Wenning, B.-L.; Pesch, D.; Timm-Giel, A.: A distributed routing approach for vehicle routing in logistic networks. In: Proceedings of the IEEE 64th Vehicular Technology Conference. 2006, published on CD

A Distributed Routing Approach for Vehicle Routing in Logistic Networks

Bernd-Ludwig Wenning
Andreas Timm-Giel
Communication Networks
University of Bremen
28359 Bremen, Germany
Email: (wenn,atg)@comnets.uni-bremen.de

Dirk Pesch
Centre for Adaptive Wireless Systems
Cork Institute of Technology
Cork, Ireland
Email: dirk.pesch@cit.ie

***Abstract*— The increasing complexity and dynamics of logistic processes is creating significant new challenges for the management of goods transport. This is leading to increased requirements for the routing of goods and transport vehicles in order to adapt to the dynamics of the changing logistics environment. Current practice of vehicle and goods routing is based on centralised planning and control. This approach is now rapidly becoming too inflexible and complex for maintaining efficient goods transport. In this paper we introduce a novel routing process which implements distributed decision making among transport vehicles, goods items, and other entities in a logistic transport network. The proposed approach enables packages (goods items) and transport vehicles to find their routes autonomously whilst reacting to dynamic changes in their environment.**

I. Introduction

The current management practice in logistic planning is to execute decision-making in a central entity [1], [2]. This requires solving large optimization problems in order to obtain the most efficient solution with regard to vehicle loads and routes. These optimization problems are mostly static with fixed assumptions regarding transport needs and the road traffic situation. When dynamic effects need to be taken into account, such as unexpected transport orders or changes in the road traffic situation, the optimization becomes a rather complex undertaking as routes and possibly also loads have to be recalculated many times for potentially large amounts of vehicles.

In order to deal with this ever more complex problem, logistic processes are currently undergoing a paradigm shift from centralised control of "non-intelligent" items in hierarchical systems towards decentralised control of "intelligent" items in heterarchical systems. Those intelligent items could either be raw materials, components or products as well as transit equipment (e.g. pallets, packages) or transportation systems (e.g. conveyors, trucks). Reichl describes such items as "things that think" [3]. This paradigm shift is facilitated by the availability of a wide range of information and communication technologies, as shown in Figure 1, that can be utilised to devolve decision making down to the level of a vehicle and indeed the individual item in the logistic chain [4], [5].

In the study reported here we are investigating the effects of autonomous decision making by each individual transport vehicle in terms of the route and the loading it wishes to take. The proposed approach has also been inspired by source routing algorithms proposed for mobile ad-hoc networks, where the source decides on the routes of packets. It turns out that the proposed approach achieves good results in terms of vehicle utilisation and throughput.

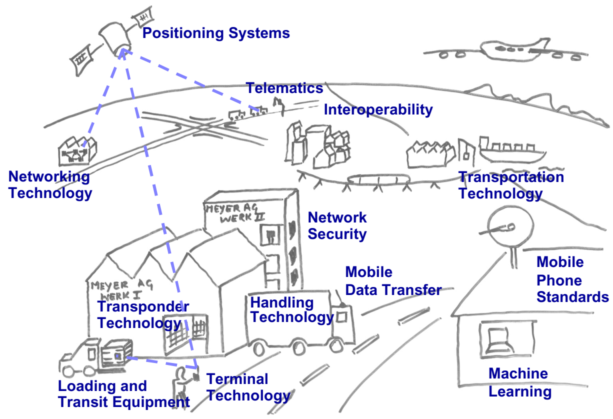

Fig. 1. Information and communications technologies available to logistic processes

II. The Routing Approach

The approach presented here assumes that routing decisions are carried out by vehicles themselves rather than in a centralised planning department. In fact, a vehicle does not just decide on its route, but also considers a choice of packages for its load. Both load and route choice influence a *reward function* which is used in optimizing network performance.

The reward introduced here is used to steer the optimisation process and is largely based on the urgency of delivery of the packages. The closer the due time $t_{due}^{(package)}$, the more urgent the package delivery becomes.

The reward function is defined as

$$R = \sum_{package \in depot} r_{package}^{(local)} - C_{vehicle}$$

with

$$r^{(local)}_{package} = we^{-c(t^{(package)}_{due} - (t_{current} + t^{package}_{travel}))} + r^{(package)}_{economic}$$

as the reward for shipping a locally available package. Here, w is a wighting factor, t_x are the respective times, and $r^{(package)}_{economic}$ represents the economic reward for delivering the package. This economic reward can include additional (positive or negative) reward parts like fees, for example, that are not directly related to the delivery time.

$$\begin{aligned} C_{vehicle} &= w_{time} \sum_{edges \in route} t_{edge} \\ &+ w_{distance} \sum_{edges \in route} d_{edge} \\ &+ w_{toll} \sum_{edges \in route} toll_{edge} \end{aligned}$$

represents the total cost, consisting of travel time t, travel distances d, and road toll $toll$, for routing the vehicle along the chosen route. Given a set of packages, the route choice is basically a traditional travelling salesman problem. However, the complexity of the problem is relatively low as the choice of packages limits the routes that the vehicle can choose. We wish to remark on a further aspect of the reward: as a late delivery should not yield more reward than an on-time delivery, a fine is added to the economic reward when the package is overdue. However, there is a cutoff (not shown in the formula) which limits the reward for overdue packages.

In order to limit the number of constraints in the investigated approach, some basic assumptions have been made:
We assume that there are no restrictions concerning driving times, i.e. vehicles can drive at any time (24/7). Furthermore, we assume that trans-shipment facilities are operational 24/7, so that loading and unloading can take place at any time. We also assume that vehicles can be served simultaneously, so that no vehicle has to wait for service at a trans-shipment point. Loading and unloading times at vertices are fixed values per package, i.e. the time taken is proportional to the number of trans-shipped packages. Empty vehicles evaluate the reward function based on the packages currently available for loading. The maximum reward must exceed a threshold before a vehicle starts loading. This is to make sure that lightly loaded vehicles do not leave a source.

III. Evaluation Scenarios

We have taken an incremental approach for the introduction of autonomous logistics starting with a very basic scenario and leading on to increased autonomy in decision making for vehicles. To do so we consider four scenarios, which introduce increasing flexibility in the decision making process. For the introduction of the scenarios, we use the following terminology

- Vertex: A location in the transport network where either several edges (roads) intersect/meet or trans-shipment (loading and/or unloading) is carried out or both.
- Active vertex: A vertex with trans-shipment facilities.
- Passive vertex: A vertex without trans-shipment facilities.
- Edge: A directed connection between vertices, e.g. roads.
- Package: A piece of good that can not be broken down into smaller parts.

A. Scenario 1

In this scenario, the most basic, the reward, based on the reward function defined above, is only determined when a vehicle is newly loaded. Only packages that are present at the same location as the vehicle are taken into account. This means, the vehicle searches for an optimal route to deliver packages from source to destination, under the constraint that the volume of packages does not exceed the vehicle's capacity. At the last vertex of the route, the vehicle has to be totally empty, and the vehicle starts a new load and route evaluation process.

B. Scenario 2

In the second scenario, the algorithm is extended to include packages which are available at the vertices along the route. This implies some kind of "lookahead" where the vehicle has to obtain knowledge about what package would be available for shipment. The vehicle sends requests to the different vertices regarding package stocks and the destinations and due times of those packages. For these requests, a concept is borrowed from communication networks: the route discovery in source routed ad-hoc networks [6]. The vehicle sends a request for information about loadable packages, this request propagates from vertex to vertex similar to a route request in communication networks. Vertices reply with an offer of suitable packages for the vehicle.

As we are now considering the possibility of collecting packages along the route, the reward function is changed to

$$R = \sum r^{(local)}_{package} + \sum r^{(distant)}_{package} - C_{vehicle}$$

where $r^{(local)}_{package}$ and $C_{vehicle}$ are the same as before, but here we have an additional term

$$r^{(distant)}_{package} = we^{-c(t^{(package)}_{due} - (t_{est-pickup} + t^{package}_{travel})} + r^{(package)}_{economic}$$

which represents the reward for those packages that can be collected along the way. A further constraint is considered here in that the vehicle's capacity must not be exceeded by the volume of the packages on any section of the route.

After having chosen a route, the vehicle must send a notification with package identifiers for those packages it intends to collect along the route to the respective active vertices on its route. This has to be done to avoid that other vehicles schedule the same packages for collection. We envisage that the usual fixedwireless communications technologies will be used for this.

C. Scenarios 3+4

Scenarios 3 and 4 use the same reward functions as scenario 2, but here the reward is evaluated more often. In scenario 3, the reward is evaluated wherever a route change is possible, but the sequence of active vertices has to be preserved. This means the algorithm can react to changes in route costs (like changed expected travel time) by re-routing the vehicle. In Scenario 4, the reward is evaluated at each (active and passive) vertex. There are no constraints concerning preservation of the sequence of active vertices, but the re-evaluation can lead to completely new routes which might have totally different sequences of vertices. This approach allows vehicles to take advantage of the opportunity to pick-up urgent packages along the way and also to evade road traffic congestion in a dynamic manner. However, precautions need to be introduced in order to maintain stability of the routing approach.

IV. Solving the Optimization Problem

The ideal route and load for a vehicle to take is the one that achieves the best possible vehicle utilisation and throughput. This means each vehicle needs to maximes the reward for delivering packages. As the solution space for the reward function can be large, especially if the network or the amount of packages is large, we propose to use an effective heuristic based on two variations of a genetic algorithm approach.

A. Genetic Algorithm No. 1

In the first genetic algorithm, which is used to find solutions to the vehicle routing problem for scenario 1, each available package at a source vertex is assigned a positive integer indentifier (ID). Based on these IDs, a set of packages that do not exceed the the capacity of the vehicle forms a phenotype of the genome. This means the phenotype consists of integer elements, which can be either a package ID or 0 - meaning “no package”. Except for the 0, each ID can only appear once in a solution, which means packages have unit size, e.g. solution

$$S = (2, 3, 0, 5, 0, 0).$$

To initialize the algorithm, a number of solutions is randomly generated. In addition, the all-zero solution - meaning no packages are loaded and the vehicle stays where it is - is added to the set of initial solutions as it could be that no solution yields a positive reward. In the experiments described later in this paper, initial population sizes between 10 and 20 solutions were used.

In order to evaluate a solution, an optimal route covering all destinations for the solution’s packages is found. For this, the current travel time estimation between the destinations is obtained by requesting the typical travel times from adjacent vertices and then the sequence of destinations is optimized by an “inner” genetic algorithm. In this study, we have considered that each active vertex has several packages for a particular destination so that the number of destinations included in one solution is not very high. This results in fast convergence of the inner genetic algorithm. In other situation with a high diversity of destinations, this approach may be inefficient.

The result of this optimization yields the required sequence of hops for this particular route, along with the estimated travel times and cost for the route based on the cost function introduced earlier. With this, the reward is calculated based on one of the two reward functions above.

In the first iteration, the selection operator promotes the best half of the solutions to the set of best solutions for the next iteration. In the following iterations, the best half of the best solutions remains in the set, while the rest is filled up with the best quarter of solutions from the current generation. In order to generate new solutions based on the current set of solutions, a crossover operator is used. The operator works as follows: The selection of the first parent is done by an iterator that goes forward through the set of best solutions while the selection of the second parent is done by an iterator which goes backwards. The first half of the offspring solution is taken from the first parent, while the second half is filled up with IDs from the second parent which are either 0 or not contained in the first half. The genetic algorithm terminates when either a specific number of runs has been reached or a convergence of the reward is discovered.

B. Genetic Algorithm No. 2

For scenarios 2 to 4, the possibility exists of loading packages along the route. As this implies that the total number of packages to be handled on a route can exceed the vehicle’s capacity, it is not possible in this scenario to take a list of packet IDs for the route as the phenotype. The genetic algorithm is constructed differently: It is based on the package destinations available at the starting active vertex. At algorithm initialisation, all package destinations for the local packages are collected. The initial population is created by listing the different destinations for packages. It is also allowed that a destination does not appear in the solution, which is represented by -1 in order to obtain the correct length for the phenotype. If the sequence of destinations leads to a route, then the route cost and the estimated travel times are determined. Based on this, the vertices along the route can be asked for packages for destinations along the determined route. This is based on the source routing approach in mobile ad-hoc networks through a look-ahead mechanism taking advantage of fast communication networks. For each of the local and remote packages, a single package reward is then calculated. As there are packages that are only travelling along a part of the route, there are some which can be combined to one “virtual package”. This creation of virtual packages is not trivial and should be subject to an optimisation as well, however, in this early implementation, a solution which does a linear search without optimising the virtual package’s reward is used. Having created the virtual packages, the overall reward is maximized by appropriate selection of virtual packages.

Having determined the overall reward for the solutions, the selection and crossover operations are carried out in the same way as those in the Genetic Algorithm No. 1. When convergence is established or the generations limit is reached, the vehicle uses the best available solution to set its route and its loading schedule.

As the packages at the remote vertices are obviously not immediately loaded, they have to be reserved for this vehicle at the vertices along the route. The vehicle announces a list of packages which it plans to collect by transmitting the list to all vertices along the route using a suitable communication technology. Each vertex extracts the package IDs relevant to itself and marks the respective packages as reserved. This way prevents a package from being scheduled for multiple vehicles.

V. Evaluation

The performance of the four scenarios and the two genetic algorithms is evaluated by a discrete-event simulation. The evaluation is based on a model of a transport network depicted in Figure 2. In this transport network, the vertices represent 18 German cities with the edges being the motorways between them. Each of the vertices is an "Active Vertex", i.e. packages

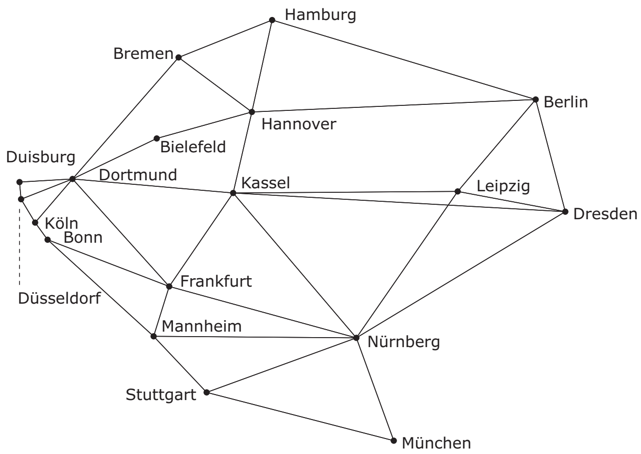

Fig. 2. Map of vertices and edges in the simulation setup

can be trans-shipped there. Furthermore, packages are generated at each vertex with generation rates proportional to the size of the city which is represented by the vertex, ranging from 2 to 34 packages per (simulated) hour. Several vehicles, each with a capacity of 60 packages and an average travel speed of 100 km per hour, are distributed among the vertices. The amount and location of those vehicles depends on the transport volume and the scenario in use.

A. Simulation Result for Scenario 1

In the first scenario, it is assumed that the vehicle evaluates the reward generated only by locally available packages. The simulations of scenario 1 showed that this routing approach is not very efficient. Here, 68 vehicles were required to handle the transport volume (even without demanding on-time delivery), while the average utilisation of those vehicles' capacity is just about 45.3%. The reason for this poor performance is rather obvious as the vehicles do not pick up any packages along a planned route and therefore travel with suboptimal use of their capacity most of the time. Additionally, this scenario is very sensitive to inhomogenously distributed pickups and deliveries. Vertices producing more packages than they receive are short on vehicles, while the vehicles concentrate on vertices which mainly receive packages. This can be overcome by either introducing the additional constraint that the vehicle has to return to its home vertex regularly or by giving the overloaded vertices the ability to request vehicles from other vertices. In the study presented here, the issue was solved by always planning round trips and therefore assuring that the vehicle comes back to its home vertex. This approach, however, increases the number of empty trips and therefore contributed to the low efficiency in terms of vehicle utilisation. Requesting vehicles from other vertices would have a similar effect.

The convergence of the Genetic Algorithm in scenario 1 obviously depends on the number of packages the vehicle is taking into account. If there are few packages than the vehicle's capacity, convergence is reached close to almost immediately. For a large number of packages, quite a large number of generations has to be created until convergence of the reward is established. However, this is not necessarily proportional to the number of packages. In fact, some of the longer optimisation runs used package amounts that were only a little larger than the vehicle's capacity. Nevertheless, the convergence in this scenario was fast, as the maximum observed was 10 generations of 15 solutions each.

B. Simulation Results for Scenario 2

The second scenario is more complicated than the first as packages that can be collected along the route have to be considered when calculating the reward. This creates on one hand the problem that the vehicle has to obtain information about the packages that can be collected along the route and on the other hand there is an interdependence between the route to be planned and the packages which are relevant for planning. Furthermore, the constraint that the vehicle can handle more packages than its capacity along the route, but not on any one segment of the route increases the complexity of the problem further.

However, scenario 2 shows much better results than scenario 1 under exactly the same conditions (also always planning round trips). The number of required vehicles is now only 52, which means a reduction of around 23.5% relative to scenario 1, while the average utilisation of the vehicle capacity has reached more than 81%. This is not surprising as the ability to load packages én route results in improved usage of the vehicle's capacity. This performance can be further improved by an optimization approach considering assembly and selection of virtual packages.

In scenario 2, convergence does largely not depend on the number of packages, but on the amount of different destinations, as the optimisation is based on the package destinations as described in section IV-B. As there are not that many destinations, convergence is reached with less generations (a maximum of 8 generations of 10 solutions each was observed). The processing time of the algorithm, however, was significantly longer than in scenario 1 (more than one order of magnitude). This is due to the creation of virtual packages, which also strongly depends on the amount of packages to be taken into account. Virtual package creation, as previously stated, was done by a linear search, which means that there is still potential for increasing performance by using alternative optimisation methods or efficient heuristics.

C. Results for Scenario 4

In scenario 4, the reward is evaluated at each vertex. This means the computational effort in each vehicle is significantly increased. Meanwhile, the stability issue that was solved by introducing round trips in Scenarios 1 and 2 is again present in this scenario. Planning of round trips does not help much in this scenario, as replanning is carried out at each vertex. Without any further constraints this might cause the vehicle to never return to its home vertex, even if the home vertex is always set to be the endpoint of the new route. Therefore, if the vehicle distribution should be kept under control by forcing vehicles to return to their home vertex, then this constraint has to be tightened compared to earlier. Possible options for constraints are a maximum travel time or a maximum travel distance. Here, the latter option has been chosen.

The simulation of scenario 4 shows performance results that lie between those of scenarios 1 and 2: 65 vehicles are required, the average capacity utilisation is 56%. An explanation for this poor performance is that the vehicles are optimising a complete route, but they are only travelling to the first vertex of this route (the first hop) before they re-evaluate the route and load. As the optimal solution for a complete route does not necessarily mean a good utilisation of the first hop and only the first hop is really used, the lower performance in comparison to scenario 2 can be explained. This was further investigated by increasing the reward for local packages and thus trying to improve the usage of the first hop. This improved the performance, but the results of scenario 2 were still not reached. A multiplication of local package rewards by a factor of 1.5 led to a reduction of required vehicles to 58 and an increase of capacity utilisation to 67%.

As the algorithm in scenario 4 is basically the same as in scenario 2, similar results were expected for its convergence and the processing time for an optimisation run. The acquired simulation results show that this is indeed the case for convergence but with a shorter processing time. The reduction in processing time is caused by less packages being available for the creation of virtual packages. This is probably a side-effect of the maximum travel distance: Routes that exceed this maximum are not evaluated but directly given a bad reward, which means long routes with high amounts of packages, which would cause a long construction of virtual packages, are more likely not evaluated at all. This observation was made independent of whether the local package reward was increased or not.

VI. Conclusions and Outlook

An approach for autonomous vehicle routing in logistic networks using decentralized decision-making is proposed. We have presented the overall approach and early results of the evaluation of some considered scenarios. While the knowledge about packages that may be collected at intermediate locations of a route can increase the efficiency of the vehicle usage, the results for the last scenario let us draw the conclusion that a continous change of decisions can even decrease performance as the decision might already be changed before its advantages can take effect.

We are currently considering an even more distributed approach for future research, where the packages themselves are assumed to have some intelligence and are able to communicate and to make routing decisions. In this approach, both the packages and the vehicles determine routes and then negotiate with each other about being transported.

Acknowledgment

This research was supported by the German Research Foundation (DFG) as part of the Collaborative Research Centre 637 “Autonomous Cooperating Logistic Processes” and the European Commission in the context of the IST Network of Excellence CRUISE (Creating Ubiquitous Intelligent Sensing Environments).

References

[1] P. Toth and D. Vigo, *The Vehicle Routing Problem*, Society for Industrial and Applied Mathematics, 2002

[2] E. Taniguchi and H. Shimamoto, *Intelligent Transportation System based Dynamic Vehicle Routing and Scheduling with Variable Travel Times*, Transportation Research Part C, 12 (2004) pp. 235 - 250

[3] H. Reichl and J. Wolf, *Things that think*, TU-Berlin - Forschung aktuell, 49 (2001) pp.1887-1898

[4] B. Scholz-Reiter, K. Windt and M. Freitag, *Autonomous Logistic Processes - New Demands and First Approaches*, in Proc. of 37th CIRP Internation Seminar on manufacturing Systems, Budapest, Hungary, 2004, pp. 357-362

[5] C.D. Tarantilis and C.T. Kiranoudis, *Using a Spatial Decision Support System for Solving the Vehicle Routing Problem*, Information and Management, 39 (2002) pp. 359-375

[6] S. Rea and D. Pesch, *Multi-Metric Routing Decisions for Ad-Hoc Networks using Fuzzy Logic*, in Proc. of IEEE International Symposium on Wireless Communication Systems, Mauritius, Sept. 2004