|  |
| --- |
| ANLO-Projekt SS13 WIN6 Zuordnungsplanun\_2004 HTWG Knstanz |
| Dokumentation der Projektarbeit über Migration des Tools Zuordnungsplanung\_2004 |
| Angelina Bader 285101; Benedikt Wölfle 285148 |

|  |
| --- |
| 02.06.2013 |

Inhaltsverzeichnis

[1. Einleitung 4](#_Toc358724825)

[2. Beschreibung IST-Zustand 5](#_Toc358724826)

[3. Zielvorgaben 5](#_Toc358724827)

[4. Umsetzung 5](#_Toc358724828)

[4.1. LP-Solver 5](#_Toc358724829)

[4.2. Ersetzen des XA-Solvers durch MOPS 7](#_Toc358724830)

[4.3. INI-Datei 11](#_Toc358724831)

[4.4. Aufräumfunktion 12](#_Toc358724832)

[1.1.1. LP-Solve 13](#_Toc358724833)

[1.1.2. MOPS 13](#_Toc358724835)

[5. Neuer IST-Zustand 14](#_Toc358724837)

[6. Budgetauswertung 15](#_Toc358724838)

[7. Fazit 15](#_Toc358724839)

# Einleitung

Zu Beginn kann gesagt werden, dass das Tool Zuordnungsplanung 2004 vor Projektstart des Zweierteams (Angelina Bader und Benedikt Wölfle) auf Windows 7 nicht lauffähig war. Daraus erfolgte der Wunsch einer Migration des Tools auf das Betriebssystem Windows 7 in der 64Bit Version. Dies soll die zukünftige Nutzung auf diesem Betriebssystem ermöglichen. Und es Studenten erleichtern weiter Verbesserungen in am Programm vorzunehmen.

# Beschreibung IST-Zustand

Im Folgenden wird kurz der IST-Zustand des Tools Zuordnungsplanung 2004 beschrieben, wie er vor der Überarbeitung war.

Das Tool nutz zur Lösung von Zuordnungsproblemen zum Beispiel im Personalbereich verschiedene Solver. Neben einer durch die Studenten selbst entwickelten Enumerationsalgorithmus, werden die Solver LP-Solve und XA zur Problemlösung eingesetzt.

Der Nutzer kann über eine Menüleiste auswählen welchen Solver er für die Lösung des eingegebenen Problems nutzen möchte. Das eingegebene Zuordnungsproblem wird dann mithilfe des gewählten Solver gelöst und das Ergebnis wird auf dem Bildschirm ausgegeben.

Das Programm lief problemlos unter XP. Nach der Portierung auf Windows 7 waren jedoch die beiden Solver LP-Solve und XA nicht mehr funktionsfähig. Bei gewähltem LP-Solver trat immer die Fehlermeldung auf, dass die „Solver-pif“- Datei nicht ausgeführt werden kann. Die der HTWG zur Verfügung gestellte Version des XA-Solver war im Allgemeinen nicht unter Windows 7 lauffähig. Alle weiteren Funktionen wie das Laden und Speichern von Eingaben im CSV-Format funktionierten jedoch weiterhin problemlos.

# Zielvorgaben

Als Ziel haben wir uns gesetzt, die Lauffähigkeit eines Solvers zu gewährleisten. In dem Fall waren es hier entweder der XA-Solver oder der LP-Solve.

Bei Nichtfunktionalität eines Solvers besteht die Möglichkeit ein Substrat einzusetzen, was bedeutet, dass ein nicht-funktionsfähiger Solver einfach durch einen funktionsfähigen ausgetauscht wird.

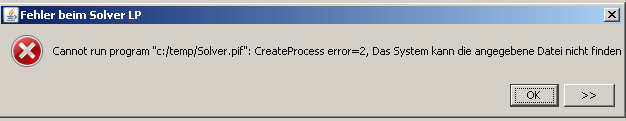
Die Pfade im Programmcode des alten Tools waren hart reincodiert, was einem schlechten Programmierstil entspricht. Dies sollte geändert werden. Die Pfade zu den einzelnen Solvern sollten nun durch eine ini.Datei gesetzt werden können. So können die Pfade an einer zentralen Stelle verwaltet werden, ohne etwas am Programmcode ändern zu müssen.

Ein weiteres Ziel war das Erstellen einer Aufräumfunktion. Alle Ergebnisse sollten nach der Berechnung des Tools aus dem temporären Speicher gelöscht werden, was zu einer Fehlerminimierung führt.

# Umsetzung

## LP-Solver

Wie bereits erwähnt kam bei der Nutzung des LP-Solvers immer folgende Fehlermeldung auf.



Nach längerer Fehleranalyse des Quellcodes sind wir auf das Problem gestoßen. Die Solver.pif Datei wurde dazu genutzt, eine Batch-Datei Namens „Solver.bat“ über ein MS-Dos Programm zu starten. Nach einer Internetrecherche war klar, dass diese Art von MS-Dos Programmen unter Windows 7 nicht mehr ausführbar sind. Um dieses Problem zu umgehen mussten wir den Quellcode so abändern, dass die „Solver.bat“-Datei nicht mehr über ein MS-Dos Programm gestartet wurde, sondern direkt durch das Java-Programm selbst ausgeführt wird. Um dies zu realisieren wurden in der Klasse SolveLP folgende Änderungen vorgenommen.



3

2

1

1. In einem ersten Schritt wurden alle Aufrufe der Datei „Solver.pif“ auskommentiert um den fehlerhaften Aufruf aus dem Programm zu entfernen. Damit in Zukunft alle Änderungen nachverfolgt werden können, wurden die auskommentierten Stellen im Quellcode gelassen und nicht gelöscht.
2. Als Ersatz für die „Solver.pif“-Datei haben wir das Programm so umgeschrieben, dass bei der Ausführung eine Batch-Datei erzeugt wird, welche die gleiche Funktion hat wie das MS-DOS Programm besitzt. Diese neue „lpbatch.bat“ ruft über Konsoleneingaben zuerst das in der INI-Datei festgelegte Arbeitsverzeichnis auf und startet in diesem dann die zuvor dorthin kopierte Datei „Solver.bat“. Um die Erzeugung auch auf langsameren Systemen gewährleisten zu können, wird im Programm immer kurzeitig (500ms) gewartet bis alle notwendigen Dateien erzeugt wurden.
3. Im dritten und letzten Schritt wird dann noch die durch die „Solver.bat“ Datei erzeugte Ausgabedatei, soweit vorhanden, eingelesen und im weiteren Programmverlauf verarbeitet. Zur Fehlervermeidung wurde hier wiederum eine Wartefunktion implementiert, welche sicherstellt, dass die notwendige „lp\_solve.out“ Datei im Arbeitsverzeichnis vorhanden ist.

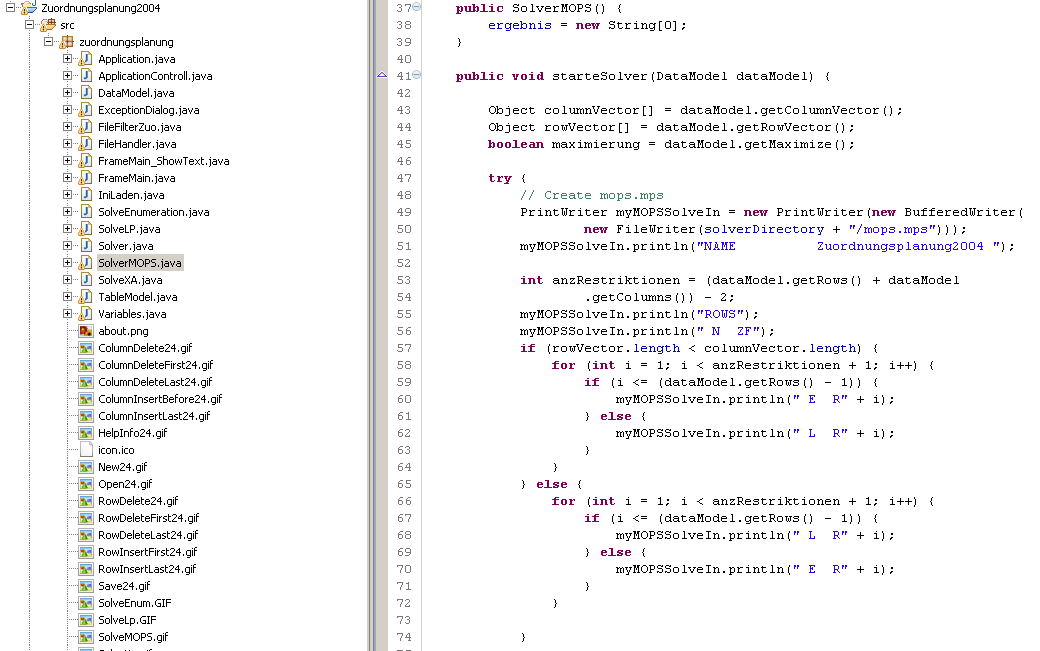
Diese 3 durch uns im Rahmen der ANLO-Vorlesung implementierten Schritte erfüllen die durch das Commitment gestellte Anforderung, mindestens einen Solver auf Windows 7 lauffähig zu machen.

## Ersetzen des XA-Solvers durch MOPS

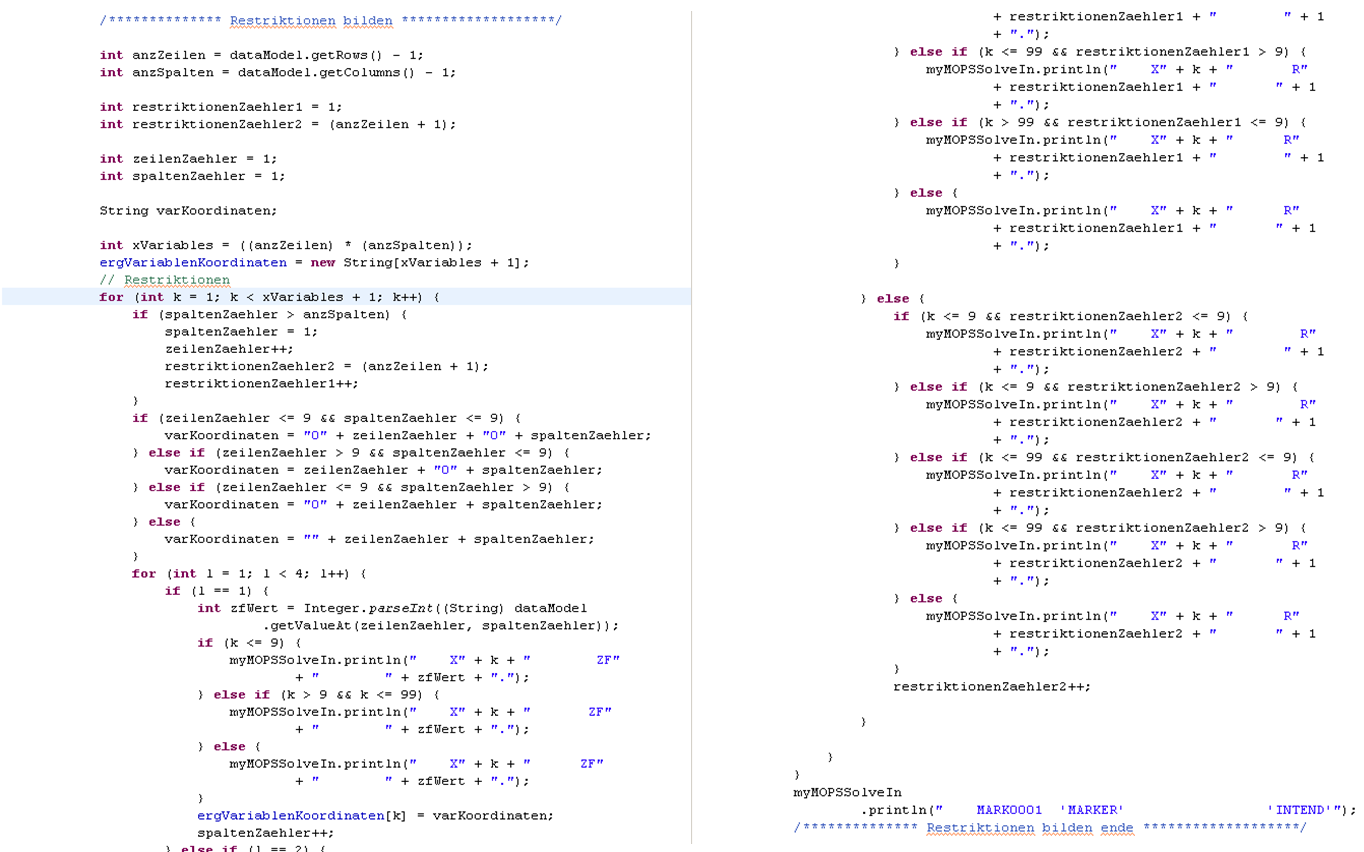
Im Rahmen des Teamprojektes „Migration der OR-Methodendatenbank auf Windows 7“ wurde festgestellt, dass der XA-Solver in seiner derzeitig verfügbaren Version nicht unter Windows 7 lauffähig ist. Aufgrund der nicht vorhandenen Möglichkeit den XA-Solver umzuprogrammieren, haben wir uns dazu entschlossen den XA-Solver durch den MOPS-Solver zu ersetzten. Der Vorteil des MOPS-Solvers liegt in der Nutzung des definierten MPS-Formates für Paramtereingaben. Durch diese Umstellung auf ein standardisiertes Eingabeformat, kann das Programm auch mit zukünftigen Solvern arbeiten, da die Grundstruktur zur Dateierzeugung somit bereits gelegt wurde.

Die Umsetzung erfolgte folgendermaßen. Wir haben eine neue Klasse namens SolverMOPS angelegt und die Programmoberfläche entsprechend angepasst. Anstatt dem XA Solver kann nun über die Menüleiste der MOPS gewählt werden. Ebenso wurde die Ausgabe entsprechend angepasst.

Da der MOPS komplett neu geschrieben wurde ist der gesamte Code im der oben genannten Klasse im Programmordner „source“ zu finden. Jedoch soll im Folgenden in wenigen Screenshots die grundsätzliche Struktur aufgezeigt werden



Bereits zu Beginn der Erezugung der MPS-Datei muss darauf geachtet werden, ob die Eingabematrix symmetrisch ist oder nicht. Je nachdem ob es mehr Bewerber, mehr Stellen gibt bzw. eine symmetrische Matrix vorliegt, ändern sich die Restriktionen. E steht hierbei für Equal(=) und L für lowerthan (<=).



Nachdem die Gleichheitszeichen für die Restriktionen gesetzt wurden, können nun die Restriktionen selbst erzeugt werden. Hierbei muss darauf geachtet werden, dass jede X-Variable nacheinander abgehandelt wird. Wir definieren also für z.B. die Variable X1 zuerst die Zielfunktion(ZF) und daraufhin alle dazugehörigen Restriktionen(R). Anhand von Zeilen- und Spaltenzählern legen wir fest, welche X-Variable abgehandelt wird. Dabei startet das Programm in der linken oberen Ecke des Programms und wandert dann zeilenweise nach rechts. Ist es am Ende einer Zeile angelangt, so springt es in die nächste. Solange bis alle Felder der Eingabematrix bearbeitet wurden.

Aufgrund dessen, dass in einer MPS-Datei jedes Leerzeichen beim einlesen der Datei entscheidend ist, mussten wir mehrere if-else if-else Anweisungen implementieren, um eine korrekte Eingabedatei zu erhalten. Hierbei wurde auf die maximal mögliche Größe der Eingabematrix von 18x18 geachtet.



Als letztes legen wir in der MPS-Datei noch die Grenzen der x-Werte fest. Über die „BOUNDS“ Eingabe stellen wir die Ganzzahligkeit der Lösung und der variablen sicher und das die X-Werte der Zielfunktion nicht größer 1 werden können.

Neben der MPS-Datei benötigt der MOPS Solver auch eine sogenannte „XMOPS“ Datei, welche z. B. festlegt ob es Maximierungs- oder Minimierungsproblem ist. Daher wird nach der MPS-Datei dies „XMOPS“ Datei auf die gleiche Art und Weise erzeugt und wir unterscheiden hierbei anhand der boolean Variabel maximierung um welche Art von Problem es sich handelt.



Nach der Dateierstellung wird die MOPS.exe, aus dem in der INI-Datei angegebenen Verzeichnis, in das Arbeitsverzeichnis kopiert. Dies geschieht wie bei dem LP-Solve über eine Batch-Datei. Nach dem kopieren wird der MOPS-Solver gestartet und auf die Ausgabedatei gewartet.

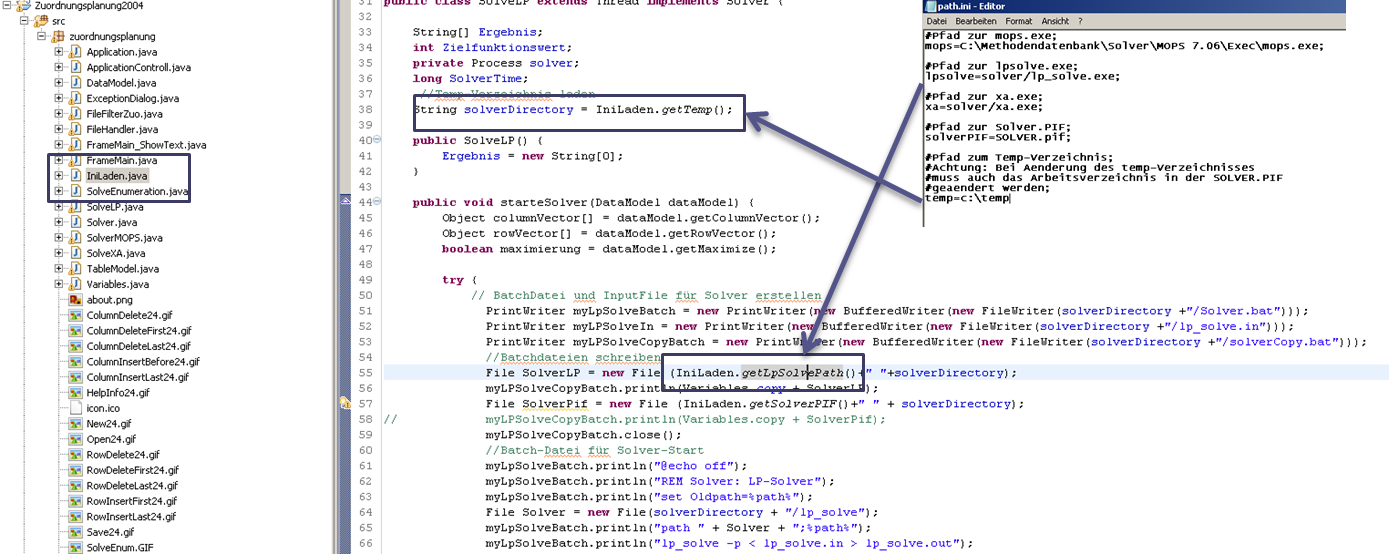
Die fertige Eingabedatei kann Beispielhaft dem Anhang entnommen werden.



Im letzten Schritt wird dann noch das Ergebnis aus der Ausgabedatei „Mops.lps“ ausgelesen und den entsprechenden Variablen aus der Eingabematrix zugewiesen um die Ausgabe nach dem Muster, BEWERBER -> STELLE, zu erhalten.

## INI-Datei

Um die Weitergabe des Programms auf andere Rechner zu erleichtern haben wir uns als weiteres Ziel gesetzt, das Programm um eine sogenannte INI-Datei zu erweitern. Über diese Datei können die Pfade für die verschiedenen Solver und das Arbeitsverzeichnis geändert werden, ohne den Programmcode selbst ändern zu müssen. Bisher waren alle Pfade im Programmcode selbst hinterlegt.

Die Implementierung dieser Funktion erfolgte durch das Erstellen der Klasse IniLaden. Innerhalb dieser Klasse werden die einzelnen Pfade aus einer Textdatei namens „Path.ini“ eingelesen und Variablen zugewiesen. Alle anderen Klassen können dann auf diese Variablen zugreifen.

Um das Programm jederzeit in seinen früheren Zustand zurückführen zu können, sind in der INI-Datei auch die Pfade für die „Solver.pif“-Datei und der Pfad für den XA-Solver hinterlegt. Bei Nutzung dieser Pfade müsste der Quellcode entsprechend angepasst werden.

Durch diese von uns implementiert INI-Datei werden Großteile der Klasse Variables nun durch die Klasse IniLaden abgedeckt.

## Aufräumfunktion

Wie auf der folgenden Programmcodeausschnitt zu sehen ist wird zunächst eine Batch Datei erzeugt. Diese enthält alle notwendigen Parameter, die für den entsprechenden Solver von Bedeutung sind. Hier galt es alle Parameter zu berücksichtigen und auf Vollständigkeit zu achten.

Anschließend wird die Batch Datei gestartet. Ein weiterer Punkt, der eine wichtige Rolle spielte waren die Berechtigungen an Hochschulrechnern, welche bei der Entwicklung oftmals Steine in den Weg legten.

Sinn der Aufräumfunktion war die Minimierung der Fehlerquote. Da es vorkommen konnte, dass nach einer nicht-erfolgreichen Berechnung das Ergebnis einer vorherigen Berechnung (welche erfolgreich war und gespeichert wurde) ausgegeben wurde, musste diese Schwachstelle behoben werden.

### LP-Solve

### 

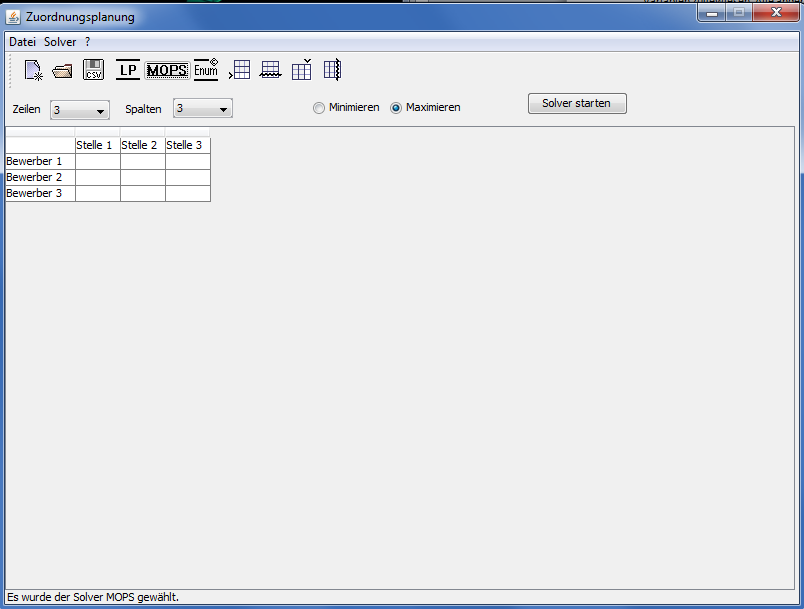
### MOPS

### 

# Neuer IST-Zustand

Nach allen vorgenommen Änderungen ist der neue IST-Zustand des Programms Zuordnungsplanung 2004 gemäß des Commitments. Der LP-Solver ist funktionsfähig, der XA-Solver wurde durch MOPS ersetzt, Pfade können nun über eine INI-Datei eingelesen und verwaltet werden und es wurde eine „Aufräumfunktion“ zur Fehlervermeidung implementiert.

Im Rahmen der Änderungen wurde ebenfalls die Oberfläche entsprechend angepasst. So wird nun überall wo vorher der auf den XA-Solver verwiesen wurde nun auf den MOPS verwiesen. Die neue Startoberfläche sieht damit wie folgt aus.



Außer den Änderungen bezüglich des MOPS-Solvers, wurden keine weiteren Änderungen an der Oberfläche vorgenommen.

Alle Funktionen wie Laden und Speichern sind voll funktionsfähig. Alle Programmänderungen sind schriftlich in diesem Dokument festgehalten oder im Quellcode kenntlich gemacht durch z.B. Kommentare.

Das Programm Zuordnungsplanung 2004 ist somit vollständig unter Windows 7 lauffähig.

# Budgetauswertung

Die Budgetauswertung in diesem Projekt sieht so aus, dass 60 Stunden pro Person zur Verfügung stehen. In Real wurden ca. 30 Stunden für die Programmierung aufgewandt (pro Person). Anschließend folgten die Fehleranalyse, das Testing und das Schreiben der Dokumentation, was wiederum 15 Stunden in Anspruch nahm. Wie zu sehen ist, konnten und mussten die zur Verfügung stehenden 60 Stunden nicht ganz ausgeschöpft werden. Jedoch ist zu beachten, dass immer mit unerwarteten Komplikationen zu rechnen ist, welche weitaus mehr Zeit in Anspruch nehmen können, was in diesem Projekt glücklicherweise nicht der Fall war. Ein durchdachter Start des Projektes konnte bereits viele Risiken und Fehlerquellen ausschließen.

# Fazit

Als Fazit lässt sich abschließend sagen, dass unser Projekt erfolgreich abgeschlossen und an das Teamprojekt übergeben werden konnte. Die Fehleranalyse des Tools hat sich jedoch aufgrund des Programmumfangs und der teilweise fehlerhaften Dokumentation als sehr schwierig herausgestellt. Wir brauchten insgesamt deutlich länger für die Fehleranalyse als erwartet und mussten somit die Programmierarbeiten in weniger Zeit erledigen als geplant.

Das Umprogrammieren wurde durch die Wahl des MOPS-Solver nicht unbedingt vereinfacht. Jedoch kamen auch gerade durch die Wahl des MOPS-Solvers viele Verbessrungsideen für das Tool erst während der Programmierung. Bei der Anbindung des MOPS-Solvers mussten wir auf viele Details achten, haben aber, dank des MPS-Formats, ein zukunftsfähiges Tool geschaffen.

Als letzte Hürde bei der Neuentwicklung standen uns noch die Hochschulberechtigungen im Wege. Das Berechtigungskonzept der Hochschule schränkt die Programmierung der Tools teilweise ein. Man muss viel Arbeit investieren bis die Berechtigungen kein Problem mehr für das Tool darstellen.

Trotz aller Probleme war es ein insgesamt erfolgreiches Projekt, welches im Rahmen des vorhanden Budgets abgeschlossen werden konnte.

# Anhang

Beispiel für die iengabedateien LP-Solve und MOPS

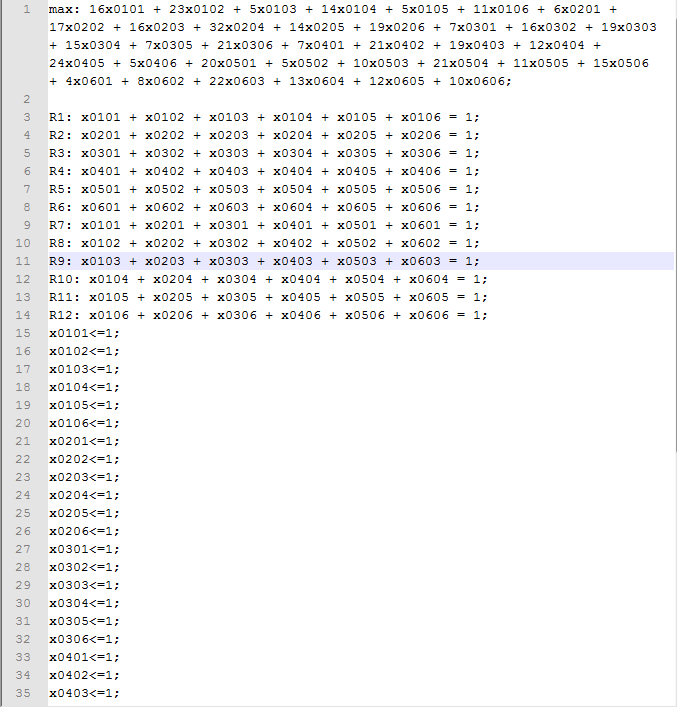
**Eingabematrix:**

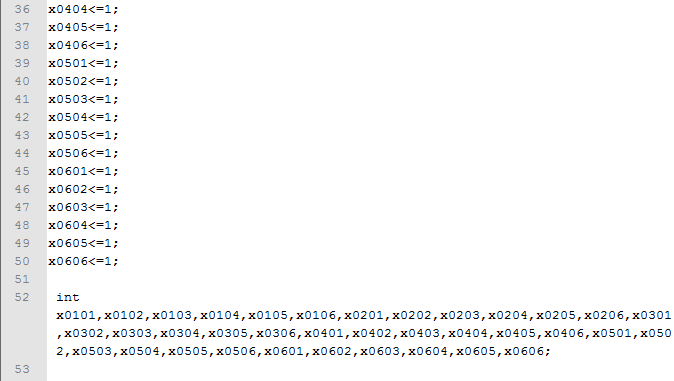
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Marketing | Vertrieb | Personalwesen | Produktion | Finanzierung5 | Vertragswesen |
| Frommburger | 16 | 23 | 5 | 14 | 5 | 11 |
| Weiß | 6 | 17 | 16 | 32 | 14 | 19 |
| Primavera | 7 | 16 | 19 | 15 | 7 | 21 |
| Broesel | 7 | 21 | 19 | 12 | 24 | 5 |
| Bauer | 20 | 5 | 10 | 21 | 11 | 15 |
| Arbor | 4 | 8 | 22 | 13 | 12 | 10 |

* **MAXIMIERUNG**

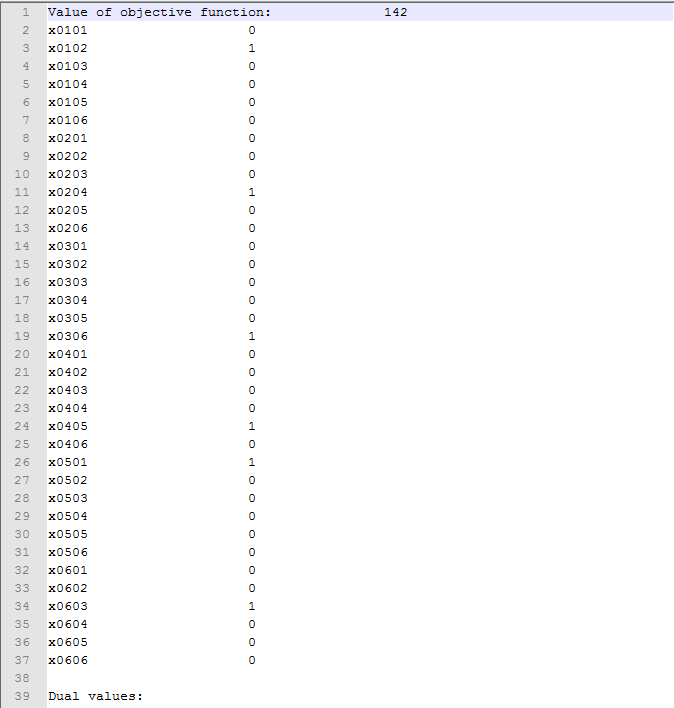
LP-Solve:

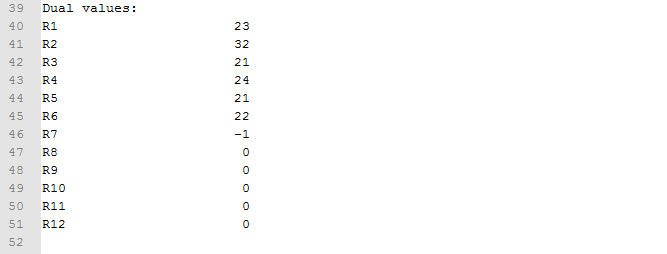
Eingabe Datei (lp\_solve.in)->





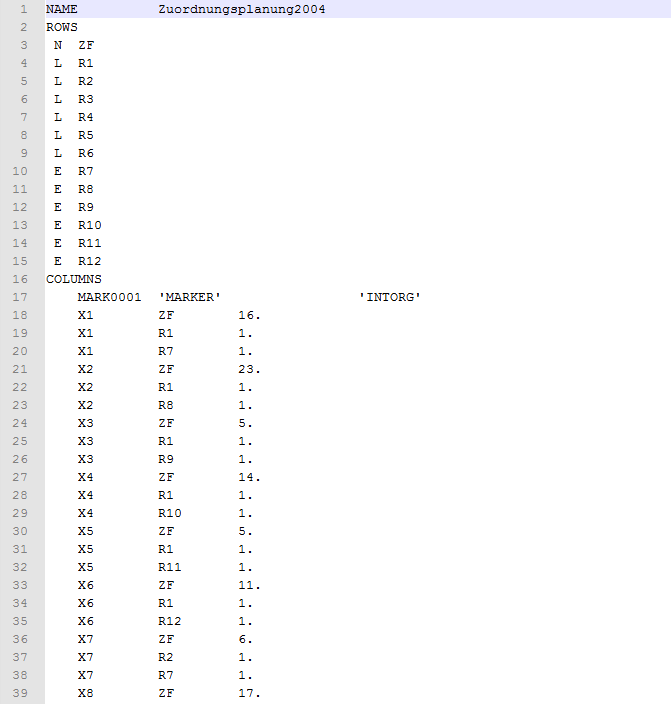
Ergebnis Datei (lp\_solve.out) ->:

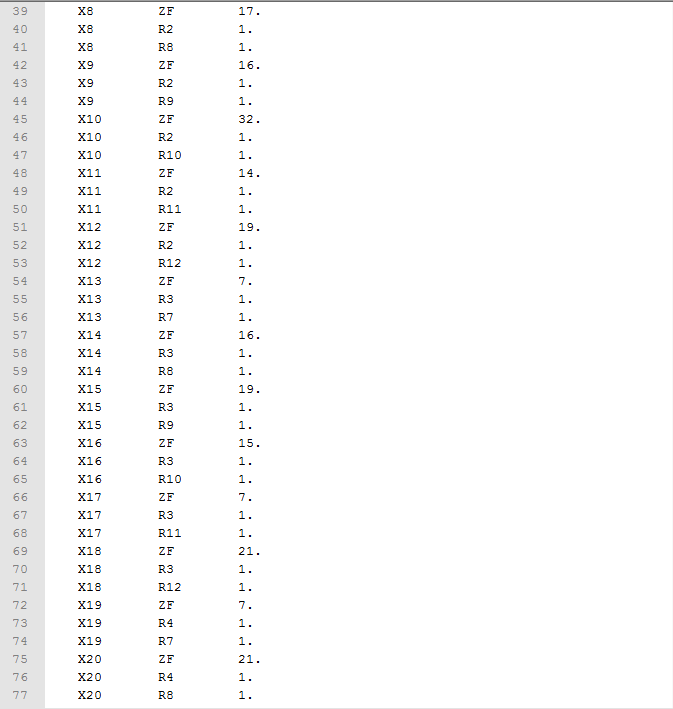


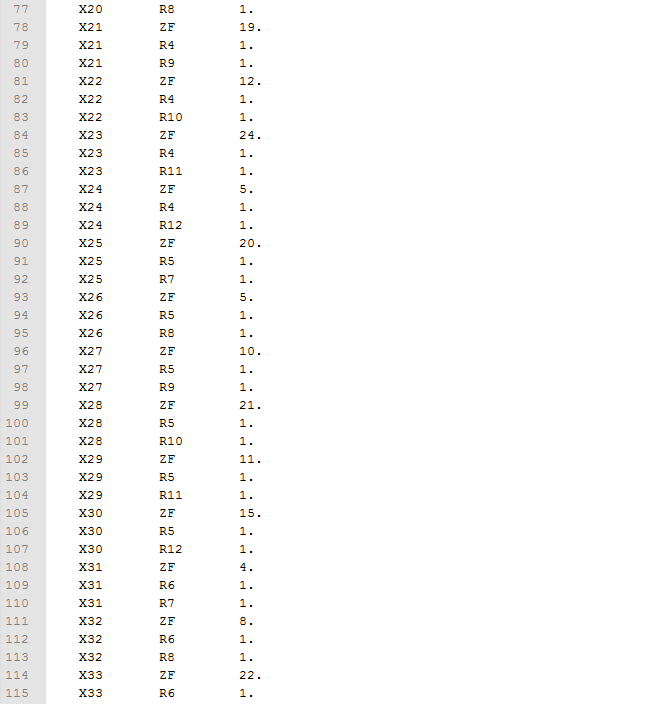


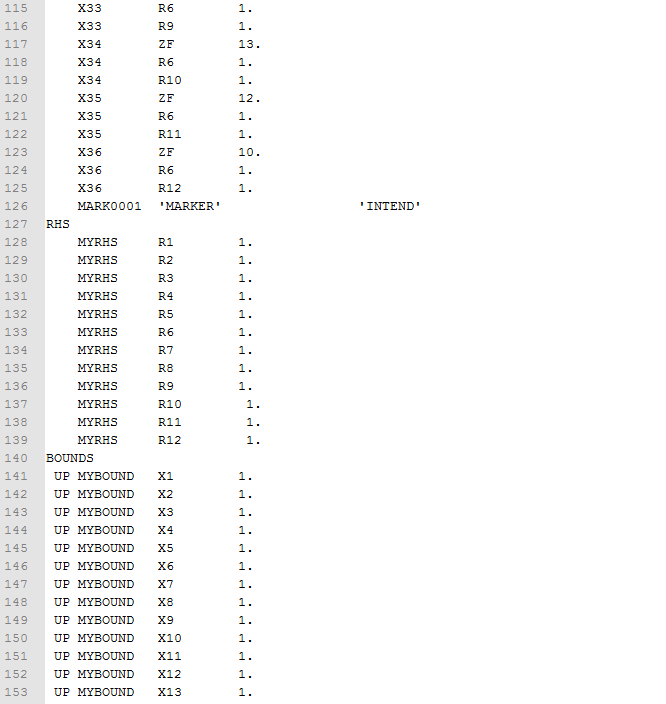
MOPS-SOLVER

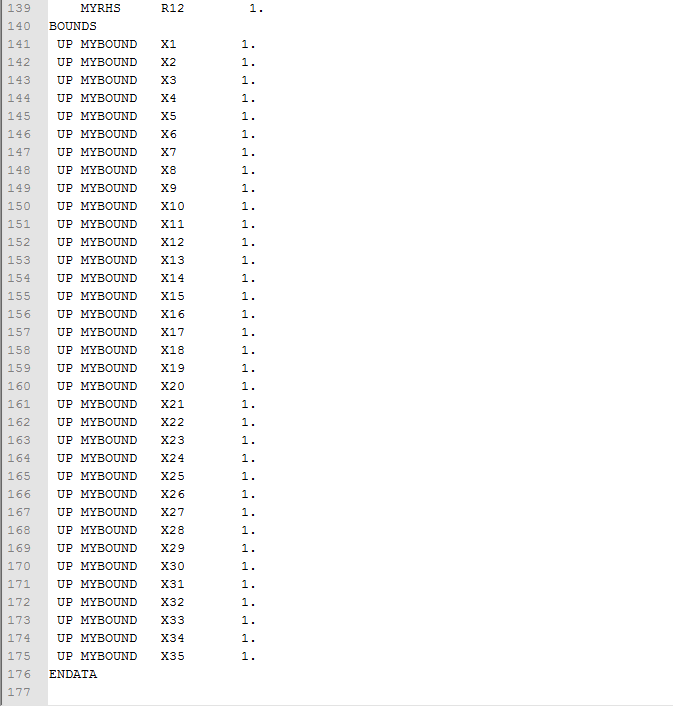
Eingabe-Datei (mops.mps)->



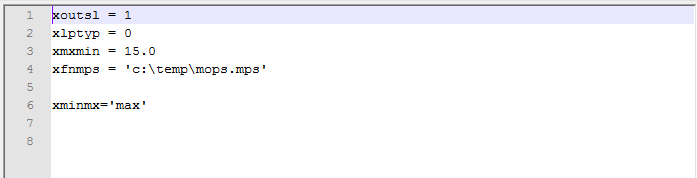




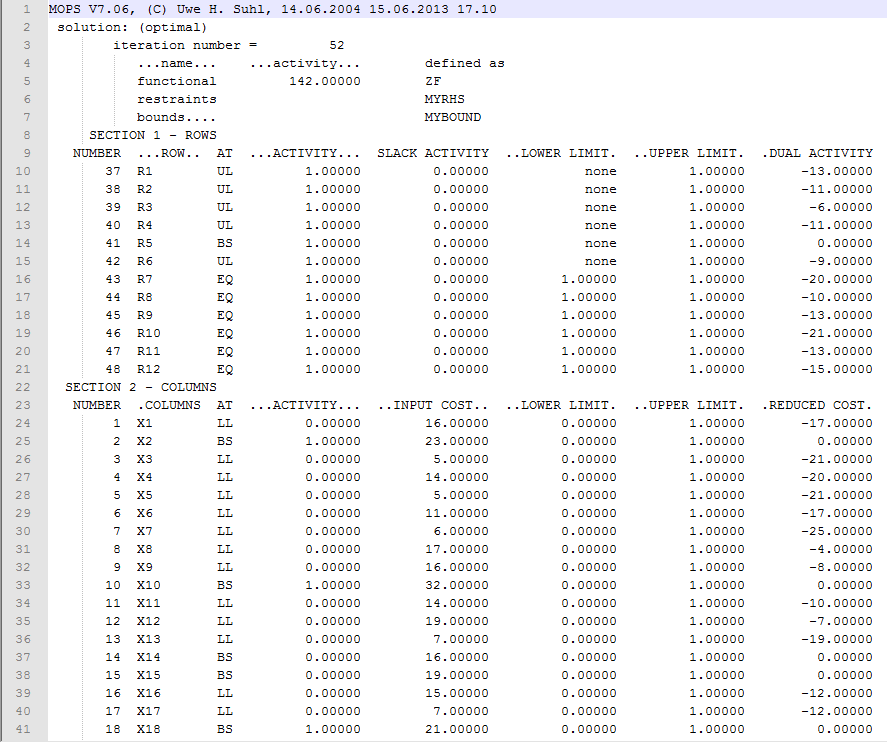


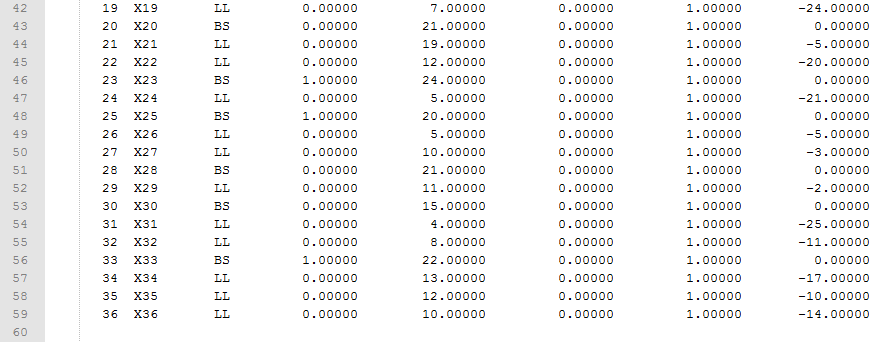


Einstellungs- DATEI (xmops.pro)->



Ergebnis-Datei (mops.lps)->





Beide Solver liefern folgendes Ergebnis:

**Maximierung**

**Zielfunktionswert: 142**

**Frommburger -> Vertrieb**

**Weiß -> Produktion**

**Primavera -> Vertragswesen**

**Broesel -> Finanzierung**

**Bauer -> Marketing**

**Arbor -> Personalwesen**