

Domänenspezifische Vorabanalyse und geschickte Generierung von Optimierungsproblemen

Benjamin Saul¹ (Betreuer: Wolf Zimmermann)

Abstract: Optimierungsprobleme lassen sich mit domänenspezifischen Sprachen beschreiben, aus denen dann z. B. Gemischt-Ganzzahlige Lineare Programme (MILP) erstellt werden können. Bei der Auswertung dieser durch entsprechende Löser erhält man jedoch nur unspezifische Fehlermeldungen. Um die Anwendbarkeit der domänenspezifischen Sprache zu fördern, soll durch entsprechende Analysen die Lösbarkeit des Problems untersucht und entsprechende Fehlermeldungen ausgegeben werden. Zusätzlich können die ermittelten Informationen bei der Generierung des MILPs angewandt werden, um dessen weitere Verarbeitung zu begünstigen.

1 Motivation

Diese Forschungsarbeit wird im Rahmen des Projektes „Hocheffiziente Pumpensysteme“ durchgeführt. Ein Pumpensystem kann vereinfacht als Graph von Komponenten, also Pumpen, Ventilen und Rohren, aufgefasst werden. Abbildung 1 zeigt ein solches Pumpensystem.

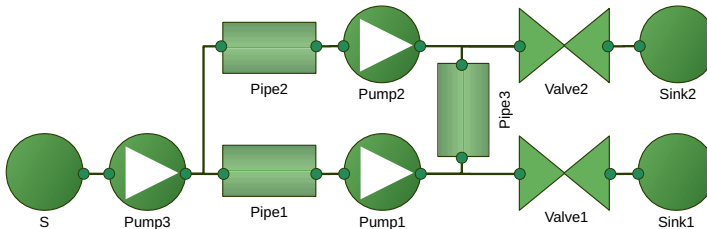


Abb. 1: Beispiel eines Pumpensystems mit Pumpen, Rohren und Ventilen.

Die Aufgabe des Projektes besteht darin, die Rahmenbedingungen eines Pumpensystems zu modellieren und daraus die energie- bzw. kostengünstigste Konfiguration der Komponenten zu bestimmen. Für die Lösung der Aufgabe wird ein Gemischt-Ganzzahliges Lineares Programm (MILP) generiert und durch Standard-Löser verarbeitet. Die entstandene Lösung wird anschließend ausgewertet, indem beispielsweise Simulationsmodelle generiert werden. Das Verfahren wird in Abbildung 2 dargestellt.

Die Modellierung von Pumpensystemen als Gemischt-Ganzzahlige Lineare Programme wird in [Pö15] vorgestellt und u. a. am Beispiel des Darmstadtiums in Darmstadt verwendet. Hier erfolgte die Modellierung des linearen Programmes jedoch von Hand.

¹ Martin-Luther-Universität Halle-Wittenberg, Institut für Informatik, 06120 Halle (Saale), Deutschland, saul@informatik.uni-halle.de

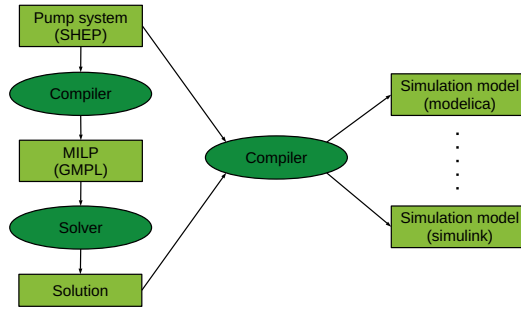


Abb. 2: Übersicht über die Werkzeugkette zur Erstellung optimaler Pumpensysteme.

Domänenspezifische Sprachen erlauben eine effiziente Beschreibung von Problemstellungen durch entsprechende Fachkräfte. Auf dieser Spezifikation erfolgen dann Berechnungen, die das Problem analysieren oder sogar lösen. Aus dem domänenspezifischen Modell kann dann z. B. das entsprechende Optimierungsproblem generiert werden.

Die Löser von Optimierungsproblemen verarbeiten nur das Ungleichungssystem, wodurch domänenspezifische Informationen verloren gehen. Gibt es keine Lösung, so ist es dadurch nur schwer nachvollziehbar, welche Ungleichungen nicht erfüllbar sind. Insbesondere, wenn die Gleichungen generiert wurden, ist dies nur durch fundierte Kenntnisse des entsprechenden Gleichungssystems zu bewältigen. Außerdem kann die Lösungszeit einen erheblichen Zeitaufwand bedeuten, welcher nur schwer vorher einschätzbar ist.

In dieser Arbeit soll daher das Optimierungsproblem in einer domänenspezifischen Beschreibung analysiert werden. Diese Analysen ermöglichen es, entsprechende Widersprüche im Modell aufzudecken, bevor diese im Löser zu Fehlern führen. Dadurch erhält der Anwender ein besseres Feedback und kann mögliche lange Lösungszeiten einsparen.

Als zusätzlicher Effekt können die ermittelten Informationen dabei helfen, Variablen und Gleichungen einzusparen, was das erzeugte Optimierungsproblem verkleinert und damit Lösungszeit einsparen kann. Es ergeben sich die folgenden Forschungsfragen:

- Lassen sich domänenspezifische Informationen verwenden, um auf die Lösbarkeit des entsprechenden Optimierungsproblems schließen zu können?
- Sind diese Methoden besser als die eines Löser für Lineare Optimierungsprobleme?
- Können durch domänenspezifische Analysen Variablen und Restriktionen eingespart werden, sodass die gesamte Lösungszeit verringert wird?

In den folgenden Abschnitten werden zunächst ähnliche Arbeiten aufgeführt. Darunter zählen insbesondere Arbeiten zur Generierung von Optimierungsproblemen sowie Arbeiten über deren Analyse. Anschließend wird der Forschungsplan vorgestellt, der die bewältigten und anstehenden Schritte der Forschungsarbeit zusammenfasst. Der Beitrag schließt mit den bisher erzielten Ergebnissen und einer Zusammenfassung.

2 Verwandte Arbeiten

Die automatische Erzeugung linearer und nichtlinearer Probleme ist notwendig, um auch große Systeme zu modellieren. Übersichtliche Formulierungen bietet die Sprache AMPL [FGK93]. Mit dieser ist es möglich, lineare Probleme in einer verkürzten Form zu beschreiben, indem z. B. Gleichungen und Daten getrennt voneinander angegeben werden können.

Auf so beschriebene Ungleichungssysteme können nur Methoden der allgemeinen Optimierung, wie in [Pa13] beschrieben, angewendet werden. Das heißt, es werden z. B. Variablen eliminiert, indem die zugehörigen Gleichungssysteme weitestgehend gelöst werden. Allerdings kann dieses Vorgehen das weitere Wissen über eventuelle Verknüpfungen von Variablen nicht verwenden, die nicht unmittelbar aus dem Ungleichungssystem hervorgehen.

Andere Ansätze bieten ein komplettes Framework an, bei dem die Modellierung in einer eigenen Sprache erfolgt, damit diese dann direkt weiterverarbeitet werden kann. Ein Beispiel hierfür ist [PJM12], bei dem eine Python-Anwendung entwickelt wurde. Hier werden Optimierungsmodelle mit bereits vorhandenen Bibliotheksfunktionen von Python gelöst. Dabei können dann direkt in der Eingabesprache Vorschläge gemacht werden, welche Lösungsalgorithmen verwendet werden sollten.

Gemischt-Ganzzahlige Lineare und Nichtlineare Probleme besitzen oft alternative Formulierungen. In [SA13] werden einige äquivalente Linearisierungen vorgestellt. Durch den Austausch der Formulierungen kann die Lösungszeit zu Gunsten von Genauigkeit verringert werden.

Für die Generierung von Optimierungsproblemen bietet es sich an, verschiedene Formulierungen für lineare und nichtlineare Zusammenhänge auswählbar zu machen. Ein Beispiel für eine neue Formulierung des Scheduling-Problems findet sich in [OAV12]. Dort werden u. a. die Gleichungen für die benötigten Laufzeiten umformuliert, was eine Verringerung der Lösungszeit mit sich brachte.

Die Lösung der Optimierungsaufgabe kann auch ohne die Generierung eines MILPs erfolgen. In [Su11] wird eine domänenspezifische Sprache vorgestellt, mit dem Quellcode für parallel rechnende Grafikkartenprozessoren beschrieben werden kann. Dieser wird dann auf die jeweilige Architektur angepasst, sodass die Hardware möglichst gut ausgenutzt wird. Die Optimierung selbst erfolgt über eigene Algorithmen innerhalb des Übersetzers.

Ein ähnliches Forschungsfeld ist die Constraint-Basierte Programmierung. Dieses Paradigma erlaubt die Formulierung von Restriktionen, welche dann Einfluss auf die davon betroffenen Variablen haben [RVBW06]. Eine Forschungsarbeit für die Anwendung der Constraint-basierten Programmierung ist [Ma12]. Hier wurde ein Verfahren vorgestellt, welches den Constraint-Ansatz für Produktionslinienmodelle nutzt. Dieser ist deutlich vielseitiger und umfangreicher nutzbar als die grafische Modellierung von Produktionslinien.

Zusammenfassend lässt sich sagen, dass für spezielle Domänen bzw. Gleichungen besonders gute Formulierungen als MILP gefunden wurden, welche so auch generiert werden sollten. Für die domänenspezifischen Analysemethoden können Methoden der Constraint-Programmierung verwendet werden, um Widersprüche zu aufzudecken. Dabei kann die Verknüpfung von Constraint-basierten Modellierungsmethoden und den äquivalenten Umformungen innerhalb eines Optimierungsproblems zu beiderseitigem Vorteil genutzt werden. Die Constraint-Programmierung erhält neue Lösungsmethoden und die Optimierungsprobleme neue Analysemethoden.

3 Aktueller Stand

Die domänenspezifische Sprache SHEP zur Beschreibung der Rahmenbedingungen von Pumpensystemen wurde im Rahmen des Projektes zusammen mit der Werkzeugkette umgesetzt. Die Sprache ermöglicht die Spezifikation von Komponententypen, z. B. Pumpen, deren Zusammenstellung in einem Pumpensystem (vgl. Abb. 1), sowie der geforderten Werte an Druck und Durchfluss an verschiedenen Stellen im System. Auf Basis dieser Sprache und insbesondere am Beispiel von Pumpensystemen sollen die weiteren Verfahren untersucht werden.

Einfache Tests auf Lösbarkeit wurden bereits umgesetzt. So werden die an verschiedenen Stellen des Modells spezifizierten Einschränkungen an Attribute geprüft. Codeabschnitt 1 zeigt einen entsprechenden Ausschnitt mitsamt den generierten Fehlermeldungen.

```
pipe Steelpipe
    length = {[20,35], [40,60], [80,100]};
end

system
    Steelpipe P1(length = [20,30]);// -> ok
    Steelpipe P2(length = [65,75]);// -> ERROR "Empty resulting interval"
    Steelpipe P3(length = [30,50]);// -> WARNING "Impossible values specified"
end
```

Code 1: Auszug einer Systemspezifikation mit drei Rohren unterschiedlicher Länge und der daraus resultierenden Fehlermeldungen.

Im Code wird zunächst der Typ `Steelpipe` definiert, dessen Länge entweder im Intervall $[20, 35]$, $[40, 60]$ oder $[80, 100]$ liegen kann. Anschließend werden im System drei dieser Rohre benutzt und deren Längen bestimmt. Dabei liegt die Länge von P1 komplett im möglichen Intervall und erzeugt somit keinen Fehler. Die Länge des Rohres P2 soll im Intervall $[65, 75]$ liegen, und ist daher mit diesem Typ nicht möglich zu erreichen. Die letzte Länge liegt genau zwischen zwei möglichen Längen, die der Typ erlaubt. Daher wurden sowohl gültige als auch ungültige Werte für die Rohrlänge spezifiziert, was auf einen möglichen Modellierungsfehler hinweist. Daher sollte eine Warnung erzeugt werden.

Das zu generierende lineare Programm wird unter anderem in [Pö15] vorgestellt und wird im Rahmen dieser Arbeit als Beispiel eines Optimierungsproblems verwendet.

4 Forschungsplan

Die folgenden Schritte sind aktuell geplant und werden demnächst umgesetzt. Dabei wird zunächst eine Lösbarkeitsanalyse auf dem domänenspezifischen Modell entwickelt, mit deren Informationen dann versucht wird, das erstellte lineare Programm zu vereinfachen.

4.1 Bestimmung der Lösbarkeit

Ein lineares Programm ist lösbar, wenn es eine Belegung der Variablen gibt, sodass alle Ungleichungen erfüllt sind. In einem Pumpensystem sind die gültigen Belegungen gerade die Betriebspunkte, die die Anlage bedienen kann. Das Berechnen von erreichbaren Betriebspunkten erfordert die Modellierung des Verhaltens der einzelnen Komponenten, um dieses dann über den Systemgraphen zu transportieren. Die Verhaltensfunktionen der Komponenten sowie die Verteilungen der propagierten Werte sind durch die Sprache bzw. das Modell definiert. Erreichbare Zustände müssen abgeschätzt werden. Dazu ist ein Algorithmus in Planung, welcher den Systemgraphen iterativ traversiert und die jeweiligen Betriebspunkte aktualisiert.

Durch die Abschätzung kann angegeben werden, ob die geforderten Werte überhaupt erreicht werden könnten. Ist dies nicht möglich, so muss aus der Analyse ersichtlich sein, welche Komponenten die geforderten Werte nicht unterstützen.

Bei dem entwickelten Verfahren muss per Beweis die Aussagefähigkeit festgestellt werden, also welche Fehler entdeckt werden können und welche nicht. Weiterhin müssen die Fehlermeldungen nach ihrer Verwertbarkeit untersucht werden. Hierzu sollten Testversuche auch durch die endgültigen Anwender erfolgen und ausgewertet werden.

4.2 Bestimmung redundanter Restriktionen und konstanter Variablen

Mit den Informationen über die erreichbaren Zustände können die Verhaltensfunktionen einzelner Komponenten angepasst werden. So muss z. B. nicht abgebildet werden, wie sich eine Pumpe für hohe Durchflüsse verhält, wenn diese dort gar nicht auftreten können. Entsprechende Intervalle sollten angepasst bzw. Stützstellen zur Interpolation weggelassen werden.

Die Untersuchung hierzu ist zweigeteilt. Zum einen kann die Anzahl der entfernten Variablen gezählt werden. Zum anderen muss der Einfluss dieser auf das Lösungsverhalten untersucht werden. Es besteht die Möglichkeit, dass der Löser die nicht benötigten Variablen schneller entdeckt, als dies durch domänenspezifische Verfahren möglich ist.

5 Zusammenfassung

Ziel dieser Arbeit ist die Entwicklung und Untersuchung von Methoden zur Analyse von Optimierungsproblemen, die sich als lineare Probleme formulieren lassen und in einer

domänenspezifischen Sprache vorliegen. Dies geschieht am Beispiel von Pumpensystemen im Rahmen des Projektes "Hocheffiziente Pumpensysteme".

Die zu entwickelnden Methoden sollen das Modell auf Lösbarkeit hin untersuchen und im Falle von Widersprüchen diese konkret durch Fehlermeldungen aufzeigen. Als Zusatz soll untersucht werden, ob durch die ermittelten Informationen das Optimierungsproblem besser formuliert werden kann, also z. B. durch Einsparung ganzzahliger Variablen.

Die zusammengetragenen Methoden sollen beispielhaft implementiert und ausgewertet werden. Dabei soll insbesondere untersucht werden, wie viel Berechnungszeit durch Voranalysen eingespart werden kann.

Falls möglich soll von der aktuellen Domäne abstrahiert werden. Dazu sollen auch andere Domänen betrachtet werden und Gemeinsamkeiten und Unterschiede herausgefunden werden. Die Verfahren können dann helfen, Fehler im Modell frühzeitig zu erkennen.

Literaturverzeichnis

- [FGK93] Fourer, Robert; Gay, David; Kernighan, Brian: *Ampl*, Jgg. 119. Boyd & Fraser, 1993.
- [Ma12] Mazo, Raúl; Salinesi, Camille; Djebbi, Olfa; Diaz, Daniel; Lora-Michiels, Alberto: Constraints: The heart of domain and application engineering in the product lines engineering strategy. *International Journal of Information System Modeling and Design IJISMD*, 3(2):50, 2012.
- [OAV12] Ostrowski, James; Anjos, Miguel F; Vannelli, Anthony: Tight mixed integer linear programming formulations for the unit commitment problem. *IEEE Transactions on Power Systems*, 27(1):39, 2012.
- [Pa13] Padberg, Manfred: *Linear optimization and extensions*, Jgg. 12. Springer Science & Business Media, 2013.
- [PJM12] Perez, Ruben E; Jansen, Peter W; Martins, Joaquim RRA: pyOpt: a Python-based object-oriented framework for nonlinear constrained optimization. *Structural and Multidisciplinary Optimization*, 45(1):101–118, 2012.
- [Pö15] Pöttgen, Philipp; Ederer, Thorsten; Altherr, Lena; Lorenz, Ulf; Pelz, Peter F: Examination and Optimization of a Heating Circuit for Energy-Efficient Buildings. *Energy Technology*, 2015.
- [RVBW06] Rossi, Francesca; Van Beek, Peter; Walsh, Toby: *Handbook of constraint programming*. Elsevier, 2006.
- [SA13] Serali, Hanif D; Adams, Warren P: A reformulation-linearization technique for solving discrete and continuous nonconvex problems, Jgg. 31. Springer Science & Business Media, 2013.
- [Su11] Sujeeth, Arvind; Lee, HyoukJoong; Brown, Kevin; Rompf, Tiark; Chafi, Hassan; Wu, Michael; Atreya, Anand; Odersky, Martin; Olukotun, Kunle: OptiML: an implicitly parallel domain-specific language for machine learning. In: *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*. S. 609–616, 2011.