Efficiency for Routing Networks Management in Supplier-customer Distribution Systems

Radu BÎLBÎIE[1]*, **Cătălin DIMON**[1,2], **Dumitru POPESCU**[1,2]

[1] Department of Automatic Control and Systems Engineering, University Politehnica of Bucharest,
313 Splaiul Independenței, 060042 Bucharest, Romania
bbbradu@yahoo.com (**Corresponding author*), catalin.dimon@upb.ro, dumitru.popescu@upb.ro

[2] Academy of Romanian Scientists, 3 Ilfov Street, 050044 Bucharest, Romania

Abstract: This paper proposes a cost-efficient solution for the routing network associated with the management of supplier-customer distribution systems. The main objective is to determine the minimum cost path in a transportation network affected by various incidents, either critical and/or non-critical. Based on the supplier and customer demands and the evaluated transport costs, the corresponding solution was computed by using Dijkstra's algorithm. The novelty of the proposed solution lies in the possibility to continuously adjust transport costs per road segments, even eliminate some road segments, and to re-generate a new optimal solution in real time (online replacement path problem).

Keywords: Supplier-client, Transport network, Reconfigurable management system, Online replacement path.

1. Introduction

The transportation problem (TP) is closely related to the distribution of products between supply and distribution locations. The TP aims to minimize the total transport cost and at the same time to optimize the problem. Since the `60s, a procedure for optimizing a routing network of the fleet of vehicles intended for gasoline supply was proposed by Dantzig & Ramser (1959).

The shortest path problem is presented in detail in (Ortega-Arranz, Llanos & Gonzalez-Escribano, 2015), starting with the classic Dijkstra's algorithm and moving to more advanced solutions that are currently applied to road network routing, including the use of heuristics and precomputation techniques.

To solve the transportation problem, the initial basic feasible solution (IBFS) is a crucial step to determine the optimal solution. However, some developed methods based on IBFS do not always produce a good initial solution. The study of Amaliah, Fatichah & Suryani (2022) presents a new method called Supply Selection Method (SSM), which is proposed to obtain a better initial solution for balanced TP. It has been compared with other IBFS-based methods, including Vogel's Approximation Method (VAM), Juman and Hoque Method (JHM), Total Opportunity Cost Matrix – Minimum Total (TOCM-MT) and Bilqis Chastine Erma (BCE) method to evaluate its performance. The SSM method was examined for 45 total cases including 31 cases from some journals, 4 randomly generated cases and 10 real data samples from XYZ company. The results of the study show that SSM provided a better initial base solution than other methods, with 41 out of 45 cases reaching the Optimal Solution. The evaluation shows that SSM provided more results with lower total minimum cost than LCM (Least Cost Method), VAM, JHM, TOCM-MT and BCE.

Various heuristic shortest path algorithms that have been developed were presented in a survey review published in 2005. The goals were to identify the main features of different heuristic strategies, develop a unifying classification framework, and summarize relevant computational experience (Fu, Sun & Rilett, 2005).

In (Huang, Lai & Cheng, 2009) the authors present various fundamental algorithms for research and development in the field of electronic design automation (EDA), from classical graphic theories to practical heuristic approaches and then up to theoretical mathematical programming techniques. Heuristic algorithms that produce suboptimal but reasonably good results are usually adopted as practical approaches. Several selected heuristic algorithms are also covered. Mathematical programming algorithms are explored, which provides theoretical support for the optimal solution of the problem and focus on mathematical programming problems that are most common in EDA applications.

For the Replacement Paths problem, a $O(TSPT(G) + m + l^2)$ time and $O(m + l^2)$ space algorithm is

https://doi.org/10.24846/v33i2y202405 ICI Bucharest © Copyright 2012-2024. All rights reserved

suggested, where $TSPT(G)$ is the asymptotic time to compute a single source shortest path tree in G (Kare, 2016).

A new robust path problem, the Online Replacement Path problem (ORP), was also studied by considering a setup in which an adversary can choose to remove a single edge in the network and reveal the identity of the failed edge just before the routing mechanism attempts to use it (Adjiashvili, Oriolo & Senatore, 2013).

The optimal or sub-optimal solution can only be used in ideal cases, in which there are no incidents affecting the road network. In case of the occurrence of such incidents, a new optimum must be determined to allow the reduction of additional costs. If the analysis is done through procedures specific to linear mathematical programming (simplex type procedures or similar ones), the reconfiguration of the optimization problem can become complex, involving an important computation effort.

In order to find an efficient solution, this paper proposes an approach based on the following considerations:

- the management of the distribution system is proposed in the form of a causal graph, each arc in the graph representing the connection between the nodes associated with suppliers and customers;
- at the level of the transport system, the minimum cost route for each supplier-customer pair is determined by using Dijkstra's algorithm, considering the state of the existing infrastructure (roads and intersections);
- in the event of an incident, the model associated with the affected road structure is reconfigured, the network segment is updated and the new trajectory is re-estimated.

The paper is organized as follows. Section 2 presents the algorithm employed for determining/adjusting the minimum cost route in a transportation network affected by various incidents (critical and/or non-critical) and in Section 3 this algorithm is illustrated through a case study involving a simple road network configuration. Section 4 presents the simulation results for 7 different road network configurations, ranging from small (10 nodes) to large (15000 nodes). Finally, Section 5 includes the conclusion of this paper and suggests possible future directions.

2. Routing Network of the Supplier-customer Distribution System

For the management model associated with the activities of supply-retail services, an oriented graph is proposed, which includes the following objects: suppliers, intermediate and final customers as well as the specific interconnection chains.

In Figure 1, a transportation network is associated with a management system. Each link between system's entities (warehouses, suppliers, customers) will have an associated transportation cost per product unit, based on existing road infrastructure.

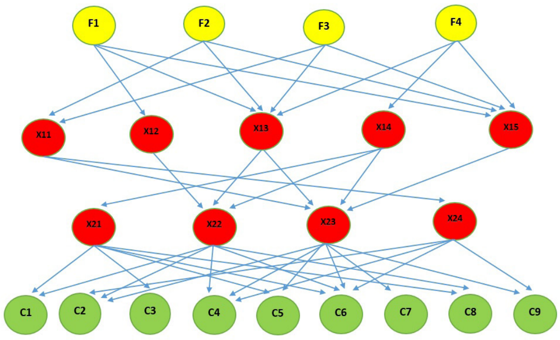

Figure 1. Representation of the transportation network in the distribution system

These costs will start with an estimated value, resulting from data collected in the past, and will be continuously updated (if necessary) according to traffic events.

After each update, the total cost for minimal path is updated accordingly and the optimal route will be kept only if the total cost increase will be lower than an "acceptable" value (that value was fixed at 1% for simulations).

In the event of the occurrence of unforeseen incidents (or high cost increase for the optimal path), which affect the optimal route, the structure of the network associated with the road infrastructure is reconfigured, a new solution is determined to avoid the affected road and the matrix of the management system is updated.

This paper shows that by selecting a strategy which is simple enough (and fast enough) to be used repeatedly, the minimum cost path from supplier to customer can be found, considering a road infrastructure prone to incidents.

Several strategies were taken into consideration, many of which were mentioned in the introductory part. As the cost for each segment is adjusted continuously, based on different signals received from road infrastructure, sensors, weather status etc., it was necessary to eliminate all strategies based on pre-computation paths or static estimations. Heuristic-based methods (like A*) can be used, under the condition of re-computing the previously estimated values, according to modifications in road infrastructure. In the A* case, the heuristic function $h(n)$ must be checked for admissibility and consistency after modifications.

Dijkstra's algorithm was selected based on its simplicity and ease of implementation. It has a relatively high standard time complexity, $O(N^2)$, but can be reduced to $O(N \log N + E)$ if one uses Fibonacci heaps. In simulations, for large-enough road networks (15000 nodes and 54684 edges) total computing time to determine minimal cost path was a few seconds on a home laptop, which is low enough for a real-time transportation scenario.

2.1 The Minimum Cost Route for the Distribution Network

The network route segment from Figure 2 is considered (from F2 supplier to C6 customer), extracted from the transport distribution network. In this reduced distribution network, the arcs represent roads of the routes, and the nodes different crossing points (localities, intersections, etc.).

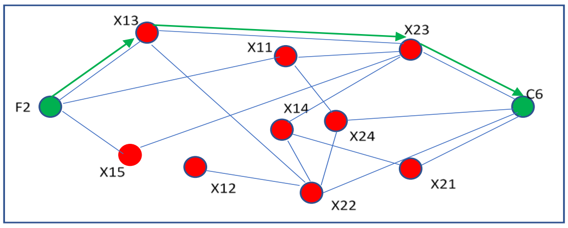

Figure 2. Minimum cost route from the distribution infrastructure

For each route the transport-cost for a product unit is associated and the minimum road segment / cost between two nodes is determined using a form of Dijkstra's algorithm (Algorithm 1), which allows stopping the search when the desired node is reached:

Algorithm 1. A form of Dijkstra's algorithm

```
function Dijkstra(G, source, target):
   Q ← new List
   foreach v in G.Nodes:
     dist[v] ← INFINITY
     prev[v] ← UNDEFINED
     Q.Add(v)
   dist[source] ← 0

   while Q is not empty:
     u ← min(dist[x]) for x in Q
     if u == target:
        return dist[u], prev[u]
     Q.Remove(u)
     for each (v in Q) and ((u, v) in G.Edges[u, v]):
        new_dist ← dist[u] + G.Edges[u, v]
        if new_dist < dist[v]:
           dist[v] ← new_dist
           prev[v] ← u

   return 0, 0
```

G is the network representation, and it contains network nodes (G.Nodes[]) and network edges (G.Edges[,]). Q is the list of unchecked nodes. dist[] is the optimal cost for the route between the source and each node, and prev[] contains the nodes that compose the route (after optimizing each route).

In order to obtain the minimum cost route from source node to target node, the prev[] table should be used. Here, for each node, the previous node on the minimum cost route can be found (Algorithm 2). So, it is necessary to go backwards, from target to source, and insert each previous node before all the others in the list representing minimum cost route:

Algorithm 2. Obtaining the minimum cost route

```
function GetPath(prev, source, target):
 S ← new List
 u ← target
 if (prev[u] not null):
    while u not null:
       S.Insert(0, u)
       u ← prev[u]
 return S
```

Sequence S is the list of vertices forming one of the shortest paths from source to target, or the empty sequence if no path exists.

Both these functions will be used with regard to the function related to incident handling, which is presented in the next section.

2.2 Routing Network with Incident Events

There are situations in which changes appear in relation to the initial chosen route from Figure 2. Some roads can become very congested or even inaccessible due to accidents or natural disasters. In these situations, the initial route must be reconfigured, and the total transport cost must be adjusted.

When an incident occurs, if the minimum cost route is affected, one of the following situations can be considered:

Case 1 - the incident affects the current road segment (Figure 3):

- The vehicle returns to the previously traveled node of the minimum cost route;
- The previous node (X23) becomes the new source node;
- The new cost and the new minimum cost route to the destination is determined;
- The new route is transmitted to the vehicle;
- The total cost is adjusted, by adding the additional cost to the cost corresponding to the journey made until the occurrence of the incident;
- The new minimum cost route is established, which contains the nodes which were already passed and the nodes to be passed.

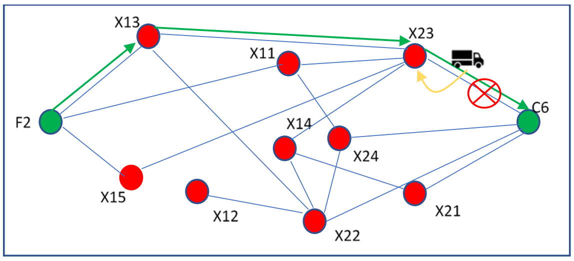

Figure 3. The incident affects the current road segment

Case 2 - the incident affects a road segment that will be followed (Figure 4):

- The new cost for both options should be determined, the next node (X23) and the previous node (X13) of the minimum cost route which becomes the new source node;
- The option that offers the minimum cost is selected;
- The new route is transmitted to the vehicle;
- The total cost is adjusted, by adding the additional cost to the cost corresponding to the journey made until the occurrence of the incident.
- The new minimum cost path is established, which contains the nodes already passed and the nodes to be passed.

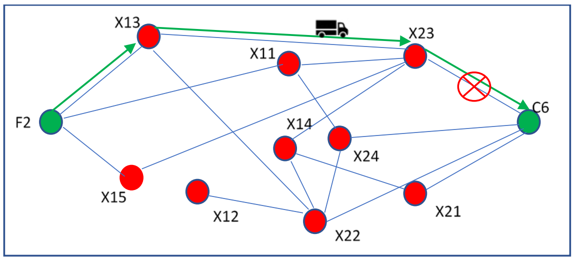

Figure 4. The incident affects a future road segment

Algorithm 3 related to incident handling was included below:

Algorithm 3. Handling an incident

```
function Incident(G, lastnode, nextnode, source, target, u, v):
  if lastnode == u:
      source ← lastnode
      dist, prev[] ← Dijkstra(G, source, target)
      S[] ← GetPath(prev, source, target)
  else:
      source ← lastnode
      dist_last, prev_last[] ← Dijkstra(G, source, target)
      S_last[] ← GetPath(prev, source, target)
      source ← nextnode
      dist_next, prev_next[] ← Dijkstra(G, source, target)
      S_next[] ← GetPath(prev, source, target)
      if dist_next <= dist_last:
          dist ← dist_next
          prev[] ← prev_next[]
          S[] ← S_next[]
      else:
          dist ← dist_last
          prev[] ← prev_last[]
          S[] ← S_last[]
```

3. Case Study

To ilustrate the mechanisms for adjusting the minimum cost route, for the presented case study a simple network between supplier (F) and customer (C) was selected, containing 8 intersections (intermediary checkpoints). Associated costs for each road segment are represented in Figure 5:

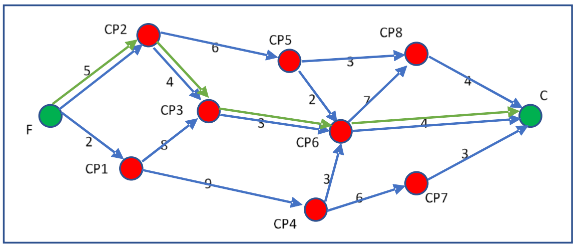

Figure 5. Initial minimum cost route

The minimum cost for this route is 16 u.p., determined by the sequence F-CP2-CP3-CP6-C.

If an incident occurs before the vehicle starts the transport on the given route, then a new minimum cost route is generated, considering the changes that have occurred. For example, if the route segment between CP6 and C becomes impassable, the new minimum cost route becomes F-CP2-CP5-CP8-C, for a total cost of 18 u.p. (Figure 6):

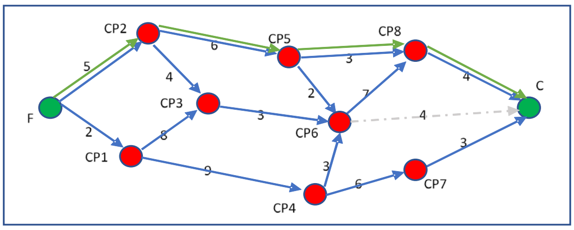

Figure 6. A new minimum cost route after an incident

If the incident occurs after starting the transport on the given route, then two situations are possible:

3.1 The Occurrence of an Incident on the Current Road Segment

The initial F-C cost was 16 u.p. on the route F-CP2-CP3-CP6-C. The occurrence of an incident on the road segment on which the carrier is located is illustrated in Figure 7.

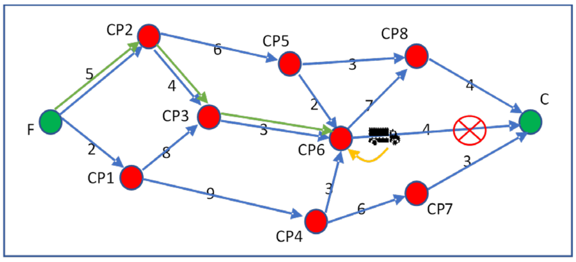

Figure 7. The occurrence of an incident on the current road segment

If an incident occurs on the CP6-C segment, then the carrier returns to the CP6 point and a new minimum cost route is generated (the CP6 node becomes the source, and C becomes the destination for Dijkstra`s algorithm) (Figure 8):

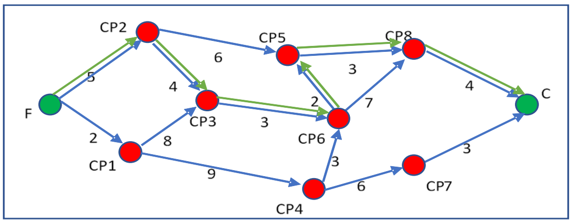

Figure 8. The new minimum cost route

The new minimum cost for F-CP2-CP3-CP6 will be 12 u.p. to which 4 u.p. are added for CP6-C (the segment where the incident occurred, which is half traveled, but twice, so the road cost is only added once) and the new minimum cost for the segment CP6-CP5-CP8-C will be 9 u.p., resulting in a total cost of 25 u.p.

In the transportation network, the new value corresponding to the route between F and C is updated, so that it reflects the new cost, to be able to generate new solutions in case new requests from customers appear.

3.2 The Occurrence of an Incident on One of the Future Road Segments

If the incident occurs on a segment that affects the optimal route but allows the continuation of the route to the next checkpoint (Figure 9), there are two continuation options: returning to the previous checkpoint or continuing to the next checkpoint. For both cases (Figure 10 and Figure 11), the new minimum cost is calculated and the variant with the lower cost is chosen.

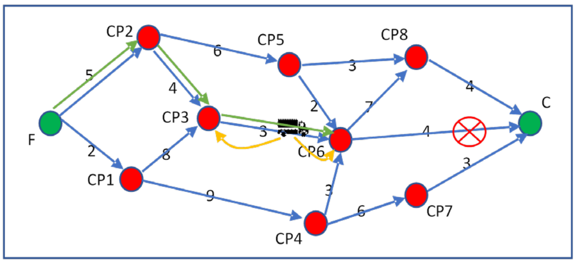

Figure 9. The incident affects one of the future road segments

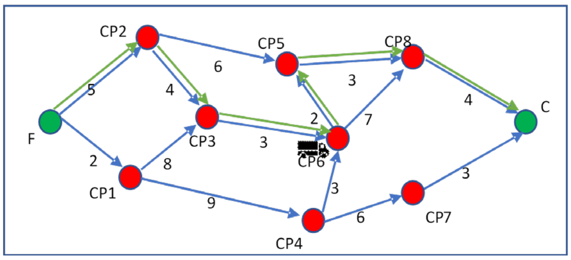

Figure 10. Continuation to the next checkpoint

ICI Bucharest © Copyright 2012-2024. All rights reserved

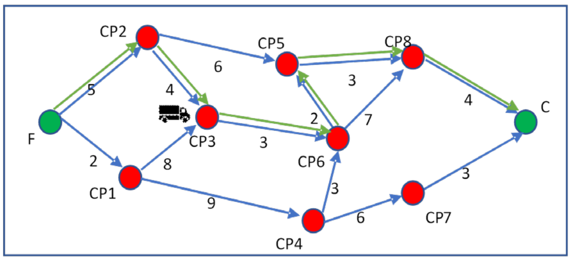

Figure 11. Return to the previous checkpoint

Alternative 1

The new total cost is 21 u.p. if the route continues up to point CP6 and then the new optimal route is calculated, the cost being generated by the sequence F-CP2-CP3-CP6-CP5-CP8-C.

Alternative 2

The new total cost is 21 u.p. (for route F-CP2-CP3-CP6-CP5-CP8-C) to which 3 u.p. are added for the segment CP3-CP6, which is half traveled, but twice, so the road cost is only added once, i.e. a total cost of 24 u.p.

Among the two options, the option with the lowest cost is chosen (Option 1), that is, continuing the road up to CP6 and then calculating the new minimum cost route.

If there was another configuration of costs per segments, for example if the cost for CP2-CP3 would have been 1 u.p. and the cost for CP2-CP5 would have been 3 u.p., a different solution would be obtained (Figure 12):

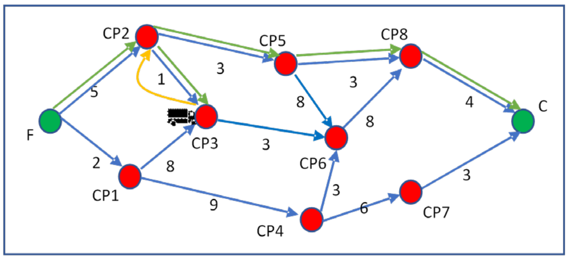

Figure 12. Another configuration of costs per segments

The new minimum cost route from point CP3 would have been CP3-CP2-CP5-CP8-C, for a cost of 11 u.p. and a total cost of 20 u.p. (by adding F-CP2-CP3 and 3 u.p. for the segment CP3-CP6, which is half traveled, but twice, so the road cost is only added once), that is lower than 21 u.p. if the road were to be continued up to point CP6 and then the new minimum cost route would be generated. For this reason, both options must be calculated and then the minimum cost option should be chosen.

4. Simulation Results

In order to estimate the duration of additional processing, a simulator was built, which is composed of 3 applications that run simultaneously (Figure 13):

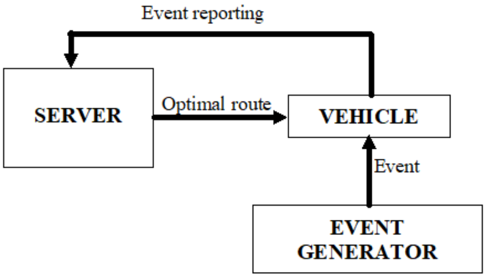

Figure 13. Event reporting

- **SERVER**

- Generates the optimal route
- Communicates the minimum cost route to the vehicle
- Receives any events from the vehicle
- Reconfigures the minimum cost route and communicates it to the vehicle

- **VEHICLE**

- Receives the minimum cost route from the server
- The route begins
- Receives events from the event generator
- Sends the events to the server
- Receives route corrections from the server
- Runs the new route

- **EVENT GENERATOR**

- Generates random events
- Communicates events to the vehicle

The event generator creates non-critical events affecting random segments, modifying transportation cost per segment and, at a random moment, a blocking event for one of the road segments from minimal cost path and communicates those to the vehicle. The vehicle transmits the events to the server. The server updates the initial data (to accurately generate new routes in case of new requests) and then reconfigures (if necessary) the minimum cost route for the vehicle in motion.

The study was carried out for different configurations of the road infrastructure: the configuration chosen for the case study (Figure 5), two slightly more complex configurations that include 27 nodes and 50 nodes, respectively and four larger configurations for networks with 1000, 5000, 10000 and 15000 nodes, respectively. The road infrastructure and costs associated with each segment were randomly generated. The "simple random sampling" method was used to generate for each node 2-4 up to 4-6 (for larger networks) connections and initial costs for every network configuration. The cost values are generated using a rand routine for a given interval.

Due to the relatively high number of simulations (1000 for each network configuration), any particular case of cost configuration has an extremely low effect on the average results.

The meaning of the elements in the tables:

SAME – the event affects the road segment on which the vehicle is located;

FUTURE – the event affects a road segment to be traveled;

BEFORE – the new minimal cost of the transport after the occurrence of an event that affects the ideal route, if the event occurs before the start of the transport;

BACK – the new optimal cost if, following the occurrence of an event, the vehicle has to return to the previous checkpoint and from there the optimal route is reconfigured;

FORWARD – the new optimal cost if, following the occurrence of an event, the vehicle can move to the next checkpoint and from there the optimal route is reconfigured.

It can be noticed that in the event of an incident affecting the initially generated minimum cost route, even if the transport has not started, the newly generated optimal one adds a cost difference of 14.7%. This is due to the reduced complexity of the road network, as it is difficult to find alternative routes with a cost which is similar to that of the ideal route. The more detailed the road infrastructure is (higher number of junctions and arches), the smaller the cost difference in the event of an incident.

The difference from the minimum cost is high if the event occurs after the vehicle has started the transport, if it has to turn back from the road to find a new route to the destination (22.9-25.1%). However, if the event does not prevent travelling to the next checkpoint, then the difference drops, reaching 15.8% (Table 1).

Table 1. Results for an infrastructure with 10 nodes/15 edges

Network with 10 nodes	**Location of the incident**	
	Same	**Future**
Number of cases	607	393
BEFORE	14.7%	
BACK	25.1%	22.9%
FORWARD	N.A.	15.8%

The differences in this case are considerably reduced, they amount to 7.8% for the generation of a new optimal solution before starting the transport and to 8.9% if the event does not prevent the movement to the next checkpoint. The cost difference remains quite high for the situation in which the vehicle must turn back from the road to find a new route to the destination (14.2-18.8%) (Table 2).

Table 2. Results for an infrastructure with 27 nodes/41 edges

Network with 27 nodes	**Location of the incident**	
	Same	**Future**
Number of cases	381	619
BEFORE	7.8%	
BACK	18.8%	14.2%
FORWARD	N.A.	8.9%

In a network of 50 nodes, the generation of a new optimal solution after the occurrence of an event involves an additional cost of only 4.5%, and if the event occurs after the start of the transport but allows the vehicle to move to the next point of the route, the cost difference is only 5.1%. For the case when the vehicle must turn back from the road the difference remains higher, 6.8-8.6% more than the value for the optimal route (Table 3).

Table 3. Results for an infrastructure with 50 nodes/144 edges

Network with 50 nodes	**Location of the incident**	
	Same	**Future**
Number of cases	243	757
BEFORE	4.5%	
BACK	8.6%	6.8%
FORWARD	N.A.	5.1%

Further on, for sufficiently developed configurations of the road network, the differences become smaller and smaller (as it can be seen in Tables 4-7).

Table 4. Results for an infrastructure with 1000 nodes/3858 edges

Network with 1000 nodes	Location of the incident	
	Same	Future
Number of cases	38	962
BEFORE	0.38%	
BACK	0.9%	0.7%
FORWARD	N.A.	0.43%

Table 5. Results for an infrastructure with 5000 nodes/18754 edges

Network with 5000 nodes	Location of the incident	
	Same	Future
Number of cases	7	993
BEFORE	0.06%	
BACK	0.33%	0.13%
FORWARD	N.A.	0.08%

Table 6. Results for an infrastructure with 10000 nodes/38726 edges

Network with 10000 nodes	Location of the incident	
	Same	Future
Number of cases	4	996
BEFORE	0.03%	
BACK	0.28%	0.09%
FORWARD	N.A.	0.037%

Table 7. Results for an infrastructure with 15000 nodes/54684 edges

Network with 15000 nodes	Location of the incident	
	Same	Future
Number of cases	3	997
BEFORE	0.02%	
BACK	0.18%	0.06%
FORWARD	N.A.	0.024%

For road networks of over 1000 nodes (Tables 4-7), the supplementary cost drops below 1%. There is also a 1% limit for various non-critical events generating additional costs; going over this limit will trigger a re-validation of the minimal cost path (or change it, if possible/necessary).

5. Conclusions

This paper aims to evaluate and propose a strategy for (repeatedly) determining the minimal cost route in a transportation network with variable costs on different road segments and prone to critical incidents.

Based on an existing network and evaluated transport costs, the corresponding optimal solution for the existing road infrastructure was found by using Dijkstra's algorithm.

For testing the effectiveness of the proposed solution, different configurations were simulated. The obtained results confirm the efficiency of the proposed approach, as well as the fact that granularity of road infrastructure becomes important in obtaining a near-optimal transport cost after a critical incident.

The novelty of this paper lies in the possibility to easily generate a new optimal solution in real time when segments from the optimal path are eliminated.

Transferring the obtained simulation results to the existing supplier-customer distribution systems in different application domains is a possible perspective for future work, for example in communication systems, military transports or medical supply networks.

REFERENCES

Adjiashvili, D., Oriolo, G. & Senatore, M. (2013) The Online Replacement Path Problem. In: Bodlaender, H. L. & Italiano, G. F. (eds.) *Algorithms - ESA 2013. Lecture Notes in Computer Science, vol. 8125*. Berlin, Heidelberg, Springer, pp. 1-12.

Amaliah, B., Fatichah, C. & Suryani, E. (2022) A Supply Selection Method for better Feasible Solution of balanced transportation problem. *Expert Systems with Applications*. 203, 117399. doi: 10.1016/j.eswa.2022.117399.

Dantzig, G. B. & Ramser, J. H. (1959) The Truck Dispatching Problem. *Management Science*. 6(1), 80-91. doi: 10.1287/mnsc.6.1.80.

Fu, L., Sun, D. & Rilett, L. R. (2005) Heuristic shortest path algorithms for transportation applications: State of the art. *Computers & Operations Research*. 33(11), 3324-3343, doi: 10.1016/j.cor.2005.03.027.

Huang, C.-Y., Lai, C.-Y. & Cheng, K.-T. (2009) Fundamentals of Algorithms. In: Wang, Laung-Terng, Chang, Y.-W. & Cheng, K.-T. (eds.) *Electronic Design Automation*. Amsterdam, Holland, Elsevier, pp. 173-234.

Kare, A. S. (2016) A Simple Algorithm For Replacement Paths Problem. *Electronic Notes in Discrete Mathematics*. 53, 307-318, doi: 10.1016/j.endm.2016.05.026.

Ortega-Arranz, H., Llanos, D. R., Gonzalez-Escribano, A. (2015) *The Shortest-Path Problem*. Cham, Springer.

This is an open access article distributed under the terms and conditions of the
Creative Commons Attribution-NonCommercial 4.0 International License.