Lineare Optimierung   
Portierung des Programms „Mischungsplanung 2004“

Doreen Brunner, Mark Deppe und Marcel Prügel

Inhalt

[1. Ausgangs –Zustand 3](#_Toc358816315)

[1.1. Mischungsplanung 2004 auf Windows XP 3](#_Toc358816316)

[1.2. Mischungsplanung 2004 auf Windows 7 3](#_Toc358816317)

[1.3. Zu lösende Aufgaben 3](#_Toc358816318)

[2.3 Mockup 4](#_Toc358816319)

[2.4 Umsetzung 4](#_Toc358816320)

[2. Die GUI / Die Tabellen 4](#_Toc358816321)

[3. Testläufe 5](#_Toc358816322)

[Testlauf 1: Test des LP Modells 6](#_Toc358816323)

[Testlauf 2: Test der Ganzzahligkeitsoption 6](#_Toc358816324)

[Testlauf 3: Einfügen einer Sorte 6](#_Toc358816325)

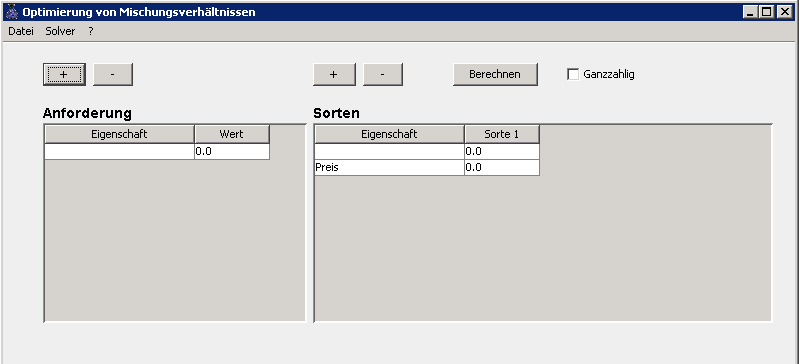
[Testlauf 4: Löschen einer Sorte 7](#_Toc358816326)

[Testlauf 5: Speichern und Laden einer Eingabe 7](#_Toc358816327)

## Ausgangs –Zustand

### Mischungsplanung 2004 auf Windows XP

* Lauffähiges Programm
* Programmiert mit Java 1.4
* Unübersichtliche Oberfläche – irreführende Benennungen



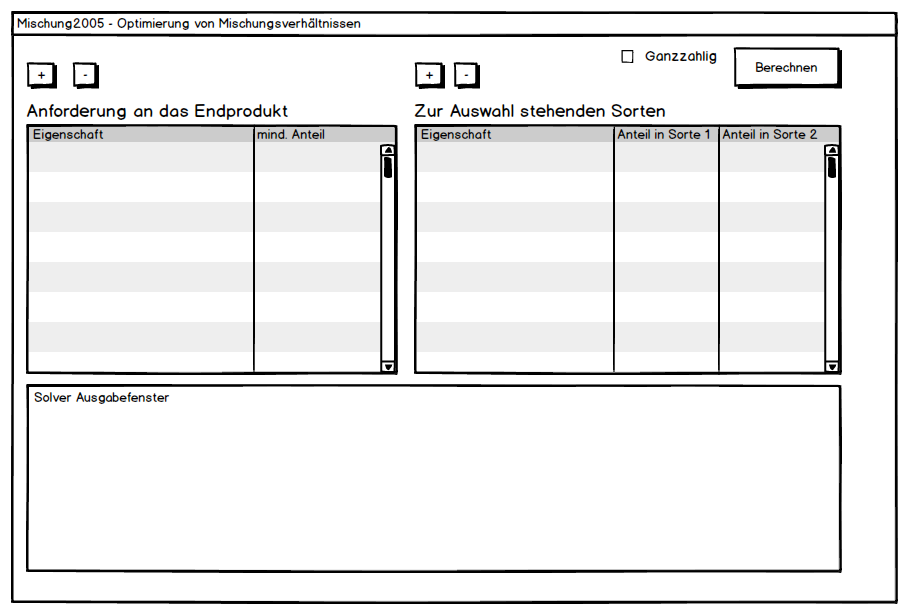
### Mischungsplanung 2004 auf Windows 7

* Konnte in Eclipse ausgeführt werden, aber Solveraufruf war nicht möglich
* Fehler beim Aufruf des Solvers
* Fehler in der Ausgabedatei
* Fehler durch sporadische Fehlermeldungen

### Zu lösende Aufgaben

* Sicherstellen der Lauffähigkeit auf Windows7
* Testen der Lauffähigkeit auf Windows7
* Dokumentation der Durchführung und Ergebnisse
* Übergabe des Programms an die Gruppe des Teamprojektes
* Optimierung der GUI auf Benutzerfreundlichkeit (Bessere Benennungen)

### Mockup

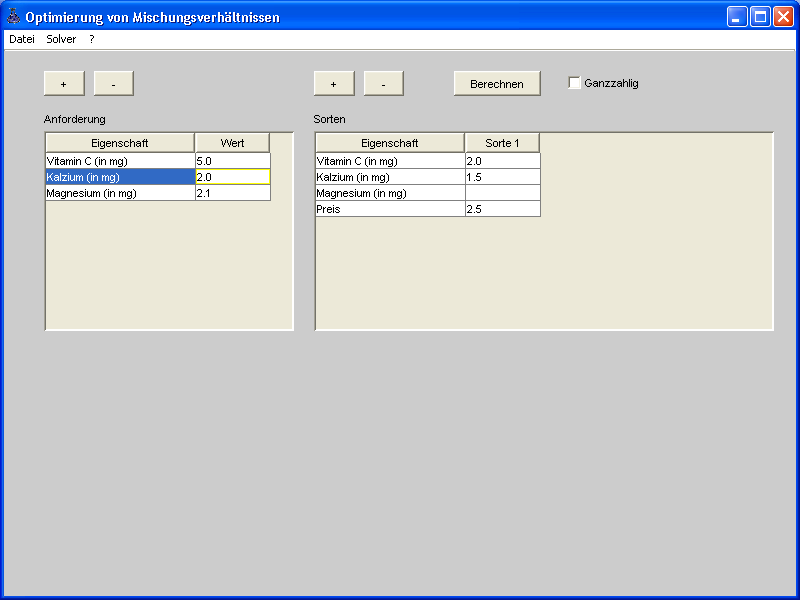


### 2.4 Umsetzung

Nach einigen Tests haben wir festgestellt dass beim Aufruf des Solvers verschieden Pfade nicht richtig übergeben worden sind, was du Fehlern führte. Des weiteren waren in der Ausgabedatei einige syntaktische Fehler. Alle Fehler wurden von uns behoben.

Insgesamt hielt sich der Aufwand dieses Programm zu portieren in Grenzen, obwohl is in der Kategorie B war. Deshalb haben wir versucht die GUI des Programmes durch bessere Benennungen zu verbessern und benutzerfreundlicher zu Gestalten. Da sich unser erstes Programm, Wagner-Whitin jedoch als viel schwerer als gedacht herausstellte (Neuprogrammierung ohne Vorlage) waren die Ressourcen begrenzt noch weitere Verbesserungen vorzunehmen.

## Die GUI / Die Tabellen



10.

9.

7.

8.

7.

6.

5.

4.

3.

2.

1.

1. Fügt eine neue Anforderung hinzu.
2. Löscht die markierte Anforderung (die Anforderung, in der sich der Cursor befindet).
3. Fügt eine neue Sorte hinzu.
4. Löscht die markierte Sorte (die Sorte, in der sich der Cursor befindet).
5. Startet die Berechnung. Im Unterem Bildschirmfester werden die Ergebnisse angezeigt.
6. Schaltet die Ganzzahligkeitsoption ein oder aus. Wenn XA ausgewählt wurde, dann ist dieses Feld inaktiv.
7. In diese Spalte wird der Name der Anforderung eingegeben.
8. In diese Spalte wird der Wert der Anforderung eingegeben.
9. Dieser Spaltenwert wird automatisch von den Anforderungen übernommen.
10. In diese Spalte werden die Werte der Sorte eingegeben.

## Testläufe

Um sicher zu gehen, dass unsere Software einwandfrei läuft, haben wir exemplarisch 5 Testläufe durchgeführt. (Natürlich haben wir viel mehr Tests durchgeführt, diese möchten wir aber nicht alle hier aufführen).

### Testlauf 1: Test des LP Modells

**Anforderungen: Sorten:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Wert** |  | **Sorte 1** | **Sorte 2** | **Sorte 3** |
| Vitamin C (in mg) | 5.0 |  | 2.0 | 3.5 | 1.5 |
| Kalzium (in mg) | 2.0 |  | 1.5 | 0.3 | 0.3 |
| Magnesium (in mg) | 4.5 |  | 0.2 | 0.0 | 1.0 |
| Vitamin A (in mg) | 0.9 |  | 0.1 | 0.3 | 0.1 |
| **Preis** |  |  | 2.5 | 3.0 | 1.7 |

**Resultat:**

|  |  |  |
| --- | --- | --- |
|  | **LP Solve** | **XA** |
| Sorte 1 | 0.14706 | 0.14706 |
| Sorte 2 | 1.4608 | 1.46078 |
| Sorte 3 | 4.4706 | 4.47059 |
| Minimale Kosten | 12.34999992 | 12.35 |

✓ Vergleichstests mit dem LP Modell und LP Interaktiv bestätigt das Ergebnis

### Testlauf 2: Test der Ganzzahligkeitsoption

Wir verwenden die gleichen Werte wie im Testlauf 1, allerdings schalten wir die Ganzzahligkeitsoption ein. Diesen Test führen wir nur mit dem LP Solve aus, da XA nur binäre Ganzzahligkeit kennt.

**Resultat:**

|  |  |  |
| --- | --- | --- |
|  | **LP Solve** | **XA** |
| Sorte 1 | 0 | - |
| Sorte 2 | 1 | - |
| Sorte 3 | 6 | - |
| Minimale Kosten | 13.19999993 | - |

✓ Der Vergleichtest mit LP Interaktiv bestätigt das Ergebnis.

### Testlauf 3: Einfügen einer Sorte

Gleiche Werte wie bei Testlauf 1, allerdings mit einer 4 Sorte, die gegenüber den anderen Sorten wesentlich attraktivere Werte besitzt.

|  |
| --- |
| **Sorte 4** |
| 2 |
| 1.4 |
| 3 |
| 0.3 |
| 1.7 |

**Resultat:**

|  |  |  |
| --- | --- | --- |
|  | **LP Solve** | **XA** |
| Sorte 1 | 0.0 | 0.0 |
| Sorte 2 | 0.0 | 0.0 |
| Sorte 3 | 0.0 | 0.0 |
| Sorte 4 | 3.0 | 3.0 |
| Minimale Kosten | 5.09999994 | 5.1 |

✓ Der Vergleichtest mit LP Interaktiv bestätigt das Ergebnis.

### Testlauf 4: Löschen einer Sorte

Wir löschen die Sorte 4 und starten erneut die Berechnung. Die Ergebnisse sollten mit denen aus Testlauf 1 identisch sein.

✓ Das Ergebnis ist mit Testlauf 1 identisch.

### Testlauf 5: Speichern und Laden einer Eingabe

Wir speichern die Eingabe auf der Festplatte ab, und beenden das Programm.

Anschließend starten wir es wieder und laden die XML Datei. Die Eingaben müssen identisch sein.

✓ Die Eingaben sind identisch.