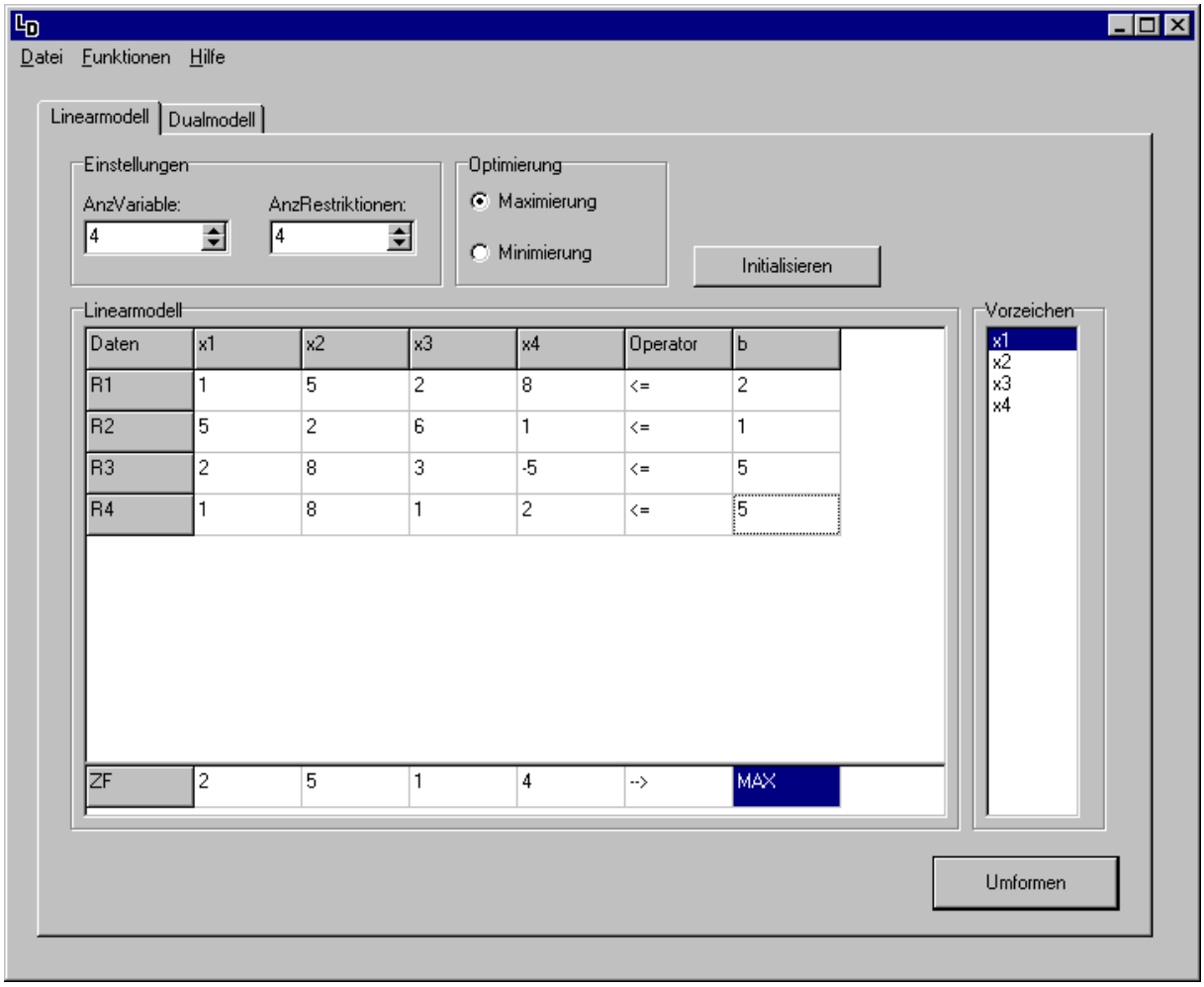
PrimalDual.exe – Die Neuerstellung

... eine Projektbetrachtung



Verfasser	Stephanie Föhrenbach – 254 908
	Markus Wittekindt – 249 963
Betreuer	Prof. Dr. M. Grütz
Veranstaltung	Operation Research 2 – WI 8
Semester	WS 2000 / 2001, FH Konstanz
Datum	Konstanz, 17.01.2001

INHALTSVERZEICHNIS

ABBILDUNGSVERZEICHNIS

1. Aufgabenstellung

Die Initiale Idee ist aus schon bestehenden Klausuraufgaben innerhalb der Vorlesung Operation Research an der Fachhochschule Konstanz entstanden.
Dort wird, neben vielen anderen Aufgaben auch oft verlangt, ein Primalmodell in das entsprechende Dualmodell umzuformen.

Diese Umformung hat für jedes mal die selbe Struktur, d.h. die Vorgehensweise ändert sich nicht. Daher bietet es sich an diesen Ablauf innerhalb eines Programmes zu automatisieren, so dass mehr Zeit für die Modellerstellung verwendet werden kann.
Des weiteren kann ein solches Programm zur Kontrolle und Überprüfung des „von Hand“ erstellten Dualmodells herangezogen werden.

Die Aufgabenstellung bestand also darin, einen neuen Solver zu erstellen, welcher ein vom Benutzer eingegebenes Primalmodell unter der Verwendung der bekannten Vorgehensweise in ein Dualmodell umformt.
Die Vorgehensweise lässt sich an verschiedenen Stellen nachlesen und ist daher nicht Bestandteil dieser Projektbetrachtung. Um nur einige Stellen zu nennen:

- Prof. Dr. M. Grütz, Vorlesungsunterlagen zum Fach Operation Research an der FHK, Kapitel 4 - Dualität
- Meyer, M. / Hansen, K., Planungsverfahren des Operations Research
- Tietze,
- In der Hilfe des Programms[1] unter dem Abschnitt „Theorie“
- ...

In einem Satz ausgedrückt lautet die Aufgabenstellung:

„Neuerstellung eines Solvers, welcher die Umformung eines Primal- in ein Dualmodell vornimmt.“

[1] Eine Textversion der Hilfedatei ist dieser Projektbetrachtung als Anhang beigefügt

2. Von der Idee zur Realisierung

Nachdem die Aufgabenstellung klar war, stellte sich die Frage wie ein solcher Solver realisiert werden kann. Einige Rahmenbedingungen waren von vorneherein gegeben, so dass sich der Lösungsweg nur innerhalb dieses Rahmens abspielen konnte.
Der Rahmen wurde abgesteckt durch folgende Vorgaben:

- Die Erstellung durfte nicht länger als ein Semester benötigen
- Für das Entwicklungstool muss der Fachbereich WI an der FHK zumindest eine Lizenz besitzen
- Das Entwicklungsteam besteht aus zwei Mitgliedern, Stephanie Föhrenbach und Markus Wittekindt (beide WI 8)

Von dem Entwicklungsteam kamen dann noch folgende Vorstellungen hinzu

- Das Programm sollte leicht handhabbar und übersichtlich gestaltet sein
- Die Oberfläche sollte „Windows – like“ sein
- Das Programm sollte unabhängig von dem Entwicklungstool laufen und so z.B. auf keinerlei Runtime Bibliotheken angewiesen sein
- Die Verarbeitung sollte schnell und stabil sein
- Die Bedienung sollte mit der Maus, aber auch über Shortcuts möglich sein

Als erster Stelle stand die Frage nach dem groben Oberflächenentwurf. Schnell war klar, dass die Darstellung der Restriktionen und der Zielfunktion innerhalb eines Grids erfolgen sollte. Dieses Grid sollte für den Benutzer leicht editierbar sein. Wichtig war auch, dass die Eingabe komplett mit der Tastatur realisiert werden kann, da erfahrungsgemäß das Wechseln zwischen den einzelnen Gridfeldern mit der Maus und das Eingeben der Werte mit der Tastatur diesen Eingabevorgang erheblich verlangsamt.
Da diese Grids den größten Anteil der Oberfläche und somit der Schnittstelle zum Benutzer darstellten konzentrierte sich die Suche nach einem geeigneten Entwicklungstool also auf solche, welche die vom Entwicklungsteam gewünschten Grids wie gewünscht bereitstellen und diese auch leicht einsetzbar sind.

Das Umfeld der möglichen Realisierungswege wurde so auf drei Entwicklungsumgebungen reduziert:

- Visual Basic 6.0
- Visual C++
- Borland Delphi 5.0

Visual Basic 6.0
Gegen Visual Basic 6.0 sprachen mehrere Punkte. Zum einen entspricht die mögliche Griddarstellung nicht ganz den Vorstellungen des Entwicklungsteams, zum zweiten greifen in Visual Basic erstellte

Programme auf Runtime Bibliotheken und DLLs zurück, also ein Punkt der von vorneherein ausgeschlossen werden sollte.

Visual C++

Visual C++ ist von den in der näheren Auswahl stehenden Varianten sicherlich das mächtigste. Doch auch hier sind einige Nachteile zu vermerken. Erstens ist die Griddarstellung wie gewünscht zwar möglich, jedoch bedarf diese einem enormen Programmieraufwand, bei dem auch auf Klassen und Methoden der Microsoft Foundation Classes zurückgegriffen werden müsste.
Zweitens können gerade Visual C++ Programme fehleranfällig sein, da durch die Komplexität des Aufbaues sich leicht Fehler einschleichen können, welche dann schwer wieder nachvollziehbar sind.

Borland Delphi 5.0

Den Zuschlag in diesem Projekt bekam Borland Delphi 5.0. Hier wies die Griddarstellung die gewünschten Eigenschaften auf. Auch die ersten Versuche ein Grid in ein Programm einzubinden gestalteten sich überraschenderweise einfach und unkompliziert. Der Vollständigkeit halber sollte hier erwähnt werden, dass innerhalb des Entwicklungsteams noch keinerlei Erfahrungen im Umgang mit Borland Delphi 5.0 oder Delphi generell vorhanden waren.
Ein weiterer ausschlaggebender Punkt war es, dass so erstellte Programme ohne Runtime Bibliotheken auskommen und so „standalone“ überall verwendet werden können.
Last but not least fiel, neben der leichten Handhabung und der Unabhängigkeit der erstellten Programme, auch noch ins Gewicht, dass der innerhalb der OR – Methodenbank der FHK verwendete Solver „Iterator“ auch in Delphi geschrieben ist.

Mit der Wahl von Borland Delphi 5.0 ergab sich auch die Chance sowohl eine neue Sprache als auch eine neue Entwicklungsumgebung kennenzulernen.

3. Merkmale der erstellten Software

Am Ende des Semesters war es dann vollbracht, es gab einen neuen Solver der den Namen PrimalDual.exe trägt. Welche Merkmale zeichnen diese Software aus?
Zusammengefasst lässt sich sagen, dass alle Vorstellungen und Erwartungen des Entwicklungsteams realisiert bzw. erfüllt wurden. Zusätzlich wurden noch einige „Leckerbissen“ eingebaut. Doch konzentrieren wir uns zuerst einmal auf die Kernfunktionalität.
Der Solver formt ein vom Benutzer eingegebenes Primalmodell in das entsprechende Dualmodell um. Im ersten Schritt der Umformung wird das komplette Dualmodell angezeigt, im Anschluss daran kann per Button eine Vereinfachung dieses Modells durchgeführt werden. Zusätzlich zu dieser Kernfunktionalität verfügt die Software über folgende Merkmale:

- Überprüfung der Eingabe
  Die vom Benutzer eingegebenen Werte werden auf Gültigkeit geprüft. Werden falsche Werte eingegeben wird dies vom System erkannt und der Benutzer darauf hingewiesen.

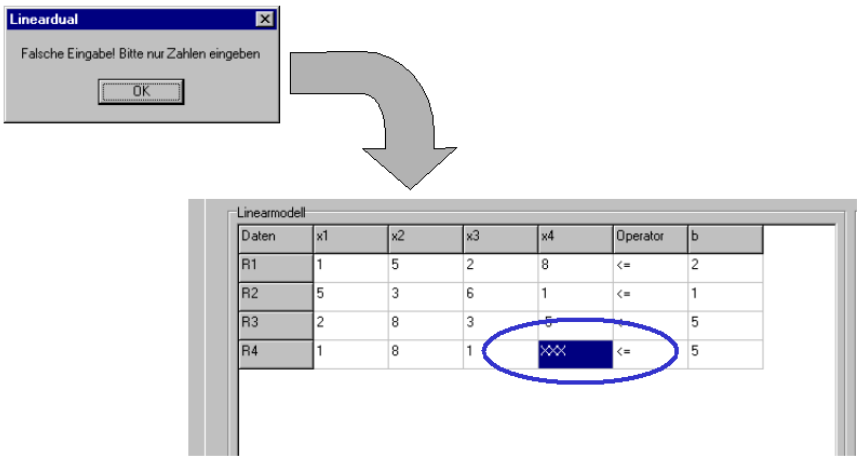


Abbildung 1 – Systemreaktion bei Falscheingaben

- Dateiabhandlung
  Dateien können über das Dateimenü geöffnet, abgespeichert und geschlossen werden. Der beim Abspeichern eingegebene Dateiname wird automatisch mit der Endung pdm versehen.

- Windowsoberfläche
  Die realisierte Oberfläche entspricht dem derzeit gängigen Windowsstil. So findet man hier Registerkarten, Radiobuttons, Grids, Funktionenmenüs, ...
  Dadurch wird dem Benutzer der Einstieg und der Umgang mit der Programm erleichtert, da auf bekannte Mechanismen zurückgegriffen wird.
  Auch ist die Bedienung des Programms zusätzlich zur Maussteuerung auch über Shortcuts möglich. Diese werden dem Benutzer ebenfalls angezeigt.

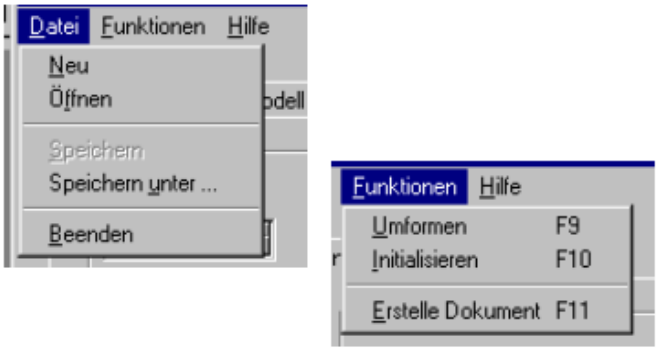


Abbildung 2 – Shortcuts im Programm

Als weitere „Leckerbissen“ wurde eine Wordanbindung und ein Hilfesystem erstellt. Durch die Wordanbindung kann vom Benutzer ein Lösungsdokument erstellt werden, welches die Inhalte der beiden Registerkarten beinhaltet.
Das Hilfesystem setzt sich aus drei Teilen zusammen. Zum ersten führt es den Benutzer in die Theorie ein, dann wird der Solver selbst und seine Verwendung beschrieben und im Anschluss daran wird noch kurz auf das Entwicklungstool und das Entwicklungsteam verwiesen.

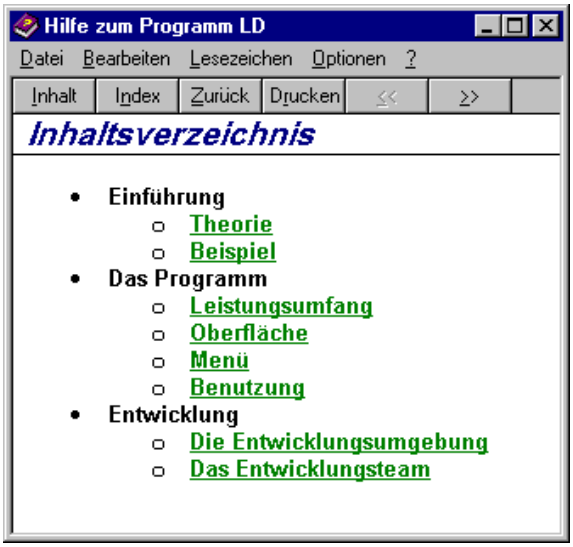


Abbildung 3 – Inhalt des Hilfesystems

Die Textversion der Hilfedatei befindet sich im Anhang dieses Dokument. Sie dient gleichzeitig auch als Anwenderdokumentation, welche die erstellte Software ausführlich beschreibt.

4. Erweiterungsmöglichkeiten

Da sich das ganze auf die Erstellung eines Prototypen beläuft ist dieser auch nicht frei von Fehlern bzw. Erweiterungsmöglichkeiten. In Anbetracht der bereits investierten Zeit und der schon bereitgestellten Funktionalität wurden das Projekt zu diesem Zeitpunkt abgeschlossen. Eine Erweiterung könnte zum Beispiel von kommenden Semestern durchgeführt werden. Hier einige Punkte an welchen angesetzt werden kann:

- Anbindung an einen Solver wie den XA oder den LP Solve.
  Denkbar ist hier sowohl das Primal- als auch das Dualmodell mit Hilfe eines solchen Solvers zu lösen.

- Die Oberfläche passt sich nicht der Größenveränderung mit der Maus an. D.h. das Programm kann zwar verkleinert, oder vergrößert werden, doch hat dies keine Auswirkung auf die Größe der einzelnen Graphikelemente innerhalb der Oberfläche.

Nach ein kleiner Hinweis des Projektteams:
Zum Einarbeiten in Borland Delphi 5.0 eignet sich hervorragend das Buch:
Borland Delphi 5 Kochbuch von den Autoren Doberenz und Kowalski erschienen 2000 im Hanser Verlag

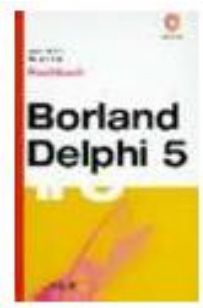


Abbildung 4 - Literaturempfehlung

Auch für Programmierer mit Kenntnissen in Borland Delphi 5.0 ist dieses Buch ein guter Lieferant für Tipps.

5. Das Entwicklungsteam

Das Entwicklungsteam bestand aus:
Stephanie Föhrenbach
Markus Wittekindt

Für beide war der Umgang mit Borland Delphi 5.0 neu und somit auch reizvoll.
Die Aufgaben wurden wie folgt verteilt:

Teammitglied	Aufgaben
Stephanie Föhrenbach	Oberflächengestaltung Wordanbindung Hilfesystem
Markus Wittekindt	Algorithmus Dateiabhandlung

Der zeitliche Umfang des Gesamtprojektes inklusive aller Ausarbeitungen belief sich auf ca. 120 Stunden, wobei sich dieser Aufwand in gleichen Teilen den einzelnen Teammitgliedern zuordnen lässt.
Die relativ hohe Stundenzahl ist auch mit darin begründet, dass zu Beginn erst eine Einarbeitung in das Entwicklungstool stattfinden musste. Zudem wurde während der Realisierung vieles ausprobiert, da es interessant ist sich in eben dieser Entwicklungsumgebung zu bewegen und die angebotenen Fähigkeiten und Möglichkeiten auszuloten.

Trotz dieses hohen Zeitaufwandes hat der „Spaßanteil“ bei weitem die frustrierenden Erlebnisse übertroffen, so dass jetzt am Ende zufrieden auf die Entwicklungsphase zurückgeblickt werden kann. Beide Teammitglieder haben viel Neues dazugelernt und auch einmal die Methoden des Operation Research aus einem anderen Blickwinkel betrachten können.

6. Anhang

1. Gedruckte Textversion der Hilfedatei
2. Datenträger auf welchem sich befindet:
    - Erstelltes Programm (inklusive Hilfesystem)
    - Elektronische Form der Textversion der Hilfedatei
    - Diese Ausarbeitung in elektronischer Form
    - Vollständige Quellcodedateien