

# Shortest Path Computations under Trajectory Constraints for Ground Vehicles within Agricultural Fields

Mogens Graf Plessen and Alberto Bemporad

**Abstract**—This paper addresses the task of finding the shortest path to a target point on the boundary of an agricultural working area given a current position and heading of a ground vehicle within that area. Constraints such as admittance of turning on the spot, lane- and corridor-constraining motion as well as repressed area minimization are taken into account. We distinguish between orchard- and vineyard-like areas and agricultural fields growing, in particular, rapeseed and cereals. For the former application, dynamic programming and label-correcting algorithms are compared and, based on a coordinate transformation, a heuristic is motivated therefore. For the latter agricultural field application, we constrain the shortest path objective by allowing agricultural machinery to only use already existing tractor-lane traces and thus introduce a repressed area minimization constraint. Therefore, a customized and novel shortest path finding method is derived, before its optimality is proven. The outcome of this work is equally applicable for autonomous as well as manually driven agricultural ground vehicles.

## I. INTRODUCTION

The agricultural sector is experiencing an increasing degree of automation in both the operation of agricultural machinery, e.g., autonomous guidance of tractors [1], [2], the processing of information [3], and the planning of logistical in-field and inter-field operations [4], [5], [6]. Within this context, this paper relates to in-field intelligent transportation systems and logistical optimization. It is applicable both to support a human driver by providing navigation guidance, as well as for the high-level path planning task in a fully autonomous tractor system.

We distinguish between two general types of plantable spaces, in the following referred to as *orchard-like areas* and *agricultural fields*. The difference is that for the former, the crop grows on bushes, vines or trees, whereby for the latter the crop is harvested on the ground and, specifically, there also exist headlands equally employed for crop growth. Examples for the former include orchards and vineyards, whereas the latter comprises in particular wheat and rapeseed plantation. For agricultural field operation we further focus on post-seeding operations such as fertilizing and spraying. The distinction between orchard-like areas and agricultural fields has implications for the operation of machinery, in particular, the importance of minimizing repressed ground area or unimportance thereof.

Shortest path planning is a well known topic in operations research. Popular approaches include dynamic programming

(DP) and label correcting algorithms (LCA) such as Dijkstra's method, and A\* as a label correcting variation [7].

For existing literature about in-field shortest path planning we refer the reader to [6] and [4]. They treat path planning of service units (SU), e.g., transport wagons, while they are supporting primary units (PU), e.g., harvesters, in the harvesting process. A two-stage optimization method is employed in [6]. In contrast, [4] extends this work to inter-field operation and reduces the optimization to one computational stage. Its focus is on graph modeling with Euclidean distances used as arc costs. For graph search, Dijkstra's method is employed and it is referred to [8] for its implementation. Conceptually, within our paper for the navigation in orchard-like areas, the same graph generation approach is taken as in [4]. However, we additionally motivate a heuristic based on a coordinate transformation that decreases the number of LCA-iterations required to find a solution. Additionally, we give details about our graph search algorithms customized to the given problem. The main contribution of this paper is the second part about shortest path navigation in agricultural fields, using only existing tractor-lane traces for minimization of repressed ground area. To the best of the authors' knowledge, both the application and solution approach have not been treated before. A customized shortest path algorithm is presented and its optimality proven.

This paper is organized as follows. In Section II and III we discuss the navigation in orchard-like areas and agricultural fields, respectively. The results of numerical experiments on real-world field-data are stated in Section IV, before concluding with Section V.

## II. NAVIGATION IN ORCHARD-LIKE AREAS

Shortest path planning is of relevance in the coordination of SUs and PUs, e.g., in the harvesting process. An alternative application is the navigation in vineyards and orchards. Relevance arises because of limited hectare coverage and the need for timely return to a depot or mobile station for refilling. For general route planning in orchards, see [9]. For experimental field results of an autonomous multi-tractor system that performs mowing and spraying operations in a citrus orchard, see [10]. An autonomous navigation system using a 2D laser scanner for straight line recognition of tree rows in an orchard application is presented in [11].

In the following, we distinguish between admitting for the ground vehicle an instantaneous turn on the spot (e.g., plant inspection with a small mobile unit) and vice versa the case of having vehicle motion constrained by small operation

MGP and AB are with IMT School for Advanced Studies Lucca, Piazza S. Francesco 19, 55100 Lucca, Italy, {mogens.plessen, alberto.bemporad}@imtlucca.it

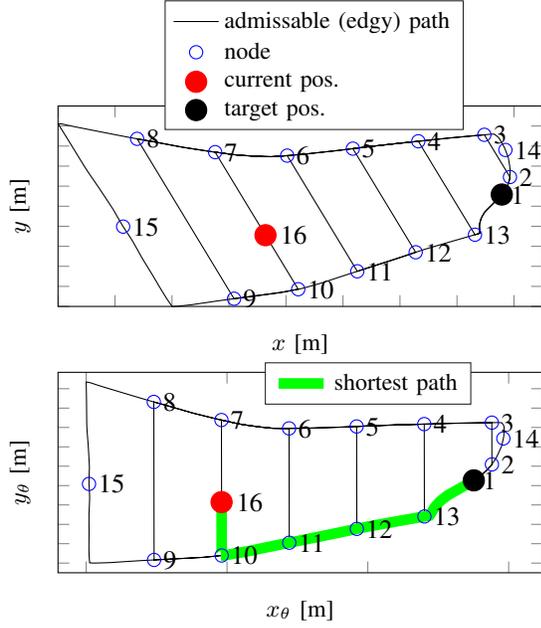


Fig. 1. Modeling of the transition graph  $T$  with  $N = 16$ , and illustration of the heuristic from Section II in a rotated coordinate  $(x_\theta, y_\theta)$ -frame.

corridors or because of larger towed implements prohibiting an instantaneous  $180^\circ$ -turn.

Let

$$\xi_i = [x_i \ y_i]^T, \quad i = 1, \dots, N$$

be the position of *node*  $i \in \mathcal{N} = \{1, \dots, N\}$  in the Universal Transverse Mercator (UTM)-coordinate system with  $x_i$  and  $y_i$  representing easting and northing-position, respectively. The traverse of an orchard or vineyard can be modeled as a concatenation of transitions between multiple nodes, see Figure 1. We further define *interior lane-segments* (e.g.,  $13 \rightarrow 4$ ,  $12 \rightarrow 5$ , etc.) and *perimetric lane-segments* (e.g.,  $9 \rightarrow 10$ ,  $10 \rightarrow 11$ , etc.). *Auxiliary nodes* (e.g., 15 and 14 in Figure 1) are introduced to ensure unique connections between any two nodes. A transition graph  $T$  can then be defined as

$$T_{ij} = \begin{cases} d_{ij}, & \text{if } \exists \text{ a direct admissible path } i \rightarrow j \\ \infty, & \text{otherwise,} \end{cases} \quad (1)$$

where  $d_{ij}$  denotes the pathlength in meters from node  $i$  to  $j$  and  $i, j \in \mathcal{N}$ . Throughout this paper, we label our current vehicle position (start) and target position (exit) by node  $s$  and  $e$  (e.g., nodes 1 and  $N$  in Figure 1), respectively. Ultimately, note that with respect to the definition, *interior lane-segments* do not necessarily have to be straights. They may also be curved and thereby shifted in parallel (*freeform*).

In case of *straight* interior lane-segments, let the admissible path-segments and all nodes be described in a new coordinate system defined by  $[x_{\theta,i} \ y_{\theta,i}]^T = R(\theta) [x_i \ y_i]^T$ , with  $R(\theta)$  being a standard rotation matrix with rotation angle  $\theta$ . We select  $\theta$  such that  $x_{\theta,m} = x_{\theta,n}$  for all  $m, n \in \mathcal{N}$  representing the end-nodes of all interior lane-segments,

thereby generating a canonical vertical lane grid as further visualized in Figure 1.

Let us define  $\Delta x_\theta = |x_{\theta,l} - x_{\theta,m}|$ , as the *operating width* of the agricultural machine, where nodes  $l$  and  $m$  represent *end-points* of two distinct straight interior lane-segments that are both connected via a perimetric lane-segment (i.e.,  $T_{lm} \neq \infty$ ). For freeform interior lane-segments, the operating width is defined similarly in a *curvilinear* coordinate system.

For the case of straight interior lane-segments, we abbreviate “*interior lane end-point*” by *ilep*, and further introduce variables  $x_\theta^{\max} = \max_{i \in \mathcal{N}} x_{\theta,i}$ ,  $x_\theta^{\min} = \min_{i \in \mathcal{N}} x_{\theta,i}$  and

$$x_\theta^{\text{last,max}} = \max_{i \in \mathcal{N}} \{x_{\theta,i} : x_{\theta,i} < \max\{x_{\theta,s}, x_{\theta,e}\}, x_{\theta,i} \text{ ilep}\},$$

$$x_\theta^{\text{last,min}} = \min_{i \in \mathcal{N}} \{x_{\theta,i} : x_{\theta,i} > \min\{x_{\theta,s}, x_{\theta,e}\}, x_{\theta,i} \text{ ilep}\}.$$

#### A. Admittance of Turning on the Spot

For determining the shortest path between node  $s$  and  $e$ , we initialize  $J_0(i) = T_{ie}$ ,  $i \in S_0$  with  $S_0 = \{i \in \mathcal{N} : T_{ie} \neq \infty\}$ , and define our DP-iteration as

$$J_k(i) = \min_{j \in S_{k-1}} \{T_{ij} + J_{k-1}(j)\}, \quad i \in S_k, \quad (2)$$

$$S_k = \{i \in \mathcal{N} : T_{ij} \neq \infty, j \in S_{k-1}\}, \quad (3)$$

which is terminated at a particular stage  $k$  on satisfaction of a criterion, that is further discussed at the end of this section. The shortest path length at DP-iteration stage  $k$  from node  $i$  to the exit node  $e$  is denoted by  $J_k(i)$ .

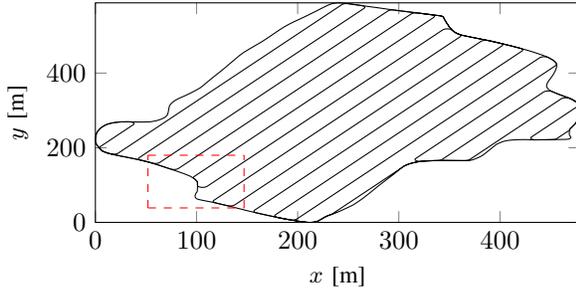
An alternative to DP is LCA [7]. Let  $d_j$  denote the label of node  $j$ , i.e., the pathlength from starting point  $s$  to node  $j$ . The general idea of all LCAs is to progressively find shorter paths from the start to every other node  $j$  and ultimately the target  $e$ . Characteristic is the test

$$d_i + T_{ij} \leq \min\{d_j, \text{UPPER}\}, \quad (4)$$

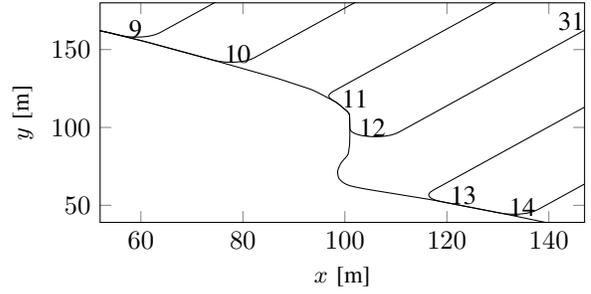
with  $\text{UPPER} = d_e$ . The satisfaction of (4) implies that a path from node  $s$  to  $j$  with  $i$  immediately before  $j$  is shorter than the current path from  $s$  to  $j$  and furthermore smaller than the currently shortest path from  $s$  to  $e$ . Thus, it is a candidate for being part of a shortest path from  $s$  to  $e$ . In case of satisfaction of (4), node  $j$  will be further examined at one of the following LCA-iterations for being a candidate for possible inclusion in the shortest path. Besides the sequence of analysis of candidate nodes (breadth-first and depth-first search, Dijkstra’s method, etc.), the adaptation of (4) by usage of heuristics (to constrain nodes added to the candidate list) is the main driver for reduction in computation times for the discovery of the shortest path.

In the following we derive a simple heuristic for our application in case of straight interior lane-segments. It can be generalized to the case of freeform interior lanes by transformation to a curvilinear coordinate system.

*Proposition 1:* Let the *shortest path* between current node  $s$  and target  $e$  be denoted by a list  $\mathcal{C} = [c_1, c_2, \dots, c_l, c_{l+1}, \dots, c_{l+L-2}, c_{l+L-1}]$  of nodes with  $c_1 =$



(a) Directed transition curves.



(b) Zoom-in and added numbering.

Fig. 2. Illustration of characteristic directed transition curves resulting from natural smoothing by an agricultural machine with limited turning radius.

$s$  and  $c_{l+L-1} = e$ . Then, it holds that

$$\begin{aligned} & \text{if } x_{\theta,s} \leq x_{\theta,e} : \\ & \left\{ \begin{array}{l} f^{\min} \leq x_{\theta,c} \leq f^{\max}, \forall c \in \mathcal{C}, \\ x_{\theta,c_{l+1}} \geq x_{\theta,c_l}, \forall 1 < l < l+L-2, \end{array} \right. \end{aligned} \quad (5)$$

$$\begin{aligned} & \text{if } x_{\theta,s} > x_{\theta,e} : \\ & \left\{ \begin{array}{l} f^{\min} \leq x_{\theta,c} \leq f^{\max}, \forall c \in \mathcal{C}, \\ x_{\theta,c_{l+1}} \leq x_{\theta,c_l}, \forall 1 < l < l+L-2 \end{array} \right. \end{aligned} \quad (6)$$

where  $f^{\min} = \max\{x_{\theta}^{\text{last,min}} - \Delta x_{\theta}, x_{\theta}^{\min}\}$  and  $f^{\max} = \min\{x_{\theta}^{\text{last,max}} + \Delta x_{\theta}, x_{\theta}^{\max}\}$ .

*Proof:* Let us consider the case  $x_{\theta,s} \leq x_{\theta,e}$ . The proof is analogous for  $x_{\theta,s} > x_{\theta,e}$ , and is by contradiction. By construction of the transition graph, at every node  $i \in \mathcal{N}$  with  $\{j \in \mathcal{N} : T_{ij} \neq \infty, i \text{ ilep}\}$ , there exists at least one  $j$  such that  $x_{\theta,j} \geq x_{\theta,i}$  and another  $j$  such that  $x_{\theta,j} \leq x_{\theta,i}$ . Thus, a traverse along the nodes connecting interior lane-segments of the transition graph in one of the two monotonous  $x_{\theta}$ -directions is always possible. Assume now the shortest path is such that at stage  $l$  and node  $c_l$  we move to node  $c_{l+1}$  with  $x_{\theta,c_{l+1}} < x_{\theta,c_l}$ . Then, to reach node  $x_{\theta,e} > x_{\theta,c_{l+1}}$ , the shortest path must pass level  $x_{\theta,c_l}$  at a stage  $l+h > l+1 > l$  with  $h > 1$  because of the connectivity of the graph. By the fact that all nodes  $\{i \in \mathcal{N}, i \text{ ilep}\}$  can be reached by traversing monotonously, we have a contradiction with respect to above assumption about the shortest path. The relation  $f^{\min} \leq x_{\theta,c} \leq f^{\max}, \forall c \in \mathcal{C}$  is then by construction a consequence considering additionally the auxiliary nodes. This concludes the proof. ■

Note that Proposition 1 does not explicitly address the transition directions from and to the starting and exit node, respectively. In general, no a priori rule can be applied since starting and exit nodes may also lie in between two interior lane end-points, and, e.g., in indentations, bays or along highly non-convex perimetric lane-segments.

Proposition 1 further implies that a shortest path from node  $s$  to  $e$  does consequently not include any turning on the spot, except at most one, at the very beginning at node  $s$ .

Proposition 1 can be employed as a heuristic for an efficient implementation of LCA. Thus, any node entering the candidate list must satisfy not only (4), but additionally (5) in case of  $x_{\theta,s} \leq x_{\theta,e}$  and (6) in case of  $x_{\theta,s} > x_{\theta,e}$ .

For DP, iterations (2) and (3) may be terminated once all nodes with  $x_{\theta}$  for which  $f^{\min} \leq x_{\theta} \leq f^{\max}$  holds have been covered. Alternatively, the DP-solutions for traversing from *all* nodes, except  $s$ , to the target node  $e$  can be stored offline in a look-up table. Then, online the two nodes neighboring the current position  $s$  are determined and the corresponding shortest paths starting from both of these neighboring nodes are looked up. The pathlengths from the current position are added respectively, and the shorter solution of the two is ultimately selected.

### B. Vehicle Motion Constrained by Operation Corridors

Shortest path planning under consideration of the current heading direction of the ground vehicle, and not permitting any turning maneuvers on the spot due to tight operation corridors, can easily be addressed by constraining the node immediately following after the starting node. Thus, besides  $c_1 = s$  and  $c_{l+L-1} = e$ , we additionally fix  $c_2$  within the shortest path as the node towards the vehicle is invariably heading given the current starting position and orientation.

## III. NAVIGATION IN AGRICULTURAL FIELDS

### A. Repressed Area Minimization Constraint

For the coverage of agricultural fields growing, e.g., wheat, rapeseed and similar crops, the transition from an interior lane-segment (lane) to the perimetric tractor-lane (headland path) is in practice *smoothed* by the agricultural machinery, mostly due to its limited turning radius. For autonomous tractor guidance, a corresponding trajectory design approach is presented in [12]. Once a coverage path has been established, see Figure 2 for illustration, it is reasonable to always follow it at every working iteration on that specific field. The reason is repressed area minimization. Consider Figure 2(b). Suppose instead of driving along the established traces  $31 \rightarrow 12 \rightarrow 11$ , the ground vehicles covers  $31 \rightarrow 12 \rightarrow 13$ . Then, at the interior lane-segment  $31 \rightarrow 12$  a small new tractor trace would be created, thereby repressing the crop at that location. While it may appear negligible at first sight, it is avoidable and may become very relevant in the sum for all lanes. This is since  $L_{\text{loss}} = 2 \cdot l_{\text{repr}} \cdot w_t \cdot g_{\text{gain}}$ , where  $L_{\text{loss}}$  [\$],  $l_{\text{repr}}$  [m],  $w_t$  [m] and  $g_{\text{gain}}$  [\$/m<sup>2</sup>] represent total monetary loss due to area repression by tractor traces, the

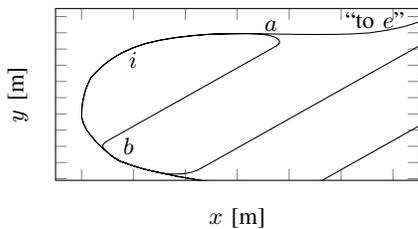


Fig. 3. Illustration of the deadlocking principle resulting from the repressed area minimization constraint.

total length of repressing tractor traces, the tire width, and the normalized gain for a particular crop, respectively.

The finding of shortest paths under the *repressed area minimization constraint* may become necessary for the refilling of fertilizers and spraying applications at a farm basis or a mobile depot located outside the agricultural field boundary. In the following, we present a solution approach.

### B. Solution Approach and Algorithm

Let the target node (exit) be located somewhere along the perimetric lane and the current tractor position anywhere along the tractor traces. We create a transition graph as in (1). Additionally, we store all admissible transitions composed of three nodes in a list  $\mathcal{U}$ . For example, regarding Figure 2(b), transitions  $11 \rightarrow 12 \rightarrow 31$ ,  $14 \rightarrow 13 \rightarrow 12$ , etc. are allowed, i.e.,  $[11, 12, 31] \in \mathcal{U}$ . In contrast,  $13 \rightarrow 12 \rightarrow 31$  and  $31 \rightarrow 12 \rightarrow 13$  are not. We further define  $\eta = [s, c_2, e]$ , where  $c_2$  analogously to Section II-B.

For our purpose we employ a modified LCA with *breadth-first* search, to which we refer as “LCAMod” in the following. Suppose we are currently analyzing transition  $i \rightarrow j$  for being element of the shortest path from  $s$  to  $e$ . Then, in the basic LCA implementation test (4) is conducted, and, in case of satisfaction, node  $j$  is added to the candidate list for further analysis of transitions starting from  $j$ . In contrast, in LCAMod, we add  $j$  to the candidate list only if (4) and  $[i_{k-1}, i, j] \in \mathcal{U}$  both hold, whereby  $i_{k-1}$  denotes the current parent-node (predecessor node) of  $i$ . We employ LCAMod as outlined in Algorithm 1, where “—” denotes one or multiple unused return values of a function call.

Feasibility or existence of a path from  $s$  to  $e$  can *always* be guaranteed. This is since there is always the solution of transitioning from  $s$  to the perimetric tractor-lane and consequently following it until the reaching of  $e$ . Noting further that  $T$  is created with nonnegative arc costs (here pathlengths), it is obvious that there exists a shortest path from  $s$  to  $e$ . Nevertheless, the single application of LCAMod is not guaranteed to find it. To see this, Figure 3 serves as illustration. The transition from node  $b$  to  $a$  directly will always be shorter (straight line) than via node  $i$ . Thus, the transition  $i \rightarrow a$  will be discarded when subsequently analyzed as a candidate. As a result, a *deadlock* is reached at node  $i$  and no path is determined with segment  $i \rightarrow a \rightarrow$  “to  $e$ ”, even though it obviously exists and may even be optimal. To resolve these (and similar) deadlocks, we propose a *restart* of LCAMod. Suppose the node removed

last from the candidate list is  $i$  with parent-node  $b$ . Then, instead of attempting to solve  $\eta = [s, c_2, e]$  by one LCAMod-call, we try solving two LCAMod-calls with  $\eta^{(1)} = [s, c_2, b]$  and  $\eta^{(2)} = [b, i, e]$ . A solution to  $\eta^{(1)}$  is guaranteed to be found by one LCAMod-call. In contrast,  $\eta^{(2)}$  may result in another second deadlock (which we resolve by another restart of LCAMod). Nevertheless, the primary deadlock (transition  $i \rightarrow a$ ) is guaranteed to be overcome. It remains to ensure that the node removed last from the candidate admits together with its parent-node a transition along the perimetric path. (For the illustrating example in Figure 3, node  $a$  may also be the node removed last). To ensure this particular transition, we therefore employ the heuristic, that, in case of a deadlock, the node removed last must together with its parent represent a transition along a perimetric lane-segment. Note that more than one restart may be required to find a path from  $s$  to  $e$  dependent on the complexity of the perimetric contour. Imagine, for example, the case of highly non-convex field shapes with multiple larger bays or indentations. Note that, however, when employing specific heuristic field coverage *patterns* repeatedly on regularly shaped fields, upper bounds on required restarts can be given. Some heuristic field coverage patterns are much more suitable than others, which is subject of ongoing work.

As outlined above, LCAMod may not immediately in one iteration find a solution to our shortest path finding problem. We therefore abbreviate the input-output relation of one LCAMod-function call as

$$(\text{UPPER}, \mathcal{C}, d_{\tilde{e}}, i, i_{k-1}) = f(s, c_2, \tilde{e}, T, \mathcal{U}),$$

where  $i$  and  $i_{k-1}$  refer to the node removed last from the candidate list and its parent when  $\text{UPPER} = \infty$ , respectively. The cost for the shortest path transition from  $s$  to  $\tilde{e}$  is denoted by  $d_{\tilde{e}}$ . Usually we have  $\tilde{e} = e$ . However, as described in Algorithm 1, we may also solve for subproblems with different temporary target nodes.  $\text{UPPER}$  always refers to the cost to the original target node  $e$ .

*Proposition 2:* The finding of a shortest path within an agricultural field under the repressed area minimization constraint can be solved optimally by applying Algorithm 1.

*Proof:* For our application the heading direction is important. We can distinguish between two cases. If a transition is from an interior lane-segment (*interior-to-perimetric*) then there always exists exactly one option to traverse to the perimetric lane. In contrast if a transition is from the perimetric lane (*perimetric-to-interior*), there are either one or two options dependent on the immediately ahead available lane-to-interior tractor trace transition: we can *always* follow the perimetric lane, and *potentially* may turn towards an interior lane-segment.

Any interior lane-segment connects two parts along the perimetric lane. Because of the *breadth-first* search for the removal of nodes from the LCAMod-candidate list for further analysis, both of these parts along the perimetric lane will be analyzed *alternately* after the first available interior-to-perimetric transition. Following above reasoning, this first interior-to-perimetric traversal initiates further a

unique transition direction, clockwise (CW) or counter-clockwise (CCW), at two points, referred to as  $\xi_1$  and  $\xi_2$ , along the perimetric lane. We introduce the binary variable  $\gamma(\xi) \in \{0, 1\}$  to indicate the transition direction as CW and CCW, respectively, initiated at  $\xi$ . Thus, because of the aforementioned *perimetric-to-interior* property, it is now possible to *always* follow the perimetric lane starting from both  $\xi_1$  and  $\xi_2$ . We can now further distinguish between two scenarios: it may occur  $\gamma(\xi_1) == \gamma(\xi_2)$  or  $\gamma(\xi_1) \neq \gamma(\xi_2)$ . In the former case, then it is easy to see that the target node is guaranteed to be reached without having to restart LCAMod. However, no remark about potential optimality can yet be made. In the latter case, a deadlock *may* result, but does not necessarily have to. In case of a deadlock, it can be resolved by *restarting* LCAMod as discussed above. By Bellman's principle of optimality the shortest path between two general locations  $A$  and  $C$  can *in general* not be described by the concatenation of, first, the shortest path from  $A$  to another location  $B$ , and, then, the shortest path from  $B$  to  $C$ . However, it is the case if a transition from  $A$  to  $C$  is *not* possible without visiting  $B$ . Thus, w.r.t. Algorithm 1, the transition via the deadlock node, achieved by restarting, must be the only possibility to find a path connecting  $s$  and  $e$  given  $T_c$ . This is the case since any other possible transition would have been detected applying the breadth-first search in combination with the perimetric-to-interior transitions.

It remains to be shown that only the iteration over various  $T_c$ , with different unique transitions to destination  $e$ , yields the shortest path from  $s$  to  $e$ . Suppose we use  $T$  directly.

---

#### Algorithm 1

---

```

1: Input: transition graph  $T$ , node numbers  $s$ ,  $c_2$  and  $e$ , and
   all lane-segment coordinates.
2: For every node  $n$  of  $N_n$  nodes adjacent to  $e$ :
3:   Create a copy  $T_c$  of  $T$ , cut all connections to  $e$  except
    $n \rightarrow e$  and conduct all following operations on  $T_c$ .
4:   Initialize UPPER =  $\infty$ ,  $d_e^{(n)} = 0$ ,  $C^{(n)} =$  empty list.
   Set  $\tilde{s} = s$  and  $\tilde{c}_2 = c_2$ .
5:   While UPPER ==  $\infty$ :
6:     (UPPER,  $C_{e(0)}^{(n)}$ ,  $d_{e(0)}^{(n)}$ ,  $i$ ,  $i_{k-1}$ ) =  $f(\tilde{s}, \tilde{c}_2, e, T_c, \mathcal{U})$ .
7:     If UPPER ==  $\infty$ :
8:       ( $-, C_{e(1)}$ ,  $d_{e(1)}$ ,  $-$ ,  $-$ ) =  $f(\tilde{s}, \tilde{c}_2, i_{k-1}, T_c, \mathcal{U})$ .
9:       (UPPER,  $C_{e(2)}$ ,  $d_{e(2)}$ ,  $-$ ) =  $f(i_{k-1}, i, e, T_c, \mathcal{U})$ .
10:      If UPPER ==  $\infty$ :
11:        Concatenate  $C_{e(1)}$  to  $C^{(n)}$ .
12:        Compute  $d_e^{(n)} = d_e^{(n)} + d_{e(1)}$ .
13:        Set  $\tilde{s} = i_{k-1}$  and  $\tilde{c}_2 = i$ .
14:      Else
15:        Concatenate  $C_{e(1)}$  and  $C_{e(2)}$  to  $C^{(n)}$ .
16:        Compute  $d_e^{(n)} = d_e^{(n)} + d_{e(1)} + d_{e(2)}$ .
17:      End If
18:      Else % shortest path found in one iteration.
19:        Set  $C^{(n)} = C_{e(0)}^{(n)}$  and  $d_e^{(n)} = d_{e(0)}^{(n)}$ .
20:      End If
21:    End While
22:  End For
23: Among the  $N_n$  solutions, select the one with shortest total
   pathlength  $d_e$  from  $s$  to  $e$ , and return the corresponding
   shortest path-coordinates suitable for navigation-aid or a
   fully autonomous tractor system application.

```

---

Then, a path from  $s$  to  $e$  may be found. In such a case, all nodes that resulted in a deadlock will have been discarded from the candidate list, eventhough these nodes may have lead to an overall shorter path. This concludes the proof. ■

## IV. NUMERICAL EXPERIMENTS

### A. Orchard-like Areas

We consider field data of a real-world orchard in Northern Germany with an inter-row space of 3.5m (operating width). The resulting shortest path trajectories are shown in Figure 4. For the case of multiple target nodes being available, we simply solve the shortest path problem for all of them, before determining the best solution. Algorithmic runtimes are given in Table I, whereby the LCA-based methods use breadth-first search. We compared breadth-first, depth-first and Dijkstra's graph search method and found breadth-first to be fastest. Eventhough the LCA-heuristic reduces the number of iterations for the given example, it was slightly slower because of the overhead in additional condition checking. All simulations are conducted on a laptop running Ubuntu 14.04 equipped with an Intel Core i7 CPU @2.80GHz×8, 15.6GB of memory, and using MATLAB 8.6 (R2015b).

### B. Agricultural Fields

The results of simulation experiments for a real-world agricultural field in Northern Germany with an operating width (inter-row space) of 24m are displayed in Figure 5. Numerical results are given in Table II. Interestingly, because of the stringent repressed area minimization constraint check, very few nodes were ultimately added to the candidate list resulting in very fast computation times. For every immediate neighbor of  $e$ , a solution is analyzed as outlined in Algorithm 1. The target node for the third experiment in Figure 5(c) has three immediate neighbors. In contrast, for the examples in Figure 5(a) and Figure 5(b), there are only two nodes immediately neighboring the exit node. This is the reason for the larger number of 215 required iterations in the third experiment. As indicated in Figure 5(c), despite a local proximity of start and target node, the shortest path between them under the repressed area minimization constraint may be considerably longer and less intuitive.

## V. CONCLUSION

We presented methods for shortest path finding for ground vehicle operation in agriculturally used areas. It was distinguished between interior lanes and perimetric paths. Constraints such as admittance of an instantaneous turn of the vehicle and, the usage of only already existing tractor traces for repressed area minimization were discussed. Corresponding solution approaches based on dynamic programming and customized label correcting algorithms were given.

Future work will discuss the influence and suitability of specific field coverage patterns for shortest path finding. In addition, tree islands prohibited from trespassing in the interior of an agricultural working area have to be accounted for. Another topic may be the comparison of repeated but

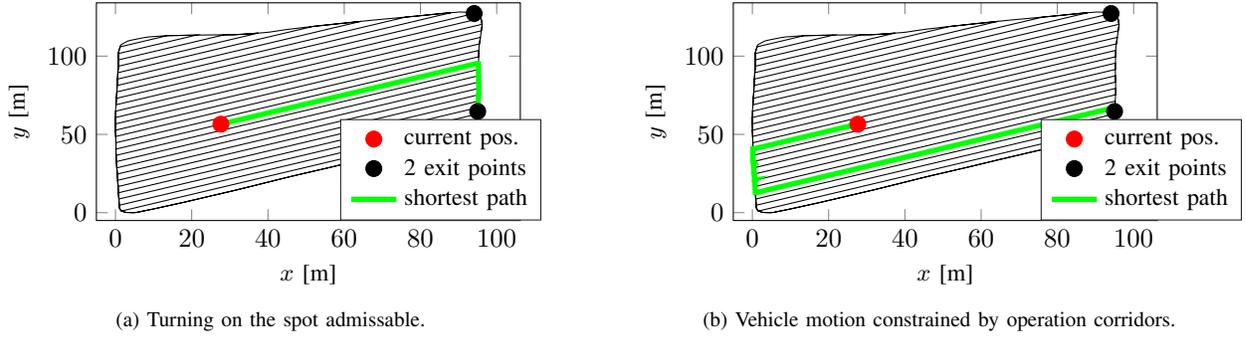


Fig. 4. Resulting shortest paths for the two experiments with an orchard-like area, see Section IV-A. (a) The ground vehicle is initially heading along the current interior lane towards negative  $x$ - and  $y$ -direction. There are two admissible exit points. In both cases (a) and (b), the exit node with smaller  $y$ -coordinate resulted in a slightly shorter traveling distance starting from  $s$ . The pathlengths for the displayed shortest paths are (a) 108m and (b) 171m.

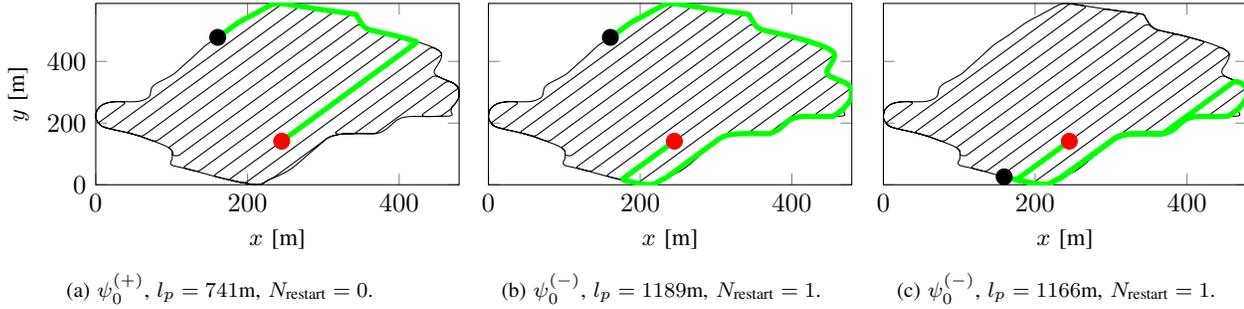


Fig. 5. Resulting shortest paths for experiments within an agricultural field under the repressed area minimization constraint, see Section IV-B. The initial heading of the ground vehicle along the current interior lane towards positive and negative  $x$ - and  $y$ -direction is indicated by  $\psi_0^{(+)}$  and  $\psi_0^{(-)}$ , respectively. The total pathlength and number of restarts required are  $l_p$  and  $N_{restart}$ , respectively. Three different scenarios are displayed.

TABLE I. Comparison of algorithms for the experiment with an orchard-like area, see Section IV-A and Figure 4. The number of required iterations and total computation time [ms] is denoted by  $N_{iter}$  and  $\bar{\tau}_{total}$ , respectively. Results for both Figures 4(a) and 4(b) are given.

		DP	LCA-baseline	LCA-heuristic
Fig. 4(a)	$N_{iter}/\bar{\tau}_{total}$	10/70.4	44/4.4	20/6.1
Fig. 4(b)	$N_{iter}/\bar{\tau}_{total}$	11/183.7	54/4.3	20/6.5

TABLE II. Comparison of the LCMod-results for the three experiments displayed in Figure 5. The total number of required LCMod-iterations and corresponding CPU-time [ms] is denoted by  $N_{iter}$  and  $\bar{\tau}_{total}$ , respectively.

	Fig. 5(a)	Fig. 5(b)	Fig. 5(c)
$N_{iter}/\bar{\tau}_{total}$	54/11.5	114/3.3	215/5.2

only partial field coverage with frequent returning to a (mobile) depot for refilling using shortest path planning and smaller as well as lighter towed implements, versus traditional one-run field coverage with larger and heavier application machinery, e.g., for storage of larger amounts of pesticides and fertilizers.

## REFERENCES

- [1] J. Backman, T. Oksanen, and A. Visala, "Navigation system for agricultural machines: nonlinear model predictive path tracking," *Computers and Electronics in Agriculture*, vol. 82, pp. 32–43, 2012.
- [2] R. Lenain, B. Thuilot, C. Cariou, and P. Martinet, "Adaptive and predictive path tracking control for off-road mobile robots," *European Journal of Control*, vol. 13, no. 4, pp. 419–439, 2007.
- [3] C. Sørensen, S. Fountas, E. Nash, L. Pesonen, D. Bochtis, S. M. Pedersen, B. Basso, and S. Blackmore, "Conceptual model of a future farm management information system," *Computers and Electronics in Agriculture*, vol. 72, no. 1, pp. 37–47, 2010.
- [4] M. A. F. Jensen, D. Bochtis, C. G. Sørensen, M. R. Blas, and K. L. Lykkegaard, "In-field and inter-field path planning for agricultural transport units," *Computers & Industrial Engineering*, vol. 63, no. 4, pp. 1054–1061, 2012.
- [5] D. Bochtis and C. Sørensen, "The vehicle routing problem in field logistics part i," *Biosystems Engineering*, vol. 104, no. 4, pp. 447–457, 2009.
- [6] D. Bochtis, C. Sørensen, and S. Vougioukas, "Path planning for in-field navigation-aiding of service units," *Computers and Electronics in Agriculture*, vol. 74, no. 1, pp. 80–90, 2010.
- [7] D. P. Bertsekas, *Dynamic programming and optimal control*, vol. 1. Athena Scientific Belmont, MA, 1995.
- [8] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*, vol. 6. MIT press Cambridge, 2001.
- [9] D. Bochtis, H. Griepentrog, S. Vougioukas, P. Busato, R. Berruto, and K. Zhou, "Route planning for orchard operations," *Computers and Electronics in Agriculture*, vol. 113, pp. 51–60, 2015.
- [10] S. J. Moorehead, C. K. Wellington, B. J. Gilmore, and C. Vallespi, "Automating orchards: A system of autonomous tractors for orchard maintenance," in *IEEE/RSJ International Conference on Intelligent Robots and Systems Workshop on Agricultural Robotics, Vilamoura, Portugal*, 2012.
- [11] O. C. Barawid, A. Mizushima, K. Ishii, and N. Noguchi, "Development of an autonomous navigation system using a two-dimensional laser scanner in an orchard application," *Biosystems Engineering*, vol. 96, no. 2, pp. 139–149, 2007.
- [12] M. Graf Plessen and A. Bemporad, "Reference trajectory planning under constraints and path tracking using linear time-varying model predictive control for agricultural machines," *Biosystems Engineering*, 2016. (to appear).
- [13] D. Bochtis and S. Vougioukas, "Minimising the non-working distance travelled by machines operating in a headland field pattern," *Biosystems Engineering*, vol. 101, no. 1, pp. 1–12, 2008.