Abstrakte Interpretation auf domänenspezifischen Sprachen für graphbasierte Optimierungsprobleme

Benjamin Saul, Wolf Zimmermann

Institut für Informatik
Martin-Luther-Universität Halle-Wittenberg
Von-Seckendorff-Platz 1
06120 Halle (Saale)

Zusammenfassung. Domänenspezifische Sprachen erlauben die prägnante Formulierung von Optimierungsproblemen. Ist die Semantik der Sprache klar definiert, so kann diese verwendet werden, um eine abstrakte Interpretation auf den Modellen durchzuführen, die die Problemstellung einschränken oder näher untersuchen können. Dieser Artikel stellt am Beispiel von graphbasierten Optimierungsproblemen vor, wie eine abstrakte Interpretation eingesetzt werden kann.

1 Einleitung

Pumpensysteme transportieren Flüssigkeiten über eine Distanz unter Berücksichtigung von Anforderungen an Druck, Durchfluss und verfügbarem Platz. Durch eine geschickte Auswahl und Kombination verschiedener Pumpen kann bis zu 60% des benötigten Stromverbrauches eingespart werden [8]. Die Aufgabe eines Ingenieurs ist es also, ein möglichst verbrauchsarmes und somit kostenoptimales System zu bestimmen. Abbildung 1 zeigt eine Druckerhöhungsanlage, eine Standardanlage wie sie in Versorgung und Industrie eingesetzt wird.

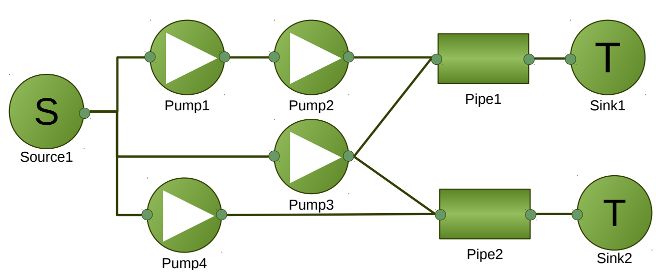


Abb. 1: Beispiel für eine Pumpenanlage, die einen Fluss von einer Quelle zu zwei Senken transportiert.

Die Beispielanlage soll Wasser von einer Quelle zu zwei unterschiedlichen Senken transportieren. Dabei überwinden Rohre die Distanz und Pumpen bauen den Druck auf. Dem Anlagenplaner stehen dabei vier Bauplätze zur Verfügung, in denen Pumpen verschiedenen Typs platziert werden können. Für einen gegebenen Ausgangsdruck in der Quelle und geforderten Drücken und Volumenströmen in den Senken müssen nun passende Pumpen ausgewählt werden. Andere Armaturen werden zur Vereinfachung vernachlässigt.

Das Auswahlverfahren der bestmöglichen Pumpen kann als gemischt-ganzzahliges lineares Programm (MILP) formuliert werden [9]. Dessen Lösung entspricht dem verbrauchsärmsten Pumpensystem, welches die gegebenen Randbedingungen erfüllt. Um diese Modellierungsmethoden besser anwendbar zu machen, wurde die domänenspezifische Sprache SHEP – Sprache für hocheffiziente Pumpensysteme – entwickelt [10]. Diese erlaubt die Beschreibung der Randbedingungen eines Pumpensystemes in einer knappen und präzisen Form. Die Lösung des MILPs geschieht durch bestehende Löser für diese Probleme, das Optimierungsergebnis wird anschließend noch ausgewertet. Abbildung 2 zeigt den Ablauf der Werkzeugkette.

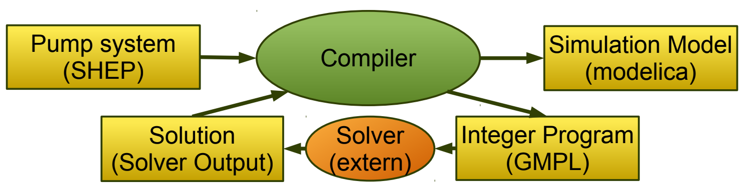


Abb. 2: Werkzeugkette für die Berechnung verbrauchsoptimaler Pumpensysteme.

Die Nutzung externer Löser hat den Vorteil, dass diese leicht austauschbar sind und hier keine Neuentwicklungen bestehender Algorithmen geschehen müssen. Allerdings haben solche Löser zwei Nachteile. Zum einen kann die Lösungszeit sehr lang sein, zum anderen werden inkonsistente Systeme nur schwer nachvollziehbar detektiert.

Eine domänenspezifische Sprache beinhaltet Informationen über Zusammenhänge, welche sonst im linearen Programm erst wieder interpretiert werden müssen. So ist zum Beispiel der Systemgraph und die verschiedenen Abhängigkeiten der Komponenten bekannt. Mit diesen Informatiionen kann eine Programmanalyse in Form einer abstrakten Interpretation auf dem SHEP Modell durchgeführt werden. Dabei sollen die folgenden Fragen beantwortet werden:

– Wie funktioniert eine abstrakte Interpretation auf domänenspezifischen Sprachen für graphbasierte Optimierung?
– Bringen die Ergebnisse der Analyse einen Vorteil in Bezug auf Problemgröße und Anwenderfreundlichkeit?

Um diese Fragen zu untersuchen, wird zunächst in Kapitel 2 die Pumpendomäne vorgestellt. In Kapitel 3 wird anschließend die domänenspezifische Sprache SHEP und ihre Semantik vorgestellt. Darauf aufbauend wird in Kapitel 4 ein Framework für eine abstrakte Interpretation erarbeitet und untersucht. Kapitel 5 stellt einige verwandte Arbeiten vor. Kapitel 6 fässt die Ergebnisse dieser Arbeit zusammen und schließt mit einem Ausblick ab.

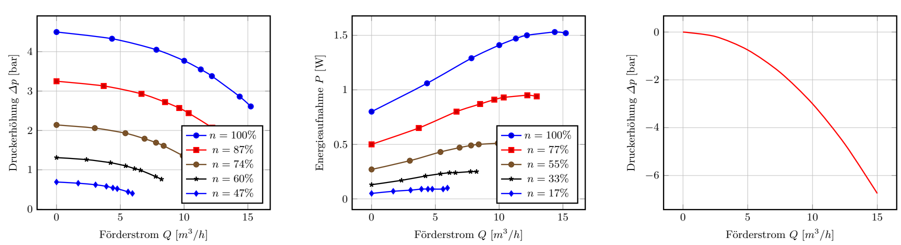
2 Domänenspezifisches Verhalten

Komponenten in Pumpensystemen haben ein unterschiedliches Verhalten. Allerdings lassen sich Gruppen von Komponenten mit ähnlichem Verhalten identifizieren. Leitende Komponenten, wie Pumpen und Rohre, leiten den Volumenstrom von ihrem *in*-Anschluss zum *out*-Anschluss. Dabei wird Volumen vom Förderstrom erhalten, dessen Richtung mit einem Vorzeichenwechsel angepasst und der Druck erhöht bzw. reduziert. Die Größe der Druckdifferenz ist dabei i. A. abhängig vom Volumenstrom, woraus sich

$$Druck_{out} = Druck_{in} + \Delta p(Fluss_{in}) \qquad \text{und} \tag{1}$$

$$Fluss_{in} = -Fluss_{out} \tag{2}$$

ergeben. Im Falle von Pumpen hängt Δp auch von deren Drehzahl ab. Durch eine Erhöhung der Rotirendrehzahl erhöht sich auch die Druckerhöhung auf Kosten des Energieverbrauches der Pumpe. Der genaue Zusammenhang zwischen den verschiedenen Größen wird durch Kennlinien spezifiziert. Kennlinien sind gemessene und interpolierte Funktionen und werden als Funktionsgraphen vom Hersteller veröffentlicht. Sie beschreiben, wie viel Druck eine Pumpe aufbauen kann, wenn sie mit einer bestimmten Drehzahl eingestellt ist und ein gewisser Förderstrom anliegt. Abbildung 3a zeigt aus [6] entnommene Kennlinien.



(a) Die Kennlinie einer Pumpe stellt den Zusammenhang zwischen Druckaufbau Δp, Förderstrom Q, Drehzahl n und Energieaufnahme P dar.

(b) Druckverlust durch ein Rohr, dargestellt als Kennlinie.

Abb. 3: Verhaltensfunktionen von Pumpen und Rohren.

Während bei Pumpen eine einzelne Pumpe durch die veränderliche Drehzahl verschiedene Kennlinien aufweist, besitzen Rohre nur eine einzelne Kennlinie.

Diese ist Abhänig vom Durchmesser, der Länge, der Lage und verschiedener Materialkonstanten und wird durch die Formel

$$\Delta p(\mathit{Fluss}) = -c \cdot \frac{\mathit{Länge} \cdot \mathit{Fluss}^2}{\mathit{Durchmesser}^3} - \mathit{Höhendifferenz} \tag{3}$$

beschrieben [7]. Dabei fässt die Konstante c mehrere Faktoren zusammen, auf die hier nicht näher eingegangen werden soll. Abbildung 3b zeigt den Druckabfall eines Rohrs als Kennlinie.

Wenn Komponenten über ihre Anschlüsse miteinander verbunden werden, so wird der darüber transportierte Volumenstrom auf die verschiedenen Anschlüsse – unter Berücksichtigung der Fließrichtung – aufsummiert. Drücke an verbundenen Stellen müssen gleich sein, da jeder Stelle im Pumpensystem genau ein Druck zugeordnet wird. Diese Zusammenhänge werden durch

$$\forall_{p \in \mathit{Anschlüsse}} : \mathit{Fluss}_p = \sum_{(a,p) \in \mathit{Kanten}} \mathit{Fluss}_{ap} - \sum_{(p,a) \in \mathit{Kanten}} \mathit{Fluss}_{pa} \quad \text{und} \tag{4}$$

$$\forall_{(a,b) \in \mathit{Kanten}} : \mathit{Druck}_a = \mathit{Druck}_b \tag{5}$$

beschrieben. Dabei wird zwischen dem durch einen Anschluss fließendem und dem über eine Verbindung geleiteten Volumenstrom unterschieden. Eine Kante im System entspricht hier der Verbindung zweier Komponenten, wobei eventuelle weitere Armaturen, welche im gebauten System notwendig sind, der Einfachheit halber nicht modelliert werden.

3 Syntax und Semantik von SHEP

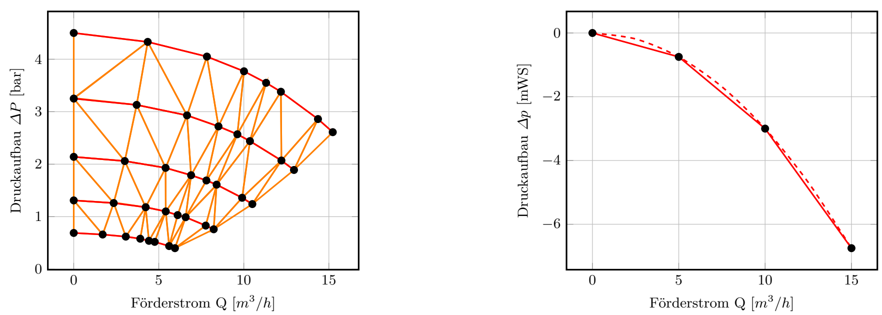
Die domänenspezifische Sprache SHEP beschreibt Pumpensysteme, ihre Komponenten und verschiedene Lastprofile, um damit eine Systemoptimierung durchzuführen [10]. Dazu können Komponententypen definiert werden. Die Definition umfasst vor allem die Beschreibung der Kennlinie, die das Verhalten der Komponente definiert. In Codeausschnitt 1 wird die in Abb. 3a gezeigte Kennlinie modelliert.

```
pump Movitec scales to {34,47,60,74,87,100}
characteristic
  speed flow  head  power;
  100   0.00  4.50  0.80;
  100   4.35  4.33  1.06;
  100   7.83  4.05  1.29;
  // ...
  ports
    in  Flange;
    out Flange;
end
```

Codeausschnitt 1: SHEP Modell der Movitec Pumpe aus Abb. 3a.

Eine Pumpe besitzt einen Typnamen, eine Kennlinie und zwei Anschlüsse. Basierend auf einer Kennlinie bei einer bestimmten Drehzahl, i. d. R. die maximale Drehzahl (100%), können die Kennlinien anderer Drehzahlen durch Ähnlichkeitsgesetze berechnet werden indem die gegebenen Datenpunkte skaliert werden [7]. Die Definition der Anschlüsse ist notwendig, damit beim Verbinden bestimmt werden kann, ob diese zueinander passen.

Die Berechnung der Druckänderung Δp erfolgt durch Interpolation der vorhandenen Datenpunkte. Mehrere Methoden sind dafür möglich, wobei sich innerhalb dieser Arbeit auf eine Triangulation geeinigt wurde, die bereits in [9] verwendet wurde. Dabei wird eine Triangulierung über alle Datenpunkte nach festen Regeln bestimmt, sodass jeweils drei Datenpunkte ihren Zwischenraum interpolieren. Für diese Modellierung sind mehrere ganzzahlige Entscheidungsvariablen notwendig, welche sich negativ auf die spätere Lösungszeit auswirken. Abbildung 4a zeigt die Triangulierung der Kennlinie aus Abb. 3a.



(a) Triangulierung der Kennlinie aus 3a.

(b) Linearisierung (durchgängig) des Druckabfalls in Rohren (gestrichelt, vgl. Abb. 3b.

Abb. 4: Linearisierung von Kennlinien für Pumpen und Rohre.

Rohre und ihr Verhalten werden durch ihre Länge und den Durchmesser bestimmt. Daher sind diese in der Spezifikation anstelle einer expliziten Kennlinie zu finden (Codeabschnitt 2).

```
pipe CommonPipe with 20 support points
  flow     = [0,100];
  diameter = {20, 40, 60} [cm];
  length   = {[20,35], [40,60], [80,100]} [m];
  ports
    in  Flange;
    out Flange;
end
```

Codeausschnitt 2: Declaration of a Pipe in SHEP

In der Typdefinition werden nur mögliche Längen festgelegt, die tatsächliche Länge wird bei jedem Rohr einzeln festgelegt und muss sich an die Intervalle der Typen halten. So kann ein Rohr vom Typ *CommonPipe* nur Längen zwischen 20 und 35, 40 und 60 oder 80 und 100 Metern annehmen.

Um den nichtlinearen Druckabfall (3) zu linearisieren, sind Sützsstellen notwendig. Diese werden in gegebener Anzahl über den zuvor bestimmten Wertebereich verteilt. Sollten keine Angaben diesbezüglich im Modell enthalten sein, so werden die Stützstellen vom Übersetzer nach gewissen Kriterien wie Genauigkeit bestimmt. Ein Beispiel für eine Interpolation des Druckabfalls in Rohren ist in Abb. 4b gegeben.

Durch die Linearisierung wird der mögliche Druckabfall überschätzt. Dadurch wird der lineare Lösungsraum gegenüber dem Nichtlinearen verkleinert, wodurch Lösungen ausgeschlossen werden könnten. Allerdings wird so verhindert, dass es Lösungen im linearen Lösungsraum gibt, welche im nichtlinearen Raum nicht verwendet werden können.

Das Pumpensystem selbst besteht aus den zu kombinierenden Komponenten, den Verbindungen zwischen diesen und diversen Lastfällen, wie Tag- und Nachtbetrieb. Codeabschnitt 5 zeigt die Spezifikation des Pumpensystems aus Abb. 1.

```
system MySys
 Source      Source1;
 Movitec     Pump1 optional;
 Movitec     Pump2 optional;
 Movitec     Pump3 optional;
 Movitec     Pump4 optional;
 CommonPipe Pipe1 (length = 3 [m]);
 CommonPipe Pipe2 (length = 5 [m]);
 Sink        Sink1, Sink2;
connections
 Source1 → Pump1 → Pump2;
 Pump2   → Pipe1 → Sink1;
 Source1 → Pump3 → Pipe1;
 Pump3   → Pipe2 → Source2;
 Source1 → Pump4 → Pipe2;
 Pipe2   → Sink2;
```

```
objective
 minimize costs
  (powerprice=0.24);
scenarios
scenario daytime
  weight 20000 [hours]:
 Source1.out.press = 0 [bar]   ;
 Sink1.in.flow      = 3 [m^3/h];
 Sink1.in.press     = 4 [bar]   ;
 Sink2.in.flow      = 5 [m^3/h];
 Sink2.in.press     = 3 [bar]   ;
end
```

Abb. 5: Beschreibung des Pumpensystems aus Abb. 1 in SHEP.

Aus dieser Spezifikation schließlich wird das lineare Programm bzw. das Gleichungssystem gebildet. Für jede angelegte Komponente werden die ihrem Typ entsprechenden Gleichungen generiert. Dies verbindet ihren eingehenden mit dem ausgehenden Anschluss.

Durch die Verbindungen – beschrieben durch den →-Operator – wird dann der Systemgraph aufgebaut. Die Verteilungsgleichungen (4) und (5) werden basierend auf den hier beschriebenen Kanten implementiert.

Eine Zielfunktion *objective* gibt an, wonach optimiert werden soll. Im Allgemeinen werden die Gesamtkosten, also Einkaufskosten und Stromverbrauch, minimiert, oder nur der Stromverbrauch über die betrachteten Lastfälle.

Ein Lastfall, oder auch *scenario*, wird durch eine Menge an Forderungen an Komponenten wie Quellen und Senken sowie einer erwarteten Betriebszeit definiert. Die Forderungen werden als Gleichung formuliert und können so direkt in das lineare Programm übernommen werden.

4 Abstrakte Interpretation

Aufgabe eines Pumpensystems ist die Erfüllung von Anforderungen an Drücken und Volumenströmen an verschiedenen Stellen im System. Erreichbare Werte definieren dabei das sogenannte Betriebsfeld einer Anlage bzw. einer Pumpe.

Definition 1. *Das* ***Betriebsfeld*** $\Omega \subseteq \mathbb{R}^2$ *eines Knotenpunktes in einem Pumpensystem gibt an, welche Drücke und Durchflüsse dort erreicht werden können. Für eine Pumpe gibt das Betriebsfeld den Bereich ihrer Kennlinien an.*

Alleinige Betrachtung der Betriebsfelder, also Vernachlässigung von Parametern wie Drehzahl oder Stromverbrauch, genügt für die Erfüllbarkeitsfrage von Pumpenanlagen. Die dadurch entstehende Abstraktion ist in Abb. 6 dargestellt.

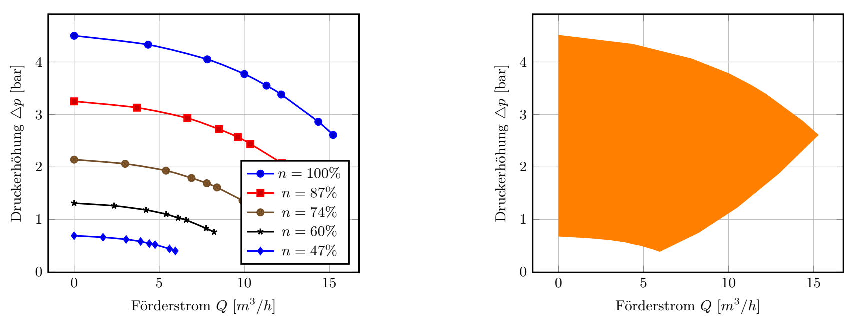
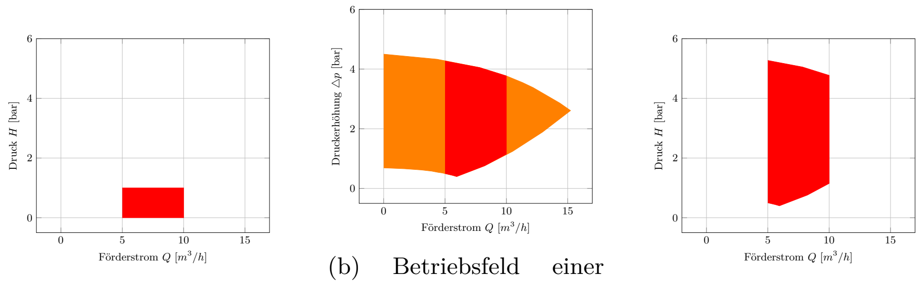


Abb. 6: Abstraktion auf das Betriebsfeld einer Pumpe durch Vernachlässigung der Kennlinien.

Die Druckänderung lässt sich über die auftretenden Mengen definieren. Für ein Pumpenbetriebsfeld $\triangle \in \Omega$ und ein eingehendes Betriebsfeld $X \in \Omega$ definieren wir die Druckänderung $P_\triangle$ entsprechend (1) und (2) als

$$P_\triangle(X) := \{(x_1, x_2 + p) : (x_1, x_2) \in X, (x_1, p) \in \triangle\}.$$

Hierbei ist zu beachten, dass die entstehende Menge auch leer sein kann. Pumpen dürfen nicht außerhalb ihrer Kennlinien arbeiten. In Abbildung 7 wird dieser Zusammenhang veranschaulicht.



(a) Eingehendes Betriebsfeld für eine Pumpe.

(b) Betriebsfeld einer Pumpe mit hervorgehobenen Werten aus Abb. 7a.

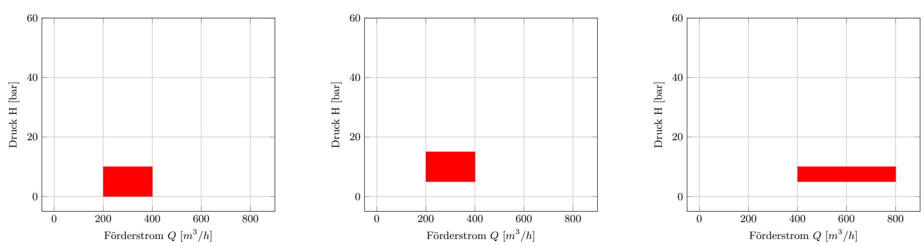
(c) Ausgehendes Betriebsfeld an der Pumpe.

Abb. 7: Druckerhöhung einer Pumpe am Beispiel.

Während Komponenten Druck verändern und den Förderstrom erhalten, ist dies bei den Verbindungen zwischen Komponenten genau anders herum. Wie in (5) und (4) beschrieben, teilen sich die Volumenströme auf während der Druck gleichgesetzt wird. Der Operator $\uplus$ beschreibt diesen Zusammenhang. Für zwei Betriebsfelder $X, Y \in \Omega$ wird definiert:

$$X \uplus Y := \{(p, x_2 + y_2) : p \in \mathbb{R}, (p, x_2) \in X, (p, y_2) \in Y\}.$$

Dabei gilt ähnlich zur Druckerhöhung, dass nur bei gleichem Druck sich die Volumenströme addieren können. Abbildung 8 verdeutlicht diesen Operator.



(a) A

(b) B

(c) $A \uplus B$.

Abb. 8: Zusammenführen zweier Betriebsfelder an einem Punkt.

Mit den oben definierten Funktionen und Operatoren ist es möglich, aus den entsprechenden Codezeilen in SHEP ein Gleichungssystem zu konstruieren. Dies wird in Abb. 9 gezeigt.

```
Movitec Pump1,Pump2,
        Pump3,Pump4;
CPipe Pipe1, Pipe2;
Source1 → Pump1;
Pump1   → Pump2;
Pump2   → Pipe1;
Pipe1   → Sink1;
Source1 → Pump3;
Pump3   → Pipe1;
Pump3   → Pipe2;
Pipe2   → Source2;
Source1 → Pump4;
Pump4   → Pipe2;
```

(a) Komponentendefinitionen und Kanten aus dem Codeabschnitt 5.

$$
\begin{aligned}
Source1.out &= \text{init}\\
Pump1.out &= P_{Movitec}(Pump1.in)\\
Pipe1.out &= P_{CPipe}(Pipe1.in)\\
Pump1.in &= S.out\\
Pump2.in &= Pump1.out\\
Pipe1.in &= Pump2.out \uplus Pump3.out\\
Pump3.in &= S.out\\
Pipe2.in &= Pump3.out \uplus Pump4.out\\
Pump4.in &= S.out\\
Sink1.in &= Pipe1.out\\
Sink2.in &= Pipe2.out
\end{aligned}
$$

(b) Gleichungssystem

Abb. 9: Gegenüberstellung von SHEP Modell und Gleichungssystem.

Die Lösung dieses Gleichungssystems ergibt sich durch Iteration. Als Startwert wird dabei Ω verwendet. Die Terminierung des Verfahrens lässt sich durch Anwendung des Fixpunktsatzes von Kleene zeigen.

Lemma 1 (Kleenes Fixpunktsatz [5]). *Sei* $(L, \sqsubseteq)$ *ein vollständige Halbordnung mit einem kleinstem Element, und sei* $f : L \to L$ *eine monotone Funktion. Dann hat* f *einen kleinsten Fixpunkt, welcher als das Supremum der aufsteigenden Kleene Kette von* f *ist.*

Die Berechnung über den $\uplus$-Operator führt zu einer Überschätzung der tatsächlich möglichen Betriebsfelder, da immer nur direkt verbundene Anschlüsse betrachtet werden. Dadurch wird nicht bestimmt, ob ein Pumpensystem so tatsächlich eine alle Anforderungen erfüllende Konfiguration besitzt. Sollte sich bei der Fixpunktiteration die leere Menge in einer Senke ergeben, so heißt dass, dass es keine gültige Lösung für das Pumpensystem geben kann. Die erforderlichen Drücke bzw. Durchflüsse können an bestimmten Stellen nicht erreicht werden. Diese Information kann an den Nutzer weitergegeben werden, um fehlerhafte Modellierung zu markieren. Wird ein leeres Betriebsfeld an einer anderen Komponenten berechnet, so kann diese Komponente ohne Verlust aus dem System entfernt werden, da sie so nicht betrieben werden kann.

Theorem 1. *Das durch die Funktion $P_\triangle$ und den Operator $\uplus$ Gleichungssystem definiert auf der Menge der Betriebsfelder eine abstrakte Interpretation.*

Beweis. Für den Beweis wird gezeigt, dass $(\Omega, \subseteq)$ ein vollständiger Verband ist und $P_\triangle$ sowie $\uplus$ monotone Funktionen bzw. Operatoren darauf sind. Durch Anwendung von Lemma 1 folgt daraus, dass das Gleichungssystem genau eine Lösung hat, welche nach Konstruktion die möglichen Betriebsfelder berechnet.

Die Grundmenge $\Omega = \mathbb{R}^2$ bildet unter $\subseteq$ einen vollständigen Verband mit Supremum $\mathbb{R}^2$ und Infimum $\emptyset$, da Reflexivität, Transitivität und Asymmetrie durch die Teilmengenbeziehung bereits gegeben sind.

Die Funktion $P_\triangle$ ist monotone: für alle $x, y \in \Omega$ und $x \subseteq y$ gilt

$$\begin{aligned} P_\triangle(x) &= \{(x_1, x_2 + p) : (x_1, x_2) \in X, \qquad (x_1, p) \in \triangle\} \\ &\subseteq \{(x_1, x_2 + p) : (x_1, x_2) \in X \cup Y, (x_1, p) \in \triangle\} \\ &= P_\triangle(y). \end{aligned}$$

Außerdem gilt für den Operator $\uplus$ und alle $x^1, x^2, y^1, y^2 \in \Omega$ mit $x^1 \subseteq x^2$ und $y^1 \subseteq y^2$

$$\begin{aligned} x^1 \uplus y^1 &= \{(p, x + y) : p \in \mathbb{R}, (p, x) \in x^1, (p, y) \in y^1\} \\ &\subseteq \{(p, x + y) : p \in \mathbb{R}, (p, x) \in x^2, (p, y) \in y^2\} \\ &= x^2 \uplus y^2. \end{aligned}$$

Somit sind die Anforderungen für Lemma 1 erfüllt und das Gleichungssystem konvergiert gegen einen Fixpunkt. □

5 Verwandte Arbeiten

Abstrakte Interpretationen wurden in [3] vorgestellt. Darin wird das Framework für eine allgemeine abstrakte Interpretation auf Programmiersprachen vorgestellt, welches vielseitige Anwendung in Programmanalysen findet ([2], [1]).

In [2] wird die abstrakte Interpretation genutzt um das Programmverhalten bzgl. Aussagen in temporaler Logik zu überprüfen. Durch Abstraktion der zu beobachtenden Aussagen können auch größere Eingaben getestet werden.

Alur et. al. beschreiben in [1] eine abstrakte Interpretation für die Untersuchung hybrider Systeme. Dabei müssen kontinuierliche Domänen diskretisiert werden, um sie zu analysieren. Dabei müssen einige Wertebereiche zusätzlich eingeschränkt werden, um eine Entscheidbarkeit zu erzwingen.

Götz et. al. zeigen einen Ansatz zur optimalen Verteilung von Lastprofilen, bei dem die eigentliche Optimierung über ein generiertes gemsicht-ganzzahliges lineares Programm erfolgt [4]. Auch hier wird versucht, die Lösungszeit durch entsprechende Analysen zu verringern.

Für die effiziente Lösung von gemischt-ganzzahligen linearen Programmen gibt es verschiedene Ansätze, die von der Problemstruktur und der Größe abhängen [11]. Diese Methoden werden in dieser Arbeit nicht weiter verfolgt, da sie unabhängig von der abstrakten Interpretation auf den generierten linearen Programmen erfolgen können.

6 Zusammenfassung und Ausblick

Diese Arbeit zeigt ein Framework für eine abstrakte Interpretation auf einer domänenspezifischen Sprache für die Optimierung von Pumpensystemen. Ein Pumpensystem ist dabei als Graph zu verstehen, bei dem die Komponenten Funktionen zwischen ihren beiden Anschlüssen definieren und die durch Kanten miteinander über gewisse Variablen verbunden sind. Die wichtigsten Variablen dabei sind Druck und Durchfluss, die auch für die Erfüllbarkeit eines Pumpensystems entscheiden. Nur wenn erforderliche Größen erreicht werden, kann das System so funktionieren.

Die eigentliche Optimierung dabei geschieht über einen externen Löser für gemischt-ganzzahlige lineare Programme, welche aus der dafür entwickelten Sprache SHEP generiert werden. Die Nachteile eines Lösers sind dabei die eventuell lange Rechenzeit sowie die schlechte Markierung von Inkonsistenzen. Diese Schwächen versucht die hier entwickelte Analyse zu beheben.

Die Abstrakte Interpretation analysiert mögliche Betriebsfelder und bestimmt so für jeden Knotenpunkt im Pumpengraphen die erreichbaren Werte von Drücken und Durchflüssen. Diese Information kann einerseits verwendet werden, um Inkonsistenzen zu erkennen. Ist ein geforderter Druck zum Beispiel nicht innerhalb des jeweiligen Betriebsfeldes, so kann die Anlage die gestellten Anforderungen nicht erfüllen. Andererseits müssen die Funktionen der einzelnen Komponenten nur für die überhaupt erreichbaren Wertebereiche modelliert werden. Dadurch können Variablen eingespart werden.

Die vorgestellten Analysen sind noch erweiterbar. Zusätzlich zu den vorwärts gerichteten Berechnungen kann für jede Pumpe beispielsweise auch die umgekehrte Funktion bestimmt werden, um den Zusammenhang zwischen Eingang und Ausgang herzustellen. Damit lassen sich deren Wertebereiche weiter einschränken. Außerdem wurden optionale Komponenten, also solche, die durch die Optimierung entfallen können, nicht weiter betrachtet.

Danksagung

Dieses Projekt wird gefördert durch das Bundesministerium für Wirtschaft und Technologie unter dem Förderkennzeichen 03ET1134C. Partner in diesem Projekt sind die Technische Universität Darmstadt und die KSB Aktiengesellschaft.

Literatur

[1] Rajeev Alur u. a. „Discrete abstractions of hybrid systems“. In: *Proceedings of the IEEE* 88.7 (2000), S. 971–984.

[2] Edmund M Clarke, Orna Grumberg und David E Long. „Model checking and abstraction“. In: *ACM transactions on Programming Languages and Systems (TOPLAS)* 16.5 (1994), S. 1512–1542.

[3] Patrick Cousot und Radhia Cousot. „Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints“. In: *Proceedings of the 4th ACM SIGACT-SIGPLAN symposium on Principles of programming languages*. ACM. 1977, S. 238–252.

[4] S Götz u. a. „Modeldriven self-optimization using integer linear programming and pseudoboolean optimization“. In: *Proceedings of ADAPTIVE* (2013), S. 55–64.

[5] Stephen Cole Kleene u. a. *Introduction to metamathematics*. Bd. 483. van Nostrand New York, 1952.

[6] KSB Aktiengesellschaft. *Baureihenheft Movitec B*. 2013. URL: `http://www.pumpentechnik-strauss.de/index_htm_files/Movitec50.pdf`.

[7] AG KSB. „Kreiselpumpen Lexikon“. In: *KSB AG, Frankenthal, Germany* (1989).

[8] Manfred Oesterle und Fred Leidig. „Methodisch sichere, schnelle Produktionsanlaufe in der Mechatronik (MESSPRO)“. In: *Schneller Produktionsanlauf in der Wertschopfungskette* Band 2 (2007).

[9] Peter Pelz u. a. *Designing Pump Systems by Discrete Mathematical Topology Optimization: The Artificial Fluid Systems Designer*. Techn. Ber. Düsseldorf: International Rotating Equipment Conference, 2012.

[10] Benjamin Saul, Christian Berg und Wolf Zimmermann. „A Domain Specific Language for Optimal Pumping Systems“. In: *Proceedings of the 1st Industry Track on Software Language Engineering*. ITSLE 2016. Amsterdam, Netherlands: ACM, 2016, S. 23–32. ISBN: 978-1-4503-4646-7. DOI: `10.1145/2998407.2998412`. URL: `http://doi.acm.org/10.1145/2998407.2998412`.

[11] Martin WP Savelsbergh. „Preprocessing and probing techniques for mixed integer programming problems“. In: *ORSA Journal on Computing* 6.4 (1994), S. 445–454.