

# Visualisierung von Optimallösungen logistischer Probleme

Matthias Forster\*

## Zusammenfassung

Viele Optimierungsprobleme aus der Logistik lassen sich mit Graphen und Netzwerken visualisieren. Bilder von Materialflüssen zeigen die Struktur eines Modells in geeigneter Weise. Das leistungsfähige Optimierungssystem GAMS (General Algebraic Modeling System) kann große Modelle lösen. GAMS stellt aber kein Werkzeug für das Zeichnen von Materialflüssen zur Verfügung. Der Autor hat das Programm Graph.gms implementiert, das als Tool zur Modellvisualisierung von GAMS aus verwendet werden kann.

## Abstract

Many optimisation problems from logistics can be visualised using graphs and networks. Pictures of material flows show the structure of a model in a suitable way. The powerful optimisation system GAMS (General Algebraic Modeling System) is able to solve big models. GAMS provides no tool that visually presents material flows. The author's program Graph.gms can be used for visualising models from GAMS.

## 1. EINLEITUNG

Viele Fragestellungen der Logistik lassen sich als Optimierungsprobleme formulieren. Das algebraische Formulieren von Modellen unterstützen Systeme wie GAMS (General Algebraic Modeling System) oder AMPL (A Mathematical Programming Language). Mithilfe von GAMS oder AMPL kann man leistungsfähige Optimierer für alle relevanten Modelltypen nutzen: lineare Programme, gemischt-ganzzahlige Probleme, nichtlineare und gemischt-nichtlineare Probleme lassen sich in übersichtlicher Form eingeben und mit einem geeigneten Optimierer lösen (AMPL 2014, GAMS 2014). Die Rechenergebnisse werden in Listenform ausgegeben, d.h. im einfachsten Fall als Liste der Variablen mit den zugehörigen Optimalwerten.

In der Regel haben algebraische Modelle aus der betrieblichen Praxis oder der Logistik 5.000 Variablen und mehr. Eine sehr übersichtliche Darstellungsform von Modell und Ergebnissen erhält man, wenn z.B. Materialflüsse grafisch präsentiert werden. Die Visualisierung des Problems erleichtert

die Kommunikation mit den Entscheidungsträgern, da die wesentlichen Inhalte des Modells ohne Variablen, Gleichungen und Ungleichungen vermittelt werden. Beispiele für die grafische Darstellung zahlreicher Modelle findet man in Ahuja et al. (1993). Die bildhafte Darstellung ist gleichzeitig Grundlage für die Präsentation der Optimallösung. So ist die einheitliche Form des Kommunizierens von Problem und Lösung gesichert. Historisch wurden GAMS und AMPL als reine Modellformulierungssprachen ohne grafische Ausgabe konzipiert und realisiert. Heute kann GAMS Ergebnisse grafisch ausgeben, aber nur als Balkendiagramme, Kreisdiagramme und Ähnliches. Das Zeichnen von Diagrammen für Flüsse, Prozesse, Netzwerke usw. ist nicht möglich. Grundlage der Modellformulierung in der Sprache GAMS bilden Mengen und Relationen. Grundlage der hier vorgestellten Visualisierung sind gerichtete Graphen, also Objekte aus einer endlichen, nichtleeren Menge von Knoten und einer Menge von (gerichteten) Kanten. In Abbildung 1 sind die Knoten des Beispiels als Ellipsen dargestellt. Die Kanten verbinden

je zwei Knoten und sind als Pfeile zu erkennen. Der Autor hat das Programm Graph.gms in der Sprache GAMS implementiert. Mit diesem Programm kann man aus einem GAMS-Modell einen Graphen erzeugen, der das Problem geeignet visualisiert. Das vorhandene GAMS-Modell ist nur durch wenige GAMS-Anweisungen zum Erzeugen des Graphen zu ergänzen. Das Prinzip soll am Beispiel eines Modells aus der Distributionsplanung vorgestellt werden. Zum Zeichnen des Graphen wird die Software yEd genutzt. Das Programm yEd ist ein Graphen-Editor und ein Werkzeug zum geeigneten Darstellen und Untersuchen von Graphen, Netzwerken und Diagrammen wie z.B. Organigrammen, Flussdiagrammen, semantischen Netzen usw. Der Graphen-Editor yEd kann die Daten im sogenannten GML-Format einlesen (Graph Modelling Language). GML-Dateien sind normale ASCII-Dateien, welche den Graphen in einer hierarchisch aufgebauten Weise repräsentieren. Das Programm Graph.gms schreibt den Graphen zu einem gegebenen GAMS-Modell in eine GML-Datei, die dann yEd in ein Bild verwandelt (yWorks 2014).

\* korrespondierender Autor

## 2. EIN DISTRIBUTIONSPROBLEM

Die Visualisierung sei an einem Beispiel aus der Distributionsplanung veranschaulicht: An mehreren Fabrikstandorten (Fab) wird jeweils eine Anzahl von Produkten (Prd) hergestellt. Diese Produkte sind an Regionalläger (ReL) zu liefern, sodass deren Nachfrage befriedigt wird. Das Ziel ist die Minimierung der gesamten Transportkosten. In der Sprache GAMS wird das Distributionssystem mithilfe von Mengen und Relationen dargestellt wie folgt:

```
set Fab/F1, F2/, ReL/R1, R2, R3/,
    Prd/P1, P2, P3, P4/;
```

```
set Produktion (Fab, Prd)/F1.(P1, P2,
    P3), F2.(P3, P4)/;
```

```
set RegLager (ReL, Prd)/R1.(P1,P2), R2.
    (P1, P2, P3, P4), R3(P3, P4)/;
```

Die GAMS-Anweisungen sind so zu lesen: Basis der Modellformulierung sind die Mengen  $Fab = \{F1, F2\}$  der beiden Fabriken,  $ReL = \{R1, R2, R3\}$  der drei Regionalläger,  $Prd = \{P1, P2, P3, P4\}$  der vier Produkte und die Relationen Produktion, einer Teilmenge von  $Fab \times Prd$ , und RegLager, einer Teilmenge von  $ReL \times Prd$ . Die Produktionsknoten bilden eine echte Teilmenge von  $Fab \times Prd$ , weil nicht jede Fabrik alle Produkte herstellen kann. Bei der Definition von Relationen in GAMS hat die Punktnotation eine besondere Bedeutung: „F1.P1“ ist hier ein geordnetes Paar, für das in der Mathematik die Schreibweise  $(F1, P1)$  üblich ist. Und „F1.(P1, P2, P3)“ steht in GAMS für die Menge  $\{(F1, P1), (F1, P2), (F1, P3)\}$ .

Abbildung 1 zeigt das Distributionssystem: Die Regionalläger R1, R2 und R3 haben Bedarfe der Güter P1, P2, P3 und P4. Diese Produkte werden an den Standorten F1 und F2 hergestellt und über geeignete Distributionswege an die Regionalläger geliefert. Ziel ist die Minimierung der Summe aus Produktions-, Transport- und Lagerkosten. Abbildung 1 stellt das Distributionssystem mit allen Produktions- und Liefermöglichkeiten dar. Fabrik F1 z. B. kann die Produkte P1, P2 und P3 herstellen. An den Orten R1 und R2 wird Produkt P1 nachgefragt. Daher gehen zwei Pfeile von F1P1 aus (nach R1P1 und nach R2P1). Die Regionalläger führen

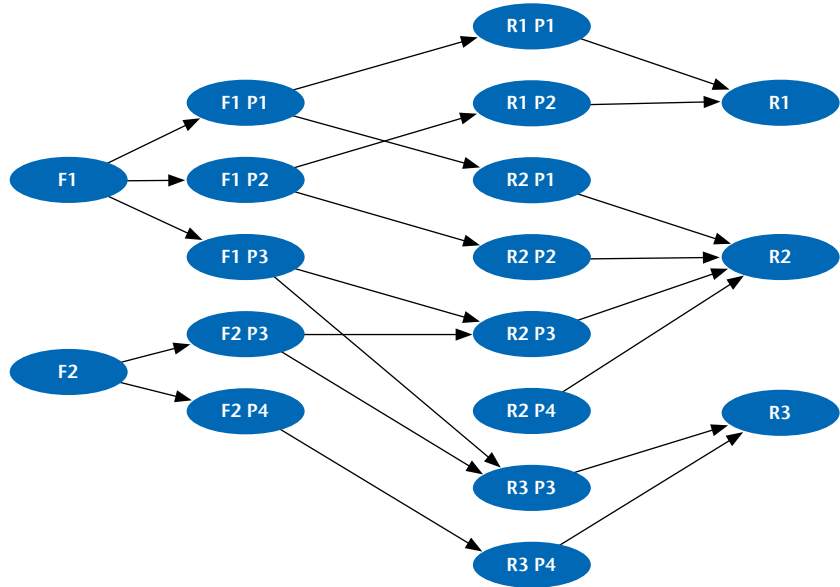


Abb. 1) Distributionssystem (Fabriken F, Produkte P und Regionalläger R).

unterschiedliche Sortimente. Daher ist RegLager eine echte Teilmenge von  $ReL \times Prd$ . In einem vollständigen GAMS-Modell sind nach den set-Anweisungen die Variablen zu deklarieren und die Nebenbedingungen (Gleichungen und Ungleichungen) zu formulieren. Für die Visualisierung haben die Variablen und Nebenbedingungen keine Bedeutung. Die wesentliche Struktur des Modells ist mit den Mengen und den Relationen gegeben. Die Eingabe der Variablen und Nebenbedingungen wird hier nicht behandelt.

Die Knoten- und Kantenmengen des Graphen der Abbildung 1 werden als weitere Mengen definiert. Zuerst sind die Menge der Knotentypen und die Menge der Knoten zu definieren. Das Zeichen \* bezeichnet in GAMS die Obermenge aller sinnvoll definierbaren Mengen, d. h. an der Stelle \* darf jedes Element stehen:

```
set Typ/TFab, TReL, TProduktion,
    TRegLager/;
```

```
set Knoten (Typ, *, *)/TFab.(F1, F2) .o,
    TReL.(R1, R2, R3).o, TProduktion.(F1.
    P1, F1.P2, F1.P3, F2.P3, F2.P4), TRegLa-
    ger.(R1.P1, R1.P2, R2.P1, R2.P2, R2.P3,
    R2.P4, R3.P3, R3.P4)/;
```

In diesem Beispiel ist die Knotenmenge eine dreistellige Relation mit dem Typ an erster Stelle. Abhängig vom Typ

folgt die Fabrik bzw. das Regionallager bzw. eine Kombination aus Fabrik und Produkt oder aus Regionallager und Produkt entsprechend Abbildung 1. Das „o“ in dem Ausdruck „TFab.(F1, F2).o“ bedeutet, dass die dritte Stelle im Falle der Fabriknoten keine Bedeutung hat. Mithilfe der Knotenmenge lassen sich mit wenigen Befehlen die Knoten nummerieren und die x- und y-Koordinaten zum Zeichnen des Graphen berechnen. Das setzt aber Kenntnisse der Sprache GAMS voraus und wird hier nicht im Detail behandelt. Die Kantenmenge wird als Teilmenge von  $Knoten \times Knoten$  definiert:

```
set Kanten(Knoten, Knoten);
```

Die Kanten sind natürlich nicht einzeln einzugeben. In GAMS lassen sich die logischen Ausdrücke verwenden, die man aus jeder höheren Programmiersprache kennt. Die Menge der Kanten ist mit logischen Ausdrücken leicht zu definieren und wird automatisch generiert. Die Knoten, Kanten und weitere Daten (x-Koordinaten, y-Koordinaten, Werte von Modellvariablen) werden als Parameter an das Programm Graph.gms übergeben. Das Programm schreibt eine GML-Datei (Graph Modelling Language), die alle Informationen des Graphen enthält und von dem Programm yEd gelesen werden kann: yEd ist ein frei verfügbares Visualisierungsprogramm zum

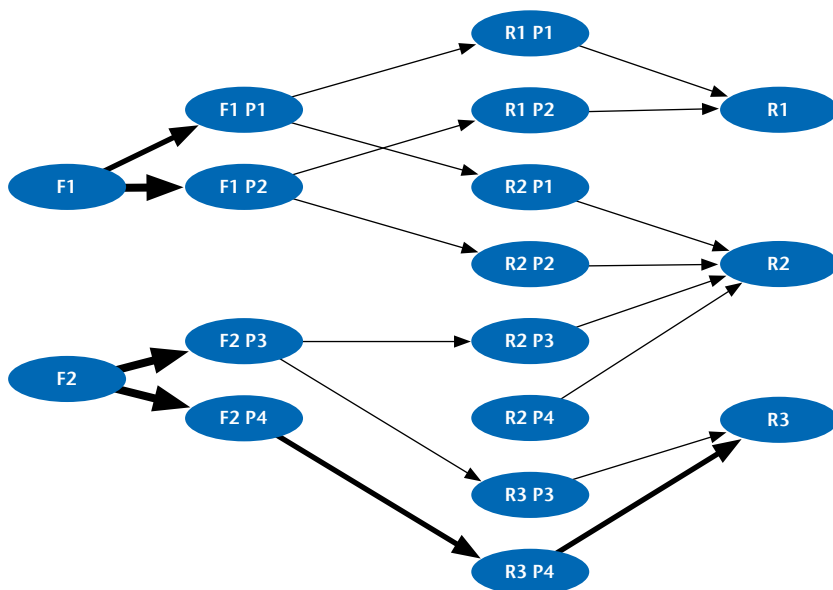


Abb. 2) Lösung (optimiertes Distributionssystem).

Zeichnen von Graphen, die gegebenenfalls auch sehr groß sein können. Es kennt zahlreiche Darstellungsoptionen, eine Zoom-Funktion und Analysemethoden, um Strukturen in Graphen zu finden. Die Abbildungen 1 und 2 wurden mit yEd hergestellt.

### 3. DIE VISUALISIERUNG DER LÖSUNG

Nach dem Aufrufen eines geeigneten Optimierers und der Bestimmung der Optimalwerte aller Entscheidungsvariablen folgt die Visualisierung der Lösung. Die Lösung wird wie das Problem als Graph mit yEd gezeichnet. Abbildung 2 zeigt den Graphen der Lösung. Im Graphen der Lösung sind nur Kanten enthalten, auf denen Material transportiert wird bzw. die eine echt positive Produktmenge symbolisieren. Die Anweisungen für das Erzeugen des „Lösungs-Graphen“ unterscheiden sich nicht von den Anweisungen zum Erzeugen des „Problem-Graphen“. Nur eine weitere Regel ist zu beachten: Die Kante zu der Variablen  $x$  ist nur dann zu zeichnen, wenn  $x > 0$  ist! Den Wert einer Variablen symbolisiert in Abbildung 2 die Breite der entsprechenden Linie. Das Bild zeigt, dass Fabrik F2 viel von den Produkten P3 und P4 herstellt und die Produktionsmengen auf die Regionalläger R2 und R3 aufteilt. Mehrperiodenmodelle sind nicht leicht zu zeichnen. Modelle mit mehreren

Planungsperioden haben sehr viele Variablen, Knoten und Kanten. Der vollständige Graph eines Mehrperiodenmodells könnte sehr unübersichtlich werden. Durch geeignete Aggregation der Lösung bekommt man auch im Falle eines Mehrperiodenmodells eine gute Darstellung. Im Beispiel hätte jeder Knoten einen weiteren Index  $t$  für die Zeit mit  $t = 1, 2, \dots, T$ . Die Variable  $x(Fab, ReL, Prd, t)$  würde in der Optimallösung angeben, welche Menge des Erzeugnisses  $Prd$  die Fabrik  $Fab$  in Periode  $t$  an das Regionallager  $ReL$  zu liefern hat. Eine Möglichkeit der Aggregation ist in diesem Fall die Berechnung der Summe

$$xsum(Fab, ReL, Prd) = \sum(t, x(Fab, ReL, Prd, t));$$

Mit dieser einzeiligen GAMS-Anweisung hat man eine neue Variable  $xsum(Fab, ReL, Prd)$ , die auch im Graphen des Einperiodenmodells vorkommt. Man kann daher den Graphen des Einperiodenmodells weiter verwenden. Aggregationen der Lösung sind bei großen Modellen aus der spezifischen Fragestellung ableitbar. Oder man zerlegt die Lösung in  $T$  Graphen, von denen jeder eine Periode  $t$  darstellt. In diesem Falle erscheinen die Lagerbestände nicht im Bild. Eigenschaften der Lösung können auch durch Farben symbolisiert werden. Beispiel: Wenn die Produktionskapazität einer Anlage

zu mehr als 90% ausgelastet ist, dann kann das eine rote Kante symbolisieren, zwischen 50% und 90% grün und unter 50% blau. Die Visualisierung mit Graphen ist sicher schneller zu interpretieren als eine Liste von Zahlen. Das Programm Graph.gms kann man vom Autor auf Anfrage erhalten. Die Ausgabe der Bilder erfolgt mit dem Programm yEd, das im Internet kostenlos zur Verfügung steht (yWorks 2014). Ein kostenloses Demo-System von GAMS ist im Internet unter [www.gams.com](http://www.gams.com) zu finden (GAMS 2014).

### 4. AUSBLICK

Das Werkzeug zur Visualisierung von Modellen und Lösungen wird zur Zeit nur für die Ausgabe genutzt. Im Prinzip könnte man auch Eingabefelder in den Graphen „einbauen“. Der Nutzer hätte dann die Möglichkeit, Modellparameter zu ändern und die Auswirkungen auf die Lösung zu untersuchen. Lineare Programme sind in der Regel nach wenigen Sekunden gelöst. Ziel könnte die interaktive Optimierung mit grafischer Lösungsausgabe sein.

### LITERATUR

- Ahuja RK et al. (1993) Network Flows. Theory, Algorithms, and Applications, 3rd edn. Prentice Hall, Upper Saddle River, NJ. ISBN: 978-0-13-617549-0
- AMPL (2014) A Mathematical Programming Language. <http://ampl.com>. Accessed 10 Sep 2014
- GAMS (2014) General Algebraic Modeling System. <http://www.gams.com/index.htm>. Accessed 10 Sep 2014
- yWorks (2014) yEd Graph Editor. <http://www.yworks.com/en/products/yfiles/yed>. Accessed 10 Sep 2014

### AUTOR

Prof. Dr. rer. pol. Matthias Forster  
Technische Hochschule Wildau  
[matthias.forster@th-wildau.de](mailto:matthias.forster@th-wildau.de)