Journal of Naval Sciences and Engineering
2021, Vol. 17, No. 2, pp. 241-263
Industrial Engineering/Endüstri Mühendisliği

RESEARCH ARTICLE

**An ethical committee approval and/or legal/special permission has not been required within the scope of this study.*

A LAGRANGEAN RELAXATION-BASED SOLUTION APPROACH FOR MULTICOMMODITY NETWORK DESIGN PROBLEM WITH CAPACITY VIOLATIONS*

Levent ERİŞKİN[1]

[1]National Defence University, Turkish Naval Academy, Department of Industrial Engineering, Istanbul, Turkey,
leriskin@dho.edu.tr; ORCID: 0000-0002-9128-2167

Received: 04.05.2021 **Accepted: 25.06.2021**

Levent ERİŞKİN

ABSTRACT

In this study, we formulate and compare two different Lagrangean relaxation-based decompositions for multicommodity network problems with penalized constraints. These problems are different versions of capacitated multicommodity network problems where capacity constraints can be violated for additional penalty costs. These costs are reflected as nonlinear terms in the objective function; hence, these problems turn out to be nonlinear mixed-integer optimization problems. To the best of our knowledge, there is no exact solution algorithm for this type of problem. We propose two kinds of Lagrangean relaxation-based decompositions and solve these problems with the subgradient algorithm. The resulting subproblems are easy to solve and the proposed algorithms can reach reasonable solutions where CPLEX solver cannot even find a solution. In the study, we also conduct a computational analysis where we compare two relaxations over various performance measures. Even though two relaxations present similar performances in terms of computation times and the number of iterations, we observed that Relaxation 1 statistically outperforms Relaxation 2.

Keywords: *Multicommodity Network Design Problem, Lagrangean Relaxation, Subgradient Algorithm, Decomposition.*

KAPASİTE İHLALLİ ÇOKLU MAL ŞEBEKE DİZAYN PROBLEMİ İÇİN LAGRANGEAN GEVŞETMESİ TABANLI BİR ÇÖZÜM YAKLAŞIMI

ÖZ

Bu çalışmada, cezalandırıcı kısıtlara sahip çoklu mal şebeke problemi için Lagrangean gevşetmesi tabanlı iki farklı ayrıştırma yaklaşımı formüle edilmekte ve karşılaştırılmaktadır. Bu problemler kapasite kısıtlarının ilave bir ceza maliyeti ile ihlal edilebilediği kapasite kısıtlı çoklu mal şebeke problemlerinin farklı versiyonlarıdır. Bu maliyetler amaç fonksiyonuna doğrusal olmayan terimler olarak yansıtılmakta, bu kapsamda bu problemler doğrusal olmayan karışık tam sayılı eniyileme problemlerine dönüşmektedir. Bilgimiz dahilinde, bu tip problemlerin çözümü için herhangi bir kesin çözüm algoritması bulunmamaktadır. Bu problemler için iki farklı Lagrangean gevşetmesi tabanlı ayrıştırma teklif etmekte ve gradyant altı algoritması ile çözmekteyiz. Ortaya çıkan alt-problemler kolaylıkla çözülebilmekte ve önerilen algoritmalar CPLEX çözücünün herhangi bir çözüm bile bulamadığı durumlar için makul sonuçlar elde etmektedir. Çalışmada ayrıca bu iki gevşetmenin farklı performans metrikleri bazında karşılaştırmasının yapıldığı bir hesaplamalı analiz de yapmaktayız. Her ne kadar iki gevşetme de çözüm süresi ve iterasyon adedi açısından benzer performanslar gösterse de Gevşetme 1'nin istatistiksel olarak Gevşetme 2'den daha üstün olduğunu gözlemledik.

Anahtar Kelimeler: *Çoklu Mal Şebeke Dizayn Problemi, Lagrangean Gevşetmesi, Gradyan Altı Algoritması, Ayrıştırma.*

1. INTRODUCTION

Multicommodity network flow (MCNF) problems are used extensively in operations research or management applications such as production scheduling and planning, transportation, and routing where more than one commodity is to be shipped over a network from a designated origin node to a destination node. There are mainly three types of MCNF problems: max MCNF problem, the max-concurrent flow problem, and min-cost MNCF problem (Wang, 2018a). In max MCNF problem sum of all flows of commodities is aimed to be maximized. Max-concurrent flow problem maximizes the percentage of satisfied demands of all commodities. Min-cost MCNF, on the other hand, aims to satisfy all demands for all commodities by finding a feasible assignment of flows to arcs. For this problem type, the major variant is incapacitated MCNF where there are no capacity limits enforced for arcs.

Even though min-cost MCNF models arise in different forms, two of them are mostly seen. These are network routing and network design problems (Wang, 2018a). Network routing problems seek a feasible assignment of flows to arcs for all commodities with a minimum cost without violating the capacity constraints of arcs. Network routing problems are usually seen in telecommunication and warehouse management applications (Yousefi Nejad Attari et al., 2020). In network design problems, we design a network on a given graph by determining which arcs to include in the network and the amount of flow on a given arc by satisfying the demands of all commodities. In capacitated version, the capacities of arcs cannot be violated. Network design problems have numerous applications in transportation, postal services, and telecommunication (Ghaffarinasab et al., 2020).

In this study, we consider a min-cost multicommodity network design problem where arc capacities can be violated for a penalty cost. The penalty cost for excess flow on an arc is reflected in the objective function on quadratic form; hence, the problem turns out to be a nonlinear mixed-integer multicommodity network flow problem. As Bektaş et al. (2010) remark, there is no exact solution method for this type of problem. In this respect, we consider two types of Lagrangean-based decompositions, one of which was proposed by Bektaş et al. (2010) as flow decomposition. These decompositions yield subproblems that can be solved efficiently compared to the original problem. To find solutions for these decompositions, we employ a subgradient algorithm. With this study, we aim to contribute to the MNCF literature by introducing various solution techniques.

The study is organized as follows: Section 2 provides a literature review for the problem. In section 3 we formulate the problem and present two different decompositions. Additionally, we give details of the algorithm based on subgradient optimization in this section. Section 4 gives details of an empirical study conducted for comparing two decompositions on a test set. Finally, we conclude in Section 5.

2. LITERATURE REVIEW

Being very popular among many scheduling, routing, and transportation applications, MNCF problems are well studied in the literature. In his paper, Wang (2018a) surveys the last three decades and provides a summary for applications and various mathematical formulations for MCNF problems. Focusing mainly on min-cost MCNF problems, he remarks that most of the MCNF problems are formulated as network routing and network design problems. In his follow-up paper, Wang (2018b) surveys MCNF solution methods that are proposed in the literature. He classifies solution methods as; primal and dual-based solution methods, approximation methods, interior-point methods, and convex programming methods.

Among solution approaches, Holmberg and Yuan (2000) propose a branch and bound algorithm for knapsack relaxation of the multicommodity capacitated network design problem. Utilizing a subgradient algorithm having additional features such as special penalty tests and cutting criteria, they show that they obtain optimal solutions in very short computation times with respect to commercial software packages. Regarding the Lagrangean-relaxation approach, Crainic et al. (2001) studied different relaxation techniques for large-scale capacitated MCNF problems. They propose two types of relaxations for the problem: by relaxing the capacity constraints they get a shortest path relaxation and by relaxing the network flow constraints they get knapsack relaxation. They utilize a bundle-based algorithm and subgradient method for solving them. They remark that both solution approaches perform well provided that the subgradient method is tuned properly. Costa (2005) focuses on applications of Benders decomposition to MCNF network design problems and presents a review of these applications. Katayama et al. (2009) propose a capacity scaling heuristic by utilizing a column generation and row generation technique for solving multicommodity capacitated network design problems. Combining row and column generation techniques, their proposed heuristic generates high-quality results based on computational experiments involving 196 problem instances. Alysson et al. (2009) compare three sets of inequalities that are used for strengthening the multicommodity capacitated network design problem formulation. They show that theoretical results apply to any network design problem for which feasible solutions are obtained by solving subproblems. Karsten et al. (2015) study a multicommodity network flow problem where a time constraint is imposed and apply it to a liner shipping network design case. The problem imposes time limits on the duration of the transit of the commodities through the network. They remark that ignoring time constraints results in significant differences in revenues compared to solving the same problem while these constraints are imposed. Considering that time constraints make the problem more complex, they propose an algorithm to reduce computation times. They show that the proposed technique solves the problem in reasonable times. Moradi et al. (2015) present a column generation algorithm for solving a bi-objective problem. Their approach is based on bi-objective simplex and Dantzig-Wolfe decomposition. They start the methodology by solving a single objective MCNF problem with Dantzig-Wolfe decomposition. Afterward, the algorithm moves from one non-dominated extreme point to another, as in simplex until there is no entering variable left. Gendron and Gouveia (2016) consider the piecewise linear multicommodity network design problem with an additional constraint enforcing that the total flow on each arc must be an integer. These types of problems are common in transportation and logistics because the total flow might be represented with vehicles or containers. They propose a formulation by using discretization which is commonly used in mixed-integer programming. They develop a Lagrangean relaxation solution approach and show that their approach is efficient and effective. Chouman et al. (2018) propose a novel branch-and-cut algorithm for solving multicommodity capacitated fixed charge network design problem. They incorporate several filtering methods to the algorithm that exploits the structure of the problem. Thus, they inhibit combinations of values of some variables. They show that filtering significantly improves the performance of the branch-and-cut algorithm. Oğuz et al. (2018) consider restricted continuous facility location problems where location of a facility can be anywhere on the planet except for in restricted regions. They model the problem as a MCNF problem and propose Benders decomposition algorithm to find the optimal solution to the model. They conduct computational experiments and show that the proposed method outperforms commercial solvers.

Among newer studies dealing with MNCF problems, Anisi and Fathabadi (2019) consider the survivable multicommodity network design with node capacities and flow restrictions. Being a variant of the multicommodity network design problem, these problems aim to minimize the cost of failure in addition to design cost. The design aims to ensure a feasible flow in case of a simultaneous failure on arcs. They utilize Benders decomposition to solve the problem as well as a new approach that considers particular failure scenarios. Guimaraes et al. (2020) studied a variant of the MNCF problem where multiple transport lines and time windows are considered. They proposed two mixed-integer programming models and two objective functions, in particular, minimization of network operational costs and minimization of travel times. Trivella et al. (2021) studied a generalization of MCNF where transit time restrictions are modeled as soft constraints and delays are penalized. Kazemzadeh et al. (2021) introduced node-based Lagrangian relaxation where the resulting subproblem decomposes by nodes.

There are not many studies that deal with nonlinear integer multicommodity network design problems. Crainic and Rousseau (1986) study a nonlinear mixed-integer multicommodity network flow problem. They present an algorithm combining heuristics and optimization. Belotti et al. (2007) consider a multicommodity network design problem with discrete node costs. Costs are defined as stepwise functions of facilities installed at these nodes. They propose a branch-and-cut algorithm for solving the problem. Bektaş et al. (2010) propose Lagrangean-based decomposition algorithms for multicommodity network design problems where arc capacities can be violated at the expense of a penalty. This penalty adds nonlinear cost to the min-cost objective function. They propose two decompositions: flow decomposition that is obtained by relaxing capacity constraints, and arc decomposition that is obtained by relaxing flow constraints. They show that with the help of a special algorithm developed to solve subproblems in arc decomposition, arc decomposition performs better in terms of convergence but worse in terms of computation time and the number of iterations. Paraskevopoulos et al. (2016) study a variant of fixed-charge multicommodity network design problem having additional congestion costs. They model the problem as a nonlinear integer programming model, and they propose two solution approaches. The first solution approach is the reformulation of the problem as a mixed-integer second-order cone program. The second uses an evolutionary algorithm combining iterated local search and scatter search. They remark that the first solution approach provides satisfactory results provided that conic representations of nonlinear terms are available. Additionally, they observe that the evolutionary algorithm is not only satisfactory but also achieves good quality solutions in short computational times.

For a recent survey regarding classification, applications, and solution methods of MNCF problems, the interested reader is referred to Salimifard and Bigharaz (2020).

3. PROBLEM FORMULATION AND OPTIMIZATION WITH LAGRANGEAN RELAXATION

We formulate the min-cost multicommodity network flow problem with capacity violations in harmony with the definition of Bektaş et al. (2010). We have a graph of $G = (N, A)$ where N corresponds to set of nodes and A corresponds to set of arcs. Two different sets $N_i^+ = \{j \in N | (i, j) \in A\}$ and $N_i^- = \{j \in N | (j, i) \in A\}$ are defined for each node. We have a set of commodities P. In this respect, we formulate the problem as follow:

Parameters:

B_{ij}: Upper bound on the amount of excess flow on each arc

f_{ij}: Fixed cost of activating a network

w^p: Quantity of commodity p that is to be sent from $o(p)$ to $d(p)$

$d_i^p = w^p$ if $i = o(p)$, $d_i^p = -w^p$ if $i = d(p)$

c_{ij}^p: The unit cost of routing the demand for commodity p over arc (i,j)

u_{ij}: Capacity of arc (i,j)

C_{ij}: Penalty cost for excess flow on arc (i,j)

Decision Variables:

x_{ij}^p: The amount of commodity p flowing on arc (i,j) where $x_{ij}^p \geq 0$

y_{ij}: Design variable for selecting arc (i,j) where $y_{ij} \in \{0,1\}$

e_{ij}: Excess flow on arc (i,j) where $e_{ij} \geq 0$.

The Model:

$$(F) \qquad Minimize \sum_{(i,j)\in A} f_{ij}\, y_{ij} + \sum_{(i,j)\in A}\sum_{p\in P} c_{ij}^p\, x_{ij}^p + \sum_{(i,j)\in A} C_{ij}(e_{ij})^2 \qquad \textbf{(1)}$$

$Subject\ to$

$$\sum_{j\in N_i{}^+} x_{ij}{}^p - \sum_{j\in N_i{}^-} x_{ji}{}^p = d_i{}^p \qquad \forall i \in N, p \in P \qquad \textbf{(2)}$$

$$x_{ij}^p \leq w^p y_{ij} \qquad \forall (i,j) \in A, p \in P \qquad \textbf{(3)}$$

$$\sum_{p\in P} x_{ij}{}^p \leq u_{ij} y_{ij} + e_{ij} \qquad \forall (i,j) \in A \qquad \textbf{(4)}$$

$$e_{ij} \leq B_{ij} y_{ij} \qquad \forall (i,j) \in A \qquad \textbf{(5)}$$

$$x_{ij}^{p} \geq 0,\ e_{ij} \geq 0, y_{ij} \in \{0,1\} \qquad \forall (i,j) \in A, p \in P \qquad \textbf{(6)}$$

The penalty term $(e_{ij})^2$ adds nonlinearity to the formulation. This term is quadratic in our formulation, however, the power of this term could be cubic or higher, as well (Bektaş et al., 2010). Constraint (2) is the flow conservation constraint, Constraint (3) ensures that flow of an arc is positive provided that it is selected, Constraint (4) enforces that total flow on an arc should be less than and equal to the sum of the capacity of that arc and excess flow on that arc, Constraint (5) imposes that maximum amount of excess flow on an arc cannot be more than a predefined value and can be positive unless it is selected.

This problem is a nonlinear mixed-integer problem and as Bektaş et al. (2010) remark, there exists no exact solution method proposed in the literature. In this respect, we define two different Lagrangean relaxations for the problem. Lagrangean relaxation aims to get rid of complicating constraints by adding them to the objective function by multiplying them with Lagrengean multipliers so that the resulting problem can be partitioned into small subproblems which can be solved relatively easily. The Lagrangean relaxation approach is classified as price-directive methods since Lagrangean multipliers place prices on the dualized constraints (Ahuja et al., 1993). In this respect, this technique aims to find proper prices so that an optimal solution to the Lagrangean subproblem provides a solution to the main problem.

3.1. Relaxation 1

This relaxation is proposed by Bektaş et al. (2010) and obtained by relaxing capacity constraints (Constraint sets (3) and (4)). By defining Lagrangean variables μ_{ij}^{p} and σ_{ij} for these constraint sets respectively, we formulate the relaxed problem as follows.

$$(LR1) \qquad Min \sum_{(i,j)\in A} [f_{ij} - \sum_{p\in P} \mu_{ij}^p w^p - \sigma_{ij} u_{ij}] y_{ij} + \sum_{(i,j)\in A} \sum_{p\in P} (c_{ij}^p + \mu_{ij}^p + \sigma_{ij})\, x_{ij}^p + \sum_{(i,j)\in A} \left[C_{ij}(e_{ij})^2 - \sigma_{ij} e_{ij}\right] \qquad \textbf{(7)}$$

$$Subject\ to \qquad (2), (5), (6) \qquad \textbf{(8)}$$

We can decompose $(LR1)$ into two subproblems. The first subproblem is defined over y and e variables. As shown by Bektaş et al. (2010), this problem can be solved by inspection.

$$(SP1) \qquad Min \sum_{(i,j)\in A} [f_{ij} - \sum_{p\in P} \mu_{ij}^p w^p - \sigma_{ij} u_{ij}] y_{ij} + C_{ij}(e_{ij})^2 - \sigma_{ij} e_{ij} \qquad \textbf{(9)}$$

$$Subject\ to \qquad (5) \qquad \textbf{(10)}$$

We define the second problem over x variables as shown below.

$$(SP2) \qquad Min \sum_{(i,j)\in A} \sum_{p\in P} (c_{ij}^p + \mu_{ij}^p + \sigma_{ij})\, x_{ij}^p \qquad \textbf{(11)}$$

$$Subject\ to \qquad (2) \qquad \textbf{(12)}$$

This problem can be decomposed into $|P|$ single commodity minimum cost network problems. We know that this problem is well-solved in the sense that an efficient algorithm is known. We can use the shortest path algorithm to solve each of these problems.

3.2. Relaxation 2

The second relaxation dualizes constraint sets (3), (4), and (5). We define nonnegative Lagrangean variables μ_{ij}^p, σ_{ij}, and γ_{ij} for these constraint sets respectively and formulate the relaxed problem as follows.

$$(LR2) \quad Min \sum_{(i,j)\in A} [f_{ij} - B_{ij}\gamma_{ij} - \sum_{p\in P} \mu_{ij}^p w^p - \sigma_{ij}u_{ij}]y_{ij} + \sum_{(i,j)\in A} \sum_{p\in P} (c_{ij}^p + \mu_{ij}^p + \sigma_{ij})\, x_{ij}^p + \sum_{(i,j)\in A} \left[C_{ij}(e_{ij})^2 - (\gamma_{ij} - \sigma_{ij})e_{ij}\right] \quad \textbf{(13)}$$

$$Subject\ to \qquad (2) \qquad \textbf{(14)}$$

This problem can be decomposed into 3 subproblems. The first problem is defined over x variables.

$$(SP1) \quad Min \sum_{(i,j)\in A} \sum_{p\in P} (c_{ij}^p + \mu_{ij}^p + \sigma_{ij})\, x_{ij}^p \quad \textbf{(15)}$$

$$Subject\ to \qquad (2) \qquad \textbf{(16)}$$

This subproblem decomposes into a set of minimum cost network flow problems for each commodity p. Therefore, we need to solve $|P|$ minimum cost network flow problems.

The second subproblem is defined over y variables and formulated as follows. This problem is an unconstrained binary optimization problem and can be solved by inspection easily.

$$(SP2) \quad Min \sum_{(i,j)\in A} [f_{ij} - B_{ij}\gamma_{ij} - \sum_{p\in P} \mu_{ij}^p w^p - \sigma_{ij}u_{ij}]y_{ij} \quad \textbf{(17)}$$

The last subproblem is defined over e variables.

$$(SP3) \quad Min \sum_{(i,j)\in A} \left[C_{ij}(e_{ij})^2 - (\gamma_{ij} - \sigma_{ij})e_{ij}\right] \quad \textbf{(18)}$$

This problem is a quadratic nonlinear programming problem. Since e_{ij} nonnegative, this is a convex function with a unique minimum. That is, the second derivative is positive and thus the solution to the first derivative gives the unique minimum. Therefore, this problem is easy to solve, too (Bektaş et al., 2010).

3.3. Optimization with Subgradient Algorithm

The subgradient algorithm is an easy and simple technique to solve non-differentiable Lagrangean multiplier problems. For a given set of multipliers, the relaxed problem provides a Lower Bound (LB) for the original problem. To obtain an Upper Bound (UB), problem F is solved while $(\mathbf{y}, \mathbf{q})$ is fixed to $(\mathbf{y}^*, \mathbf{q}^*)$ where these values correspond to the solution to the relaxed problem. In this case, the resulting problem turns out to be a linear programming problem. The subgradient algorithm is shown in Figure 1.

<u>Define:</u>

t: Number of iterations

s^t: Step size at each iteration

SP_1, SP_2, SP_3: Optimal objective values for $SP1, SP2, SP3$, respectively.

W_{LD}: Optimal objective value for the Lagrangean problem.

$\phi^t = \begin{pmatrix} \mu_{ij}{}^p \\ \sigma_{ij}{}^p \end{pmatrix}$: Vector of Lagrangean multipliers for Relaxation 1.

$\phi^t = \begin{pmatrix} \mu_{ij}{}^p \\ \sigma_{ij}{}^p \\ \gamma_{ij}{}^p \end{pmatrix}$: Vector of Lagrangean multipliers for Relaxation 2.

$g^t = \begin{pmatrix} g_{ijp}{}^1 \\ g_{ij}{}^2 \\ g_{ij}{}^3 \end{pmatrix}$: Vector of subgradients for $SP1, SP2, SP3$, respectively.

---

1. Initialize $\phi^t = \phi^0$
2. $LB = -\infty$ and $UB = \infty$
3. $t = 1$
4. While $gap = \frac{UB-LB}{UB} \geq \varepsilon$ do

4.1 Solve Lagrangean dual

4.1.1 Solve $SP1$, get $\bar{y}_{ij}$ and $\bar{e}_{ij}$ (Relaxation 1)
Solve $SP1$, get $\bar{x}_{ij}^{p}$ (Relaxation 2)

4.1.2 Solve $SP2$, get $\bar{x}_{ij}^{p}$ (Relaxation 1)
Solve $SP2$, get $\bar{y}_{ij}$ (Relaxation 2)

4.1.3 Solve $SP3$, get $\bar{e}_{ij}$ (Relaxation 2)

4.2 $W_{LD} = SP_1 + SP_2$ (Relaxation 1)
$W_{LD} = SP_1 + SP_2 + SP_3$ (Relaxation 2)

4.3 If $W_{LD} > LB$ then

$LB = W_{LD}$

4.4 Solve $\{F_r = F|\bar{y}_{ij}$ and $\bar{e}_{ij}$ values are fixed$\}$ and get ${F_r}^*, {x_{ij}}^{p*}$

4.5 $$UB = {F_r}^* - \sum_{(i,j)\in R} f_{ij} \quad \text{where } R = \{(i,j) \in A | {y_{ij}}^* = 1 \ and \sum_{p\in P} {x_{ij}}^{p*} = 0\}$$

4.6 Calculate subgradients

4.6.1 $${g_{ijp}}^1 = \bar{x}_{ij}^{p} - w^p \bar{y}_{ij} \quad \forall (i,j) \in A, p \in P$$
$${g_{ij}}^2 = \sum_{p\in P} x_{ij}^{p} - u_{ij}\bar{y}_{ij} - \bar{e}_{ij} \quad \forall (i,j) \in A$$
$${g_{ij}}^3 = \bar{e}_{ij} - B_{ij}\bar{y}_{ij} \quad \forall (i,j) \in A$$
$$g^t = \begin{pmatrix} {g_{ijp}}^1 \\ {g_{ij}}^2 \end{pmatrix} \text{ (Relaxation 1)}$$

$$g^t = \begin{pmatrix} {g_{ijp}}^1 \\ {g_{ij}}^2 \\ {g_{ij}}^3 \end{pmatrix} \text{(Relaxation 2)}$$

4.7 Calculate step length

$$s^t = \lambda \frac{UB - W_{LD}}{\|g^t\|^2}$$

4.8 Update Lagrangean multipliers

$$\phi^{t+1} = \phi^t + s^t g^t$$

4.9 $t = t + 1$

5. End while

Figure 1. Subgradient algorithm for the relaxations.

Simplicity of the subgradient algorithm has made it a popular option for solving Lagrangean multiplier problems. At each iteration, the algorithm takes a small step from the current point in the direction opposite to a subgradient. The most important parameter in the algorithm is the step length. One option is to use a constant step length. This option guarantees convergence; however, the convergence is too slow. In this respect, we employ a dynamic step length which provides faster convergence (Wolsey, 1998).

4. COMPUTATIONAL ANALYSIS

We have performed a computational analysis to compare the performances of these two relaxations. For this purpose, we used the first 36 instances defined in Crainic et al. (2001) which are also used by Bektaş et al. (2010). As done by Bektaş et al. (2010), we reduced capacities of arcs in the instances as $u'_{ij} = u_{ij}/2$ and set penalty costs C_{ij} to twice the flow cost of each arc. Subproblems are solved by using IBM ILOG CPLEX 12.5. Additionally, original problem is solved with IBM ILOG CPLEX 12.5 to evaluate lower bounds achieved. For both relaxations, subgradient algorithm is stopped whenever the gap does not improve for 30 consecutive iterations. The algorithm is coded with Java and run using an Intel Core i7 2.6 GHz. 8 GB RAM computer. Computational results are shown in Table 1.

Table 1. Computational results.

				Relaxation 1				**Relaxation 2**			
Inst.	N	A	P	i	t	g_s	g_0	i	t	g_s	g_0
1.1	10	35	10	128	6.90	0.67	0.67	154	5.10	0.24	0.24
1.2	10	35	10	29	3.14	2.63	0.62	50	3.34	1.80	1.01
1.3	10	35	10	14	1.83	16.61	6.43	22	1.89	9.34	4.99
1.4	10	35	10	18	1.87	29.19	-	18	1.78	28.81	-
1.5	10	35	10	81	3.84	35.86	31.56	94	4.37	33.86	32.22
1.6	10	35	10	95	5.52	27.31	-	92	3.90	27.24	-
2.1	10	35	25	25	3.57	53.78	48.02	22	2.98	19.90	18.48
2.2	10	35	25	102	5.56	71.63	-	102	5.56	32.93	-
2.3	10	35	25	64	6.54	67.94	65.30	62	4.31	41.91	40.53
2.4	10	35	25	33	4.30	72.78	54.32	43	3.125	45.55	45.16
2.5	10	35	25	52	5.38	70.66	52.72	37	3.57	50.01	49.48
2.6	10	35	25	33	4.04	60.20	42.06	117	7.20	40.46	39.94
3.1	10	35	50	25	4.71	70.92	48.45	33	6.45	37.56	35.02
3.2	10	35	50	32	5.29	63.56	31.45	100	10.09	32.98	32.04
3.3	10	35	50	52	8.57	69.43	-	77	6.187	39.29	-
3.4	10	35	50	32	5.59	75.27	52.97	31	3.39	56.93	53.56
3.5	10	35	50	48	5.76	67.75	47.43	85	5.62	45.67	44.53
3.6	10	35	50	52	6.52	50.27	-	149	11.25	36.99	-
4.1	10	60	10	44	4.33	6.99	3.50	67	6.00	11.33	0.66
4.2	10	60	10	27	3.44	19.61	10.17	23	2.10	12.39	0.98
4.3	10	60	10	132	5.92	9.25	14.45	178	6.60	8.49	8.49
4.4	10	60	10	45	4.53	4.87	0.32	46	3.26	8.36	1.07
4.5	10	60	10	150	6.63	8.81	7.45	34	2.73	12.83	0.47
4.6	10	60	10	17	2.56	38.86	12.78	70	3.85	32.25	13.10
5.1	10	60	25	41	5.64	28.01	17.39	37	4.2	20.32	12.61
5.2	10	60	25	115	13.2	20.38	7.73	34	3.73	20.29	3.11
5.3	10	60	25	70	7.41	50.30	10.85	59	5.14	15.54	10.24
5.4	10	60	25	45	4.32	74.49	-	45	4.32	22.79	-
5.5	10	60	25	44	5.69	67.46	-	59	5.73	49.68	-
5.6	10	60	25	54	6.2	60.84	-	48	3.406	58.12	-
6.1	10	60	50	34	7.12	67.38	23.22	88	12.39	27.15	22.87
6.2	10	60	50	62	11.8	63.52	-	44	7.03	35.69	-

-Continuation of the Table 1.

6.3	10	60	50	74	14.02	40.25	-	69	5.68	25.77	-
6.4	10	60	50	34	5.54	81.09	-	84	15.17	36.33	-
6.5	10	60	50	42	7.24	75.78	-	82	13.23	40.62	-
6.6	10	60	50	69	7.76	77.51	-	69	7.77	37.46	-

In Table 1, columns 2-4 correspond to the size of the instance in terms of the number of arcs, nodes, and commodities, respectively. Next two main columns present results for Relaxation 1 and 2. Under these columns, we provide the number of iterations performed for solution (i), computation time (CPU time) of the algorithm in seconds (t), gap value (in %) calculated based on UB and LB difference (g_s) and gap value (in %) calculated based on optimal value achieved by CPLEX solver and LB difference (g_0). The dashed lines under the last columns indicate those instances for which CPLEX cannot obtain optimal values. As clearly seen, CPLEX fails to find an optimal solution for 13 problems (36% of the problem set). As problem size increases, the performance of the CPLEX decreases, as expected. Table 1 indicates that computation times for both relaxations are less than a minute (maximum being 16 seconds) while we observe a slight increase as problems get more complex. To compare performances of these two relaxations visually, we plot performance measures vs. instances as shown in Figures 2-5. The gap g_0 is computed and graphed only for those instances that CPLEX solves optimally.

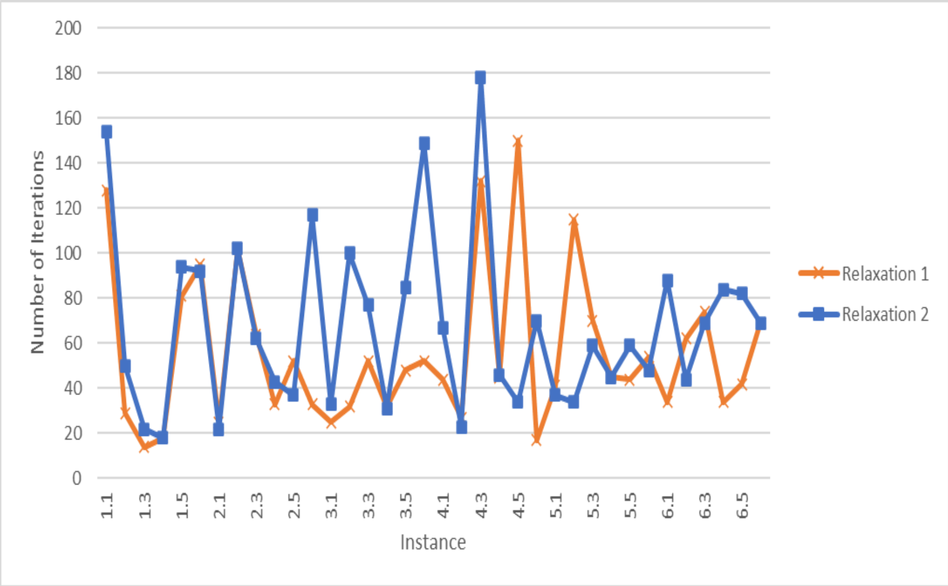

Figure 2. Number of iterations vs. instances.

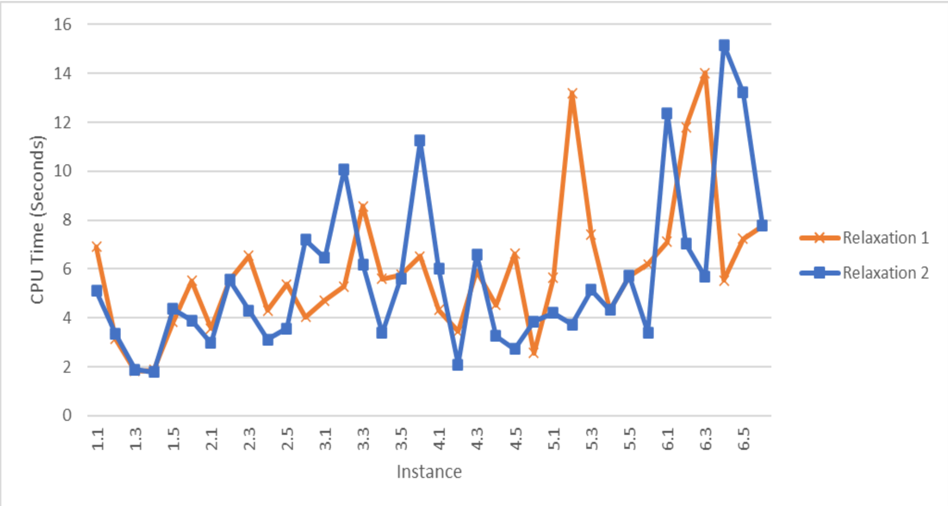

Figure 3. CPU times (in seconds) vs. instances.

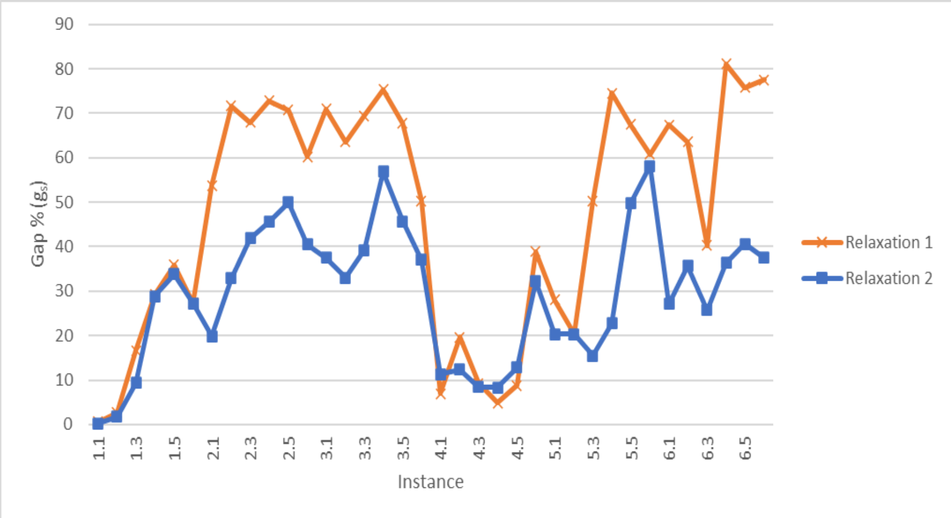

Figure 4. Gaps (g_s) vs. instances.

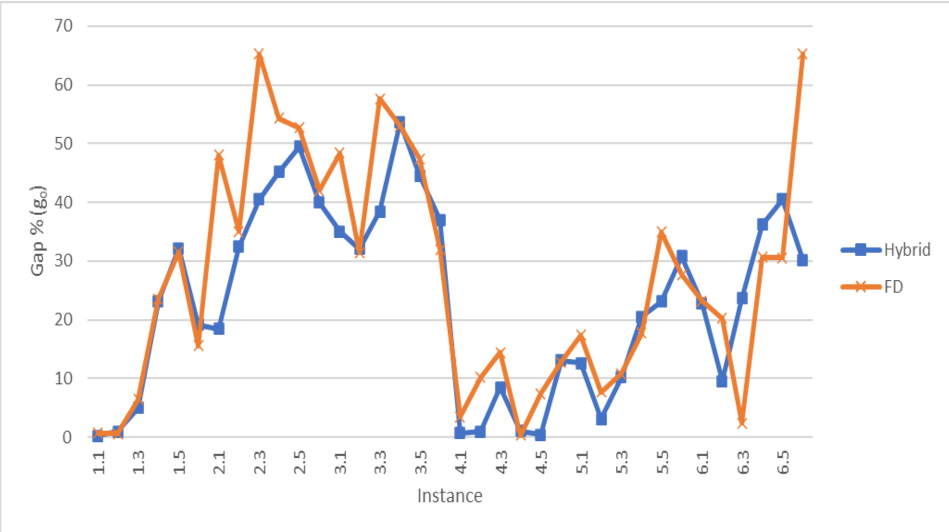

Figure 5. Gaps (g_0) vs. instances.

5. CONCLUSION

In this study, we consider a min-cost multicommodity network design problem where arc capacities can be violated for a penalty cost. When the penalty term is quadratic or has higher power, the problem turns out to be a nonlinear mixed-integer problem that does not have an exact solution method proposed in the literature. To solve this problem efficiently, we consider two different Lagrangean relaxations which decompose the original problem into smaller and easy to solve subproblems. One of these relaxations was proposed by Bektaş et al. (2010). We propose another relaxation and compare this relaxation with the former one. We implemented a computational study over a set of instances and solved these instances with the CPLEX solver and two relaxation approaches. Our computational study has shown that the CPLEX solver cannot obtain a solution for %36 of the test instances while two relaxation-based approaches achieve solutions with reasonable gaps. We also observe from the computational study that, our relaxation (Relaxation 2) outperforms that of Bektaş et al. (2010) (Relaxation 1) in terms of performance measures gap g_0 and gap g_s. Hence, this relaxation can be used for solving aforementioned problems which do not have exact solution methods and cannot be solved by on-the-shelf optimizers efficiently.

We used a subgradient algorithm to optimize the Lagrangean problem. One of the drawbacks of this algorithm is that it requires fine-tuning of parameters to achieve satisfactory results. Particularly, we observed that parameter λ that is used to calculate step length should be tuned carefully and values varying between 0.05 and 0.9 provide good results. Additionally, initial values of Lagrangean multipliers have a dramatic impact on the performance of the algorithm. Hence, as future work, a starting heuristic that will find suitable parameter settings would increase the performance of the algorithm.

As we stated previously, gap g_0 improves better than the gap g_s as problem size increases. This result indicates that we need to have a better procedure to generate UBs for the algorithm. Therefore, this could be another future work for this study.

REFERENCES

Ahuja, R. K., Magnanti, T. L. and Orlin, J. B. (1993). *Network Flows: Theory, Algorithms, and Applications,* New Jersey, Prentice Hall.

Anisi, M., and Fathabadi, H. S. (2019). "Survivable multi-commodity network flow design: case of node capacities and arc failure". *International Journal of Operational Research,* 35(3), 355–365. doi:10.1504/IJOR.2019.10022713.

Bektaş, T., Chouman, M. and Crainic, T. G. (2010). "Lagrangean-based decomposition algorithms for multicommodity network design problems with penalized constraints". *Networks,* 55(3), 171–180.

Belotti, P., Malucelli, F., and Brunetta, L. (2007). "Multicommodity network design with discrete node costs". *Networks,* 49(1), 90–99.

Chouman, M., Crainic, T. G., and Gendron, B. (2018). "The Iimpact of filtering in a branch-and-cut algorithm for multicommodity capacitated fixed charge network design". *EURO Journal on Computational Optimization,* Vol. 6, 143–184.

Costa, A. M. (2005). "A survey on benders decomposition applied to fixed-charge network design problems". *Computers & Operations Research,* 32(6), 1429–1450. doi:10.1016/j.cor.2003.11.012.

Costa, A. M., Cordeau, J.-F. and Gendron, B. (2009). "Benders, metric and cutset inequalities for multicommodity capacitated network design". *Computational Optimization and Applications,* Vol. 42, 371–392. doi:10.1007/s10589-007-9122-0.

Crainic, T. G., and Rousseau, J.-M. (1986). "Multicommodity, multimode freight transportation: A general modeling and algorithmic framework for the service network design problem". *Transportation Research Part B: Methodological,* 20(3), 225–242. doi:10.1016/0191-2615(86)90019-6.

Crainic, T. G, Frangioni, A., and Gendron, B. (2001). "Bundle-based relaxation methods for multicommodity capacitated fixed charge network design". *Discrete Applied Mathematics,* Vol. 112, Issues 1–3, 73–99. doi:10.1016/S0166-218X(00)00310-3.

Gendron, B., and Gouveia, L. (2016). "Reformulations by discretization for piecewise linear integer multicommodity network flow problems". *Transportation Science,* 51(2), 629–649.

Ghaffarinasab, N., Zare Andaryan, A., and Ebadi Torkayesh, A. (2020). "Robust single allocation p-hub median problem under hose and hybrid demand uncertainties: models and algorithms". *International Journal of Management Science and Engineering Management*, 15(3), 184-195.

Guimarães, L. R., de Sousa, J. P., and Prata, B. D. A. (2020). "Variable fixing heuristics for the capacitated multicommodity network flow problem with multiple transport lines, a heterogeneous fleet and time windows". *Transportation Letters*, 1-10.

Holmberg, K., and Yuan, D. (2000). "A Lagrangian heuristic based branch-and-bound approach for the capacitated network design problem". *Operations Research,* 48(3), 461–481.

Karsten, C. V., Pisinger, D., Røpke, S., and Brouer, B. D. (2015). "The time constrained multi-commodity network flow problem and its application to liner shipping network design". *Transportation Research Part E: Logistics and Transportation Review,* Vol. 76, 122–138. doi:10.1016/j.tre.2015.01.005.

Katayama, N., Chen, M., and Kubo, M. (2009). "A capacity scaling heuristic for the multicommodity capacitated network design problem". *Journal of Computational and Applied Mathematics,* 232(1), 90–101. doi:10.1016/j.cam.2008.10.055.

Kazemzadeh, M. R. A., Bektaş, T., Crainic, T. G., Frangioni, A., Gendron, B., and Gorgone, E. (2021). "Node-based Lagrangian relaxations for multicommodity capacitated fixed-charge network design". *Discrete Applied Mathematics*. doi:10.1016/j.dam.2020.12.024.

Moradi, S., Raith, A., and Ehrgott, M. (2015). "A bi-objective column generation algorithm for the multi-commodity minimum cost flow problem". *European Journal of Operational Research,* 244(2), 369–378. doi:10.1016/j.ejor.2015.01.021.

Oğuz, M., Bektaş, T., and Bennell, J. A. (2018). "Multicommodity flows and Benders decomposition for restricted continuous location problems". *European Journal of Operational Research*, 266(3), 851–863. doi:10.1016/j.ejor.2017.11.033.

Paraskevopoulos, D. C., Gürel, S., and Bektaş, T. (2016). "The congested multicommodity network design problem". *Transportation Research Part E: Logistics and Transportation Review*, Vol. 85, 166–187. doi:10.1016/j.tre.2015.10.007.

Salimifard, K., and Bigharaz, S. (2020). "The multicommodity network flow problem: State of the art classification, applications, and solution methods". *Operational Research*, 1-47. doi:10.1007/s12351-020-00564-8.

Trivella, A., Corman, F., Koza, D. F., and Pisinger, D. (2021). "The multi-commodity network flow problem with soft transit time constraints: Application to liner shipping". *Transportation Research Part E: Logistics and Transportation Review*, *150*, 102342.

Yousefi Nejad Attari, M., Ebadi Torkayesh, A., Malmir, B., and Neyshabouri Jami, E. (2020). "Robust possibilistic programming for joint order batching and picker routing problem in warehouse management". *International Journal of Production Research*, 59(2), 1-19.

Wang, I.-L. (2018a). "Multicommodity Network Flows : A Survey , Part I : Applications and Formulations". *International Journal of Operational Research*, 15(4), 145–153. doi:10.6886/IJOR.201812.

Wang, I.-L. (2018b). "Multicommodity Network Flows : A Survey, Part II : Solution Methods". *International Journal of Operational Research*, 15(4), 155–173.

Wolsey, L. A. (1998). *Integer Programming*. New York, John Wiley & Sons.