WS Network Design Problem with Nonlinear Pricing Solved by Hybrid Algorithm

Dušan Hrabec[1(✉)], Pavel Popela[2], and Jan Roupec[2]

[1] Faculty of Applied Informatics, Tomas Bata University, Nad Stráněmi 4511, 760 05 Zlín, Czech Republic
hrabec@fai.utb.cz

[2] Faculty of Mechanical Engineering, Brno University of Technology, Technická 2896/2, 616 69 Brno, Czech Republic
{popela,roupec}@fme.vutbr.cz

Abstract. The aim of the paper is to introduce a wait-and-see (WS) reformulation of the transportation network design problem with stochastic price-dependent demand. The demand is defined by hyperbolic dependency and its parameters are modeled by random variables. Then, a WS reformulation of the mixed integer nonlinear program (MINLP) is proposed. The obtained separable scenario-based model can be repeatedly solved as a finite set of MINLPs by means of integer programming techniques or some heuristics. However, the authors combine a traditional optimization algorithm and a suitable genetic algorithm to obtain a hybrid algorithm that is modified for the WS case. The implementation of this hybrid algorithm and test results, illustrated with figures, are also discussed in the paper.

Keywords: Stochastic transportation model · Network-design problem · Nonlinear pricing · Wait-and-see approach · Genetic algorithm · Hybrid algorithm

1 Introduction

The transportation network design problem (TNDP) remains a challenging research topic in transportation planning. From constructing new roads, pipelines, power lines, etc. to determining the optimal road toll, TNDP has provided valuable information for capital investment in transportation [1,7,18]. Various approaches have been used to solve TNDP. Steenbrink [17] and Magnanti and Wong [8] reviewed a number of the network design problems (NDP's) and some earlier algorithms. LeBlanc [7] proposed a branch-and-bound procedure to solve the problem but the algorithm did not perform well in large-scale problems. For a detailed review of solution techniques see, e.g., [1,11].

This paper presents a hybrid algorithm for the solution of a scenario-based wait-and-see (WS) stochastic mixed integer nonlinear program (MINLP), which models the design of a transportation network under price-sensitive stochastic demand. Regarding the solution technique, we mention our direct approach

J. Handl et al. (Eds.): PPSN XIV 2016, LNCS 9921, pp. 655–664, 2016.
DOI: 10.1007/978-3-319-45823-6_61

derived from modeling ideas (e.g., [13]). Due to the growing popularity of pricing strategies development and further applications in industry, we follow up on our previous modeling ideas presented in [4], where we modeled a mixed integer linear program with linearly price-dependent stochastic demand. So, we extend our previous model from [4] into a more complex case with a nonlinear (hyperbolic) price-demand dependency and, therefore, we also modify the previously used algorithm [4,13].

2 Stochastic TNDP with Pricing Solved by WS Approach

In this section, we develop the above mentioned MINLP which represents the design of a transportation network under price-sensitive stochastic demand. Note, that in our case, the network consists of three components: supply, demand, and transition parts of the system, see [2]. Before we deal with the stochastic problem and its WS reformulation, we shortly review the hyperbolic pricing function [10].

2.1 Pricing

Consider a price-setting firm that faces a price-dependent demand function, $b_i(p_i)$, describing the dependency between price p_i and demand b_i for each customer denoted by i. To capture real-world situations, we will further define the demand function as $b_i(p_i) = \alpha_i p_i^{-\beta_i}$, where $\alpha_i > 0$ and $\beta_i > 1$, see Fig. 1.

This means that the selling prices are decision variables, and so we want to find the optimal price p_i^* for each customer i.

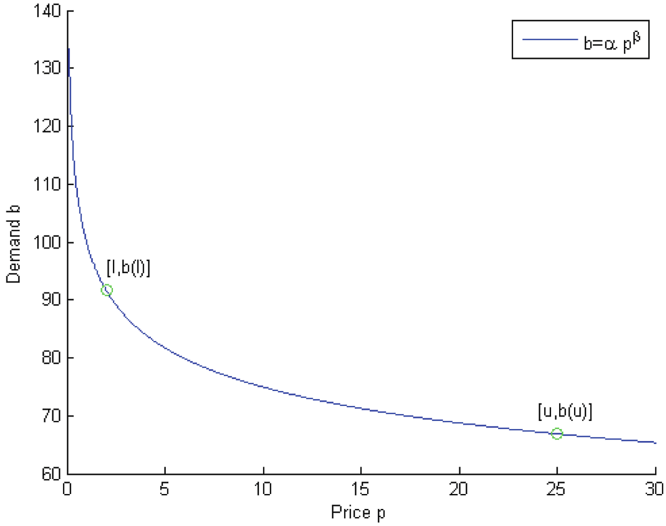

Fig. 1. Example of a hyperbolic demand-pricing function.

2.2 Stochastic Demand and the WS Approach

In real-world problems, the customer demand information is often uncertain and varying. This situation is usually modeled by one of the following deterministic reformulations: (a) the here-and-now (HN) approach, which means that the decisions are made before the demand is observed, see [15] and, specifically, [13]; (b) the wait-and-see (WS) approach, which means that the demand is known at the decision point. An interested reader can also find useful references to fundamental concepts of stochastic programming, e.g., in [6,15].

In this paper, we approach the stochastic TNDP with pricing using the WS scenario-based approach. The scenario-based approach assumes that we have enough observations of the parameters $\alpha_{i,s}$ and $\beta_{i,s}$ (one combination of the observations represents one particular scenario for each customer). In order to develop the mathematical model, we define the following (decision) variables, index sets and parameters.

- The decision variables:
  $x_{e,s}$: amount of the product to be transported on edge e in scenario s,
  $\delta_{e_n,s} \in \{0,1\}$: 1 if new edge e_n is built in scenario s, 0 otherwise,
  $p_{i,s}$: unit selling price for customer i in scenario s,
- second-stage variables:
  $y^+_{i,s}$: shortages for customer i in scenario s,
  $y^-_{i,s}$: leftovers for customer i in scenario s,
- index sets:
  E : set of edges, $e \in E$,
  E_n : set of new (built) edges, $e_n \in E_n,\ E_n \subset E$,
  i : set of customers (or locations with a non-zero demand), $i \in I$,
  j : set of production locations (or warehouses), $j \in J$,
  k : set of traffic nodes, $k \in K$,
  V : set of all nodes (vertices) in the network, $v \in V$,$V = I \cup J \cup K$,
  S : set of all possible scenarios, $s \in S$, $s = 1, 2, \ldots, m$,
- and parameters:

  $A_{v,e}$: incidence matrix, $A_{v,e} \begin{cases} 1 & \text{if edge } e \text{ leads to node } v, \\ -1 & \text{if edge } e \text{ leads from node } v, \\ 0 & \text{otherwise,} \end{cases}$
  $b_{v,s}$: the demand in node v for scenario s,
  c_e : unit transporting cost on edge e,
  d_{e_n} : cost of building of a new edge e_n,
  r^+_i, r^-_i : unit penalty cost for shortages/leftovers at customer node i,
  l, u : lower and upper bound for selling prices,
  $\alpha_{i,s}, \beta_{i,s}$: scenario-based (and demand-related) parameters.

Then, we formulate the stochastic TNDP with nonlinear pricing, which we reformulate using WS approach, and so, we solve the model repeatedly, i.e., once for each scenario:

$\forall s \in S:$

$$\max \sum_{i\in I}(\sum_{e\in E} A_{i,e}x_{e,s})p_{i,s} - \sum_{e\in E} c_e x_{e,s} - \sum_{e_n\in E_n} d_{e_n}\delta_{e_n,s} - \sum_{i\in I}(r_i^- y_{i,s}^- + r_i^+ y_{i,s}^+) \quad (1)$$

$$\sum_{e\in E} A_{i,e}x_{e,s} = b_{i,s} - y_{i,s}^+ + y_{i,s}^-, \ \forall i \in I, \quad (2)$$

$$\sum_{e\in E} A_{j,e}x_{e,s} = b_{j,s}, \quad \forall j \in J, \quad (3)$$

$$\sum_{e\in E} A_{k,e}x_{e,s} = b_{k,s}, \quad \forall k \in K, \quad (4)$$

$$x_{e_n,s} \le \delta_{e_n,s}\sum_{j\in J}(-b_j), \quad \forall e_n \in E_n, \quad (5)$$

$$y_{i,s}^+ \le b_{i,s}, \quad \forall i \in I, \quad (6)$$

$$x_{e,s} \ge 0, \quad \forall e \in E, \quad (7)$$

$$\delta_{e_n,s} \in \{0,1\}, \quad \forall e_n \in E_n, \quad (8)$$

$$y_{i,s}^+,\ y_{i,s}^- \ge 0, \quad \forall i \in I, \quad (9)$$

$$p_{i,s} \ge l, \quad \forall i \in I, \quad (10)$$

$$p_{i,s} \le u, \quad \forall i \in I, \quad (11)$$

$$b_{i,s} = \alpha_{i,s}p_{i,s}^{-\beta_{i,s}}, \quad \forall i \in I. \quad (12)$$

The objective function (1) maximizes the total profit, which is the revenue minus all the costs (transportation, network design and penalties for leftovers and shortages). Equations (2–4) are balance constraints, i.e. amount entering a node is equal to the demand plus the amount leaving; in addition, in the constraint (2) we consider quantities presenting leftovers and shortages, respectively. (5) guarantees that there will be no transported amount on non-built edges. (6) is a constraint on shortages, i.e. any shortage can not be higher than related demand. (7)–(11) state domains of decision variables, while Eq. (12) states the hyperbolic dependency between price and demand (see Fig. 1).

Obviously, the problem (1)–(12) is nonlinear, but it seems that the exact solvers deal with a linearized (MILP) version of it. Such nonlinear problems often requires a heuristic approach, especially large scale problems. Therefore, we further propose a hybrid algorithm in Sects. 3 and 4.

3 Hybrid Algorithm for the WS Approach

The above-mentioned model was coded in GAMS and solved by the BARON, MINOS and CPLEX solvers for suitable test instances. The obtained results are considered acceptable. The next solution attempt targeted large test problems using the same techniques; however, this led to an increase of the required computational time.

Due to the above, the decision to utilize previous experience was made, see [3,13]. This resulted in the implementation of a modified hybrid algorithm combining the GAMS code with a selected genetic algorithm (GA). The C++ implementation concentrating on the GAMS-GA interface is developed for the updated GA, as it was discussed in [12]. This can also be replaced by other GAs [9]. The principles of the following algorithmic scheme follow the papers [3,13].

1. Initialize the computer environment for parallel computations.
2. Define the scenario-based GAMS model and load the model and data into *.gms files for each scenario. Specify control parameters for the GA so that one instance is created for each scenario. The parameters can be defined either by the user (e.g., the population size) or inherited from the GAMS code (e.g., how many edges in the network should be taken into account).
3. Build an initial population for each GA instance. Specifically, the initial values of 0–1 variables must be generated and copied in the `$INCLUDE` files, from which they are read by the GAMS code.
4. The GAMS model is repeatedly solved (in parallel, two loops, one for scenarios and one by population size) by using the MINOS solver. Each run solves the program for the fixed values of 0–1 variables. The profit (or, alternatively, cost) function values are computed (initially in 3. and then in 8.).
5. The best results obtained from GAMS in 4. are saved for comparisons.
6. The termination conditions for the algorithm are tested (in parallel) and the algorithm is terminated if they are met. Otherwise the algorithm proceeds until the last scenario solution is obtained.
7. Input values for the GA from GAMS results are generated, see step 4. Specifically, the profit function values for each member of population of the GA are received from results of the GAMS runs in 4.
8. The GA run leads to an update of the set of 0–1 variables (population), see [12] for details.

Broadly speaking, the GA works with 0–1 variable $\delta_{e_n,s}$ for each scenario s, while MINOS solves the remaining nonlinear problem (NLP) for the fixed binary variable δ, i.e. MINOS computes optimal $x_{e,s}$, $p_{i,s}$ as well as value of the objective function. Afterwards, the value of objective/fitness function (1) is sent back for the solution assessment and then, according to 6., the algorithm continues.

4 Description of the Utilized Genetic Algorithm

This section shortly reviews key ideas of the utilized GA that works as the main part of the hybrid algorithm, see Sect. 5. It follows the previous ideas of one of the authors [12]; see also [13] for its extension.

In general, we consider a set of genetic operators containing: the crossover operator, the mutation operator, and eventually other problem dependent or implementation dependent operators. All these operators generate descendants from parents. The parent selection operator and the genetic operators have a probabilistic character and the deletion operator is usually deterministic. The fitness value f is a non-negative number which captures a relative measure of the quality of every individual in the current population. The run of our GA can be described using the following steps: (1) Generation of the initial population (random generation is often used) composed of individuals. (2) Computation of fitness function values related to 1). (3) Parent selection and generation of offspring. (4) Creation of the new population by using deletion operator and addition of offspring generated in the previous step. (5) Mutation. (6) If the stopping rule is not satisfied, go to step 3), otherwise continue to 7). (7) The result is the best individual in the population. It is usually advantageous to use some redundancy in genes, and then the physical length of the genes can be greater than one bit. Such a type of redundancy by shades was introduced by Ryan [14]. To prevent degeneration and the deadlock in a local extreme, a limited lifetime of individuals can be used. This limited lifetime is implemented via a death operator [12], which represents something like a continual restart of the GA. Many GAs are implemented on a population consisting of haploid individuals (each individual contains one chromosome). However, in nature, many living organisms have more than one chromosome and there are mechanisms used to determine dominant genes. Sexual recombination generates an endless variety of genotype combinations that increases the evolutionary potential of the population. Since it increases the variation among the offspring produced by an individual, this improves the probability that some of them will be successful in varying and often unpredictable environments. The modeling of sexual reproduction is quite simple. The population is divided into two parts - males and females. One parent from each part is selected for crossover. The sex of the individual is stored in the special gene; this gene is not mutated. The sex of the descendant is determined by a crossover of the sexual genes of parents, the descendant is placed into the corresponding part of population. The replacement scheme is associated with another problem. To ensure monotonous behavior the incremental replacement (steady-state replacement) was introduced. We can use least-fit member replacement where one (or more) elements with the worst fitness is replaced, or we can replace randomly chosen element(s). Therefore, the elitism brings a way to keep monotony while generational replacement is used. One or several best individuals represent the elite. The whole elite is directly taken into the next iteration.

So, the GA used in the paper for problem related computations uses ranking selection, haploid chromosomes, shadows and limited lifetime, as described above. We used uniform crossover and the probability of mutation of every gene was 5 %. Every 01 variable was stored in one gene having length of 3 bits. This redundant coding uses the shades technique mentioned above. The population size was 20 individuals; such a low value was chosen in relation to the computational complexity of evaluation of the fitness. The maximum number of iterations was limited to 50. The maximum lifetime of individual was set to 5 iterations.

5 Computations and Results

Figure 2 represents an initial visualization of an example. The example shows a distribution network: bold lines are existing edges and dash lines are possible

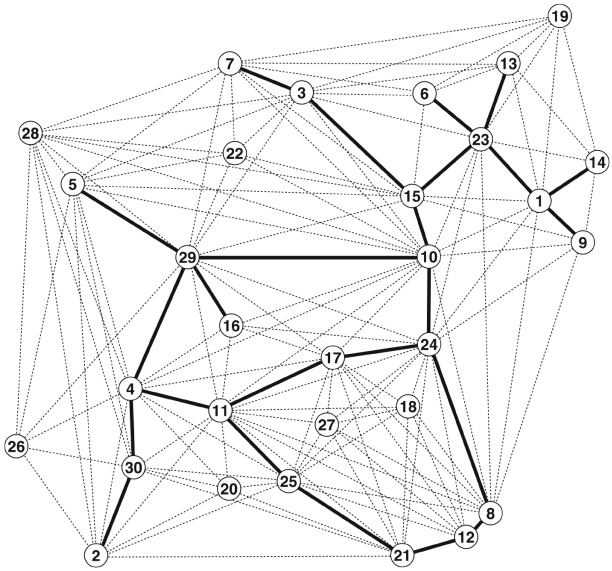

Fig. 2. Input network structure for the WS case [4].

edges that can be switched on by 0–1 variables, nodes 1–14 present customers, 15,16 production nodes, 17–30 transition nodes.

The main idea of the hybrid algorithm is based on the solution of a stochastic program for various sequences of the fixed $0-1$ variables repeatedly for each scenario. This extends the idea of [13] with modifications of the hybrid algorithm in Sect. 3. So, the optimal objective function values are obtained together with these sequences of zeros and ones. They serve as the input fitness value plus elements of the populations for the GA instances that utilizes its own above mentioned steps that are hidden within the GA structure. Updated sequences of zeros and ones are generated by the GA and sent to the GAMS through the updated `$INCLUDE` file and the computational loop continues until a satisfactory improvement of the network design is obtained. For the purpose of future comparison, we have utilized the test examples from [4]. The comparison between MINOS and of the proposed hybrid solution will be subject of our future research, but we have already shown on other MINLP problems that usage of exact solvers is not applicable in real (large) problems due to a huge computational time [4]. Therefore, using of the hybrid approach has one more reason in the MINLP's.

Results are described in Fig. 3 where the thicknesses of lines represent frequencies of usage in m scenarios, and hence, probabilities that variables x_e related to edges are non-zeros. The fixed lines are drawn as dash lines to emphasize the role of edges generated by the WS computations. We may also see that the stochastic demand usually requires new edges to bring the necessary adaptation in the results. In comparison with the HN solutions (cf. [13]) it can be done in a more flexible and cheaper way. Figure 3 also shows that only

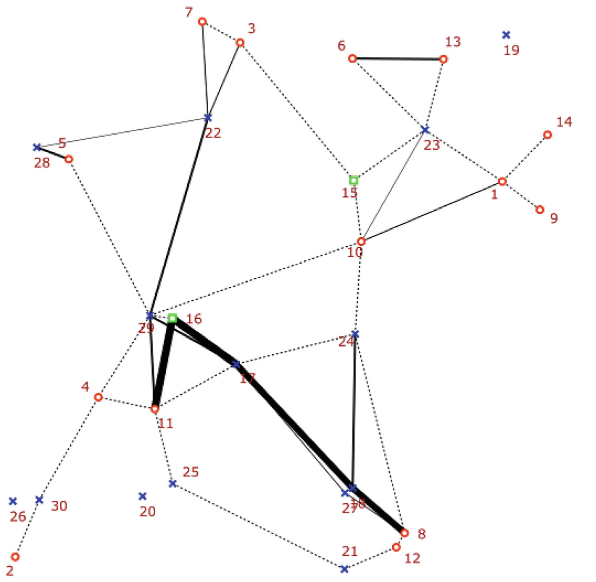

Fig. 3. Visualization of results for the hybrid algorithm for 100 scenarios.

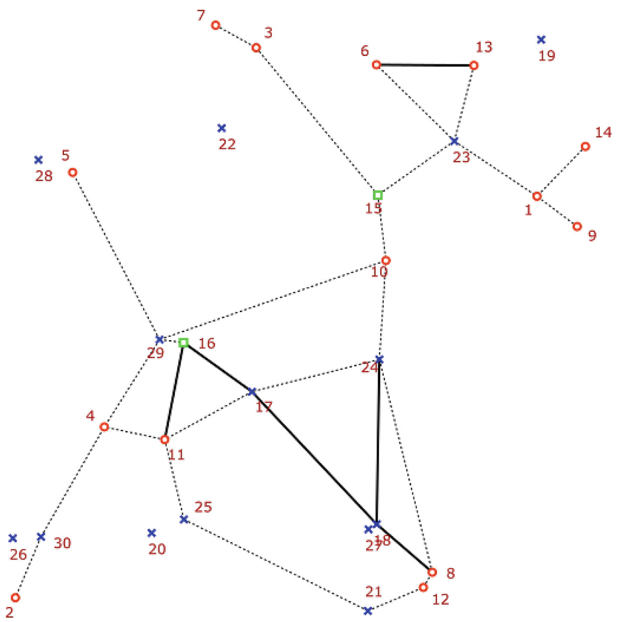

Fig. 4. Visualization of results from GAMS for 1 scenario.

suboptimality has been reached by computations for some scenarios, as extra unnecessary edges are switched on by the GA runs (e.g., 5–28).

To compare the obtained results, due to extreme time requirements of finding a traditional GAMS MINLP solution, we utilized one scenario case and provide a visualization of the result in Fig. 4. We leave further comparison of time requirements as well as values of objective functions for our further research.

6 Conclusions and Further Research

The paper presents a WS reformulation of a TNDP with stochastic price-dependent demands. The proposed mixed-integer nonlinear model is solved with the original hybrid algorithm involving GA for the solution of the WS network design problem. The previously introduced hybrid algorithm (see [4,13]) has been modified and successfully tested. This reconfirms our conclusions in [13] about the portability of the approach to other problems.

In our further research work, we plan to compare (or improve) the proposed hybrid algorithm with similar ideas dealing with differential evolution, specifically multi-chaotic success-history based parameter adaptation for differential evolution [5], which is a novel version of the standard GA that, hopefully, may achieve better computational results for our MINLP problems. Moreover, some obvious suboptimalities (see, e.g., Fig. 3) produced by the GA can easily be eliminated by appending a local search procedure to the GA run.

Similar mixed integer (nonlinear) stochastic programs may appear in many application areas, including NDP [11], traffic networks [3] or waste management problems [16]. Therefore, the suggested hybrid algorithm can be modified and widely applied.

Acknowledgments. This work was supported by the EEA Programme and Norway Grants, within the institutional cooperation project Nr. NF-CZ07-ICP-4-345-2016.

References

1. Babazadeh, A., Poorzahedy, H., Nikoosokhan, S.: Application of particle swarm optimization to transportation network design problem. J. King Saud Univ. Sci. **23**, 293–300 (2011)
2. Ghiani, G., Laporte, G., Musmanno, R.: Introduction To Logistic Systems Planning And Control. Wiley: Interscience Series in Systems and Optimization. Wiley, Chichester (2004)
3. Holešovský, J., Popela, P., Roupec, J.: On a disruption in congested networks. In: Proceedings of the 18th International Conference of Soft Computing MENDEL, pp. 191–196, Brno, Czech Republic (2013)
4. Hrabec, D., Popela, P., Roupec, J., Jindra, P., Novotný, J.: Hybrid algorithm for wait-and-see transportation network design problem with linear pricing. In: Proceedings of 21st International Conference on Soft Computing, MENDEL 2015, pp. 183–188, Brno, Czech Republic (2015)

5. Hrabec, D., Viktorin, A., Šomplák, R., Pluháček,M., Popela, P.: A heuristic approach to facility location problemfor waste management: a case study. In: Proceedings of 22nd International Conference on Soft Computing, MENDEL 2016, pp. 61–66, Brno, Czech Republic (2016)
6. Kall, P., Wallace, S.W.: Stochastic Programming, 2nd edn. Wiley, Chichester (1994)
7. LeBlanc, L.J.: An algorithm for discrete network design problem. Transp. Sci. **9**, 183–199 (1975)
8. Magnanti, T.L., Wong, R.T.: Network design and transportation planning: models and algorithms. Transp. Sci. **18**, 1–55 (1984)
9. Matoušek, R.: HC12: the principle of CUDA implementation. In: Proceedings of the 16th International Conference on Soft Computing MENDEL 2010, pp. 303–308, Brno, Czech Republic (2010)
10. Petruzzi, N.C., Dada, M.: Pricing and the newsvendor problem: a review with extensions. Oper. Res. **47**(2), 183–194 (1999)
11. Poorzahedy, H., Rouhani, O.M.: Hybrid meta-heuristic algorithms for solving network design problem. Eur. J. Oper. Res. **182**, 578–596 (2007)
12. Roupec, J.: Advanced genetic algorithms for engineering design problems. Eng. Mech. **17**(5–6), 407–417 (2011)
13. Roupec, J., Popela, P., Hrabec, D., Novotný, J., Olstad, A., Haugen, K.K.: Hybrid algorithm for network design problem with uncertain demands. In: Proceedings of the World Congress on Engineering and Computer Science, WCECS 2013, vol. 1, pp. 554–559, San Francisco, USA (2013)
14. Ryan, C.S.: Polygenic inheritance scheme. In: Proceedings of the 3rd International Conference on Soft Computing MENDEL 1997, pp. 140–147, Brno, Czech Republic (1997)
15. Shapiro, A., Dentcheva, D., Ruszczynski, A.: Lectures on Stochastic Programming: Modeling and Theory. SIAM, Philadelphia (2009)
16. Šomplák, R., Procházka, V., Pavlas, M., Popela, P.: The logistic model for decision making in waste management. Chem. Eng. Trans. **35**, 817–822 (2013)
17. Steenbrink, P.A.: Optimization of Transport Network. Wiley, New York (1974)
18. Tiratanapakhom, T., Kim, H., Nam, D., Lim, Y.: Braess' Paradox in the uncertain demand and congestion assumed stochastic transportation network design problem. KSCE J. Civil Eng. 1–10 (2016)