Ant Systems in Navigation and Logistics

Prof. Dr. Sebastian Iwanowski

FH Wedel, University of Applied Systems, Germany
Guest Lecturer at Haaga-Helia Ammattikorkeakoulu
Class Innovation Topics, Friday
18. September 2015

Ant Systems in Navigation and Logistics

➡ 1. Traditional approach revised: Dijkstra and A*

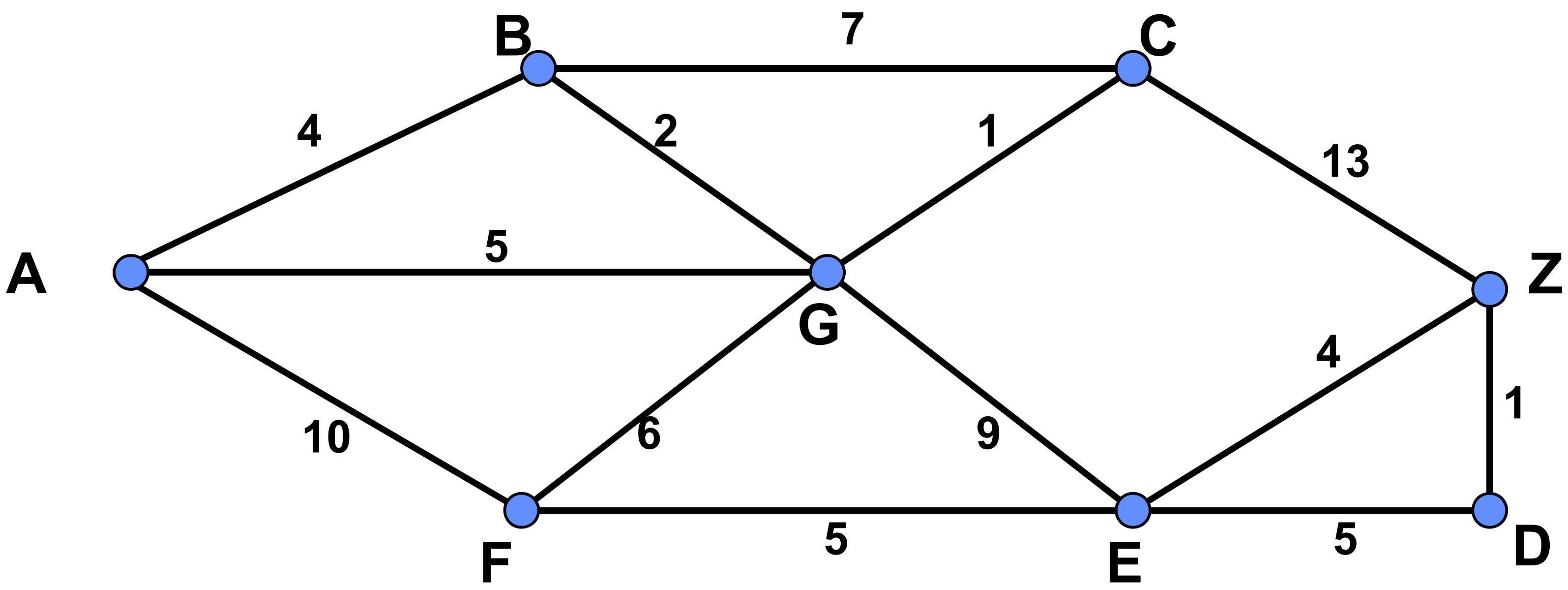
2. Discussion and extension of ant homework assignment

3. How to bring ant systems into practice

4. Logistics problems and the use of ant systems

5. Conclusion: Lessons learnt

Solution to Homework assignment 1: Dijkstra



Shortest path from G to Z: G → E → Z (13 units)

Node (distance from G, direct predecessor):

A(5,G)		A(5,G)		**[A(5,G)]**						
B(2,G)		**[B(2,G)]**								
[C(1,G)]										
D(∞)	→	D(∞)	→	D(∞)	→	D(∞)	→	D(∞)	→	D(14,E)
E(9,G)		E(9,G)		E(9,G)		E(9,G)		**[E(9,G)]**		
F(6,G)		F(6,G)		F(6,G)		**[F(6,G)]**				
Z(∞)		Z(14,C)		Z(14,C)		Z(14,C)		Z(14,C)		**[Z(13,E)]**

Acceleration of Dijkstra: A*

A* algorithm for weighted graphs

(Generalisation of Dijkstra's algorithm) ***(State evaluation = Node evaluation)***

Requirement for edge weights: All edge lengths must be nonnegative.

Requirement for heuristic function $h_B(u)$ for estimating the real distance $d_B(u)$ to target node B:

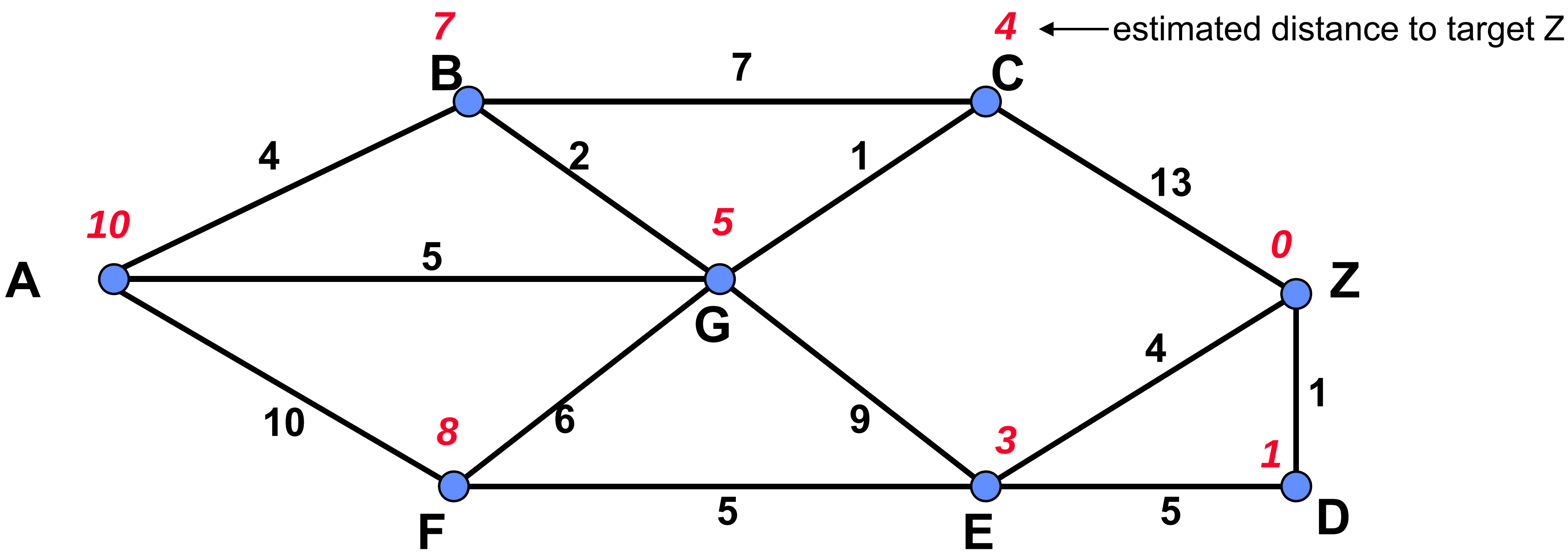
Admissibility: $h_B(u) \leq d_B(u)$

Monotonicity: $h_B(u) \leq h_B(v) + \mathit{length}(u,v)$

Algorithm for the search of a path from A to B having minimal total edge length:

- Put A into the set **`Done`**. Label A by *distance*(A) := 0.
  Put all other nodes into the set **`YetToCompute`**.
  Label all neighbors N of A by *distance* (N) := *length* (A,N) and
  heuristic (N) := *distance* (N) + h_B(N)
  and all other nodes by *distance* (V) := ∞ and *heuristic* (V) := ∞.
- Repeat:
  Choose node V from **`YetToCompute`** with minimum *heuristic* (V)
  and shift V to the set **`Done`**.
  Update all neighbors N of V that are still in **`YetToCompute`**:
  distance (N) := min {*distance* (N), *distance* (V) + *length* (V,N)}.
  heuristic (N) := *distance* (N) + h_B(N) (if update is necessary).
  until V = B

Example for A* algorithm



Shortest path from G to Z: G → E → Z (13 units)

Node (real distance from G, direct predecessor, estimated distance to target):

A(5,G,15)		A(5,G,15)		A(5,G,15)		A(5,G,15)
B(2,G,9)		B(2,G,9)				
C(1,G,5)						
D(∞)	→	D(∞)	→	D(∞)	→	D(14,E,15)
E(9,G,12)		E(9,G,12)		E(9,G,12)		
F(6,G,13)		F(6,G,14)		F(6,G,14)		F(6,G,14)
Z(∞)		Z(14,C,14)		Z(14,C,14)		Z(13,E,13)

Ant Systems in Navigation and Logistics

1. Traditional approach revised: Dijkstra and A*
2. Discussion and extension of ant homework assignment
3. How to bring ant systems into practice
4. Logistics problems and the use of ant systems
5. Conclusion: Lessons learnt

Homework assignment 2: Ant systems

Consider the following network and the corresponding pheromone tables:

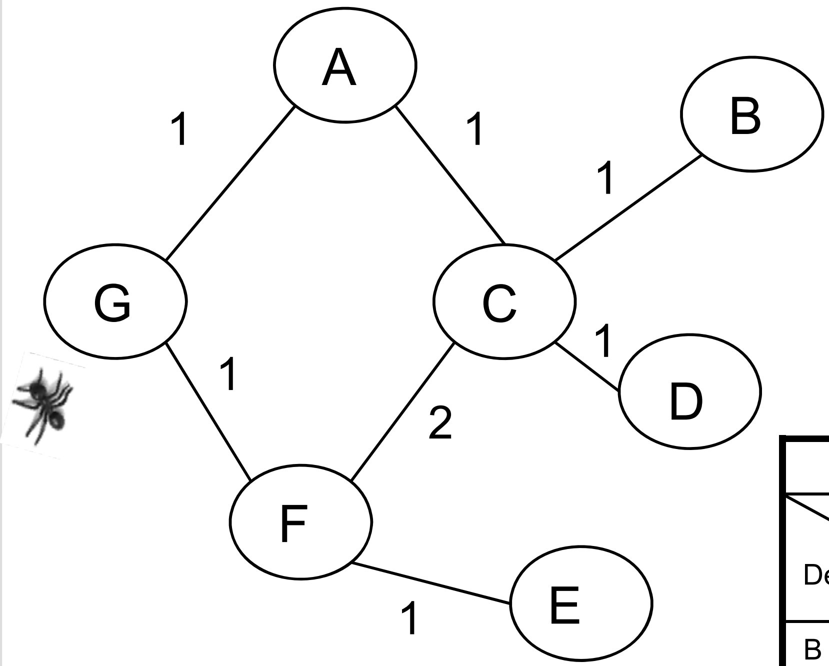


Table for F (extract)

Dest \ Next	C	G	E
B	0.5	0.35	0.15

Table for A (extract)

Dest \ Next	C	G
B	0.9	0.1

Table for C (extract)

Dest \ Next	A	B	D	F
B	0.05	0.85	0.05	0.05

Table for G (extract)

Dest \ Next	A	F
B	0.5	0.5

i) Generate an ant in G with destination B and let it run via A and C to B. Update the pheromone tables according to the presented method with the constants c1=c2=1, but without the restriction that at least 0,05 must remain as remainder probability.

ii) Generate now a second ant in G with destination B and let it run via F and C. Update the pheromone tables according to the presented method as in i).

iii) Exchange the order of i) and ii): What do you observe?

i) +ii) First via A, then F

	A	F
G->B	0,5	0,5

via A:

ΔP = 1,33

	A	F
G->B	0,79	0,21

via F:

ΔP = 1,25

	A	F
G->B	0,35	0,65

iii) First via F, then via A

	A	F
G->B	0,5	0,5

via F:

ΔP = 1,25

	A	F
G->B	0,22	0,88

via A:

ΔP = 1,33

	A	F
G->B	0,67	0,33

Appearently it matters which ant moves first!

$$\Delta P_{d,B} = \frac{1}{t_{d,B}} + 1 \qquad P_{B,l} = \frac{P_{B,l}}{1 + \Delta P_{d,B}} \qquad P_{B,d} = \frac{P_{B,d} + \Delta P_{d,B}}{1 + \Delta P_{d,B}}$$

Multiple execution:

i) First via A, then via F

	A	F
G->B	0,5	0,5

via A:

$\Delta P = 1{,}33$

	A	F
G->B	0,79	0,21

via F:

$\Delta P = 1{,}25$

	A	F
G->B	0,35	0,65

via A:

$\Delta P = 1{,}33$

	A	F
G->B	0,72	0,28

via F:

$\Delta P = 1{,}25$

	A	F
G->B	0,32	0,68

via A:

$\Delta P = 1{,}33$

	A	F
G->B	0,71	0,29

via F:

$\Delta P = 1{,}25$

	A	F
G->B	0,32	0,68

This remains invariant for future runs.

Average values:

	A	F
G->B	0,515	0,485

$$\Delta P_{d,B} = \frac{1}{t_{d,B}} + 1 \qquad P_{B,l} = \frac{P_{B,l}}{1 + \Delta P_{d,B}} \qquad P_{B,d} = \frac{P_{B,d} + \Delta P_{d,B}}{1 + \Delta P_{d,B}}$$

Multiple execution:

ii) First via F, then via A

	A	F
G->B	0,5	0,5

via F:

ΔP = 1,25

	A	F
G->B	0,22	0,88

via A:

ΔP = 1,33

	A	F
G->B	0,67	0,33

via F:

ΔP = 1,25

	A	F
G->B	0,30	0,70

via A:

ΔP = 1,33

	A	F
G->B	0,70	0,30

via F:

ΔP = 1,25

	A	F
G->B	0,31	0,69

via A:

ΔP = 1,33

	A	F
G->B	0,70	0,30

This remains invariant for future runs.

Average values:

	A	F
G->B	0,505	0,495

$$\Delta P_{d,B} = \frac{1}{t_{d,B}} + 1 \qquad P_{B,l} = \frac{P_{B,l}}{1 + \Delta P_{d,B}} \qquad P_{B,d} = \frac{P_{B,d} + \Delta P_{d,B}}{1 + \Delta P_{d,B}}$$

Interpretation: Among 100 ants 2 more ants run via A than via F

Development of average value under consideration that 2 more ants run via A:

	A	F
G->B	0,51	0,49

via A:

ΔP = 1,33

	A	F
G->B	0,79	0,21

via A:

ΔP = 1,33

	A	F
G->B	0,91	0,09

This is only the best case! **For the average case assume only:**

	A	F
G->B	0,80	0,20

For future runs: Among 5 ants 4 run via A, 1 via F

$$\Delta P_{d,B} = \frac{1}{t_{d,B}} + 1 \qquad P_{B,l} = \frac{P_{B,l}}{1 + \Delta P_{d,B}} \qquad P_{B,d} = \frac{P_{B,d} + \Delta P_{d,B}}{1 + \Delta P_{d,B}}$$

For the average case assume only:

	A	F
G->B	0,80	0,20

For future runs: Among 5 ants 4 run via A, 1 via F

Exact computation of average value:

1) Best case

via F:

ΔP = 1,25

	A	F
G->B	0,35	0,65

via A:

ΔP = 1,33

	A	F
G->B	0,72	0,28

via A:

ΔP = 1,33

	A	F
G->B	0,88	0,12

via A:

ΔP = 1,33

	A	F
G->B	0,95	0,05

via A:

ΔP = 1,33

	A	F
G->B	0,98	0,02

$$\Delta P_{d,B} = \frac{1}{t_{d,B}} + 1 \qquad P_{B,l} = \frac{P_{B,l}}{1 + \Delta P_{d,B}} \qquad P_{B,d} = \frac{P_{B,d} + \Delta P_{d,B}}{1 + \Delta P_{d,B}}$$

For the average case assume only:

	A	F
G->B	0,80	0,20

For future runs: Among 5 ants 4 run via A, 1 via F

Exact computation of average value:

2) Worst case

via A: **$\Delta P = 1,33$**

	A	F
G->B	0,91	0,09

via A: **$\Delta P = 1,33$**

	A	F
G->B	0,96	0,04

via A: **$\Delta P = 1,33$**

	A	F
G->B	0,98	0,02

via A: **$\Delta P = 1,33$**

	A	F
G->B	0,99	0,01

via F: **$\Delta P = 1,25$**

	A	F
G->B	0,44	0,56

$$\Delta P_{d,B} = \frac{1}{t_{d,B}} + 1 \qquad P_{B,l} = \frac{P_{B,l}}{1 + \Delta P_{d,B}} \qquad P_{B,d} = \frac{P_{B,d} + \Delta P_{d,B}}{1 + \Delta P_{d,B}}$$

For the average case assume only:

	A	F
G->B	0,80	0,20

For future runs: Among 5 ants 4 run via A, 1 via F

Exact computation of average value:

3) Further case

via A: **ΔP = 1,33**

	A	F
G->B	0,91	0,09

via F: **ΔP = 1,25**

	A	F
G->B	0,40	0,60

via A: **ΔP = 1,33**

	A	F
G->B	0,74	0,26

via A: **ΔP = 1,33**

	A	F
G->B	0,88	0,12

via A: **ΔP = 1,33**

	A	F
G->B	0,95	0,05

$$\Delta P_{d,B} = \frac{1}{t_{d,B}} + 1 \qquad P_{B,l} = \frac{P_{B,l}}{1 + \Delta P_{d,B}} \qquad P_{B,d} = \frac{P_{B,d} + \Delta P_{d,B}}{1 + \Delta P_{d,B}}$$

For the average case assume only:

	A	F
G->B	0,80	0,20

For future runs: Among 5 ants 4 run via A, 1 via F

Exact computation of average value:

4) Further case

via A:

$\Delta P = 1,33$

	A	F
G->B	0,91	0,09

via A:

$\Delta P = 1,33$

	A	F
G->B	0,96	0,04

via F:

$\Delta P = 1,25$

	A	F
G->B	0,43	0,57

via A:

$\Delta P = 1,33$

	A	F
G->B	0,76	0,24

via A:

$\Delta P = 1,33$

	A	F
G->B	0,90	0,10

$$\Delta P_{d,B} = \frac{1}{t_{d,B}} + 1 \qquad P_{B,l} = \frac{P_{B,l}}{1 + \Delta P_{d,B}} \qquad P_{B,d} = \frac{P_{B,d} + \Delta P_{d,B}}{1 + \Delta P_{d,B}}$$

For the average case assume only:

	A	F
G->B	0,80	0,20

For future runs: Among 5 ants 4 run via A, 1 via F

Exact computation of average value:

5) Further case

via A:

ΔP = 1,33

	A	F
G->B	0,91	0,09

via A:

ΔP = 1,33

	A	F
G->B	0,96	0,04

via A:

ΔP = 1,33

	A	F
G->B	0,98	0,02

via F:

ΔP = 1,25

	A	F
G->B	0,44	0,56

via A:

ΔP = 1,33

	A	F
G->B	0,76	0,24

$$\Delta P_{d,B} = \frac{1}{t_{d,B}} + 1 \qquad P_{B,l} = \frac{P_{B,l}}{1 + \Delta P_{d,B}} \qquad P_{B,d} = \frac{P_{B,d} + \Delta P_{d,B}}{1 + \Delta P_{d,B}}$$

For the average case assume only:

	A	F
G->B	0,80	0,20

For future runs: Among 5 ants 4 run via A, 1 via F

Exact computation of average value:

Summary of the 5 cases:

	A	F
G->B	0,98	0,02

	A	F
G->B	0,95	0,05

	A	F
G->B	0,90	0,01

	A	F
G->B	0,76	0,24

	A	F
G->B	0,44	0,56

Exact average:

	A	F
G->B	0,81	0,19

$$\Delta P_{d,B} = \frac{1}{t_{d,B}} + 1 \qquad P_{B,l} = \frac{P_{B,l}}{1 + \Delta P_{d,B}} \qquad P_{B,d} = \frac{P_{B,d} + \Delta P_{d,B}}{1 + \Delta P_{d,B}}$$

Ant Systems in Navigation and Logistics

1. Traditional approach revised: Dijkstra and A*
2. Discussion and extension of ant homework assignment
3. How to bring ant systems into practice
4. Logistics problems and the use of ant systems
5. Conclusion: Lessons learnt

Artificial Ant Systems

How do we simulate ant behaviour for the routing problem?

Different pheromones for different destinations

- Each node has got a routing table
- This looks exactly like routing tables in a computer network

Continuous simulation from each source to each destination

- For n nodes serving as potential source and destination, there are n^2 different routing problems
- For each routing problem, there must be a continuous launch of ants with subsequent update of the corresponding pheromones.

This requires huge computational capacities

But there is hope for the future:

- The system should run on a central server, not at the clients.
- Cloud computing provides huge capacities.

Artificial Ant Systems

Which principles are worth to be integrated in any future navigation system?

Eager computing:

- Answer is computed before query is asked.
- This enables the integration of very recent information in actual decisions.

Off-board middleware:

- Dynamic data is collected from all of the country and distibuted to each user.
- Overall communication from m providers to n users is reduced from $m \cdot n$ to $m+n$.

Statistical compression of individual information:

- Pheromones store compressed information collected from many segments.
- The pheromones do not only consider the next segment ahead but the entire remaining trip.

Local information first:

- Changes close to the current position arrive earlier than changes far away.
- This provides the user with more recent information for the relevant part.

Open Street Map

The wiki approach to a map

Everybody is able to contribute to the map

- This makes the map the most accurate one in regions with a lot of IT prone people.

The sources for representation and interfaces are open

- This enables the integration of new services.
- In openstreetmap.de, routing functionality is already provided.

Business advantages

- Any user may use any service of open street map without restrictions.
- Even an integration in a commercial software is allowed.
- Public authorities support OpenStreetMap.

Open Street Map

Preprocessing OpenStreetMap data for ant simulation

OSM data is represented on a very low level

- Road segments do not go from corner to corner, but from geopoint to geopoint using a straight line in between. A way is a sequence of such road segments.
- Ways are classified in different categories of roads.
- Road corners consist of several nodes representing the different turning opportunities.

This requires a data abstraction

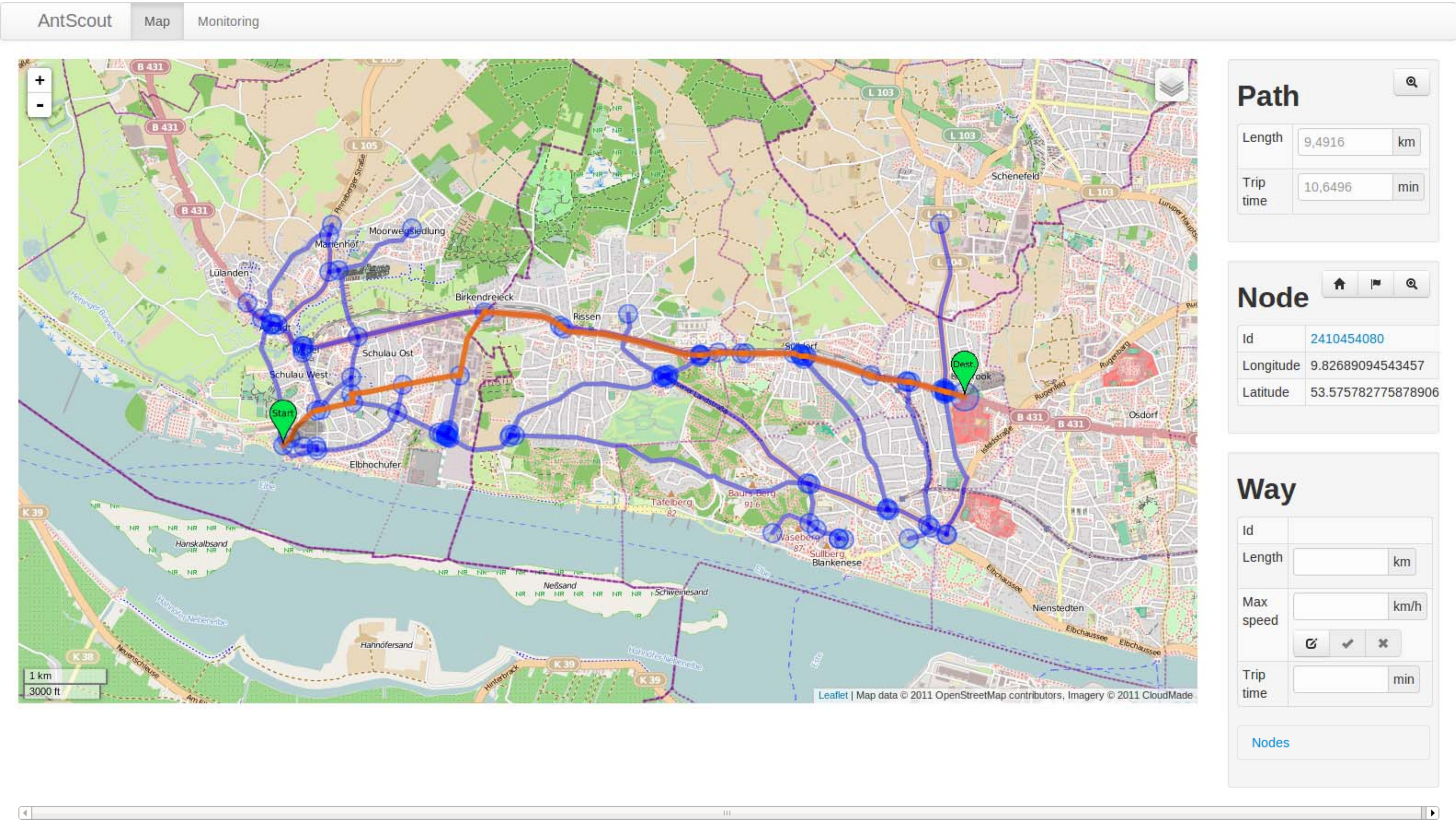
- In order to get a feasible number of nodes, only data within a predefined geographic rectangle is taken.
- Only road segments of a certain category are considered.
- Only nodes adjacent to at least 3 considered road segements are considered.

On current notebooks, our implementation works for maps with at most 150 nodes.

AntScout: An ant system on OpenStreetMap

User functionality

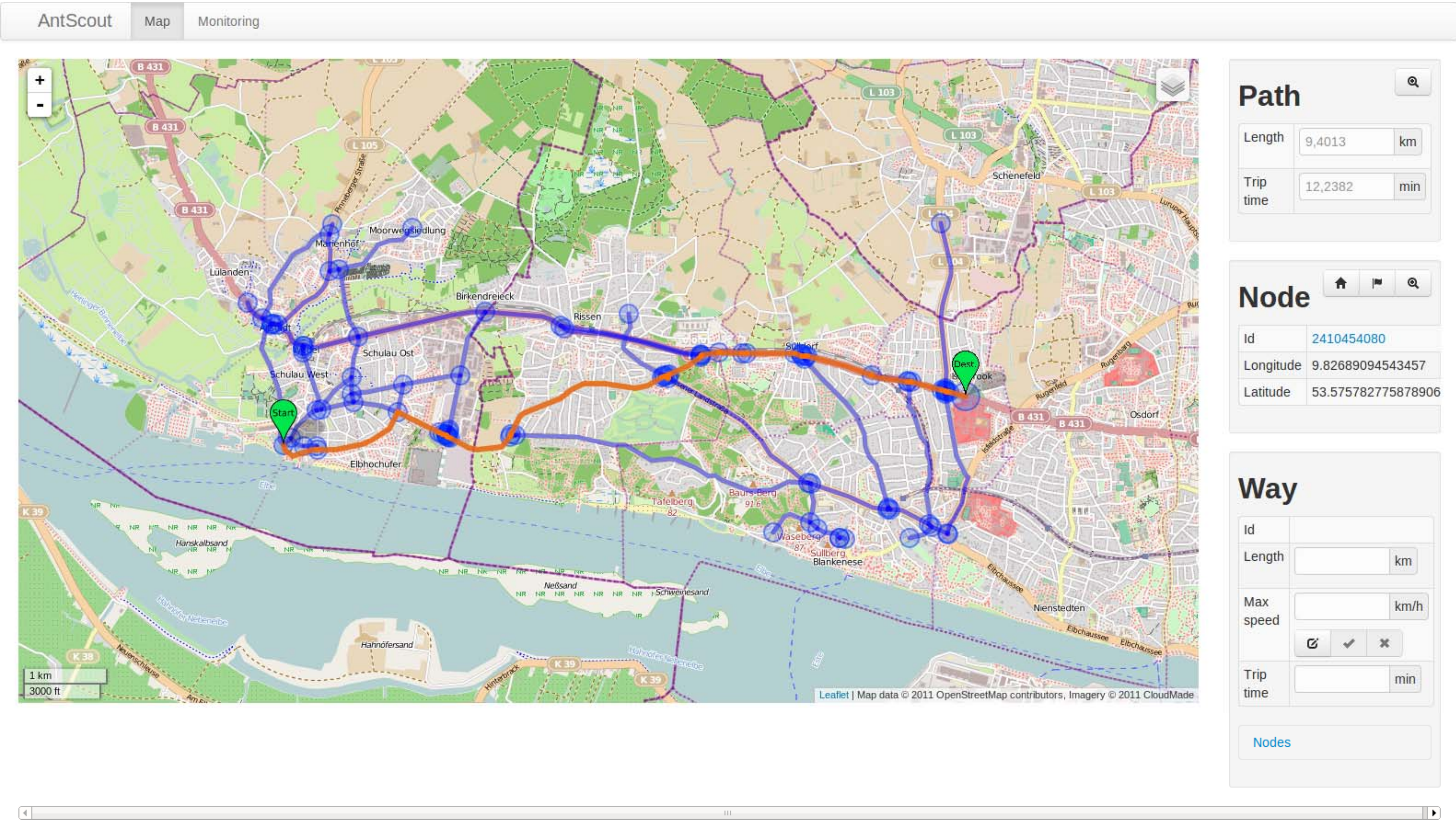
- Perform navigation queries on the preprocessed map



AntScout: An ant system on OpenStreetMap

User functionality

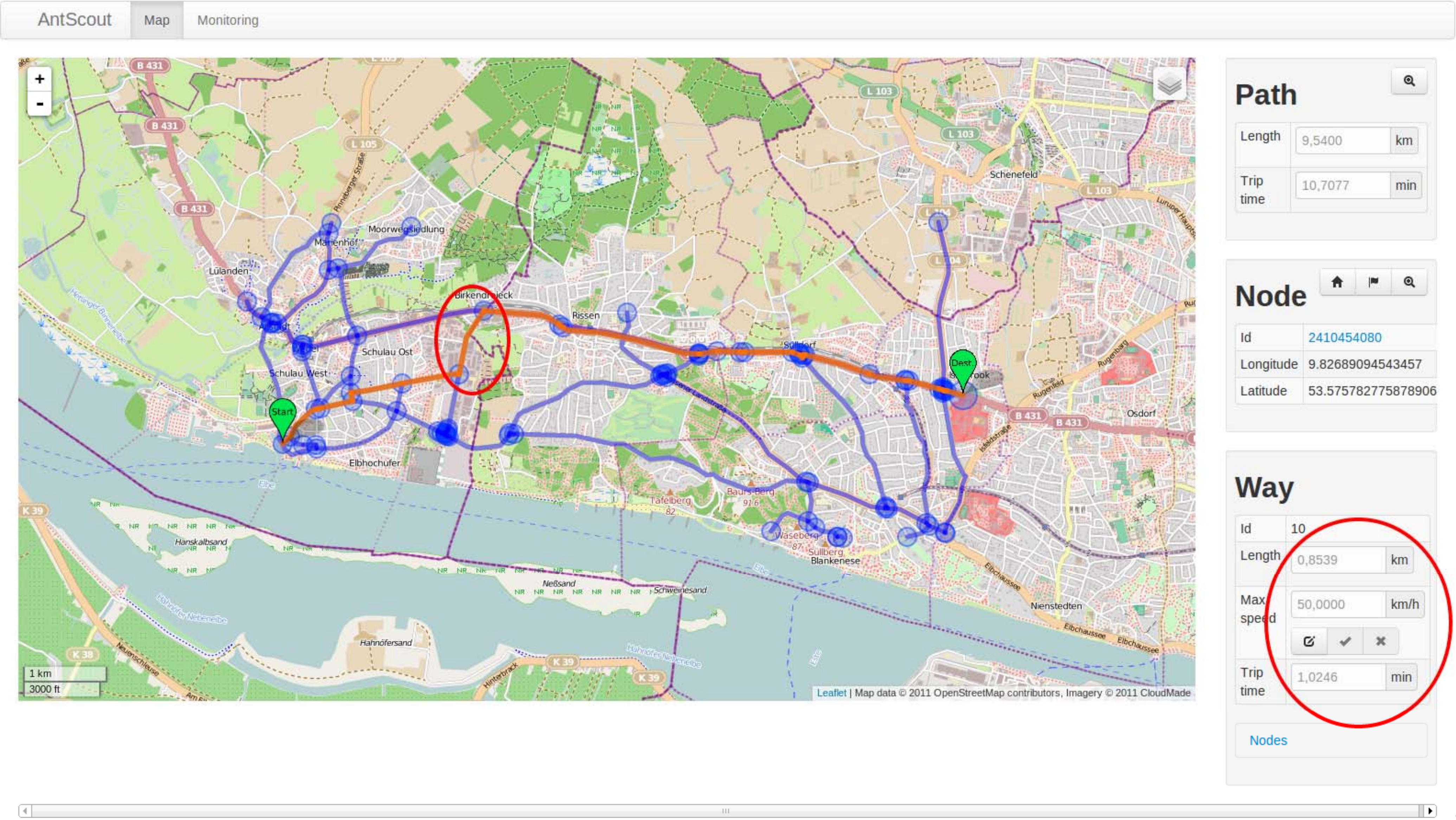
- Perform navigation queries on the preprocessed map



AntScout: An ant system on OpenStreetMap

Operator functionality

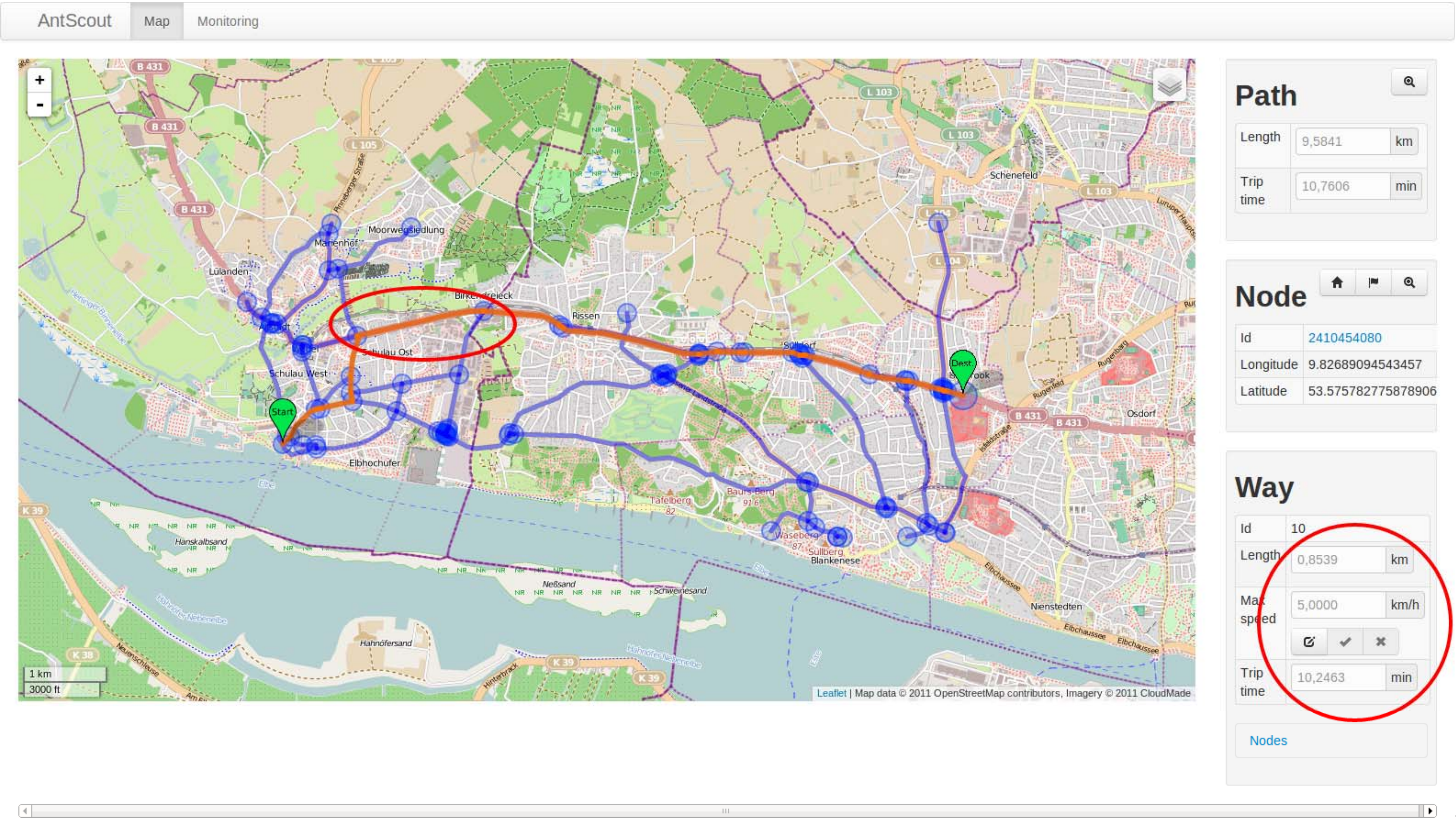
- Change the quality of any selected road segment



AntScout: An ant system on OpenStreetMap

Operator functionality

- Change the quality of any selected road segment



AntScout: An ant system on OpenStreetMap

Contributions to the state-of-the-art of ant systems

- **First ant system running on a real map**
- **Ants are modelled as node states instead of independent units**
- **Ant launch frequency is dependent from the distance between source and destination**

Interesting observations testing the map

- **The suggested route is flickering nondeterministically between routes of similar quality.**
- **Ant systems tend to prefer routes with fewer decision alternatives.**

AntScout: An ant system on OpenStreetMap

Tasks for future development

Include hierarchies for better feasibility

- Summarize nodes belonging to the same corner
- Introduce supernodes for long-distance travelling

Improve algorithm for better convergence

- Give a higher priority to ants using nodes with more decision alternatives.

Establish traffic control instead of traffic information

- Exploit the fact that ant systems tend to oscillate between routes of almost the same quality.

2015 (still ongoing): Master thesis project on AntScout +

Swarm intelligence vs. traditional algorithmics

What do ant systems solve ?

Given the location of some food supplies and the nest:
Find the shortest path between the nest and each food supply!

- Ants do not always find the optimal solution.
- Ants work in a non-deterministic way.
- Simulation of ant systems needs a lot of storage and does not guarantee any time limit.

How does traditional algorithmics solve this problem?

- with deterministic algorithms providing an optimal solution in time quadratic in the number of network nodes, in practice even better (Dijkstra, A*)

Why should we use ant systems ?

Swarm intelligence vs. traditional algorithmics

What do ants better than traditional algorithmics ?

Ants solve the following variant of the path finding problem:

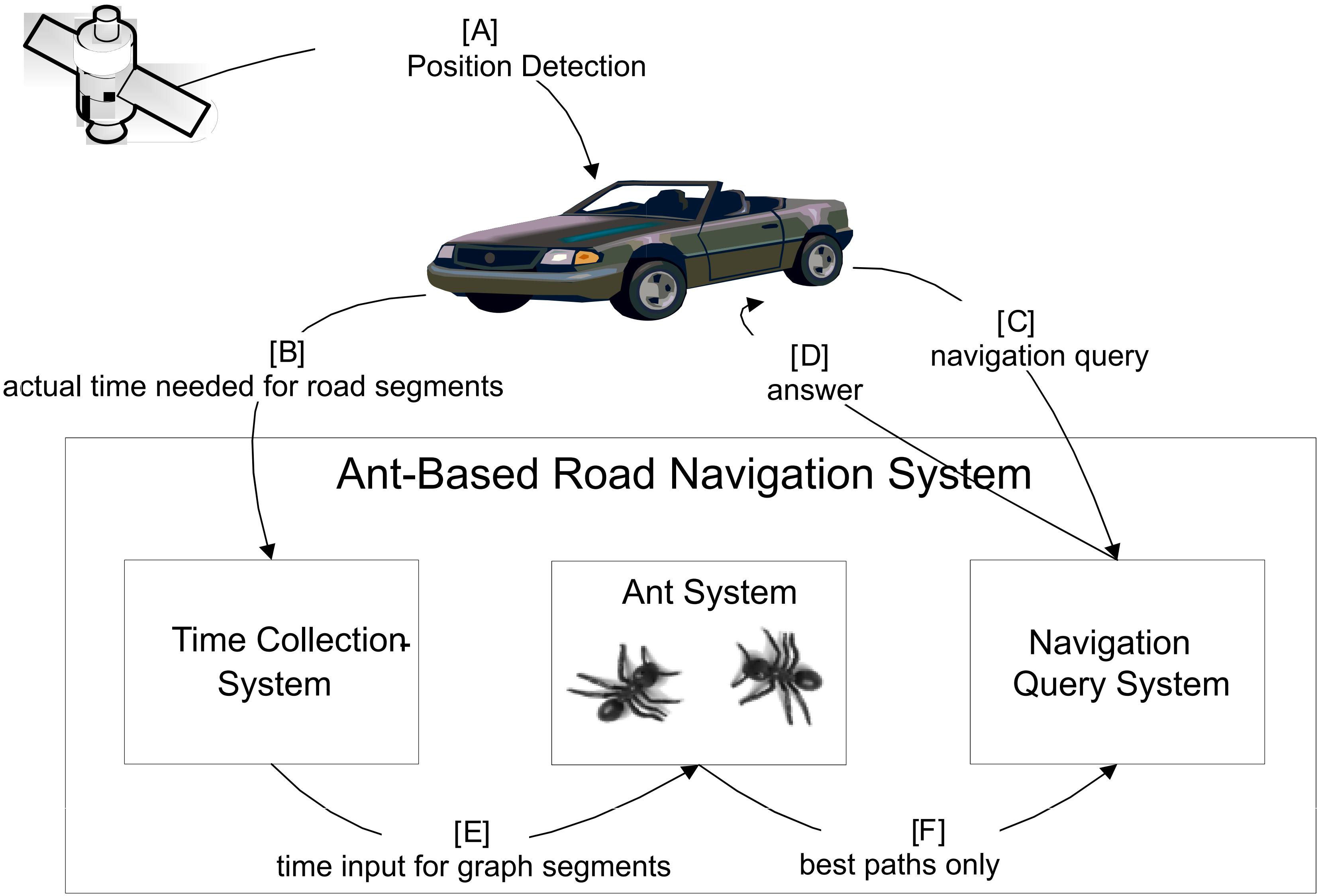
- The parameters of the underlying graph are only known in a local scope.
- Edge costs may change any time (without the possibility of prediction).

This problem is the typical problem of road navigation.

Traditional algorithmics does not deal with such unspecified problems!

This is the chance for the applications of ant systems !

Street navigation: General Software Architecture



proposed in master thesis of Thomas Walther, 2006 (in German)

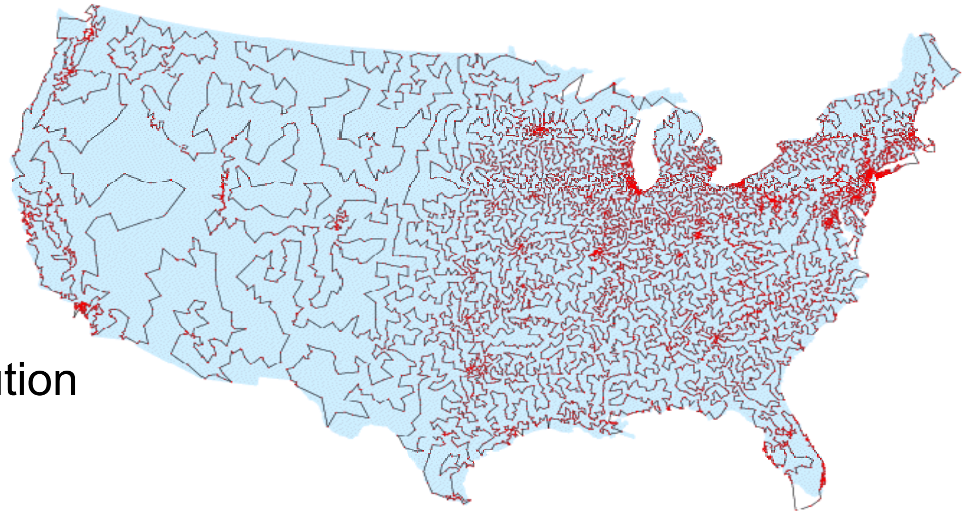
Ant Systems in Navigation and Logistics

1. Traditional approach revised: Dijkstra and A*
2. Discussion and extension of ant homework assignment
3. How to bring ant systems into practice
4. Logistics problems and the use of ant systems
5. Conclusion: Lessons learnt

The Traveling Salesman Problem (TSP)

Traveling Salesman Problem (TSP):

Given a set of cities with mutual distances:
Find the shortest round trip passing each city at least once!



This is a very prominent problem in algorithmics:

Problem is NP-complete!

- Almost no hope for an efficient optimal solution

This holds even for the static case!

Solution concepts:

- Probabilistic
- Special heuristics with no sound mathematical proof

This is what ant systems also do !

Applications of Traveling Salesman Problem

More general case in logistics:

- Observe time windows
- Observe load capacities
- … (further application specific restrictions)

FH Wedel Internal Project: Hamburg Tourist Information

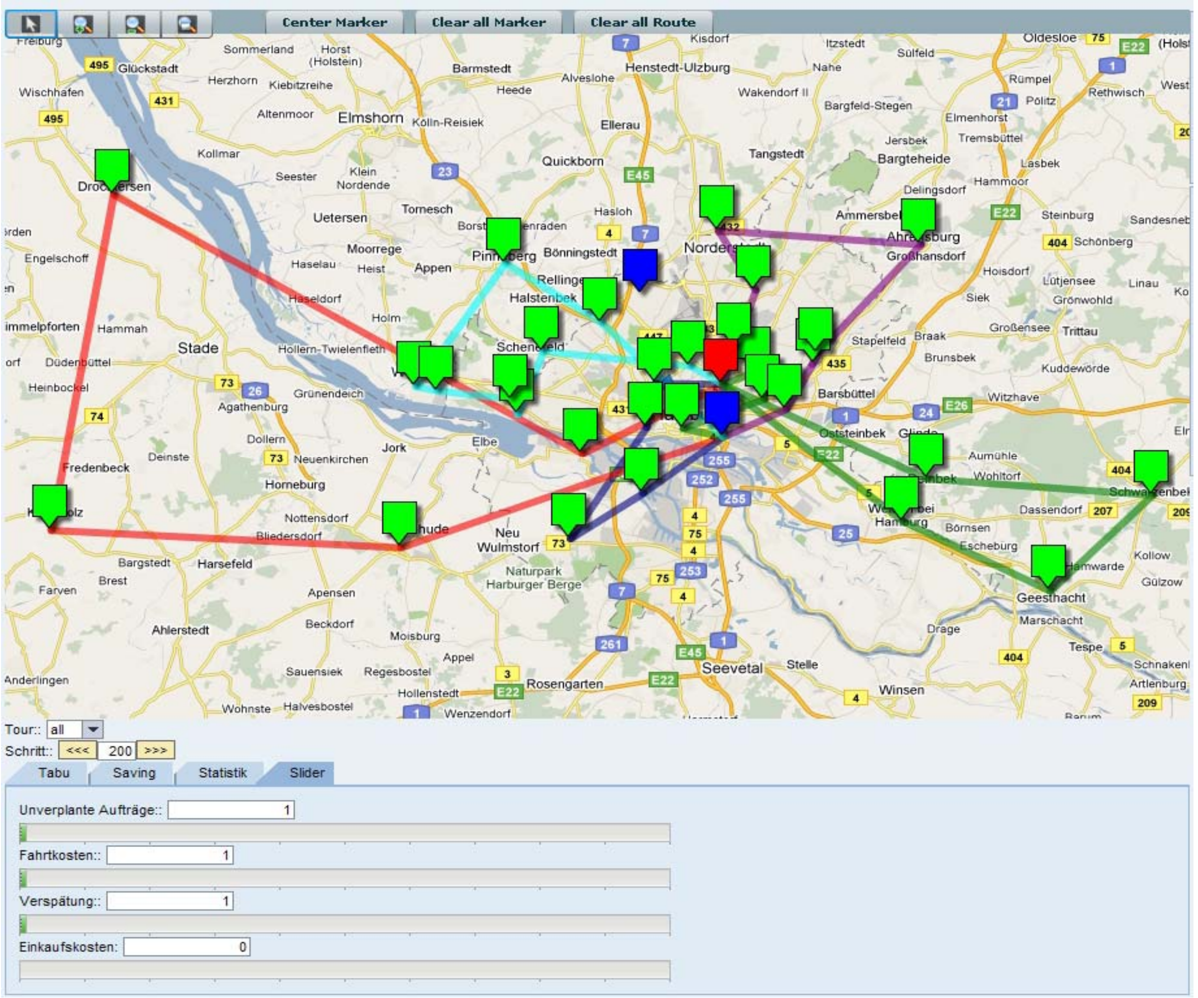
http://vsrv-studprojekt2.fh-wedel.de:8080/touristinformationsystem/home

Bachelor Thesis 2011 at FH Wedel

Application: Oil and gas delivery

Problems for operation:

- Trucks are not where they should be
- Customer changes his order parameters
- New orders have come in



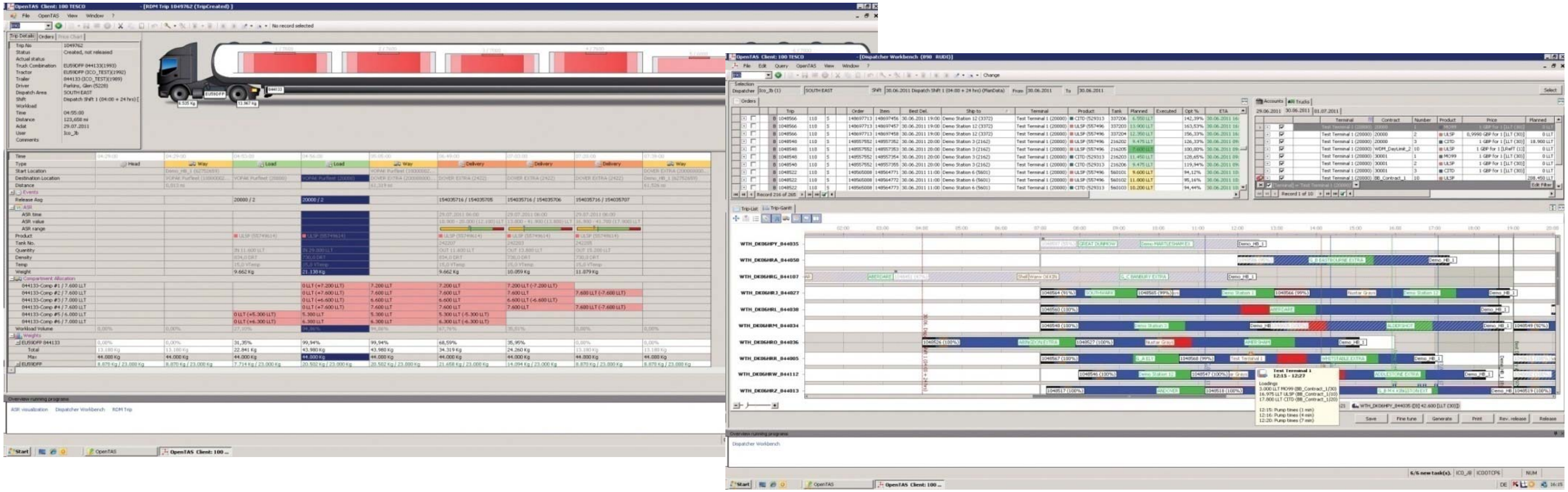
Solution (2010/2011):

- Several FH Wedel graduation theses for a software supplier using SAP
- One using an ant system!

discussion und details in bachelor thesis of Christopher Blöcker, 2011 (in German), outline in publication 2012 (in English)

Integration into existing scheduling system

Application: Oil and gas delivery



Further software requirements:

- Software had to be integrated in running SAP environment
- Answer had to come quickly after an unexpected event occured
- Drivers always had to know where to head next

  discussion und details in bachelor thesis of Christopher Blöcker, 2011 (in German),
  outline in publication 2012 (in English)

Tourist Information System (TIS)

Internal project at FH Wedel (several stages, still running)

more information at: http://www.fh-wedel.de/mitarbeiter/iw/eng/r-d/done/sw-projects/hti/

Current functionality:

- Tourist chooses places of interest from category tree.
- TIS gives details on request from external providers (photos, texts, maps)
- Tourist specifies his preferences (desired staying time, order of places in tour, time windows, etc.).
- System takes default values from the categories/places for all non-specified properties

Tourist Information System (TIS)

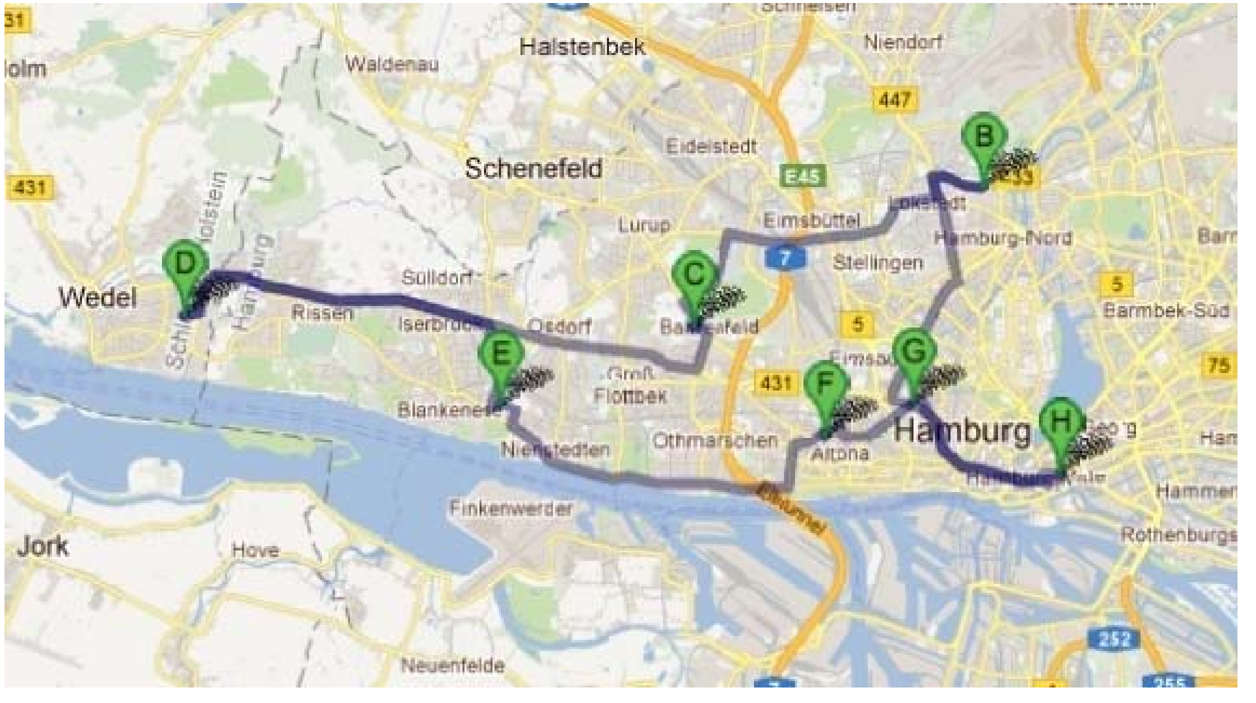
Internal project at FH Wedel (several stages, still running)

more information at: http://www.fh-wedel.de/mitarbeiter/iw/eng/r-d/done/sw-projects/hti/

Current functionality:

- TIS computes a complete tour for individual passengers (pedestrians, car) and for public transport users (automatic connection to Hamburg public transport web services). The tour may be computed for a preselected order als well as for an order automatically scheduled by a tour optimiser.

This is a TSP generalisation!



Tourist Information System (TIS)

Potential future functionality:

- other cities (e.g. cities of partner universities)

- Tourist uses TIS on tour with his smartphone

- TIS gives infos to tourist on tour about items in his current vicinity which he selected to be interesting in advance.

This is dynamic:
A job for ant systems!

- TIS gives infos also on temporal events such as theater performances, etc.

- Tourist may perform bookings on-line / during his tour via TIS.

Ant Systems in Navigation and Logistics

1. Traditional approach revised: Dijkstra and A*

2. Discussion and extension of ant homework assignment

3. How to bring ant systems into practice

4. Logistics problems and the use of ant systems

➡ 5. Conclusion: Lessons learnt

Ant Algorithms and their Applications

When should ant algorithms be applied ?

- For navigation only in the dynamic case: When rapid and unexpected changes matter
- For logistics even in the static case: When the computation of the mathematically best solution is not feasible or not necessary