A bi-objective approach to discrete cost-bottleneck location problems*

Sune Lauth Gadegaard† and Lars Relund Nielsen

Department of Economics and Business Economics, Aarhus University, Fuglesangs Allé 4, DK-8210 Aarhus V, Denmark.

Andreas Klose

Department of Mathematics, Aarhus University, Ny Munkegade 118, DK-8000 Aarhus C, Denmark.

October, 2016

Abstract: This paper considers a family of bi–objective discrete facility location problems with a cost objective and a bottleneck objective. A special case is, for instance, a bi-objective version of the (vertex) p-centdian problem. We show that bi-objective facility location problems of this type can be solved efficiently by means of an ε-constraint method that solves at most $(n-1)\cdot m$ minisum problems, where n is the number of customer points and m the number of potential facility sites. Additionally, we compare the approach to a lexicographic ε-constrained method that only returns efficient solutions and to a two–phase method relying on the perpendicular search method. We report extensive computational results obtained from several classes of facility location problems. The proposed algorithm compares very favorably to both the lexicographic ε-constrained method and to the two phase method.

Keywords: discrete facility location; bi-objective optimization; ε-constrained method; lexicographic optimization.

1 Introduction

Single objective location analysis usually distinguishes between two major types of objective functions. Whilst the objective of a *minisum* location problem consists in minimizing average (weighted) costs, a solution to a *minimax* location problem aims at minimizing the maximal (weighted) distance between customer points and facilities. We will refer to the two objectives as a *cost objective* and a *bottleneck objective*, respectively. On networks, the prototype cost and bottleneck location models are the p-median (Hakimi, 1965) and

*Preprint of S.L. Gadegaard, A. Klose and L.R. Nielsen, *A bi-objective approach to discrete cost-bottleneck location problems* in Annals of Operations Research, doi:10.1007/s10479-016-2360-8

†Corresponding author, email: sgadegaard@econ.au.dk.

p-center problem (Hakimi, 1964), respectively. The survey papers by Reese (2006) and Revelle, Eiselt, and Daskin (2008) review the extensive literature on these two important network location problems. Within the subject of discrete facility location, the simplest and also most studied problem with cost–objective is the uncapacitated facility location problem proposed by Balinski (1965). For discrete facility location the bottleneck–objective has not received as much attention as the cost–objective, but Dearing and Newruck (1979) study a capacitated facility location problem with such a bottleneck–objective.

As the cost–objective focuses on minimizing the total/average cost of supplying demand points from supply points, it often provides solutions in which remote and sparsely populated areas are discriminated in terms of accessibility compared to centrally situated and highly populated areas. On the other hand, locating facilities according to the bottleneck objective (focussing on equity rather than efficiency) may cause a large increase in the total cost, thus generating a substantial loss in cost efficiency. This fact was recognized by Halpern (1976) who minimized a convex combination of the two objectives such that both efficiency and equity were expressed in the resulting solution. On a network, this model is known as the centdian problem. Halpern (1978) showed that when placing *one* facility on an undirected graph so as to minimize the centdian objective, only a finite set composed of the set of vertices and the set of so–called local centers need to be examined. For the case where multiple facilities should be placed, this finite set needs to be expanded to the (still finite) set presented in Pérez-Brito, Moreno-Pérez, and Rodríguez-Martín (1997).

Recognizing the fact that only supported efficient solutions can be found using convex combinations of the objectives, it seems obvious to apply a bi-objective approach to the cost–bottleneck problem. Instead of only considering a fixed convex combination of the two objectives, the entire set of non-dominated outcomes will then be generated. The need to balance efficiency and equity does, however, also arise in the case of other, similar problems. In case of the transportation problem, the literature distinguishes between the classical Hitchcock-type and the bottleneck transportation problem (Garfinkel and Rao, 1971; Hammer, 1969). Whilst the objective of the former is to minimize total transportation cost, the latter aims at minimizing the maximum time required to transport all supplies to the destinations. In this case it seems relevant to find a balance between these two objectives, leading to the bi-objective "bottleneck-cost" transportation problem studied by Pandian and Nataraja (2011). This problem can in fact be solved in polynomial time. Combining a cost and a bottleneck objective is also highly relevant in the context of discrete facility location problems such as, for instance, the uncapacitated and capacitated facility location problems. The bottleneck objective may then, in particular, refer to a customer service objective that aims at keeping maximum delivery times small. Facility location models also play a role in the area of supplier selection (Current and Weber, 1994). In this case, the cost objective may refer to the total procurement cost, whereas the bottleneck objective is to minimize the (largest) lead time.

In the literature on planar and network multi–objective facility location problems, most papers concentrate on specific problems and on methodological and theoretical results (see e.g. Hamacher and Nickel (1996) for planar and Hamacher, Labbé, and Nickel (1999) for network location problems). The opposite is true for discrete multi–objective facility location, where the attention has primarily been on applications (see e.g. the survey in Nickel, Puerto, and Rodríguez-Chía (2005)). However, a few contributions with a methodological perspective on multi–criteria discrete location problem exists. In Ross and Soland (1980) the set of Pareto optimal solutions is characterized by rewriting the location problem as a generalized assignment problem with an additional constraint. However, the authors argue not to generate all efficient solutions, and propose an *interactive* approach instead. Fernández and Puerto (2003) consider a multi–objective version of the uncapacitated facility location problem (UFLP). A multi–objective dynamic programming approach is proposed based on a decomposition of the UFLP into a facility selection problem and a demand allocation problem. We have, however, not been able to find any papers investigating, in a discrete bi–objective setting, the combination of a cost objective and a bottleneck. For surveys on multi–objective facility location problems we refer the reader to Nickel et al. (2005) and Farahani, SteadieSeifi, and Asgari (2010).

In this paper we, therefore, suggest a bi-objective approach to balance cost minimization and maximum transportation times for a family of discrete facility location problems comprising many well studied location problems as special cases. We establish the computational complexity of the problem and prove the problem to be tractable. Furthermore, we propose a scheme for solving the problem by means of an ε–constrained method. We suggest two ways to accommodate the issue of generating weakly efficient solutions: First, a simple change of the cost matrix that imposes the non–linear ε–constraint on the bottleneck objective. Second, we outline a scheme based on lexicographic branch–and–bound which generates no weakly efficient solutions. Last, we compare the ε–constraint algorithm to a two–phase implementation. The main contributions of the paper can thus be summarized as follows:

1. We propose a family of discrete bi–objective facility location problems, encompassing many problems known from the literature.

2. We show that even though the problems are generally $\mathcal{NP}$–hard, they are computationally tractable in a certain sense.

3. We propose a simple methodology, based on the ε–constraint method, for solving these problems.

4. Through extensive experiments we show that the methodology is indeed very efficient compared to other algorithms from the literature.

The remaining of this paper is organized as follows: Section 2 introduces the cost-bottleneck location problem and Section 3 outlines the preliminaries of bi–objective combinatorial optimization and establishes the computational complexity of the problem. Section 4 outlines the ε–constraint algorithm proposed to solve the problem and finally, results from extensive computational experiments are reported in Section 5.

2 The bi–objective cost-bottleneck location problem

A large number of facility location problems are special cases of the general integer program

$$
\begin{aligned}
\min\ & \sum_{i\in I}\sum_{j\in J} c_{ij}x_{ij} + \sum_{i\in I} f_i y_i && \text{(1a)}\\
\text{s.t.:}\ & \sum_{i\in I} x_{ij} = 1, \quad \forall j \in J, && \text{(1b)}\\
& x_{ij} - y_i \le 0, \quad \forall i \in I,\ j \in J, && \text{(1c)}\\
& (\mathbf{x}_i, y_i) \in \mathcal{X}_i, \quad \forall i \in I, && \text{(1d)}\\
& y \in \mathcal{Y}. && \text{(1e)}
\end{aligned}
$$

Here J is the set of demand points which need to be served by a set of open facilities picked among the potential facility sites in I. The cost of servicing all of customer j's demand from facility i amounts to $c_{ij} \ge 0$ while opening a facility i results in a fixed charge of $f_i \ge 0$. It is without loss of generality assumed that both c_{ij} and f_i are non–negative integers. Constraints (1b) ensure that all demand at customer j is covered by allocating the demand to at least one open facility while constraints (1c) ensure that demand can only be allocated to open facilities. Constraints (1d) restrict the possible assignments $\mathbf{x}_i = (x_{ij})_{j\in J}$ of demand points to facilities. Possible assignments are *single sourced* if demand points are assigned to only one facility, that is $\mathcal{X}_i \subseteq \{0,1\}^{|J|+1}$. Otherwise we have $\mathcal{X}_i \subseteq [0,1]^{|J|} \times \{0,1\}$. Finally, $\mathcal{Y} \subseteq \{0,1\}^{|I|}$ may introduce further restrictions on the locational decisions y_i.

Special cases of the program (1) comprise among others the uncapacitated facility location problem, the capacitated facility location problem with and without single-sourcing, and the p–median problem.

When the sum of costs is minimized, the relation to the individual demand point is not taken into account meaning that in a spacial setting some customers might be located far from the open facilities. To overcome this issue we introduce yet another objective, namely to minimize the travel time of the *worst* assignment, effectively introducing the bottleneck objective

$$\min \max_{i\in I, j\in J} \{t_{ij} \ :\ x_{ij} > 0\}. \tag{2}$$

It is assumed that $t_{ij} \ge 0$ for all $i \in I$ and all $j \in J$. In this setting, one can consider the t_{ij} as the *travel time* from demand point j to facility i while c_{ij} is the *cost* incurred by this assignment. The introduction of this additional objective function leads to the bi–objective combinatorial optimization (BOCO) problem

$$\min \left(\sum_{i\in I}\sum_{j\in J} c_{ij}x_{ij} + \sum_{i\in I} f_i y_i \ , \ \max_{i\in I, j\in J}\{t_{ij} \ : \ x_{ij} > 0\} \right) \tag{3a}$$

$$\text{s.t.:} \ \sum_{i\in I} x_{ij} = 1, \quad \forall j \in J, \tag{3b}$$

$$x_{ij} - y_i \le 0, \quad \forall i \in I, \ j \in J,$$

$$(\mathbf{x}_i, y_i) \in \mathcal{X}_i, \quad \forall i \in I, \tag{3c}$$

$$y \in \mathcal{Y}, \tag{3d}$$

which will be referred to as the bi–objective cost-bottleneck location problem (BO–CBLP).

3 Preliminaries

In the remainder of this paper, we will adopt the notation from Ehrgott (2005) to induce orderings on $\mathbb{R}^2$. Let $z^1, z^2 \in \mathbb{R}^2$. Then

$$\begin{aligned} z^1 \le z^2 &\Leftrightarrow z^1_k \le z^2_k, & & k = 1,2 \text{ and } z^1 \ne z^2 \\ z^1 <_{\text{lex}} z^2 &\Leftrightarrow z^1 \le z^2 \text{ and } z^1_q < z^2_q, & & \text{where } q = \min\{k = 1,2 \ : \ z^1_k \ne z^2_k\} \\ z^1 \le_{\text{lex}} z^2 &\Leftrightarrow z^1 = z^2 \text{ or } z^1 <_{\text{lex}} z^2 \end{aligned}$$

If $z^1 \le z^2$ we say that z^1 *dominates* z^2. Similarly, if $z^1 \le_{\text{lex}} z^2$ we say that z^1 *lexicographically dominates* z^2. Note the implication that if $z^1 \le z^2$ then $z^1 \le_{\text{lex}} z^2$.

The focus of this section will be on a generic BOCO problem of the form

$$\min\{(f_1(x), f_2(x)) \ : \ x \in \mathcal{X}\}. \tag{4}$$

The objective functions $f_i : \mathcal{X} \to \mathbb{R}$, $i = 1, 2$, are defined over the mixed integer set $\mathcal{X} \subseteq \{0,1\}^{n_1} \times [0,1]^{n_2}$, where n_1 and n_2 are non–negative integers with $n_1 > 0$. The set $\mathcal{X}$ is the set of feasible solutions, also referred to as the feasible set in *decision space*. The image of $\mathcal{X}$, $\mathcal{Z} \coloneqq f(\mathcal{X}) \subseteq \mathbb{R}^2$, where $f(x) \coloneqq (f_1(x), f_2(x))$, is referred to as the feasible set in *outcome space*. It is not obvious what is meant by (4), so to clarify this we use the concept of Pareto optimality or efficiency (see Ehrgott (2005)):

Definition 1. A feasible solution $\hat{x} \in \mathcal{X}$ is called Pareto optimal or *efficient* if there does not exist any $\bar{x} \in \mathcal{X}$ such that $f(\bar{x}) \le f(\hat{x})$. The image $\hat{z} = f(\hat{x})$ is then called *non-dominated.*

A feasible solution $\hat{x} \in \mathcal{X}$ is called *weakly efficient* if there is no $x \in \mathcal{X}$ such that $f_1(x) < f_1(\hat{x})$ *and* $f_2(x) < f_2(\hat{x})$. The corresponding outcome vector $\hat{z} = f(\hat{x})$ is then called weakly non–dominated.

From Definition 1 we have that a solution $\hat{x}$ with $f(\hat{x}) \leq_{\text{lex}} f(x)$ for all $x \in \mathcal{X}$ is efficient (see Ehrgott (2005) for a proof).

Let $\mathcal{X}_E$ denote the set of efficient solutions and let $\mathcal{Z}_N = f(\mathcal{X}_E)$ be the image of this set. The set of non-dominated solutions $\mathcal{Z}_N$ will also be referred to as the *non–dominated frontier*. We distinguish between efficient solutions which are supported, extreme supported, and non–supported (see Ehrgott (2005)).

Definition 2.

1. A solution $\hat{x} \in \mathcal{X}$ is a *supported* efficient solution if there exists a $\lambda \in (0, 1)$ such that $\hat{x}$ is an optimal solution to

$$\min\{(\lambda f_1(x) + (1 - \lambda) f_2(x) \ : \ x \in \mathcal{X}\}$$

The corresponding outcome vector, $\hat{z} := f(\hat{x})$, is called a *supported* non–dominated outcome vector. The set of supported non–dominated outcomes is denoted $\mathcal{Z}_{sN}$.

2. A supported efficient solution $\hat{x}$, with $\hat{z} = f(\hat{x})$, is called an *extreme* supported efficient solution if $\hat{z}$ is an extreme point of $\text{conv}(\mathcal{Z}_N)$. The outcome vector $\hat{z}$ is then called an *extreme* supported non–dominated outcome vector.

3. If $\hat{x} \in \mathcal{X}_E$ and $\hat{z} := f(\hat{x}) \notin \mathcal{Z}_{sN}$, then $\hat{x}$ is said to be an *unsupported* efficient solution and $\hat{z}$ is called an *unsupported* non–dominated outcome.

If two feasible solutions $x_1, x_2 \in \mathcal{X}$ map into the same outcome vector, $z \in \mathcal{Z}$, that is $f(x_1) = f(x_2) = z$, the solutions x_1 and x_2 are called *equivalent*. The literature is not always precise on the outcome of a proposed algorithm and we therefore employ the following definition from Hansen (1980):

Definition 3. A *complete* set $\mathcal{C}$ is a set of efficient solutions such that all $x \in \mathcal{X} \backslash \mathcal{C}$ are either dominated by or equivalent to at least one $\hat{x} \in \mathcal{C}$. Moreover we distinguish between two types of complete sets:

1. A *minimal complete set*, $\mathcal{C}_{\min}$ is a complete set without equivalent solutions. Any complete set contains a minimal complete set.

2. The *maximal complete set*, $\mathcal{C}_{\max}$, is the complete set including all efficient solutions, i.e., $\mathcal{C}_{\max} = \mathcal{X}_E$.

In this paper the focus is on generating a set $\mathcal{C}_{\min}$, i.e., the efficient solutions found for (4) constitute a minimal complete set. Finally, the concept of computational tractability is defined:

Definition 4 (Ehrgott (2005))**.** The BOCO problem (4) is called *intractable* if the cardinality of $\mathcal{Z}_N$ can be exponential in the size of the input and tractable otherwise.

3.1 The complexity of the BO–CBLP problem

Despite the relatively simple nature of the problem, the BO–CBLP (3) is a difficult BOCO problem. The computational complexity of the BO–CBLP is easily established in Proposition 1 as the uncapacitated facility location problem is a special case known to be $\mathcal{NP}$–hard (see e.g. Cornuejols, Nemhauser, and L.A. (1990) for a proof).

Proposition 1. *The BO–CBLP* (3) *is* $\mathcal{NP}$*–hard.*

However, despite the fact that the BO–CBLP is $\mathcal{NP}$–hard, the problem is in fact *tractable* in the sense of Definition 4:

Proposition 2. *The BO–CBLP is tractable.*

Proof. As the travel time matrix $(t_{ij})_{i\in I, j\in J}$ comprises at most $|I| \times |J|$ different values, the bottleneck objective (2) can attain no more than $|I| \times |J|$ different values, implying $|\mathcal{Z}_N| \leq |I| \times |J|$, which is polynomial in the input size. $\square$

Proposition 2 tells us, that even though the BO–CBLP is $\mathcal{NP}$–hard, we only need to solve a polynomial number of $\mathcal{NP}$–hard problems in order to generate the non–dominated frontier, $\mathcal{Z}_N$.

4 Solution methodologies

One of the most well–known approaches for establishing the complete set of Pareto optimal solutions to a bi–objective optimization problem is probably the *ε-constrained method.* This method turns one of the objectives into a constraint. The scalar ε represents the upper bound on the objective, and by varying this scalar in an appropriate way, the complete efficient frontier can be generated. Recently, Bérubé, Gendreau, and Potvin (2009) successfully applied this method to the traveling salesman problem with profits. The literature also suggests a number of variations of this method. Two recent versions are: Filippi and Stevanato (2013) combine the weighted sum scalarization technique with the ε-constrained method and show that exactly $2|\mathcal{Z}_N| - 1$ single objective optimization problems have to be solved in order to produce the entire set of non-dominated outcomes. Two box algorithms based on a combination of lexicographic optimization and the ε-constrained method (the lexicographic ε-constrained method) are proposed by Hamacher, Pedersen, and Ruzika (2007). The proposed algorithm also solves at most $2|\mathcal{Z}_N| - 1$ lexicographic optimization problems.

In this section we describe an ε–constrained method for finding a minimal complete set for the BO–CBLP. The two ε–constrained problems arising in terms of the general BOCO (4) are

$$\min\{f_i(x) \;:\; x \in \mathcal{X},\; f_j(x) \leq \varepsilon_j\}, \tag{P_j^i}$$

where $i, j = 1, 2$ and $i \neq j$. Furthermore, $\varepsilon_j = \hat{f}_j - \gamma > 0$ where $\hat{f}_j$ is the solution value in the second objective of a feasible solution and $\gamma > 0$ is small number that guarantees an

Input: Functions f_1 and f_2 and a feasible set $\mathcal{X}$.
Output: A solution $(\mathcal{C}_{\min}, \mathcal{Z}_N) \subseteq \mathcal{X} \times \mathcal{Z}$ to (4) or a proof that $\mathcal{X}_E = \emptyset$.

Step 0: (Initialization) Set $\mathcal{C}_{\min} = \emptyset$, $\mathcal{Z}_N = \emptyset$, $\varepsilon = \infty$ and $k = 1$.

Step 1: (Subproblem) If $\min\{f_1(x) : x \in \mathcal{X}, \quad f_2(x) \leq \varepsilon\}$ is feasible let x^k be an optimal solution. Else go to *Step 3*.

Step 2: (Update) Set $\mathcal{C}_{\min} = \mathcal{C}_{\min} \cup \{x^k\}$, $\mathcal{Z}_N = \mathcal{Z}_N \cup \{f(x^k)\}$, $\varepsilon = f_2(x^k) - \gamma$ and $k = k + 1$. Return to *Step 1*.

Step 3: Remove dominated solutions from $\mathcal{C}_{\min}$ and their outcome vectors from $\mathcal{Z}_N$ and return $(\mathcal{C}_{\min}, \mathcal{Z}_N)$ as an optimal solution.

Algorithm 1: Summary of the ε–constraint based algorithm.

improvement of the second objective. It is well known that all non–dominated solutions can be found by varying the ε–parameter in an appropriate manner (see e.g. Ehrgott (2005)). BOCO problems yield a straightforward variation scheme for the ε parameter. One simply constructs a sequence where ε is initially set equal to a substantially large value and progressively lowered. An outline of a generic ε–constrained method is given in Algorithm 1. Note that one needs a method for defining the parameter γ in the program (P_j^i) such that no efficient solutions are missed. If the objective moved to the constraints, f_j, is guaranteed to map to the integers, γ takes the value of 1. In Section 4.1 we propose a straightforward way to handle this for the BO–CBLP where we have not made the assumption that $f_2 : \mathcal{X} \to \mathbb{Z}$.

A major drawback of the ε–constrained method is that the set of solutions produced by Steps 1 and 2 in Algorithm 1 usually contains (weakly) dominated solutions, such that the method presumably solves far more ε–constrained programs than actually required. One way of preventing this is to obtain a *lexicographically* optimal solution to the ε–constrained subproblem in *Step 1* in Algorithm 1. This way the solution obtained in this step is guaranteed to be efficient, and *Step 3* can be skipped. We describe how such a lexicographically optimal solution can be obtained in Section 4.2. Another weak point of the ε–constrained algorithm is that the ε–constraint might ruin the structure of the underlying problem. This issue is addressed in Section 4.1. Another approach would be to apply the augmented ε–constraint approach (see Mavrotas (2009)), but as this requires a weighting of the two objectives we did not do so as the computational results showed that this slows down the computations significantly (see Section 5.5).

4.1 The ε–constrained method for BO–CBLP

In the ε–constrained method for BO–CBLP we choose to move the non–linear bottleneck–objective into the constraints and obtain the problem

$$\min \sum_{i\in I}\sum_{j\in V} c_{ij}x_{ij} + \sum_{i\in I} f_i y_i \tag{5a}$$

$$\text{s.t.:} \sum_{i\in I} x_{ij} = 1, \quad \forall j \in J \tag{5b}$$

$$x_{ij} - y_i \leq 0, \quad \forall i \in I,\ j \in J, \tag{5c}$$

$$\max_{i\in I, j\in J}\{t_{ij} \ :\ x_{ij} > 0\} \leq \varepsilon, \tag{5d}$$

$$(\mathbf{x}_i, y_i) \in \mathcal{X}_i, \quad \forall i \in I \tag{5e}$$

$$y \in \mathcal{Y}. \tag{5f}$$

The program (5a)–(5f) differentiates from the general facility location problem (1) only in the non–linear constraint (5d). The presence of this constraint might ruin the structure of the problem and, furthermore, the program cannot be handed directly to a MILP–solver.

Fortunately, for the BO–CBLP the ε–constraint (5d) can be taken into account by simple variable elimination:

Lemma 1. *The constraints*

$$x_{ij} = 0, \quad \forall i \in I,\ j \in J : t_{ij} > \varepsilon \tag{6}$$

are equivalent to the constraint $\max_{i\in I, j\in J}\{t_{ij} \ :\ x_{ij} > 0\} \leq \varepsilon$*.*

This variable elimination, constraints (6), can easily be implemented while maintaining the structure of the general facility location problem, simply by changing the cost matrix as follows

$$c_{ij} = \begin{cases} c_{ij}, & \text{if } t_{ij} \leq \epsilon \\ M, & \text{otherwise,} \end{cases}$$

where M is a sufficiently large number. With this transformation of the cost matrix, the ε–constrained facility location problem, (5a)–(5f), can be solved as an ordinary facility location problem as links (i, j) will not be used unless the travel time on the arc is less than or equal to ε. This leads to the ε–constrained algorithm for the BO-CBLP problem given in Algorithm 2. The adaptation of the generic ε–constrained algorithm to the BO–CBLP happens in *Step 1* and *Step 2* of Algorithm 2. Instead of solving an ε–constrained problem in *Step 1*, we simply solve the single objective *cost modified* facility location problem which preserves its structure. In *Step 2* the ε–constraint is "added" to the model by changing the cost matrix. Note that we do not need to explicitly define the parameter γ as strict improvement is ensured by eliminating all x_{ij}–variables which does lead to an improvement of the bottleneck objective.

Input: Cost matrix c_{ij} and travel time matrix t_{ij}.
Output: A solution $(\mathcal{C}_{\min}, \mathcal{Z}_N)$ to the BO-CBLP problem (3).

<u>*Step 0:*</u> (Initialization) Set $\mathcal{C}_{\min} = \emptyset$, $\mathcal{Z}_N = \emptyset$, $\varepsilon = \infty$, $k = 1$, and $M = \infty$.

<u>*Step 1:*</u> (Subproblem) Solve the facility location problem (1) with assignment cost matrix c. Let (x^k, y^k) be an optimal solution and set $z_1^k = \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij}^k + \sum_{i \in I} f_i y_i^k$ and $z_2^k = \max_{i \in I, j \in J} \{t_{ij} : x_{ij}^k > 0\}$. If $z_2^k = M$ go to *Step 3* else go to *Step 2*.

<u>*Step 2*</u> (Update) Set $\mathcal{C}_{\min} = \mathcal{C}_{\min} \cup \{(x^k, y^k)\}$, $\mathcal{Z}_N = \mathcal{Z}_N \cup \{(z_1^k, z_2^k)\}$ and

$$\varepsilon = \max_{i \in I, j \in J} \{t_{ij} : t_{ij} < z_k^2\}.$$

Update the cost matrix according to

$$c_{ij} = \begin{cases} c_{ij}, & \text{if } t_{ij} \leq \varepsilon \\ M, & \text{otherwise,} \end{cases}$$

and set $k = k + 1$. Return to *Step 1*.

<u>*Step 3*</u> Remove dominated solutions from $\mathcal{C}_{\min}$ and their outcome vectors from $\mathcal{Z}_N$ and return $(\mathcal{C}_{\min}, \mathcal{Z}_N)$ as an optimal solution.

Algorithm 2: Summary of the ε–constrained method applied to the BO–CBLP.

Rather intriguingly, this implies that a specialized algorithm for the single objective minisum facility location problem (1) can be used to solve the subproblems arising in *Step 1* of Algorithm 2, often resulting in quite large problem instances solved relatively fast.

Furthermore, as the bottleneck objective $\max_{i\in I, j\in J}\{t_{ij} \,:\, x_{ij} > 0\}$ attains at most $|I| \times |J|$ different values, we have Proposition 3:

Proposition 3. *The ε–constraint algorithm in Algorithm 2 solves at most* $(|I| - 1) \times |J|$ *single objective facility location problems.*

Proof. The maximum number of problems solved is reached if the algorithm fixes only one x_{ij}–variable in each iteration. However, each customer point $j \in J$ must have at least one possible assignment for the problem to be feasible. This implies that the algorithm performs at most $(|I| - 1) \times |J|$ iterations. □

Corollary 1. *The BO–CBLP is $\mathcal{NP}$–hard if and only if the single objective facility location problem* (1a)–(1e) *is $\mathcal{NP}$–hard.*

Proof. The $\Leftarrow$ direction of the bi–implication follows immediately. The converse follows from the fact that if the single objective facility location problem (1a)–(1e) can be solved in polynomial time, then so can the BO–CBLP by Proposition 3. □

4.2 Solving a lexicographic BOCO problem

In this section we describe how a lexicographic optimization problem can be dealt with (the problem is also discussed by Ralphs, Saltzman, and Wiecek (2006) who apply a weighted Chebyshev norm approach). To make the exposition as general as possible, we use the terminology of the general BOCO problem (4).

One simple possibility is to employ a scalarization of the BOCO that only puts little weight on the second objective, obtaining the scalarized problem

$$\min\{\lambda f_1(x) + (1-\lambda) f_2(x) \,:\, x \in \mathcal{X}, f_2(x) \le \varepsilon\}$$

where λ is very close to one. This approach does, however, suffer from "bad scaling" of the objective function coefficients which often leads to problems that are very hard to solve and numerically unstable.

Another approach is to implicitly enumerate all optimal solutions to the ε–constrained problem

$$\min\{f_1(x) \,:\, x \in \mathcal{X}, f_2(x) \le \varepsilon\}. \tag{7}$$

This can be done by modifying the branch–and–cut algorithm used to solve the ε–constraint problem. To that end, let x^* be the current incumbent of the ε–constrained problem and let $z_1^* = f_1(x^*)$ be the corresponding solution value. A node in the branching tree is then pruned only if it is infeasible or if it shows a lower bound *strictly* greater than z_1^*. Conversely, if the lower bound of the node equals z_1^* then the first objective cannot strictly improve in subsequent subproblems. However, there might exist solutions improving the second objective. Solutions in the subsequent subproblems need to strictly improve the second objective in order to improve the solution and therefore the constraint $f_2(x) \leq f_2(x^*) - \gamma$ can be used as a local cutting plane (or branching constraint).

The incumbent is updated during the modified branch–and–cut algorithm whenever a feasible solution, $\bar{x}$, is found such that $f_1(\bar{x}) < z_1^*$ or (non–exclusively) such that $f_1(\bar{x}) \leq z_1^*$ *and* $f_2(\bar{x}) < f_2(x^*)$.

This way, all optimal solutions to (7) are implicitly enumerated and the lexicographically best solution is found. This guarantees that the solution returned is efficient.

4.2.1 Lexicographic branch–and–bound applied to the BO–CBLP problem

The lexicographic combinatorial optimization problem that needs to be solved in each iteration of the ε–constrained algorithm in Algorithm 2 is

$$\begin{aligned} \operatorname{lex\,min} \quad & \left(\sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} + \sum_{i \in I} f_i y_i \ , \ \max\{t_{ij} \ : \ x_{ij} > 0\} \right) \\ \text{s.t.:} \quad & \sum_{i \in I} x_{ij} = 1, \quad \forall j \in J, \\ & x_{ij} - y_i \leq 0, \quad \forall i \in I, \ j \in J, \\ & \max\{t_{ij} \ : \ x_{ij} > 0\} \leq \varepsilon, \\ & (\mathbf{x}_i, y_i) \in \mathcal{X}_i, \quad \forall i \in I, \\ & y \in \mathcal{Y}. \end{aligned}$$

The branch–and–cut algorithm used to solve this problem is implemented as a best first search. A node in the branching tree is pruned if the node is LP–infeasible or if a lower bound, say LB, is strictly greater than the incumbent, z_1^* (tolerances are used to ensure numerical stability). If, on the other hand, $LB \leq z_1^*$, it is necessary to distinguish between two different cases.. In the first case, when the node is integer feasible, a single child node is created by adding the branching constraints $x_{ij} = 0$ for all $i \in I$ and $j \in J$ where $t_{ij} \geq \max_{i \in I, j \in J}\{t_{ij} \ : \ x_{ij}^* > 0\}$. In the second case, when the node shows a fractional LP solution, the node is separated using a variable dichotomy by selecting an integer infeasible variable and creating two child nodes: one forcing the variable to zero, the other to one. The constraints described above that force x_{ij} to zero in order to improve the bottleneck objective are added to both child nodes as well. To determine the branching variable, we adopt the *Pseudocost branching* rule described in the excellent paper by Achterberg, Koch, and Martin (2005).

5 Computational results

The purpose of the computational tests is fourfold: first, we examine whether it is worthwhile to compute a lexicographically optimal solution to the subproblems arising in the ε–constraint algorithm. Secondly, we examine the extent to which the proposed ε–constrained algorithm is efficient for solving different BO–CBLP problems. Next, we test if the methodology is appropriate for solving these problems. This, is done by comparing the ε–constrained method to an implementation of the two–phase method. And finally, a customized solver for the single source capacitated facility location problem is used to test to which extent such a solver can be used to speed up the computation of the non–dominated set. In total, 1398 different problems have been solved and four algorithmic approaches have been tested.

All algorithms have been coded in C++11 and compiled using the GNU GCC compiler with optimization option O3. All the experiments were carried out on a Fujitsu Esprimo Q920 desktop with 16GB RAM and a 2.20 GHz Intel Core i7-4785T processor running a 64 bit version of Linux Ubuntu. The lexicographic branch–and–cut algorithm described in Sections 4.2 and 4.2.1 was coded using the branch– and incumbent callbacks of the C++ API of CPLEX concert technology. The integer feasible solutions are kept in an external data-structure and CPLEX is then told to reject all solutions such that it does not terminate when a zero gap is obtained. The ParallelMode switch in CPLEX is set to *deterministic*. The absolute and the relative optimality gaps are set to 0.0. All other parameters are at their default values. The code, instances, and detailed results for each instance are all publicly available (Gadegaard, Klose, and Nielsen, 2016).

To ease the reading, we summarize our test statistics in Table 1. For all tests except the ones carried out with the lexicographic branch–and–bound algorithm, we used a time limit of one hour for the generation of the non–dominated frontier.

5.1 Test classes

We report the results of tests conducted on three different classes of facility location problems known from the literature. The problem classes are the uncapacitated facility location problem, and the capacitated facility location problem with and without single-source constraints. In the following, we give a short description of the three problem classes.

5.1.1 The capacitated facility location problem

The capacitated facility location problem (CFLP) is a widely studied combinatorial optimization problem. It consists of opening a set of facilities and assigning demand points to these facilities in such a way that all capacities are respected and the cost of assigning demand points and opening facilities is minimized. The CFLP assumes that

1. Each demand point has a fixed and known demand, $d_j > 0$.
2. Each facility has a fixed and known capacity, $s_i > 0$, which must be respected.
3. Each customer does not need to be assigned to a single facility; its demand can be split between several open facilities.
4. The cost of assigning a customer j to facility i depends linearly on the fraction of the demand d_j transported on the link (i, j).

Table 1: Description of the column headings

Heading	Description
#	Number of instances over which the averages are taken in the corresponding row.
Size	Displays the size of the instances of the corresponding row as $\lvert I\rvert \times \lvert J\rvert$.
Time	Reports the average time consumption in CPU seconds for calculating the entire frontier.
% times	Average of the percentage $\frac{\text{CPU time}[\varepsilon-\text{alg}]}{\text{CPU time [two-phase method]}}100$.
Δz_2	Reports the average difference in the second coordinate of the lexicographic minima. Note that $\Delta z_2 + 1$ gives an upper bound on the number of non–dominated outcomes.
$\lvert\mathcal{Z}_N\rvert$	Reports the average number of non–dominated solutions for the instances of the corresponding row.
N	Contains the average number of dominated solutions generated by the algorithm. That is, the number of *unnecessary* iterations of the ε–constrained algorithm.
$\lvert\mathcal{Z}_N\rvert^{P2}/\lvert\mathcal{Z}_N\rvert^{P1}$	Reports the average ratio between the points found in phase two of the two–phase method and those found in phase one.
Superscript	Indicates the number of instances of that particular (row,column)–combination could be solved within one hour of computation time. If no superscript, all instances were solved.
—	Indicates that none of the instances of that particular (row,column)–combination could be solved within one hour of computation time.

This leads to the set of possible assignments, $\mathcal{X}_i$, being defined as

$$\mathcal{X}_i = \{(\mathbf{x}_i, y_i) \in [0,1]^{|J|} \times \{0,1\} \ : \ \sum_{j \in J} d_j x_{ij} \leq s_i y_i\}$$

The set $\mathcal{Y}$ is equal to $\{y \in \{0,1\}^{|I|} \ : \ \sum_{i \in I} s_i y_i \geq \sum_{j \in J} d_j\}$. The constraints $x_{ij} - y_i \leq 0$ and $\sum_{i \in I} s_i y_i \geq \sum_{j \in J} d_j$ are implied by the other constraints, but they often strengthen the LP relaxation. We denote the BO–CBLP arising from the CFLP the *capacitated* bi–objective cost-bottleneck location problem (capacitated BO–CBLP). The instances for the capacitated BO–CBLP are a subset of the test bed created for the paper Klose and Görtz (2007) consisting of 45 instances ranging from 100 customers and 100 facility sites to 200 customers and 200 facility sites.

5.1.2 The uncapacitated facility location problem

The uncapacitated facility location problem (UFLP) is very similar to the CFLP. As the name suggests it is a version of the CFLP where all facilities have sufficient capacity to potentially serve all demand points. Therefore, no additional restrictions on the assignments are needed and the UFLP can be defined by the sets $\mathcal{X}_i$ given by

$$\mathcal{X}_i = \mathbb{R}^{|J|+1}.$$

As there is no restriction on the number of facilities to be open in a feasible solution either, we have $\mathcal{Y} = \{0,1\}^{|I|}$. We denote the BO–CBLP arising from the UFLP the *uncapacitated* bi–objective cost–bottleneck location problem (uncapacitated BO–CBLP). For the uncapacitated BO–CBLP we use a slightly larger subset of the instances created by Klose and Görtz (2007) consisting of 60 instances ranging in size from 100 customers and 100 facilities to 100 customers and 500 facility sites.

5.1.3 The single source capacitated facility location problem

The single source capacitated facility location problem (SSCFLP) is also a variant of the CFLP where each demand point has to be assigned to exactly one open facility. The SSCFLP can be described in terms of the general facility location problem (1) by setting

$$\mathcal{X}_i = \{(\mathbf{x}_i, y_i) \in \{0,1\}^{|J|+1} \ : \ \sum_{j \in J} d_j x_{ij} \leq s_i y_i\},$$

and the extra requirement on the y-variables is given by

$$\mathcal{Y} = \{y \in \{0,1\}^{|I|} \ : \ \sum_{i \in I} s_i y_i \geq \sum_{j \in J} d_j\}.$$

Note that the constraint defining the set $\mathcal{Y}$ and the constraints $x_{ij} - y_i \leq 0$ are redundant as was the case for the capacitated BO–CBLP. Note that even the feasibility problem for the SSCFLP is $\mathcal{NP}$–hard, and that the constraint $\sum_{i \in I} s_i y_i \geq \sum_{j \in J} d_j$ is not a sufficient condition for a feasible solution to exist. We denote the resulting bi–objective location problem the *single source capacitated* bi–objective cost–bottleneck location problem (single source capacitated BO–CBLP).

In order to test the algorithms, we include the test bed proposed in Holmberg, Rönnqvist, and Yuan (1999) that consists of 71 instances as well as 57 instances from the testbed used in Díaz and Fernández (2002).

5.2 Cost structures

We use the cost matrix and cost vector provided by the instances for the travel costs and the fixed opening costs, respectively. Regarding the travel times we generate three new instances for each instance with travel costs defined as follows:

1. $t_{ij} = c_{ij}$. This suggests that travel time is equivalent to travel distance/cost. A plot of this cost structure, here referred to as C1, is provided in Figure 1(a).

2. $t_{ij} = \max\{0, c_{ij} + U(-d, d)\}$. Here $U(k_1, k_2)$ denotes a discrete uniform distribution on the interval $[k_1, k_2]$. This implies that travel times are positively correlated with the travel cost, but that there is some *noise* which increases (decreases) the travel time for some distances. This is, for example, the case in geographically challenging countries like Denmark. A plot of cost structure C2 is given in Figure 1(b).

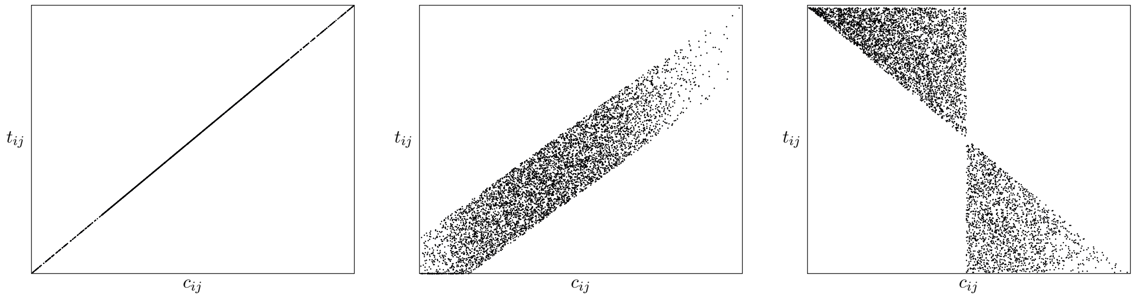
3. Finally, we generate the t_{ij}'s such that $\operatorname{corr}(t_{ij}, c_{ij}) < 0$, suggesting that large values of c_{ij} lead to small values of t_{ij} and vice versa. Such instances occur in e.g. supplier selection problems (see Current and Weber (1994) for more on location problems used in supplier selection problems). One can think of c_{ij} as the cost of procuring the required amount per period of product j from supplier i, where f_i is a fixed cost of placing an order, whilst t_{ij} is the delivery time. This model then minimizes total cost and the time until the last order arrives. We have generated the travel times in the following way: let $C_{\max}$ and $C_{\min}$ be the largest and the smallest assignment costs, respectively. Then

$$\begin{aligned} c_{ij} &< \frac{C_{\max} - C_{\min}}{2} \Rightarrow t_{ij} = U(C_{\max} + C_{\min} - c_{ij}, C_{\max}) \\ c_{ij} &\geq \frac{C_{\max} - C_{\min}}{2} \Rightarrow t_{ij} = U(C_{\min}, C_{\max} + C_{\min} - c_{ij}) \end{aligned}$$

   The cost structure C3 is illustrated in Figure 1(c).

5.3 Performance of the lexicographic branch–and–bound approach

We carried out the experiments for the lexicographic branch–and–bound procedure only for the capacitated BO–CBLP as this problem exhibits characteristics of all three problem



(a) Cost structure C1. (b) Cost structure C2. (c) Cost structure C3.

Figure 1: A plot of the three different cost structures for the assignment of customers to facilities.

Table 2: Results obtained using the lexicographic branch–and–bound algorithm.

		Time		
#	Size	C1	C2	C3
$R = 3$				
5	100x100	1101.34	1865.48	—
5	100x200	1945.25	—	—
$R = 5$				
5	100x100	5242.64	—	5817.79
$R = 10$				
5	100x100	7274.67	—	6137.56

$R = \sum_{i \in I} s_i / \sum_{j \in J} d_j$
Ci = cost structure $i = 1, 2, 3$

classes, and only the locational decisions, the y–variables, need to be integer. We found that only some of the smaller problems could be solved directly using the lexicographic branch–and–bound algorithm. As can be seen in Table 2 we succeeded in solving some problems with 100 facility sites and up to 200 customers. It is quite obvious that when the ratio between the total capacity and total demand becomes larger, the lexicographic branch–and–bound algorithm becomes more time consuming and less effective. This seems to be due to the larger number of feasible solutions to the problems, as the algorithm has a very hard time fathoming branching nodes after branching on integer feasible nodes. When we increase the size of the instances, the time consumption increases drastically. We were not able to solve larger instances as the branching tree became too large to fit in memory. The implemented lexicographic branch–and–bound algorithm thus seems unsuitable as a solution procedure for these problem types. Even though no weakly efficient solutions are generated, enumerating all optimal solutions of the subproblems of the ε–constrained algorithm becomes prohibitive even for smaller problems. Therefore, we did not perform

Table 3: Summary of the results obtained for the capacitated BO–CBLP.

		Time			Δz_2			$\lvert\mathcal{Z}_N\rvert$			N		
#	Size	C1	C2	C3	C1	C2	C3	C1	C2	C3	C1	C2	C3
$R=3$													
5	100×100	17.91	81.24	45.35	34.80	117.20	32.60	8.20	43.00	33.60	0.00	0.00	0.00
5	100×200	68.92	243.62	130.02	29.40	74.40	31.00	13.40	44.80	32.00	0.00	0.00	0.00
5	200×200	291.73	775.31	285.59	37.00	91.60	34.80	18.80	53.40	35.80	0.00	0.00	0.00
$R=5$													
5	100×100	52.24	150.11	85.05	53.00	90.40	36.40	20.80	50.60	35.40	0.00	0.00	0.00
5	100×200	122.42	569.50	353.63	39.80	117.00	33.80	19.40	64.80	34.80	0.00	0.00	0.00
5	200×200	437.54	1163.85	425.62	39.60	82.60	32.40	21.20	60.80	33.40	0.00	0.00	0.00
$R=10$													
5	100×100	67.89	166.90	213.00	53.00	82.60	37.20	23.80	38.20	38.00	0.00	0.00	0.00
5	100×200	231.57	893.59[4]	2521.07[3]	67.60	82.60	32.20	34.00	51.80	33.20	0.00	0.00	0.00
5	200×200	384.43	1070.58	584.13	55.80	111.20	27.60	22.00	64.20	36.00	0.00	0.80	0.00

$R = \sum_{i\in I} s_i / \sum_{j\in J} d_j$
Ci = cost structure $i = 1, 2, 3$

further tests with the lexicographic branch–and–bound algorithm, and the following results are carried out using CPLEX as a single objective solver for the subproblems.

5.4 Performance of the ε–constrained algorithm

In this section we report the results obtained with the ε–constrained algorithm using CPLEX as a single objective solver for the subproblems arising in the ε–algorithm described in Algorithm 2.

5.4.1 Results for the capacitated BO–CBLP

We report the results obtained when applying the ε-algorithm to the 45 instances in the capacitated BO-CBLP class that could be solved within one hour of computation time. Table 3 shows that problems of up to 200 facilities and 200 customers could be solved. Furthermore, the number of non-dominated solutions does not vary much within each cost structure, but substantial variations are seen between cost structures: when considering the cost structure C2 many more non–dominated points are generated compared to C1 and C3. Intuitively, one would expect cost structure C3 to produce more non–dominated points as the coefficients are negatively correlated. However, when we divide the range of the coefficients t_{ij} into three equal segments, it turns out that for the cost structure C3, about two thirds of the entries in the travel time matrix lie in the largest third. This means that many variables are fixed in the first couple of iterations. Thus, producing solutions improving the center/bottleneck objective quickly becomes infeasible, implying smaller values of $|\mathcal{Z}_N|$ for this cost structure. By comparison, for cost structures C1 and C2, about 60 percent of the travel time coefficients lie in the middle third. This explains why the cost structure C3 produces less efficient solutions than expected.

Another interesting point in relation to cost structure C3 is that in many cases the upper bound on the number of non–dominated outcomes given by $\Delta z_2 + 1$ is strict. That is, there

Table 4: Summary of the results obtained for the uncapacitated BO–CBLP.

#	Size	Time			Δz_2			$\lvert\mathcal{Z}_N\rvert$			N		
		C1	C2	C3	C1	C2	C3	C1	C2	C3	C1	C2	C3
15	100×100	19.73	55.96	30.18	84.53	137.87	34.73	27.00	57.33	35.20	0.07	0.40	0.00
15	100×200	74.71	227.93	145.45	77.33	122.20	32.33	27.67	62.87	33.33	0.00	0.33	0.00
15	200×200	459.46	1190.83	791.66	67.80	117.93	34.07	39.27	82.00	35.07	0.00	0.53	0.00
15	500×500	402.88	1464.44	1043.93	25.47	76.73	29.33	23.13	69.73	30.33	0.00	0.40	0.00

Ci = cost structure $i = 1, 2, 3$

is almost always a non–dominated solution for each value between $z_2^{\min} = \min\{z_2 : z \in \mathcal{Z}_N\}$ and $z_2^{\max} = \max\{z_2 : z \in \mathcal{Z}_N\}$ of the bottleneck objective. This seems to be due to the antagonistic relationship between the objectives when the coefficients c_{ij} and t_{ij} are negatively correlated: The cost objective tries to pick the assignments with small values of c_{ij} resulting in long travel times for that assignment.

Note that strikingly few weakly efficient solutions are generated for all of the three cost structures. The low numbers of weakly efficient solutions confirm the suitability of the ε–constraint approach for these cost–bottleneck location problems.

5.4.2 Results for the uncapacitated BO–CBLP

We were able to solve slightly larger problems of the uncapacitated BO–CBLP class compared to the capacitated version. This means that instances ranging from 100 facilities and 100 customers to instances with 100 facilities and 500 customers were solved to optimality. As there is no capacity limit on the facilities in this problem type, the instances are grouped by size only.

In Table 4, we see the same pattern as for the capacitated BO–CBLP, namely that the cost structure C2 yields larger values of $|\mathcal{Z}_N|$. And again, the number of non–dominated outcomes in cost structure C3 reaches the upper bound $\Delta z_2 + 1$ in most of the instances. Furthermore, a slightly higher number of non–dominated solutions is generated compared to the capacitated BO–CBLP. This phenomenon was expected as there is no capacity constraints to conflict with the fixation of assignment variables.

The reader should again note the remarkably small number of weakly efficient solutions which underpins the appropriateness of the ε–constraint approach.

5.4.3 Results for the single source capacitated BO–CBLP

The 71 problems from Holmberg et al. (1999) are divided into five subsets and the 57 instances of Díaz and Fernández (2002) are divided into 7 subsets based on the dimensions of the problems. Table 5 summarizes the results obtained for the single source capacitated BO–CBLP.

As opposed to the capacitated and the uncapacitated cost-bottleneck location problems, the size of $\mathcal{Z}_N$ is significantly larger for the instances with negatively correlated travel costs and travel times compared to the two other cost structures. The number of weakly efficient

Table 5: Summary of the results obtained for the single source capacitated BO–CBLP.

		Time			Δz_2			$\lvert\mathcal{Z}_N\rvert$			N		
#	Size	C1	C2	C3	C1	C2	C3	C1	C2	C3	C1	C2	C3
Intances from (Holmberg et al., 1999)													
12	10×50	0.13	0.42	8.75	60.42	161.83	574.25	3.75	9.92	137.33	0.00	0.08	0.08
12	20×50	0.45	1.08	44.21	93.17	142.00	571.25	12.08	16.00	172.33	0.00	0.00	0.33
16	$10 \times 90^{\dagger}$	0.93	3.50	46.22	53.20	81.27	143.93	10.27	15.40	93.53	0.00	0.07	0.47
15	30×150	0.80	10.42	16.87	0.00	162.06	42.38	1.00	25.19	43.38	0.00	0.44	0.00
16	30×200	13.10	41.22	957.86^{15}	88.44	176.00	335.69	23.25	53.00	313.80	0.19	0.13	0.27
Instances from Díaz and Fernández (2002)													
6	10×20	0.48	0.72	4.01	31.83	43.00	83.33	4.33	7.67	33.67	0.00	0.00	0.17
11	15×30	2.19	3.02	16.67	32.27	39.36	88.82	10.45	13.45	45.73	0.00	0.09	0.18
8	20×40	4.93	7.12	51.78	25.38	34.25	93.29	12.00	15.00	59.25	0.00	0.00	0.00
8	20×50	372.47	353.99	45.52^{6}	29.50	40.00	92.33	12.50	18.25	64.33	0.13	0.13	0.67
8	30×60	27.41	49.62	153.80	33.25	39.88	94.63	17.67	23.78	74.11	0.22	0.44	0.33
8	30×75	84.55^{7}	225.38^{7}	1407.9^{7}	28.00	44.50	94.60	16.00	28.14	78.43	0.14	0.00	0.57
8	30×90	492.08	706.92^{6}	220.43^{6}	25.38	30.50	95.33	15.00	17.00	81.17	0.13	0.17	0.17

† These instances ranges from 10×90 to 30×70.
Ci = cost structure $i = 1, 2, 3$

solutions described in column N is again remarkably small, which suggests that the problem instances for the single source case also only have a few alternative optimal solutions and that, therefore, the ε–constraint method is a well–suited approach.

It is interesting to note, that for the Holmberg et al. instances of size 30×150, only one efficient solution is found for all of the 15 instances with cost structure C1. This happens, because the c_{ij}'s are relatively large compared to the fixed opening costs, f_i, leading to more facilities being open in an optimal solution to the minisum problem. As the travel time coefficients $t_{ij} = c_{ij}$ in cost structure C1, the cost objective and the bottleneck objectives are not in conflict at all, and hence, the optimal solution to the cost objective, turns out to be optimal for the bottleneck objective as well.

5.5 Comparison with the two–phase method

In order to validate the effectiveness of the ε–constrained method proposed in this paper, we have implemented a two–phase method for solving the BO–CBLP as well. We have chosen to implement this solution methodology as it is probably the most widely used solution method for bi–objective combinatorial optimization next to the ε–constrained method (for a thorough treatment of two–phase methods the reader is referred to Przybylski, Gandibleuz, and Ehrgott (2011) and references therein).

In the first phase of the two–phase method, the set $\mathcal{Z}_{sN}$ is generated by solving weighted sum scalarized versions of the BO–CBLP for different weight vectors. Phase two consists of a method capable of generating the remaining non–dominated outcomes, $\mathcal{Z}_{nN}$. We have implemented the so–called "Perpendicular Search Method" (PSM) suggested by Chalmet, Lemonidis, and Elzinga (1986) for generating the non–extreme supported non–dominated outcomes. Contrary to the ε–constrained method, the PSM algorithm computes no weakly non–dominated solutions. However, similar to the ε–constrained method the implementation is straightforward.

To meet the potential critique that the two–phase method is badly implemented, we here mention that between 99.6% and 100.0% of the running time was spent by CPLEX solving the subproblems. We also note that a first implementation where the second phase consisted of ranking solutions using no–good inequalities performed very poorly. Thus, we chose to implement the PSM method instead.

Using the two–phase method and the PSM method requires the non–linear min–max objective to be linearized. For the SSCFLP and the UFLP this is easily done by replacing the objective with a new continuous variable ρ and including the set of constraints:

$$\rho \geq \sum_{i \in I} t_{ij} x_{ij}, \quad \forall j \in J.$$

For the CFLP, where the x_{ij}–variables are continuous, we need to introduce a set of new binary variables ξ_{ij} equaling one if and only if $x_{ij} > 0$. The following constraints are then added to the formulation of the CFLP:

$$\begin{aligned} \rho &\geq t_{ij}\xi_{ij}, \quad \forall j \in J, \\ \xi_{ij} &\geq x_{ij}, \quad \forall i \in I,\ j \in J, \\ \xi_{ij} &\in \{0,1\}, \quad \forall i \in I,\ j \in J. \end{aligned}$$

The results obtained with the two–phase method has been aggregated in Table 6. If the algorithm failed to solve all instances corresponding to a row in Table 6 within one hour of computation time, the number of actually solved instances is indicated in the "%–time" columns using superscript. It is obvious that the overall performance of the two-phase method is very poor compared to the ε–constrained method. In fact, the ε–constrained method is 2 to 160 times faster than the two–phase method on average. For the uncapacitated as well as for the capacitated BO–CBLP, the relative performance across the cost structures seems stable. However, for the single source capacitated BO–CBLP, the ε–constraint algorithm becomes better relative to the two–phase method when the coefficients c_{ij} and t_{ij} become negatively correlated. The explanation for this behavior is easily found in the columns entitled "$|\mathcal{Z}_N|^{P2}/|\mathcal{Z}_N|^{P1}$". These columns display the ratio between the number of solutions found in phase one and phase two, respectively. This ratio increases significantly for most of the instances in cost structure C3 indicating that only few extreme supported non–dominated outcomes exists for these problems. As this ratio grows, the two–phase method loses its power as the first phase cannot divide the search space into sufficiently small regions for the second phase to be effective.

Furthermore, it was only possible to solve the small instances of the capacitated BO–CBLP and the uncapacitated BO–CBLP. In particular, we cannot solve many of the capacitated BO–CBLP within one hour of computation time. This is mainly due to the linearization of the objective function which requires the introduction of $|I| \times |J|$ new binary variables as well as $|I| \times |J|$ new constraints. In the linearization of the min–max objective in the uncapacitated and the single source capacitated BO–CBLP, only one additional continuous variable and $|J|$ new constraints are needed. Therefore these problems scale

Table 6: Comparison of the results obtained with the two phase method and the ε–constrained method.

#	Size	% time			$\lvert\mathcal{Z}_N\rvert^{P2}/\lvert\mathcal{Z}_N\rvert^{P1}$		
		C1	C2	C3	C1	C2	C3
Capacitated BO–CBLP							
$R = 3$							
5	100×100	0.62^{3}	0.47^{1}	—	1.16	1.83	—
$R = 5$							
5	100×100	—	—	—	—	—	—
$R = 10$							
5	100×100	1.84^{1}	3.83^{1}	—	1.22	1.20	—
Uncapacitated BO–CBLP							
15	100×100	3.55	3.56	3.60	2.15	1.96	0.95
15	100×200	3.33	4.18	2.76	1.61	2.03	0.43
Single source capacitated BO–CBLP							
Intances from (Holmberg et al., 1999)							
12	10×50	16.58	18.59	4.75	0.32	1.11	49.72
12	20×50	11.33	17.07	2.38	0.83	1.67	70.13
16	$10 \times 90^{\dagger}$	17.58	12.80	2.07	0.96	1.60	40.15
15	30×150	38.07	2.91	1.03	0.00	2.46	8.16
16	30×200	1.19	1.12	—	2.27	4.25	—
Intances from (Díaz and Fernández, 2002)							
6	10×20	8.51	8.47	10.46	0.25	0.62	5.63
11	15×30	14.63	15.85	4.72	0.96	1.53	7.61
8	20×40	0.72	0.84	3.99^{5}	0.97	1.03	6.79
8	20×50	12.01^{6}	15.82^{5}	8.97^{3}	1.20	1.71	6.98
8	30×60	55.96^{5}	22.55^{4}	13.12^{3}	1.05	1.18	4.90
8	30×75	—	—	—	—	—	—
8	30×90	—	—	—	—	—	—

† These instances ranges from 10×90 to 30×70.
$R = \sum_{i \in I} s_i / \sum_{j \in J} d_j$
Ci = cost structure $i = 1, 2, 3$

Table 7: Summary of the results obtained for the single source capacitated BO–CBLP using a specialized solver.

		Time			$\lvert\mathcal{Z}_N\rvert$			N		
#	Size	C1	C2	C3	C1	C2	C3	C1	C2	C3
12	10×50	0.26	0.74	11.20	3.75	9.50	104.00	0.00	0.00	0.00
12	20×50	1.44	3.79	92.81	1.44	3.79	92.81	0.00	0.08	0.17
16	$10 \times 90^\dagger$	0.73	9.95	12.30	1.00	21.44	22.06	0.00	0.38	0.00
15	30×150	6.45	12.11	36.58	8.40	12.07	53.60	0.00	0.07	0.00
16	30×200	33.02	93.63	1086.10	17.44	39.19	161.88	0.00	0.13	0.06
6	10×20	0.81	1.10	4.12	4.00	6.83	22.50	0.00	0.00	0.00
11	15×30	3.36	4.69	17.75	7.82	9.55	28.45	0.09	0.18	0.18
8	20×40	5.78	6.89	34.33	8.25	11.00	34.75	0.00	0.00	0.13
8	20×50	9.77	14.20	73.01	8.50	11.88	37.25	0.00	0.00	0.13
8	30×60	19.78	202.18	48.49	11.25	13.75	40.88	0.13	0.00	0.25
8	30×75	17.51	23.71	159.28	10.00	15.75	42.50	0.00	0.00	0.00
8	30×90	26.80	33.70	408.78	9.38	12.50	44.25	0.00	0.00	0.13

† These instances ranges from 10×90 to 30×70.
Ci = cost structure $i = 1, 2, 3$

slightly better. It should, however, be very clear, that the two–phase method is also less suited for these problems than the ε–constraint method.

5.6 Utilizing a customized solver for the single source capacitated BO–CBLP

As a quite efficient solver for the single source capacitated facility location problem was available to the authors, we wanted to test the capabilities of the method. The specialized solver is based on a cut–and–solve framework with upper bounds generated using a local branching heuristic. The lower bound of the problem is strengthened by exact knapsack separation (the code is available on request). In Table 7, we report the results obtained using this specialized solver as a black box engine for solving the subproblems arising in each iteration of the ε–constraint algorithm.

With this solver as black box engine, we were easily able to solve the largest instances in both of the testbeds and we saved a significant amount of time on the largest instances. The smaller instances were often solved in less time using CPLEX (see Table 5), however. This basically boils down to our implementation: The implementation using CPLEX fixes the x_{ij}–variables and *resolves* the model. This allows CPLEX to utilize basis, incumbent, and branching information from previous problems. When we use the specialized solver, the problem is solved *from scratch* every time, implying that no information from previously solved problems is used.

6 Conclusion

In this paper we investigated the very general bi-objective cost–bottleneck location problem. We proved that even though the problem is $\mathcal{NP}$–hard, it is in fact tractable, in the sense that the size of the efficient frontier of the bi–objective problem is always limited by a polynomial in the input size.

We proposed a scheme for solving the BO-CBLP that relies on an ε–constrained method. We suggested two ways to accommodate the issues of generating weakly efficient solutions in the ε–constraint: First, a simple change of the cost matrix which is both a necessary and sufficient condition in order to impose the non–linear ε–constraint on the bottleneck objective. This implies, the structure of the underlying location problem is kept, and specialized solvers can be used as a subroutine reducing the computation time. Secondly, we outlined a scheme based on lexicographic branch–and–bound leading to no weakly efficient solutions being generated. Furthermore, the number of iterations performed by the algorithm is bounded by a polynomial in the input size.

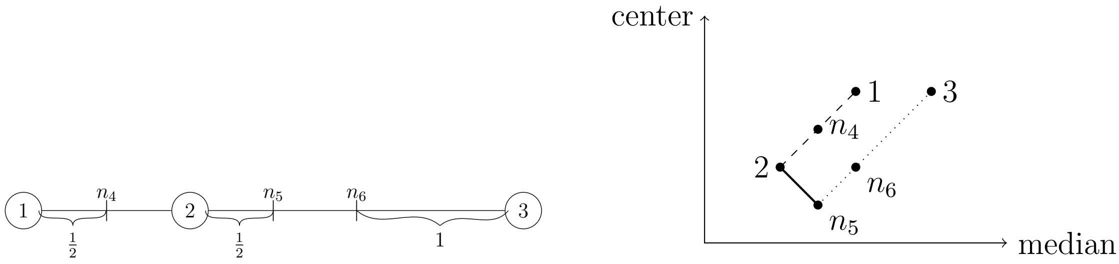
Through extensive computational tests we have shown that the proposed method is capable of efficiently and exactly solving even large BO–CBLPs. In addition, the tests showed that remarkably few weakly non–dominated solutions exist for these very large combinatorial optimization problems. This leads to the lexicographic branch–and–bound based algorithm being very inferior compared to the ε–constrained algorithm. Furthermore, the proposed algorithm outperforms a two–phase method by several orders of magnitude as well.

In the appendix we also linked the BO–CBLP to the p–centdian location problem studied in the literature. We found that the bi–objective p–centdian problem *on a graph* might have an uncountably infinite number of Pareto optimal solutions and noted that given a computationally efficient way of solving the BO–CBLP, the vertex–p–centdian problem can be solved efficiently for all values of the scaling parameter λ. As many very efficient algorithms for the p-median problem exist, and the p–centdian on a graph can be reduced to a discrete problem, our algorithm may constitute an efficient way of solving the p–centdian problem.

Directions for further research include the testing of the approach for the p–centdian problem. That is, to investigate if it is in fact possible to solve large problem instances of the p–centdian problem using an algorithm tailored for the p–median problem. It would also be interesting to apply the methodology on other types of more complex facility location models, such as multi–stage and dynamic models.

7 Acknowledgments

The authors would like to thank Professor Kim Allan Andersen for insightful comments and suggestions. This work was supported by a grant from Købmand Ferdinand Sallings Mindefond.



(a) Illustration of the graph used in the proof of Theorem 2.

(b) Solutions in outcome space for the graph used in the proof of Theorem2.

Figure 2: Illustrations of the example used in the proof of Theorem 2.

A A link to the weighted p–centdian problem

The p–centdian problem is a combination of the p–median and the p–center problems where the objective function is a convex combination of the median/cost and the center/bottleneck objectives. Given a graph $G = (V, E)$, let $P(G)$ be the set of all points on G and $d(l, h)$ be the distance of a shortest path between points l and h on G (note that the points l and h might be *interior* points on the edges of G). Furthermore, given a set of points $S \subseteq P(G)$ we define $d(S, h) = \min\{d(l, h) \ : \ l \in S\}$. In addition, let $w_j \geq 0$ and $v_j \geq 0$ be weights of the node $j \in V$ representing, for example, the number of customers or some other measure of attractiveness of the node j. For $0 \leq \lambda \leq 1$ the p–centdian problem may then be stated as

$$\begin{aligned} &\min \lambda \sum_{j \in V} w_j d(S, j) + (1 - \lambda) \max_{j \in V} v_j d(S, j) \\ &\text{s.t.: } S \subseteq P(G) \\ &\qquad |S| = p \end{aligned} \tag{8}$$

It has been shown in Pérez-Brito et al. (1997) that there exists a finite set of points on G containing an optimal solution to the p–centdian. In what follows, the problem

$$\min\{\left(\sum_{j \in V} w_j d(S, j), \max_{j \in V} v_j d(S, j) \right) \ : \ S \subseteq P(G), \ |S| = p\}$$

is denoted the bi–objective p–centdian problem. It turns out that there can be infinitely many Pareto optimal solutions to this problem.

Theorem 2. *The set of Pareto optimal solutions to the bi–objective p–centdian problem on a network can be uncountably infinite even for $p = 1$ when $S \subseteq P(G)$.*

Proof. We show the result by giving an example having this property. Consider the graph given in Figure 2(a), where the edge lengths are given by $d(1, 2) = d(2, 1) = 1$ and $d(2, 3) = d(3, 2) = 2$. Furthermore, suppose $p = 1$, that is, we want to place one facility on

G. The intersection points on G are denoted by n_4, n_5, and n_6. It is easily verified that locating a facility at node 2 is an optimal solution to the 1–median problem with outcome vector $(3, 2)$ whereas placing a facility at n_5 is an optimal solution for the 1–center problem, resulting in the outcome vector $(3.5, 1.5)$. As the solutions are unique optimal solutions they are *efficient*. In Figure 2(b) the outcome vectors for all points on the graph G have been plotted. All points on the edge $(1, 2)$ map into the dashed line, while all solutions on the edges (n_5, n_6) and $(n_6, 3)$ map into the dotted line. However, all points on the edge $(2, n_5)$ map into the solid line which is non–dominated. As there are an uncountable infinite number of points on this edge, the result follows. □

Note that the bi–objective p–centdian is *not* covered by the general BO–CBLP as the set I of potential facility sites has to be finite in the definition of the BO–CBLP. However, letting the sets I and J equal the set V, the (vertex) p–centdian problem (8) can be stated as the following *scalarized* version of the BO–CBLP:

$$\begin{aligned}
\min\ & \lambda \left(\sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} + \sum_{i \in I} f_i y_i \right) + (1-\lambda) \max_{i \in I, j \in J} \{ t_{ij} \ : \ x_{ij} > 0 \} \\
\text{s.t.:}\ & \sum_{i \in I} x_{ij} = 1, \quad \forall j \in J, \\
& (\mathbf{x}_i, y_i) \in \mathcal{X}_i, \quad \forall i \in I, \\
& y \in \mathcal{Y},
\end{aligned}$$

where $\mathcal{X}_i = \{(\mathbf{x}_i, y_i) \in \{0,1\}^{|J|+1} \ : \ x_{ij} \leq y_i\}$, $\mathcal{Y} = \{y \in \mathbb{R}^n \ : \ \sum_{i \in I} y_i = p\}$, $f_i = 0$, $c_{ij} = w_j d(i, j)$, and $t_{ij} = v_j d(i, j)$ for all $i \in I$ and $j \in J$. In many practical applications it will suffice to consider placing facilities only at the nodes of the graph. For $0 < \lambda < 1$ a solution to the vertex–p–centdian problem corresponds to a supported efficient solution of the BO–CBLP (3) with the sets I, J, $\mathcal{X}_i$, $\mathcal{Y}$ and the coefficients f_i, c_{ij}, and t_{ij} defined as above. This means that solving the BO–CBLP yields a solution to the vertex–p–centdian problem *for each value of* $\lambda > 0$. Most literature considers special structured graphs such as trees when solving the p–centdian problem. This approach offers a means to solve the vertex–p–centdian problem for all values of $0 < \lambda < 1$ on a general graph with no assumption other than that facilities should be located on nodes only.

References

References

Achterberg, T., Koch, T., Martin, A., 2005. Branching rules revisited. Operations Research Letters 33 (1), 42 – 54.

Balinski, M., 1965. Integer programming: Methods, uses, computations. Management Science 12 (3), 253–313.

Bérubé, J.-F., Gendreau, M., Potvin, J.-Y., 2009. An exact ϵ-constraint method for bi-objective combinatorial optimization problems: Application to the traveling salesman problem with profits. European Journal of Operational Research 194 (1), 39–50.

Chalmet, L., Lemonidis, L., Elzinga, D., 1986. An algorithm for the bi-criterion integer programming problem. European Journal of Operational Research 25 (2), 292 – 300.

Cornuejols, G., Nemhauser, G., L.A., W., 1990. The uncapacitated facility location problem. In: Mirchandani, P., Francis, R. (Eds.), Discrete Location Theory. Wiley–Interscience, New York.

Current, J., Weber, C., 1994. Application of facility location modeling constructs to vendor selection problems. European Journal of Operational Research 76 (3), 387 – 392.

Dearing, P., Newruck, F., 1979. A capacitated bottleneck facility location problem. Management Science 25 (11), 1093–1104.

Díaz, J., Fernández, E., 2002. A branch-and-price algorithm for the single source capacitated plant location problem. Journal of the Operational Research Society 53, 728–740.

Ehrgott, M., 2005. Multicriteria Optimization, 2nd Edition. Springer Berlin, Heidelberg.

Farahani, R., SteadieSeifi, M., Asgari, N., 2010. Multiple criteria facility location problems: A survey. Applied Mathematical Modelling 34 (7), 1689–1709.

Fernández, E., Puerto, J., 2003. Multiobjective solution of the uncapacitated plant location problem. European Journal of Operational Research 145 (3), 509–529.

Filippi, C., Stevanato, E., 2013. Approximation schemes for bi-objective combinatorial optimization and their application to the TSP with profits. Computers & Operations Research 40 (10), 2418 – 2428.

Gadegaard, S., Klose, A., Nielsen, L., 2016. A bi–objective approach to discrete cost–bottleneck location problems. `https://github.com/SuneGadegaard/BOCBLP`, commit: 384088cd374e7629c29dbe97b8d07f0a2c0dfbdb.

Garfinkel, R. S., Rao, M., 1971. The bottleneck transportation problems. Naval Research Logistics Quarterly 18 (4), 465–472.

Hakimi, S., 1965. Optimum distribution of switching centers in a communication network and some related graph theoretic problems. Operations Research 13 (3), 462–475.

Hakimi, S. L., 1964. Optimum locations of switching centers and the absolute centers and medians of a graph. Operations Research 12 (3), 450–459.

Halpern, J., 1976. The location of a center-median convex combination on an undirected tree*. Journal of Regional Science 16 (2), 237–245.

Halpern, J., 1978. Finding minimal center-median convex combination (cent-dian) of a graph. Management Science 24 (5), 535–544.

Hamacher, H., Labbé, M., Nickel, S., 1999. Multicriteria network location problems with sum objectives. Networks 33 (2), 79–92.

Hamacher, H. W., Nickel, S., 1996. Multicriteria planar location problems. European Journal of Operational Research 94 (1), 66–86.

Hamacher, H. W., Pedersen, C. R., Ruzika, S., 2007. Finding representative systems for discrete bicriterion optimization problems. Operations Research Letters 35 (3), 336 – 344.

Hammer, P., 1969. Time minimizing transportation problems. Naval Research Logistics Quarterly 16 (3), 345–357.

Hansen, P., 1980. Bicriterion path problems. In: Fandel, G., Gal, T. (Eds.), Multiple Criteria Decision Making Theory and Application. Vol. 177 of Lecture Notes in Economics and Mathematical Systems. Springer Berlin Heidelberg, pp. 109–127.

Holmberg, K., Rönnqvist, M., Yuan, D., 1999. An exact algorithm for the capacitated facility location problems with single sourcing. European Journal of Operational Research 113 (3), 544–559.

Klose, A., Görtz, S., 2007. A branch-and-price algorithm for the capacitated facility location problem. European Journal of Operational Research 179 (3), 1109–1125.

Mavrotas, G., 2009. Effective implementation of the ε–constraint method in multi-objective mathematical programming problems. Applied Mathematics and Computation 213 (2), 455–465.

Nickel, S., Puerto, J., Rodríguez-Chía, A., 2005. Mcdm location problems. In: Figueira, J., Greco, S., Ehrgott, M. (Eds.), Multiple Criteria Decision Analysis: State of the Art Surveys. International Series in Operations Research and Management Science. Springer, Berlin, Heidelberg, Ch. 19, pp. 761–795.

Pandian, P., Nataraja, G., 2011. A new method for solving bottleneck-cost transportation problems. International Mathematical Forum, Vol. 6, 2011, no. 10, 451 - 460 6 (10), 451–460.

Pérez-Brito, D., Moreno-Pérez, J., Rodríguez-Martín, I., 1997. Finite dominating set for the p-facility cent-dian network location problem. Studies in Locational Analysis 11, 27–40.

Przybylski, A., Gandibleuz, X., Ehrgott, M., 2011. The two phase method for multiobjective combinatorial optimization problems. In: Mahjoub, R. (Ed.), Progress in Combinatorial Optimization. ISTE Ltd and John Wiley Sons Inc, pp. 559–595.

Ralphs, T., Saltzman, M., Wiecek, M., 2006. An improved algorithm for solving biobjective integer programs. Annals of Operations Research 147 (1), 43–70.

Reese, J., 2006. Solution methods for the p-median problem: An annotated bibliography. Networks 48 (3), 125–142.

Revelle, C., Eiselt, H., Daskin, M., 2008. A bibliography for some fundamental problem categories in discrete location science. European Journal of Operational Research 184 (3), 817–848.

Ross, G., Soland, R., 1980. A multicriteria approach to the location of public facilities. European Journal of Operational Research 4 (5), 307 – 321.