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

ROBERT BURLACU

ABSTRACT. We investigate the generalized red refinement for \( n \)-dimensional simplices that dates back to Freudenthal [4] 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 [10].

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 [6, 7, 9, 11, 13]. 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 a priori given accuracy as in [14, 15]. Complementary, optimal polynomial relaxations of one-dimensional functions are constructed in [13]. For up to three-dimensional functions, explicit approximation techniques for general nonlinear functions are proposed in [12]. 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 [16] 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 [17]. Among the most important ones is the incremental method of Markowitz and Manne, see [10], which is originally developed for one-dimensional functions. A generalization to higher dimensions is described in [8, 18] and [5]. Supplementary to this, an extension to relaxations is given in [6].

In this paper, we consider the MINLP solution method proposed in [3, 5] and further developed in [2] that tackles an MINLP by solving a series of adaptively refined piecewise linear relaxations. The relaxations are based on linear approximations and
completely contain the graph of the function. They are defined on simplices, which in turn are defined by several vertices. The authors 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 [4]. 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.

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 finally prove that adaptively red refined piecewise linear relaxations can be modeled by the generalized incremental method.

2. Preliminaries

The aim of this article is to prove that the MINLP solution method proposed in [3] (Algorithm 1) is convergent if combined with the generalized red refinement procedure. The main idea of the MINLP solution approach is to use piecewise linear functions to construct MIP relaxations of the underlying MINLP. An iterative Algorithm 1 [3] is developed to find a global optimal solution by solving these relaxations, which are adaptively refined. With the domain \( D_f \) of a nonlinear function \( f \) that is contained in the MINLP, the refinement is performed on a simplicial triangulation \( T \) of \( D_f \). In [3], the classical longest-edge bisection is used for this. 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 relevant definitions and theorems from [3]. With a \( \delta \)-precise refinement procedure, Algorithm 1 [3] is both correct and convergent:

**Definition 2.1.** The refinement procedure in Algorithm 1 [3] is called \( \delta \)-precise, if for an arbitrary sequence \((S^i) \in T_i\) of simplices \( S^i \) that are refined by the refinement procedure with initial triangulation \( T_0 \) of \( D_f \) and given \( \delta > 0 \), there exists an index \( N \in \mathbb{N} \), such that

\[
\text{diam}(S^N) < \delta
\]

holds, where \( \text{diam}(S^N) := \sup_{x',x'' \in S^N} \{\|x' - x''\|}\).

**Proposition 2.2** (Theorem 3.6, [3]). *If the refinement procedure in Algorithm 1 [3] is \( \delta \)-precise for every \( \delta > 0 \), then Algorithm 1 [3] is correct and terminates after a finite number of steps.*
It is therefore sufficient to prove that the generalized red refinement is $\delta$-precise to show that its combination with Algorithm 1 [3] yields a correct and convergent algorithm.

The piecewise linear relaxations in Algorithm 1 [3] are modeled by the generalized incremental method; see [5]. There are two main ideas of the generalized incremental model. At first, any point $x^S$ inside a simplex $S$ with its vertex set $V(S) = \{\bar{x}_0^S, \ldots, \bar{x}_d^S\}$ can be expressed either as a convex combination of its vertices or equivalently as

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

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. 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 $T = \{S_1, \ldots, S_n\}$ are ordered such that $S_i \cap S_{i+1} \neq \emptyset$ for $i = 1, \ldots, n-1$, and

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

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

In this article, we consider Algorithm 3.1 that is proposed in [1, 4] as the refinement procedure in Algorithm 1 [3]. 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 $V(S_l) = \{\bar{x}_0, \bar{x}_1, \bar{x}_2\}$ that has to be refined. Let the scalar $\delta$ 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 $\text{Sym}_2$ has two permutations: the identity $\tau_1 = \text{id}$ and the permutation $\tau_2: \{1, 2\} \rightarrow \{2, 1\}$. The identity fulfills the condition in Line 9 for all $0 \leq k \leq 2$. 
Algorithm 3.1 Generalized red refinement of a simplex S

Input: A simplex S with \( \mathcal{V}(S) = \{ \vec{x}_0, \ldots, \vec{x}_d \} \) and a scalar \( \delta \).

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

1: Set \( e \leftarrow \) longest edge of \( S \).
2: if \( \|e\| < \delta \) then
3: return \( \mathcal{V}(S) \).
4: else
5: Set \( i \leftarrow -1 \).
6: for \( 0 \leq k \leq d \) do
7: Set \( v_0 \leftarrow \frac{1}{2} (\vec{x}_0 + \vec{x}_k) \).
8: for \( \tau \in \text{Sym}_d \) do
9: if \( \tau^{-1}(1) < \cdots < \tau^{-1}(k) \) and \( \tau^{-1}(k+1) < \cdots < \tau^{-1}(d) \) then
10: for \( 1 \leq l \leq d \) do
11: Set \( v_l \leftarrow v_{l-1} + \frac{1}{2} (\vec{x}_{\tau(l)} - \vec{x}_{\tau(l)-1}) \).
12: end for
13: end if
14: end for
15: return \( \mathcal{V}(S^0), \ldots, \mathcal{V}(S^{2^d-1}) \).
19: end if

Since \( \tau_2 \) does not satisfy Line 9 for \( k = 0 \), we obtain the first corner sub-simplex \( S_0^0 \) for \( \tau_1 \). The vertices of \( S_0^0 \) are

\[
v_0 = \vec{x}_0, \quad v_1 = v_0 + \frac{1}{2} (\vec{x}_1 - \vec{x}_0), \quad v_2 = v_1 + \frac{1}{2} (\vec{x}_2 - \vec{x}_1); \quad (3)
\]

see Figure 1 for an illustration.

For \( k = 1 \) both permutations \( \tau_1 \) and \( \tau_2 \) comply with Line 9. Equivalent to \( k = 0 \), with \( \tau_1 \) we obtain the simplex \( S_1^0 \) as in (3), while now \( v_0 = \vec{x}_0 + \frac{1}{2} (\vec{x}_1 - \vec{x}_0) \). We thus obtain the corner sub-simplex \( S_1^0 \) simply by translating the corner sub-simplex \( S_0^0 \) by the vector \( \frac{1}{2} (\vec{x}_1 - \vec{x}_0) \). For \( \tau_2 \), we compute the vertices of the simplex \( S_2^0 \) as

\[
v_0 = \frac{1}{2} (\vec{x}_1 - \vec{x}_0), \quad v_1 = v_0 + \frac{1}{2} (\vec{x}_2 - \vec{x}_1), \quad v_2 = v_1 + \frac{1}{2} (\vec{x}_1 - \vec{x}_0); \quad (4)
\]

Finally, for \( k = 1 \) again only the identity \( \tau_1 \) fulfills the condition in Line 9. We obtain the simplex \( S_2^1 \) as in (3) with \( v_0 = \vec{x}_0 + \frac{1}{2} (\vec{x}_2 - \vec{x}_0) \). The corner sub-simplex \( S_1^0 \) again corresponds to a translation of \( S_1^0 \) by the vector \( \frac{1}{2} (\vec{x}_2 - \vec{x}_0) \).

We now prove the \( \delta \)-preciseness of the refinement procedure and show how to model a refined triangulation by the generalized incremental method afterward. Let \( \mathcal{T}_k \) 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 \( \mathcal{T}_{i-1} \) are refined.
Lemma 3.3. Let $S \in \mathbb{R}^d$ be a simplex of $T_0$ and $e$ the longest edge of $S$. Then, the longest edge of any simplex of $\tilde{T}_l$ 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 $\tilde{T}_{\tilde{N}}$ is a refinement of every triangulation obtained by applying Algorithm 3.1 to $T_0$ with $\delta$ as input parameter.

Proof. Let $e_0$ be the longest edge of all simplices of $T_0$. With Lemma 3.3 and $\tilde{N} := \max \left\{ 0, \left\lceil \frac{\ln (\frac{||e_0||}{\delta})}{\ln(2)} \right\rceil \right\}$, 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 $\tilde{T}_{\tilde{N}}$ is bounded by $\delta$. Since Algorithm 3.1 only refines simplices with a longest edge larger than $\delta$ and no simplex in $\tilde{T}_{\tilde{N}}$ has an edge longer than $\delta$, it follows by the pigeonhole principle that $\tilde{T}_{\tilde{N}}$ is the finest refinement of $T_0$ that is achievable.

Theorem 3.5. Algorithm 3.1 as refinement procedure in Algorithm 1 [3] is $\delta$-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 $T_0$.

Proof. We count every single simplex that has to be refined to achieve $\tilde{T}_{\tilde{N}}$ 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 $(S^i) \in \tilde{T}_l$ of simplices has an element $S^k$
with index $k \leq m(2^{\tilde{N}d} - 1) + 1$, such that $S^k \in \tilde{T}_N$, since $\tilde{T}_N$ is a refinement of every triangulation obtained by Algorithm 3.1 with parameter $\delta$. Therefore, simplex $S^k$ has the $\delta$-preciseness property (1), as no simplex in $\tilde{T}_N$ has an edge longer than $\delta$. \hfill \Box

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

**Corollary 3.6.** Algorithm 1 [3] together with Algorithm 3.1 as refinement procedure is correct and terminates after a finite number of steps.

4. **Incremental Method for Redefined 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 $S = \{S^0, \ldots, S^{2^d-d-1}\}$ be a refinement of a simplex $S$ by Algorithm 3.1 with $V(S) = \{\bar{x}_0, \ldots, \bar{x}_d\}$. Then, each simplex of the subset of the corner sub-simplices $S' = \{S^0, \ldots, S^{d}\}$ of $S$ contains a vertex of the simplex $S$, i.e., $\bar{x}_j \in V(S')$ for all $j = 0, \ldots, d$. Moreover, for each pair of simplices $S^i, S^k \in S'$ 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 $\text{id} \in \text{Sym}_d$ always fulfills the conditions from Line 9 of Algorithm 3.1. Let $S^{ij}$ be the simplex that is constructed using the starting vertex $v_0 = \frac{1}{2}(\bar{x}_0 + \bar{x}_j)$ and $\tau = \text{id}$, where $j = 0, \ldots, d$. Due to the telescoping sum in Line 11, it follows that

$$v_j = \frac{1}{2}(\bar{x}_0 + \bar{x}_j) + \frac{1}{2}(\bar{x}_1 - \bar{x}_0) + \frac{1}{2}(\bar{x}_2 - \bar{x}_1) + \cdots + \frac{1}{2}(\bar{x}_j - \bar{x}_{j-1}) = x_j$$

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

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 $\tau = \text{id}$. 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 $S^{ij}$ and $S^{ik}$, respectively, are equal to $m_{jk}$. \hfill \Box

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

$$S = \{x \in \mathbb{R}^d : a_j^\top x \leq b_j \text{ with } a_j \in \mathbb{R}^d \text{ and } b_j \in \mathbb{R} \text{ for } j = 0, \ldots, d\}. \quad (6)$$

We now describe the set $S''$ 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 \( S_1 \) 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 disjoint, we can describe \( S_1 \) 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

\[
S' = \left\{ x \in \mathbb{R}^d : a_j^\top x \leq b_j, \ a_j^\top x \geq \frac{1}{2}(b_j + a_j^\top \bar{x}_j) \right\}.
\]

It follows that the union of all simplices of \( S' \) is a convex polytope.

The consistency of the triangulation of \( S \) translates to the triangulation of \( S' \), because only the corner sub-simplices are omitted. It is shown by [5] that a triangulation with the property that for each nonempty subset \( \hat{S} \subseteq \mathcal{T} \) of a triangulation \( \mathcal{T} \) there exist simplices \( S_1 \in \hat{S} \) and \( S_2 \in \mathcal{T} \setminus \hat{S} \) 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 \( S' \) has this property. To this end, we consider a nonempty subset \( \hat{S} \) of \( S' \). Each facet of a \( d \)-dimensional simplex consists of \( d \) vertices of the simplex. Let \( \hat{S}_F \) be the set of all facets of the simplices of \( \hat{S} \), where the simplices are described by the convex hull of its vertices. Then, each facet in \( \hat{S}_F \) is either a facet of \( S' \) (in the sense of convex hulls) or a common facet of two simplices of \( \hat{S} \).

This is due to the consistency of the triangulation. Since \( \hat{S} \subseteq S' \), however, there must be a facet in \( \hat{S}_F \) that is a common facet of two simplices \( S_1, S_2 \) such that \( S_1 \in \hat{S} \) and \( S_2 \in S' \setminus \hat{S} \). \( \square \)

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 \( S' \) and \( S'' \) 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 \( \mathcal{T} = \{S_1, \ldots, S_n\} \) and \( S_l \) be the simplex that has to be refined, while \( \bar{x}_0^{S_l}, \ldots, \bar{x}_d^{S_l} \) are its labeled vertices. We first consider \( d \geq 4 \) and \( d = 2, 3 \) afterward. The \( 2^d \) simplices \( S_0^{S_l}, \ldots, S_{2^d-1}^{S_l} \), 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 \( \bar{x}_0^{S_l} \in V(S_l^1) \).

We first show that the set \( S' \) of the corner sub-simplices has an ordering with the required properties. The corner sub-simplices \( S_l^1 \in S' \) 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 $\bar{x}^l_j = m_{jk}$ and $\bar{x}^l_k = m_{jk}$.

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

$$(S^l_0, \ldots, S^l_{d-1-(d+1)})$$

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

$$m_{jk} = \bar{x}^l_0 \quad \text{and} \quad m_{st} = \bar{x}^l_d.$$

We now link the orderings of $S'$ and $S''$ 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 $j \neq s \land j \neq t$. With $s \neq t$, we have the following three cases for the node $s$: $s = j$, $s = k$, or $s \neq j \land s \neq k$. Consequently, due to $j \neq k$ and $s \neq t$ only the following five cases are possible for the nodes $j, k, s,$ and $t$:

$$j = s \land k \neq t, \quad j = s \land k = t, \quad j \neq s \land k = t, \quad k = s, \quad j \neq s \land k \neq t. \quad (9)$$

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

$$H = \begin{cases} 
(0, j, t, R, k, d), & \text{if } j = s \land k \neq t, \quad (10a) \\
(0, R, j, k, d), & \text{if } j = s \land k = t, \quad (10b) \\
(0, j, s, R, t, d), & \text{if } k = s, \quad (10c) \\
(0, j, s, R, k, t, d), & \text{otherwise.} \quad (10d) 
\end{cases}$$

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 $S^l_i$, where $i = 0, \ldots, d - 1$ and of the simplex $S^l_{d-1-(d+1)}$, where $i = 1, \ldots, d$, is permissible. Thus, we can assume that $m_{st} \neq m_{0d}$ and that $m_{jk} \neq m_{0a}$ if $m_{st} = m_{ad}$. 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 $S^l_j \in S'$. As showed above, this ordering has the properties (O1) and (O2). We now insert the ordering (8) of the simplex set $S''$ into the one of $H$ after the
simplex that corresponds to the vertex $j$. The union of $S'$ and $S''$ 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 $S'$ and $S''$ via the midpoints $m_{jk}$ and $m_{st}$ finally leads to an overall ordering $(S_l^0, \ldots, S_l^{2^d-1})$ 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 $S'$ and $S_l^3$–$S_l^6$ the four center sub-simplex of $S''$. Finally, the orderings of $S'$ and $S''$ 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 $T'$ as

$$(S_1, \ldots, S_{l-1}, S_l^0, \ldots, S_l^{2^d-1}, S_{l+1}, \ldots, S_n).$$

(11)

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

$$\bar{x}_d^{S_l_{-1}} = \bar{x}_0^{S_l^0} \quad \text{and} \quad \bar{x}_d^{S_l^{2^d-1}} = \bar{x}_0^{S_{l+1}}$$

holds. Therefore, the simplices $S_{l-1}$, $S_l^0$ and $S_l^{2^d-1}$, $S_{l+1}$ are linked as required in (O2). Altogether, due to the inheritance from $T$, we conclude that the ordering (11) of the simplices of $T'$ has the properties (O1) and (O2).

5. 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 for the thorough reading of the paper by Lena Hupp.

References


---

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

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