Multiple Objective Minimum Cost Flow Problems: A Review

Horst W. Hamacher[*], Christian Roed Pedersen[†], Stefan Ruzika[*]

18th February 2005

Abstract

In this paper, theory and algorithms for solving the multiple objective minimum cost flow problem are reviewed. For both the continuous and integer case exact and approximation algorithms are presented. In addition, a section on compromise solutions summarizes corresponding results. The reference list consists of all papers known to the authors which deal with the multiple objective minimum cost flow problem.

Keywords: Multiple objective optimization; network flows; efficient solution; representative systems.

[*] Fachbereich Mathematik, Universität Kaiserslautern, Kaiserslautern, Germany; E-mail: {hamacher, ruzika}@mathematik.uni-kl.de. The research has been partially supported by Deutsche Forschungsgemeinschaft (DFG) grant HA 1737/7 "Algorithmik großer und komplexer Netzwerke" and by New Zealand's Julius von Haast award.

[†] Department of Operations Research, University of Aarhus, Aarhus, Denmark; E-Mail: roed@imf.au.dk.

1 Introduction

In this paper we review papers which simultaneously cover two important modeling tools.

This is on one hand the class of *network flow problems*, which for several reasons are among the most studied and applied operations research models. They can be used to tackle a large set of real-world problems and occur as important building blocks and subproblems in complex models. Furthermore, they possess nice theoretical properties such as the integrality of its basic solutions. Network flow problems are known to be solvable by special algorithms which deal even with large scale problems in moderate computing time.

On the other hand we consider problems with more than one objective. As in the case of network flow problems, these *multiple objective optimization problems* are very important as modeling tool, since very few real-world problems have only a single objective to be taken into account. In general, these objectives will be in conflict, such that there is no solution which simultaneously optimizes all objectives. Therefore one is generally interested to find solutions which have the property that none of the objectives can be improved without worsening one of the other. Finding all or a suitable subset of these *Pareto* or *efficient solutions* is the goal of multiple objective optimization.

Given the importance and the wealth of literature which exists for both of these problem classes it is surprising that relatively few authors have worked on combining them to consider *multiple objective network flow problems.* But network flow problems are inherently multiple objective in nature. Applications within transportation planning faces conflicting criteria like minimization of cost for selected routes, minimization of arrival times at the destination points, minimization of deterioration of goods, maximization of safety, etc. Since these applications typically require flow values to be integer, one also has a prominent example showing the need to deal with integer multiple objective optimization problems. Network flow problems are obviously a very good starting point for this research, since the network flow feasibility polyhedron is known to have the integrality property.

To initiate more research in this - in our opinion - very important but neglected area, we review in this paper theoretical results and solution algorithms for the class of multiple objective minimum cost flow problems. We have chosen not to consider any of the various special cases of the minimum cost network flow problem (shortest path, assignment, transportation problem, etc.). All of these subclasses have received considerable attention in the operations research or - more specific - in the network design and routing literature and deserve their own reviews, since there exist various algorithms exploiting the special structure of those problems. For excellent existing reviews we refer the interested reader, for instance, to [CM86, CM93, UT94, Skr00]. Although some of these survey papers include sections on the general multiple objective minimum cost flow problem, we consider it necessary to deal with this subject by itself to be able to elaborate more extensively on some details which could be of interest to the reader.

We tried our best to include all relevant literature which is available as publication in a journal or as report via internet. But we apologize in advance to any author whose paper we left out unintendedly.

The paper is organized as follows. In Section 2 we briefly introduce the standard notation from multiple objective programming and network flow theory. A review of exact and approximate methods for the continuous multiple objective minimum cost flow problem is given in Section 3. Section 4 contains a discussion of the same topics for the case of integer valued flows. In Section 5 we summarize the results on compromise solutions. In Section 6 we summarize all results in tabular form and conclude the paper by pointing out some current lines of research.

2 Multiple Objective Linear Programs and Network Flows

The goal of this section is to introduce terminology and basic results well-known from multiple objective linear programming and network flow theory. For a more thorough introduction to these topics appropriate references are Steuer [Ste86], Ehrgott [Ehr00] and Ahuja, Magnanti, and Orlin [AMO93].

2.1 Multiple Objective Linear Programs

A *multiple objective linear program* (MOLP) is given by

$$\begin{aligned} \min\ & Cx \\ \text{s.t.}\ & x \in \mathcal{X} \end{aligned} \tag{$MOLP$}$$

where $C = (c^1, \ldots, c^p)^T$ with rows $c^1, \ldots, c^p$ denotes a $p \times m$ linear objective matrix, $x \in \mathbb{R}^m$ the vector of variables, and $\mathcal{X}$ the set of feasible solutions, subsequently called the *decision space.* For MOLPs, this set is a polyhedron, i.e.,

$$\mathcal{X} := \mathcal{P} := \{x \in \mathbb{R}^m : Ax = b,\ x \geq 0\}$$

where A is an $n \times m$ constraint matrix and $b \in \mathbb{R}^n$ the right-hand side vector. We refer to

$$\mathcal{Y} := C\mathcal{X} := \{y := Cx \in \mathbb{R}^p : x \in \mathcal{X}\}.$$

as the *objective space.* By adding integrality requirements to the variables in $\mathcal{X}$ we obtain a *multiple objective linear integer program (MOLIP).* For both MOLP and MOLIP we assume that the objective functions are conflicting, that is, we exclude the existence of an *ideal solution* $x \in \mathcal{X}$ which minimizes all p objectives simultaneously.

Minimizing a vector-valued objective function needs some more explanation since there is no canonical ordering defined in $\mathbb{R}^p$ for $p \geq 2$. Throughout this article we use the Pareto concept of optimality for MOLPs and MOLIPs which is based on the following two binary relations $<$ and $\leq$. Let $y^1, y^2 \in \mathbb{R}^p$. Then

$$\begin{aligned} y^1 \leq y^2 \quad &:\Leftrightarrow \quad y^1_k \leqq y^2_k \quad \forall\, k = 1, \ldots, p \quad \textit{and} \quad y^1 \neq y^2 \\ y^1 < y^2 \quad &:\Leftrightarrow \quad y^1_k < y^2_k \quad \forall\, k = 1, \ldots, p. \end{aligned}$$

A point $y^2 \in \mathbb{R}^p$ is called *dominated* by $y^1 \in \mathbb{R}^p$ if $y^1 \leq y^2$. The *Pareto cone* is defined as $\mathbb{R}^p_{\geqq} := \{y \in \mathbb{R}^p : y \geqq 0\}$.

According to the Pareto concept of optimality, the *efficient* or *Pareto set*, $\mathcal{X}_E$, and the *weakly efficient* or *weakly Pareto set*, $\mathcal{X}_{wE}$, are defined as

$$\begin{aligned} \mathcal{X}_E \quad &:= \quad \{x \in \mathcal{X} : \text{there exists no } \bar{x} \in \mathcal{X} : C\bar{x} \leq Cx\} \\ \mathcal{X}_{wE} \quad &:= \quad \{x \in \mathcal{X} : \text{there exists no } \bar{x} \in \mathcal{X} : C\bar{x} < Cx\}. \end{aligned}$$

The images of these sets under the vector-valued linear mapping C

$$\mathcal{Y}_N := C\mathcal{X}_E \quad \text{and} \quad \mathcal{Y}_{wN} := C\mathcal{X}_{wE}$$

are called the *nondominated set* and the *weakly nondominated set*, respectively. In Figure 1 examples of nondominated and dominated objective vectors are indicated by crosses and dots, respectively. Following the terminology of Steuer [Ste86] we use the denotation

$$Z^{\geq} := \textit{conv}(\mathcal{Y}_N) + \mathbb{R}^p_{\geqq}$$

(the shaded grey area of Figure 1) where *conv* is the convex hull operator. For MOLP and MOLIP the *efficient frontier* is defined as the set $\{y \in \textit{conv}(\mathcal{Y}_N) : \textit{conv}(\mathcal{Y}_N) \cap (y + (-\mathbb{R}^p_{\geqq})) = y\}$. For MOLP the efficient frontier is identical with the set $\mathcal{Y}_N$. In the case of $p = 2$ objectives, the efficient frontier of MOLP is known to be piecewise linear and convex. Its breakpoints are the *extreme nondominated points* which are images of *extreme efficient solutions* in the decision space. For MOLIP the efficient frontier is the nondominated set of its continuous relaxation. If a nondominated objective vector is on the efficient frontier it is called a *supported* nondominated objective vector. Otherwise it is an *unsupported* nondominated objective vector. We shall denote the corresponding solutions in decision space *supported efficient solutions* and *unsupported efficient solutions*, respectively. It is important to notice that for MOLP only supported efficient solutions exist, whereas for MOLIP unsupported efficient solutions may exist even if the constraint matrix for the considered MOLIP instance is totally unimodular.

It is well known, that each supported efficient solution for MOLP can be found as an optimal solution to the *weighted sum problem*, $\min\{\lambda Cx : x \in \mathcal{X}\}$, for a certain $\lambda \in]0, 1[^p$ (Geoffrion [Geo68] and Isermann [Ise74]).

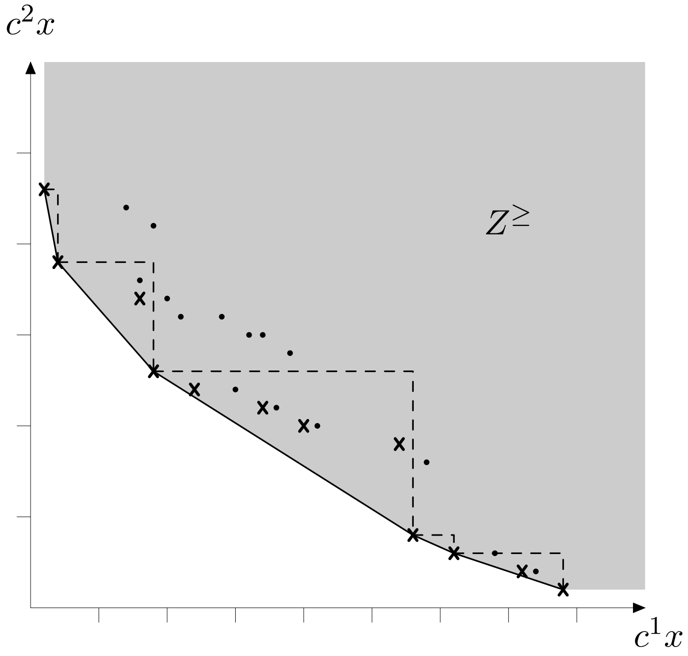

Figure 1: $Z^{\geq}$ for a biobjective linear integer program with nondominated (cross) and dominated (dot) objective vectors. Unsupported nondominated objective vectors are located in its interior.

Two different notions of connectivity based on topology and graph theory, respectively, are used in the context of multiple objective programming.

We call a set S *topologically connected* if there does not exist nonempty open sets S_1 and S_2 such that $S \subseteq S_1 \cup S_2$ and $S_1 \cap S_2 = \emptyset$. For MOLP the efficient set $\mathcal{X}_E$ and the efficient frontier $\mathcal{Y}_N$ are topologically connected as shown by Naccache [Nac78] and Warburton [War83], respectively. In contrast, neither $\mathcal{X}_E$ nor $\mathcal{Y}_N$ are topologically connected for MOLIP.

Let $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ denote the *adjacency graph of MOLP* where $\mathcal{N}$ is the set of efficient basic feasible solutions. An edge between two nodes of $\mathcal{N}$ is included in $\mathcal{A}$ if and only if the corresponding efficient basic feasible solutions can be obtained from each other by a single pivot operation. Isermann [Ise79] showed that the adjacency graph $\mathcal{G}$ of MOLP is connected. This graph theoretical connectivity property of the set of all basic solutions in $\mathcal{X}_E$ makes it possible to find the entire efficient and nondominated sets by simple pivot exchange arguments for MOLPs (see Figure 2).

The same figure illustrates that the adjacency graph is, in general, not connected for the integer version MOLIP. Consequently, it is not possible for MOLIPs to generate the efficient set by "traveling" across the adjacency graph only. This result was shown in the special case of biobjective shortest path problems by Przybylski, Gandibleux, and Ehrgott [PGE05]. Obviously, it is depending on the definition of the adjacency graph in the integer case. Other definitions of the node and edge set may salvage connectivity. Investigations on suitable definitions and the algorithmic consequences are currently under research.

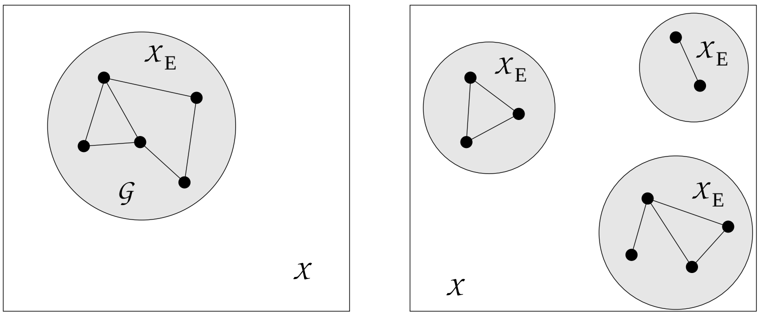

Figure 2: Illustration of adjacency graph: connectedness for MOLP (left) and disconnectedness for MOLIP (right).

2.2 Minimum Cost Flow Problem

Consider a directed network $G = (N, A)$ with node set N and arc set A. Let $n = |N|$ and $m = |A|$. Each arc $(i, j) \in A$ is associated with a nonnegative lower and a positive upper bound capacity referred to as l_{ij} and u_{ij}, respectively. Let $b \in \mathbb{Z}^n$ be a vector of demand (if $b_i < 0$, $i \in N$) and supply (if $b_i > 0$, $i \in N$) satisfying $\sum_{i \in N} b_i = 0$. If $b_i = 0$ for some $i \in N$, node i is a transshipment node. A function $x : A \to \mathbb{R}$ is called a *(network) flow* if it satisfies the *flow conservation constraints*

$$\sum_{\{j:(i,j)\in A\}} x_{ij} - \sum_{\{j:(j,i)\in A\}} x_{ji} = b_i \quad \text{for all } i \in N,$$

and the *capacity constraints*

$$l_{ij} \leq x_{ij} \leq u_{ij} \quad \text{for all } (i, j) \in A.$$

The set of all flows satisfying the flow and the capacity constraints is the *flow polyhedron*, denoted $\mathcal{P}_{flow}$. Results from classic network flow theory are not part of this survey. In particular, we will freely apply well-known results as the relation between basic flows and trees, the optimality criterion using negative cycles and the resulting cycle cancelling algorithm, shortest augmenting paths and path algorithms, the out-of-kilter procedure etc.

The *multiple objective minimum cost flow problem (MMCF)* can be concisely stated as the following mathematical program:

$$\min\{(c^1, \ldots, c^p)^T x : \quad x \in \mathcal{P}_{flow}\} \tag{$MMCF$}$$

where $c^1, \ldots, c^p \in \mathbb{R}^m$ are nonnegative integer cost vectors. Note that the objective function $Cx = (c^1, \ldots, c^p)^T x$ is composed of p linear functions whereas the decision space remains the same as for the classic minimum cost flow problem, subsequently denoted MCF. Obviously, MCF is a special case of MMCF (with $p = 1$) and MMCF is a special case of MOLP. Another special case which is addressed in most papers on MMCF is the *biobjective minimum cost flow problem (BMCF)*. MMCF is in general a continuous problem, i.e., the flows x_{ij} may take on fractional values. If we want to enforce integrality we solve the *multiple objective minimum cost integer flow problem (MMCIF)*

$$\begin{aligned} &\min\{(c^1,\ldots,c^p)^T x: \quad x \in \mathcal{X}_{flow}\} \\ &\quad \text{where} \quad \mathcal{X}_{flow} := \mathcal{P}_{flow} \cap \mathbb{Z}^m. \end{aligned} \tag{$MMCIF$}$$

Both MMCF and MMCIF become easy if the componentwise ordering is replaced by the *lexicographical ordering.* In the latter, two vectors are compared by looking at the first component in which they differ. The order of this component then defines the lexicographical order of the vectors. Virtually all algorithms for the classic minimum cost flow problem can be carried over to the lexicographical minimum cost flow problem (see, for instance, in a more general context, Hamacher [Ham80] or Calvete and Mateo [CM95]).

For the bicriteria minimum cost flow problem Ruhe [Ruh88a] uses a pathological graph introduced by Zadeh [Zad73] to show that there can be an exponential number of extreme nondominated objective vectors on the frontier, which makes even the continuous case *intractable* in general. Furthermore, BMCF is known to be a generalization of the bicriteria shortest path problem (BSP) which is $\mathcal{NP}$-hard and has a $\#\mathcal{P}$-complete decision problem. These results obviously carry over to MMCF. More on general complexity results can be found in the book of Ehrgott [Ehr00].

3 Continuous Multiple Objective Minimum Cost Flow Problem

Similar to the situation in linear programming, there are many more papers on the continuous multiple objective minimum cost flow problem MMCF than on its integer counterpart. An obvious reason is that some of the techniques of MOLP can just be specialized to MMCF. In the first section papers are considered which aim at computing all efficient solutions. Methods for finding a representative system for the efficient set are discussed in the second section.

3.1 Exact Methods

The papers reviewed in this section present algorithms for finding the entire set of efficient solutions of BMCF - either in decision space or in objective space ([MP84, LP91, PHL92, SNGM00, SNGM03]). We found no papers on an exact solution method for MMCF with more than two objectives. All but one paper ([SNGM03]) exploit the topological connectivity of the set of efficient solutions.

Since MMCF is a special case of MOLP, the graph theoretical connectivity result of

Isermann [Ise79] holds as well. Basic feasible solutions of MMCF (subsequently called *basic flows*) correspond to spanning trees ([AMO93]) and a pivot operation corresponds to the insertion of a non-tree edge into the tree and the deletion of an edge in the tree. Hence, two basic feasible flows are adjacent if their tree representations have $n-2$ arcs in common. Due to the graph theoretical connectivity property for MMCF, one can start with an arbitrary efficient basic feasible flow and obtain the entire efficient frontier by iteratively performing network pivot operations, i.e., by exchanging one arc in trees at a time.

Entering a nonbasic arc (u,v) in the tree representation of an efficient basic flow x^k yields a unique cycle W along which δ_{uv} units of flow can be send before a new basic feasible flow x^l is obtained. Note $\delta_{uv} = \min\{r_{ij} : (i,j) \in W\}$ equals the minimal capacity of the arcs in W in the incremental graph $G(x^k)$. The only flow change between x^k and x^l is on the arcs of this unique cycle. Only the introduction of nonbasic arcs that have out-of-kilter status with respect to at least one of the criteria can lead to other efficient flows ([Ehr99]). If two adjacent basic feasible flows x^k and x^l are both efficient, then so are all of their convex combinations. Any such convex combination can be found for $\delta_{uv} > 0$ by sending a suitable amount of flow $0 \leq \Delta < \delta_{uv}$ through the cycle. In terms of polyhedral structure of MMCF this corresponds to moving along the *efficient edge* connecting x^k and x^l in $\mathcal{P}_{flow}$ ([LP91]).

Malhotra and Puri [MP84] provide a generalization of the out-of-kilter method to solve BMCF with a uniform capacity for all arcs, i.e., $u_{ij} = u$ for all arcs $(i,j) \in A$. As the authors state, this idea can be modified to address general BMCF. The efficient frontier is built in a left-to-right fashion, starting with the lexicographical minimum for the first objective (see Figure 1). Due to the definition of a lexicographical minimal flow, all arcs are in-kilter with respect to the first objective. But, since by our general assumption BMCF does not have an ideal point some arcs are out-of-kilter for the second objective. The algorithm strives for an efficient flow with all arcs being in-kilter for the second objective, i.e., a lexicographical minimal flow for the second objective. Therefore, for a given flow the arcs being out-of-kilter for the second objective (and in-kilter for the first objective) are said to be *eligible arcs.* For each eligible arc Pareto cycles including this arc are found in the incremental graph of the current flow and added to the current flow. The obtained flows are argued to be efficient points on the frontier of BMCF. This argumentation can easily be seen to be wrong and in fact not only the generation of dominated objective vectors but also possible oversights of extreme points on the efficient frontier might occur. Even for the small network given by Malhotra and Puri themselves it can be seen that an extreme nondominated objective vector exists, which is not found by the proposed algorithm. The incorrect output of the algorithm is partly due to the fact that only one set of dual node variables (node prices) is introduced.

This mistake is corrected by - among others - Lee and Pulat [LP91], Pulat et al.

[PHL92] and Sedeño-Noda and González-Martín [SNGM00] who use one set of dual variables for each objective. These papers are based on the following idea of Gass and Saaty [GS55] and its generalization by Geoffrion [Geo67]. BMCF is formulated as the parametric programming problem $P(\lambda)$:

$$\begin{aligned} &\min \; f(x) = \left(c^1 + \lambda c^2\right) x \\ &\text{s.t.} \;\; x \in \mathcal{P}_{flow} \end{aligned} \tag{$P(\lambda)$}$$

where $0 \leq \lambda \leq \infty$ (or $\delta \leq \lambda \leq \phi$ where δ and ϕ are suitably chosen lower and upper bounds). It is well known, that all optimal solutions for $P(\lambda)$ are efficient solutions of BMCF, and that conversely all efficient solutions of BMCF can be found as optimal solutions of some $P(\lambda)$ problem [Geo67].

Similar to [MP84], the algorithm by **Lee and Pulat** [LP91] implements a revised version of the out-of-kilter method. Initially uniform weights are placed on both objectives and the resulting single objective minimum cost flow problem, $\min\{(c^1+c^2)x : x \in \mathcal{P}_{flow}\}$ is solved by the out-of-kilter method after which the flow is adjusted to be basic by a rerouting procedure or node price adjustment procedure. Then from this compromise solution the frontier is searched to the left by considering arcs that are out-of-kilter with respect to the first objective and to the right by considering arcs that are out-of-kilter for the second objective. To perform the move from one basic flow to another the labeling procedure from the single criterion out-of-kilter method is modified and a procedure for ensuring the attainment of a basic feasible flow is applied. The arc entering the basis is chosen upon a determination of the smallest ratio between reduced costs for the two criteria. This corresponds to choosing an arc that results in the steepest slope of the line between two consecutive points on the efficient frontier. Since it is mistakenly believed that efficient basic feasible flows necessarily yield extreme nondominated objective vectors, the authors neglect to focus on the case of multiple nonbasic arcs having the same minimal value of the ratio of the two reduced costs. Consequently, possible nonextreme nondominated objective vectors are generated.

The algorithm of **Pulat, Huarng, and Lee** [PHL92] is conceptually the same as the one by Malhotra and Puri, since it again utilizes adjacency results for searching the frontier in a left-to-right fashion. Apart from the addition of a second set of node prices another distinction is that Pulat et al. use the network simplex method to solve BMCF in its parametric programming formulation $P(\lambda)$. Pulat et al. realize that some efficient basic feasible flows may not correspond to extreme nondominated objective vectors. However, since the authors aim at a complete description of all efficient basic feasible flows, in case of several arcs yielding a minimal ratio of the reduced costs, all of them are introduced in the current basic feasible flow. As noted above, all nonbasic efficient flows can be found by sending flow in the unique cycle between any two adjacent efficient basic feasible flows. Hence, a complete description of all efficient basic feasible flows is required to find an explicit expression for the entire efficient set.

Sedeño-Noda and González-Martín [SNGM00] build on the ideas of the three preceding papers. $P(\lambda)$ is solved in a left-to-right fashion using the network simplex method for the single criterion optimizations. The desired output of the algorithm is the set of extreme nondominated objective vectors. In each iteration from a list S of arcs yielding the minimal ratio of the reduced costs one arc is chosen to enter the basic tree of the current efficient basic feasible flow x. This will result in a new efficient basic feasible flow, $\hat{x}$ which corresponds either to an extreme nondominated objective vector or a nonextreme nondominated objective vector. The former case is identified if all other arcs in S fulfill the optimality condition of the second objective, and in this case the obtained point is stored. In the latter case, arcs violating the optimality condition for the second objective still exist in S, and one of these is introduced into the basic tree of $\hat{x}$ yielding a new efficient basic feasible flow. If the list S is empty, an extreme nondominated objective vector has been identified, and a new list of arcs is built. The algorithm stops, if no more arcs are out-of-kilter with respect to the second objective.

The papers reviewed above all exploit the topological connectivity of the efficient set. In contrast, **Sedeño-Noda and González-Martín** [SNGM03] modify a method proposed by Aneja and Nair [AN79], which iteratively applies the weighted sum problem. The method was originally designed for the bicriteria transportation problem. The method by Sedeño-Noda and González-Martín starts by finding the two lexicographical minima. For each two identified consecutive points on the frontier a search for additional extreme nondominated objective vectors in the triangle between these points and the corresponding local ideal point $(c^{1^\star}, c^{2^\star})$, is performed using Lagrangian theory and the network simplex method. The algorithm finds the extreme point closest to the line between the ideal point and some appropriately chosen aspiration level vector z with $z_k > c^{k^\star}$, $k = 1, 2$. Whenever a new point is identified two new areas are introduced for future search. The output of the procedure is the set of all extreme nondominated objective vectors.

Four of the five described algorithms have been implemented, all using NETGEN [KNS74] as problem generator. In fact, three of the algorithms have been implemented by Sedeño-Noda and González-Martín on the same sample sets making a direct comparison possible [LP91, SNGM00, SNGM03]. In total 180 networks (five replications of 36 setups) with the number of nodes ranging between 25 and 40, the number of arcs ranging from 100 to 400 and the cardinality of maximum capacity on the arcs ranging from 10 to 100000 were considered. For these examples, the method by Sedeño-Noda and González-Martín [SNGM00] proved superior in all instances. The method by Sedeño-Noda and González-Martín [SNGM03] performs better than the method by Lee and Pulat [LP91] for sparse networks. Pulat et al. [PHL92] test their algorithm on 90 networks grouped in three different setups with the number of nodes ranging from 10 to 100 and the number of arcs ranging from 30 to 500.

3.2 Approximate Methods

As mentioned in Section 2.2, Ruhe [Ruh88a] shows that the exact computation of the efficient frontier is, in general, intractable, since there may exist an exponential number of extreme nondominated objective vectors. Therefore, several approaches are proposed to find representative subsets of the efficient frontier. All these approaches aim at the development of computationally appealing algorithms. The literature following this thought can be divided into two groups: The first group argues for the computation of a limited number of points leading to a final preferred solution in reasonable computation time. We have chosen to present this group in Section 5 along with the algorithms finding compromise solutions for MMCIF. The second group approximates the frontier as a whole with provable good quality. All approaches in the second group are applicable to BMCF only and utilize sets L and U which "sandwich" the efficient frontier (see Figure 3), i.e., $\mathcal{Y}_N \subseteq \big((L + \mathbb{R}^2_{\geqq}) \cap (U + (-\mathbb{R}^2_{\geqq}))\big)$.

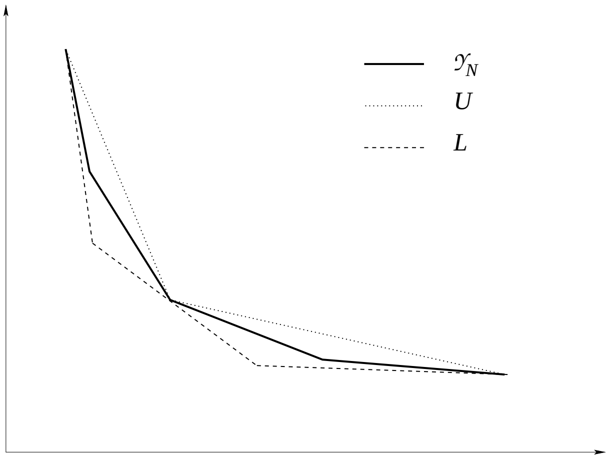

Figure 3: Illustration of sandwich idea.

All articles that use the sandwich idea for BMCF follow the algorithm of **Burkard, Hamacher, and Rote** [BHR91] for approximating univariate convex functions $f : \mathbb{R} \rightarrow [a, b]$. We therefore present next this idea in its general framework.

It is assumed that for any $t \in [a, b]$ the left and right derivatives $f^-(t)$ and $f^+(t)$ are obtainable. Let $a = t_1 < t_2 \cdots < t_n = b$ be a finite partition of the interval $[a, b]$. We refer to t_i, $i = 1, \ldots, n$, as breakpoints. Two piecewise linear functions $l(t)$ and $u(t)$ approximate $f(t)$ from above and from below, where

$$u(t) := f(t_i) + \frac{f(t_{i+1}) - f(t_i)}{t_{i+1} - t_i}(t - t_i) \tag{1}$$

$$l(t) := \max\{f(t_i) + f^+(t_i)(t - t_i),\ f(t_{i+1}) + f^-(t_{i+1})(t - t_{i+1})\} \tag{2}$$

for $t_i \leq t \leq t_{i+1},\ i = 1, 2, \ldots, n-1$. At any stage of the algorithm, $l(t)$ and $u(t)$ satisfy

$l(t) \leq f(t) \leq u(t)$ for all $t \in [a,b]$.

The error of the current approximation is measured in [BHR91] by $\max\{u(t) - l(t) : t \in [a,b]\}$. Let t_l and t_r be two consecutive breakpoints in the current partition satisfying $t^* \in [t_l, t_r]$ where $t^* = \arg\max\{u(t) - l(t) : t \in [a,b]\}$. Thus, $[t_l, t_r]$ is an interval with the largest error. A new breakpoint $t_{new} := \frac{t_r - t_l}{2}$ is added to the current, finite partition of the interval $[a,b]$ and the value $f(t_{new})$ is found. This choice of the new breakpoint is referred to as the *interval bisection rule.* The approximating functions $l(t)$ and $u(t)$ are updated due to Equations (1) and (2). This scheme is iteratively repeated until the error of the approximation falls below a prescribed value. It is shown that the approximating functions $l(t)$ and $u(t)$ converge uniformly to $f(t)$. Furthermore, the error of the approximation decreases quadratically with the number of breakpoints and, given an accuracy level, an upper bound on the number of breakpoints needed for satisfying this accuracy level can be obtained.

When transferring the interval bisection rule to BMCF, one objective function is treated as the independent variable t. The other objective function plays the role of $f(t)$. In other words, one objective function is considered a convex function of the other objective function. Note that evaluating $f(t_{new})$ for some given t_{new} corresponds to minimizing one objective function subject to a fixed value of the other objective function (which is known as the ε-constraint method). In terms of BMCF this means solving MCFs with one additional side constraint.

Ruhe [Ruh88b] (see also the monography [Ruh91] of Ruhe) introduces the Hausdorff distance between the lower and upper approximation to measure the error of the approximation. For $L := \{(t, l(t)) \mid t \in [a,b]\}$ and $U := \{(t, u(t)) \mid t \in [a,b]\}$ the Hausdorff distance between L and U is

$$d(L,U) := \max\big\{\sup_{x\in L}\inf_{y\in U} ||y-x||_2, \sup_{y\in U}\inf_{x\in L} ||y-x||_2\big\}.$$

In contrast to the error measure in [BHR91], the Hausdorff distance is invariant under rotation and does not favor one objective function over the other.

In Ruhe [Ruh88b] new breakpoints are generated by applying the so-called *chord rule*, which was originally introduced by Aneja and Nair [AN79] in the context of computing the efficient extreme points of a bicriteria transportation problem. Given an interval $[t_l, t_r]$ with the maximal error, the new breakpoint is computed by $t_{new} := \arg\min(f(t) - k \cdot t)$ where k is the slope of the upper approximating function $u(t)$, $t \in [t_l, t_r]$.

Using the chord rule for the approximation of $\mathcal{Y}_N$ for BMCF, a weighted sum problem (which corresponds to solving MCF) has to be solved in each iteration which is in general easier than solving an ε-constraint problem.

Fruhwirth, Burkard, and Rote [FBR89] introduce two new rules, the *angle bisection* and the *slope bisection rule* for generating breakpoints. As in the chord rule, the new breakpoint is $t_{new} := \arg\min(f(t) - k \cdot t)$, but the rules vary in the choice of the parameter k. For the angle bisection rule, k equals the slope of the bisector of the outer angle of the triangle formed by the graphs $(t, u(t))$ and $(t, l(t))$, $t \in [t_l, t_r]$. In case of the slope bisection rule, k is the mean of the slopes of the two piecewise linear functions that determine $u(t)$ in $[t_l, t_r]$.

For chord, angle bisection and slope bisection rule, the error decreases quadratically with the number of breakpoints. Consequently, an upper bound on the number of MCF evaluations can be derived to obtain a given accuracy level when applying any of the three rules (see [Ruh88b] and [FBR89]).

Fruhwirth, Burkard, and Rote [FBR89], Burkard, Rote, Ruhe, and Sieber [BRRS89], and Ruhe [Ruh91] report on numerical studies comparing different partition rules. The chord rule and the angle bisection rule are compared with a special implementation of the angle bisection rule which exploits the bound on the number of MCFs that have to be solved to construct an approximation with a desired accuracy level. A primal network simplex algorithm is employed to solve the occuring MCFs. Nine network instances were generated with $n = 100, 400, 800$ and $m = 10n$. Capacities and costs for the arcs were taken independently at random from the intervals $[10, 500]$ and $[1, 500]$, respectively. The special implementation of the angle bisection rule outperforms the angle bisection as well as the chord rule in terms of memory and time consumption.

A derivative-free modification of the sandwich approximation approach was proposed by **Yang and Goh** [YG97]. For each interval $[t_i, t_{i+1}]$, the upper approximating function is computed as in Equation (1). The lower approximating function consists of a piecewise linear function that is parallel to the upper approximation. New breakpoints are computed with the chord rule. The algorithm is applied to bicriteria quadratic minimum cost flow problems.

In **Ruhe and Fruhwirth** [RF90] the sandwich algorithm is used in order to compute an ε-optimal approximation for BMCF. Here, a subset $S \subset \mathcal{P}_{flow}$ is called ε-optimal if for all $x \in \mathcal{P}_{flow}$ there is a solution $\hat{x} \in S$ such that $c^i\hat{x} \leq (1 + \varepsilon)c^i x$ for $i = 1, 2$. In their pseudo-polynomial time algorithm Ruhe and Fruhwirth modify the rule for determining additional breakpoints. Instead of solving only one MCF as in all previous sandwich algorithms, two MCFs are solved in each iteration. Two general questions are addressed: Given a value ε, an ε-optimal set of small cardinality is determined, and, given the cardinality of S, an ε-optimal set having a high level of accuracy is computed. Ruhe and Fruhwirth compare two realizations of their algorithm numerically. Networks are generated with $n = 600$ and $m = 6000,\ 9000,\ 12000$ and $n = 900$ and $m = 9000,\ 18000$, respectively. The results are averaged over twenty instances for each of the five network sizes. Capacities and costs are chosen randomly distributed in $[500, 5000]$ and $[1, 1500]$, respectively. The numerical study shows that already for relatively small sized approximating sets the accuracy level is quite high.

4 Integer Multiple Objective Minimum Cost Flow Problem

All approaches for finding all integer efficient solutions of MMCIF address problems with two objective functions only and comprise two phases:

Phase 1: Find all supported integer efficient flows.

Phase 2: Find all unsupported integer efficient flows.

Some of the articles concentrate on Phase 1 only. These approaches can be considered as approximations of the integer nondominated set. Analogously to the structure in Section 3, these approximate approaches are discussed in Section 4.2, while the exact approaches are discusssed in Section 4.1. When presenting the latter group of papers, we, therefore, focus on Phase 2 only assuming that Phase 1 has already been solved. Several general ideas are employed to implement Phase 2. Firstly, structural results are used such as the fact that each efficient flow is an extreme efficient solution of BMCIF with modified capacities. Secondly, it is observed that consecutive supported solutions on the efficient frontier define "triangles of interest" restricting the search area for undiscovered nondominated points. Thirdly, well-known concepts from different research areas like the ε-constraint method and the branch-and-bound method are combined and applied in the biobjective flow context. Three approaches for finding all efficient/nondominated points for BMCIF are presented subsequently. Although literature shows that two of them lack in argumentation, they contain a thorough analysis of some structural properties of BMCIF and therefore, we have chosen to present their main ideas.

In the beginning of Section 4.1 we discuss some general results for BMCIF which will be needed in describing algorithms in Sections 4.1 and 4.2.

4.1 Exact Methods

Lee and Pulat [LP93] elaborate on the cycle relationship among basic feasible flows and investigate the structure of solutions of BMCIF. Let x^t and x^{t+1} be two efficient basic solutions of BMCF with objective vectors y^t and y^{t+1} which are topologically adjacent in $\mathcal{Y}_N$. Due to the total unimodularity of the constraint matrix, x^t and x^{t+1} are integral. Consider the basic tree T of x^t and let (u, v) be the nonbasic arc whose inclusion in T yields x^{t+1}. It is well-known that the union of T and (u, v) contains a unique cycle. Let $\delta_{uv} \in \mathbb{N}$ denote the maximum amount of flow that can be sent along this cycle. The number of supported integer nondominated nonextreme points on the line connecting y^t and y^{t+1} equals $\delta_{uv} - 1$. Furthermore, such integer points are equidistant from each other [PHL92, LP93]. Figure 4 illustrates this result. Here, $\delta_{uv} = 3$. Augmenting the cycle induced by (u, v) by 1 and 2 units results in the points P and Q, respectively.

Let $(i, j) \neq (u, v)$ be a nonbasic arc in the bases of x^t and x^{t+1} whose arc flow x_{ij} is equal to its lower bound l_{ij}. Assume it is feasible to introduce arc (i, j) into the bases of x^t and x^{t+1}. Let $x^{t'}$ and $x^{t+1'}$ be the resulting flows when sending one unit of flow along the cycles originating from adding arc (i, j) to the basic trees of x^t and x^{t+1}, respectively. Then, $x^{t'}$ and $x^{t+1'}$ are efficient flows for a modified BMCF with respect to an increased lower bound $l'_{ij} := l_{ij} + 1$ (see Figure 4). Note that the bases of x^t and $x^{t'}$ (x^{t+1} and $x^{t+1'}$) are the same. Hence, it is possible to introduce the nonbasic arc (u, v) in the basis of $x^{t'}$ to obtain $x^{t+1'}$.

Let $y^k := Cx^k$ for $k = t, t', t+1$, and $t + 1'$. It is shown in Lee and Pulat [LP93] that the slopes of the two lines joining y^t and y^{t+1} and $y^{t'}$ and $y^{t+1'}$, respectively, are equal. Furthermore, the number of integer flows lying on each of these lines differ by at most one. The distances between these integer points lying on the parallel lines are the same. In Figure 4 the number of points on the line connecting $y^{t'}$ and $y^{t+1'}$ is one less than the number of points on the line connecting y^t and y^{t+1}. The distances between points on these lines are equal, e.g., $||y^{t'} - R||_2 = ||y^t - P||_2$.

Note that these results can be generalized when increasing the lower bound l_{ij} of nonbasic arc (i, j) by more than one unit assuming feasibility is maintained. These generalizations are indicated by the dashed lines in Figure 4.
Obviously, analogous arguments can be used if $x_{ij} = u_{ij}$ instead of $x_{ij} = l_{ij}$.

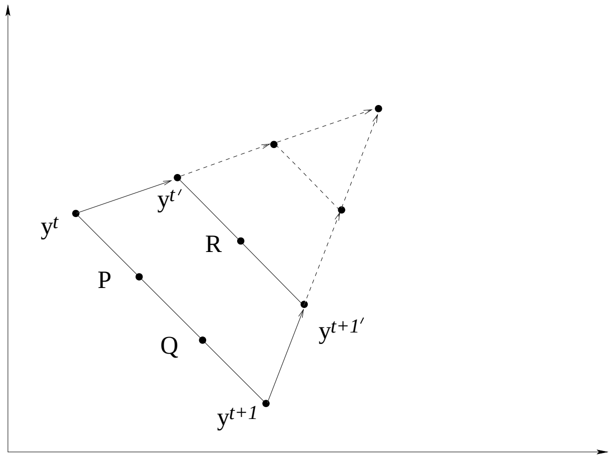

Figure 4: Illustration of the structure of $\mathcal{Y}$.

Based on this idea, Lee and Pulat [LP93] prove that any efficient integer flow $\hat{x} \in \mathcal{X}_E$ can be obtained as a supported efficient solution with respect to modified lower and upper capacities on some arcs. Ehrgott [Ehr99] generalizes this result to MMCFs with more than two objectives.

Lee and Pulat [LP93] assume that all extreme nondominated points and all other supported nondominated points are explicitly known. The results above are used to compute candidate points located inside the triangles defined by consecutive supported nondominated points. Instead of explicitly testing all combinations of modified lower and upper capacities, an implicit search for efficient flows is performed.

For each supported integer efficient extreme point x^t it is checked whether a nonbasic arc $(i,j) \neq (u,v)$ can enter the basic tree of x^t possibly leading to an unsupported efficient solution $x^{t'}$. Integer points on the line that connects $y^{t'}$ and $y^{t+1'}$ are generated as well. Rules are established that exclude nonbasic arcs from consideration which yield candidates being dominated by previously found candidates. For example, it is observed that if $y^{t'} \in y^t + \mathbb{R}^2_{\geqq}$ and the number of integer points on the line connecting $y^{t'}$ and $y^{t+1'}$ is less than or equal to the number of integer points on the line connecting y^t and y^{t+1}, then all points resulting from introducing (i,j) to the basic tree of x^t are dominated (see Figure 5). Thus, not all feasible flows are generated - i.e. the search is, indeed, implicit. After all candidates have been generated by checking all supported efficient extreme points and all possible nonbasic arcs, a filtering process deletes dominated points from the candidate set. It is claimed that the remaining points are nondominated and all nondominated points have been found. The algorithm was implemented and tested on six different problem setups. The number of nodes and arcs vary between 10 to 20 and 20 to 50, respectively. Five different instances have been randomly generated and solved for each setup. For more details, the reader is referred to the PhD thesis of Lee [Lee89].

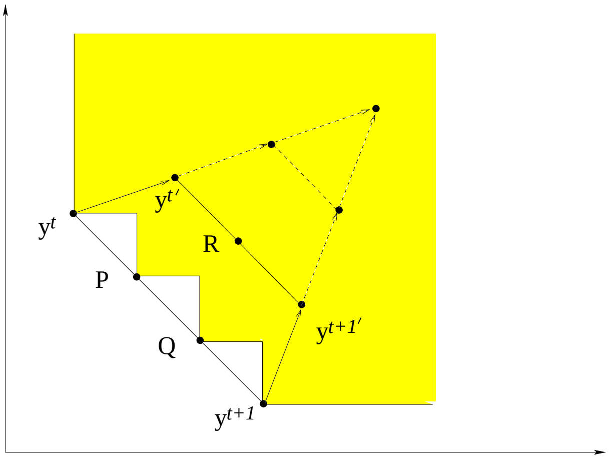

Figure 5: Illustration of a rule eliminating dominated points.

Huarng, Pulat, and Ravindran [HPR92] extend the algorithm of Lee and Pulat focusing on degeneracy phenomena.

Sedeño-Noda and González-Martín [SNGM01] claim that the algorithm stated in [LP93] is incorrect. They argue that Lee and Pulat might miss some efficient flows since their algorithm introduces only two nonbasic arcs (i,j) and (u,v) at a time whereas more than two arcs are needed in general.

Sedeño-Noda and González-Martín use a left-to-right approach starting with the efficient solution that is optimal with respect to objective function c^1 to find unsupported efficient integer flows. Each efficient flow found by the algorithm is associated with a set of nonbasic candidate arcs the introduction of which might possibly yield an unsupported efficient flow. Filtering rules similar to those in [LP93] are utilized to reduce the number of nonbasic arcs in these candidate sets that are stored for further investigation. In each iteration, a new integer efficient flow x^{new} is obtained by adding a nonbasic arc (i, j) to the basic tree of a previously found efficient flow x^{old}. The left-to-right approach requires that Cx^{new} has the smallest c^1-value among all undiscovered nondominated objective vectors. This is implemented by checking all known efficient flows with nonempty candidate sets. Among all combinations of known efficient flows and nonbasic candidate arcs, the one yielding the nondominated objective vector with smallest c^1-value is chosen. The candidate set of x^{old} is assigned to x^{new} while the arc (i, j) is deleted from the candidate set of x^{old}. Doing this, ensures that combinations of nonbasic variables are taken into account when searching for unsupported nondominated points. The procedure stops when all candidate sets are empty.

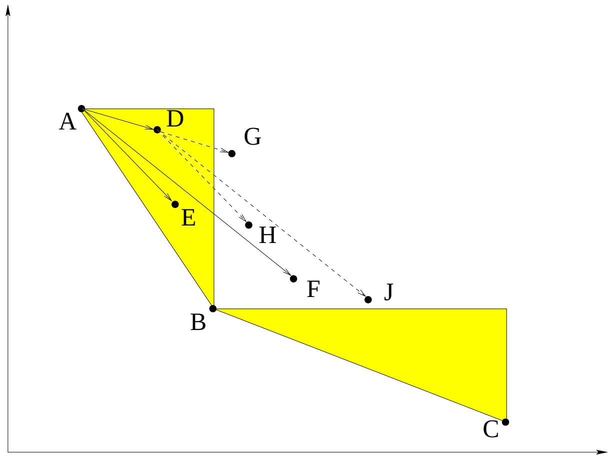

Figure 6: Illustration of the algorithm by Sedeño-Noda and González-Martín.

Figure 6 illustrates the algorithm: Suppose that points A, B, and C are the points obtained after Phase 1. The candidate nonbasic arcs associated with point A yield points D, E, and F. In the first iteration, point D is generated since it has smallest c^1-value among all undiscovered nondominated points. The candidate set of D is the same as the one of A as indicated by the dashed arrows. [SNGM01] do not investigate the points G, H, and J any further, since these points are dominated.

In their numerical tests, 180 random samples are generated with 10 to 25 nodes and 20 to 100 arcs. Sedeño-Noda and González-Martín observe that the number of unsupported nondominated objective vectors increases with the size of the networks. In their study, some 90% of all efficient flows are unsupported.

Przybylski, Gandibleux, and Ehrgott [PGE05] show the incorrectness of the approach suggested by Sedeño-Noda and González-Martín, since the latter authors implicitly assume that the adjacency graph is connected. But [PGE05] give a counterexample proving that the graph theoretical connectivity property does not hold for BMCIF. They conclude that it is not possible to find all nondominated objective vectors for BMCIF by a straightforward application of the network simplex algorithm.

The result of [PGE05] indicates that a different definition of an adjacency graph is needed to establish the graph theoretical connectivity result for MMCIF. One way to do this is based on the result of **Hamacher, Pedersen, and Ruzika** [HPR05]. They prove that any efficient flow x can be obtained from another efficient flow x' by addition of an efficient incremental circulation from the incremental graph of x'. Furthermore, it is shown that no such addition of an efficient flow and an efficient circulation can return a non-efficient flow. Hence the modified adjacency graph containing all integer efficient flows is complete and thus trivially connected. Current research focuses on a characterization of a minimal set of incremental flows which are sufficient to maintain this connectivity property. By the result of [PGE05] it is known that cycle flows do not suffice.

Figueira [Fig02] proposes a branch-and-bound approach to find all nondominated objective vectors for BMCIF. Each pair of extreme nondominated objective vectors defines a triangle in the objective space where unsupported nondominated objective vectors may be located. Each of these triangles are searched for unsupported nondominated objective vectors in a left-to-right approach. The efficient flows found by the algorithm are stored in a list L with decreasing c^2-values. The ε-constraint problem P_1 is solved using a branch-and-bound method to decide whether there are unsupported nondominated objective vectors having c^2-values less than the first element $\hat{x}$ and larger than the second element in the list L.

$$\begin{aligned} \min\ & c^1 x \\ \text{s.t.}\ & c^2 x \leq c^2 \hat{x} - 0.5 \\ & x \in \mathcal{X}_{flow} \end{aligned} \tag{P_1}$$

All subproblems occuring in the branch-and-bound procedure are stored in a list W. Problem P_1 is the current problem and its continuous relaxation is solved. Let x^{rel} denote the optimal value of the relaxed problem. If x^{rel} is integer, the optimal solution of P_1 is found and x^{rel} is known to be efficient. If not, branching is performed to find an integer solution as described in the following. Note that $y^{rel} = Cx^{rel}$ is a vector located on the efficient frontier and can thus be determined by sending a fractional value along the cycle that connects the two nondominated extreme points in the triangle of interest. Let (u, v) denote the arc that defines this cycle. Two new problems P_2 and P_3 are set up by introducing to P_1 the constraints $l_{uv} \leq x_{uv} \leq \lfloor x_{uv}^{rel} \rfloor$ and $\lfloor x_{uv}^{rel} \rfloor + 1 \leq x_{uv} \leq u_{uv}$, respectively. The costs of the continuous relaxations of P_2 and P_3 are computed using the network simplex algorithm. Both problems are added to the list W and branching is now performed on these two problems. Three rules are set up for fathoming: A subproblem is discarded from consideration when it has no feasible solution or when its cost is greater than the cost of the current best solution. If a subproblem P_i has an integer solution with a cost lower than the cost of the current best solution, then all subproblems having a cost greater than the cost of P_i are removed from W and the solution of P_i becomes the current best solution. The algorithm iterates until an optimal integer solution to problem P_1 is found. Whenever an optimal solution of P_1 is found, the first element of L is deleted. If the solution of problem P_1 is a new efficient solution, it is inserted in the front of L and the algorithm proceeds.

4.2 Approximate Methods

Two lines of thoughts are distinguishing the authors providing a representation set for the solutions of MMCIF. Some interpret this problem as the calculation of all integer efficient points on the efficient frontier [LP91, Nik98]. Others compute only a limited amount of compromise solutions [CM96, Ehr99, Ham95, MG98, Nik01, Nik03]. The latter group of papers will be reviewed in Section 5.

Compared with algorithms looking for all efficient solutions of BMCIF, papers dealing with the determination of the integer flows on the efficient frontier of BMCF face a conceptually easier task. In fact, all integer flows on the efficient frontier can be found by methods designed for solving the continuous BMCF only involving considerations of network cycles already discussed in Section 3, in particular using the ideas of [LP91, PHL92].

Lee and Pulat [LP91] extend their algorithm for determining all efficient extreme points in decision space for the continuous BMCF to find all integer efficient points on the efficient frontier. While moving from one efficient extreme point solution, x^t with objective value y^t to the topologically adjacent x^{t+1} with objective value y^{t+1} an integer value δ_{uv} is sent along the nonbasic arcs (u, v). All intermediate integer points, I^t, on the frontier line piece $[y^t, y^{t+1}]$ are determined as

$$I^t = y^t + \alpha \frac{y^{t+1} - y^t}{\delta_{uv}}, \text{ where } \alpha = 1, \ldots, \delta_{uv} - 1.$$

Nikolova [Nik98] develops an algorithm finding all supported integer efficient solutions of BMCF with a designated source node s and sink node t. A flow of q units are to be shipped from s to t. She proves that each supported efficient solution of BMCIF with flow value q can be obtained as a sum of a basic feasible solution x with flow value $v(x) = q$, and a Pareto circulation flow of value 0 in the incremental graph $G(x)$. The description of the algorithm offers no details on how to apply the weighted sum method.

5 Finding Compromise Solutions

A group of authors refer to the theoretical complexity of generating all efficient solutions of MMCF and MMCIF proven by Ruhe [Ruh88a]. The difficulty faced by a decision maker to choose from a large set of offered efficient flows is also discussed. Based on this insight, several methods for generating only a limited amount of compromise solutions are proposed.

Obviously, it is easy to compute just a single efficient solution, since lexicographical minimum cost flows are known to be efficient. Although an arbitrary lexicographic solution is in general not a good compromise, **Calvete and Mateo** ([CM95] and [CM96]) elaborate on this idea.

[CM96] deduct complementary slackness optimality conditions for the lexicographical ordering and specify two algorithms for finding a lexicographical minimum cost flow: lexicographical generalizations of the out-of-kilter method and of the primal-dual method ([AMO93]). Numerical comparisons (using test networks with 2, 3, 5 and 15 objective functions and n and m varying between 27 and 102 and 250 and 1000, respectively) between these algorithms indicate that the out-of-kilter algorithm outperforms the primal-dual method. An application in water resource planning is discussed.

A sequential approach for solving the lexicographic MMCF problem is introduced in [CM96]. First, the network flow problem regarding the objective function with the highest priority is solved by using some MCF algorithm. The flow on arcs having non-zero reduced costs is fixed to the current value. This ensures that the current flow remains optimal for the objective function currently under consideration throughout the algorithm. The flow structure is still preserved when fixing variables. The new network flow problem is then solved regarding the objective function with the second highest priority, and so on. This iterative procedure terminates when all objective functions have been considered or when the flow is fixed to some value on every arc. Two different strategies for fixing flow on arcs are specified and numerically compared using the same instances as in Calvete and Mateo [CM95]. The solution obtained with either of the algorithms by Calvete and Mateo is a supported integer efficient optimal flow.

Nikolova [Nik01, Nik03] develops two interactive algorithms for finding an integer compromise solution of MMCF with side constraints. In [Nik01] the side constraints are imposed initially by a decision maker but may be altered during the algorithm to obtain feasibility. In [Nik03] the side constraints are imposed during the algorithm.

Nikolova [Nik03] initially finds a solution of the weighted sum method with equal weights. In each iteration of the algorithm the decision maker is asked to classify which criteria should be improved, which criteria can be worsened and which criteria should be kept at the same value. Exploiting the same ideas of fixing flow on arcs and reducing the considered network as Calvete and Mateo [CM96], smaller MMCFs and BMCFs are solved using the weighted sum method and a left-to-right approach resembling the one found in Pulat et al. [PHL92], respectively. The identified flow is a supported integer efficient flow.

Nikolova [Nik01] solves in each iteration of the interactive algorithm a BMCF by a left-to-right approach. The two objectives to be considered are a weighted sum of already considered objectives and an objective for which the decision maker's constraints have not been fulfilled, respectively. If infeasibility is revealed during the algorithm the decision maker is asked to change the requirements. The obtained solution is a supported integer efficient flow.

Hamacher [Ham95] develops a polynomial time algorithm for solving the K best integer MCF. This algorithm is extended to a solution procedure for the *max ordering* (MO) flow problem that minimizes the worst of the single objective functions. Considering as before p objectives to be minimized simultaneously and letting $\mathcal{X}_{flow}$ denote the set of integer feasible flows for the flow problem the MO flow problem can formally be stated as

$$\min_{x \in \mathcal{X}_{flow}} \max_{q=1,\ldots,p} c^q x, \text{ where } c^q x = \sum_{\{(i,j) \in A\}} x_{ij} c^q_{ij}.$$

It is noted that there exists at least one flow which is both a MO flow and efficient. The crucial part of the developed ranking algorithm is to obtain the second best solution, y for a MCF given the best solution, x. This is achieved utilizing the network structure by determination of a proper minimal cycle, C in the incremental graph of x, $G(x)$. This cycle is added to x yielding y. The tree structure of the algorithm is built by consecutive changes of lower and upper arc bounds, ensuring intactness of the network structure. By applying the ranking method for a weighted sum of the p objectives a lower bound for the MO flow problem is improved iteratively until an integer MO solution has been identified.

Ehrgott [Ehr99] extends the ideas of Hamacher in the search for a solution to the integer lexicographical max ordering network flow problem. A lexicographical max ordering solution is known to be a max ordering flow as well as a Pareto flow.

This is based on the fact that if $x^1, \ldots, x^K$ are the K best flows with respect to the first objective, i.e., $c^1x^1 \leq c^1x^2 \leq \cdots \leq c^1x^K \leq c^1x \; \forall x \in \mathcal{X}_{flow}$. Then $c^1x^K > \min_{x \in \mathcal{X}_{flow}} \max_{q=1,\ldots,p} c^q x$ implies that all max ordering solutions are contained in $\left\{x^1, \ldots, x^K\right\}$.

Therefore, Ehrgott proposes to identify the lexicographical max ordering solution by applying the ranking method for MCF of [Ham95] followed by a suitable sorting and selection of the obtained solutions.

Mustafa and Goh [MG98] find an integer efficient compromise solution of BMCIF and for the tricriteria MCIF problem (TMCIF). The software package DINAS [OSZ92], originally designed for solving transshipment problems involving facility location, is used to find a non-integral efficient compromise solution. An integral solution of BMCIF is obtained by rounding all non-integral flow values on arcs to their nearest integer value. This is justified by the fact that non-integrality in an efficient solution for the bicriteria case can only occur on a unique cycle between two adjacent efficient extreme points with objective vectors on the efficient frontier. Only in the case where the deviation from the nearest integer value is 1/2, the cycle needs to be traced to get the correct flow direction on the arcs. The rounding process corresponds to sliding to the nearest integer point on the efficient frontier, and hence the identified solution is a supported integer efficient flow. The result is generalized to TMCIF obtaining also an efficient supported solution. The proposed algorithm is applied to the practical problem of assigning courses to faculty members.

Figueira, M'Silti, and Tolla [FMT98] provide an interactive method for finding a robust solution of MMCIF. An initial solution is found applying the augmented weighted Tchebycheff method [SC83, Ste86]. When using the augmented weighted Tchebycheff method as a scalarization technique, a single objective MCF with side constraints has to be solved. The obtained solution is presented to the decision maker, who builds an indifference area using thresholds around the obtained nondominated point in the objective space. Further points located in the indifference area are identified by a heuristic method based on Lagrangian duality and subgradient techniques. The authors claim to be able to perform a test of the quality of the obtained solutions from the decision maker's point of view without explicitly knowing the decision maker's utility function. The outcome of the algorithm is the most robust solution of the ones in the indifference area. This solution need not be an efficient flow for MMCF. The algorithm was implemented and tested on bicriteria transportation problems with 200 nodes and the number of arcs ranging between 1308 and 2900.

The method by **Sedeño-Noda and González-Martín** [SNGM03] that finds all efficient extreme points for BMCF (described in Section 3.1), consists of multiple application of a subroutine performing guided search. As pointed out in [SNGM03], this subroutine on its own serves as a solution method for generating an efficient extreme point located in a desired region of the objective space.

Sun [Sun03] investigates computational efficient algorithms for generating not only one but a finite number of nondominated solutions with the augmented weighted Tchebycheff method. The network simplex method with side constraints proposed by Chen and Saigal [CS77] is capable of solving the occuring MCFs with side constraints. Three strategies are proposed to produce good starting solutions for this special network simplex algorithm:

1. The weighted sum method.
2. The Tchebycheff method.
3. A sequential combination of the weighted sum method and the Tchebycheff method, i.e., a solution obtained with the weighted sum method is used as a starting solution for the Tchebycheff method.

The optimal solution of the three strategies is transformed into an initial basic feasible flow for the augmented weighted Tchebycheff problem. The network simplex method with side constraints uses the starting solution and solves the augmented weighted Tchebycheff problem to optimality.

The three strategies are numerically compared with the alternative of applying the network simplex method with side constraints directly. For testing, networks with $p = 3$, 5, and 7 criteria and $n = 100$, 200, 300, 400, 500 and $m = 20n$ are generated. Costs are taken randomly from the interval $[1, 15]$, while all lower capacities are set to be zero and the upper capacities are randomly chosen from the interval $[100, 500]$. For each of the 15 parameter setups, 10 instances are solved. Strategy 3 outperforms the other algorithms with respect to CPU times in almost all test problems which is due to a reduced number of iterations needed by the special simplex method for network problems with side constraints.

6 Conclusion

In this paper, we have reviewed papers dealing with multiple objective minimum cost flow problems both in the continuous (MMCF) and the integer version (MMCIF).

To make a direct comparison of the solution procedures for the multiple objective minimum cost flow problem possible, the main features of the reviewed articles are concisely listed in Table 3. For this purpose, we adapt the classification scheme of Ehrgott and Gandibleux [EG02]. The articles are grouped analogously as in Sections 3 to 5 and listed in alphabetical order within each group. Table 3 has six columns. In the first column, we refer to the article under consideration. The number of objective functions in this article is listed in the second column. Here, we distinguish between approaches designed for biobjective (denoted by 2) and general multiple objective problems (denoted by p). Note that all articles consider sum objectives. The third and fourth column provide information about the type of the problem and the applied solution methods, respectively. Tables 1 and 2 explain the abbreviations used in these columns. In the last two columns, a "+" indicates that examples are included in the paper and that the algorithm has been implemented by the authors, respectively.

Most of the approaches to find all efficient flows (exact methods) are of theoretical rather than practical interest: Since the number of efficient flows is, in general, exponential in the size of the network, an explicit enumeration of all efficient flows is out of the question. This is also true, if one is interested in an analytic (implicit) description of all efficient flows using only extreme efficient flows, since their number is exponential as well.

For the continuous problem MMCF, algorithms for finding a representative set of all efficient flows (approximate methods) generate results with quality guarantees. All of these algorithms are based on the sandwich idea. A corresponding approach for the integer case MMCIF, is missing in the current literature, but will be discussed in the forthcoming paper of Hamacher, Pedersen, and Ruzika [HPR05].

An important problem which is not well understood at this point of time is the graph theoretical connectivity in MMCIF. If the adjaceny graph has node set $\mathcal{N}$ equal to all integer efficient flows and arc set $\mathcal{A}$ defined by $(i, j) \in \mathcal{A}$ if and only if flow i can be obtained by flow j through a change along a single cycle, then the results of Przybylski, Gandibleux, and Ehrgott [PGE05] show the non-connectivity of this graph. An open problem is to find a reasonable definition of node adjacency which establishes connectivity. Obviously, the adjacency definition of Hamacher, Pedersen, and Ruzika [HPR05] using efficient residual flows is just a first step in this research area.

An important observation in the multiple objective flow literature is the fact that almost all papers deal exclusively with the case of two objectives. Obviously, this is due to the fact that the efficient frontier has in this situation the form of a piecewise linear function, which is easy to describe and to analyze. The question, which results can be generalized to more than two objectives is currently under research.

Entry	Explanation
E	Finding the efficient set
e	Finding a subset of the efficient set
SE	Finding the supported efficient solutions
Ê	Finding an approximation of the efficient set
lex	Solving the lexicographical problem
MO	Solving the max ordering problem
lexMO	Solving the lexicographical max ordering problem
C	Finding a compromise solution

Table 1: Entries for *Problem Type*.

Entry	Explanation
SP	Exact algorithm specifically designed for the problem
BB	Algorithm based on branch and bound
εC	Algorithm based on the ε-constraint method
IA	Interactive method
A	Approximate algorithm with worst case performance bound
LP	Method based on linear programming

Table 2: Entries for *Solution Method*.

Paper	Number of Objectives	Problem Type	Solution Method	Example	Implemented
Continuous MMCF					
[BRRS89]	2	Ê	A	-	+
[FBR89]	2	Ê	A	-	+
[LP91]	2	E	SP	+	-
[MP84]	2	E	SP	+	-
[PHL92]	2	E	LP	+	+
[Ruh91]	2	Ê	A	+	+
[Ruh91]	2	E	SP	+	+
[RF90]	2	Ê	A	-	+
[SNGM00]	2	E	LP	+	+
[SNGM03]	2	E	SP	+	+
Integer MMCF					
[Fig02]	2	E	SP/BB/εC	+	-
[HPR92]	2	E	SP	+	+
[LP91]	2	SE	SP	-	-
[LP93]	2	E	SP	+	+
[Nik98]	2	SE	SP	-	-
[SNGM01]	2	E	SP	+	+
Compromise solutions					
[CM95]	p	lex	SP	+	+
[CM96]	p	lex	SP	+	+
[Ehr99]	p	lexMO	SP	+	-
[FMT98]	p	C	IA	+	+
[Ham95]	p	MO	SP	-	-
[MG98]	2(3)	C	SP	+	-
[Nik01]	p	C	SP/IA	-	-
[Nik03]	p	C	SP/IA	-	-
[SNGM03]	2	C	SP	+	-
[Sun03]	2	e	SP	-	+

Table 3: Classification for the reviewed papers.

References

[AMO93] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. *Network Flows: Theory, Algorithms, and Applications.* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1993.

[AN79] Y.P. Aneja and K.P.K. Nair. Bicriteria transportation problem. *Management Science*, 25(1):73–78, 1979.

[BHR91] R.E. Burkard, H.W. Hamacher, and G. Rote. Sandwich approximation of univariate convex functions with an application to separable convex programming. *Naval Research Logistics Quarterly*, 38(6):911–924, 1991.

[BRRS89] R.E. Burkard, G. Rote, G. Ruhe, and N. Sieber. Algorithmische Untersuchungen zu bikriteriellen kostenminimalen Flüssen in Netzwerken. *Wissenschaftliche Zeitschrift der technischen Hochschule Leipzig*, 13(6):333–341, 1989.

[CM86] J. Current and H. Min. Multiobjective design of transportation networks: taxonomy and annotation. *European Journal of Operational Research*, 26:187–201, 1986.

[CM93] J. Current and M. Marsh. Multiobjective transportation network design and routing problems: taxonomy and annotation. *European Journal of Operational Research*, 65:4–19, 1993.

[CM95] H.I. Calvete and P.M. Mateo. An approach for the network flow problem with multiple objectives. *Computers and Operations Research*, 22(9):971–983, 1995.

[CM96] H.I. Calvete and P.M. Mateo. A sequential network-based approach for the multiobjective network flow problem with preemptive priorities. In M. Tamiz, editor, *Multi-Objective Programming and Goal Programming - Theory and Applications*, volume 432 of *Lecture Notes in Economics and Mathematical Systems*, pages 74–86. Springer Verlag, Berlin, 1996.

[CS77] S. Chen and R. Saigal. A primal algorithm for solving a capacitated network flow problem with additional linear constraints. *Networks*, 7(1):59–79, 1977.

[EG02] M. Ehrgott and X. Gandibleux. *Multiple Criteria Optimization: State of the Art Annotated Bibliographic Surveys.* Kluwer Academic Publishers, Boston, Massachusetts, 2002.

[Ehr99] M. Ehrgott. Integer solutions of multicriteria network flow problems. *Investigação Operacional*, 19:229–243, 1999.

[Ehr00] M. Ehrgott. *Multicriteria optimization*, volume 491 of *Lecture Notes in Economics and Mathematical Systems.* Springer-Verlag, Berlin, 2000.

[FBR89] B. Fruhwirth, R.E. Burkard, and G. Rote. Approximation of convex curves with application to the bicriterial minimum cost flow problem. *European Journal of Operational Research*, 42:326–338, 1989.

[Fig02] J. Figueira. On the integer bi-criteria network flow problem: A branch-and-bound approach. *Cahier du LAMSADE n^o 191, Université Paris-Dauphine*, 2002.

[FMT98] J. Figueira, H. M'Silti, and P. Tolla. Using mathematical programming heuristics in a multicriteria network flow context. *Journal of the Operational Research Society*, 49:878–885, 1998.

[Geo67] A.M. Geoffrion. Solving bicriterion mathematical programs. *Operations Research*, 15:39–54, 1967.

[Geo68] A.M. Geoffrion. Proper efficiency and the theory of vector maximization. *Journal of Mathematical Analysis and Applications*, 22(3):618–630, 1968.

[GS55] S. Gass and T. Saaty. The computational algorithm for the parametric objective function. *Naval Research Logistics Quarterly*, 2(1/2):39–45, 1955.

[Ham80] H.W. Hamacher. Algebraic flows in regular matroids. *Discrete Applied Mathematics*, 2(1):27–38, 1980.

[Ham95] H.W. Hamacher. A note on k best network flows. *Annals of Operations Research*, 57:65–72, 1995.

[HPR92] F. Huarng, P. Pulat, and A. Ravindran. An algorithm for bicriteria integer network flow problem. In *Proceedings of the 10th International Conference on Multiple Criteria Decision Making, Taipei, Taiwan*, volume 3, pages 305–318, 1992.

[HPR05] H.W. Hamacher, C.R. Pedersen, and S. Ruzika. Representative systems for bicriteria minimum cost flow problems. *Forthcoming paper*, 2005.

[Ise74] H. Isermann. Proper efficiency and the linear vector maximum problem. *Operations Research*, 22:189–191, 1974.

[Ise79] H. Isermann. The enumeration of the set of all efficient solutions for a linear multiple objective program. *Naval Research Logistics Quaterly*, 26:123–139, 1979.

[KNS74] D. Klingman, K. Napier, and J. Stutz. NETGEN: A program for generating large scale capacitated assignment, transportation, and minimum cost flow network problems. *Management Science*, 20(5):814–821, 1974.

[Lee89] H. Lee. *Integer solutions of bicriteria network flow problems.* PhD thesis, University of Oklahoma, Norman, OK, 1989.

[LP91] H. Lee and P.S. Pulat. Bicriteria network flow problems: Continuous case. *European Journal of Operational Research*, 51:119–126, 1991.

[LP93] H. Lee and P.S. Pulat. Bicriteria network flow problems: Integer case. *European Journal of Operational Research*, 66:148–157, 1993.

[MG98] A. Mustafa and M. Goh. Finding integer efficient solutions for bicriteria and tricriteria network flow problems using DINAS. *Computers and Operations Research*, 25(2):139–157, 1998.

[MP84] R. Malhotra and M.C. Puri. Bi-criteria network problem. *Cahiers du Centre d'Études Recherche Opérationelle*, 26(1/2):95–102, 1984.

[Nac78] P. Naccache. Connectedness of the set of nondominated outcomes. *Journal of Optimization Theory and Applications*, 25:459–467, 1978.

[Nik98] M. Nikolova. Properties of the effective solutions of the multicriteria network flow problem. *Problems of Engineering Cybernetics and Robotics*, 47:104–111, 1998.

[Nik01] M. Nikolova. An approach for finding Pareto optimal solutions of the multicriteria network flow problem. *Cybernetics and Information Technologies*, 2:56–62, 2001.

[Nik03] M. Nikolova. A classification based approach for finding Pareto optimal solutions of the multicriteria network flow. *Cybernetics and Information Technologies*, 3(1):11–17, 2003.

[OSZ92] W. Ogryczak, K. Studziński, and K. Zorychta. DINAS: a computer-assisted analysis system for multiobjective transshipment problems with facility location. *Computers and Operations Research*, 19(7):637–648, 1992.

[PGE05] A. Przybylski, X. Gandibleux, and M. Ehrgott. The biobjective integer minimum cost flow problem - incorrectness of Sedeño-Noda and González-Martín's algorithm. Computers and Operations Research, in print, 2005.

[PHL92] P.S. Pulat, F. Huarng, and H. Lee. Efficient solutions for the bicriteria network flow problem. *Computers and Operations Research*, 19(7):649–655, 1992.

[RF90] G. Ruhe and B. Fruhwirth. ε-optimality for bicriteria programs and its application to minimum cost flows. *Computing*, 44:21–34, 1990.

[Ruh88a] G. Ruhe. Complexity results for multicriterial and parametric network flows using a pathological graph of Zadeh. *Zeitschrift für Operations Research*, 32(1):9–27, 1988.

[Ruh88b] G. Ruhe. *Flüsse in Netzwerken - Komplexität und Algorithmen.* PhD thesis, Technische Hochschule Leipzig, Sektion Mathematik und Informatik, 1988.

[Ruh91] G. Ruhe. *Algorithmic Aspects of Flows in Networks.* Kluwer Academic Publishers, Dordrecht, 1991.

[SC83] R.E. Steuer and E.-U. Choo. An interactive weighted Tchebycheff procedure for multiple objective programming. *Mathematical Programming*, 26(1):326–344, 1983.

[Skr00] A.J.V. Skriver. A classification of bicriteria shortest path (BSP) algorithms. *Asia-Pacific Journal of Operational Research*, 17:199–212, 2000.

[SNGM00] A. Sedeño-Noda and C. González-Martín. The biobjective minimum cost flow problem. *European Journal of Operational Research*, 124:591–600, 2000.

[SNGM01] A. Sedeño-Noda and C. González-Martín. An algorithm for the biobjective integer minimum cost flow problem. *Computers and Operations Research*, 28:139–156, 2001.

[SNGM03] A. Sedeño-Noda and C. González-Martín. An alternative method to solve the biobjective minimum cost flow problem. *Asia-Pacific Journal of Operational Research*, 20(2):241–260, 2003.

[Ste86] R.E. Steuer. *Multiple-Criteria Optimization: Theory, Computation, and Application.* John Wiley & Sons, New York, 1986.

[Sun03] M. Sun. Procedures for finding nondominated solutions for multiple objective network programming problems. *Transportation Science*, 37(2):139–152, 2003.

[UT94] E.L. Ulungu and J. Teghem. Multi-objective combinatorial optimization problems: a survey. *Journal of Multi-Criteria Decision Analysis*, 3:83–104, 1994.

[War83] A. Warburton. Quasiconcave vector maximization: Connectedness of the sets of Pareto-optimal and weak Pareto-optimal alternatives. *Journal of Optimization Theory and Applications*, 40:537–557, 1983.

[YG97] X.Q. Yang and C.J. Goh. A method for convex curve approximation. *European Journal of Operational Research*, 97:205–212, 1997.

[Zad73] N. Zadeh. A bad network for the simplex method and other minimum cost flow algorithms. *Mathematical Programming*, 5:255–266, 1973.