Operations Research am Beispiel des Einsatzes von Generatoren

Reduktion des Aufwands für die Modellmodellierung mithilfe moderner Modellierungssprachen

Berechnungen im Operations Research basieren in der Regel auf grossen Datenmodellen. Während die Berechnungen auf Grund des Einsatzes immer leistungsfähigerer Rechner in immer kürzerer Zeit durchgeführt werden können, kann sich die Modellformulierung und -wartung über Wochen erstrecken. Daher ist gerade bei grösseren Modellen der Einsatz von geeigneten Modellierungssprachen und von Instrumenten der Datenverwaltung, der Modellverifikation, der automatischen Modelldokumentation, sowie von Instrumenten, die den gesamten Lebenszyklus und die Evolution eines Modells unterstützen, unerlässlich. Am Beispiel des Einsatzes von Generatoren in einem Elektrizitätswerk wird im vorliegenden Beitrag die an der Universität Freiburg entwickelte Modelliersprache LPL[1] vorgestellt.

Operations Research[2] hat sich – besonders, wenn es der mathematischen Optimierung nahe steht – in den letzten Jahrzehnten hauptsächlich auf die mathematischen Lösungsalgorithmen konzentriert. Zusammen mit immer leistungsfähigeren Computern wurde es so erst möglich, grosse Probleme zu lösen.

Tony Hürlimann

Allerdings wurden Fragen des Modellmanagements, also der Modellformulierung, -bildung und -wartung, lange stiefmütterlich behandelt und sind immer noch relativ unterentwickelt. Vor allem in der praktischen Anwendung des Operations Research, wo grössere Modelle unumgänglich sind, ist der Einsatz von Instrumenten des Modellmanagements unerlässlich. Wenn die Lösung eines Modells auf einem Grosscomputer höchstens Minuten dauert, die Modellformulierung und -wartung sich aber über Wochen, ja Monate erstreckt, wie das durchaus üblich ist, so besteht hier ein krasses Missverhältnis. Der Einsatz von Operations-Research-Methoden in der Praxis hängt damit direkt von der Investition ab, die es braucht, um solche Modelle zu erstellen und zu warten.

Da ein Modell nicht von Anfang an in einer Form vorliegt, wie sie der Lösungsalgorithmus verlangt, braucht es für die praxisorientierte Anwendung des Operations Research mehr als nur effiziente Lösungsalgorithmen. Solche Einsichten haben sich in der letzten Zeit nur zögernd durchgesetzt. Schon 1970 wurden effiziente und mächtige Matrix- and Report-Generatoren entwickelt. Allerdings arbeitet man mit diesen Instrumenten typischerweise so, dass der Modellierer ein Computerprogramm schreibt, welches das Modell in eine vom Matrix-Generator lesbare Form bringt. Wird das Modell geändert, muss auch dieses Übersetzungsprogramm neu geschrieben oder abgeändert werden. Ein solches Vorgehen verlangt vom Modellierer nicht nur Kenntnisse in der Programmierung, sondern ist auch mit einem erheblichen Aufwand verbunden. Zudem liegt das Modell oft als prozedurales Programm vor, das eher vom Programmierer als vom Modellierer interpretiert werden kann. Dadurch wird die eigentliche Struktur des Problems undurchsichtig.

Eine Modellierungssprache als Instrument einer Modellformulierung umgeht diese Nachteile, da sie eine Formulierung erlaubt, die dem Modellentwickler näher steht. Die Modellierungssprache ist sicher nur ein – wenn auch wichtiger – Bestandteil einer Modellumgebung. Dazu gehören nämlich auch Instrumente der Datenverwaltung eines Modells, der Modellverifikation, der automatischen Modelldokumentation, sowie Instrumente, die den gesamten Lebenszyklus und die Evolution eines Modells unterstützen [1].

Am Departement für Informatik der Universität Freiburg (Schweiz) wurde die Modellierungssprache LPL entwickelt, die inzwischen bereits bei Grossfirmen wie Holcim (strategischen Standortplanung) oder der Bank Wegelin (z.B. Portfolio-Optimierung) im praktischen Einsatz steht. Sie ist bestens geeignet, um grosse mathematische Modelle mit Tausenden von Variablen und Beschränkungen konzise abzubilden und zu warten. Sie besitzt einen deklarativen und einen algorithmischen Sprachteil – enthält also auch alle Elemente einer (prozeduralen) Programmiersprache – sowie eine offene Schnittstelle zu den meisten Lösungsalgorithmen und zu Datenbanken für das Daten-Handling.

Ein in LPL beschriebenes Modell besteht aus einer einfachen Syntax – ähnlich wie die mathematische Notation mit indexierten Ausdrücken. Mit einem eingebauten mächtigen Indexmechanismus können Modellstruktur und Modellgrösse sehr flexibel, d.h. unabhängig von den Datenmengen, gestaltet werden.

Dank der Aufspaltung des Modells in Modellstruktur und Modelldaten ist die

Tageszeit-Zone	Leistung [MW]
Mitternacht bis 6 Uhr	15 000
6 – 9 Uhr	30 000
9 – 15 Uhr	25 000
15 – 18 Uhr	40 000
18 Uhr bis Mitternacht	27 000

Tabelle I Datentabelle zum Stromverbrauch

Formulierung der Modellstruktur für kleine und grosse Modelle identisch, die Modellgrösse wird allein durch die Datenmenge bestimmt, die als Input von unabhängigen Textdateien oder aus Datenbanken automatisch eingelesen werden können. Verschiedene Funktionen in der Modellierumgebung erlauben ausserdem das Modelldebugging, automatische Modelldokumentation, sowie Datenbank-Generierung.

Bei LP/MIP-Modellen[3] wird der MPS-Standard-Code[4] schnell produziert, und die Schnittstelle zu kommerziellen LP/MIP-Lösungsalgorithmen (CPLEX[5], Xpress[6] u.a.) ist offen und kann vom Benutzer selber definiert werden.

Nachfolgend sollen die Grundideen von LPL anhand eines einfachen Modells vorgestellt werden.

Modellbeispiel aus der Elektrizitätswirtschaft: Einsatzplanung von Stromgeneratoren

In einem Elektrizitätswerk müssen ständig Stromgeneratoren ein- und ausgeschaltet werden, um die stündlichen Schwankungen des Stromverbrauchs zu decken. Dabei wird ein Tag in 5 Zeitzonen eingeteilt, in welchen ein konstanter, geschätzter Stromverbrauch angenommen wird (Tabelle I).

Weiter soll vereinfachend angenommen werden, dass im Elektrizitätswerk drei verschiedene Generatorentypen im Einsatz sind: Zwölf Generatoren vom Typ 1, zehn vom Typ 2 und fünf vom Typ 3. Während des Betriebs liefert jeder Generatortyp eine minimale Strommenge, kann aber ein bestimmtes Maximum nicht übersteigen. Die Betriebskosten über dem Minimum sind höher als auf dem Minimum. Zudem müssen Anschaltkosten in Betracht gezogen werden. Die Daten sind in der Tabelle II zusammengestellt.

Um die geschätzte Nachfrage jederzeit decken zu können, ist vorgesehen, dass eine plötzliche Zunahme von 15% der Nachfrage abgefangen werden kann, ohne dass neue Generatoren in Betrieb genommen werden müssen.

Die Frage lautet nun: Welche Generatoren müssen zu jeder Tageszeit in Betrieb sein, bzw. in Betrieb genommen werden, wenn die Betriebskosten zu minimieren sind? Ausserdem interessiert uns die Frage, welches Grenzkosten des Stroms zu jeder Tageszeit sind.

Dieses Problem kann als ein lineares Optimierungsmodell folgendermassen formuliert werden.

Lineares Optimierungsmodell

Gegeben sind die *i* Generatortypen (*i* = 1, ..., N mit N = 3) und die *t* Zeitzonen (*t* = 1, ..., T mit T = 5, zyklisch). Daneben sind liegen die folgenden Daten vor.

- m_i Minimaler Betriebsoutput pro Generatortyp *i* [MW]
- M_i Maximale Kapazität des Generatortyps *i* [MW]
- C_i Minimale Betriebskosten pro Generatortyp *i* [CHF/h]
- E_i Zusätzliche Betriebskosten pro MW über dem Minimum je Typ *i* [CHF/(h·MW)]
- F_i Anschaltkosten pro Generatortyp *i* [CHF]
- L_i Anzahl Generatoren des Typs *i* [-]
- D_t Geschätzte Stromnachfrage in der Zeitzone *t* [MW]
- N_t Länge der Zeitzone *t* [h]

Als Unbekannte liegen schliesslich folgende Grössen vor:

- $n_{i,t}$ Anzahl zum Zeitpunkt *t* in Betrieb stehender Generatoren vom Typ *i* [-]
- $s_{i,t}$ Anzahl zum Zeitpunkt *t* gestartete Generatoren vom Typ *i* [-]
- $x_{i,t}$ Stromproduktion von Generatoren des Typs *i* zum Zeitpunkt *t* [MW]

Die zu minimierende Funktion ist in Formel 1 dargestellt. Minimiert werden dabei die Kosten, d.h. die Summe aus zusätzlichen Betriebskosten über dem Minimum, minimalen Betriebskosten und Anschaltkosten.

$$\sum_{i=1}^{N}\sum_{t=1}^{T}(N_t\cdot(E_i\cdot(x_{i,t}-m_i\cdot n_{i,t})+C_i\cdot n_{i,t})+F_i\cdot s_{i,t}) \qquad (1)$$

Für die Berechnung müssen folgende Restriktionen berücksichtigt werden:

– Restriktionen hinsichtlich der Nachfragedeckung: die gesamte Produktion sämtlicher Generatoren muss für jedes Zeitintervall mindestens der geschätzten Nachfrage entsprechen (Formel 2).

$$\sum_{i=1}^{N} x_{i,t} \geq D_t \qquad (2)$$

mit $t = 1,\ldots,T$

– Restriktionen hinsichtlich der Produktionsmarge: die maximale Kapazität sämtlicher in einem bestimmten Zeitintervall in Betrieb stehenden Generatoren muss mindestens 15% über der geschätzten Nachfrage liegen (Formel 3).

$$\sum_{i=1}^{N} M_i\cdot n_{i,t} \geq \frac{115}{100}\cdot D_t \qquad (3)$$

mit $t = 1,\ldots,T$

– Restriktionen hinsichtlich der Stromproduktion: in einem bestimmten Zeitintervall ist die Stromproduktion der einzelnen in Betrieb stehenden Generatoren eines Generatortypen beschränkt durch ihre Minimalproduktion und durch die maximale Kapazität (Formel 4).

$$m_i\cdot n_{i,t} \leq x_{i,t} \leq M_i\cdot n_{i,t}$$

mit $i = 1,\ldots,N,\ t = 1,\ldots,T$ (4)

– Restriktionen hinsichtlich der Anzahl gestarteten Generatoren: die Anzahl gestarteter Generatoren vom Typ *i* im Zeitintervall *t* entspricht mindestens der Anzahl in Betrieb stehenden Generatoren im gleichen Zeitintervall abzüglich jener des vorangehenden Zeitintervalls (Formel 5).

$$s_{i,t} \geq n_{i,t} - n_{i,t-1}$$

mit $i = 1,\ldots,N,\ t = 1,\ldots,T$ (5)

– Restriktionen hinsichtlich der Anzahl in Betrieb stehender Generatoren: die maximale Anzahl der in Betrieb ste-

	Typ 1	Typ 1	Typ 3
Minimale Kapazität [MW]	850	1250	1500
Maximale Kapazität [MW]	2000	1750	4000
Minimale Betriebskosten [CHF/h]	1000	2600	3000
Zusätzliche Betriebskosten pro MW über dem Minimum [CHF/(h·MW)]	2.–	1.30	3.–
Anschaltkosten	2000	1000	500
Anzahl Generatoren	12	10	5

Tabelle II Daten zu den Stromgeneratorentypen

```
MODEL Strom "Stromproduktion";
  SET
    i                    "Generatorentypen";
    t STRING tName      "Zeitzonen";
  PARAMETER
    m{i}          "minimaler Betriebsmenge pro Generatortyp i";
    M{i}          "maximale Kapazitaet des Generatortyps t";
    C{i}          "min. Betriebskosten/Std pro Generatortyp i";
    E{i}          "Extra Betriebskosten/GW/Std. ueber dem Minimum";
    F{i}          "Anschaltkosten pro Generatortyp i";
    L{i}          "Anzahl von Generatoren des Typs i";
    D{t}          "geschaetzte Stromnachfrage zur Zeit t";
    N{t}          "Laenge der Zeitzone t (in Stunden)";
  VARIABLE
    x{i,t}                "Stromproduktion des Typs i zur Zeit t";
    INTEGER n{i,t}      "Anzahl Generatoren vom Typ i in Betrieb zur Zeit t";
    INTEGER s{i,t}      "Anzahl gestartete Generatoren vom Typ i zur Zeit t";
  CONSTRAINT
    Nachfrage{t}: SUM{i} x >= D;
    Extrakapazitaet{t}: SUM{i} M*n >= 1.15*D;
    Output{i,t}: m*n <= x <= M*n;
    Gestartet{i,t}: s >= n - n[i,(#t+t-2)%#t+1];
    ObereSchranke{i,t}: n <= L >= s;
  MINIMIZE Kosten: SUM{i,t} (N*E*(x-m*n) + N*C*n + F*s);
  WRITE n, s, x, Kosten;

  MODEL DATA aData "Daten für fünf Tagesperiode und drei Generatorentypen";
    READ FROM 'strom.dat' '%:Tabelle';
    READ '%1' : ROW{t} (t , tName , D, N);
    READ '%2' : ROW{i} (i , m , M , C , E , F , L);
  END
END
```

Quelle: Tony Hürlimann

Bild 1 LPL-Code für das Stromgeneratoren-Modell

henden Generatoren kann die Anzahl der vorhandenen Generatoren des gleichen Typs nicht überschreiten (Formel 6).

$$n_{i,t} \leq L_i \quad \text{mit } i = 1,\ldots,N,\ t = 1,\ldots,T \qquad (6)$$

– Restriktionen hinsichtlich der Variablen x, s, und n: die Variablen x_{it}, s_{it} und n_{it} dürfen keine negativen Werte annehmen und müssen ganzzahlig sein (Formel 7).

$$\begin{aligned} x_{i,t} &\geq 0 \quad \text{und ganzzahlig} \\ s_{i,t} &\geq 0 \quad \text{und ganzzahlig} \\ n_{i,t} &\geq 0 \quad \text{und ganzzahlig} \end{aligned} \qquad (7)$$

Da die Zeitzonen zyklisch definiert sind, d.h. die Vorperiode der ersten Zeitzone die letzte Zeitzone des Vortages ist, und die nachfolgende Periode der letzten Zeitzone die erste Zeitzone des nächsten Tages ist, gilt die Restriktion $s_{i,t} \geq n_{i,t} - n_{i,t-1}$ auch für $t = 1$, die gegeben ist als $s_{i,1} \geq n_{i,1} - n_{i,T}$.

Diese Struktur dieses Modells kann direkt in der LPL-Sprache wiedergegeben werden. Das LPL-Modell, welches dieses Modell repräsentiert, ist in Bild 1 aufgelistet.

Das Modell in Bild 1 besteht aus einer Liste von Deklarationen: Indexmengen, Datentabellen, Unbekannten, einer minimierende Funktion und lineare Beschränkungen. Diese werden in der Modelliersprache durch die reservierten Wörter SET, PARAMETER VARIABLE, CONSTRAINT usw. eingeleitet und durch einen Strichpunkt abgeschlossen. Es ist wichtig zu sehen, dass diese Formulierung vollständig unabhängig von den Daten ist. Die Daten selber sind extern – hier in der Text-Datei strom.dat – definiert (Bild 2). Die Beschreibung ist somit unabhängig von der Grösse des Modells. Werden beispielsweise die Tageszeiten feiner eingeteilt, ändert sich nichts an der Logik des Modells, nur die externe Datenmenge wird vergrössert.

Die LPL-Formulierung zeigt noch einige weitere Merkmale, die sie von der mathematischen Formulierung in Bild 1 unterscheidet: so stehen Kommentare in Hochkommata ("..."). Ferner werden die Indizes nicht tief gestellt sondern geklammert dargestellt, und Zeichen wie beispielsweise das Summen-Symbol Σ sind durch reservierte Wörter wie SUM ersetzt. Schliesslich wird die LPL-Formulierung noch durch drei weitere Anweisungen erweitert:

– mit der Anweisung READ werden die Modelldaten aus externen Dateien (z.B. Datenbanken) eingelesen;
– die Anweisung MINIMIZE bewirkt, dass das Modell gelöst wird;
– durch die Anweisung WRITE werden entsprechende Resultate-Tabellen generiert.

Nach Aufruf des LPL-Compilers und des Lösungsalgorithmus werden die Resultate-Tabellen der drei Variablen n, s, x sowie die berechneten Kosten in eine Datei geschrieben (Bild 3). In der interaktiven Modellierumgebung können diese Daten natürlich auf verschiedene Arten betrachtet werden.

Das Resultat in Bild 3 zeigt, dass in allen Zeitzonen alle 12 Generatoren des Typs 1 und zusätzlich in der ersten Zeitzone 3 Generatoren des Typs 2 in Betrieb sind (n $\{i, t\}$). Auf Grund der Nachfrage müssen in der zweiten Zeitzone 5 Generatoren des zweiten Typ hinzugeschaltet werden und einer in der Zeitzone 4. Ebenso müssen zwei Generatoren des Typs 3 in der Zeitzone 4 hinzugeschaltet werden (s $\{i, t\}$). Der Output x in jeder

```
 (* Datenfile strom.dat zum Modell strom.lpl *)

Tabelle 1 : Stromverbrauchskurve
(* Zeitzonen              geschaetzte Nachfrage   Anzahl
                                  (in Gigawatt)        Stunden
 --------------------------------------------------- *)
 t1 'Mitternacht bis 6 Uhr'           15              6
 t2 '6 - 9 Uhr'                       30              3
 t3 '9 - 15 Uhr'                      25              6
 t4 '15 - 18 Uhr'                     40              3
 t5 '18 Uhr bis Mitternacht'          27              6

Tabelle 2 : Daten zu den Generatorentypen
(*Typ  minimale  maximale  Kosten bei    extra Kosten pro  Start    Anzahl
       Kapa.(MW) Kapa.(MW) min. Betrieb  produzierte MW    kosten   Generatoren
 ------------------------------------------------------------------------- *)
   G1    850      2000         1.0            2000           2.0        12
   G2   1250      1750         2.6            1300           1.0        10
   G3   1500      4000         3.0            3000           0.5         5
```

Quelle: Tony Hürlimann

Bild 2 LPL-Daten für das Stromgeneratoren-Modell

```
n{i,t}      t1         t2         t3         t4         t5
G1          12         12         12         12         12
G2           3          8          8          9          9
G3           0          0          0          2          0

s{i,t}      t1         t2         t3         t4         t5
G1           0          0          0          0          0
G2           0          5          0          1          0
G3           0          0          0          2          0

x{i,t}      t1         t2         t3         t4         t5
G1     10.2000    16.0000    11.0000    21.2500    11.2500
G2      4.8000    14.0000    14.0000    15.7500    15.7500
G3      0.0000     0.0000     0.0000     3.0000     0.0000

Kosten    988.5400
```

Quelle: Tony Hürlimann

Bild 3 Ausgabe und Resultat des Stromgeneratoren-Modells

Zeitzone entspricht gerade der Nachfrage (Daten aus Bild 2), wobei die Generatoren des Typs 1 in der vierten Zeitzone die grösste Kapazität erreichen, die aber noch 2,5 MW unter der maximalen Kapazität der 12 Generatoren liegt. Die Gesamtkosten betragen rund 989 Schweizer Franken.

Abschliessende Bemerkungen

Dieses einfache Beispiel zeigt nur einen kleinen Ausschnitt der vielen Möglichkeiten, die die Software LPL bietet. LPL ist zurzeit die einzige im deutschen Sprachraum entwickelte Modellierungssprache der mathematischen Optimierung[7]. Zu den weiteren Besonderheiten dieser strukturierten mathematische Modellierungs- und Programmiersprache gehört, dass sowohl deklaratives als auch algorithmisches Wissen und automatische Modellierung der Dokumentation unterstützt und sehr konzise formuliert werden können. Weitere Stärken finden sich in der logischen Modellierung und der Verwendung von Indexstrukturen, die es ermöglichen, grosse lineare und nichtlineare mathematische Optimierungsprobleme zu implementieren, zu warten, zu modifizieren und zu dokumentieren[8].

Die Entwicklung von LPL war von Anfang an motiviert durch den praktischen Einsatz von grossen Modellen. Verschiedene LP-Modelle mit 8000 Beschränkungen, 20 000 Unbekannten und einer Matrixbesetzung von 70 000 Elementen werden am Departement für Informatik an der Universität Freiburg (Schweiz) im Auftrage des Bundesamtes für Landesvorsorge gewartet. Da diese Modelle noch Mitte der 80er-Jahren einem Grossrechner zur Lösung übergeben werden mussten, stand für die LPL-Sprache die automatische Produktion des MPS-Codes im Vordergrund. Dank der raschen Entwicklung der Personal Computer und den darauf implementierten Lösungsalgorithmen (CPLEX, Xpress, XA[9] u.a.) ist es heute möglich, solche Modelle lokal auf dem PC innert Sekunden zu lösen. Damit rückt der Modellierungszyklus (Modell ändern – lösen – Resultate generieren) immer mehr in den Vordergrund. Umso wichtiger werden Modellierungswerkzeuge, die es dem Benützer erlauben, grosse Modelle zu manipulieren, zu warten und zu dokumentieren. Die Software LPL unterstützt diesen Zyklus, indem es automatisch einen externen Lösungsalgorithmus aufrufen und die Lösung zur weiteren Verarbeitung übernehmen kann. Der eingebaute Tabellengenerator produziert anschliessend die gewünschten Resultate-Tabellen. Der Modellierer muss sich daher nicht mehr mit den vielen technischen Details befassen, sondern kann sich der eigentlichen Modellierung widmen. Die praktische Erfahrung mit den genannten Modellen hat gezeigt, dass sich der Modellierungszyklus (Modellmodifikationen-Lösen-Resultate) von einigen Stunden oder Tagen auf einige Minuten reduziert hat.

Im Rahmen eines laufenden Forschungsprojektes am Department of Computer Science der Universität Freiburg (Schweiz) werden neue Modellierungsstrukturen in LPL implementiert und getestet; parallel dazu existiert eine stabile, kommerzielle LPL-Version zur Nutzung in praktischen Problemen.

Referenzen

[1] Hürlimann, T.: Mathematical Modeling and Optimization. An Essay for the Design of Computer-Based Modeling Tools. Kluwer Academic Publ., (Applied Optimization 31), 1999

Angaben zum Autor

PD. Dr. **Tony Hürlimann** ist seit 1985 wissenschaftlicher Mitarbeiter und Dozent am Departement für Informatik der Universität Freiburg (Schweiz) und Inhaber der im Jahr 2004 gegründeten Firma Virtual Optima (www.virtual-optima.com) in Freiburg.
Universität Freiburg, Site Regina Mundi,
Rue Faucigny 2, Departement für Informatik,
1700 Fribourg, tony.huerlimann@unifr.ch

[1] LPL: Linear Programming Language. LPL ist eine Modelliersprache, die es erlaubt, grosse Modelle in der üblichen mathematischen Notation zu formulieren.
[2] Operations Research: Suche nach einer bestmöglichen, bzw. optimalen Entscheidung unter Berücksichtigung von Nebenbedingungen.
[3] LP/MIP: Linear Program/Mixed Integer Program; Bezeichnungen aus dem Operations Reaserch für lineare Modelle.
[4] MPS: Standardformat von IBM aus den 70er-Jahren, mit welchem lineare Modell beschrieben werden können.
[5] CPLEX: www.ilog.com; kommerzielle Lösungssoftware.
[6] Xpress: www.dashoptimization.com; kommerzielle Lösungssoftware.
[7] Zu den vielen existierenden Optimierungs-Programmen wie z.B. OPF-Programme (OPF: Optimal Power Flow) bestehen zwei essenzielle Unterschiede: – die Modellformulierung und -struktur ist direkt in der mathematischen Sprache modelliert, transparent und schnell modifizierbar sowie klar von den Daten separiert (nicht in einem längeren Programm-Code mit Daten-Code vermischt); – das Modell kann den schnell verändernden kommerziellen Lösungsalgorithmen zur Lösung (z.B. CPLEX) übergeben werden.
[8] Das Benutzerhandbuch und weitere Dokumente, die auf der Internetseite www.virtual-optima.com verfügbar sind, geben eine detaillierte Beschreibung des Modellierungssystems, das auch eine grafische Benutzeroberfläche besitzt. Eine eingeschränkte, freie Version kann heruntergeladen und getestet werden.
[9] XA: www.sunsetsoft.com; kommerzielle Lösungssoftware.

Operations Research à l'exemple de l'utilisation de générateurs

Réduction du travail de modélisation grâce aux langages modernes de modélisation

Les calculs en Operations Research sont généralement basés sur de grands modèles de données. Tandis que des ordinateurs de plus en plus performants permettent d'effectuer les calculs de plus en plus rapidement, la formulation et l'entretien des modèles peuvent prendre des semaines. C'est pourquoi les modèles plus importants exigent que l'on ait recours à des langages de modélisation adéquats et à des instruments de gestion des données, de vérification et de documentation automatique des modèles ainsi qu'à des instruments assistant tout le cycle de vie d'un modèle et son évolution. A l'exemple de l'utilisation de générateurs dans une centrale électrique, l'article présente le langage de modélisation LPL développé à l'université de Fribourg.

articles spécialisés