ON REFINEMENT STRATEGIES FOR SOLVING MINLPs BY PIECEWISE LINEAR RELAXATIONS: A GENERALIZED RED REFINEMENT

ROBERT BURLACU[1,2]

ABSTRACT. We investigate the generalized red refinement for n-dimensional simplices that dates back to Freudenthal [5] in a mixed-integer nonlinear program (MINLP) context. We show that the red refinement meets sufficient convergence conditions for a known MINLP solution framework that is essentially based on solving piecewise linear relaxations. In addition, we prove that applying this refinement procedure results in piecewise linear relaxations that can be modeled by the well-known incremental method established by Markowitz and Manne [13]. Finally, numerical results from the field of alternating current optimal power flow demonstrate the applicability of the red refinement in such MIP-based MINLP solution frameworks.

1. INTRODUCTION

Solving general MINLPs is to this day a very challenging task. The backbone of most approaches in this context is still branch-and-bound. In the last two decades, however, various methods have been proposed that tackle non-convex MINLPs by piecewise convex relaxations without direct branching of the continuous variables, see for example [7, 8, 12, 14, 16]. Although these approaches are sometimes rather different, they all need to address the following two problems: the construction of reasonable relaxations of the nonlinear functions and the incorporation of these relaxations into a mixed-integer linear program (MIP) or convex nonlinear program (NLP).

One way to obtain such relaxations is to compute an optimal linearization of a nonlinear function with respect to the number of breakpoints and an a priori given accuracy as in [17, 18]. Complementary, optimal polynomial relaxations of one-dimensional functions are constructed in [16]. For up to three-dimensional functions, explicit approximation techniques for general nonlinear functions are proposed in [15]. The main drawback of all these methods, however, is that the number of simplices in the approximation grows exponentially with the dimension of the function. We refer to the approach from [19] that avoids this problem in that the piecewise linear approximation is not required to interpolate the original function at the vertices of the triangulation.

There are many different ways to model piecewise linear functions as an MIP. A detailed overview of the various models is presented by [20]. Among the most important ones is the incremental method of Markowitz and Manne, see [13], which is originally developed for one-dimensional functions. A generalization to higher dimensions is described in [11, 21] and [6]. Supplementary to this, an extension to relaxations is given in [7].

In this paper, we consider the MINLP solution method proposed in [3, 6] and further developed in [2] that tackles an MINLP by solving a series of MIP relaxations that are based on piecewise linear functions and completely contain the graph of the function. The functions are defined on simplices, which in turn are defined by several vertices. The authors develop an iterative algorithm to find a global optimal solution of the MINLP by solving these MIP relaxations, which are adaptively refined. They present rather general convergence conditions for MINLP solution algorithms that rely on the adaptive refinement of piecewise linear relaxations. They show that the classical longest-edge bisection fulfills these conditions and therefore is suitable for such a solution framework. In addition, they prove that triangulations that are constructed by successively applying the longest-edge bisection lead to piecewise linear relaxations that can always be modeled by the already mentioned generalization of the incremental method.

We extend this result by another refinement strategy for n-dimensional simplices: the generalized red refinement introduced by Freudenthal in [5]. This procedure is to some extent an n-dimensional generalization of the well-known red-green refinement, which is used for two-dimensional simplices. We show that the red refinement meets the convergence conditions from [3]. Moreover, we prove that the generalized incremental method is suitable to model piecewise linear relaxations that are obtained by iteratively applying the red refinement. Finally, numerical results from the field of alternating current optimal power flow demonstrate the applicability of the red refinement in an MINLP solution framework as presented in [3].

This article is structured as follows. We introduce all necessary definitions and theorems of previous works in Section 2. Section 3 shows that the red refinement fulfills the convergence conditions from [3]. In Section 4, we prove that adaptively red refined piecewise linear relaxations can be modeled by the generalized incremental method. Section 5 presents some numerical results that illustrate the practicability of the red refinement procedure. We conclude this work in Section 6.

2. Preliminaries

The aim of this article is to prove that the generalized red refinement procedure is suitable for an MINLP solution framework such as in [3].

We consider an MINLP problem as an optimization problem of the following type:

$$\begin{aligned}\min_{x}\quad & c^\top x\\ \text{s.t.}\quad & Ax \le b,\\ & f_i(x) \le 0 \qquad \text{for all } i \in \{1,\dots,k\},\\ & l \le x \le u,\\ & x \in \mathbb{R}^q \times \mathbb{Z}^p,\end{aligned} \tag{P}$$

where k, q, $p \in \mathbb{N}$. First, $Ax \le b$ represents the linear constraints, while the nonlinear constraints are described by continuous nonlinear real-valued functions $f_i\colon \mathbb{R}^{q+p} \to \mathbb{R}$ for $i = 1,\dots,k$. The variables x are bounded from below and above by $l, u \in \mathbb{R}^{q+p}$. Moreover, we denote by $\mathcal{F}$ the set of all nonlinear functions $f_i(x)$. Let $D_f \subset \mathbb{R}^{q+p}$ be the domain of a nonlinear function $f \in \mathcal{F}$. Since each variable in (P) has lower and upper bounds, the domain D_f is a compact set. We consider it to be a d-dimensional box with its edges parallel to the coordinate axes, while $d \leq q+p$. Equality constraints, i.e., constraints of type $f_i(x) = 0$, are implicitly contained in (P) by simply adding the constraints $f_i(x) \leq 0$ and $-f_i(x) \leq 0$. Please note that we are not restricted to a linear objective function $c^\top x$, because we can include any nonlinear objective function $f\colon \mathbb{R}^{q+p} \to \mathbb{R}$ by substituting $f(x)$ with a variable $y \in \mathbb{R}$ and adding $f(x) \leq y$ as a constraint to the MINLP problem. Due to $\max c^\top x = -\min -c^\top x$, any maximization problem can be transformed to a minimization problem. Thus, (P) represents a general formal description of MINLP problems.

The main idea of the MINLP solution approach from [3] is to use piecewise linear functions to construct MIP relaxations of the underlying MINLP. An iterative algorithm is developed to find a global optimal solution by solving these relaxations, which are adaptively refined. With the domain D_f of $f \in \mathcal{F}$, the refinement is performed on a simplicial triangulation $\mathcal{T}$ of D_f. A more formal description of this method is given in Algorithm 2.1.

In [3], the classical longest-edge bisection is used for the refinement procedure in Line 19 of Algorithm 2.1. In this paper, we consider the generalized red refinement instead. Please note that the theoretical results are part of the author's PhD thesis [2].

In order to be equipped with all the necessary ingredients for the following proofs, we first introduce the most relevant definitions and theorems from [3]. First, with a *δ-precise* refinement procedure, Algorithm 2.1 is both correct and convergent:

Definition 2.1. The refinement procedure in Algorithm 2.1 is called *δ-precise*, if for an arbitrary sequence $(S^i) \in \mathcal{T}_i$ of simplices S^i that are refined by the refinement procedure with initial triangulation $\mathcal{T}_0$ of D_f and given $\delta > 0$, there exists an index $N \in \mathbb{N}$, such that

$$\operatorname{diam}(S^N) < \delta \tag{1}$$

holds, where $\operatorname{diam}(S^N) \coloneqq \sup_{x',x'' \in S^N}\{\|x' - x''\|\}$.

Proposition 2.2 (Theorem 3.6, [3])**.** *If the refinement procedure in Algorithm 2.1 is δ-precise for every $\delta > 0$, then Algorithm 2.1 is correct and terminates after a finite number of steps.*

It is therefore sufficient to prove that the generalized red refinement is δ-precise in order to show that its combination with Algorithm 2.1 yields a correct and convergent algorithm.

Moreover, we must prove that piecewise linear relaxations that are obtained by iteratively applying the red refinement can be described by an MIP model. In Algorithm 2.1, the piecewise linear relaxations are modeled by the generalized incremental method, see [6]. There are two main ideas of the generalized incremental model. At first, any point x^S inside a simplex S with its vertex set $\mathcal{V}(S) = \{\bar{x}_0^S, \ldots, \bar{x}_d^S\}$ can be expressed either as a convex combination of its vertices or equivalently as

$$x^S = \bar{x}_0^S + \sum_{j=1}^{d} (\bar{x}_j^S - \bar{x}_0^S)\delta_j^S \tag{2}$$

with $\sum_{j=1}^{d} \delta_j^S \leq 1$ and $\delta_j^S \geq 0$ for $j = 1, \ldots, d$.

The other main idea is that all simplices of a triangulation are ordered in such a way that the last vertex of any simplex is equal to the first vertex of the next one.

Algorithm 2.1 Global optimization of an MINLP by solving adaptively refined MIP relaxations

Input: An MINLP problem P of type (P), upper bounds $\epsilon_f^0 > 0$ for the absolute linearization errors in the piecewise linear approximations used to construct the initial relaxation and the maximal absolute linearization errors $\epsilon_f > 0$ for all $f \in \mathcal{F}$.

Output: If P is feasible, the algorithm returns an optimal solution x of an MIP relaxation Π of P with $|f(x_f) - y_f| \leq \epsilon_f$ for all $f \in \mathcal{F}$ and $c^\top x \leq c^\top x'$ for any feasible point x' of P. If no such MIP relaxation Π of P exists, this is reported by returning *infeasible*.

```
Set $\mathcal{F} \leftarrow$ all nonlinear functions that are contained in $P$.
Set $D_f, \epsilon_f^0, \epsilon_f \leftarrow$ the domain $D_f$ and the error bounds $\epsilon_f^0$, $\epsilon_f$ for all $f \in \mathcal{F}$.
Compute an initial piecewise linear approximation $\phi_f^0$ of $f \in \mathcal{F}$ satisfying the
    upper bound $\epsilon_f^0$ for all $f \in \mathcal{F}$.
Set $i \leftarrow 0$.
repeat
    Construct an MIP relaxation $\Pi^i$ of $P$ from $\phi_f^i$ for all $f \in \mathcal{F}$.
    Solve $\Pi^i$.
    if $\Pi^i$ is feasible then
        Set $x^i \leftarrow$ optimal solution of $\Pi^i$.
    else
        return infeasible.
    end if
    Set stop $\leftarrow$ true.
    for all $f \in \mathcal{F}$ do
        Set $x_f^i \leftarrow$ projection of $x^i$ on $D_f$.
        Set $y_f^i \leftarrow$ value of the variable for the relaxed function value of $f$.
        if $|f(x_f^i) - y_f^i| > \epsilon_f$ then
            Set $S_f^i \leftarrow$ simplex of triangulation $\mathcal{T}(\phi_f^i)$ that contains $x_f^i$.
            Set $\mathcal{T}(\phi_f^{i+1}) \leftarrow$ refinement of $\mathcal{T}(\phi_f^i)$ (by refining $S_f^i$).
            Set $\phi_f^{i+1} \leftarrow$ piecewise linear approximation according to $\mathcal{T}(\phi_f^{i+1})$.
            Set stop $\leftarrow$ false.
        else
            Set $\phi_f^{i+1} \leftarrow \phi_f^i$.
        end if
    end for
    Set $i \leftarrow i + 1$.
until stop.
return $x^{i-1}$.
```

In this way, we can construct a Hamiltonian path and model the piecewise linear approximation along this path. It is known that modeling piecewise linear functions by the generalized incremental method is possible if an ordering of the simplices with the following properties is available:

(O1) The simplices in $\mathcal{T} = \{S_1, \dots, S_n\}$ are ordered such that $\mathcal{V}(S_i) \cap \mathcal{V}(S_{i+1}) \neq \emptyset$ for $i = 1, \dots, n-1$, and

(O2) for each simplex S_i its vertices $\bar{x}_0^{S_i}, \dots, \bar{x}_d^{S_i}$ can be labeled such that $\bar{x}_d^{S_i} = \bar{x}_0^{S_{i+1}}$ for $i = 1, \dots, n-1$.

Hence, we only have to show that a red refined triangulation has properties (O1) and (O2) to utilize the generalized incremental method.

3. Convergence Result

Algorithm 3.1 Generalized red refinement of a simplex S

Input: A simplex S with $\mathcal{V}(S) = \{\bar{x}_0, \dots, \bar{x}_d\}$ and a scalar δ.

Output: If the longest edge of S is greater than δ, a set of 2^d simplices $\{S^0, \dots, S^{2^d-1}\}$ with $S = \cup_{i=0}^{2^d-1} S^i$ and $\operatorname{int}(S^i) \cap \operatorname{int}(S^j) = \emptyset$ for all $i \neq j$ is returned. Otherwise no refinement is performed.

```
Set $e \leftarrow$ longest edge of $S$.
if $\|e\| \leq \delta$ then
  return $\mathcal{V}(S)$.
else
  Set $i \leftarrow -1$.
  for $0 \leq k \leq d$ do
    Set $v_0 \leftarrow \frac{1}{2}(\bar{x}_0 + \bar{x}_k)$.
    for $\tau \in \mathrm{Sym}_d$ do
      if $\tau^{-1}(1) < \cdots < \tau^{-1}(k)$ and $\tau^{-1}(k+1) < \cdots < \tau^{-1}(d)$ then
        for $1 \leq l \leq d$ do
          Set $v_l \leftarrow v_{l-1} + \frac{1}{2}(\bar{x}_{\tau(l)} - \bar{x}_{\tau(l)-1})$.
        end for
        Set $i \leftarrow i + 1$.
        Set $\mathcal{V}(S^i) \leftarrow \{v_0, \dots, v_d\}$ ; $S^i \leftarrow \mathrm{conv}(\mathcal{V}(S^i))$.
      end if
    end for
  end for
  return $\mathcal{V}(S^0), \dots, \mathcal{V}(S^{2^d-1})$.
end if
```

In this article, we consider Algorithm 3.1 that is proposed in [1, 5] as the refinement procedure in Algorithm 2.1. It is a generalization of the red refinement strategy, which originally was only developed for triangles. It is known that the generalized red refinement procedure always delivers a triangulation of a simplex (that has to be refined) by 2^d sub-simplices. Moreover, the triangulation is *consistent*, i.e., the intersection of any two sub-simplices is either empty or a common lower-dimensional simplex with respect to the vertex sets. Consequently, a consistent triangulation does not allow for hanging nodes, i.e., nodes that are contained in the vertex set of a simplex S, but not in all vertex sets of the simplices that are adjacent to S. We again point out that the theoretical results in this and the subsequent section are part of the author's PhD thesis [2]. We first illustrate the refinement by Algorithm 3.1 using an example in dimension two.

Example 3.2. We consider a simplex S_l of some triangulation of a two-dimensional nonlinear function with vertex set $\mathcal{V}(S_l) = \{\bar{x}_0, \bar{x}_1, \bar{x}_2\}$ that has to be refined. Let the scalar δ be sufficiently large such that a refinement is performed by Algorithm 3.1. Note that for $k \leq 1$ the condition $\tau^{-1}(1) < \cdots < \tau^{-1}(k)$ is considered to be fulfilled. The same applies for the condition $\tau^{-1}(k+1) < \cdots < \tau^{-1}(d)$ in case of $k \geq d-1$.

First, the symmetry group Sym_2 has two permutations: the identity $\tau_1 = \mathrm{id}$ and the permutation $\tau_2 \colon \{1,2\} \to \{2,1\}$. The identity fulfills the condition in Line 9 for all $0 \leq k \leq 2$.

Since τ_2 does not satisfy Line 9 for $k=0$, we obtain the first corner sub-simplex S_l^0 for τ_1. The vertices of S_l^0 are

$$v_0 = \bar{x}_0, \quad v_1 = v_0 + \frac{1}{2}(\bar{x}_1 - \bar{x}_0), \quad v_2 = v_1 + \frac{1}{2}(\bar{x}_2 - \bar{x}_1); \tag{3}$$

see Figure 1 for an illustration.

For $k=1$ both permutations τ_1 and τ_2 comply with Line 9. Equivalent to $k=0$, with τ_1 we obtain the simplex S_l^1 as in (3), while now $v_0 = \bar{x}_0 + \frac{1}{2}(\bar{x}_1 - \bar{x}_0)$. We thus obtain the corner sub-simplex S_l^1 simply by translating the corner sub-simplex S_l^0 by the vector $\frac{1}{2}(\bar{x}_1 - \bar{x}_0)$. For τ_2, we compute the vertices of the simplex S_l^2 as

$$v_0 = \frac{1}{2}(\bar{x}_1 - \bar{x}_0), \quad v_1 = v_0 + \frac{1}{2}(\bar{x}_2 - \bar{x}_1), \quad v_2 = v_1 + \frac{1}{2}(\bar{x}_1 - \bar{x}_0). \tag{4}$$

Finally, for $k=1$ again only the identity τ_1 fulfills the condition in Line 9. We obtain the simplex S_l^3 as in (3) with $v_0 = \bar{x}_0 + \frac{1}{2}(\bar{x}_2 - \bar{x}_0)$. The corner sub-simplex S_l^3 again corresponds to a translation of S_l^0 by the vector $\frac{1}{2}(\bar{x}_2 - \bar{x}_0)$.

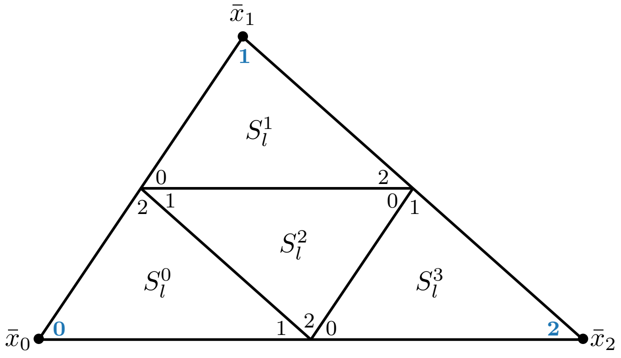

FIGURE 1. A red refinement of a two-dimensional simplex with corresponding ordering and labeling of the four sub-simplices. The labeled vertices of the original simplex that is refined are marked in bold blue.

We now prove the δ-preciseness of the refinement procedure and show how to model a refined triangulation by the generalized incremental method afterward. Let $\widetilde{\mathcal{T}}_k$ be the refined triangulation of an initial triangulation $\mathcal{T}_0$ of D_f obtained by applying Algorithm 3.1 such that in every iteration $i \leq k$ all simplices of $\widetilde{\mathcal{T}}_{i-1}$ are refined.

Lemma 3.3. *Let $S \in \mathbb{R}^d$ be a simplex of $\mathcal{T}_0$ and e the longest edge of S. Then, the longest edge of any simplex of $\widetilde{\mathcal{T}}_l$ contained in (the set) S is bounded by $\frac{1}{2^l}\|e\|$ with $l \in \mathbb{N}$.*

Proof. The lemma follows directly from Line 11 of Algorithm 3.1, since

$$\left\| \frac{1}{2}(\bar{x}_{\tau(l)} - \bar{x}_{\tau(l)-1}) \right\| \leq \frac{1}{2}\|e\|$$

are the lengths of the edges of the sub-simplices that are constructed during the first refinement step. Applying this recursively finishes the proof. □

Lemma 3.4. *Let $\delta > 0$, then there is an $\tilde{N} \in \mathbb{N}$, such that $\widetilde{\mathcal{T}}_{\tilde{N}}$ is a refinement of every triangulation obtained by applying Algorithm 3.1 to $\mathcal{T}_0$ with δ as input parameter.*

Proof. Let e_0 be the longest edge of all simplices of $\mathcal{T}_0$. With Lemma 3.3 and $\tilde{N} := \max\{0, \lceil \ln(\frac{\|e_0\|}{\delta})/\ln(2)\rceil\}$, the proof works equivalently to the one of Theorem 3.4 from [3]: After at most $\tilde{N}$ refinement steps the longest edge of any simplex of $\widetilde{\mathcal{T}}_{\tilde{N}}$ is bounded by δ. Since Algorithm 3.1 only refines simplices with a longest edge larger than δ and no simplex in $\widetilde{\mathcal{T}}_{\tilde{N}}$ has an edge longer than δ, it follows by the pigeonhole principle that $\widetilde{\mathcal{T}}_{\tilde{N}}$ is the finest refinement of $\mathcal{T}_0$ that is achievable. □

Theorem 3.5. *Algorithm 3.1 as refinement procedure in Algorithm 2.1 is δ-precise for every $\delta > 0$. With $\tilde{N}$ as in Lemma 3.4, the number of refinement steps N as in Definition 2.1 is bounded from above by*

$$N := m(2^{\tilde{N}d} - 1) + 1,$$

where m is the number of simplices contained in $\mathcal{T}_0$.

Proof. We count every single simplex that has to be refined to achieve $\widetilde{\mathcal{T}}_{\tilde{N}}$ from Lemma 3.4 and obtain

$$m(1 + 2^d + 2^{2d} + \ldots + 2^{\tilde{N}d-1}) = m(2^{\tilde{N}d} - 1)$$

refinements in total. The rest of the proof follows by the pigeonhole principle as in the proof of Theorem 3.5 from [3]: Every sequence $(S^i) \in \mathcal{T}_i$ of simplices has an element S^k with index $k \leq m(2^{\tilde{N}d} - 1) + 1$, such that $S^k \in \widetilde{\mathcal{T}}_{\tilde{N}}$, since $\widetilde{\mathcal{T}}_{\tilde{N}}$ is a refinement of every triangulation obtained by Algorithm 3.1 with parameter δ. Therefore, simplex S^k has the δ-preciseness property (1), as no simplex in $\widetilde{\mathcal{T}}_{\tilde{N}}$ has an edge longer than δ. □

The next corollary follows directly from Proposition 2.2 and Theorem 3.5.

Corollary 3.6. *Algorithm 2.1 together with Algorithm 3.1 as refinement procedure is correct and terminates after a finite number of steps.*

4. Incremental Method for Red Refined Piecewise Linear Relaxations

We now show that a piecewise linear approximation that results from applying Algorithm 3.1 can also be modeled with the generalized incremental method. We first prove two lemmata that are used afterward to prove the main result of this section.

Lemma 4.1. *Let $\mathcal{S} = \{S^0, \ldots, S^{2^d-1}\}$ be a refinement of a simplex S by Algorithm 3.1 with $\mathcal{V}(S) = \{\bar{x}_0, \ldots, \bar{x}_d\}$. Then, each simplex of the subset of the corner sub-simplices $\mathcal{S}' = \{S^{i_0}, \ldots, S^{i_d}\}$ of $\mathcal{S}$ contains a vertex of the simplex S, i.e., $\bar{x}_j \in \mathcal{V}(S^{i_j})$ for all $j = 0, \ldots, d$. Moreover, for each pair of simplices $S^{i_j}, S^{i_k} \in \mathcal{S}'$ the midpoint m_{jk} of the edge with endpoints $\bar{x}_j$ and $\bar{x}_k$ is contained in both vertex sets of the simplices.*

Proof. First, the identity id $\in \mathrm{Sym}_d$ always fulfills the conditions from Line 9 of Algorithm 3.1. Let S^{i_j} be the simplex that is constructed using the starting vertex $v_0 = \frac{1}{2}(\bar{x}_0 + \bar{x}_j)$ and $\tau = \mathrm{id}$, where $j = 0, \ldots, d$. Due to the telescoping sum in Line 11, it follows that

$$v_j = \underbrace{\underbrace{\underbrace{\frac{1}{2}(\bar{x}_0 + \bar{x}_j)}_{v_0} + \frac{1}{2}(\bar{x}_1 - \bar{x}_0)}_{v_1} + \frac{1}{2}(\bar{x}_2 - \bar{x}_1)}_{\vdots} + \cdots + \frac{1}{2}(\bar{x}_j - \bar{x}_{j-1}) = x_j \tag{5}$$

is contained in the vertex set of S^{i_j}.

Furthermore, due to the telescoping sum in (5), we can rewrite Line 11 as

$$v_l \leftarrow \frac{1}{2}(\bar{x}_0 + \bar{x}_j) + \frac{1}{2}(\bar{x}_l - \bar{x}_0) = \frac{1}{2}(\bar{x}_j + \bar{x}_l)$$

for $\tau = \mathrm{id}$. Since $m_{jk} = \frac{1}{2}(\bar{x}_j + \bar{x}_k)$, we conclude that the vertices v_k and v_j that occur during the construction of S^{i_j} and S^{i_k}, respectively, are equal to m_{jk}. □

Lemma 4.2. *Let $\mathcal{S}'' = \{S^0, \ldots, S^{2^d-1-(d+1)}\}$ be a refinement of a simplex S by Algorithm 3.1 without the $d+1$ corner sub-simplices of $\mathcal{S}'$ as in Lemma 4.1. Then, the union of all simplices of $\mathcal{S}''$ is a (convex) polytope and the triangulation of $\mathcal{S}''$ has an ordering with the properties (O1) and (O2).*

Proof. Alternatively to the vertex description, we can describe the simplex S by its half-space representation

$$S = \left\{ x \in \mathbb{R}^d \, : \, a_j^\top x \leq b_j \;\text{ with } a_j \in \mathbb{R}^d \text{ and } b_j \in \mathbb{R} \text{ for } j = 0, \ldots, d \right\}. \tag{6}$$

We now describe the set $\mathcal{S}''$ by adding the inequalities that separate all corner sub-simplices from the set S as in (6). For a vertex of S exactly d inequalities in (6) are tight. Due to Lemma 4.1, the vertex set of the corner sub-simplex $S^{i_j} \in \mathcal{S}'$ consists of the vertex $\bar{x}_j$ and all midpoints m_{jk} with $k = 0, \ldots, d$. Let $a_j^\top x \leq b_j$ be the inequality that is not tight for $\bar{x}_j$. Naturally, it is tight for all other vertices of S and it follows that

$$a_j^\top m_{jk} = \frac{1}{2} a_j^\top (\bar{x}_k + \bar{x}_j) = \frac{1}{2}(b_j + a_j^\top \bar{x}_j). \tag{7}$$

Moreover, since the red refinement also delivers a triangulation of S, where all interiors of the sub-simplices are disjunct, we can describe S^{i_j} by substituting $a_j^\top x \leq b_j$ with $a_j^\top x \leq \frac{1}{2}(b_j + a_j^\top \bar{x}_j)$ in (6). Therefore, by separating all corner sub-simplices, we obtain the half-space description

$$\mathcal{S}'' = \Big\{ x \in \mathbb{R}^d \, : \, a_j^\top x \leq b_j, \; a_j^\top x \geq \frac{1}{2}(b_j + a_j^\top \bar{x}_j) \\ \text{with } a_j \in \mathbb{R}^d \text{ and } b_j \in \mathbb{R} \text{ for } j = 0, \ldots, d \Big\}.$$

It follows that the union of all simplices of $\mathcal{S}''$ is a convex polytope.

The consistency of the triangulation of S translates to the triangulation of $\mathcal{S}''$, because only the corner sub-simplices are omitted. It is shown by [6] that a triangulation with the property that for each nonempty subset $\tilde{\mathcal{S}} \subsetneq \mathcal{T}$ of a triangulation $\mathcal{T}$ there exist simplices $S_1 \in \tilde{\mathcal{S}}$ and $S_2 \in \mathcal{T} \setminus \tilde{\mathcal{S}}$ such that S_1, S_2 have d common vertices, always yields a triangulation with the properties (O1) and (O2). Therefore, we only have to prove that the triangulation of $\mathcal{S}''$ has this property. To this end, we consider a nonempty subset $\tilde{\mathcal{S}}$ of $\mathcal{S}''$. Each facet of a d-dimensional simplex consists of d vertices of the simplex. Let $\tilde{\mathcal{S}}_{\mathrm{F}}$ be the set of all facets of the simplices of $\tilde{\mathcal{S}}$, where the simplices are described by the convex hull of its vertices. Then, each facet in $\tilde{\mathcal{S}}_{\mathrm{F}}$ is either a facet of $\mathcal{S}''$ (in the sense of convex hulls) or a common facet of two simplices of $\tilde{\mathcal{S}}$. This is due to the consistency of the triangulation. Since $\tilde{\mathcal{S}} \subsetneq \mathcal{S}''$, however, there must be a facet in $\tilde{\mathcal{S}}_{\mathrm{F}}$ that is a common facet of two simplices S_1, S_2 such that $S_1 \in \tilde{\mathcal{S}}$ and $S_2 \in \mathcal{S}'' \setminus \tilde{\mathcal{S}}$. □

Endowed with Lemma 4.1 and 4.2, we are now ready to prove the main result of this section.

Theorem 4.3. *Let $\mathcal{T}$ be a triangulation with the properties (O1) and (O2). Then, any triangulation $\mathcal{T}'$ obtained by applying Algorithm 3.1 to $\mathcal{T}$ maintains the properties (O1) and (O2).*

Proof. For the following proof, we first show that there is an ordering of the sets $\mathcal{S}'$ and $\mathcal{S}''$ from Lemma 4.1 and 4.2 with the properties (O1) and (O2). The second part of the proof merges these two orderings to obtain an overall ordering with the properties (O1) and (O2).

Let $\mathcal{T} = \{S_1, \dots, S_n\}$ and S_l be the simplex that has to be refined, while $\bar{x}_0^{S_l}, \dots, \bar{x}_d^{S_l}$ are its labeled vertices. We first consider $d \geq 4$ and $d = 2, 3$ afterward. The 2^d simplices $S_l^0, \dots, S_l^{2^d-1}$, into which S_l is divided by the red refinement, have due to Lemma 4.1 the property that the corner sub-simplices contain the vertices of S_l. Without loss of generality, let $\bar{x}_j^{S_l} \in \mathcal{V}(S_l^j)$.

We first show that the set $\mathcal{S}'$ of the corner sub-simplices has an ordering with the required properties. The corner sub-simplices $S_l^j \in \mathcal{S}'$ yield a complete graph $G = (V, E)$ with the simplices as the node set V and the midpoints m_{jk} as the edge set E connecting the simplices S_l^j and S_l^k. Due to $m_{jk} = m_{kj}$, we assume in the following for the notation m_{jk} that $j < k$ holds. For each Hamiltonian path in G, we can use the path itself as an ordering of the simplices that correspond to the nodes of G. An edge connecting two consecutive nodes of the path corresponds to a common midpoint of two consecutive simplices. Therefore, the ordering naturally has property (O1), which indicates that two consecutive simplices have at least one common vertex. The ordering has the property (O2) as well, which states that the last vertex of any simplex is equal to the first vertex of the next one: With two consecutive simplices S_l^j and S_l^k that correspond to two consecutive nodes of the Hamiltonian path, we only have to set $\bar{x}_d^{S_l^j} = m_{jk}$ and $\bar{x}_0^{S_l^k} = m_{jk}$.

Moreover, it follows from Lemma 4.2 that there is an ordering

$$(S_l^{i_0}, \dots, S_l^{i_{2^d-1-(d+1)}}) \tag{8}$$

of the simplex set $\mathcal{S}''$ with the properties (O1) and (O2). Since the vertices of the simplices of $\mathcal{S}''$ are the midpoints m_{jk}, there must be two midpoints m_{jk} and m_{st} with

$$m_{jk} = \bar{x}_0^{S_l^{i_0}} \quad \text{and} \quad m_{st} = \bar{x}_d^{S_l^{i_{2^d-1-(d+1)}}}.$$

We now link the orderings of $\mathcal{S}'$ and $\mathcal{S}''$ to obtain an overall ordering with the properties (O1) and (O2). Let R be a Hamiltonian path in the sub-graph of G that consists of the vertices $V \setminus \{0, j, k, s, t, d\}$. Such a path is always attainable, because any sub-graph of a complete graph is also complete. With $j \neq k$ one of the following three cases applies to the node j: $j = s$, $j = t$, or $j \neq s \wedge j \neq t$. With $s \neq t$, we have the following three cases for the node s: $s = j$, $s = k$, or $s \neq j \wedge s \neq k$. Consequently, due to $j \neq k$ and $s \neq t$ only the following five cases are possible for the nodes j, k, s, and t:

$$j = s \,\wedge\, k \neq t, \quad j = s \,\wedge\, k = t, \quad j \neq s \,\wedge\, k = t, \quad k = s, \quad j \neq s \,\wedge\, k \neq t. \tag{9}$$

The case $k = s$ is equivalent to the case $j = t$, since the inverse of the ordering (8) also has the properties (O1) and (O2).

Keeping the cases in (9) in mind, we define the path

$$H = \left\{ \begin{array}{lll} (0, j, t, R, k, d), & \text{if } j = s \,\wedge\, k \neq t, & \text{(10a)} \\ (0, R, j, k, d), & \text{if } j = s \,\wedge\, k = t, & \text{(10b)} \\ (0, j, s, R, t, d), & \text{if } k = s, & \text{(10c)} \\ (0, j, s, R, k, t, d), & \text{otherwise.} & \text{(10d)} \end{array} \right.$$

Please note that if $t = d$ in (10a), we can again invert the ordering (8) such that $t \neq d$ and $k = d$. The same can be applied in case of $k = d$ in (10c) and (10d). Moreover, any permutation of the labeling $\bar{x}_i$ of the simplex $S_l^{i_0}$, where $i = 0, \dots, d-1$ and of the simplex $S_l^{i_{2^d-1-(d+1)}}$, where $i = 1, \dots, d$, is permissible. Thus, we can assume that $m_{st} \neq m_{0d}$ and that $m_{jk} \neq m_{0u}$ if $m_{st} = m_{ud}$. This guarantees us that the path H is always a Hamiltonian path.

The path H corresponds to an ordering, where the vertices in H are the corner sub-simplices $S_l^j \in \mathcal{S}'$. As showed above, this ordering has the properties (O1) and (O2). We now insert the ordering (8) of the simplex set $\mathcal{S}''$ into the one of H after the simplex that corresponds to the vertex j. The union of $\mathcal{S}'$ and $\mathcal{S}''$ corresponds to the set of all 2^d sub-simplices into which S_l is divided by the red refinement. Therefore, the resulting ordering covers all 2^d sub-simplices. The merging of the two orderings of $\mathcal{S}'$ and $\mathcal{S}''$ via the midpoints m_{jk} and m_{st} finally leads to an overall ordering $(S_l^{r_0}, \dots, S_l^{r_{2^d-1}})$ that has the properties (O1) and (O2).

In case of $d = 2$, we use the ordering $(S_l^0, S_l^1, S_l^2, S_l^3)$, where S_l^0, S_l^1, and S_l^3 are the three corner sub-simplices and S_l^2 the remaining center sub-simplex, see again Figure 1 for an illustration.

For $d = 3$, we assume without loss of generality that the vertex $\bar{x}_3$ of the simplex that has to be refined is contained in S_l^7. We use the ordering $(S_l^0, S_l^1, S_l^2, S_l^3, S_l^4, S_l^5, S_l^6, S_l^7)$, where S_l^0, S_l^1, S_l^2, and S_l^7 are the four corner sub-simplices contained in $\mathcal{S}'$ and S_l^3–S_l^6 the four center sub-simplex of $\mathcal{S}''$. Finally, the orderings of $\mathcal{S}'$ and $\mathcal{S}''$ are linked via $m_{jk} = m_{02}$ and $m_{st} = m_{27}$.

With these orderings for the refined simplex S_l and all dimensions $d \geq 2$, we complete the proof as follows. We order the simplices of $\mathcal{T}'$ as

$$(S_1, \dots, S_{l-1}, S_l^{r_0}, \dots, S_l^{r_{2^d-1}}, S_{l+1}, \dots, S_n). \tag{11}$$

Since the corner sub-simplices of S_l yield a complete graph, we can order them such that

$$\bar{x}_d^{S_{l-1}} = \bar{x}_0^{S_l^{r_0}} \qquad \text{and} \qquad \bar{x}_d^{S_l^{r_{2^d-1}}} = \bar{x}_0^{S_{l+1}}$$

holds. Therefore, the simplices S_{l-1}, $S_l^{r_0}$ and $S_l^{r_{2^d-1}}$, S_{l+1} are linked as required in (O2). Altogether, due to the inheritance from $\mathcal{T}$, we conclude that the ordering (11) of the simplices of $\mathcal{T}'$ has the properties (O1) and (O2). □

5. Numerical Results

In this section, we numerically analyze to what extent the red refinement procedure can be used beneficially in an MINLP solution approach that is based on Algorithm 2.1. The key element of Algorithm 2.1 is to solve MIP relaxations that provide dual bounds for the MINLP problem due to the relaxation property. With increasingly finer MIP relaxations, the resulting dual bounds also become tighter. Concurrently, the approximation error decreases, which yields an MIP solutions that is feasible for the MINLP problem from a practical point of view, e.g., if the error is smaller than 10^{-6}.

Naturally, any objective value of a global optimal solution of an MIP relaxation provides a dual bound for the MINLP problem. At the same time, the dual bound of any MIP relaxation is also a dual bound for the MINLP. Typically, in a solution framework that is based on Algorithm 2.1 this is exploited by solving a very fine MIP relaxation. It is not necessary to solve this MIP to global optimality, since each dual bound of the MIP that is obtained during the solution process already provides a dual bound for the MINLP. Therefore, we can employ a very fine initial MIP relaxation in Algorithm 2.1 also to find tight dual bounds.

One way to construct an initial relaxation is to specify a fixed number of simplices for each domain D_f with $f \in \mathcal{F}$ and apply the refinement strategy successively to each simplex until the desired number of simplices is attained. This corresponds to an triangulation scheme such as $\widetilde{\mathcal{T}}_k$ in Lemma 3.3.

In the following, we pursue such a triangulation strategy and compare the red refinement with the longest-edge bisection (LEB) from [3]. To this end, we consider all 20 alternating current (AC) optimal power flow (OPF) instances from the NESTA benchmark set (v0.7.0) with up to 300 buses, see [4]. These instances are subdivided into three operating conditions: standard, active power increase (API), and small angle difference (SAD). We use the Extended Conic Quadratic Formulation of [10] to model an AC OPF instance. Thus, we obtain a total amount of 59 MINLP problems, since for the case "nesta_case_6_ww" no active power increase instance is given.

We skip the details of the MINLP models in order not to go beyond the scope of this article. The nonlinearities of the MINLP problems consist of quadratic monomials, bivariate products, and one-dimensional trigonometric functions. The triangulations that are constructed by the red refinement and the longest-edge bisection differ for the domains of the bivariate products only, as they are identical for one-dimensional domains.

All computations are carried out using a Python implementation of Algorithm 2.1 on a cluster using 4 cores of a machine with two Xeon E3-1240 v6 "Kaby Lake" chips (4 cores, HT disabled) running at 3.7 GHz with 32 GB of RAM. The run time limit is set to 4 hours for each instance with a global relative optimality gap of 0.01 %. We utilize Gurobi 9.1.1 as MIP solver, see [9].

5.1. 8 **Simplices.**

At first, we use 8 simplices, i.e., segments in case of one-dimensional and triangles in case of two-dimensional functions, to construct an initial MIP relaxation. Table 1 depicts the results for all 59 NESTA instances. Here, d_{LEB} and d_{RR} are the best dual bounds of the MIP relaxations constructed by the LEB and the red refinement (RR). Correspondingly, T_{LEB} and T_{RR} are the run times until the MIP relaxations are solved to global optimality, while TL indicates that the time limit of 4 hours has been exceeded.

A common measure for comparing two different MINLP solution approaches is the shifted geometric mean. The shifted geometric mean of n numbers $t_1, \dots, t_n$ with shift s is defined as $\left(\prod_{i=1}^{n} (t_i + s)\right)^{1/n} - s$. It has the advantage that it is neither affected by very large outliers (in contrast to the arithmetic mean) nor by very small outliers (in contrast to the geometric mean). We use a typical shift $s = 10$ for both the run times and the dual bounds. Please note that if an instance runs into the time limit, we use a run time of 4 hours for the calculation of the shifted geometric mean.

The geometric mean in Table 1 is 36.73 for d_{LEB} and 36.88 for d_{RR}. Regarding the run times, we have 256.12 s for T_{LEB} and 251.95 s for T_{RR}. Therefore, both refinement strategies are of equal quality, while the red refinement is slightly favorable, since it has tighter dual bounds, faster run times and is able to solve an additional instance to global optimality, namely the SAD instance “nesta_case_29_edin”.

TABLE 1. Results summary for the AC OPF MIP relaxations with 8 simplices for each nonlinearity.

nesta_case_	standard				API				SAD			
	d_{LEB}	d_{RR}	T_{LEB} (s)	T_{RR} (s)	d_{LEB}	d_{RR}	T_{LEB} (s)	T_{RR} (s)	d_{LEB}	d_{RR}	T_{LEB} (s)	T_{RR} (s)
3_lmbd	50.12	51.25	0.10	0.10	3.56	3.60	0.10	0.10	53.71	53.81	0.10	0.10
4_gs	0.99	0.99	0.10	0.10	7.23	7.23	0.10	0.10	2.71	2.78	0.10	0.10
5_pjm	112.67	112.67	0.10	0.10	28.14	28.14	0.10	0.10	237.77	240.81	0.10	0.10
6_ww	29.72	29.76	0.10	0.10					29.87	29.89	0.10	0.10
6_c	0.13	0.13	0.10	0.10	7.37	7.38	0.10	0.10	0.14	0.15	0.10	0.10
9_wscc	45.66	45.66	0.10	0.10	6.31	6.32	0.10	0.10	49.50	49.64	0.10	0.10
14_ieee	1.91	1.91	2.38	3.44	2.80	2.80	0.10	0.10	1.95	1.95	2.27	2.52
24_ieee_rts	564.57	565.50	14.74	29.98	48.83	48.88	2.00	3.39	638.83	638.38	54.07	93.55
29_edin	192.70	192.67	205.64	189.72	2 683.85	2 684.09	113.83	66.82	257.39	268.93	TL	954.43
30_fsr	1.68	1.71	20.51	14.78	0.69	0.70	8.58	18.42	2.34	2.44	1 301.17	991.37
30_as	4.46	4.49	16.79	26.95	4.12	4.17	11.86	19.57	5.23	5.38	1 304.96	830.28
30_ieee	0.93	0.93	37.98	28.80	2.99	2.97	192.66	120.62	1.00	0.99	1 293.06	6 593.64
39_epri	902.66	902.47	15.46	13.17	67.70	68.04	9.29	24.13	906.26	905.35	73.70	21.88
57_ieee	8.24	8.42	TL	TL	11.33	11.45	TL	TL	8.49	8.54	TL	TL
73_ieee_rts	1 677.63	1 679.92	TL	TL	162.64	162.87	3 568.03	5 978.80	1 951.89	1 949.64	TL	TL
89_pegase	34.20	34.28	TL	TL	12.07	12.07	TL	TL	34.53	34.04	TL	TL
118_ieee	25.64	25.78	TL	TL	42.62	42.71	TL	TL	25.90	26.12	TL	TL
162_ieee_dtc	31.49	31.65	TL	TL	46.86	47.03	TL	TL	31.49	31.67	TL	TL
189_edin	0.00	0.00	TL	TL	4.56	4.90	TL	TL	0.00	0.00	TL	TL
300_ieee	123.36	123.08	TL	TL	158.63	158.94	TL	TL	123.35	123.97	TL	TL

5.2. 32 **Simplices.**

Table 2 shows the results for all 59 NESTA instances if we use 32 simplices to construct the initial MIP relaxation. As the size of the relaxations increases, less instances are solved to global optimality. The geometric mean in Table 2 is 39.49 for d_{LEB} and 39.50 for d_{RR}. Regarding the run times, we have 1587.48 s for T_{LEB} and 1529.38 s for T_{RR}. Therefore, both refinement strategies are of equal quality, while the red refinement is again slightly favorable, since it has faster run times.

Table 2. Results summary for the AC OPF MIP relaxations with 32 simplices for each nonlinearity.

nesta_case_	standard				API				SAD			
	d_{LEB}	d_{RR}	T_{LEB} (s)	T_{RR} (s)	d_{LEB}	d_{RR}	T_{LEB} (s)	T_{RR} (s)	d_{LEB}	d_{RR}	T_{LEB} (s)	T_{RR} (s)
3_lmbd	55.74	55.79	5.44	0.10	3.61	3.62	0.10	0.10	57.02	56.84	0.10	0.10
4_gs	1.11	1.11	0.10	0.10	7.57	7.58	0.10	0.10	3.10	3.12	0.10	0.10
5_pjm	145.62	146.20	5.43	4.50	29.71	29.74	5.57	4.75	256.02	255.74	0.10	0.10
6_ww	31.05	31.06	0.10	4.12					31.10	31.11	2.05	17.65
6_c	0.21	0.22	9.84	8.13	8.05	8.06	1.51	2.70	0.23	0.23	38.55	21.17
9_wscc	52.38	52.53	15.54	13.72	6.52	6.53	1.11	0.10	54.32	54.09	2.06	0.10
14_ieee	2.37	2.38	1 974.83	1 855.49	3.17	3.17	578.62	314.10	2.37	2.38	4 727.75	6 592.35
24_ieee_rts	610.67	611.18	TL	TL	51.67	52.21	391.57	312.74	739.56	717.78	8 574.45	1 590.11
29_edin	238.01	219.58	TL	TL	2 831.18	2 836.86	TL	TL	298.35	317.08	TL	TL
30_fsr	3.42	3.54	TL	TL	1.43	1.39	TL	TL	3.21	3.43	TL	TL
30_as	5.92	5.72	TL	TL	5.11	4.99	TL	TL	6.70	6.81	TL	TL
30_ieee	1.18	1.14	TL	TL	3.51	3.56	TL	TL	1.27	1.20	TL	TL
39_epri	937.39	939.92	TL	TL	70.13	70.48	TL	TL	949.65	942.87	TL	TL
57_ieee	9.05	9.00	TL	TL	12.08	12.11	TL	TL	9.13	9.24	TL	TL
73_ieee_rts	1 771.64	1 773.22	TL	TL	167.33	167.44	TL	TL	2 117.09	2 102.50	TL	TL
89_pegase	34.90	35.53	TL	TL	12.34	12.19	TL	TL	36.57	36.12	TL	TL
118_ieee	27.48	27.52	TL	TL	46.34	46.38	TL	TL	28.38	28.79	TL	TL
162_ieee_dtc	33.39	33.59	TL	TL	49.64	49.87	TL	TL	33.40	33.71	TL	TL
189_edin	0.30	0.43	TL	TL	5.65	5.33	TL	TL	0.27	0.50	TL	TL
300_ieee	133.85	132.28	TL	TL	169.75	170.46	TL	TL	132.86	133.81	TL	TL

5.3. 128 **Simplices for Bivariate Products.**

As pointed out before, a very fine initial MIP relaxation can also be utilized to obtain tight dual bounds. With a high number of simplices for the triangulations the MIP size drastically increases. For one-dimensional domains the LEB and RR are identical. Since we are primarily interested in the difference between LEB and RR, we now use 128 simplices to construct the initial MIP relaxation, but only for all two-dimensional domains that correspond to the bivariate products of the problem. In case of one-dimensional functions, we only use 2 segments.

Table 3 contains the results for all 59 NESTA instances. The geometric mean in Table 3 is 23.36 for d_{LEB} and 23.40 for d_{RR}. Regarding the run times, we have 451.67 s for T_{LEB} and 330.84 s for T_{RR}. Hence, in the case of very fine initial relaxations the RR can be very advantageous. Although the dual bounds are of the same quality, the run time is significantly better. This suggests that with the RR, we are able to find tight dual bounds at shorter run times.

The numerical results in this section demonstrate that red refinement can be applied advantageously for initial relaxations in Algorithm 2.1. In particular, when the red refinement is used to construct very fine MIP relaxations to obtain tight dual bounds it is more favorable than the longest-edge bisection for MINLP problems that arise in the field of AC OPF.

6. Conclusion

In this paper, we showed that the generalized red refinement for n-dimensional simplices can be utilized for solving MINLP problems by piecewise linear relaxations. We proved that the red refinement meets sufficient convergence conditions for such an MINLP solution framework as proposed in [3]. Furthermore, we showed that applying

Table 3. Results summary for the AC OPF MIP relaxations with 128 triangles for each two-dimensional function and 2 segments for each one-dimensional function.

nesta_case_	standard				API				SAD			
	d_{LEB}	d_{RR}	T_{LEB} (s)	T_{RR} (s)	d_{LEB}	d_{RR}	T_{LEB} (s)	T_{RR} (s)	d_{LEB}	d_{RR}	T_{LEB} (s)	T_{RR} (s)
3_lmbd	32.70	32.70	1.02	0.10	3.48	3.48	0.10	0.10	33.62	33.62	0.10	0.10
4_gs	0.85	0.85	0.10	0.10	5.67	5.67	0.10	0.10	2.28	2.28	1.43	0.10
5_pjm	104.00	104.00	0.10	0.10	26.84	26.84	0.10	0.10	229.00	229.14	0.10	4.69
6_ww	22.06	22.06	5.63	0.10					22.35	22.35	3.98	0.10
6_c	0.00	0.00	1.38	2.43	4.18	4.19	6.83	4.92	0.00	0.00	3.65	6.58
9_wscc	23.51	23.51	3.28	2.20	4.95	4.95	2.24	1.55	25.50	25.46	6.63	4.50
14_ieee	0.00	0.00	32.43	28.60	0.03	0.04	1 785.20	2 470.70	0.00	0.00	123.81	159.88
24_ieee_rts	487.42	487.43	1 124.07	551.03	39.55	39.54	819.27	374.45	531.29	529.03	TL	296.79
29_edin	122.83	122.79	TL	TL	2 468.70	2 469.18	TL	TL	190.88	197.72	TL	TL
30_fsr	0.00	0.00	16.37	1.57	0.00	0.00	5.19	1.97	0.00	0.00	197.79	28.18
30_as	2.85	2.85	21.67	1.91	1.31	1.31	315.90	303.17	2.85	2.85	325.92	758.83
30_ieee	0.00	0.00	83.70	24.99	0.00	0.00	88.74	6.50	0.00	0.00	746.26	337.23
39_epri	692.08	692.07	111.55	137.20	57.96	57.96	173.01	81.29	700.63	700.64	2 065.10	967.07
57_ieee	0.00	0.00	1 651.80	1 770.00	0.00	0.00	432.73	652.38	0.15	0.16	TL	TL
73_ieee_rts	1 413.38	1 413.50	TL	TL	125.79	125.87	TL	TL	1 487.14	1 522.29	TL	TL
89_pegase	16.04	16.04	2 581.74	1 602.08	6.37	6.37	5 114.17	TL	16.04	16.04	11 324.58	TL
118_ieee	3.88	3.87	TL	TL	9.89	9.94	TL	TL	4.74	4.71	TL	TL
162_ieee_dtc	11.02	11.04	TL	TL	18.45	18.47	TL	TL	11.31	11.26	TL	TL
189_edin	0.00	0.00	12 633.28	481.34	0.00	0.00	12 633.28	3 402.69	0.00	0.00	12 633.28	3 317.47
300_ieee	22.65	23.36	TL	TL	40.53	40.40	TL	TL	24.46	24.66	TL	TL

this refinement procedure results in piecewise linear relaxations that can be modeled by the well-known incremental method established by Markowitz and Manne [13].

Numerical results from the field of alternating current optimal power flow illustrated that the application of the red refinement procedure as an alternative to the longest-edge bisection can be advantageous in practice. Especially, if the MIP relaxation is primarily used to obtain tight dual bounds, the red refinement seems to be favorable.

The most important subject of future research is to what extent the combination of the two refinement strategies provides further potential for improvement. One possibility is to implement a hybrid approach that starts with an initial relaxation constructed by the red refinement and subsequently refines it by applying the longest-edge bisection.

7. Acknowledgements

The author thanks the Deutsche Forschungsgemeinschaft for their support within Project B07 of the Sonderforschungsbereich/Transregio 154 “Mathematical Modelling, Simulation and Optimization using the Example of Gas Networks”. This research has been performed as part of the Energie Campus Nürnberg and is supported by funding of the Bavarian State Government. Finally, the author is very grateful to Kevin-Martin Aigner for fruitful discussion regarding the numerical results and to Lena Hupp for the thorough reading of the paper.

References

[1] J. Bey. “Simplicial grid refinement: on Freudenthal’s algorithm and the optimal number of congruence classes”. In: *Numerische Mathematik* 85.1 (Mar. 1, 2000), pp. 1–29. doi: 10.1007/s002110050475.

[2] R. Burlacu. “Adaptive Mixed-Integer Refinements for Solving Nonlinear Problems with Discrete Decisions”. PhD thesis. Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), 2020.

[3] R. Burlacu, B. Geißler, and L. Schewe. “Solving mixed-integer nonlinear programmes using adaptively refined mixed-integer linear programmes”. In: *Optimization Methods and Software* (2019), pp. 37–64. DOI: 10.1080/10556788.2018.1556661.

[4] C. Carleton, D. Gordon, and P. Scott. “NESTA, The NICTA Energy System Test Case Archive”. In: *CoRR* abs/1411.0359 (2014). URL: http://arxiv.org/abs/1411.0359.

[5] H. Freudenthal. “Simplizialzerlegungen von Beschränkter Flachheit”. In: *Annals of Mathematics* 43.3 (1942), pp. 580–582. DOI: 10.2307/1968813.

[6] B. Geißler. “Towards Globally Optimal Solutions of MINLPs by Discretization Techniques with Applications in Gas Network Optimization”. PhD thesis. FAU Erlangen-Nürnberg, 2011.

[7] B. Geißler, A. Martin, A. Morsi, and L. Schewe. “Using Piecewise Linear Functions for Solving MINLPs”. In: pp. 287–314. DOI: 10.1007/978-1-4614-1927-3_10.

[8] M. Gugat, G. Leugering, A. Martin, M. Schmidt, M. Sirvent, and D. Wintergerst. “Towards simulation based mixed-integer optimization with differential equations”. In: *Networks* (2018). DOI: 10.1002/net.21812.

[9] L. Gurobi Optimization. *Gurobi Optimizer Reference Manual*. 2020. URL: http://www.gurobi.com.

[10] R. A. Jabr. “Radial distribution load flow using conic programming”. In: *IEEE Transactions on Power Systems* 21.3 (2006), pp. 1458–1459. DOI: 10.1109/TPWRS.2006.879234.

[11] J. Lee and D. Wilson. “Polyhedral methods for piecewise-linear functions. I. The lambda method”. In: *Discrete Appl. Math.* 108.3 (2001), pp. 269–285. DOI: 10.1016/S0166-218X(00)00216-X.

[12] A. Lundell, A. Skjäl, and T. Westerlund. “A reformulation framework for global optimization”. In: *Journal of Global Optimization* 57.1 (2013), pp. 115–141. DOI: 10.1007/s10898-012-9877-4.

[13] H. M. Markowitz and A. S. Manne. “On the Solution of Discrete Programming Problems”. In: *Econometrica* 25.1 (1957), pp. 84–110.

[14] A. Martin, M. Möller, and S. Moritz. “Mixed Integer Models for the Stationary Case of Gas Network Optimization”. In: *Mathematical Programming* 105.2 (Feb. 1, 2006), pp. 563–582. DOI: 10.1007/s10107-005-0665-5.

[15] R. Misener and C. A. Floudas. “Piecewise-linear approximations of multidimensional functions”. In: *J. Optim. Theory Appl.* 145.1 (2010), pp. 120–147. DOI: 10.1007/s10957-009-9626-0.

[16] A. Morsi. “Solving MINLPs on Loosely-Coupled Networks with Applications in Water and Gas Network Optimization”. PhD thesis. Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), 2013.

[17] S. Rebennack and J. Kallrath. “Continuous piecewise linear delta-approximations for bivariate and multivariate functions”. In: *J. Optim. Theory Appl.* 167.1 (2015), pp. 102–117. DOI: 10.1007/s10957-014-0688-2.

[18] S. Rebennack and J. Kallrath. “Continuous piecewise linear delta-approximations for univariate functions: computing minimal breakpoint systems”. In: *J. Optim. Theory Appl.* 167.2 (2015), pp. 617–643. DOI: 10.1007/s10957-014-0687-3.

[19] R. Rovatti, C. D'Ambrosio, A. Lodi, and S. Martello. "Optimistic MILP modeling of non-linear optimization problems". In: *European J. Oper. Res.* 239.1 (2014), pp. 32–45. DOI: 10.1016/j.ejor.2014.03.020.

[20] J. P. Vielma, S. Ahmed, and G. L. Nemhauser. "Mixed-Integer Models for Non-separable Piecewise-Linear Optimization: Unifying Framework and Extensions". In: *Operations Research* 58.2 (2010), pp. 303–315. DOI: 10.1287/opre.1090.0721.

[21] D. Wilson. "Polyhedral methods for piecewise-linear functions". PhD thesis. University of Kentucky, 1998.

[1] Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Discrete Optimization, Cauerstr. 11, 91058 Erlangen, Germany

[2] Energie Campus Nürnberg, Fürther Str. 250, 90429 Nuremberg, Germany