

A Bi-objective Column Generation Algorithm for the Multi-Commodity Minimum Cost Flow Problem

S. Moradi¹, A. Raith¹ and M. Ehrgott²

¹ *Department of Engineering Science, University of Auckland, New Zealand
s.moradi@auckland.ac.nz, a.raith@auckland.ac.nz*

² *Department of Management Science, Lancaster University, United
Kingdom m.ehrgott@lancaster.ac.uk*

Abstract

We present a column generation algorithm for solving the bi-objective multi-commodity minimum cost flow problem. This method is based on the bi-objective simplex method and Dantzig-Wolfe decomposition. The method is initialised by optimising the problem with respect to the first objective, a single objective multi-commodity flow problem, which is solved using Dantzig-Wolfe decomposition. Then, similar to the bi-objective simplex method, our algorithm iteratively moves from one non-dominated extreme point to the next one by finding entering variables with the maximum ratio of improvement of the second objective over deterioration of the first objective. Our method reformulates the problem into a bi-objective master problem over a set of capacity constraints and several single objective linear fractional sub-problems each over a set of network flow conservation constraints. The master problem iteratively updates cost coefficients for the fractional sub-problems. Based on these cost coefficients an optimal solution of each sub-problem is obtained. The solution with the best ratio objective value out of all sub-problems represents the entering variable for the master basis. The algorithm terminates when there is no entering variable which can improve the second objective by deteriorating the first objective. This implies that all non-dominated extreme points of the original problem are obtained. We report on the performance of the algorithm on several directed bi-objective network instances with different characteristics and different numbers of commodities.

Keywords: Network flows, bi-objective multi-commodity minimum cost flow problem, Dantzig-Wolfe decomposition, column generation, bi-objective simplex method

1. Introduction

The multi-commodity minimum cost flow problem (*MCMCF*) is a network optimisation problem where several commodities need to be sent from their source nodes to their sink nodes. Individual commodities share arcs and compete for the capacity of the arcs. The *MCMCF* problem can be modelled as a linear optimisation problem with two sets of constraints: The flow conservation constraints and the capacity constraints which tie the commodities together. These constraints have a special block diagonal shape. Taking advantage of this special structure, several decomposition approaches for solving the problem have been developed (see [1] and references therein). In many application contexts of network models such as transportation, assignment, transshipment and location problems, there is more than one objective that has to be taken into account. These objectives include time, cost, risk, environmental concerns etc. Thus, multi-objective flow models are more appropriate for modelling real-world decision making situations than single objective models [2, 3, 4, 5, 6, 7, 8]. In this paper we consider the bi-objective multi-commodity minimum cost flow problem (*BMCMCF*). We propose a decomposition algorithm that is based on the bi-objective simplex algorithm and employs a new generalisation of Dantzig-Wolfe decomposition to bi-objective linear programmes.

Let $G := (\mathcal{V}, \mathcal{A})$ be a directed network with a set of nodes or vertices $\mathcal{V} := \{1, 2, \dots, n\}$ and a set of arcs $\mathcal{A} \subseteq \mathcal{V} \times \mathcal{V}$ with $|\mathcal{A}| = m$. Furthermore, let $(c_a^{1,k}, c_a^{2,k})$ be the pair of unit flow costs on arc $a \in \mathcal{A}$ for commodity k and x_a^k represent the amount of flow of commodity k going through arc $a \in \mathcal{A}$. Let \mathbf{E} be the node arc incidence matrix of the network and let $\mathbf{x}^k := (x_a^k \text{ for } a \in \mathcal{A})$ be the flow vector for commodity k . Let \mathbf{b}^k be the demand vector for each commodity k and \mathbf{u} be the vector of arc capacities. By defining cost vectors $\mathbf{c}^{1,k} := (c_a^{1,k} \text{ for } a \in \mathcal{A})$ and $\mathbf{c}^{2,k} := (c_a^{2,k} \text{ for } a \in \mathcal{A})$ the *BMCMCF* problem can be written as the following bi-objective linear programme

$$\begin{aligned}
\min \quad & z(\mathbf{x}) := \begin{pmatrix} z^1(\mathbf{x}) := \sum_{k=1}^q (\mathbf{c}^{1,k})^T \mathbf{x}^k \\ z^2(\mathbf{x}) := \sum_{k=1}^q (\mathbf{c}^{2,k})^T \mathbf{x}^k \end{pmatrix} \\
\text{s.t.} \quad & \mathbf{E}\mathbf{x}^1 & & & = \mathbf{b}^1 \\
& \mathbf{E}\mathbf{x}^2 & & & = \mathbf{b}^2 \\
& & & & \dots \\
& & & \mathbf{E}\mathbf{x}^q & = \mathbf{b}^q \\
& \mathbf{I}\mathbf{x}^1 + \mathbf{I}\mathbf{x}^2 + \dots + \mathbf{I}\mathbf{x}^q + \mathbf{I}\mathbf{s} & = & \mathbf{u} \\
& \mathbf{x}^k, \mathbf{s} \geq \mathbf{0}, \text{ for all } k := 1, 2, \dots, q,
\end{aligned} \tag{1}$$

where \mathbf{I} is an $m \times m$ identity matrix and \mathbf{s} is a vector of slack variables. We assume that $\sum_{i=1}^n b_i^k = 0$, $k := 1, 2, \dots, q$, otherwise the problem is infeasible. The first q sets of constraints represent flow conservation at the n nodes for all q commodities. A value $b_i^k > 0$, $b_i^k < 0$, or $b_i^k = 0$, respectively, indicates that node i is a supply node, a demand node, or a transshipment node for commodity k . The next set of m constraints ensures that the overall flow of commodities along each arc amounts to at most the arc capacities.

The *BMCMCF* problem (1) is a bi-objective linear optimisation problem which can be solved by existing bi-objective linear programming algorithms [9], such as the bi-objective simplex method, see e.g. [10] and Section 3.1. The specially structured block diagonal constraint matrix of problem (1) permits the application of the Dantzig-Wolfe decomposition method. This has been done in the single objective case, see Section 3.2, In this paper, we generalise this approach to the bi-objective case. We demonstrate how Dantzig-Wolfe decomposition can be used to generate columns of the *BMCMCF* problem in the context of the bi-objective simplex method, thereby extending our preliminary results in [11].

By integrating the bi-objective simplex method with the Dantzig-Wolfe decomposition method we present a new method for solving the *BMCMCF* problem which we shall refer to as the bi-objective simplex decomposition (*BOSD*) method.

This paper is organised as follows: Recent literature is briefly discussed in Section 2. Necessary mathematical background as well as the bi-objective simplex method and standard Dantzig-Wolfe decomposition method are ex-

plained in Section 3. In Section 4, we introduce our proposed *BOSD* method. Finally numerical results are illustrated in Section 5.

2. Literature

There does not exist a lot of research on multi-objective *MCMCF* problems, so we also review research conducted on multi-objective minimum cost flow problems with only a single commodity. The most recent survey on multi-objective minimum cost flow problems is by Hamacher et al. [12]. We will therefore briefly mention only newer published work on multi-objective network flow problems, one of which considers multiple commodities.

Sedeño-Noda et al. [13] present a change of variables method to solve the bi-objective *undirected* two-commodity minimum cost flow problem. They formulate the problem as follows:

$$\begin{aligned} \min \quad z(\mathbf{x}) &:= \begin{pmatrix} z^1(\mathbf{x}) := \sum_{k:=1,2} \sum_{(i,j) \in \mathcal{A}} c_{ij}^k x_{ij}^k \\ z^2(\mathbf{x}) := \sum_{k:=1,2} \sum_{(i,j) \in \mathcal{A}} d_{ij}^k x_{ij}^k \end{pmatrix} \\ \text{s.t.} \quad \sum_{\{j:(i,j) \in \mathcal{A}\}} x_{ij}^k - \sum_{\{j:(j,i) \in \mathcal{A}\}} x_{ji}^k &= \begin{cases} b^k & \text{if } i = s^k \\ 0 & \text{if } i \in \mathcal{V} - \{s^k, t^k\}, k := 1, 2 \\ -b^k & \text{if } i = t^k \end{cases} \\ |x_{ij}^1| + |x_{ij}^2| &\leq u_{ij}, \text{ for all } (i, j) \in \mathcal{A}. \end{aligned} \tag{2}$$

By using absolute values for the amount of flow in the last set of constraints of (2), the values of the flow can be negative for each edge. The method by Sedeño-Noda et al. splits the problem into two bi-objective minimum cost flow problems with a single commodity and uses the parametric network simplex method to solve these problems. This method cannot be extended to more than two commodities and also works only for *undirected* two-commodity problems.

Eusébio et al. [14] develop a primal–dual bi-objective simplex algorithm for the bi-objective single commodity network flow problem that is based on the bi-objective primal–dual simplex algorithm of Ehrgott et al. [15] but uses reduced cost information to avoid redundancy. They report that their method does not perform efficiently on large scale instances.

Raith and Ehrgott [16] present an algorithm to compute a complete set of efficient solutions for the bi-objective (single commodity) integer minimum cost flow (*BIMCF*) problem based on the two-phase method, see [17] for a recent survey on the two-phase method. In Phase 1 they use a parametric network simplex algorithm [18] to compute all integer solutions the images of which are extreme points of the boundary of $\text{conv}(\mathcal{Z})$, where $\text{conv}(\cdot)$ is the convex hull of its argument and \mathcal{Z} is the image in objective space of the feasible set. Since they solve an integer bi-objective optimisation problem, the images of some of the efficient solutions lie in the interior of $\text{conv}(\mathcal{Z})$. In Phase 2, these remaining efficient solutions are computed using a k best flow algorithm [19] on single objective weighted sum problem. Since multi-objective integer problems are harder than continuous ones, *BIMCF* instances solved in literature all have small size and all algorithms mentioned here perform well only for small and medium sized instances.

Eusébio and Figueira [20] present and implement an algorithm for finding all supported efficient solutions to the *BIMCF*. Their method is based on a negative-cycle algorithm used in single objective minimum cost flow problems [21] applied to a sequence of parametric problems. They prove that all supported efficient solutions are connected via a chain of zero-cost cycles in the incremental graph constructed from basic feasible solutions corresponding to extreme efficient solutions.

Eusébio and Figueira [22] solve a sequence of ε -constraint problems [23] in the context of finding all the non-dominated solutions of the *BIMCF* problem. The integer optimal solutions to the ε -constraint problems are obtained by a branch-and-bound method. Similar to [14], this method performs well only on small or medium size instances.

With the exception of work on *undirected* bi-objective two-commodity *MCMCF* problem [13] there does not exist any research on the multi-objective *MCMCF* problems, in particular there is no published research on Dantzig-Wolfe decomposition for multi-objective network flow problems.

3. Background

In this section, the necessary mathematical background is introduced. We also summarise the bi-objective simplex method and the application of standard Dantzig-Wolfe decomposition to the single objective *MCMCF* problem. Consider a bi-objective linear optimisation problem (*BLP*)

$$\begin{aligned}
\min \quad & z(\mathbf{x}) := \begin{pmatrix} z^1(\mathbf{x}) := (\mathbf{c}^1)^T \mathbf{x} \\ z^2(\mathbf{x}) := (\mathbf{c}^2)^T \mathbf{x} \end{pmatrix} \\
\text{s.t.} \quad & \mathbf{A}\mathbf{x} = \mathbf{b}, \\
& \mathbf{x} \geq \mathbf{0},
\end{aligned} \tag{3}$$

where $(\mathbf{c}^1)^T, (\mathbf{c}^2)^T \in \mathbb{R}^n$ are $1 \times n$ objective or criteria vectors. The feasible set in decision space $\mathcal{X} := \{\mathbf{x} \in \mathbb{R}^n : \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$ is defined by the $m \times n$ constraint matrix \mathbf{A} and right hand side vector $\mathbf{b} \in \mathbb{R}^m$.

Let $\mathcal{Z} := \{(z^1(\mathbf{x}), z^2(\mathbf{x})) : \mathbf{x} \in \mathcal{X}\}$ be the image of \mathcal{X} under the objective functions. A feasible solution $\hat{x} \in \mathcal{X}$ of *BLP* (3) is *efficient* and if only if there does not exist any $x' \in \mathcal{X}$ with $(z^1(\mathbf{x}'), z^2(\mathbf{x}')) \leq (z^1(\hat{\mathbf{x}}), z^2(\hat{\mathbf{x}}))$ and $z(\mathbf{x}') \neq z(\hat{\mathbf{x}})$. The image of an efficient solution $z(\hat{\mathbf{x}}) := (z^1(\hat{\mathbf{x}}), z^2(\hat{\mathbf{x}}))$ is called a *non-dominated point*. Since model (3) is a linear model, the images of all the efficient solutions lie on the boundary of $\text{conv}(\mathcal{Z})$. These solutions are called *supported efficient solutions* and can be obtained by solving (single objective) weighted sum problems [24]

$$\min_{x \in \mathcal{X}} \lambda z^1(x) + (1 - \lambda) z^2(x)$$

for some $0 < \lambda < 1$. The supported efficient solutions which define an extreme point of \mathcal{Z} are called *extreme efficient solutions*. By \mathcal{X}_{ex} and \mathcal{Z}_{ex} we denote the set of all extreme efficient solutions and non-dominated extreme points.

3.1. Bi-objective simplex method

Modelling the *BMCMCF* problem (1) as a linear programme permits the use of the standard bi-objective simplex method. Ehrgott [10] gives a comprehensive explanation of this method and we use the same notation here. The bi-objective simplex method initially starts by optimising the problem with respect to the first objective. The method then iteratively moves from one non-dominated extreme point to the next one by finding entering variables with the maximum ratio of improvement of the second objective over the deterioration of the first objective, see also selection of t in step 5 of Algorithm 1. The method stops when all the non-dominated extreme points are obtained. The initial solution may be weakly efficient and the algorithm may find some efficient solutions that do not define non-dominated extreme points. These solutions can be easily discarded at the

end of the algorithm. Note that the resulting set might still contain solutions which map to the same point in objective space. All non-dominated points can then be obtained as convex combinations of the obtained extreme non-dominated points. The procedure is stated as Algorithm 1.

Algorithm 1 (Parametric simplex for *BLP*)

- 1: **Input:** Data \mathbf{A} , \mathbf{b} , \mathbf{c}^1 and \mathbf{c}^2 for a *BLP*.
 - 2: Obtain an optimal solution x_o and an optimal basis \mathcal{B} with respect to the first objective component and set $\mathcal{X}_{ex} := \{x_o\}$.
 - 3: Compute $\mathcal{N} := \{1, \dots, n\} \setminus \mathcal{B}$, $\tilde{\mathbf{A}} := \mathbf{A}_{\mathcal{B}}^{-1} \mathbf{A}$, $\tilde{\mathbf{b}} := \mathbf{A}_{\mathcal{B}}^{-1} \mathbf{b}$ and $(\tilde{\mathbf{c}}^j)^T := (\mathbf{c}^j)^T - (\mathbf{c}_{\mathcal{B}}^j)^T \tilde{\mathbf{A}}$ for $j := 1, 2$.
 - 4: **while** $\mathcal{I} := \{i \in \mathcal{N} : \tilde{c}_i^2 < 0, \tilde{c}_i^1 \geq 0\} \neq \emptyset$ **do**
 - 5: $t \in \operatorname{argmax} \left\{ i \in \mathcal{I} : \frac{-\tilde{c}_i^2}{\tilde{c}_i^1 - \tilde{c}_i^2} \right\}$. // Entering variable selection
 - 6: $r \in \operatorname{argmin} \left\{ l \in \mathcal{B} : \frac{\tilde{b}_l}{\tilde{\mathbf{A}}_{tl}} > 0 \right\}$. // Leaving variable selection
 - 7: $\mathcal{B} \leftarrow (\mathcal{B} \setminus \{r\}) \cup \{t\}$ and update \mathcal{N} , $\tilde{\mathbf{A}}$, $\tilde{\mathbf{b}}$, $\tilde{\mathbf{c}}^j$ for $j := 1, 2$.
 - 8: Compute new efficient solution x_{new} and set $\mathcal{X}_{ex} \leftarrow \mathcal{X}_{ex} \cup \{x_{new}\}$.
 - 9: **end while**
 - 10: Discard from \mathcal{X}_{ex} any points the images of which are not extreme.
 - 11: **Output:** \mathcal{X}_{ex} , a set of pre-images of all non-dominated extreme points.
-

3.2. Dantzig-Wolfe decomposition method for the *MCMCF* problem

Tomlin [25] has devised an algorithm for solving the *MCMCF* problem based on the Dantzig-Wolfe decomposition method. In the Dantzig-Wolfe decomposition method the original problem, a single objective version of (1), is reformulated into a master problem (*MP*) over a set of capacity (complicating) constraints, and q sub-problems, each over a set of flow conservation constraints, $\mathbf{E}\mathbf{x}^k = \mathbf{b}^k$ for $k := 1, 2, \dots, q$. The *MP MCMCF* can be written as follows:

$$\begin{aligned}
 \min \quad & z^1(\mathbf{x}) := (\mathbf{c}^1)^T \mathbf{x} \\
 \text{s.t.} \quad & \mathbf{A}_{hard} \mathbf{x} + \mathbf{I} \mathbf{s} = \mathbf{u}, \\
 & \mathbf{x}^k \in \mathcal{X}_k, \\
 & \mathbf{x}, \mathbf{s} \geq \mathbf{0},
 \end{aligned} \tag{4}$$

where $\mathcal{X}_k := \{\mathbf{x}^k : \mathbf{E}\mathbf{x}^k = \mathbf{b}^k\}$, $\mathbf{x} := [\mathbf{x}^1, \dots, \mathbf{x}^q]$ and $\mathbf{A}_{hard} := [\mathbf{I}, \dots, \mathbf{I}]$ is composed of q identity matrices of size $m \times m$. Starting with a basic feasible solution, the *MP* (4) iteratively updates cost coefficients for the sub-problems. Based on these cost coefficients an optimal solution of each sub-problem is obtained. These solutions are the most improving columns (corresponding to non-basic variables) to enter the *MP* basis. This method continues until no sub-problem can find a column to improve the *MP* and it is guaranteed that the optimal solution of the *MCMCF* problem is obtained. Let $\mathcal{Y}_k := \{\mathbf{y}_1^k, \mathbf{y}_2^k, \dots, \mathbf{y}_{n_k}^k\}$ be the extreme points of feasible set $\mathcal{X}_k := \{\mathbf{x}^k : \mathbf{E}\mathbf{x}^k = \mathbf{b}^k\}$ for $k := 1, 2, \dots, q$ of the single objective version of problem (1). Then any feasible \mathbf{x}^k can be expressed as a convex combination of the elements of \mathcal{Y}_k as follows:

$$\mathbf{x}^k := \sum_{j=1}^{n_k} \lambda_j^k \mathbf{y}_j^k, \quad \sum_{j=1}^{n_k} \lambda_j^k = 1, \quad \lambda_j^k \geq 0, \quad j := 1, 2, \dots, n_k. \quad (5)$$

Substituting (5) for \mathbf{x}^k in the *MCMCF* (1) with the single objective z^1 , we get the following problem *MP* (6), where we indicate the dual variables on the right hand side of the vertical line.

$$\begin{aligned} \min \quad & z^1(\boldsymbol{\lambda}) := \sum_{k=1}^q \sum_{j=1}^{n_k} \lambda_j^k ((\mathbf{c}^{1,k})^T \mathbf{y}_j^k) \\ \text{s.t.} \quad & \sum_{j=1}^{n_1} \lambda_j^1 = 1 && \alpha^1 \\ & \sum_{j=1}^{n_2} \lambda_j^2 = 1 && \alpha^2 \\ & \dots && \dots \\ & \sum_{j=1}^{n_q} \lambda_j^q = 1 && \alpha^q \\ & \sum_{j=1}^{n_1} \lambda_j^1 (\mathbf{I}\mathbf{y}_j^1) + \sum_{j=1}^{n_2} \lambda_j^2 (\mathbf{I}\mathbf{y}_j^2) + \dots + \sum_{j=1}^{n_q} \lambda_j^q (\mathbf{I}\mathbf{y}_j^q) + \mathbf{I}\mathbf{s} = \mathbf{u} && \mathbf{w} \\ & \lambda_j^k \geq 0, \text{ for all } j := 1, 2, \dots, n_k, k := 1, 2, \dots, q. && \end{aligned} \quad (6)$$

Suppose that we have a basic feasible solution for problem (6), in terms of the λ_j^k variables. Let $(\boldsymbol{\alpha}, \mathbf{w})$ be the vector of dual variables for this basic

feasible solution where $\boldsymbol{\alpha}$ has q components and \mathbf{w} has m components. For this solution to be optimal, it must also be feasible for the dual problem. This condition is called dual feasibility and for problem (6) can be stated as:

1. $-w_a \geq 0$ for each arc $a \in A$, and
2. $(\mathbf{c}^{1,k} - \mathbf{w})^T \mathbf{y}^k - \alpha^k \geq 0$ for each λ_j^k , for all $j := 1, 2, \dots, n_k$, $k := 1, 2, \dots, q$.

If the first condition is violated for any arc the related slack variable s_a will be introduced into the basis of the *MP*. If the second condition is violated the corresponding λ_j^k variable is a candidate to enter the basis of the *MP*. For commodity k the most improving column \mathbf{y}_j^k (non-basic λ_j^k variable) to enter the basis can be found by solving the sub-problem

$$\begin{aligned} \min \quad & g(\mathbf{y}^k) := (\mathbf{c}^{1,k} - \mathbf{w})^T \mathbf{y}^k - \alpha^k \\ \text{s.t.} \quad & \mathbf{E}\mathbf{y}^k = \mathbf{b}^k, \\ & \mathbf{y}^k \geq \mathbf{0}. \end{aligned} \tag{7}$$

If we consider the case in which the network has a single source and sink for each commodity $k := 1, 2, \dots, q$, which we may do without loss of generality, then the sub-problem (7) requires finding a minimum cost flow in a network with no capacity on arcs and with arc costs $(\mathbf{c}^{1,k} - \mathbf{w})$. By adding all slack variables s_a with negative $-w_a$ to the basis of the *MP*, we ensure $-w_a \geq 0$ for all $a \in A$; see step 5 in Algorithm 2. Therefore, $(\mathbf{c}^{1,k} - \mathbf{w})$ is positive for all arcs $a \in A$. In this case sub-problem (7) is a shortest path problem which can be solved by many efficient shortest path algorithms [26]. If the optimal solution \mathbf{y}_j^k of problem (7) has a negative objective value, non-basic variable λ_j^k is an entering variable for the basis of the *MP*. After finding the entering variable, it is added to the basis of the *MP* and the leaving variable is determined in the usual way in the simplex algorithm. By pivoting, the basis inverse, dual variables, and right-hand-side are updated. The method continues until the dual feasibility conditions are satisfied, which means there does not exist any candidate variable to enter the basis and an optimal solution of the *MCMCF* problem is obtained. This procedure is stated as Algorithm 2

4. Bi-objective simplex decomposition method

In this section we present the proposed *BOSD* method for solving the *BM-CMCF* problem. The *BOSD* method reformulates the *BMCMCF* problem

Algorithm 2 (Dantzig-Wolfe decomposition method for the *MCMCF* problem)

- 1: **Input:** Data \mathbf{c}^1 , \mathbf{E} , \mathbf{b} , \mathbf{u} , m and q for a linear *MCMCF* problem.
 - 2: $\mathbf{A}_{hard} := [\mathbf{I}_m, \dots, \mathbf{I}_m] \in \mathbb{R}^{m \times mq}$. Find an initial feasible solution for the *MP* (6), in which the constraints are expressed as $\mathbf{A} \begin{pmatrix} \boldsymbol{\lambda} \\ \mathbf{s} \end{pmatrix} = \begin{pmatrix} \mathbf{1} \\ \mathbf{u} \end{pmatrix} =: \mathbf{u}'$, and obtain the master basis \mathcal{B} .
 - 3: ~~Compute $\tilde{\mathbf{A}}_{hard} := \mathbf{A}_{hard, \mathcal{B}}^{-1} \mathbf{A}$, $\tilde{\mathbf{u}} := \mathbf{A}_{hard, \mathcal{B}}^{-1} \mathbf{u}$ and $(\boldsymbol{\alpha}, \mathbf{w}) := \hat{\mathbf{c}}_{\mathcal{B}}^1 \mathbf{A}_{hard, \mathcal{B}}^{-1}$~~
 Compute $\tilde{\mathbf{A}} := \mathbf{A}_{\mathcal{B}}^{-1} \mathbf{A}$, $\tilde{\mathbf{u}} := \mathbf{A}_{\mathcal{B}}^{-1} \mathbf{u}'$ and $(\boldsymbol{\alpha}, \mathbf{w}) := \hat{\mathbf{c}}_{\mathcal{B}}^1 \mathbf{A}_{\mathcal{B}}^{-1}$, where $\hat{c}_j^{1,k} := (\mathbf{c}^1)^T \mathbf{y}_j^k$ for λ_j^k variables.
 - 4: $\mathcal{T} := \emptyset$. // \mathcal{T} is the set of entering variable candidates
 - 5: **if** $-w_a < 0$, $a \in \mathcal{A}$ **then**
 - 6: $\mathcal{T} \leftarrow \mathcal{T} \cup \{s_a\}$.
 - 7: **else**
 - 8: For each commodity k ($k := 1, 2, \dots, q$) solve sub-problem (7) and find the optimal solution \mathbf{y}_j^k .
 - 9: **if** $g(\mathbf{y}_j^k) < 0$ **then**
 - 10: $\mathcal{T} \leftarrow \mathcal{T} \cup \{\lambda_j^k\}$.
 - 11: **end if**
 - 12: **end if**
 - 13: **if** $\mathcal{T} \neq \emptyset$ **then**
 - 14: Choose $t \in \mathcal{T}$ and find $\tilde{\mathbf{A}}_{hard, t}$, $\tilde{\mathbf{A}}_t$ as an entering column. // Entering variable selection
 - 15: $r \in \operatorname{argmin} \left\{ l \in \mathcal{B} : \frac{\tilde{u}_l}{\tilde{\mathbf{A}}_{hard, tl}}, \tilde{\mathbf{A}}_{hard, tl} \geq 0 \right\}$.
 $r \in \operatorname{argmin} \left\{ l \in \mathcal{B} : \frac{\tilde{u}'_l}{\tilde{\mathbf{A}}_{tl}}, \tilde{\mathbf{A}}_{tl} > 0 \right\}$. // Leaving variable selection
 - 16: $\mathcal{B} \leftarrow (\mathcal{B} \setminus \{r\}) \cup \{t\}$ and update $\tilde{\mathbf{A}}_{hard}$, $\tilde{\mathbf{A}}$, $\tilde{\mathbf{u}}'$, and $(\boldsymbol{\alpha}, \mathbf{w})$.
 - 17: Go to step 3.
 - 18: **end if**
 - 19: Compute optimal solution x_{opt} from \mathcal{B} .
 - 20: **Output:** An optimal solution x_{opt} of the *MCMCF* problem.
-

into a bi-objective master problem (*BMP*) over a set of capacity constraints and q sub-problems, each over a set of network flow conservation constraints, $\mathbf{E}\mathbf{x}^k = \mathbf{b}^k$ for $k := 1, 2, \dots, q$. The *BMP* *BMCMCF* can be written as bi-objective linear problem (8)

$$\begin{aligned} \min \quad & z(\mathbf{x}) := \begin{pmatrix} z^1(\mathbf{x}) := (\mathbf{c}^1)^T \mathbf{x} \\ z^2(\mathbf{x}) := (\mathbf{c}^2)^T \mathbf{x} \end{pmatrix} \\ \text{s.t.} \quad & \mathbf{A}_{hard} \mathbf{x} + \mathbf{I} \mathbf{s} = \mathbf{u}, \\ & \mathbf{x}^k \in \mathcal{X}_k, \\ & \mathbf{x}, \mathbf{s} \geq \mathbf{0}, \end{aligned} \tag{8}$$

where $\mathcal{X}_k := \{\mathbf{x}^k : \mathbf{E}\mathbf{x}^k = \mathbf{b}^k\}$, $\mathbf{x} := [\mathbf{x}^1, \dots, \mathbf{x}^q]$ and $\mathbf{A}_{hard} := [\mathbf{I}, \dots, \mathbf{I}]$ is composed of q identity matrices of size $m \times m$. The *BOSD* method works with the *BMP*, while sub-problems generate the columns to move from one non-dominated extreme point to the next one.

Applying the change of variables (5), the *BMCMCF* problem (1) can be written as *BMP* (9). Notice that each constraint now has two dual variables, one associated with each of the two objective functions. These are listed to the right hand side of the vertical line.

$$\begin{aligned} \min \quad & z(\boldsymbol{\lambda}) := \begin{pmatrix} z^1(\boldsymbol{\lambda}) := \sum_{k=1}^q \sum_{j=1}^{n_k} \lambda_j^k ((\mathbf{c}^{1,k})^T \mathbf{y}_j^k) \\ z^2(\boldsymbol{\lambda}) := \sum_{k=1}^q \sum_{j=1}^{n_k} \lambda_j^k ((\mathbf{c}^{2,k})^T \mathbf{y}_j^k) \end{pmatrix} \\ \text{s.t.} \quad & \sum_{j=1}^{n_1} \lambda_j^1 = 1 && \text{dual} \\ & \sum_{j=1}^{n_2} \lambda_j^2 = 1 && \alpha^{1,1} \quad \alpha^{2,1} \\ & \dots && \alpha^{1,2} \quad \alpha^{2,2} \\ & \sum_{j=1}^{n_q} \lambda_j^q = 1 && \dots \\ & \sum_{j=1}^{n_1} \lambda_j^1 (\mathbf{I} \mathbf{y}_j^1) + \sum_{j=1}^{n_2} \lambda_j^2 (\mathbf{I} \mathbf{y}_j^2) + \dots + \sum_{j=1}^{n_q} \lambda_j^q (\mathbf{I} \mathbf{y}_j^q) + \mathbf{I} \mathbf{s} = \mathbf{u}. && \alpha^{1,q} \quad \alpha^{2,q} \\ & \lambda_j^k \geq 0, \quad \text{for all } j := 1, 2, \dots, n_k, k := 1, 2, \dots, q. && \mathbf{w}^1 \quad \mathbf{w}^2 \\ & && \tag{9} \end{aligned}$$

The *BOSD* method initially starts by obtaining a solution which is minimal with respect to the first objective component. In this step, problem (9) becomes a single objective *MCMCF* problem, and the standard Dantzig-Wolfe decomposition method, as explained in Section 3.2, can be applied. Suppose that we have an initial extreme efficient solution of (9) and let \mathcal{B} , $(\boldsymbol{\alpha}^1, \mathbf{w}^1)$ and $(\boldsymbol{\alpha}^2, \mathbf{w}^2)$ respectively, be the corresponding basis for *MP* and the vectors of dual variables for the first and second objective. Vectors $\boldsymbol{\alpha}^1$ and $\boldsymbol{\alpha}^2$ have q components and \mathbf{w}^1 and \mathbf{w}^2 have m components. The *BOSD* method then iteratively generates entering variables which have a maximal ratio of improvement of the second objective function over deterioration of the first. This ratio is analogous to the ratio needed in the entering variable selection in step 5 of Algorithm 1. Slack variable s_a ($a \in \mathcal{A}$) is a candidate for introduction to the master basis if

$$-w_a^2 < 0 \text{ and } -w_a^1 \geq 0.$$

Non-basic variable λ_j^k is a candidate for introduction to the master basis if

$$\begin{aligned} (\mathbf{c}^2 - \mathbf{w}^2)^T \mathbf{y}^k - \alpha^{2,k} &< 0 \\ (\mathbf{c}^1 - \mathbf{w}^1)^T \mathbf{y}^k - \alpha^{1,k} &\geq 0. \end{aligned}$$

The ratio for slack variable s_a can be easily obtained from

$$\mu_a := \frac{-w_a^2}{w_a^1 - w_a^2}. \quad (10)$$

For commodity k the most improving column \mathbf{y}_j^k (variable λ_j^k) can be generated by solving the following fractional optimisation sub-problem:

$$\begin{aligned} \max \quad g^k(\mathbf{y}^k) &:= \frac{-((\mathbf{c}^2 - \mathbf{w}^2)^T \mathbf{y}^k - \alpha^{2,k})}{(\mathbf{c}^1 - \mathbf{w}^1)^T \mathbf{y}^k - \alpha^{1,k} - ((\mathbf{c}^2 - \mathbf{w}^2)^T \mathbf{y}^k - \alpha^{2,k})} \\ \text{s.t.} \quad \mathbf{E}\mathbf{y}^k &= \mathbf{b}^k, \\ (\mathbf{c}^2 - \mathbf{w}^2)^T \mathbf{y}^k - \alpha^{2,k} &< 0, \\ (\mathbf{c}^1 - \mathbf{w}^1)^T \mathbf{y}^k - \alpha^{1,k} &\geq 0, \\ \mathbf{y}^k &\geq \mathbf{0}. \end{aligned} \quad (11)$$

Here, an optimal solution of the linear fractional sub-problem (11) is obtained by applying the Charnes-Cooper variable transformation method [27].

Applying the Charnes-Cooper transformation as follows,

$$\mathbf{L}^k := \frac{1}{(\mathbf{c}^1 - \mathbf{w}^1)^T \mathbf{y}^k - \alpha^{1,k} - ((\mathbf{c}^2 - \mathbf{w}^2)^T \mathbf{y}^k - \alpha^{2,k})} \cdot \mathbf{y}^k,$$

$$r := \frac{1}{(\mathbf{c}^1 - \mathbf{w}^1)^T \mathbf{y}^k - \alpha^{1,k} - ((\mathbf{c}^2 - \mathbf{w}^2)^T \mathbf{y}^k - \alpha^{2,k})};$$

translates the fractional sub-problem (11) to an equivalent linear programme (12)

$$\begin{aligned} \max \quad & -((\mathbf{c}^2 - \mathbf{w}^2)^T \mathbf{L}^k - \alpha^{2,k} r) \\ \text{s.t.} \quad & \mathbf{E} \mathbf{L}^k = \mathbf{b}^k r, \\ & (\mathbf{c}^2 - \mathbf{w}^2)^T \mathbf{L}^k - \alpha^{2,k} r < \mathbf{0}, \\ & (\mathbf{c}^1 - \mathbf{w}^1)^T \mathbf{L}^k - \alpha^{1,k} r \geq \mathbf{0}, \\ & (\mathbf{c}^1 - \mathbf{w}^1)^T \mathbf{L}^k - \alpha^{1,k} r - ((\mathbf{c}^2 - \mathbf{w}^2)^T \mathbf{L}^k - \alpha^{2,k} r) = 1, \\ & r \geq 0. \end{aligned} \tag{12}$$

If the optimal solution \mathbf{y}_j^k of problem (11) has a positive objective value, non-basic variable λ_j^k is a candidate to enter the basis. Among the entering variable candidates we choose the one with maximum ratio, i.e. maximum value of (10) or maximum objective function value of (11). The identified entering variable is added to the basis of the *MP* and the leaving variable is determined according to standard simplex rules as in step 12 of Algorithm 3. The *BOSD* method continues until there does not exist any entering variable which can improve the second objective by deteriorating the first objective, which implies that all non-dominated extreme points are obtained. Note that the resulting set might still contain solutions that map to the same point in objective space, as in Algorithm 1. The *BOSD* Algorithm is stated as Algorithm 3.

5. Computational results

In this section, we perform computational experiments with the *BOSD* method on several directed bi-objective network instances with different characteristics and different numbers of commodities. All numerical tests are performed on a Microsoft Windows 7 Enterprise Service Pack 1 computer

Algorithm 3 (Bi-Objective Simplex Decomposition Algorithm)

- 1: **Input:** Data \mathbf{E} , \mathbf{b} , \mathbf{c}^1 , \mathbf{c}^2 , m and q for a *BMCMCF* problem.
 - 2: $-\mathbf{A}_{hard} := [\mathbf{I}_m, \dots, \mathbf{I}_m] \in \mathbb{R}^{m \times mq}$. Obtain an optimal solution x_o and an optimal master basis \mathcal{B} with respect to the first objective of problem (9), in which the constraints are expressed as $\mathbf{A} \begin{pmatrix} \boldsymbol{\lambda} \\ \mathbf{s} \end{pmatrix} = \begin{pmatrix} \mathbf{1} \\ \mathbf{u} \end{pmatrix} =: \mathbf{u}'$, by standard Dantzig-Wolfe decomposition and set $\mathcal{X}_{ex} := \{x_o\}$.
 - 3: Compute $\tilde{\mathbf{A}}_{hard} := \mathbf{A}_{hard, \mathcal{B}}^{-1} \mathbf{A}$, $\tilde{\mathbf{u}} := \mathbf{A}_{hard, \mathcal{B}}^{-1} \mathbf{u}$, $(\boldsymbol{\alpha}^1, \mathbf{w}^1) := \hat{\mathbf{c}}_{\mathcal{B}}^1 \mathbf{A}_{hard, \mathcal{B}}^{-1}$ and $(\boldsymbol{\alpha}^2, \mathbf{w}^2) := \hat{\mathbf{c}}_{\mathcal{B}}^2 \mathbf{A}_{hard, \mathcal{B}}^{-1}$
 Compute $\tilde{\mathbf{A}} := \mathbf{A}_{\mathcal{B}}^{-1} \mathbf{A}$, $\tilde{\mathbf{u}}' := \mathbf{A}_{\mathcal{B}}^{-1} \mathbf{u}'$, $(\boldsymbol{\alpha}^1, \mathbf{w}^1) := \hat{\mathbf{c}}_{\mathcal{B}}^1 \mathbf{A}_{\mathcal{B}}^{-1}$ and $(\boldsymbol{\alpha}^2, \mathbf{w}^2) := \hat{\mathbf{c}}_{\mathcal{B}}^2 \mathbf{A}_{\mathcal{B}}^{-1}$, where $\hat{c}_j^{1,k} := \mathbf{c}^{1,k} \mathbf{y}_j^k$ and $\hat{c}_j^{2,k} := \mathbf{c}^{2,k} \mathbf{y}_j^k$ for λ_j^k variables.
 - 4: $\mathcal{T} := \emptyset$ and $\mathcal{I} := \{a \in \{1, 2, \dots, m\} : -w_a^2 < 0, -w_a^1 \geq 0\}$. // \mathcal{T} is the set of entering variable candidates
 - 5: **if** $\mathcal{I} \neq \emptyset$ **then**
 - 6: $t_1 \in \operatorname{argmax} \left\{ i \in \mathcal{I} : \mu_i = \frac{-w_i^2}{w_i^1 - w_i^2} \right\}$, $\mathcal{T} \leftarrow \mathcal{T} \cup \{s_{t_1}\}$.
 - 7: **end if**
 - 8: For each commodity k solve fractional optimisation problem (11) and find an optimal solution \mathbf{y}_j^k .
 - 9: $t_2 \in \operatorname{argmax} \{k \in \{1, 2, \dots, q\} : g^k(\mathbf{y}_j^k) > 0\}$, $\mathcal{T} \leftarrow \mathcal{T} \cup \{\lambda_j^{t_2}\}$.
 - 10: **if** $\mathcal{T} \neq \emptyset$ **then**
 - 11: Choose $t \in \mathcal{T}$ with the maximum ratio and find $\tilde{\mathbf{A}}_{hard, t}$ as an entering column. // Entering variable selection
 - 12: $r \in \operatorname{argmin} \left\{ l \in \mathcal{B} : \frac{\tilde{u}_l}{\tilde{\mathbf{A}}_{hard, tl}} \tilde{\mathbf{A}}_{hard, tl} > 0 \right\}$.
 $r \in \operatorname{argmin} \left\{ l \in \mathcal{B} : \frac{\tilde{u}'_l}{\tilde{\mathbf{A}}_{tl}} \tilde{\mathbf{A}}_{tl} > 0 \right\}$. // Leaving variable selection
 - 13: $\mathcal{B} \leftarrow (\mathcal{B} \setminus \{r\}) \cup \{t\}$ and update $\tilde{\mathbf{A}}_{hard}$, $\tilde{\mathbf{A}}$, $\tilde{\mathbf{u}}'$, $(\boldsymbol{\alpha}^1, \mathbf{w}^1)$ and $(\boldsymbol{\alpha}^2, \mathbf{w}^2)$.
 - 14: Compute new efficient solution x_{new} and set $\mathcal{X}_{ex} \leftarrow \mathcal{X}_{ex} \cup \{x_{new}\}$.
 - 15: Go to step 3.
 - 16: **end if**
 - 17: Discard from \mathcal{X}_{ex} any points the images of which are not extreme.
 - 18: **Output:** \mathcal{X}_{ex} , a set of pre-images of all non-dominated extreme points.
-

with Intel (R) Xeon(R) CPU, 2.67 GHz, 6.00 GB RAM and 64-bit Operating System. The method is implemented in MATLAB R2013b. All single objective linear sub-problems were solved by GUROBI 5.6 [28]. We provide computational results obtained from several types of directed bi-objective network instances with one, two, three and five commodities.

We consider four groups of directed network instances. Instances N01–N12, F01–F12 and G01–G12 have the same structure as the bi-objective single commodity instances used in [16], which we modify to include several commodities. Instances of groups B01–B05 and G13–G15 have a larger number of nodes and arcs compared to the instances used in [16]. Instances of groups N01–N12, F01–F12 and B01–B05 are directed network instances generated by the NETGEN [29] generator which is modified to include a second objective function and multiple commodities. Table 1 shows NETGEN parameters for the generation of each set of networks, such as number of nodes, arcs, sources and sinks, etc. There are 30 instances for each set of parameters. Groups N01–N12 have varying total supply $\sum_{i \in \mathcal{V}: b_i > 0} b_i$, which increases as the number of nodes in the network increases. Groups F01–F12 have fixed total supplies of 500 and groups B01–B12 have fixed total supplies of 2100. Finally, instances of groups G01–G15 consist of networks with a grid structure. In these networks, nodes are arranged in a rectangular grid with given parameters height h , width w , maximum cost c_{max} , maximum capacity u_{max} and a total supply $\sum_{i \in \mathcal{V}: b_i > 0} b_i$. All grid instances are listed in Table 2. Again there are 30 instances for each set of parameters.

Figure 1 shows the non-dominated extreme points for one instance of class F12, with 80 nodes and 400 arcs, for one, two, three and five commodities which illustrates the trade-off between objective functions. This is typical for all examples. The large number of points for each commodity clearly illustrates the non-dominated set in objective space in each case, without additional lines to indicate the convex combinations of non-dominated extreme points. From Figure 1 we can also see that the number of non-dominated extreme points increases with the number of commodities. We note that curves do not exhibit a monotonic translation in any particular direction as the number of commodities increases.

In Table 3 the average CPU time t_{avg} as well as the average number of non-dominated extreme points $|\mathcal{Z}_{ex}|$ for different numbers of commodities are presented. For the groups comprised of small and medium sized instances, the average CPU times for solving the instances are between 0.06 and 19.45 seconds. For instances of groups B01–B05 and G13–G15, the average CPU

Table 1: NETGEN test instances.

Name	Nodes	Arcs	Sources	Sinks	Total supply $\sum_{i \in \mathcal{V}: b_i > 0} b_i$	Transshipment sources	Transshipment sinks
N01/F01	20	60	9	7	450/500	4	3
N02/F02	20	80	9	7	450/500	4	3
N03/F03	20	100	9	7	450/500	4	3
N04/F04	40	120	18	14	900/500	9	7
N05/F05	40	160	18	14	900/500	9	7
N06/F06	40	200	18	14	900/500	9	7
N07/F07	60	180	27	21	1350/500	14	10
N08/F08	60	240	27	21	1350/500	14	10
N09/F09	60	300	27	21	1350/500	14	10
N10/F10	80	240	35	38	1750/500	17	14
N11/F11	80	320	35	38	1750/500	17	14
N12/F12	80	400	35	38	1750/500	17	14
B01	100	400	40	50	2100	25	35
B02	200	800	40	50	2100	25	35
B03	300	1200	40	50	2100	25	35
B04	400	1600	40	50	2100	25	35
B05	500	2000	40	50	2100	25	35

time is between 12.90 and 648.58 seconds. The average number of non-dominated extreme points $|\mathcal{Z}_{ex}|$ increases logarithmically for both increases in the number of nodes and the number of arcs, as shown in Figures 2. The average CPU time t_{avg} increases quadratically, for both increases in the number of nodes and the number of arcs, as shown in Figures 3. By comparing the average CPU time t_{avg} between the groups N01–N12 and the groups F01–F12 we can see that t_{avg} increases as the total supply $\sum_{i \in \mathcal{V}: b_i > 0} b_i$ increases.

Although there is a positive correlation between the number of non-dominated extreme points $|\mathcal{Z}_{ex}|$ and the number of commodities, the average CPU time does not always increase when the number of commodities increases, which can be seen for instances of groups N08, N10, N11, N12 and F12. This may be due to the fact that by increasing the number of commodities there is a larger number of edges incident on each extreme point of the feasible polyhedron in decision space, hence it is more likely that we find an adjacent point with distinct objective value, i.e. the problem tends to have less degeneracy for our instances with two and three commodities compared to single commodity instances. To illustrate this, Table 4 presents t_{avg} , $|\mathcal{Z}_{ex}|$, the average number of iterations I_{avg} and the ratio $I_{avg}/|\mathcal{Z}_{ex}|$ for instances N08, N10, N11, N12 and F12. One iteration refers to generation of an en-

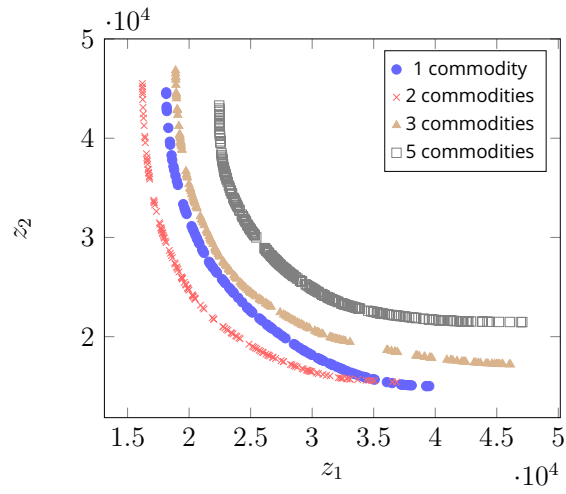


Figure 1: Non-dominated extreme points of one instance of class F12.

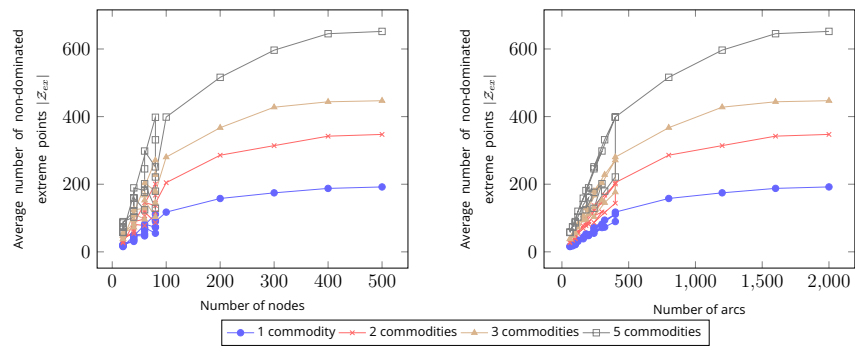


Figure 2: The average number of non-dominated extreme points $|Z_{ex}|$ increases logarithmically as the number of nodes and the number of arcs increases.

Table 2: Grid test instances.

Name	h	w	Nodes	Arcs	c_{max}	u_{max}	Total supply $\sum_{i \in \mathcal{V}: b_i > 0} b_i$
G01	4	5	20	62	100	250	500
G02	5	8	40	134	100	250	500
G03	6	10	60	208	100	250	500
G04	8	10	80	284	100	250	500
G05	6	10	60	208	100	375	500
G06	6	10	60	208	100	500	500
G07	6	10	60	208	25	250	500
G08	6	10	60	208	50	250	500
G09	8	10	80	284	100	375	500
G10	8	10	80	284	100	500	500
G11	8	10	80	284	25	500	500
G12	8	10	80	284	50	500	500
G13	10	10	100	360	50	50	500
G14	15	20	300	1130	50	50	500
G15	20	25	500	1910	50	50	500

tering column and updating the master problem, see step 14 of Algorithm 3. It is possible that updating the master problem does not change the current efficient solution and therefore I_{avg} can be bigger than $|\mathcal{Z}_{ex}|$. In Figure 4 we compare t_{avg} , $|\mathcal{Z}_{ex}|$, I_{avg} and $I_{avg}/|\mathcal{Z}_{ex}|$ of one of the sets of instances, N12, with different numbers of commodities. The figure is typical for the other sets. From Table 4 and Figure 4 it can be seen that t_{avg} does not always increase when the number of commodities increases. This happens as the ratio $I_{avg}/|\mathcal{Z}_{ex}|$ decreases sharply when the number of commodities increases. Therefore, the main factors influencing the computation time are the size of the network and the total supply.

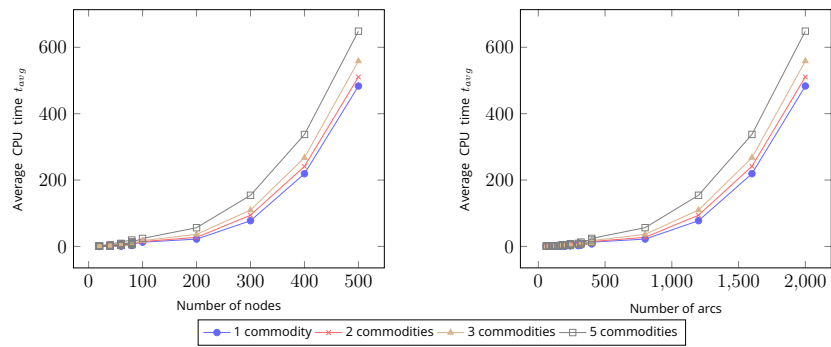


Figure 3: The average CPU time t_{avg} increases quadratically as the number of nodes and the number of arcs increases.

Table 3: Results for different sets of network instances.

Name	1 commodity		2 commodities		3 commodities		5 commodities	
	t_{avg} (s)	$ \mathcal{Z}_{ex} $	t_{avg} (s)	$ \mathcal{Z}_{ex} $	t_{avg} (s)	$ \mathcal{Z}_{ex} $	t_{avg} (s)	$ \mathcal{Z}_{ex} $
N01	0.13	16.52	0.26	29.07	0.38	38.25	0.69	57.26
N02	0.17	17.73	0.25	35.27	0.40	47.93	0.87	72.97
N03	0.18	21.83	0.27	38.67	0.46	55.50	1.02	85.60
N04	0.56	34.61	0.73	60.04	1.06	82.48	2.00	119.96
N05	0.77	43.66	1.04	76.66	1.47	107.93	2.78	158.56
N06	1.09	49.73	1.28	89.20	1.88	124.43	4.01	189.27
N07	2.14	53.10	2.17	97.10	3.22	128.65	4.74	181.27
N08	2.96	68.59	2.72	123.54	3.59	167.25	6.46	245.54
N09	3.67	82.04	3.88	145.50	4.87	203.31	9.18	298.21
N10	4.96	72.91	4.50	137.08	5.04	179.77	8.23	251.39
N11	6.13	93.81	5.88	167.77	7.45	228.08	13.00	331.46
N12	11.22	111.21	10.25	200.60	11.55	271.10	19.45	398.03
F01	0.11	15.57	0.21	26.62	0.32	37.79	0.90	58.41
F02	0.19	20.30	0.33	34.80	0.42	47.83	0.96	74.20
F03	0.20	21.30	0.39	40.63	0.61	55.50	1.03	89.43
F04	0.54	30.90	0.90	54.24	0.96	72.44	1.60	102.80
F05	0.62	38.57	1.61	69.27	1.24	93.53	2.18	139.83
F06	0.76	48.13	2.27	81.40	1.56	112.83	2.78	160.10
F07	1.40	47.16	2.44	77.92	2.19	96.10	2.94	124.33
F08	1.53	62.07	2.45	105.93	2.96	132.52	4.23	175.90
F09	2.31	71.79	2.99	118.17	3.99	152.38	5.84	201.55
F10	2.72	55.04	3.59	86.65	3.92	103.62	4.84	128.64
F11	3.77	73.11	4.27	116.39	5.62	145.18	7.60	179.50
F12	7.26	89.54	6.76	143.93	8.21	176.48	12.18	221.36
B01	12.90	117.50	14.70	204.67	17.42	280.17	24.15	398.67
B02	22.05	157.83	27.31	285.83	36.32	367.17	56.41	516.17
B03	77.43	174.50	94.09	314.17	109.62	427.83	154.06	596.50
B04	219.05	187.67	241.30	342.17	267.33	443.83	337.36	645.17
B05	483.29	192.00	510.60	347.33	558.47	447.00	648.58	652.00
G01	0.06	9.90	0.12	15.97	0.20	21.00	0.41	30.47
G02	0.18	19.67	0.36	35.23	0.54	45.53	1.15	59.77
G03	0.48	30.57	0.69	50.33	1.11	65.40	2.23	86.63
G04	1.12	40.37	1.43	66.63	1.99	82.60	3.79	113.17
G05	0.52	30.73	0.72	52.77	1.04	64.27	2.11	86.87
G06	0.50	31.77	0.72	52.23	1.05	65.57	1.92	87.57
G07	0.47	29.70	0.70	46.57	1.07	55.70	1.82	72.93
G08	0.46	31.83	0.71	49.17	1.35	64.53	1.95	83.63
G09	1.12	40.73	1.43	65.83	2.15	81.87	3.48	110.50
G10	1.01	40.17	1.39	64.80	1.99	82.47	3.57	110.53
G11	1.00	37.00	1.36	57.80	1.95	72.07	3.31	91.53
G12	0.99	40.50	1.45	66.43	2.56	82.87	3.46	103.53
G13	2.35	51.33	3.02	77.07	4.09	97.10	6.45	125.97
G14	70.45	108.27	79.86	160.30	89.53	196.47	108.37	262.87
G15	411.96	101.83	429.69	153.27	453.41	155.43	534.41	236.27

Table 4: Additional results for groups N08, N10, N11, N12 and F12.

Name	1 commodity			2 commodities			3 commodities			5 commodities						
	t_{avg} (s)	$ Z_{ex} $	$\frac{I_{avg}}{ Z_{ex} }$	t_{avg} (s)	$ Z_{ex} $	$\frac{I_{avg}}{ Z_{ex} }$	t_{avg} (s)	$ Z_{ex} $	$\frac{I_{avg}}{ Z_{ex} }$	t_{avg} (s)	$ Z_{ex} $	$\frac{I_{avg}}{ Z_{ex} }$				
N08	2.96	68.59	267.14	3.89	2.72	123.54	183.68	1.49	3.59	167.25	211.14	1.26	6.46	245.54	274.81	1.12
N10	4.96	72.91	478.95	6.57	4.50	137.08	303.54	2.21	5.04	179.77	279.41	1.55	8.23	251.39	316.22	1.26
N11	6.13	93.81	360.58	3.84	5.88	167.77	259.35	1.55	7.45	228.08	301.88	1.32	13.00	331.46	382.38	1.15
N12	11.22	111.21	415.14	3.73	10.25	200.60	311.40	1.55	11.55	271.10	328.47	1.21	19.45	398.03	446.07	1.12
F12	7.26	89.54	167.5	1.87	6.76	143.93	180.93	1.26	8.21	176.48	197.59	1.12	12.18	221.36	245.52	1.11

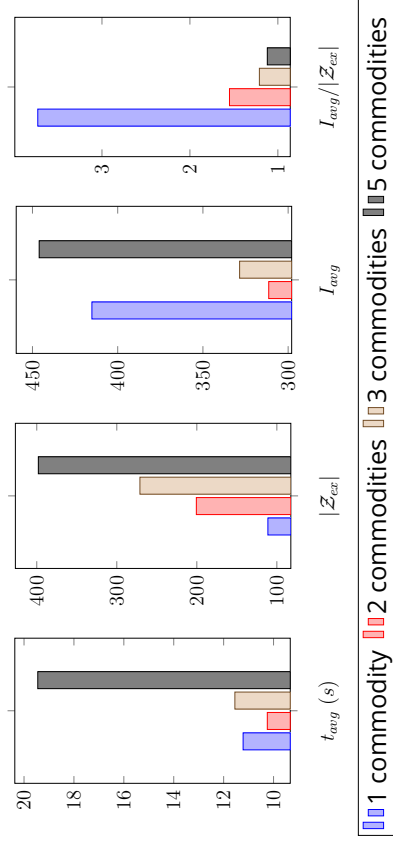


Figure 4: Comparing t_{avg} , $|Z_{ex}|$, I_{avg} and $I_{avg}/|Z_{ex}|$ for N12 with several commodities.

6. Conclusion and future work

In this paper, by integrating Dantzig-Wolfe decomposition with the bi-objective simplex method, we present a new method for solving the *BMCMCF* problem. The method is the first application of ration (Dantzig-Wolfe decomposition) in bi-objective optimisation. The method reformulates the problem into a bi-objective master problem over a set of capacity constraints and several single objective linear fractional sub-problems each over a set of network constraints. After finding an optimal solution of the problem with respect to the first objective, the linear fractional sub-problems generate entering columns for the basis of the master problem to move from one extreme non-dominated point to the next one. We investigate the performance of our method on different sets of bi-objective network instances with several commodities. According to our computational results, the number of commodities has a positive correlation with the average number of non-dominated extreme points and has a strong negative correlation with the ratio of the number of iterations over the number of non-dominated extreme points ($I_{avg}/|\mathcal{Z}_{ex}|$).

This means that increasing the number of commodities does not necessarily increase the average CPU time.

In the literature on Dantzig-Wolfe decomposition for single objective multi-commodity flow problems, it is generally assumed that the problems are structured with a single source and single sink for each commodity [25]. The sub-problems are then shortest path problems. For our *BMCMCF* problem we can also assume we have a single source and single sink for each commodity. In the bi-objective case, the sub-problem (11) involves finding a minimum cost flow in a network with no capacity on arcs and with fractional objective function

$$\frac{(\mathbf{c}^2 - \mathbf{w}^2)^T \mathbf{y}^k - \alpha^{2,k}}{(\mathbf{c}^1 - \mathbf{w}^1)^T \mathbf{y}^k - \alpha^{1,k} - ((\mathbf{c}^2 - \mathbf{w}^2)^T \mathbf{y}^k - \alpha^{2,k})}$$

Due to the fractional objective function, and due to the fact that the column generation sub-problem needs to find variables with reduced cost that are negative for one and positive for the other objective the arising sub-problems are no longer easily solved as shortest path problems. In the future, we will address the application of methods, which are able to exploit the network structure of the sub-problems.

Since the proposed method is based on the bi-objective simplex method, an extension of the approach to three or more objectives may be possible by investigating the multi-objective simplex method. However, finding entering variables to the basis in the multi-objective simplex method requires the solution of single objective linear programmes that do not necessarily have a network structure, see e.g. [10]. Such a potential extension to more than two objectives is also the subject of further research.

Acknowledgement

This research has been partially supported by the Marsden Fund, grant number 9075362506 and by the European Union Seventh Framework Programme (FP7-PEOPLE-2009-IRSES) under grant agreement 246647 and by the New Zealand Government as part of the Optimization and its Applications in Learning and Industry project. The authors would like to express their gratitude to Dr. Jon Pearce for his invaluable contribution to the development of the code.

- [1] A. A. Assad, Multicommodity network flows—a survey, *Networks* 8 (1) (1978) 37–91.
- [2] Y. Aneja, K. Nair, Bicriteria Transportation Problem, *Management Science* 25 (1) (1979) 73–78.
- [3] J. Current, M. Marsh, Multiobjective transportation network design and routing problems: Taxonomy and annotation, *European Journal of Operational Research* 65 (1) (1993) 4–19.
- [4] J. Current, H. Min, Multiobjective design of transportation networks: Taxonomy and annotation, *European Journal of Operational Research* 26 (2) (1986) 187–201.
- [5] S. M. Lee, M. J. Schniederjans, A Multicriteria Assignment Problem: A Goal Programming Approach, *Interfaces* 13 (4) (1983) 75–81.
- [6] L. J. Moore, B. W. Taylor III, S. M. Lee, Analysis of a transshipment problem with multiple conflicting objectives, *Computers & Operations Research* 5 (1) (1978) 39–46.

- [7] S. M. Lee, G. I. Green, C. S. Kim, A multiple criteria model for the location-allocation problem, *Computers & Operations Research* 8 (1) (1981) 1–8.
- [8] E. L. Ulungu, J. Teghem, Multi-objective combinatorial optimization problems: A survey, *Journal of Multi-Criteria Decision Analysis* 3 (2) (1994) 83–104.
- [9] S. Moradi, A. Raith, M. Ehrgott, The Linear Bi-Objective Multi-Commodity Minimum Cost Flow Problem, in: *Proceedings of 46th Annual ORSNZ Conference*, Wellington, 238–247, John Haywood (Ed.), 2012.
- [10] M. Ehrgott, *Multicriteria Optimization*, Springer-Verlag, Berlin, Heidelberg, 2005.
- [11] S. Moradi, A. Raith, M. Ehrgott, A Bi-Objective Decomposition Method for Solving the Bi-Objective Multi-Commodity Minimum Cost Flow Problem, *Proceedings of the 47th Annual Conference of the ORSNZ*, Hamilton, 2013.
- [12] H. Hamacher, C. Pedersen, S. Ruzika, Multiple objective minimum cost flow problems: A review, *European Journal of Operational Research* 176 (3) (2007) 1404–1422.
- [13] A. Sedeño-Noda, C. González-Martín, S. Alonso-Rodríguez, The biobjective undirected two-commodity minimum cost flow problem, *European Journal of Operational Research* 164 (1) (2005) 89–103.
- [14] A. Eusébio, J. Figueira, M. Ehrgott, A primal dual simplex algorithm for bi-objective network flow problems, *4OR* 7 (3) (2008) 255–273.
- [15] M. Ehrgott, J. Puerto, A. Rodríguez-Chía, Primal dual Simplex Method for Multiobjective Linear Programming, *Journal of Optimization Theory and Applications* 134 (3) (2007) 483–497.
- [16] A. Raith, M. Ehrgott, A two-phase algorithm for the biobjective integer minimum cost flow problem, *Computers & Operations Research* 36 (6) (2009) 1945–1954.

- [17] A. Przybylski, X. Gandibleux, M. Ehrgott, The two-phase method for multiobjective combinatorial optimization problems, in: A. R. Mahjoub (Ed.), *Progress in Combinatorial Optimization*, ISTE Wiley, London, 559–596, 2011.
- [18] A. Sedeño-Noda, C. González-Martín, The biobjective minimum cost flow problem, *European Journal of Operational Research* 124 (2000) 591–600.
- [19] H. Hamacher, A note on K best network flows, *Annals of Operations Research* 57 (1995) 65–72.
- [20] A. Eusébio, J. Figueira, On the computation of all supported efficient solutions in multi-objective integer network flow problems, *European Journal of Operational Research* 199 (1) (2009) 68–76.
- [21] R. Ahuja, T. Magnanti, J. Orlin, *Network Flows: Theory, Algorithms, and Applications*, Prentice Hall, Englewood Cliffs, 1993.
- [22] A. Eusébio, J. Figueira, Finding non-dominated solutions in bi-objective integer network flow problems, *Computers & Operations Research* 36 (9) (2009) 2554–2564.
- [23] V. Chankong, Y. Y. Haimes, *Multiobjective decision making: theory and methodology*, 8, North-Holland, 1983.
- [24] H. Isermann, Proper Efficiency and the Linear Vector Maximum Problem, *Operations Research* 22 (1) (1974) 189–191.
- [25] J. A. Tomlin, Minimum-Cost Multicommodity Network Flows, *Operations Research* 14 (1) (1966) 45–51.
- [26] G. Gallo, S. Pallottino, Shortest path algorithms, *Annals of Operations Research* 13 (1) (1988) 1–79.
- [27] A. Charnes, W. W. Cooper, Programming with linear fractional functionals, *Naval Research Logistics Quarterly* 9 (2) (1962) 181–186.
- [28] Gurobi Optimization Inc., *Gurobi Optimizer Reference Manual*, URL <http://www.gurobi.com>, 2014.

- [29] D. Klingman, A. Napier, J. Stutz, NETGEN: A Program for Generating Large Scale Capacitated Assignment, Transportation, and Minimum Cost Flow Network Problems, *Management Science* 20 (5) (1974) 814–821.