node conflicts 1, 2, and 3 are excluded. Therefore, we have just to show that conflicts of type 1 never occur.

We observe that, once a vehicle starts moving horizontally, it is concordant and below the highest occupied level. The DA will push it horizontally until it reaches the column of its destination. In fact, we can show that:

Lemma 1. *All horizontal moves of a given vehicle take place at the same level.*

If a conflict of type 1 between vehicles v and w occurs at time t, then $L_{t-1}(w) < L_{t-1}(v)$. As all vehicles starts at the same level, there has been a time before $t-1$ when w moved horizontally while v continued moving vertically. At time $t-1$, w is moving (again) vertically, so that, according to Lemma 1, it is vertically aligned to its destination, that is, $H_t(w) = H_{t-1}(v) = 0$. Also, v is moving horizontally meaning that $H_{t-1}(v) > 0$ and $H_t(v) \geq 0$. Actually, $H_t(v)$ cannot be 0, otherwise v and w would have the same destination, which contradicts our assumptions. We just showed that:

Lemma 2. *If a conflict of type* 1 *between vehicles v and w arises at time t, then $H_t(w) = 0$ and $H_t(v) > 0$.*

The two lemmas allow us to prove the following property:

Property 1. *The Dispatching Algorithm does not generate conflicts.*

Proof. Let us assume a conflict of type 1 occurs at time t and at level l between vehicles v and w. As DA never let any vehicle idle, it means that both v and w have performed t moves: l vertical moves and $t-l$ horizontal moves. From Lemma 2, $H_0(w) = t-l$ and $H_0(v) > t-l$, that is, $H_0(v) > H_0(w)$. As observed before, there should be a level d corresponding to a time d when w started moving horizontally and v continued moving vertically. We have $H_d(w) = H_0(w)$ and $H_d(v) = H_0(v)$, that is, $H_d(v) > H_d(w)$. We have two cases: if v and w were discordant at level d, then $H_d(w) < H_d(v) \leq K$ and v could not move horizontally; if v ad w are both concordant at level d, then, if the DA pushes w horizontally, then v too should be pushed horizontally, as $H_d(v) > H_d(w) \geq K$. In any case, we have a contradiction, proving that the DA excludes conflicts of type 1. As observed before, conflicts of type 2, 3, 5 and 6 never occur and this excludes also conflicts of type 4. □

We observe that at each iteration, the DA pushes each vehicle on one of its Manhattan paths without stopping. Also, no conflicts are generated and, according to Observation 1, the routings are optimal.

Concerning the computational complexity, the number of iterations is bounded by $m + n$, the length of the longest Manhattan path on a $m \times n$ grid. At each iteration, $|V|$ push operations are performed in constant time and the algorithm converges in $O(|V|(m + n))$. It is possible to show that K is strictly decreasing at each iteration. Observing that $K = 0$ means that all vehicles are vertically aligned to their destination, the number of necessary horizontal levels is limited by the maximum initial horizontal distance (plus 2, actually), which in turn is at most n. Also, the number of vehicles is bounded by n, as no two vehicles can start at the same node. We proved that:

Property 2. *Given an $m \times n$ grid, the DA converges in polynomial time $O(n^2)$ to an optimal solution of the CQP problem on grid networks.*

4 Future work

The Congested network Quickest Path (CQP) problem has been considered. It has relevant applications, for example, to the routing of Automated Guided Vehicles or to the dispatching of airport groundside equipment. In this abstract we have considered the CQP problem on grid networks with some assumptions on the grid layout and on the initial vehicle arrangement. For this special case, we have proposed a conflict-free Dispatching Algorithm (DA) able to solve the problem to optimality in polynomial time. The algorithm is very efficient and could be implemented by a centralized real-time dispatcher. The efficiency and simplicity of the DA may suggest to organize the routing network as a grid with one-way horizontal lanes. This should be applicable, especially for cases where a new network for Automated Guided Vehicles has to be settled. The remaining assumptions, in particular the one concerning the horizontal alignment of vehicle starting positions, may be too restrictive. In this case, we may use the DA as a building block of a procedure working as follows: first, vehicles are guided towards a common horizontal level, then the DA is applied. Of course the first phase is a difficult task: the problem should be further studied, and, in case, heuristics may be devised. For example, in order to reduce the time needed to align vehicles, the heuristic may partition vehicles in subsets having destinations lying on disjoint intervals of vertical lanes; then each partition is pushed towards a common level (which may be different for different sets) in the related vertical lanes interval, and then the DA works separately for each subset. Even under these hypotheses, the assumptions remain restrictive and we are interested in extending the results to more general CQP problems. For example, the extension to grid networks with arbitrary destination levels is trivial, as the same DA works. Non-trivial are the extensions to more general cases, including grids with arbitrary vehicle origins and destinations (useful also to model non-synchronized starts), non-complete grids (modeling possible non-transit zones within the routing area), two-ways horizontal lanes (when the one-way assumption cannot be implemented) etc. Ideally, we would like to consider the CQP problem on arbitrary networks. This is the object of ongoing and future research.

References

[1] K. Andersson, F. Carr, E. Feron and W.D. Hall, *Analysis and Modeling of Ground Operations at Hub Airports*, proceedings of *Air Traffic Management R&D Seminar, Napoli, June 13-16, 2000.*

[2] E. Gawrilow, E. Köhler, R.H. Möhring and B. Stenzel, *Dynamic routing of automated guided vehicles in real time*, in *Mathematics: Key Technology for the Future. Joint Projects between Universities and Industry 2004-2007*, W. Jäger and H.J. Krebs, eds., Springer, London, 2008, pp. 165–178.

[3] J.B. Gotteland, N. Durand, J.M. Alliot and E. Page, *Aircraft Ground Traffic Optimization*, proceedings of *4th International Air Traffic Management R&D Seminar, December 3-7, 2001.*

[4] A.G. Marin, *Airport management: taxi planning*, Annals of Operations Research **143** (2006) pp. 191–202.

[5] B. Pesic, N. Durand and J.M. Alliot, *Aircraft Ground Traffic Optimization using Genetic Algorithms*, proceedings of *GECCO, San Francisco, July 7-11, 2001*.

[6] P.C. Roling and H.G. Visser, *Optimal Airport Surface Traffic Planning Using Mixed Integer Linear Programming*, International Journal of Aerospace Engineering (2008) doi:10.1155/2008/732828.