

HTWG Konstanz

Innerbetriebliche Standortplanung 1.1

Anwendung der linearen Optimierung

Jenny Schwarz 288236

Sarah Amann 288182

30.06.2015

Inhaltsverzeichnis

Einleitung	3
Aufgabenstellung	4
Aufgabenstellungen	4
Lauffähigkeit	4
Eingabe eines Solverpfades	5
Hilfefunktion	5
Sonstiges	6
Rahmenbedingungen	6
Tätigkeiten	7
Lauffähigkeit	7
Hilfefunktion	10
Interne Berechnung	12
Handlungsempfehlung	13
Abbildungsverzeichnis	14

Einleitung

Im Rahmen der Veranstaltung „Anwendung der linearen Optimierung“ bei Herrn Prof. Dr. Grütz müssen in Zweierteams Methoden der Methodenbank angepasst und optimiert werden. Zwei Studierende müssen dabei bestehende Methoden den Anforderungen entsprechend anpassen und eventuell um weitere Funktionen erweitern.

Aufgabenstellung

Im Sommersemester 2015 bestand ein Projekt daraus die Methode „Innerbetriebliche Standortplanung 1.0“ zu optimieren. Dabei galt es die Methode zu verbessern und den Anforderungen entsprechend anzupassen. Wir haben vor Beginn des Projektes das Pflichtenheft und die mit dem Projekt verbundenen Aufgaben und Anforderungen mit Professor Grütz und Serkan Önnisan abgesprochen.

Aufgabenstellungen

Im Folgenden werden die entstandenen Aufgabenstellungen erläutert.

Lauffähigkeit

Zu Beginn des Projektes konnte kein LP-Ansatz gelöst werden. Die Methode erlaubte eine Eingabe, jedoch gab die Methode keine Lösung aus. Als oberste Priorität des Projektes haben wir daher die **Lauffähigkeit** der Methode gesetzt.

The screenshot shows a software window titled "Innerbetriebliche Standortplanung". It has a menu bar with "Datei", "Bearbeiten", "Berechnungsart", and "Hilfe". Below the menu is a tab labeled "Eingabe" with a dropdown menu showing "Ergebnis". The main area is divided into two sections: "Zuordnungsmatrix" and "Zuordnungsergebnis".

Zuordnungsmatrix: A table with 7 rows (A-B, A-C, A-D, B-C, B-D, C-D) and 6 columns (1-2 2-1, 1-3 3-1, 1-4 4-1, 2-3 3-2, 2-4 4-2, 3-4 4-3). The cell for A-B in the first column contains the value 24.

Zuordnungsergebnis: A table with 2 rows (Standort, Maschine) and 4 columns (A, B, C, D). The cell for Maschine in the first column (A) contains the value 1.

At the bottom, a status bar shows "Dauer: 0,064296 sec" and "Optimales Ergebnis mit einem TAW von 249".

Abbildung 1: Ergebnisausgabe

Eingabe eines Solverpfades

Als zweiter Schritt sollte die Eingabe eines **Solverpfades** ermöglicht werden. Diese Anpassung sollte unter „Einstellungen“ erfolgen.

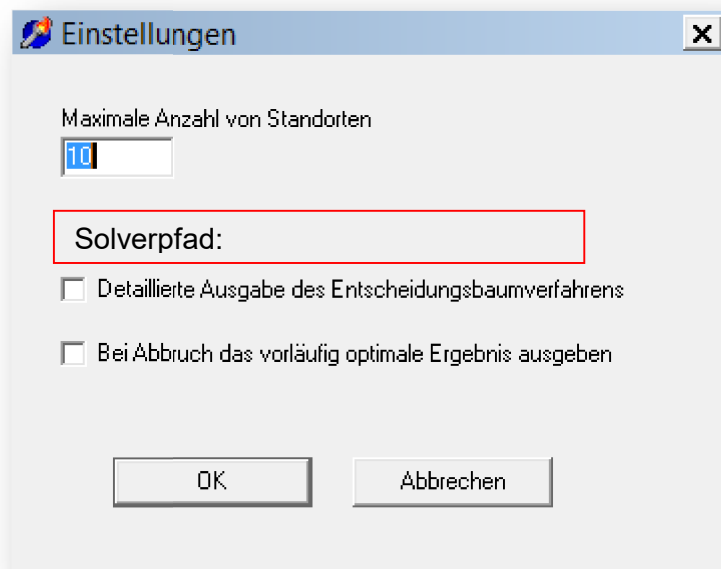


Abbildung 2: Einstellungen mit optionaler Solverpfadeingabe

Nach dem ersten Einblick in die Methode haben wir erkannt, dass die Methode die Berechnung intern implementiert hat und keinen externen Solver eingebunden hat. Aus diesem Grund haben wir diese Anforderung verworfen und uns auf die anderen Aufgaben konzentriert.

Hilfefunktion

Des Weiteren bestand die Anforderung nach einer **Hilfefunktion** innerhalb der Methode. Aus diesem Grund sollte neben der Funktion „Über...“ unter dem Reiter „Hilfe“ ein neues Item „Hilfe“ eingebunden werden. Diese Hilfefunktion soll Unterstützung zur Anwendung der Methode und Interpretation der Lösung geben.

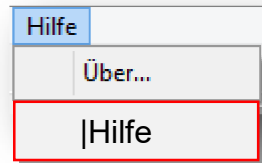


Abbildung 3: Überfunktion und neue Hilfefunktion

Sonstiges

Da die Aufgabe zur Lauffähigkeit des Programmes an oberster Stelle stand, haben wir zusätzlich dazu die Lauffähigkeit auf **Windows 7 und 8** als Aufgabe definiert.

Die Änderungen am bestehenden Code wurden mit **Kommentaren** versehen und gekennzeichnet.

Rahmenbedingungen

Durch die Anpassungen und Änderungen am bestehenden Code sollen alle nicht betroffenen Funktionalitäten nicht beeinträchtigt werden.

Als große Herausforderung im Projekt haben wir von Anfang an die Programmiersprache des Programms **Pascal** gesehen. Da wir diese Programmiersprache nicht kannten, mussten wir uns anfangs in das Programm einarbeiten und die Sprache verstehen lernen. Das Problem mit der Programmiersprache begleitete uns während des gesamten Projektes.

Tätigkeiten

Lauffähigkeit

Um mit der Wiederherstellung der Lauffähigkeit beginnen zu können, musste das Programm zunächst einmal von Delphi4 auf einen kostenlosen Compiler importiert werden. Das Problem an Delphi4 war, dass man lediglich für 14 Tage eine kostenlose Testversion nutzen kann. Da dies aber zu wenig Zeit war haben wir uns für Lazarus entschieden und haben dort die Änderungen im Programm vorgenommen. Dabei ergab sich, dass das Programm auf Bibliotheken verweist, die es gar nicht mehr gibt, oder nicht durch Lazarus unterstützt werden. Das Entfernen dieser Bibliotheken hatte keine Auswirkungen auf das Programm und wurden demnach auch nicht mehr gebraucht.

Als nächstes musste die Ergebnisansicht im Designer angepasst werden. Die schwarze Färbung der Ergebnisse rührte daher, dass auch im Designer die Felder schwarz hinterlegt waren. Diese wurden dann auf farblos gesetzt.

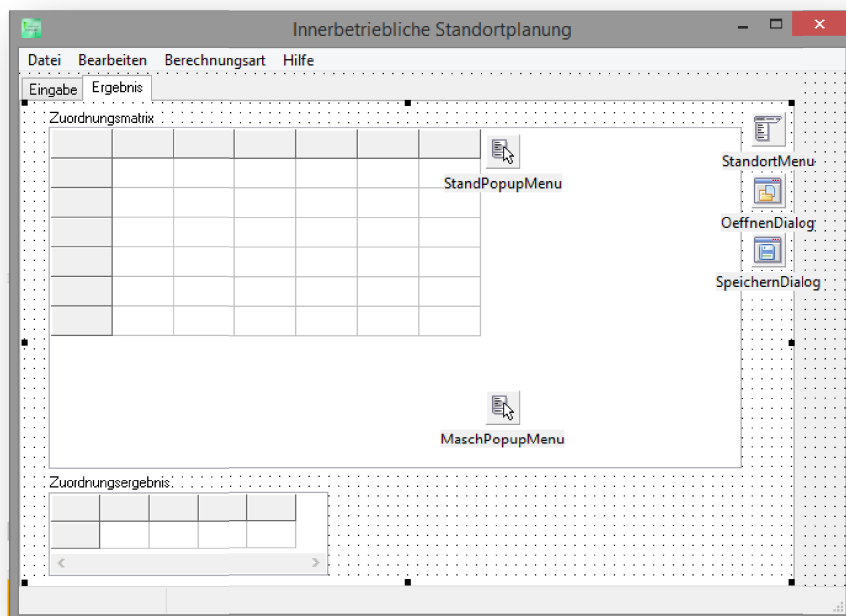


Abbildung 4: Designer Ergebnisansicht

Dach nach wurden die Ergebnissicht zwar nicht mehr schwarz eingefärbt, aber trotzdem wurden keine Ergebnisse angezeigt. Stattdessen wurde der Inhalt unter Ergebnis komplett ausgeblendet. Das konnte dann durch gelöst werden, indem „Formular.Enabled“ und „Formualr.ErgebnisSicht.TabVisible“ auf ‚true‘ gesetzt wurden.

```
procedure TFormular.Berechnen(N: Integer);  
var Daten: TDaten;  
begin  
    Daten := TDaten.Init(N);  
    Daten.EingabeDaten(StandorteGrid, MaschinenGrid);  
    Daten.BerechneMatrix();  
    Formular.Enabled := False;  
  
    QueryPerformanceFrequency(d);  
    QueryPerformanceCounter(Von);  
  
    if EnumRadio.Checked then  
        Daten.RechneEnumeration();  
    else  
        Daten.RechneEntscheidungsbaum();  
  
        Formular.Enabled := True;  
        Formular.ErgebnisSicht.TabVisible := True;  
    end;  
end;
```

Abbildung 5: Enabled und TabVisible

```
procedure TFormular.ChangeCheck();  
begin  
    if EingabeChanged = False then  
        begin  
            SetStatusInfo('');  
            StatusBar.Panels[0].Text := '';  
            // ErgebnisSicht.TabVisible := False;  
            EingabeChanged := True;  
        end;  
    if DateiOn then  
        SpeichernItem.Enabled := True;  
end;
```

Abbildung 6: Auskommentieren von TabVisible

Damit der Hintergrund Thread problemlos die Berechnung starten kann, musste der Aufruf von „resume“ auf „start“ geändert werden. So wird die Berechnung als Hintergrund Thread nebenläufig gestartet und führt seine Berechnungen im Hintergrund durch.

```
Berechnung := TDakin.Create(ZMatrix, AnzOrte);  
Berechnung.Start; // von Resume zu Start geändert
```

Abbildung 7: Hintergrundthread Berechnung starten

Hilfefunktion

Der Reiter „Hilfe“ enthielt bisher keine Hilfe, sondern lediglich Informationen über den Entwickler. Durch das Einbinden eines weiteren Items „Hilfe“ sollte eine Hilfestellung zur Anwendung der Methode und als Interpretationshilfe der Lösung bereitgestellt werden.

Bedingt durch die teilweise fehlenden Kenntnisse der Programmiersprache Pascal, konnte jedoch kein weiteres Item eingebunden werden. Nach Absprache mit Serkan Önnisan wurde daher aus dem Item „Über...“ die neue Hilfe. In der neuen Hilfefunktion wurden die Informationen des alten Infobuttons übernommen.

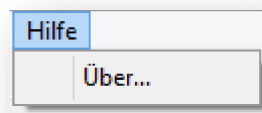


Abbildung 8: altes "Über..." für Information, nun als Hilfefunktion



Abbildung 9: alte Informationen unter "Über..."

Über...

Version	Author	Datum
1.0	Marc Junghänel	04. Oktob

Abbildung 10: integrierte Informationen in der Hilfe

Als Darstellung der Hilfefunktion haben wir uns nach Absprache mit Serkan Önnisan für eine HTML-Seite entschieden. Die HTML-Datei wird im Code mit `OpenDocument()` in der Klasse `standort.pas` geöffnet.

```
procedure TFormular.InfoClick(Sender:
begin
// Vorher: Inform.ShowModal;
// Neue Änderung ruft die Hilfe im W
OpenDocument('Hilfe\Hilfe.html');
```

Abbildung 11: Einbindung der HTML-Hilfe-Datei

Die HTML-Datei ist im Ordner „Hilfe“ unter „Programm“ zu finden und kann entsprechend verschiedener Änderungen jederzeit angepasst werden.

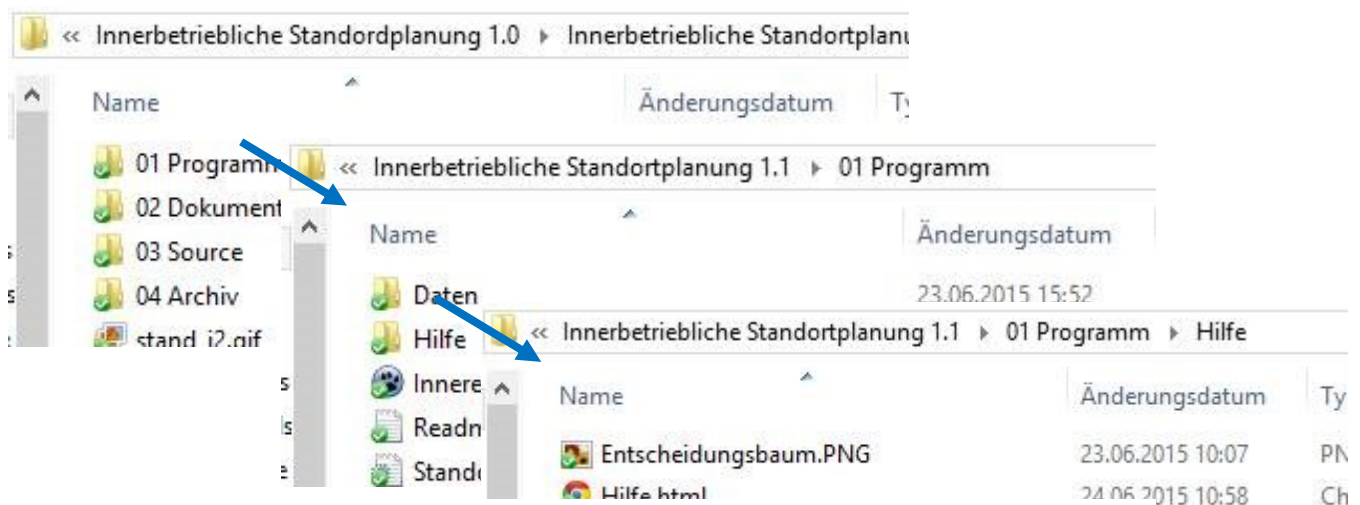


Abbildung 12: Ordnerstruktur mit Hilfedatei

Interne Berechnung

Die Methode nutzt keinen externen Solver, sondern führt die Berechnung des Optimierungsproblems intern durch. Diese wird im Dokument „Berechnung“ im Ordner „02 Dokumentation“ näher erläutert.

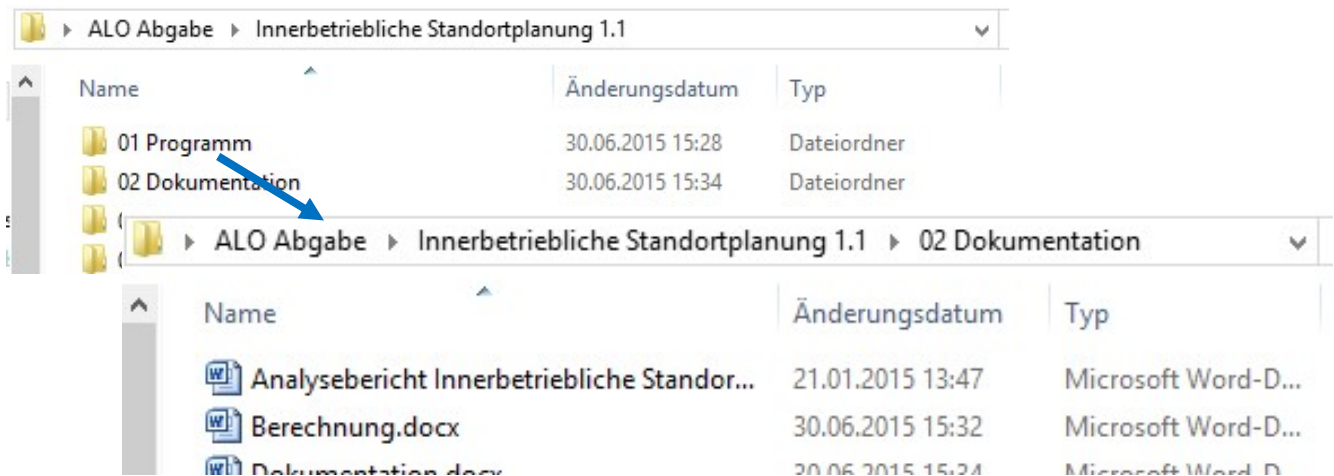


Abbildung 13: Ordnerstruktur Berechnung

Handlungsempfehlung

Da die veraltete Programmiersprache Pascal an der Hochschule nicht mehr gelehrt wird, ist eine Bearbeitung nur durch intensives Einlesen und Einarbeiten in diese Sprache möglich. Aus diesem Grund würden wir eine Neuprogrammierung dieser Methode in eine andere Sprache, beispielsweise Java, empfehlen. Eine solche Neuentwicklung kann sicherlich im Rahmen eines ALO-Projektes durchgeführt werden.

Abbildungsverzeichnis

Abbildung 1: Ergebnisausgabe.....	4
Abbildung 2: Einstellungen mit optionaler Solverpfadeingabe	5
Abbildung 3: Überfunktion und neue Hilfefunktion	6
Abbildung 4: Designer Ergebnisansicht.....	7
Abbildung 5: Enabled und TabVisible	8
Abbildung 6: Auskommentieren von TabVisible	8
Abbildung 7: Hintergrundthread Berechnung starten.....	9
Abbildung 8: altes "Über..." für Information, nun als Hilfefunktion.....	10
Abbildung 9: alte Informationen unter "Über..."	10
Abbildung 10: integrierte Informationen in der Hilfe	11
Abbildung 11: Einbindung der HTML-Hilfe-Datei	11
Abbildung 12: Ordnerstruktur mit Hilfedatei	11
Abbildung 13: Ordnerstruktur Berechnung	12