

Anwendung Linearer Programmierung

Mischungsplanung V 3.1

Optimierung von Mischungsverhältnissen

Datei Solver ?

☐ Ganzzahlig

Anforderung	
Name	Wert
Vitamin C	5.0
Kalzium	2.0
Magnesium	4.5
Vitamin A	0.9

Sorten			
Eigenschaft	Sorte 1	Sorte 2	Sorte 3
Vitamin C	2.0	3.5	1.5
Kalzium	1.5	0.3	0.3
Magnesium	0.2	0.0	1.0
Vitamin A	0.1	0.3	0.1
Preis	2.5	3.0	1.7

Marius Germann
Matrikel-Nr. 289890

Benedikt Goos
Matrikel-Nr. 289839

Inhaltsverzeichnis

1.	Beschreibung der Methode.....	3
2.	Nutzungshinweise	3
2.1	Aufbau der Registerkarten:	3
2.2	Bedienungsoberfläche:.....	4
2.3	Konfiguration:.....	5
3.	Beschreibung des LP-Ansatzes:	5
4.	Infos über verwendete Solver:	6

1. Beschreibung der Methode

Das Methodenbankprogramm „Mischungsplanung V 3.1“ dient zur Berechnung von Mischungsverhältnissen. Durch das Eingeben einer gewünschten Mischung wird eine Anforderung definiert. Danach können beliebig viele Sorten beschrieben werden, mit deren Hilfe diese Mischung zusammengestellt werden soll. Das Programm ist an den LP_SOLVE angeschlossen, welcher dann das abgeleitete LP-Modell löst. Anschließend wird das Ergebnis ausgegeben.

Beispiel für eine Anwendung:

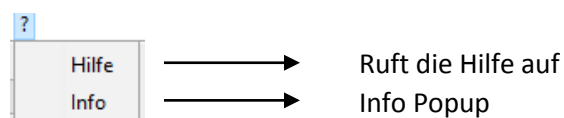
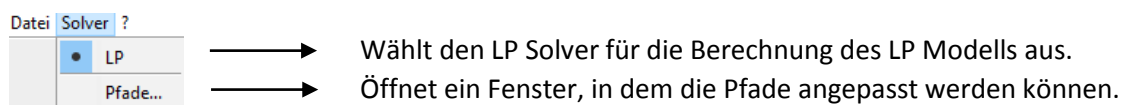
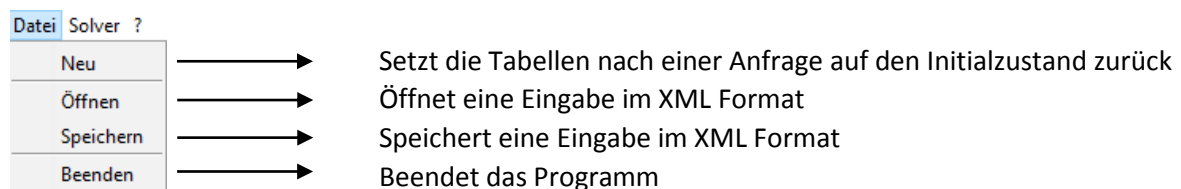
→ Metalllegierung, welche aus verschiedenen Erzen zusammengestellt werden soll.

→ Teemischung, welche aus bestimmten Kräutern hergestellt werden soll.

2. Nutzungshinweise

Das Programm besitzt die drei Registerkarten „Datei“, „Solver“ und „?“. Diese besitzen folgende Funktionen.

2.1 Aufbau der Registerkarten:



2.2 Bedienungsoberfläche:

Optimierung von Mischungsverhältnissen

Datei Solver ?

1 2 3 4 5 6

7 8

9 10

11 12

Anforderung

Name	Wert
Vitamin C	5.0
Kalzium	2.0
Magnesium	4.5
Vitamin A	0.9

Sorten

Eigenschaft	Sorte 1	Sorte 2	Sorte 3
Vitamin C	2.0	3.5	1.5
Kalzium	1.5	0.3	0.3
Magnesium	0.2	0.0	1.0
Vitamin A	0.1	0.3	0.1
Preis	2.5	3.0	1.7

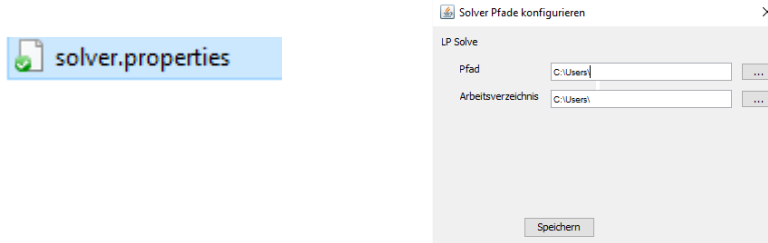
Lösung des LP-Ansatzes

Sorte	Sorte 1	Sorte 2	Sorte 3	Kosten
Sorte 1	0.14706			
Sorte 2		1.4608		
Sorte 3			4.4706	
minimale Kosten:				12.34999992

1. Fügt eine neue Anforderung hinzu.
2. Löscht die markierte Anforderung.
3. Fügt eine neue Sorte hinzu.
4. Löscht die markierte Sorte.
5. Startet die Berechnung. Anschliessend werden Ergebnisse ausgegeben (Nr.11/12)
6. Schaltet die Ganzzahligkeitsoption ein oder aus.
7. In dieser Spalte wird der Name der Anforderung eingegeben.
8. In dieser Spalte wird der Wert der Anforderung eingegeben.
9. Dieser Spaltenwert wird automatisch von den Anforderungen übernommen
10. In dieser Spalte werden die Werte der Sorte eingegeben.
11. Ausgabe der Ergebnisse, wieviel von der jeweiligen Sorte benötigt wird.
12. Ausgabe der Ergebnisse, wie hoch die Kosten sind.

2.3 Konfiguration:

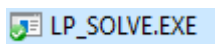
Die Pfade und die Arbeitsverzeichnisse für den Solver lassen sich über Solver > Pfade... einstellen. Die Pfade werden in der Datei solver.properties abgelegt.



In das Arbeitsverzeichnis werden folgende Dateien abgelegt:

- start_lpsolve.bat
- lpsolve_batchOutputStream.txt
- lpsolve_batchErrorStream.txt
- lp_ein.lp
- lp_aus.txt

Wichtig ist das der Solverpfad so eingegeben wird, so dass in dem Ordner die Datei LP_Solve.exe liegt, da ansonsten das Programm nicht funktioniert.



3. Beschreibung des LP-Ansatzes:

Das zugrunde gelegte OR-Modell beschreibt ein Minimierungsproblem. Jede Sorte hat einen Preis und die zusammengestellte Mischung sollte einen möglichst geringen Gesamtpreis aufweisen. Somit entsprechen die Preise der Sorten den c-Werten in der Zielfunktion.

Die Anzahl der Restriktionen ergibt sich aus der Anzahl der Anforderungen, welche die Mischung erfüllen muss. Die „aij“ entsprechen den Eigenschaften der Sorten. Dabei steht das i für die Anforderung und das j für die Sorte. Hierzu ein Beispiel zur Verdeutlichung des LP-Ansatzes.

Anforderungen:

Name	Wert
Vitamin C (in mg)	5.0
Kalzium (in mg)	2.0
Magnesium (in mg)	4.5
Vitamin A (in mg)	0.5
Preis	

Sorten:

Sorte 1	Sorte 2	Sorte 3
2.0	3.5	1.5
1.5	0.0	0.3
0.2	0.0	1.0
0.1	2.0	0.1
2.5	3.0	1.7

Hieraus lässt sich nun folgendes Modell bilden:

Zielfunktion: $2.5 x_1 + 3.0 x_2 + 1.7 x_3 \rightarrow \min$

Restriktion 1: $2.0 x_1 + 3.5 x_2 + 1.5 x_3 \geq 5.0$

Restriktion 2: $1.5 x_1 + 0.0 x_2 + 0.3 x_3 \geq 2.0$

Restriktion 3: $0.2 x_1 + 0.0 x_2 + 1.0 x_3 \geq 4.5$

Restriktion 4: $0.1 x_1 + 2.0 x_2 + 0.1 x_3 \geq 0.5$

4. Infos über verwendete Solver:

Der SolverCaller ist ein Framework zum Aufruf eines Solvers, der Probleme der linearen Programmierung löst. Er wurde von Helmut Lindinger im Rahmen dieser Veranstaltung 2001 entwickelt. Der Caller kann über klar definierte Schnittstellen angesprochen werden. Hierfür möchten wir auf die sehr gute Dokumentation des SolverCallers verweisen, die sich als ZIP-File im Unterordner \model\solverCaller befindet.

Der SolverCaller benötigt für jeden Solver ein eigenes LP Modell in Form eines Stringarrays. Wir haben das Interface LPBuilder definiert, welches die Methode createLPModell(Eingabe eingabe, boolean ganzzahlig) deklariert. Aus der Eingabe soll das erforderliche Stringarray erzeugt werden. Die Klasse LPBuilderLPSolve implementiert das Interface für den LPSolve.

Die Methode startSolver() extrahiert die Eingabe mit getEingabeFromTables() aus den Tabellen der GUI. Anschliessend ruft sie die createLPModell (...) Methode auf und erhält das LPModell, das der SolverCaller benötigt, um daraus die Eingabedateien für die Solver zu erzeugen.