

Konrad-Zuse-Zentrum
für Informationstechnik Berlin

Takustraße 7
D-14195 Berlin-Dahlem
Germany

SVENJA USLU AXEL WERNER

**A Two-Phase Method
for the Biobjective k -Architecture
Connected Facility Location Problem
and Hypervolume Computation**

Research supported by DFG within project GR 883/18-1

ZIB Report 15-08 (February 2015)

Herausgegeben vom
Konrad-Zuse-Zentrum für Informationstechnik Berlin
Takustraße 7
D-14195 Berlin-Dahlem

Telefon: 030-84185-0
Telefax: 030-84185-125

e-mail: bibliothek@zib.de
URL: <http://www.zib.de>

ZIB-Report (Print) ISSN 1438-0064
ZIB-Report (Internet) ISSN 2192-7782

A Two-Phase Method for the Biobjective k -Architecture Connected Facility Location Problem and Hypervolume Computation

Svenja Uslu

Axel Werner

February 25, 2015

We apply customized versions of the ε -constraint Method and the Two-Phase Method to a problem originating in access network planning. We introduce various notions of quality measures for approximated/partial sets of nondominated points, utilizing the concept of hypervolume for biobjective problems. We report on computations to assess the performance of the two methods in terms of these measures.

1 Introduction

There are a few general algorithms to solve a *Biobjective Optimization Problem*, i.e. to determine a complete set of nondominated points, also called a *Pareto set*, of a general optimization problem with two objective functions. Arguably the most popular one is the *ε -constraint Method* [6], which works in a very straightforward way and still seems to be among the best biobjective solving methods available. Another algorithm, the *Two-Phase Method* [12], has, to the best of our knowledge, mainly been applied to problems, for which there are efficient (polynomial) algorithms solving the single-objective counterparts; see [10] as well as [7] and the references therein.

In this paper, we apply customized versions of these two algorithms to a problem originating in access network planning, which can be formulated as a biobjective *Integer Linear Program* (ILP). We compare the two algorithms with respect to the general performance, also in terms of the *hypervolume*, which is a general measure for the approximation of nondominated sets of points in the objective space, see [13].

For biobjective mixed integer programming problems, there are some more recent general algorithms [1, 2, 3], which we do not consider here. Throughout the paper, we largely use the standard terminology from the theory multiobjective optimization; as a reference see,

for instance, [5]. We assume a biobjective optimization problem given as a minimization problem

$$\min\{f(\sigma) \mid \sigma \in \mathcal{P}\},$$

where \mathcal{P} denotes the set of feasible solutions and $f : \mathcal{P} \rightarrow \mathbb{R}^2$ with $f(\sigma) = (f_1(\sigma), f_2(\sigma))^T$ represents the two objective functions.

2 Methods and Problem

In the following we give a brief description of both algorithms, the ε -constraint Method as well as the Two-Phase Method, in very general terms. Subsequently we provide a short introduction to the problem we apply the methods to, and finally a more detailed view of how both methods work in the setting of our problem.

2.1 Epsilon-Constraint and Two-Phase Method

Both, the ε -constraint Method and the Two-Phase Method, compute the Pareto set of biobjective problems by solving a number of associated single-objective problems, which are obtained by weighting the two objectives in certain ways.

Definition 1. Let $\min\{f(\sigma) \mid \sigma \in \mathcal{P}\}$ be a biobjective optimization problem and $\lambda \in \mathbb{R}_{\geq}^2$. Then a problem of the form $\min\{\lambda^T f(\sigma) \mid \sigma \in \mathcal{P}\}$ is called a *weighted sum problem*.

Recall that an *efficient solution* for a given biobjective optimization problem is a feasible solution such that no other feasible solution exists that has a smaller or equal value for both objective functions; any efficient solution σ gives rise to a *nondominated point* $f(\sigma)$ in the objective space. There are two distinguished nondominated points: The point $\hat{z}^r = (f_1(\sigma), f_2(\sigma))^T$ for an efficient solution σ with the smallest possible value of f_1 and, analogously, $\hat{z}^s = (f_1(\sigma), f_2(\sigma))^T$ for an efficient solution σ with the smallest possible value of f_2 . These two points also define the *ideal point* $(\hat{z}_1^r, \hat{z}_2^s)$ and the *nadir point* $(\hat{z}_1^s, \hat{z}_2^r)$.

The ε -constraint Method is probably the most straightforward way to solve biobjective problems and works quite well for most problems, in particular those that allow the addition of upper-bound type constraints (cf. [7]). It works by imposing an upper bound ε on one of the objectives and successively reducing ε to obtain new nondominated points. Assumed that the objective f_2 is the one to be bounded, the value for ε in the first iteration can be chosen to be \hat{z}_2^r , or – in case the computation of \hat{z}^r is difficult – any upper bound on f_2 for efficient solutions. In each iteration a weighted sum problem with weight $\lambda = (1, 0)^T$ has to be solved, and the ε for the next iteration set to the f_2 value of the found solution, reduced by a suitably small, positive constant. If we assume that all coefficients and variables in the selected objective function are integral then the reduction in each iteration is usually taken to be the greatest common divisor of these coefficients.

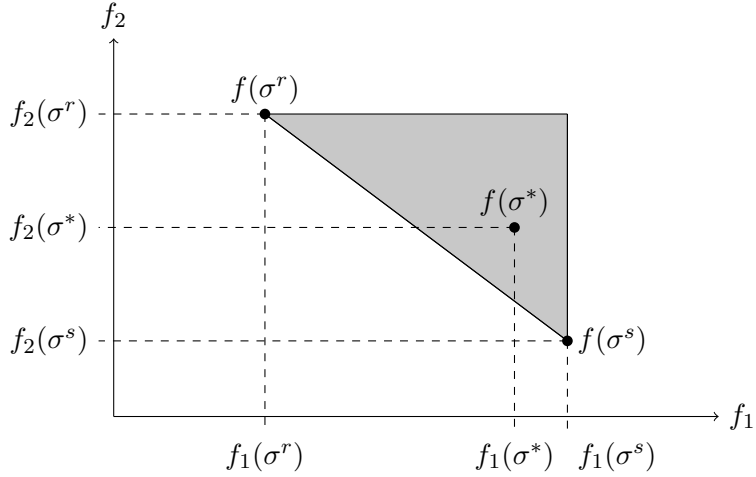


Figure 1: The objective value of a nonsupported efficient solution σ^* is in a triangle defined by the objective values of $\sigma^r, \sigma^s \in S$.

The Two-Phase Method was first described by Ulungu and Teghem [12] in 1995 as a solution method for biobjective combinatorial optimization problems. It uses the fact that the set of efficient solutions can be divided into two disjoint sets, the supported and the nonsupported efficient solutions.

Definition 2. A feasible solution $\sigma' \in \mathcal{P}$ for a biobjective problem $\min\{f(\sigma) \mid \sigma \in \mathcal{P}\}$ is a *supported efficient solution* if there exists a vector $\lambda \in \mathbb{R}_{\geq}^2$ such that σ' is an optimal solution for the weighted sum problem $\min\{\lambda^T f(\sigma) \mid \sigma \in \mathcal{P}\}$. Any efficient solution that is not a supported efficient solution is called *nonsupported efficient solution*. If there exist two linear independent weight vectors such that σ' is optimal for both associated weighted sum problems (equivalently, there exists a weight vector such that σ' is optimal for the associated weighted sum problem and $f(\sigma) = f(\sigma')$ for all optimal solutions to this weighted sum problem), σ' is called an *extremal supported efficient solution*.

In the first phase, the coordinates of the ideal and the nadir point have to be computed, or, equivalently, the nondominated points \hat{z}^r and \hat{z}^s . Then the supported efficient solutions are computed by solving several weighted sum problems with appropriate weights. In particular, the resulting set S of supported efficient solutions contains a solution for each extremal nondominated point.

In the second phase, the nonsupported efficient solutions are determined. By their definition and the properties of the set S , each objective point corresponding to a nonsupported efficient solution must be located in a triangle defined by two consecutive solutions in S (see Figure 1). More precisely: if σ^* is a nonsupported efficient solution

then there exist $\sigma^r, \sigma^s \in S$ and $\lambda \in \mathbb{R}_{\geq}^2$ with

$$\begin{aligned} f_1(\sigma^r) &< f_1(\sigma^*) < f_1(\sigma^s), \\ f_2(\sigma^s) &< f_2(\sigma^*) < f_2(\sigma^r), \\ \lambda^T f(\sigma^*) &> \lambda^T f(\sigma^r) = \lambda^T f(\sigma^s). \end{aligned}$$

The nonsupported efficient solutions are determined in the second phase by exploiting specific properties of the problem at hand. We explain in Section 2.3 how this is done for our problem.

2.2 Biobjective k -Architecture Connected Facility Location

The solution methods mentioned above can, in principle, be applied to any biobjective optimization problem, for which algorithms are available to solve associated weighted sum problems with upper-bound type constraints on the objective functions added to the formulation of the feasible set of solutions. The problem we want to consider for the rest of this paper is taken from telecommunication network optimization and can be formulated as an ILP, which makes it applicable for both ε -constraint Method and Two-Phase Method.

The problem, which we call the *Biobjective k -Architecture Connected Facility Location Problem* (*Bi- k -ArchConFL*), abstracts the planning of access networks with multiple transmission technologies (or “architectures”, such as fiber, copper, and wireless). Given a deployment area for such a network, we can represent the problem by constructing a directed graph $G = (V, A)$, where the node set V contains nodes $C \subset V$ representing customers, nodes $F = F^1 \cup \dots \cup F^k \subset V$ representing potential facility locations for each technology with opening costs $c_i^l \geq 0$ for all $i \in F^l$, $l = 1, \dots, k$, nodes $Q \subset V$ representing potential central office locations with opening costs $c_q \geq 0$ for all $q \in Q$, and further Steiner nodes. The arc set A consists of core arcs $A_c \subset A$ corresponding to forward and backward arcs for each section of the deployment area where a fiber connection can be laid out, and assignment arcs $A^l \subset V$ for each technology $l = 1, \dots, k$ such that each arc $(i, j) \in A^l$ represents a possible connection of customer $j \in C$ to facility $i \in F^l$ using technology l ; each core arc or assignment arc a again comes with a cost value $c_a \geq 0$ or $c_a^l \geq 0$, respectively. Each customer $j \in C$ has a demand value $d_j \geq 0$, which can be interpreted as the number of connections that is required to serve the corresponding customer location.

A feasible solution consists of a set of selected assignment arcs representing served customers, a set of opened facilities such that each facility with outgoing selected assignment arcs is opened, a set of selected core arcs, and a set of opened central offices such that for each opened facility i there is a directed path from an opened central office to i . The selected assignment arcs and opened facilities each represent the use of a technology $l \in \{1, \dots, k\}$; hereby we assume that technology 1 is the “preferred” technology to serve the customers. In the context of fiber-optic access networks, this is the connection using a fiber to the home of the customer: For each customer j a facility $i \in F^1$ is

placed at the same node such that i has a single outgoing assignment arc (i, j) . In order to serve the customer using this “preferred” technology, this facility has to be opened, which implies that core arcs have to be selected that connect the facility, which, in turn, increases the cost considerably. The trade-off between serving many customers with this “costly” technology and reducing costs in general creates the biobjective nature of the optimization problem.

The following ILP formulation for Bi- k -ArchConFL is a direct generalization of the single-objective version of the special case of two technologies (fiber/copper) in [8]:

$$\min \quad \sum_{(i,j) \in A_{rc}} c_{ij} x_{ij} + \sum_{l=1}^k \sum_{(i,j) \in A^l} c_{ij}^l x_{ij}^l + \sum_{l=1}^k \sum_{i \in F^l} c_i^l y_i^l \quad (1)$$

$$\sum_{l=2}^k \sum_{j \in C} d_j z_j^l \quad (2)$$

$$\text{s.t.} \quad \sum_{l=1}^k z_j^l = 1 \quad \forall j \in C \quad (3)$$

$$\sum_{i \in F_j^l} x_{ij}^l = z_j^l \quad \forall j \in C, l = 1, \dots, k \quad (4)$$

$$x_{ij}^l \leq y_i^l \quad \forall j \in C, i \in F_j^l, l = 1, \dots, k \quad (5)$$

$$x(\delta^-(W)) \geq y_i^l \quad \forall W \subseteq V \setminus C, i \in F^l \cap W, l = 1, \dots, k \quad (6)$$

$$x_a, y_i^l, z_j^l \in \{0, 1\} \quad \forall a \in A_c, i \in F^l, j \in C, l = 1, \dots, k \quad (7)$$

$$x_{ij}^l \in \{0, 1\} \quad \forall i \in F_j^l, j \in C, l = 1, \dots, k \quad (8)$$

For a detailed explanation of the model and possible alternatives to the connectivity constraints (6) we refer to [8]. Here we just mention a few specialities occurring in the biobjective setting.

Objective (1) aims to minimize the total cost. Objective (2) minimizes the demand that is covered by technologies other than the preferred technology 1; due to Constraint (3), which states that each customer has to be connected by some technology, this maximizes the demand covered by technology 1. Constraint (3) can be stated as an inequality in case not every customer has to be connected. Obviously, for the biobjective problem, setting all variables to 0 would then be a feasible solution that dominates every other feasible solution, but is surely not desired from the practical point of view. This could be remedied by either introducing coverage requirements for some (or all) technologies (as done in [8] for the single-objective case), or, more generally, by aiming to maximize the total coverage (or the coverage for each technology) and hence transforming the problem into a 3-objective problem (or $(k + 1)$ -objective problem, respectively). For this paper, we stick to the biobjective case. Note, however, that then the problem can equivalently

be modelled by replacing (3) by the corresponding inequality and (2) by

$$\sum_{j \in C} d_j(1 - z_j^1) = D - \sum_{j \in C} d_j z_j^1,$$

where D is the total demand of all customers.

2.3 Application of the two methods to the problem

Application of the ε -constraint Method for the Bi- k -ArchConFL Problem is straightforward. We choose the coverage function (2) as the objective to be bounded by the decreasing ε , since it expectedly takes much smaller values than the cost function (1) and more likely exhibits a coefficient structure that allows a non-trivial greatest common divisor. In each iteration we then solve the ILP

$$\begin{aligned} \min \quad & \sum_{(i,j) \in A_{rc}} c_{ij} x_{ij} + \sum_{l=1}^k \sum_{(i,j) \in A^l} c_{ij}^l x_{ij}^l + \sum_{l=1}^k \sum_{i \in F^l} c_i^l y_i^l \\ \text{s.t.} \quad & \sum_{l=1}^k \sum_{j \in C} d_j z_j^l \leq \varepsilon \\ & (3) - (8) \end{aligned}$$

for the currently valid upper bound ε . To avoid generating weakly dominated points, the obtained optimal solution value f_1^* is then taken as an upper bound on the cost objective and a second ILP

$$\begin{aligned} \min \quad & \sum_{l=1}^k \sum_{j \in C} d_j z_j^l \\ \text{s.t.} \quad & \sum_{(i,j) \in A_{rc}} c_{ij} x_{ij} + \sum_{l=1}^k \sum_{(i,j) \in A^l} c_{ij}^l x_{ij}^l + \sum_{l=1}^k \sum_{i \in F^l} c_i^l y_i^l \leq f_1^* \\ & \sum_{l=1}^k \sum_{j \in C} d_j z_j^l \leq \varepsilon \\ & (3) - (8) \end{aligned}$$

is solved. This yields an efficient solution and a nondominated point (f_1^*, f_2^*) . The same could, in principle, also be achieved by solving just one ILP with a suitable weight vector; this, however, seems to be computationally worse than our two-step process, probably due to the skew weight vector necessary (in a reasonable instance, (1) will take much higher values than (2)) and the involved numerical issues.

Note that the computation of the points \hat{z}^r and \hat{z}^s , and therefore the ideal and nadir point can be done in the same fashion: We just have to solve the above two ILPs without

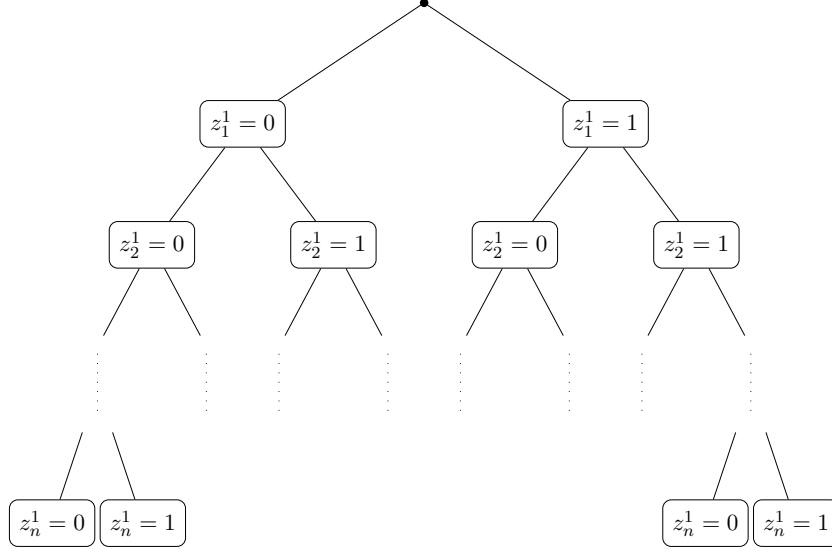


Figure 2: Each branch in the binary search tree represents a possible set of fixations. The leaves represent a complete enumeration of all combinations of fixations.

the ε bound constraints to obtain the point \hat{z}^r . This can be done before the first iteration of the ε -constraint Method to obtain a starting value for ε . The analogous procedure with exchanged objectives yields the point \hat{z}^s .

As for the Two-Phase Method, there is no general algorithm for the Second Phase, but Ulungu and Teghem [12] propose to use the structure of the underlying combinatorial problem to compute the nonsupported efficient solutions. In our case, we perform an enumeration of solutions by fixing binary decision variables to 0 or 1 and combining these fixations. We repeatedly reoptimize weighted sum problems with additional constraints representing the fixations. In our implementation, we fix the z_j^1 variables that indicate if customer j is connected by architecture 1 or not.

Let $n := |C|$ be the number of customers. Assume we are searching for new nondominated solutions in the triangle defined by two consecutive solutions $\sigma^r, \sigma^s \in S$ (cf. Figure 1), which define a fixed weight vector $\lambda = (f_2(\sigma^r) - f_2(\sigma^s), f_1(\sigma^s) - f_1(\sigma^r))^T \in \mathbb{R}_{\geq}^2$. Instead of enumerating all 2^n possible combinations, we fix the n variables iteratively and exclude unnecessary fixations during the search. We can illustrate this fixation procedure with a binary search tree (Figure 2).

Remark 3. Let the variables z_1^1, \dots, z_i^1 be fixed and σ an optimal solution to the corresponding weighted sum ILP (IP1). Let σ' be an optimal solution of a weighted sum ILP (IP2) that obeys the same fixations of z_1^1, \dots, z_i^1 and arbitrary many other fixations of the variables z_{i+1}^1, \dots, z_n^1 . Then we have $\lambda^T f(\sigma) \leq \lambda^T f(\sigma')$. If not, σ would not be an optimal solution of (IP1) since by construction of (IP2), σ' is also feasible to (IP1) and would have a smaller objective value.

We developed several criteria to decide if a combination of fixations can lead to an efficient solution. One of them is the following:

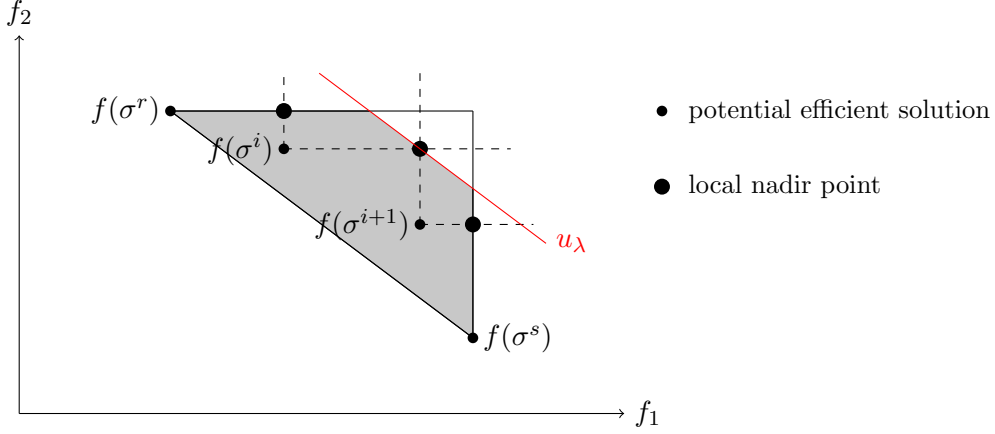


Figure 3: The upper bound u_λ reduces the search space for potential nonsupported nondominated points in the triangle depending on the already found nondominated points.

Observation 4. Let the variables z_1^1, \dots, z_i^1 already be fixed to 0 or 1. Let $D := \sum_{j=1}^n d_j$ be the total demand, $D_{\text{fixed}} := \sum_{j=1}^i d_j z_j^1$ the demand that is already fixed to be served by architecture 1 and $D_{\text{poss}} := \sum_{j=i+1}^n d_j$ the sum of demands of customers that are not yet fixed. If $D - D_{\text{fixed}} \leq f_2(\sigma^s)$ or $D - D_{\text{fixed}} - D_{\text{poss}} \geq f_2(\sigma^r)$, then a reoptimization with any fixation that contains the current fixations will not lead to an efficient solution. We can therefore exclude the current fixation and its lower subtree from the search tree.

If all preliminary criteria of the fixations are met, a reoptimization is done and we proceed depending on the feasibility and the optimal solution of the weighted sum problem. If it is infeasible or the weighted objective value is bigger than some upper bound, the corresponding subtree can be ignored, since adding more fixations can only increase the weighted objective value. If there exists an optimal solution, it has to be verified if this might be an efficient solution or not. There are several lower and upper bounds for this purpose.

We store in the set T the potentially efficient solutions found in the triangle defined by σ^r and σ^s so that $\sigma^r, \sigma^s \in T$ and there exist no $\sigma, \sigma' \in T$ such that σ dominates σ' . We assume that $T = \{\sigma^r = \sigma^1, \dots, \sigma^q = \sigma^s\}$ is ordered by increasing f_1 value.

Definition 5 (Local nadir point). Given T as above, for each $i \in \{1, \dots, q-1\}$ the point $(f_1(\sigma^{i+1}), f_2(\sigma^i))$ is called a *local nadir point*.

As we are looking for efficient solutions only in the current triangle, we can use the following upper bound introduced by Tuytens et al [11]: u_λ represents a line parallel to $\overline{f(\sigma^r)f(\sigma^s)}$, going through the “farthest” local nadir point (see Figure 3) and is defined by

$$u_\lambda := \max_{i=1, \dots, q-1} \{\lambda_1 f_1(\sigma^{i+1}) + \lambda_2 f_2(\sigma^i)\}.$$

Observation 6. Let σ be a feasible nondominated solution with $f(\sigma)$ inside the current triangle. Then $\lambda f(\sigma) < u_\lambda$.

We conclude that if for a feasible solution σ we have $\lambda f(\sigma) \geq u_\lambda$, then σ is dominated by an already found solution in T . Furthermore, suppose that we found the solution σ by solving a weighted sum problem with fixations of the variables z_1^1, \dots, z_i^1 . Since the weighted function value increases by adding fixations (see Remark 3), any additional fixation of one of the variables z_{i+1}^1, \dots, z_n^1 will not lead to an efficient solution and is therefore unnecessary. We ignore the lower subtree of this fixation in the search.

There is another way to avoid solving unnecessary ILPs. Let $\sigma \in T$ be an already found nondominated solution. Then those weighted sum problems with fixations satisfied by σ will only lead to σ as optimal solution again. Instead of solving such a problem, we add a fixation that is not satisfied by σ and reoptimize afterwards. This way, we are certain that the obtained optimal solution is a new one.

These criteria allow us to reduce the number of reoptimizations considerably from the 2^n in the worst case. In practice, there are often a lot more optimizations excluded before computing them. At the end of the second phase, we obtain a minimal complete efficient set, i.e. we find all nondominated points with at least one nondominated solution for each of them.

In general, this version of the Two-Phase Method performs well for problems for which reoptimization (possibly with variable fixations) can be done fast. For our problem, this is not the case, since we have to solve an integer linear program for each reoptimization. Indeed, it is obvious that in total for the Two-Phase Method more ILPs have to be solved than using the ε -constraint Method. However, the division in first and second phase provides another advantage that we examine closer in the next section.

3 Hypervolume and Gaps

The ε -constraint Method computes nondominated points starting in the very upper left corner of the Pareto set and gradually working its way towards the lower right; this way, quite some time is spent in the beginning before an interesting region of solutions is entered. The Two-Phase Method, on the other hand, initially computes the extremal nondominated points, which are usually evenly distributed between \hat{z}_r and \hat{z}_s . This has the advantage that already early in the process a quite close approximation of the complete Pareto set is available, making it possible to interrupt the computation and thereby making a well predictable error. In this section we want to make this precise.

The *hypervolume indicator* was introduced by Zitzler and Thiele, see [13], for instance. We present this concept in a slightly different way than in the literature (cf. [2]) extending the notion a bit. The basic definitions apply to a general multiobjective optimization problem. We denote the *ideal point* by \hat{z}^I and the *nadir point* by \hat{z}^N ; B is the box $B := [\hat{z}^I, \hat{z}^N] = [\hat{z}_1^I, \hat{z}_1^N] \times \dots \times [\hat{z}_d^I, \hat{z}_d^N]$. For points in the objective space, we occasionally use the notation $z = f(\sigma)$ for a feasible solution σ (where the z is not to be confused with the variables z_j^l of the ILP model).

Definition 7. Given a set Z of points in objective space \mathbb{R}^d , $d \geq 2$, the *hypervolume* $I_H(Z)$ is the (d -dimensional) volume of B dominated by points in Z ,

$$I_H(Z) := \text{vol}_d \left(\bigcup_{z \in Z} (z + \mathbb{R}_{\geq}^d) \cap B \right) = \text{vol}_d \left(\bigcup_{z \in Z} [z, \hat{z}^N] \right)$$

Given a set $\hat{P} = \{\sigma^1, \dots, \sigma^n\}$ of feasible solutions, we denote by $I_H(\hat{P})$ the hypervolume of the corresponding objective points, $I_H(\hat{P}) := I_H(\{f(\sigma^1), \dots, f(\sigma^n)\})$.

Remark 8. If P is a complete set of efficient solutions (i. e., $f(P)$ is the Pareto set), then $I_H(P) \geq I_H(\hat{P})$ for all sets of nondominated points \hat{P} . In other words: the hypervolume of each set \hat{P} of efficient solutions is a lower bound on the hypervolume of P .

Definition 9. Let \hat{P} be a set of feasible solutions and P a complete set of efficient solutions. The *absolute hypervolume gap* (of \hat{P}) is defined as

$$\delta_H(\hat{P}) := I_H(P) - I_H(\hat{P}) \geq 0.$$

The (*relative*) *hypervolume gap* (of \hat{P}) is

$$\gamma_H(\hat{P}) := \frac{\delta_H(\hat{P})}{I_H(P)}.$$

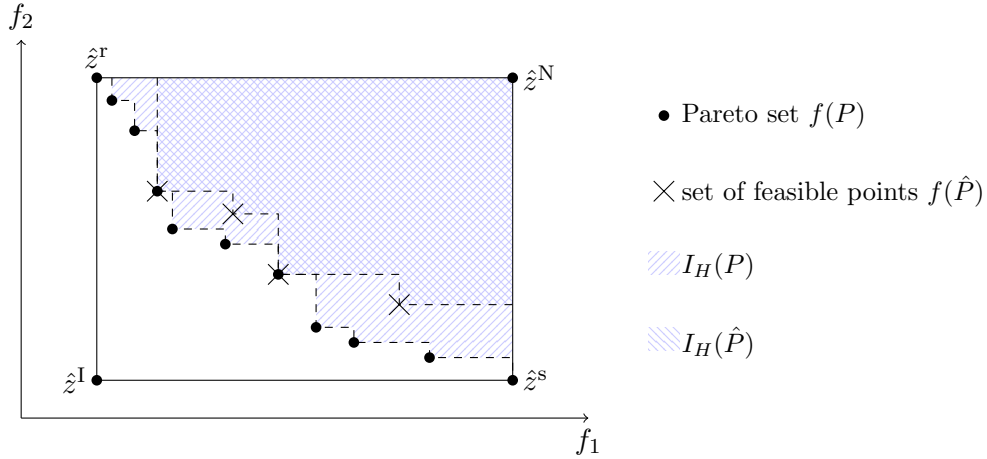


Figure 4: The hypervolume of the Pareto set $f(P)$ and of a set of feasible points $f(\hat{P})$. The difference between $I_H(P)$ and $I_H(\hat{P})$ is the absolute hypervolume gap $\delta_H(\hat{P})$

Remark 10. Let $A \subseteq B$ such that the interior of A contains no feasible solutions and $[\hat{z}^I, z] \subseteq A$ for all $z \in A$. Then

$$I_H^-(A) := \text{vol}_d(B \setminus A) = \left(\prod_{i=1}^d (\hat{z}_i^N - \hat{z}_i^I) \right) - \text{vol}_d(A)$$

is an upper bound on the hypervolume of the Pareto set P : $I_H(P) \leq I_H^-(A)$. Hence (by Remark 8) $I_H(\hat{P}) \leq I_H^-(A)$ for each set \hat{P} of feasible solutions.

Definition 11. Let \hat{P} be a set of feasible solutions and $A \subseteq B$ as above. The *absolute hypervolume bound gap* is defined as

$$\delta_H^-(\hat{P}, A) := I_H^-(A) - I_H(\hat{P}) \geq 0$$

and the *(relative) hypervolume bound gap* is

$$\gamma_H^-(\hat{P}, A) := \frac{\delta_H^-(\hat{P}, A)}{I_H(\hat{P})}.$$

It follows directly from the definitions that

$$\delta_H^-(\hat{P}, A) \geq \delta_H(\hat{P})$$

and

$$\gamma_H^-(\hat{P}, A) \geq \gamma_H(\hat{P})$$

for \hat{P} and A as in Definition 11.

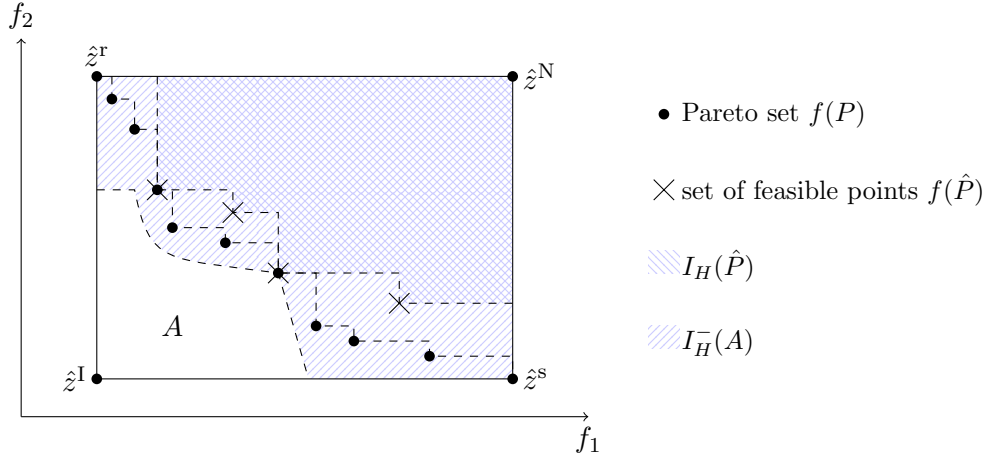


Figure 5: The hypervolume of a set of feasible points $f(\hat{P})$ and a set A that contains no feasible points in its interior and $[\hat{z}^I, z] \subseteq A$ for all $z \in A$. The difference between $I_H^-(A)$ and $I_H(\hat{P})$ is the absolute hypervolume bound gap $\delta_H^-(\hat{P}, A)$.

During the execution of a specific algorithm there are various possibilities to derive sets A that yield bound gaps $\gamma_H^-(\hat{P}, A)$ for a current set (of possibly efficient solutions) \hat{P} .

Example 12 (Adjusted hypervolume, cf. [2]). Let $\hat{P} = \{\sigma^1, \dots, \sigma^n\}$ be a set of efficient solutions. Then

$$A_a := B \cap \bigcup_{i=1}^n (f(\sigma^i) - \mathbb{R}_{\geq}) = \bigcup_{i=1}^n [\hat{z}^I, f(\sigma^i)]$$

is a set containing no feasible solutions in its interior with $[\hat{z}^I, z] \subseteq A_a$ for all $z \in A_a$ and

$$I_H^-(A_a) = \text{vol}_d(B \setminus A_a) = \text{vol}_d(B) - \text{vol}_d\left(\bigcup_{i=1}^n [\hat{z}^I, f(\sigma^i)]\right)$$

represents an upper bound on $I_H(P)$ (see Figure 6).

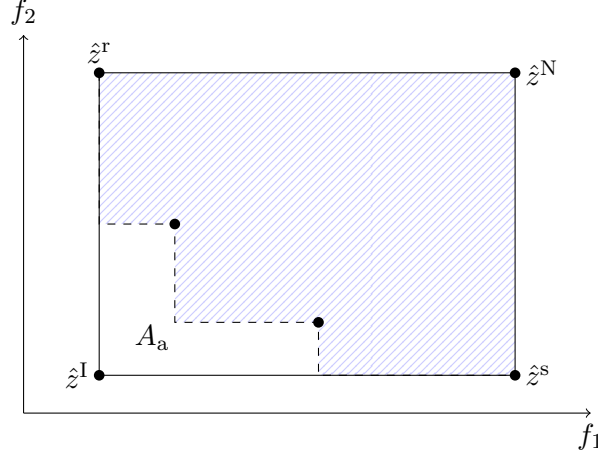


Figure 6: The adjusted hypervolume and the upper bound $I_H^-(A_a)$.

Example 13 (ε -constraint Method upper bound). The ε -constraint Method for biobjective problems determines the efficient solutions with their increasing f_1 -value. The upper bound of the adjusted hypervolume can thus be improved:

Let $\hat{P} = \{\sigma^1, \dots, \sigma^n\}$ be a set of efficient solutions, $z^i = f(\sigma^i)$, $i = 1, \dots, n$ and $Z := \{z^1, \dots, z^n\}$ such that there is no nondominated point z' with $z'_1 \leq z_1^n$ and $z' \notin Z$. Then

$$A_\varepsilon := B \setminus \left(\bigcup_{z \in Z} [z, \hat{z}^N] \cup [(z_1^n + \delta, \hat{z}_2^I), \hat{z}^N] \right) = \bigcup_{k=1}^{n-1} [\hat{z}^I, (z_1^{k+1}, z_2^k)] \cup [\hat{z}^I, (z_1^n + \delta, z_2^n)]$$

is a set containing no feasible solutions in its interior with $[\hat{z}^I, z] \subseteq A_\varepsilon$ for all $z \in A_\varepsilon$ (see Figure 7), where δ is a suitable small constant. If all coefficients of the z_1 objective are integral, δ can be chosen to be the greatest common divisor of the coefficients, analogous to the choice of ε . Note that for this the ε -constraint Method has to be performed in such a way that the solution computed in each iteration is not a weakly efficient solution, cf. Section 2.3. Hence

$$I_H^-(A_\varepsilon) = \text{vol}_2(B \setminus A_\varepsilon) = \text{vol}_2 \left(\bigcup_{z \in Z} [z, \hat{z}^N] \cup [(z_1^n + \delta, \hat{z}_2^I), \hat{z}^N] \right)$$

represents an upper bound on $I_H(P)$.

Example 14 (End-of-first-phase upper bound). After termination of the first phase of the Two-Phase Method, all extremal supported nondominated points are known. By using this property, we find an upper bound:

Given a biobjective problem, let \hat{P} be a set of efficient solutions so that $Z = f(\hat{P}) = Z_{\text{extsN}}$ is the set of all extremal supported nondominated points. Then

$$A_{P1} := B \setminus \text{conv}(Z \cup \{\hat{z}^N\})$$

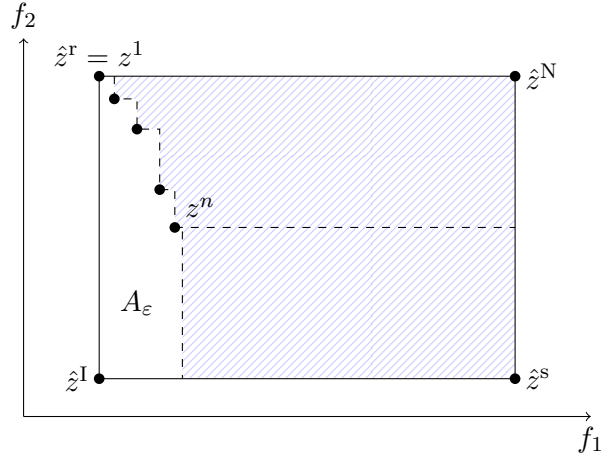


Figure 7: The ε -constraint Method upper bound $I_H^-(A_\varepsilon)$.

is a set that contains no feasible solutions in its interior with $[\hat{z}^I, z] \subseteq A_{P1}$ for all $z \in A_{P1}$ and

$$I_H^-(A_{P1}) = \text{vol}_2(B \setminus A_{P1}) = \text{vol}_2(\text{conv}(Z \cup \{\hat{z}^N\}))$$

is an upper bound on $I_H(P)$; see Figure 8. Note that this bound is in general not valid

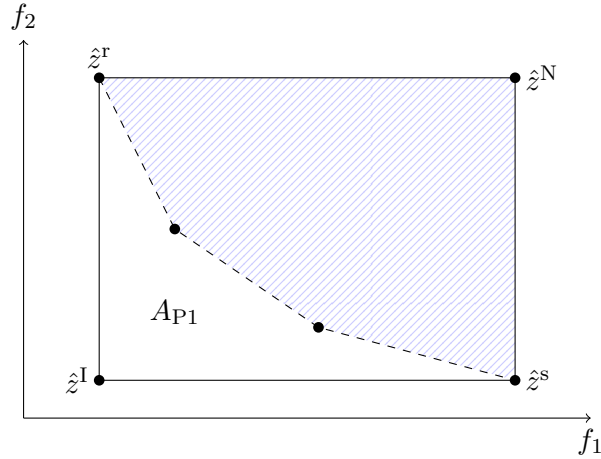


Figure 8: End-of-first-phase upper bound $I_H^-(A_{P1})$

for multiobjective problems with 3 or more objectives, since, in general, for a set Z of nondominated points in \mathbb{R}^d , $d \geq 3$, the set $A = B \setminus \text{conv } Z$ can intersect the region dominated by Z and $[\hat{z}^I, z]$ for $z \in A$ is not necessarily fully contained in A .

Example 15 (Second-phase upper bound). During the second phase, the triangles in the objective space defined by two consecutive extremal supported nondominated points are considered iteratively to find the missing efficient solutions. The upper bound $I_H^-(A_{P1})$ can therefore be improved:

Let \hat{P} be a set of efficient solutions so that $Z := f(\hat{P}) = Z_{\text{extsN}} \cup Z_t$ where $Z_{\text{extsN}} = \{z^1, \dots, z^n\}$ is the set of all extremal supported nondominated points, $1 \leq t \leq n-1$, and $Z_t = \{z^t = z^{t,1}, z^{t,2}, \dots, z^{t,l} = z^{t+1}\}$ with $z_1^{t,1} < \dots < z_1^{t,l}$ and $z^{t,i}$ is situated in the triangle defined by consecutive solutions z^t and z^{t+1} for $1 \leq i \leq l$. Let $A \subseteq B$ with $A \supseteq A_{P1}$ be a set that contains no feasible solution and $[\hat{z}^I, z] \subseteq A$ for all $z \in A$. Then

$$A_{P2} := A \cup \bigcup_{k=1}^{l-1} [\hat{z}^I, (z_1^{t,k+1}, z_2^{t,k})]$$

contains no feasible solution in its interior with $[\hat{z}^I, z] \subseteq A_{P2}$ for all $z \in A_{P2}$ and

$$I_H^-(A_{P2}) = \text{vol}_2(B \setminus A_{P2})$$

is an upper bound on $I_H(P)$; see Figure 9. Obviously, $A_{P2} \supseteq A \supseteq A_{P1}$ and hence

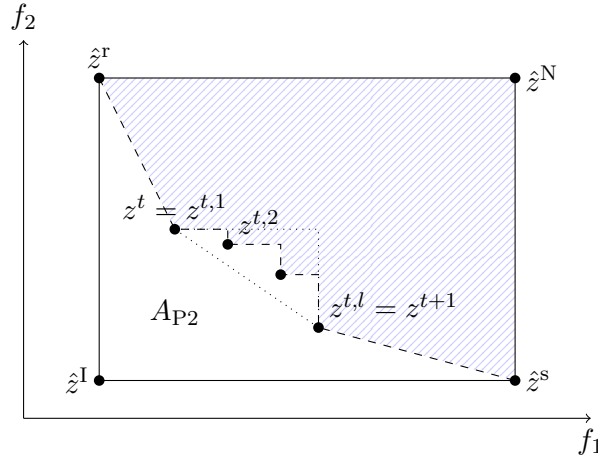


Figure 9: Second-phase upper bound $I_H^-(A_{P2})$

$I_H^-(A_{P2}) \leq I_H^-(A) \leq I_H^-(A_{P1})$, that is, the second-phase upper bound improves upon the first-phase upper bound and can iteratively be strengthened when a new triangle t is considered.

4 Computations

In the following, we report on some results obtained from computations using our implementations of the algorithms described in Section 2. In all computations, the number of technologies is $k = 2$. The instances used are derived from four different networks that represent typical (rather small) deployment areas for fiber-optic access networks. For each of these four networks, a number of different instances for the Bi- k -ArchConFL Problem were constructed by varying certain parameters, such as the density of the facilities and the number of assignment arcs for architecture 2. Table 1 gives an impression of the sizes of the instances. Note that the number of nodes and arcs contained in

Network	# instances	# nodes	# arcs	# customers	# facilities
rotdorn	18	254	512–907	91	106–157
tuberlin	21	384	644–829	39	54–109
vehlefan	52	895	1640–4847	238	266–407
atlantis	18	1001	1880–3981	345	361–447

Table 1: Overview of the test instances and their properties

the network is not the only decisive factor concerning the solvability of the instances: the **tuberlin** instances tend to be easier solvable than the **rotdorn** instances and similarly for the two bigger networks (cf. Figure 11); the first observation can possibly be explained by the much smaller number of customers and facilities in **tuberlin**, but there is no straightforward explanation (at least not in terms of numbers) for the bigger networks.

The test computations were performed on an Ubuntu 14.04 system running on an Intel Xeon X5672 quad core 3.2 GHz CPU with 48 GB memory. In our implementations we used CPLEX 12.6 as solver for the single-objective ILPs, with parallelization limited to one thread. We set a total time limit of 21600 s (6 h) for all computations.

The total runtimes in general (see Figure 10) show that for all instances the time limit is hit for the Two-Phase Method. The ε -constraint Method finishes within 6 h for nearly all of the smaller instances (37 out of 39, see Figure 11), but for none of the larger ones. The first phase of the Two-Phase Method, however, is completed much faster on average than both the second phase and the ε -constraint Method, as Figures 10 and 11 show (although it is outperformed by the ε -constraint Method on one instance of each of the two sets).

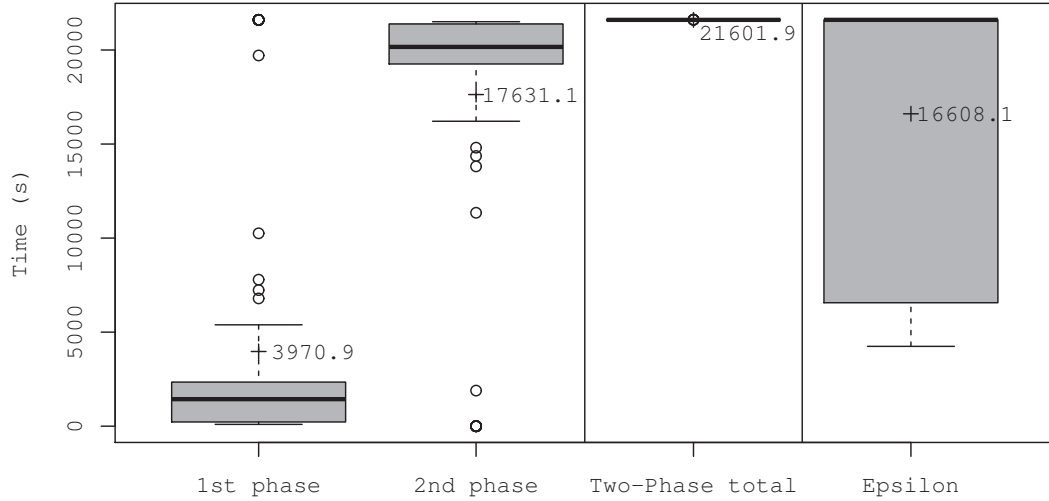


Figure 10: Runtimes for all 109 instances; a + mark indicates the (arithmetic) mean value

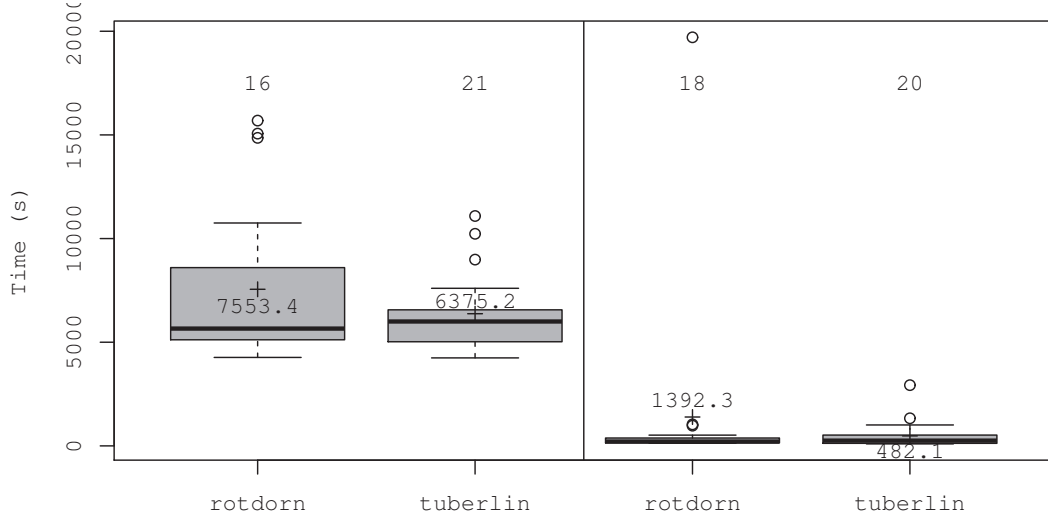
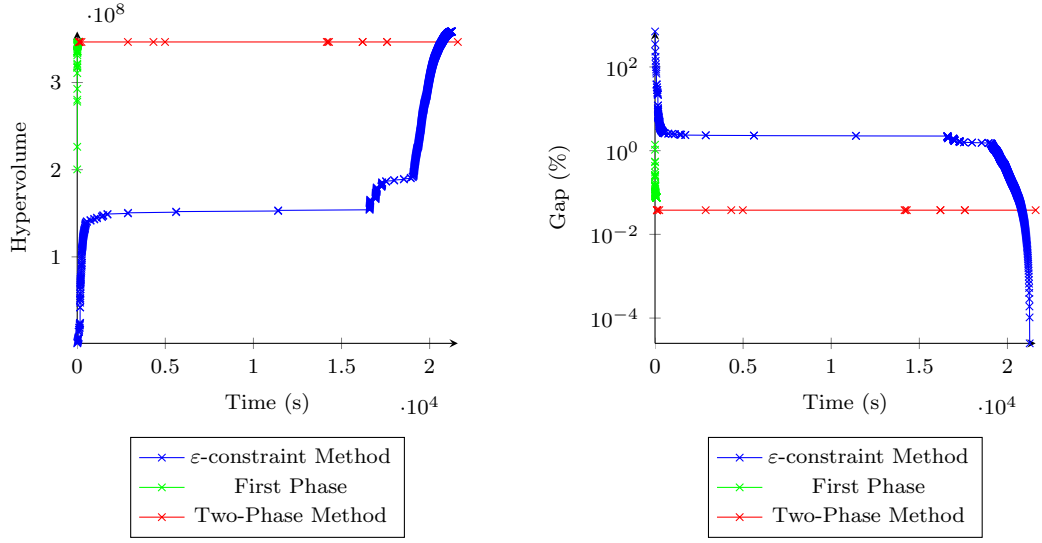


Figure 11: Runtimes of the ε -constraint Method for instances that have been solved within the time limit of 6 h (21600 s) on the left and runtimes of the First Phase for instances that have completed the First Phase within the time limit on the right

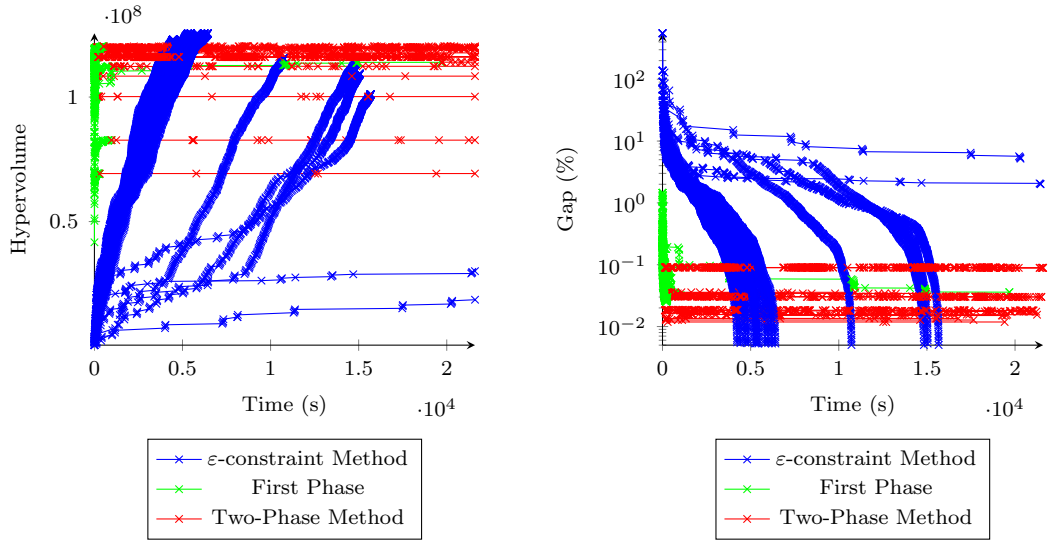
Hence one would expect that the hypervolume (and the hypervolume bound gap with suitable upper bound set A) increases (decreases, respectively) faster over the runtime for the Two-Phase Method than it does for the ε -constraint Method. A typical scenario for the development of the hypervolume (bound gap) during the first and second phase (using the sets A_a and A_{P2} , respectively), compared to the ε -constraint Method (using A_ε) is shown in Figure 12(a).

The precise behaviour obviously very much depends on the instance at hand; Figure 12(b) shows the trend for several comparable instances. The ε -constraint Method overtakes the Two-Phase Method eventually, both with respect to the hypervolume and the gap; at relatively early stages of the computations, however, i.e., after runtimes of below 1 to 4 hours for most of the shown instances, both the hypervolume and the gap are much better for the Two-Phase Method, due to the fast execution of the first phase and the more precise upper bound.

Accordingly, for those instances that have not been completely solved, the hypervolume bound gap at the end of the computation time (see Figure 13) is much lower for the Two-Phase Method than for the ε -constraint Method. Figure 14 shows that, somewhat surprisingly, the remaining hypervolume bound gap for the Two-Phase Method does not seem to vary greatly with the size of the instance. This cannot be said for the size of the solutions, as Figure 15 shows. The number of nondominated points seems to depend on the instance size and also – maybe even more significantly – on the type of the core network and the number of customers and assignment arcs in relation to the core network size. For the larger instances, no complete nondominated set could be obtained by the ε -constraint Method; however, the number of extremal solutions computed during the first phase (see Figure 16), seems to be a good indicator of the total solution size, as it



(a) ...for a typical run of an **atlantis** instance (N.B. The computation was performed on a faster computer, hence the ε -constraint Method managed to compute the full nondominated set within the time limit, in contrast to the other results described in this section.)



(b) ...for all instances of the **rotdorn** network

Figure 12: Hypervolume over the runtime...

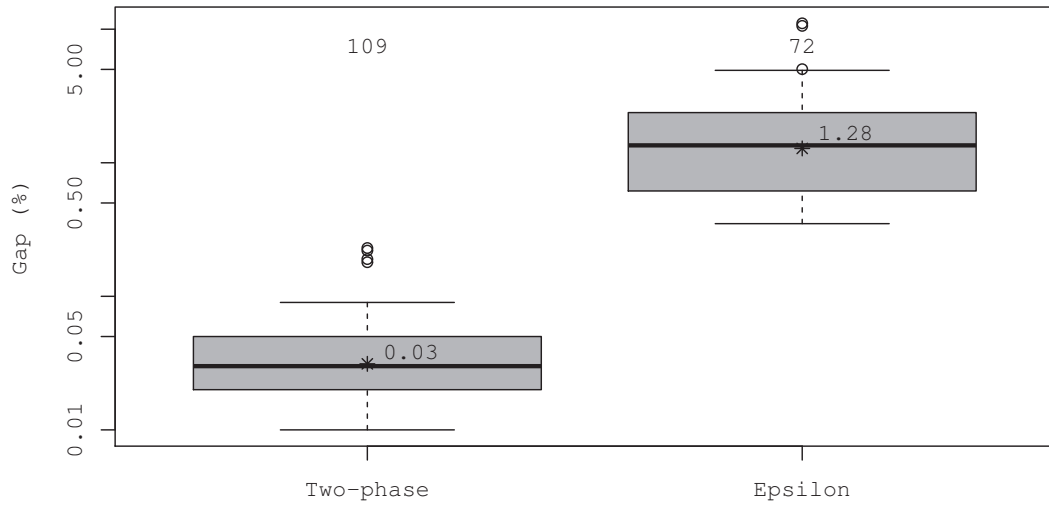


Figure 13: Relative hypervolume bound gap after 6 h for all instances that hit the time limit; a * mark indicates the geometric mean value

shows a comparable trend for the smaller networks.

To compute a *complete* nondominated set of points, the ε -constraint Method is much better suited than the Two-Phase Method. The main reason is that in every iteration of the ε -constraint Method, in which two weighted sum problems are solved, a new nondominated point is obtained; that is, the number of ILPs solved during the execution is linear in the size of the Pareto set. For the Two-Phase Method, on the other hand, many ILPs have to be solved for the variable fixings in the second phase, of which only

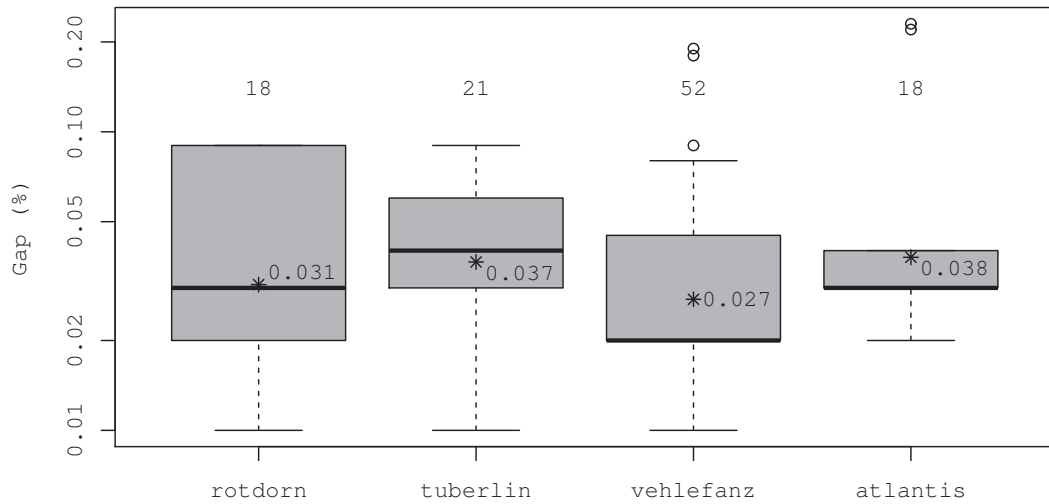


Figure 14: Relative hypervolume bound gap of the Two-Phase Method after 6 h for all instances that hit the time limit

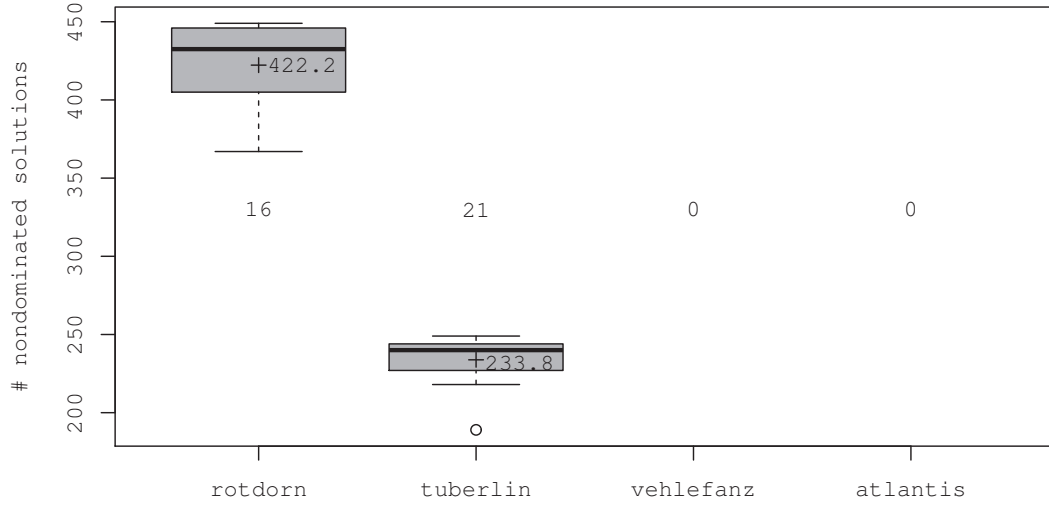


Figure 15: Size of the solutions (complete set of nondominated points) for all instances solved by the ε -constraint Method within the time limit

a small fraction yield nondominated points in the end. Accordingly, the number of calls to CPLEX's `solve` routine (see Figure 17) is much higher for the Two-Phase Method than for the ε -constraint Method.

To get an impression of how far the second phase progressed at the end of the time limit, we can have a look at the number of triangles that have been processed in the second phase. Figure 18 shows, for each of the four networks, the average over the numbers of processed triangles as well as the total numbers of triangles. The processed triangles are

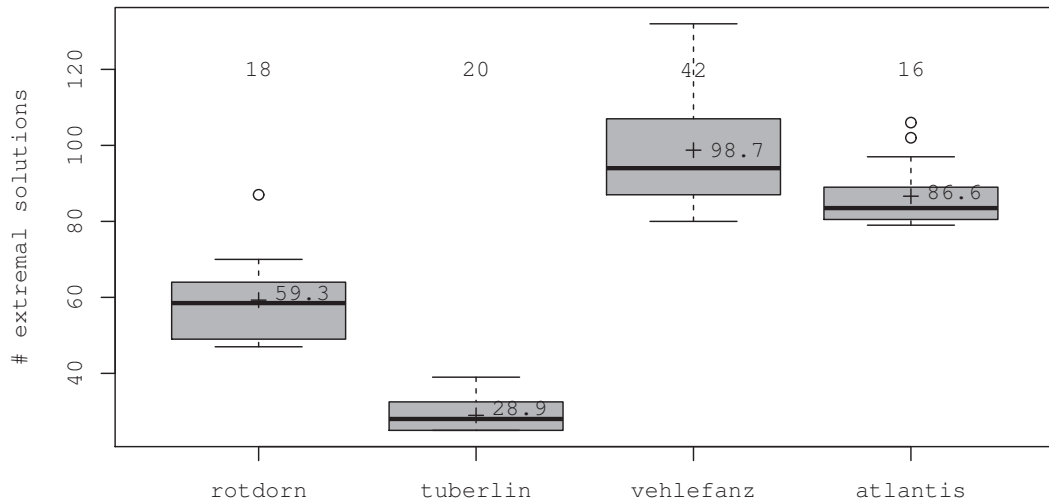


Figure 16: Number of extremal solutions for all instances for which the First Phase terminated within the time limit

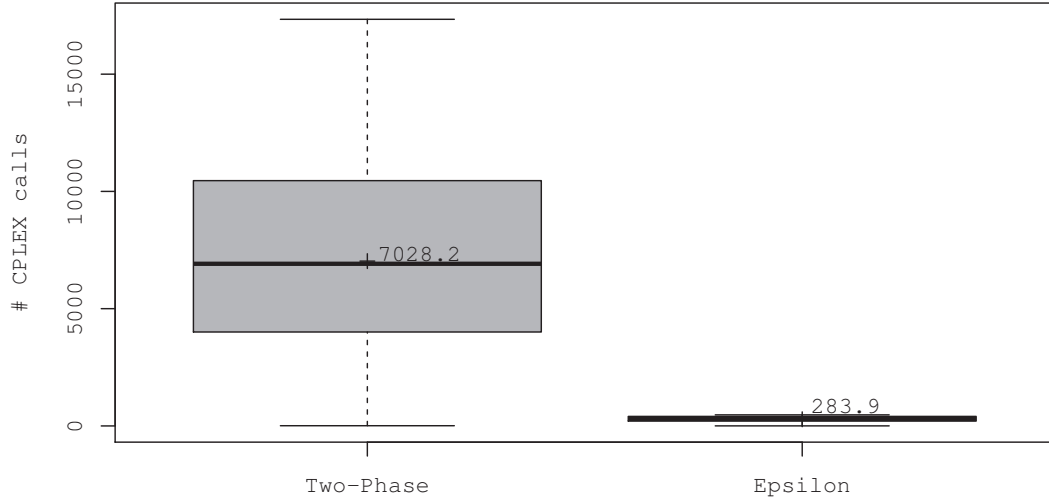


Figure 17: Number of CPLEX calls for all instances

those that are completely scanned, plus, possibly, the last triangle, whose examination might have been interrupted by the time limit. Figure 19 shows the relative number of triangles that were processed during the second phase. While for the bigger networks the second phase rarely managed to process more than the first triangle, for most of the smaller instances at least a few triangles were completed; in our implementation the second phase starts processing triangles located in the middle of the box B , i.e. those that are defined by supported efficient solutions with f_2 value around $(\hat{z}_2^N - \hat{z}_2^I)/2$, and works its way from there to the next triangle on the left and right alternatingly.

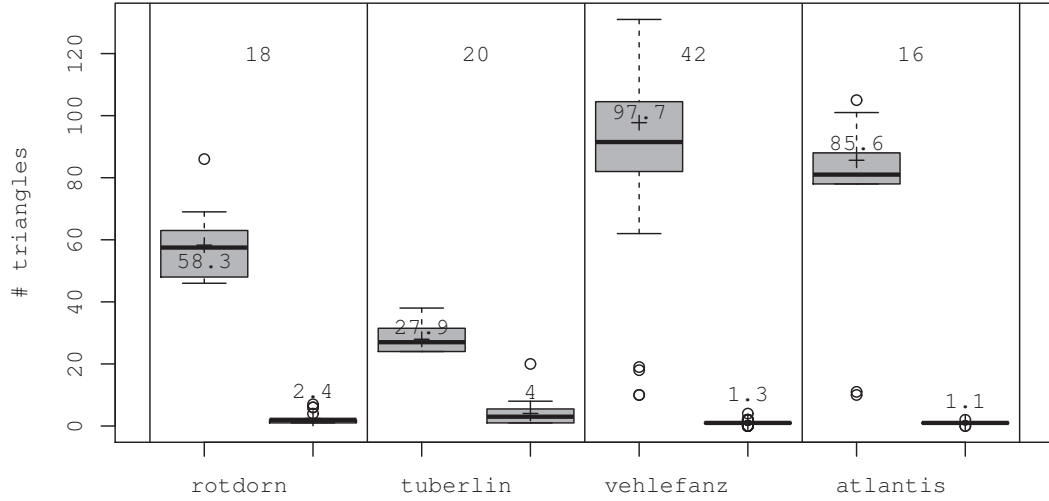


Figure 18: Total number of triangles and number of processed triangles in the second phase for the different networks for all instances for which the second phase was started

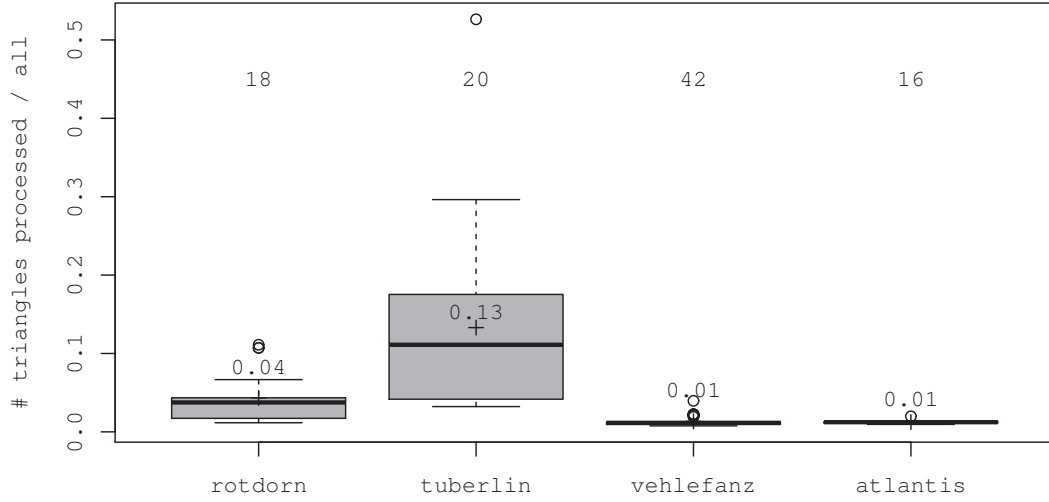


Figure 19: Fraction of the number of processed triangles for the same instances as in Figure 18

Obviously, various other strategies for selecting triangles in interesting regions to be processed first can be devised.

Note that Figures 18 and 19 also suggest that the **tuberlin** instances tend to be easier solvable, also for the Two-Phase Method, than the **rotdorn** instances, despite the slightly larger core network, cf. Figure 11.

5 Conclusions and Outlook

We have described an ε -constraint Method and a Two-Phase Method specifically tailored for solving the Bi- k -ArchConFL Problem. While the Two-Phase Method cannot compete with the ε -constraint Method in terms of efficiency, when it comes to computing a complete set of nondominated points, it can quite well be used to obtain a good approximation – in terms of the hypervolume – in a comparably short time. This is mainly due to the fact that the first phase runs quite fast and already provides a close upper bound for the hypervolume of the nondominated set; from there, selected parts of the approximated nondominated set can be explored in the second phase, such that a more precise approximation in the most interesting regions can be obtained.

The performance of other solution methods have also been assessed in terms of hypervolume measures, see [2, 3]. Obviously, the concept of hypervolume is also applicable for problems with 3 and more objectives, cf. [4].

Finally, for another strategy to obtain approximated nondominated sets with gap information, one could interrupt the solution process of the weighted sum models after a certain time instead of solving them to optimality in each iteration and use the obtained primal solutions with their respective gaps to deduce upper bounds for the hypervolume.

References

- [1] Natasha Boland, Hadi Charkhgard, and Martin Savelsbergh. The triangle splitting method for biobjective mixed integer programming. *Lecture Notes in Computer Science*, 8494:162–173, 2014.
- [2] Natasha Boland, Hadi Charkhgard, and Martin Savelsbergh. A criterion space search algorithm for biobjective mixed integer programming: The rectangle splitting method. Technical report, 2015. http://www.optimization-online.org/DB_HTML/2013/08/3987.html.
- [3] Natasha Boland, Hadi Charkhgard, and Martin Savelsbergh. A criterion space search algorithm for biobjective mixed integer programming: The triangle splitting method. Technical report, 2015. http://www.optimization-online.org/DB_HTML/2013/11/4107.html.
- [4] Natasha Boland, Hadi Charkhgard, and Martin Savelsbergh. The L -shape search method for triobjective integer programming. Technical report, 2015. http://www.optimization-online.org/DB_HTML/2014/05/4337.html.
- [5] Matthias Ehrgott. *Multicriteria Optimization*. Springer, 2nd edition, 2005.
- [6] Y. Haimes, L. Lasdon, and D. Wismer. On a bicriterion formulation of the problems of integrated system identification and system optimization. *IEEE Transactions on Systems, Man, and Cybernetics*, 1:296–297, 1971.
- [7] Markus Leitner, Ivana Ljubić, and Markus Sinnl. A computational study of exact approaches for the bi-objective prize-collecting steiner tree problem. *INFORMS Journal on Computing*, 27(1):118–134, 2015.
- [8] Markus Leitner, Ivana Ljubić, Markus Sinnl, and Axel Werner. On the two-architecture connected facility location problem. *Electronic Notes in Discrete Mathematics*, 41:359–366, 2013.
- [9] Markus Leitner, Ivana Ljubić, Markus Sinnl, and Axel Werner. New exact and heuristic MIP-approaches for bi-objective integer programs: Application to bi-objective FTTx-network design problems. Technical report, 2015. in preparation.
- [10] Antony Przybylski, Xavier Gandibleux, and Matthias Ehrgott. Two phase algorithms for the bi-objective assignment problem. *European Journal Of Operational Research*, 185(2):509–533, 2008.
- [11] D. Tuytens, J. Teghem, Ph. Fortemps, and K. Van Nieuwenhuyse. Performance of the MOSA Method for the Bicriteria Assignment Problem. *Journal of Heuristics*, 6(3):29–310, 2000.
- [12] E.L. Ulungu and J. Teghem. The two phases method: An efficient procedure to solve bi-objective combinatorial optimization problems. *Foundations of Computing and Decision Sciences*, 20(2):149–165, 1995.
- [13] Eckart Zitzler, Lothar Thiele, Marco Laumanns, Carlos M. Fonseca, and Viviane Grunert da Fonseca. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132, 2003.