

# Parallele Simulation mittels Multithreading bei der optimalen Auslegung regenerativer Energieanlagen

Nikola Thill

Nikola.Thill@TU-Ilmenau.de

Technische Universität Ilmenau, Institut für Automatisierungs- und Systemtechnik  
Postfach 10 05 65, 98684 Ilmenau

## Kurzfassung

Dieser Beitrag beschäftigt sich mit der Kopplung von Dymola und Matlab und der parallelen Simulation mittels Multithreading (MT) bei der optimalen Auslegung regenerativer Energieanlagen. Anhand eines Beispiels zur optimalen Dimensionierung einer hybriden Modellanlage bestehend aus Solarkollektoren, Solarspeicher und Photovoltaikmodulen, wird mithilfe der dafür implementierten graphischen Benutzeroberfläche [3] die Kopplungsmöglichkeit via DDE (Dynamic Data Exchange)-Schnittstelle und die parallele Simulation (mittels MT) während der Optimierung mit der Genetischen-Algorithmen-Toolbox von Pohlheim [5] für diskrete Optimierungsvariablen demonstriert.

## 1 Einleitung

Im Fachgebiet „Dynamik und Simulation ökologischer Systeme“ der Technischen Universität Ilmenau wurde die Modelica-Bibliothek RECOMB (Renewable Energy Components Modelling and OptiMization of Buildings) entwickelt, die unter der Simulationsumgebung Dymola genutzt werden kann [1]. RECOMB stellt eine Vielzahl von Modellen für Energieversorgungskomponenten und Gebäude zur Verfügung, die sich für die Modellierung komplexer Anlagen verwenden lassen [2]. Dabei ist es möglich, Anlagen zu konzipieren, die mit nur einer regenerativen Energieversorgungskomponente betrieben werden und den Restenergie- bzw. Restwärmebedarf mithilfe konventioneller Energieversorgungskomponenten decken oder hybride Anlagen, die ihre Energie aus mehr als einer regenerativen Versorgungskomponente beziehen.

In dem hier vorgestellten Beispiel wird ein Teil des Energie- und Warmwasserbedarfs eines 4-Personen-Haushalts von einer solarthermischen Anlage zur Brauchwassererwärmung und einer Photovoltaikanlage, die mit Überschusseinspeisung arbeitet, gedeckt. Die dafür notwendigen Lastgänge für den Strombedarf ( $3951 \text{ kWh/a}$ ) und den Warmwasserbedarf ( $3158 \text{ kWh/a}$ ), sowie die Komponentenmodelle für Solarkollektor, Warmwasserspeicher und die Photovoltaikmodule stehen in der Dymola-Bibliothek RECOMB zur Verfügung und werden in Dymola zu einem Gesamtanlagenmodell verschaltet.

## 2 Optimale Anlagendimensionierung

### 2.1 Die Programmkopplung

Die Modellierung mit dem Programmpaket Dymola [1] basiert auf den Prinzipien der Objektorientierung. Dafür können Gleichungen in impliziter Form hinterlegt werden, die Dymola automatisch in ein flaches Modell übersetzt. Weiterhin bietet Dymola Schnittstellen zu anderen Programmen an, wodurch die Funktionalität der Modellierung und Simulation komplexer physikalischer Systeme von Dymola genutzt und in andere Programmpakete integriert werden kann. Die Optimierung wird unter Matlab mit der Genetischen-Algorithmus-Toolbox [5] durchgeführt, wobei die einzelnen dafür notwendigen Simulationen unter Dymola aufgerufen werden. Matlab und Dymola kommunizieren dabei über die DDE-Schnittstelle beider Programme. Speziell dafür wird die graphische Benutzeroberfläche DiSiOSS (Dimensioning, Simulation and Optimization of Solar Systems) [3] verwendet, die die Kommunikation erheblich erleichtert. Unter anderem können mit DiSiOSS ausgewählte Trajektorien grafisch dargestellt und der Barwert der zusammengestellten Anlage berechnet werden, wodurch eine Beurteilung der Wirtschaftlichkeit ermöglicht wird. Die genaue Zusammenarbeit der einzelnen Programme zeigt Abbildung 1.

### 2.2 Das Optimierungsproblem

Das hier behandelte Optimierungsproblem besteht aus den Beschränkungen für die Optimierungsvariablen und den Gütekriterien der einzelnen Anlagenkomponenten. Da hier die jeweilige Größe der Komponenten der Anlage variiert wird, handelt es sich um eine diskrete Optimierung der Entwurfparameter. Die Einzelgütekriterien beschreiben die Wirtschaftlichkeit jeder Anlagenkomponente durch das Betriebsergebnis/Barwert. Dieses mehrkriterielle Problem wird mithilfe der Methode der gewichteten Summe von Einzelzielfunktionen (Methode I) aggregiert und erlaubt dem Projektanten somit die Bevorzugung einer bestimmten Technologie, was mit einer einkriteriellen Betrachtung nicht möglich wäre. Dafür wird durch die Variation der Gewichte  $w$  eine Pareto-Front erstellt, die als Entscheidungshilfe dient [4]. Alternativ wird die Formulierung einer anderen Zielfunktion vorgestellt, wobei das mehrkriterielle Problem ebenfalls in ein einkriterielles überführt wird, hier jedoch über die Minimierung eines Teilkriteriums und die Überführung der anderen Teilkriterien in  $\epsilon$ -Ungleichungsbeschränkungen. Dieses Vorgehen wird als Kompromiss-Methode oder Methode der  $\epsilon$ -Beschränkungen bezeichnet (Methode II). Die Pareto-Front kann bei dieser Methode über die Variation der  $\epsilon$ -Beschränkungen konstruiert werden [4]. Allgemein lässt sich das Optimierungsproblem wie folgt beschreiben:

$$\begin{aligned} & \text{minimiere } \underline{J}(\underline{x}) \\ & \text{mit } \underline{g}(\underline{x}) \leq 0 \\ & \text{und } \underline{h}(\underline{x}) = 0, \end{aligned}$$

wobei  $\underline{x} \in Z$ ,  $\underline{J}(\underline{x})$ ,  $\underline{g}(\underline{x})$  und  $\underline{h}(\underline{x}) \in R$  sind. Bezogen auf das hier vorgestellte Beispiel bedeutet das, dass eine Menge von Zielfunktionen mit diskreten Optimierungsvaria-

blen minimiert werden soll. Die einzelnen Größen der Anlagenkomponenten, die Fläche der Solarkollektoren  $A_{ST}$  und der Photovoltaikmodule  $A_{PV}$  und das Volumen des Speichers  $V_{ST}$ , stellen hierbei die Optimierungsvariablen dar. Dem Bewertungskriterium wird eine Barwertberechnung ( $B_W$ ) zugrunde gelegt, dem sich die Berechnung des Betriebsergebnisses  $B_E$  anschließt. Der Wert  $B_E$  gibt Auskunft über den am Ende der Lebensdauer der Anlage zu erwartenden Gewinn und setzt sich aus der Differenz von  $B_W$  und dem Gesamtkapitaldienst  $K_{Dg}$  zusammen. Somit sind  $B_E$  und  $B_W$  identisch, wenn kein Kredit für die Tatigung der Investition benotigt wird. Der Aufbau der Zielfunktion fur Methode I ist den Gleichungen (1)-(5) zu entnehmen. Dabei konkurrieren beide Technologien um ein festes Platzangebot  $A_{max}$ , das in diesem Fall auf  $60 m^2$  festgelegt wird. Untersuchungen haben gezeigt, dass bei der photovoltaischen Stromerzeugung mit uberschusseinspeisung immer die grotmogliche Flache die optimale Losung darstellt. Deshalb entfallt  $A_{PV}$  als Optimierungsvariable und wird ausschlielich aus der Differenz von  $A_{max}$  und  $A_{ST}$  berechnet.

$$\min_{A_{ST}, V_{ST}} J_{ST} = -B_{EST}(A_{ST}, V_{ST}) \quad (1)$$

$$\min_{A_{PV}} J_{PV} = -B_{EPV}(A_{PV}) \quad (2)$$

$$\min_{A_{ST}, V_{ST}} J(A_{ST}, A_{PV}, V_{ST}) = \min_{A_{ST}, V_{ST}} \{w_1 \cdot J_{ST} + w_2 \cdot J_{PV}\} \quad (3)$$

$$\sum_{i=1}^2 w_i = 1, w \in (0, 1) \quad (4)$$

$$A = A_{ST} + A_{PV} = A_{max}, 0 \leq A_{ST} \leq 18m^2, 0.1 \leq V_{ST} \leq 3m^3 \quad (5)$$

$$\min_{A_{ST}, V_{ST}} J_{ST}, \text{ mit } J_{PV} \leq \epsilon \quad (6)$$

Damit wird sichergestellt, dass die Groe fur die Photovoltaikmodule maximal gewahlt wird. Die zu minimierende Zielfunktion und deren Beschrankung fur Methode II ist in Gleichung (6) zu sehen, wobei das Betriebsergebnis fur den solarthermischen Teil der Anlage unter der  $\epsilon$ -Ungleichungsbeschrankung fur das Betriebsergebnis der Photovoltaikanlage minimiert wird. Auch in diesem Fall ist eine Konkurrenz zwischen den Technologien erwunscht und sinnvoll (siehe Gleichung (5)).

### 2.3 Das Optimierungsverfahren

Zur Losung des Optimierungsproblems wird die Genetische-Algorithmen-Toolbox von Pohlheim [5] eingesetzt, die fur den speziellen Fall mit diskreten Optimierungsvariablen eine Option bereitstellt. Diese Option heit *tbx3int* und beinhaltet die optimalen Einstellungen fur die ublichen Teilschritte evolutionarer Algorithmen, wie Mutation, Selektion und Rekombination. Als Abbruchkriterium wird *RunningMean* verwendet, das den aktuellen Zielfunktionswert mit dem Durchschnitt der Zielfunktionswerte der letzten 15 Generationen vergleicht und bei ubereinstimmung abbricht. Pro Generation werden zehn Individuen generiert. Die jeweiligen Anlagengroen mussen in das Dymola-Modell eingesetzt werden,

dieses wird in Matlab übersetzt (DDE-Schnittstelle), simuliert und abschließend die Zielfunktion berechnet. Bei dieser Vorgehensweise bietet sich eine Parallelisierung an, da die Zielfunktionswerte der zehn verschiedenen Konfigurationen unabhängig voneinander bestimmt werden können.

### 3 Parallele Simulation mittels Multithreading (MT)

Für zeitsparendes MT bei der Optimierung müssen mehrere Voraussetzungen gegeben sein. Auf der Hardware-Seite sind ein von Intel angebotener Pentium 4 Prozessor mit Hyper-Threading-Technologie (HTT) und ein HTT-fähiges Mainboard notwendig, auf der Software-Seite vervollständigt ein HTT/MT-fähiges Betriebssystem wie Windows XP Professional die Anforderungen. Um mehrere Prozesse (Threads) gleichzeitig in Matlab auszuführen, ist es erforderlich, ein weiteres Programm einzubinden, das die Thread-Teilung übernimmt, da Matlab diese Funktionalität nicht bietet. Dafür wird eine Java-Applikation [7] in ein ausführbares Java-Archiv übersetzt und in Matlab ausgeführt. In dieser Java-Applikation wird der Haupt-Thread in zwei Teil-Threads geteilt und in jedem die von Dymola bei der Übersetzung generierte Datei im jeweiligen Pfad ausgeführt. Somit ist es möglich die Simulationen, die in jeder Generation notwendig sind, jeweils paarweise parallel auszuführen und dabei Zeit zu sparen. Diese Art von paralleler Abarbeitung wird auch mit Thread-Parallelität oder Makroparallelität bezeichnet [9, 8].

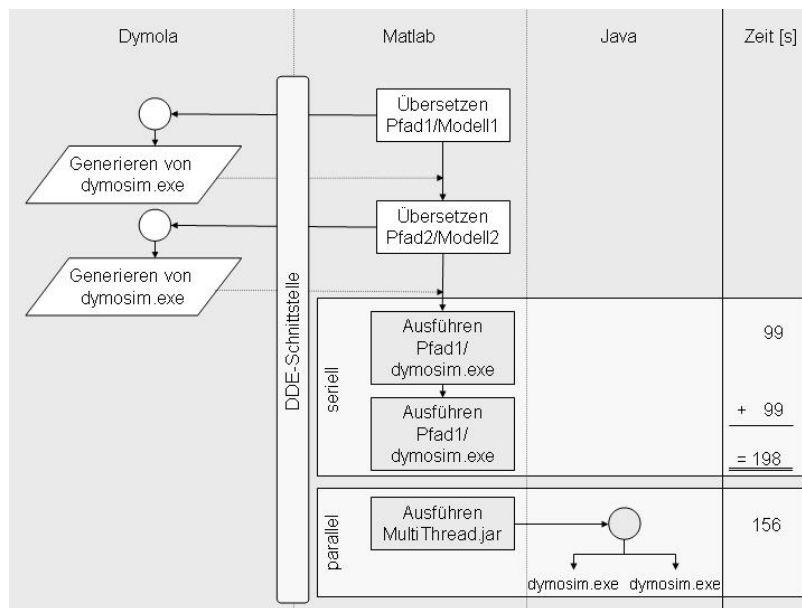


Abbildung 1: Ablauf der seriellen und parallelen Simulationen

Abbildung 1 veranschaulicht den generellen Ablauf der seriellen und der parallelen Simulationen und den dazugehörigen Zeitaufwand. Die Zeitmessungen wurden mit einem

Pentium 4 Prozessor mit 3 GHz (HTT) und 1 GB RAM durchgeführt. Die statistisch ermittelte Durchschnittszeit beträgt 99 Sekunden für eine Einzelsimulation und 156 Sekunden ( $\equiv$  78 Sekunden pro Simulation) für zwei parallele Simulationen. Somit kann durch die parallele Simulation mittels Multithreading ca. 21 % der Zeit pro Simulation eingespart werden, wenn die Hardware HTT/MT-fähig ist.

## 4 Ergebnisse

Die Verwendung diskreter Optimierungsvariablen resultiert in einer ebenfalls diskreten Menge möglicher Lösungen der Zielfunktionen, von denen nur ein Teil die Pareto-Front bildet. Abbildung 2 zeigt die Pareto-Front (links) und die pareto-optimalen Konfigurationen (rechts). Methode I lieferte dabei jedoch nur die mit 'o' gekennzeichneten pareto-optimalen

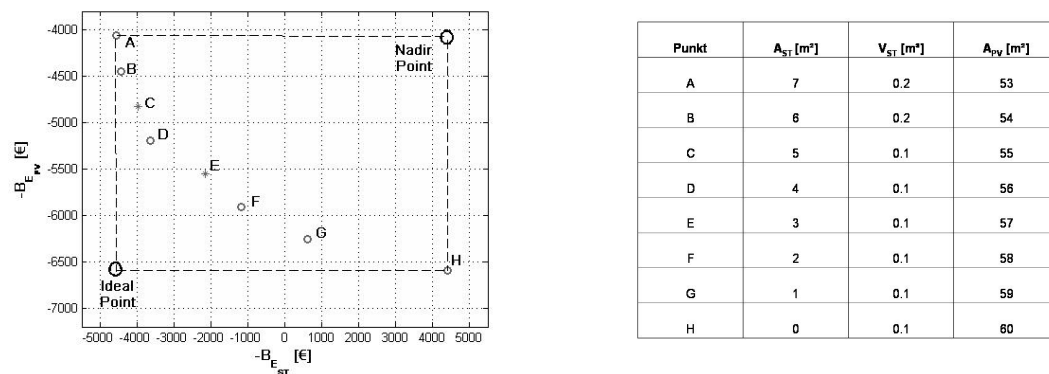


Abbildung 2: Pareto-Front

Werte, da diese Methode eine konvexe Lösungsmenge voraussetzt und die konkaven Abschnitte nicht finden kann. Durch die Minimierung der gewichteten Distanz zwischen einem Referenzpunkt und der aktuellen Konfiguration lassen sich auch die mit '\*' markierten Konfigurationen ermitteln. Die Wahl des Referenzpunktes ist für den Erfolg der Suche entscheidend. In diesem Fall wurden die Randpunkte D und F der konkaven Stelle verwendet, um den Referenzpunkt Z zu bestimmen, der den Streckenmittelpunkt zwischen D und F darstellt ( $\underline{Z} = \underline{D} + \frac{1}{2} \cdot (\underline{F} - \underline{D})$ ). Die nun zu minimierende Gleichung lautet:

$$\min_{A_{ST}, V_{ST}} J = \sqrt{((J_{ST} - z_1) * w_1)^2 + ((J_{PV} - z_2) * w_2)^2}, \text{ mit } \underline{Z} = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}. \quad (7)$$

Auch hier gelten die Bedingungen, wie sie in Gleichung (5) definiert wurden. Die zweite vorgestellte Methode (II) ist für konkave bzw. teilweise konkave Lösungsmengen geeignet und kann alle Punkte der Pareto-Menge auffinden. Zur Unterstützung der Entscheidung für eine der pareto-optimalen Lösungen, kann mit der Distanz-Methode optimiert werden. Der Referenzpunkt ist in diesem Fall der *Ideal Point* (siehe Abbildung 2 links) der durch Optimierung der Einzelanlagen berechnet wird. Das Ergebnis dieser Optimierung ist die Konfiguration D, die allerdings nur einen Kompromiss der Einzelwirtschaftlichkeiten darstellt. Ist

nur eine Einzellösung erwünscht, mit der die wirtschaftlichste Konfiguration ermittelt wird, sind Gleichung (1) und (2) ungewichtet zu addieren und diese Summe als Zielfunktion zu minimieren. Die somit erhaltene Konfiguration (B, siehe Abbildung 2 rechts) liefert das maximale Betriebsergebnis für die hybride Anlage ohne die Möglichkeit einer Präferenzierung einer Technologie durch den Projektanten.

## 5 Zusammenfassung und Ausblick

Durch die Parallelisierung der Simulations-Threads bei der Optimierung mittels HTT wird eine Zeiteinsparung von ca. 21 % pro Simulation im Vergleich zu der seriellen Vorgehensweise erreicht. Zukünftig soll die hier optimal dimensionierte hybride Anlage um weitere Komponenten erweitert werden, wie beispielsweise eine Wärmepumpe und ein Biodiesel-Blockheizkraftwerk. Dadurch wird die Simulationszeit der Modelle ansteigen und eine effektive Ausnutzung der angebotenen Rechenressourcen notwendig. Abschließend wird ein Planungswerkzeug entwickelt werden, das die optimale Dimensionierung einer beliebigen hybriden Anlage nach verschiedenen Kriterien ermöglicht.

## Literatur

- [1] *Dynasim*: Dymola Multi-Engineering Modeling and Simulation. Dymola User Manual, Lund: 2004.
- [2] *Hoffmann, Ch.; Kisser, M.; Kahler, J.; Puta, H.*: Objektorientierte Simulation von Energieversorgungssystemen auf der Basis regenerativer Energien. 48. Internationales Wissenschaftliches Kolloquium, TU Ilmenau, 22.-25.9.2003, Tagungsband S. 661-663.
- [3] *Hoffmann, Ch.; Kahler, J.; Thill, N.; Puta, H.*: Objektorientierte Simulation von Gebäuden mit Modelica. 48. Internationales Wissenschaftliches Kolloquium, TU Ilmenau, 22.-25.9.2003, Tagungsband S. 664-666.
- [4] *Collette, Y.; Siarry, P.*: Multiobjective Optimization. Berlin: Springer, 2003.
- [5] *Pohlheim, H.*: Entwicklung und systemtechnische Anwendung evolutionärer Algorithmen. Aachen : Shaker, 1998.
- [6] <http://www.geatbx.com>
- [7] *Bell, D.; Parr, M.*: Java für Studenten - Grundlagen der Programmierung. München : Pearson Studium, 2003.
- [8] *Burger, D., et al.*: Billion-Transistor Architectures. IEEE Computer 9 (1997), S. 46-49.
- [9] *Eggers, S., et al.*: Simultaneous Multithreading. IEEE Micro 5 (1997), S. 12-19.