HOCHSCHULE
KONSTANZ
TECHNIK, WIRTSCHAFT
UND GESTALTUNG

# Anwendung der Linearen Optimierung


## - Sensibilitätsanalyse -


| | |
|---|---|
| Vorgelegt am: | 29.06.2016, Konstanz |
| Eingereicht von: | Sven Reisenhauer (Matrikel-Nr.: 286522) |
| | Can Satilmis (Matrikel-Nr.: 288905) |
| Betreuer: | Prof. Dr. Grütz |

# Inhaltsverzeichnis

# 1 Einleitung

Im Rahmen der stattgefunden Veranstaltung „*Anwendung der linearen Optimierung*“ an der HTWG Konstanz, gilt es eine Projektdokumentation zu schreiben, die anschließend nach Abgabe als Prüfungsleistung gewertet wird. Wir haben das uns zugeteilte Tool SensiOR 1.0 weiterentwickelt und auf SensiOR 1.1 versioniert.

Diese Dokumentation geht auf den Ist-Zustand und alle Erweiterungen bzw. Veränderungen zur aktuellen Version SensiOR 1.1 ein. Außerdem beschreiben wir die Vorgehensweise und Umsetzung, aber auch die Probleme, welche im Verlauf des Projektes aufgetreten sind.

# 2 Projekt (Nr. 4)

## 2.1 Projektbeschreibung

Bei dieser Anwendung SensiOR kann mit Hilfe der Sensibilitätsanalyse die Schwankungsbereiche des c- und b-Vektors berechnen werden. Bei der Sensibilitätsanalyse fehlt eine notwendige Hilfefunktion. Nach der Berechnung eines LP Problems können die eingegebenen Werte nicht mehr verändert werden, da direkt das Ergebnis angezeigt wird.

## 2.2 Projektziel

Aus der Beschreibung ergibt sich folgende Aufgabenstellung:

- Eine Hilfefunktion soll implementiert werden
- Nach dem Lösen eines LP Problems soll der Schritt zurück ermöglicht werden, um eingegebene Daten wieder anpassen zu können

## 2.3 Rahmenbedingungen des Projekts

- Das Tool wird in C# weiterentwickelt
- Entwicklungsumgebung ist Visual Studio 2015

### 2.3.1 Projektentwickler

| **Name** | **Matrikelnummer** | **Studiengang** |
|---|---|---|
| Can Satilmis | 288905 | Wirtschaftsinformatik B.Sc. |
| Sven Reisenhauer | 286522 | Wirtschaftsinformatik B.Sc. |

# 3 Funktionsumfang des Tools SensiOR

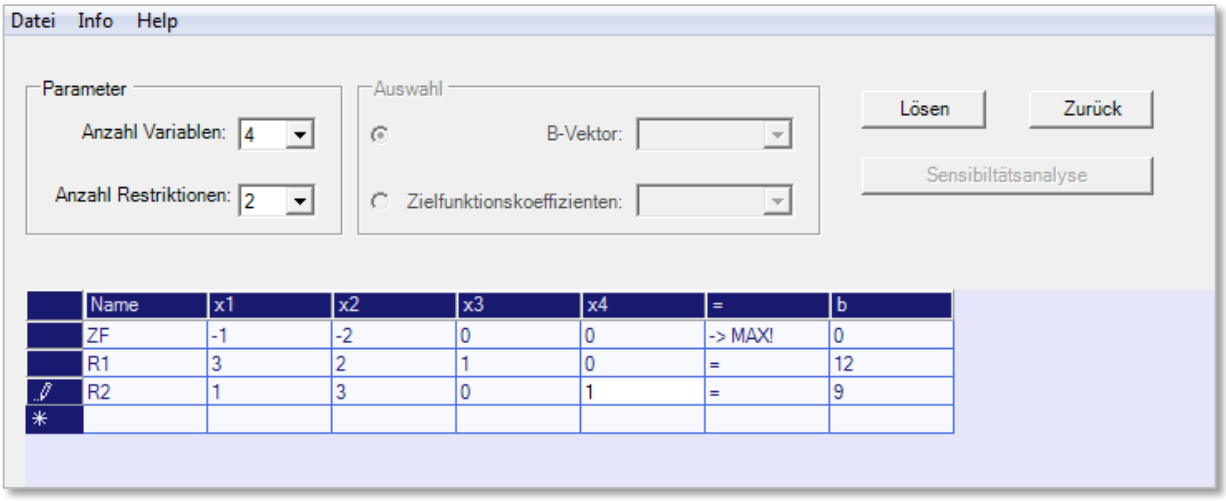
Mit Hilfe dieses Tools werden auf Gleichheit normierte LP Ansätze mittels Einführung von Schlupfvariablen gelöst. Da der Optimierungsalgorithmus auf dem des Iterators basiert, müssen dabei die Zielfunktionskoeffizienten mit negativen Vorzeichen eingegeben werden.

Verdeutlicht wird das durch folgenden LP Ansatz:

$1x_1 + 2x_2 \rightarrow$ max!

$3x_1 + 2x_2 \leq 12$

$1x_1 + 3x_2 \leq 9$



Durch Einführung von Schlupfvariablen und der Multiplikation der Zielfunktion mit -1 wird das Problem in SensiOR gelöst.

Das Tool ermittelt nun mit dem Klick auf den Button *„Lösen“* das Optimaltableau. Im Anschluss kann die Sensibilitätsanalyse durch die Wahl des entsprechenden c – bzw. b-Indexwert mit einem Klick auf *„Sensibilitätsanalyse“* ausgeführt werden.

Bisher ist es nur möglich, Restriktionen mit kleiner-gleich-Relationen und einer Hilfsvariable pro Restriktion zu lösen.

LP Ansätze welche

- Ist- gleich Relationen
- Größer-gleich Relationen
- Näherungsweise Lösungen
- Nichtlineare Probleme
- Minimierungsprobleme

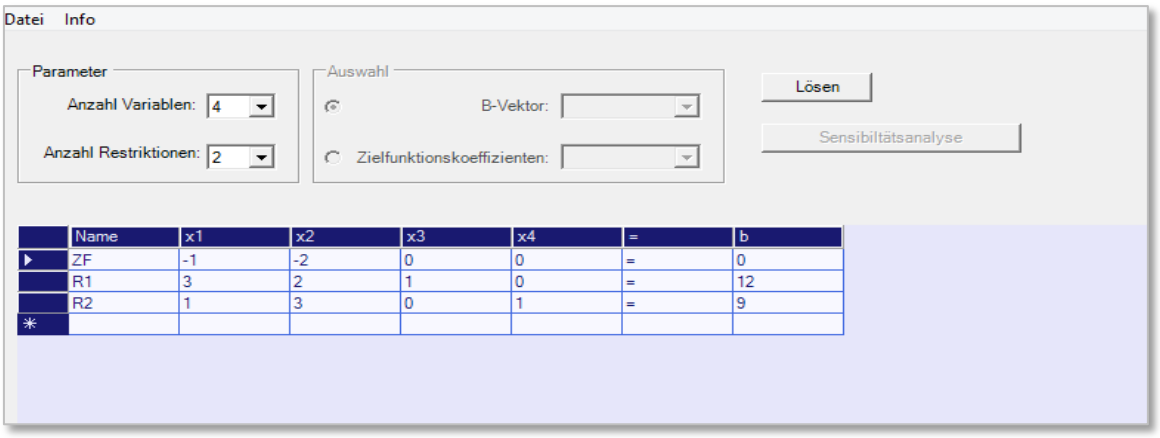
enthalten, können bisher mit dem Tool nicht gelöst werden, da der Solver nicht in der Lage ist, ein Optimaltableau dafür zu ermitteln.

# 4 Ist-Zustand

## 4.1 Zurück-Funktion

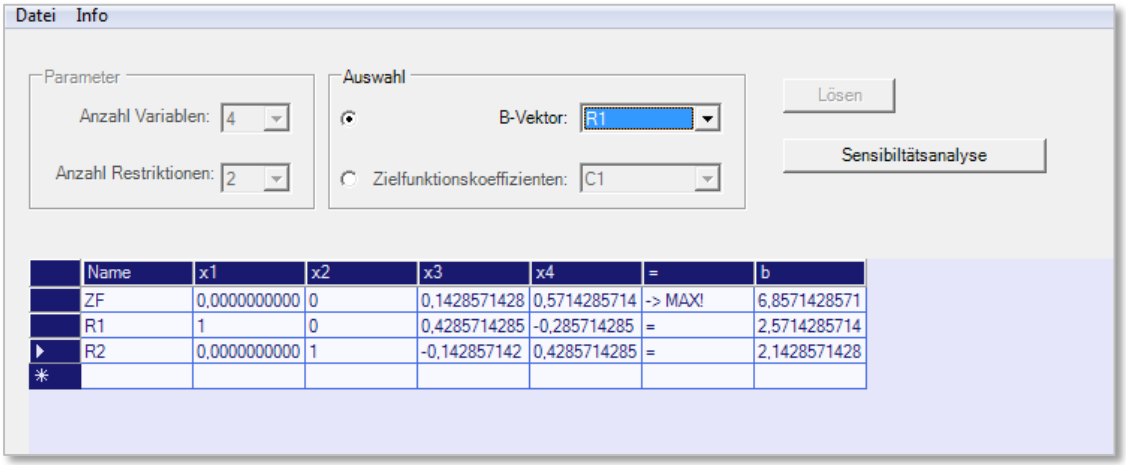
In der Version 1.0 fehlt diese wichtige Funktionalität. Das macht die Nutzung der Anwendung sehr mühsam. Was das für die Handhabung des Tools bedeutet, werden wir in diesem Kapitel grafisch darstellen.

Wurden die Parameter nun eingegeben, kann man sich das Optimaltableau mit *„Lösen“* berechnen lassen



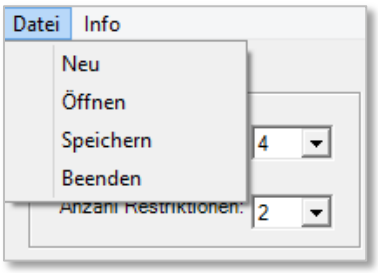

| | Name | x1 | x2 | x3 | x4 | = | b |
|---|---|---|---|---|---|---|---|
| ▶ | ZF | -1 | -2 | 0 | 0 | = | 0 |
| | R1 | 3 | 2 | 1 | 0 | = | 12 |
| | R2 | 1 | 3 | 0 | 1 | = | 9 |
| ✳ | | | | | | | |

Nach Klicken auf den Button *„Lösen“* bekommen wir das Ergebnis ausgeworfen. Anschließend kann auf „*Sensibilitätsanalyse*“ geklickt werden, um das fertige Endergebnis anzuzeigen. Problematisch wird es allerdings, wenn der Nutzer einen Wert nachträglich verändern möchte. Es ist nicht möglich, zurück zum Ausgangstableau zu springen und einen Wert zu verändern.




| | Name | x1 | x2 | x3 | x4 | = | b |
|---|---|---|---|---|---|---|---|
| | ZF | 0,0000000000 | 0 | 0,1428571428 | 0,5714285714 | -> MAX! | 6,8571428571 |
| | R1 | 1 | 0 | 0,4285714285 | -0,285714285 | = | 2,5714285714 |
| ▶ | R2 | 0,0000000000 | 1 | -0,142857142 | 0,4285714285 | = | 2,1428571428 |
| ✳ | | | | | | | |

Stattdessen musste der Nutzer ein neues leeres Sheet öffnen und seine Daten zum Tableau neu eingeben.



### 4.2 Hilfefunktion

Ein weiteres Problem bestand darin, dass keine Hilfefunktion implementiert war. Das bedeutet, dass der Nutzer keinerlei Hilfe oder Informationen zur richtigen Anwendung des Tools erhält. Mühsam muss erahnt werden, welche Art von LP-Ansätzen das Tool überhaupt verarbeiten kann und wie die richtige Eingabesyntax aussieht.

# 5 Projektumsetzung

## 5.1 Umsetzung im Allgemeinen

Die Funktionserweiterung, die das Tool erhalten sollte, war klar definiert, dennoch mussten wir uns zu Projektierungsbeginn Gedanken machen, mit welcher Logik das umgesetzt werden soll.

Zudem bestand die Schwierigkeit darin, das Tool in C#, einer für uns sehr fremden Programmiersprache, zu entwickeln. Doch war das auch die Chance, unser Kenntnisgebiet zu erweitern.

## 5.2 Technische Umsetzung

### 5.2.1 Zurück-Funktion

Technisch haben wir uns darauf geeinigt, die Daten bei Klick auf den „*Lösen*“-Button zwischen zu speichern. Zwischengespeichert werden in diesem Fall nicht nur die Werte, welche der Nutzer zuvor eingegeben hat, auch die Anzahl der Reihen und Spalten mussten explizit gespeichert werden. Die nachfolgende Abbildung zeigt den Code mit dem dies realisiert wurde.

```
499         const int DATATABLE_COLUM_COUNT = 75;
500         const int DATATABLE_ROW_COUNT = 75;
501         int m_nDataTableColums;
502         int m_nDataTableRows;
503         String[] m_strzDataTable_Matrix = new String[DATATABLE_COLUM_COUNT * DATATABLE_ROW_COUNT];
504         String[] m_strzDataTable_BValue = new String[DATATABLE_ROW_COUNT];
505
506         private void clear_datatable_data()
507         {
508             for( int i = 0; i < DATATABLE_ROW_COUNT; i ++)
509             {
510                 for( int j = 0; j < DATATABLE_COLUM_COUNT; j ++ )
511                     m_strzDataTable_Matrix[i * DATATABLE_COLUM_COUNT + j] = "0";
512                 m_strzDataTable_BValue[i] = "0";
513             }
514
515             m_nDataTableColums = 0;
516             m_nDataTableRows = 0;
517         }
518
519         private void get_datatable_data()
520         {
521             m_nDataTableColums = theDataTable.Columns.Count - 3;
522             m_nDataTableRows = theDataTable.Rows.Count;
523
524             for (int i = 0; i < m_nDataTableRows; i++)
525             {
526                 for (int j = 0; j < m_nDataTableColums; j++)
527                     m_strzDataTable_Matrix[i * DATATABLE_COLUM_COUNT + j] = theDataTable.Rows[i][j + 1].ToString();
528                 m_strzDataTable_BValue[i] = theDataTable.Rows[i][theDataTable.Columns.Count - 1].ToString();
529             }
530         }
531
532         private void set_datatable_data()
533         {
534             for (int i = 0; i < m_nDataTableRows; i++)
535             {
536                 for (int j = 0; j < m_nDataTableColums; j++)
537                     theDataTable.Rows[i][j + 1] = m_strzDataTable_Matrix[i * DATATABLE_COLUM_COUNT + j];
538                 theDataTable.Rows[i][theDataTable.Columns.Count - 1] = m_strzDataTable_BValue[i];
539             }
```

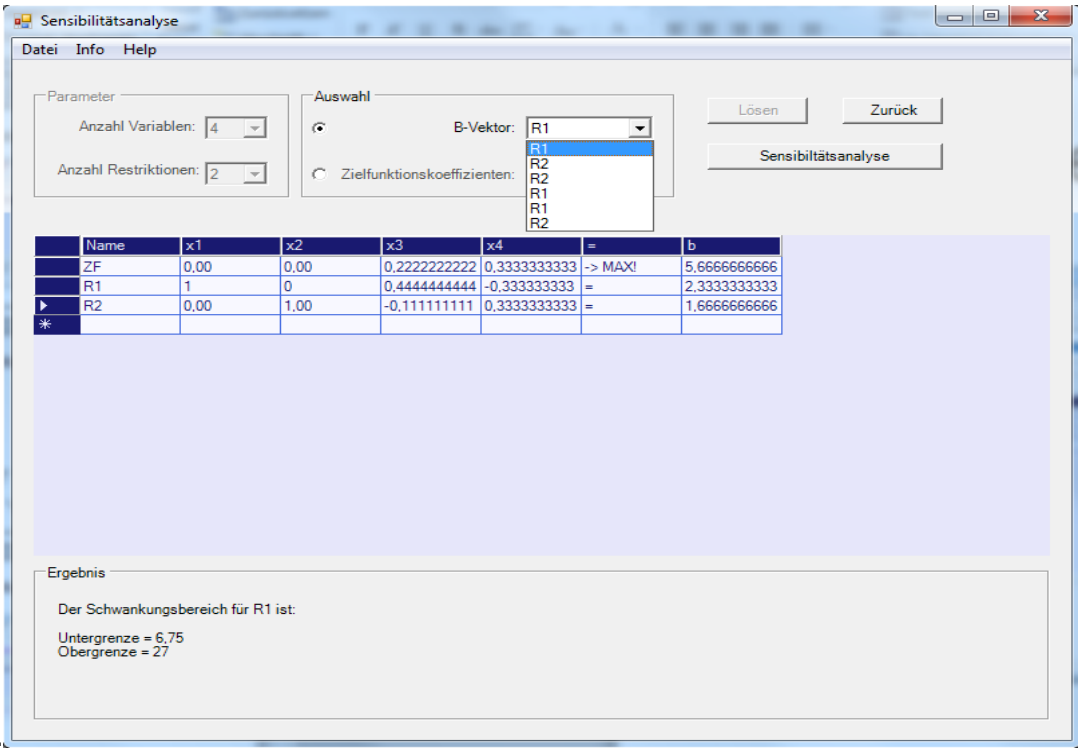
Des Weiteren haben wir festgelegt, was bei Klick auf „*Lösen*“ passiert:

```
    //Button Sensibilitätsanalyse enable macher
    b_analyse.Enabled=true;

    //Groupbox enable machen
    gb_auswahl.Enabled=true;

    //Parameter disablen
    gb_parameter.Enabled=false;

    //Lösenbutton disablen
    b_solve.Enabled =  false;

    b_back.Enabled = true;

    //Nach dem Lösen das Grid sperren
    inputtable.Enabled= false;
}
```

Wird auf „*Lösen*“ geklickt, wird der Button „Sensibilitätsanalyse“ aktiviert, um eine anschließende Sensibilitätsanalyse zu ermöglichen. Die „*Groupbox*“, in welcher ausgewählt werden kann, welcher Indexwert in der Sensibilitätsanalyse berechnet werden soll, wird aktiviert. Zeitgleich wird die „*Groupbox“* der Parameter deaktiviert.

Außerdem wurde der „*Lösen*“-Button deaktiviert und die „*Grid*“, in der die Werte vom Nutzer eingegeben werden können, gesperrt.

Zunächst hatten wir das Problem, dass die Anzahl der anzuwählenden Optionen im Dropdownmenü stieg, wenn Werte nachträglich verändert wurden und erneut auf „*Lösen*“ geklickt wurde.




Dies konnten wir jedoch mit folgendem Befehl umgehen.

```
691                 cb_auswahl_b.Items.Clear();
692                 cb_auswahl_zf.Items.Clear();
```

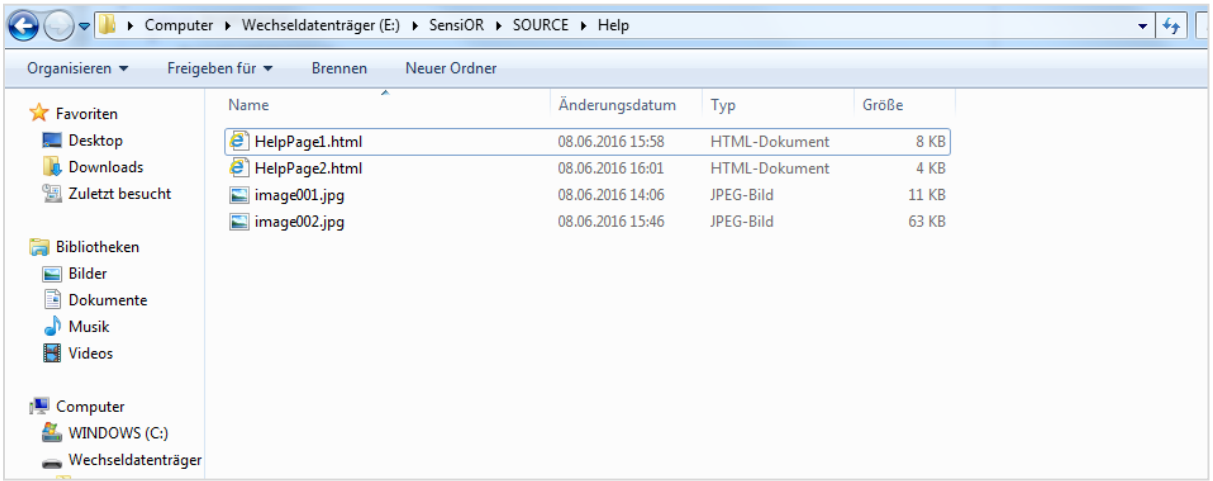
Außerdem musste der grafisch sichtbare „*Zurück*“-Button erstellt werden.

```
402             // b_back
403             //
404             this.b_back.Location = new System.Drawing.Point(621, 32);
405             this.b_back.Name = "b_back";
406             this.b_back.Size = new System.Drawing.Size(75, 23);
407             this.b_back.TabIndex = 17;
408             this.b_back.Text = "&Zurück";
409             this.b_back.Click += new System.EventHandler(this.b_back_Click);
```

### 5.2.2 Hilfefunktion

Für die Hilfefunktion erstellten wir zunächst ein Word-Dokument, welches dann als HTML-Datei exportiert wurde. Diese HTML-Dateien und dazugehörige Grafiken wurden anschließend im Space SensiOR eingebunden.

```
namespace SensiOR
{
    public partial class HandbuchWindow : Form
    {
        public HandbuchWindow()
        {
            InitializeComponent();


            string curDir = System.IO.Directory.GetCurrentDirectory();
            this.m_webBrowser.Url = new Uri(String.Format("file:///{0}/Handbuch/Handbuch.htm", curDir));
        }

        private void m_webBrowser_DocumentCompleted(object sender, WebBrowserDocumentCompletedEventArgs e)
        {
            string curDir = System.IO.Directory.GetCurrentDirectory();
            System.Windows.Forms.HtmlDocument document = this.m_webBrowser.Document;
            String strText = this.m_webBrowser.DocumentText;
            if (document != null && document.All["image001"] != null)
            {
                document.All["image001"].SetAttribute("src", String.Format("file:///{0}/Handbuch/Handbuch-Dateien/image001.png", curDir));
            }
            if (document != null && document.All["image002"] != null)
            {
                document.All["image002"].SetAttribute("src", String.Format("file:///{0}/Handbuch/Handbuch-Dateien/image002.png", curDir));
            }
            if (document != null && document.All["image003"] != null)
            {
                document.All["image003"].SetAttribute("src", String.Format("file:///{0}/Handbuch/Handbuch-Dateien/image003.png", curDir));
            }
            if (document != null && document.All["image004"] != null)
            {
                document.All["image004"].SetAttribute("src", String.Format("file:///{0}/Handbuch/Handbuch-Dateien/image004.png", curDir));
            }
            if (document != null && document.All["image005"] != null)
            {
                document.All["image005"].SetAttribute("src", String.Format("file:///{0}/Handbuch/Handbuch-Dateien/image005.png", curDir));
            }
            if (document != null && document.All["image006"] != null)
            {
                document.All["image006"].SetAttribute("src", String.Format("file:///{0}/Handbuch/Handbuch-Dateien/image006.png", curDir));
            }
            if (document != null && document.All["image007"] != null)
            {
                document.All["image007"].SetAttribute("src", String.Format("file:///{0}/Handbuch/Handbuch-Dateien/image007.png", curDir));
            }

        }
    }
}
```

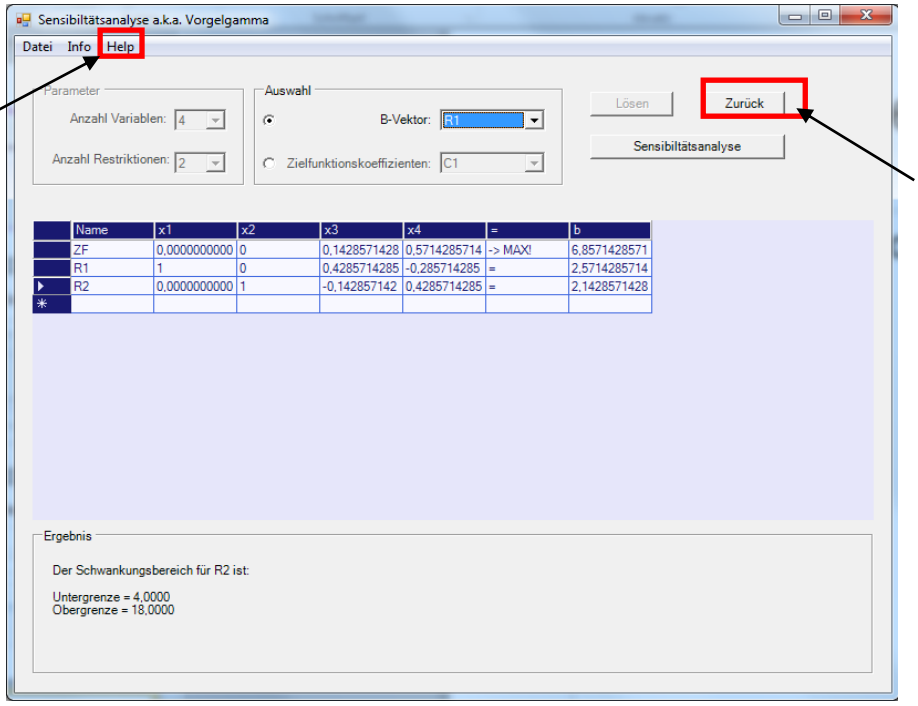



Die obige Abbildung zeigt die Ordnerstruktur der HTML- und Grafikdokumente.

Im Anschluss haben wir die Versionsnummer und Infos aktualisiert.

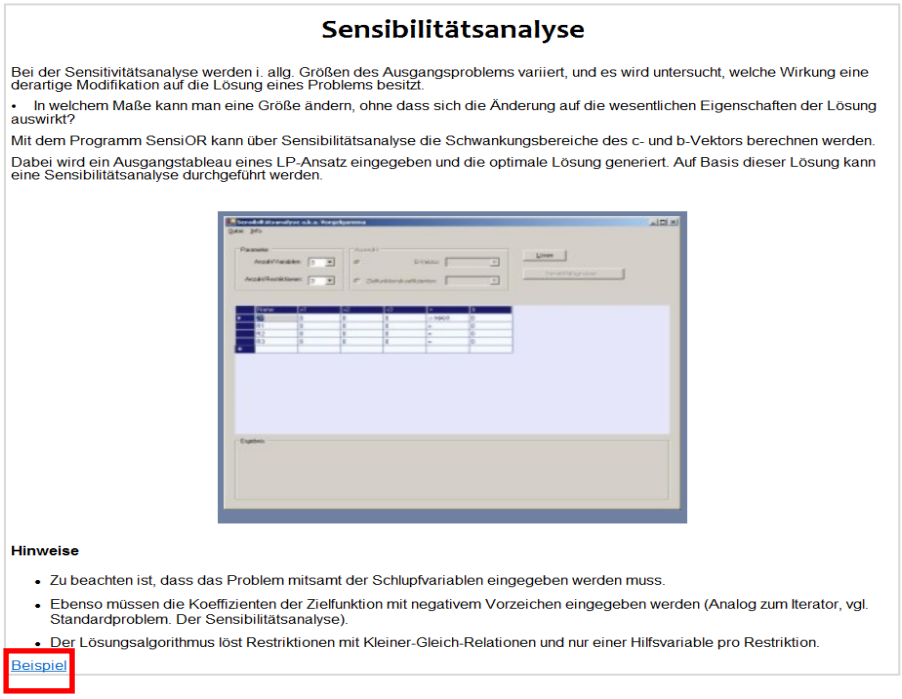
# 6 Ergebnisse

Hilfefunktion implementiert




Ein „zurück“ Button wurde implementiert.

Mit Klick auf diesen werden die zuvor eingegebenen Zahlen wieder ausgegeben und können verändert werden




Es wurde ein Beispiel als Link in die Hilfefunktion eingefügt, welches eine Beispielaufgabe im gleichen Fenster aufruft.

# 7 Verbesserungssvorschläge

## 7.1 Allgemeine Vorschläge

Aktuell ist die Anwendung eine MFC-Applikation, die inzwischen weitestgehend von WPF abgelöst wurde. Das Problem dieser Applikation ist, dass sie nur auf Windows basierten Systemen läuft, da ein .NET Framework benötigt wird.

Aus diesem Grund wäre eine Neuentwicklung in Java vorteilhaft, um plattformunabhängig zu sein.

Aus diesem Grund schlagen wir für das nächste Entwicklungsteam eine Neuentwicklung in Java vor, um die Anwendung plattformunabhängig zu machen.

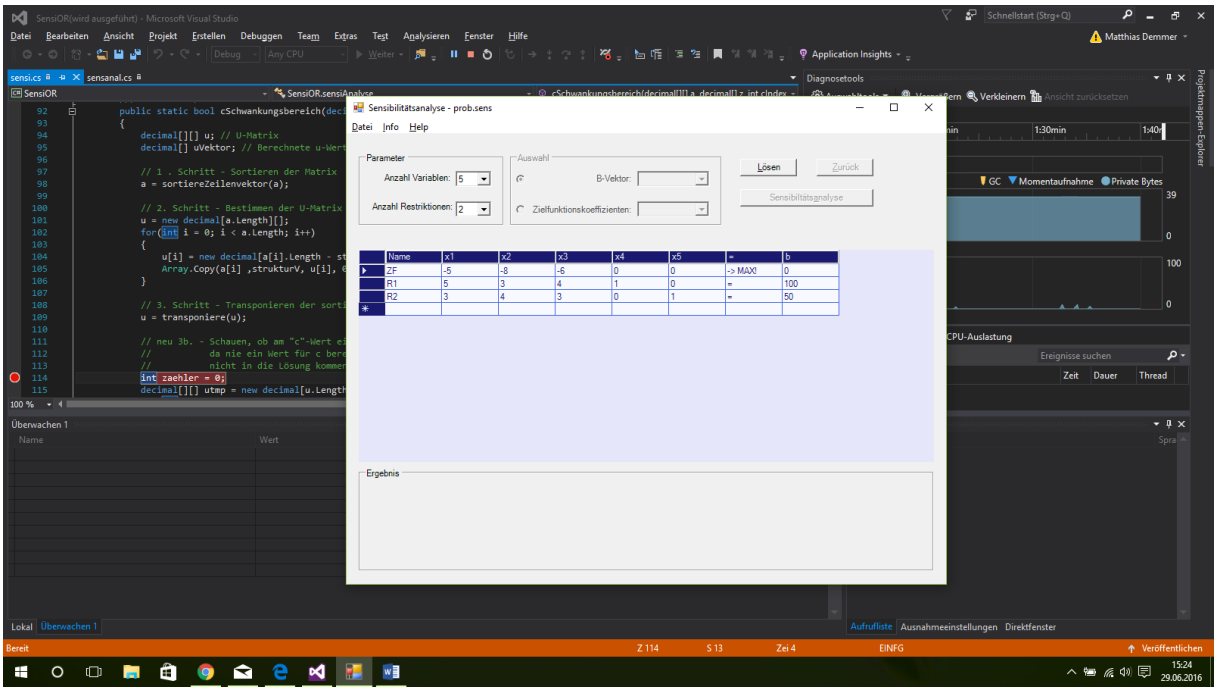
Weitere Vorteile sind:

- Studienübergreifende Programmiersprache
- Bessere Implementierung als Web-Anwendung

## 7.2 Weitere Vorschläge und deren Ansätze

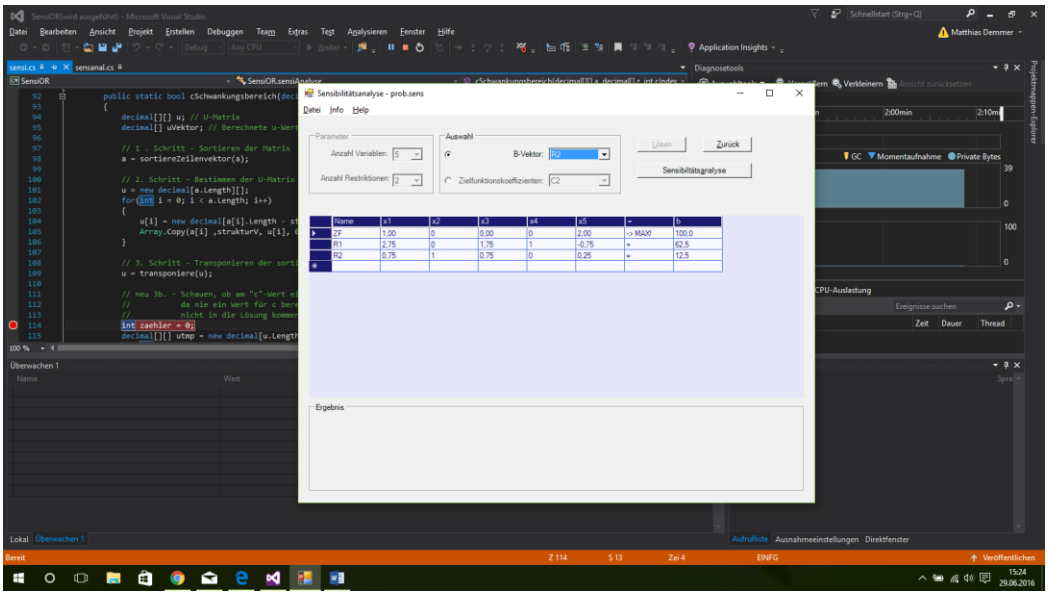
Im Laufe der Umsetzung des Projektes fanden wir einen weiteren Bug, welchen wir über die angeforderte Zielsetzung hinaus versucht haben zu beseitigen. Das Problem besteht darin, dass die Ober- und Untergrenzen bei manchen LP-Ansätzen falsch ausgegeben werden. Zunächst wurde vermutet, dass LP-Ansätze querbeet betroffen waren. Nach einigen Versuchen zeigte sich jedoch, dass das Problem alle LP-Ansätze mit Variablenanzahl $\geq 3$ betrifft.

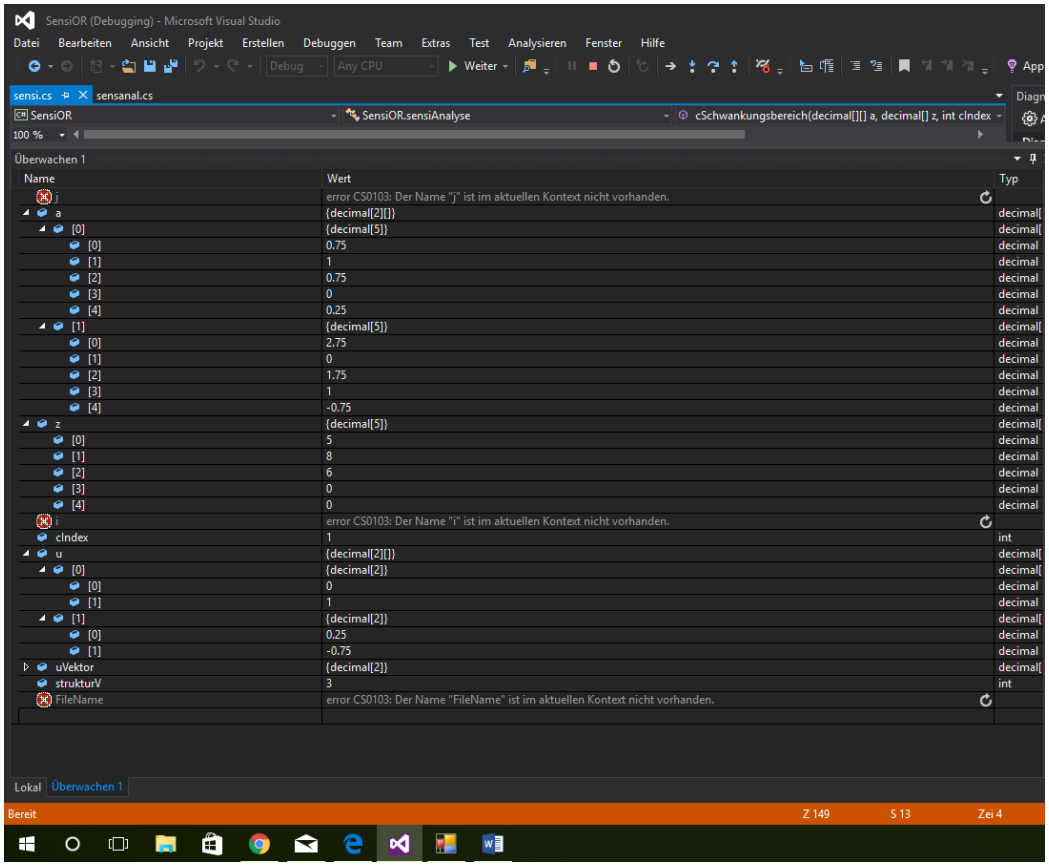
Für folgendes LP-Modell haben wir ein Beispiel gestartet, um die Vorgehensweise des Codes zu verdeutlichen:




Das Ergebnis wurde berechnet. Bis hier ist alles noch so wie es sein soll.

Das Ergebnis ist für x2 = 12,5






Zu sehen sind die Werte der u-Matrix am Ende der Berechnung ( 0 , 1 ; 0.25 , -0.75 ).

a sind die Restriktionen

z sind die zielfunktionswerte.

Der u-Vektor ist dann das Ergebnis (Untergrenze 0, Obergrenze 1,66667).

Das wird so berechnet:

| | | | | |
|---|---|---|---|---|
| u-Vektor[0] = | 0 * 5 | = | 0 / 1 * -1 | = 0 |
| u-Vektor[1] = | 0,25 * 5 | = | 1,25 / -0,75 * -1 | = 1,66667 |

Für die kommenden Semester haben wir außerdem die wichtigen Stellen des Codes zusammengefasst und beschrieben, um das Lösen dieser Problemstellung zu vereinfachen.
Das Problem konnten wir leider nicht beheben, aber wir sind überzeugt, dass wir auf der richtigen Spur sind.
Die Vorgehensweise des Codes ist entschlüsselt und auf den vorherigen Seiten abgebildet bzw. auf nachfolgenden Seiten nochmal auf Reverse Engineering Ebene dargestellt.
Der Code wurde von uns klar strukturiert und kommentiert und ist auf folgendem Link zu finden:

http://pastebin.com/hPZepwWC

| | |
|---|---|
| Zeile 4 – 10: | Beschreibungen der Parameter |
| Zeile 17: | Sortieren |
| Zeile 20 – 28: | u-Matrix Bestimmung |
| Zeile 30 – 44: | Löschen von Zeilen mit c-Wert 0 |
| Zeile 47 – 65: | u-Vektor berechnen (Multiplikation mit zf-Wert) |
| Zeile 67 – 87: | Grenzen suchen und ausgeben |
| Rest: | Bestimmung der Grenzen (gehört zu Zeile 67 – 87) |

| | |
|---|---|
| a = | Restriktionen (die berechneten Werte aus der Lösung) |
| z = | Zielfunktion (z.B. 5,6,8,0,0) also die Werte die eingetippt wurden |
| c-Index = | zu Berechnender c-Wert |
| strukturV = | Variablen – Schlupfvariablen (z.B. beim Problemfall 3) |

# 8 Fazit

Als Studenten der Wirtschaftsinformatik gewannen wir aufgrund des Projekts, einen Einblick in die Entwicklung einer Erweiterung einer bereits verwendeten Software und bekamen die Möglichkeit, die damit verbundenen Aufgaben selbständig zu lösen. Durch das Wissen, dass wir an Tools arbeiten, welche auf professionellem Niveau genutzt werden, waren wir auch sehr motiviert und haben das Projekt sehr ernst genommen.

Doch das Umsetzen unserer Ziele, bereitete zu Beginn einige Probleme. Aufgrund dessen, da wir beide zuvor noch nie mit der Programmiersprache C# gearbeitet haben, erforderte es eine gewisse Einarbeitungszeit. Durch das Programmieren, mit einer für uns unbekannten Programmiersprache, konnten wir allerdings wertvolle Erfahrungen sammeln und zeitgleich unsere persönlichen Soft-Skills wie beispielsweise Teamwork weiter ausbauen.

Alle anfallenden Aufgaben konnten gelöst werden und darüber hinaus haben wir weitere alte Bugs entfernt. Des Weiteren standen wir kurz vor dem Durchbruch bei der Findung des bereits bekannten Problems, dass manche LP-Ansätze falsche Ergebnisse ausliefern. Unser Wissen wurde in diesem Dokument für das nächste Entwicklungsteam festgehalten und kann als vorbereitende Dokumentation genutzt werden.