Die approbierte Originalversion dieser Diplom-/Masterarbeit ist in der Hauptbibliothek der Technischen Universität Wien aufgestellt und zugänglich.
http://www.ub.tuwien.ac.at

The approved original version of this diploma or master thesis is available at the main library of the Vienna University of Technology.
http://www.ub.tuwien.ac.at/eng

TECHNISCHE UNIVERSITÄT WIEN
Vienna University of Technology

Institut für
Fertigungstechnik und
Hochleistungslasertechnik

Diplomarbeit

Simulationsbasierte Optimierung der Betriebsstrategie von energietechnischen Anlagen

ausgeführt zum Zwecke der Erlangung des akademischen Grades eines

Diplom-Ingenieurs unter der Leitung von

Univ. Prof. Dipl.-Ing. Dr. techn. Friedrich Bleicher

(Institut für Fertigungstechnik und Hochleistungslasertechnik)

Dipl.-Ing. Benjamin Mörzinger

(Institut für Fertigungstechnik und Hochleistungslasertechnik)

eingereicht an der Technischen Universität Wien

Fakultät für Maschinenwesen und Betriebswissenschaften

von

Florian Kloibhofer, BSc

01226868 (066.482)

Donatistraße 9

3321 Stephanshart

Stephanshart, im Februar 2018

_____

Florian Kloibhofer

Ich habe zur Kenntnis genommen, dass ich zur Drucklegung meiner Arbeit unter der Bezeichnung

Diplomarbeit

nur mit Bewilligung der Prüfungskommission berechtigt bin.

Ich erkläre weiters an Eides statt, dass ich meine Diplomarbeit nach den anerkannten Grundsätzen für wissenschaftliche Abhandlungen selbstständig ausgeführt habe und alle verwendeten Hilfsmittel, insbesondere die zugrunde gelegte Literatur, genannt habe.

Weiters erkläre ich, dass ich dieses Diplomarbeitsthema bisher weder im In- noch Ausland einer Beurteilerin/einem Beurteiler zur Begutachtung in irgendeiner Form als Prüfungsarbeit vorgelegt habe und, dass diese Arbeit mit der vom Begutachter beurteilten Arbeit übereinstimmt.

Stephanshart, im Februar 2018

_____

Florian Kloibhofer

Danksagung

Besonders danken möchte ich Dipl.-Ing. Benjamin Mörzinger für die gute Zusammenarbeit und Unterstützung. Er ist mir immer mit Rat und Tat zur Seite gestanden. Ohne seine fachliche Unterstützung wäre diese Diplomarbeit nicht möglich gewesen. Auch dem Institut für Fertigungstechnik und Hochleistungslasertechnik der TU Wien möchte ich für die Zusammenarbeit während der Diplomarbeit danken. Das tolle Arbeitsklima habe ich sehr geschätzt.

Außerdem danke ich der Fa. Infineon für die gute und konstruktive Zusammenarbeit. Durch das Feedback und den laufenden Informationsaustausch konnten die Simulationsmodelle immer weiter verbessert werden und schlussendlich ein gutes Ergebnis erzielt werden. Ein Dankeschön geht ebenso an die Fa. Siemens, die die Simulationssoftware für die Diplomarbeit zur Verfügung gestellt hat.

Danken möchte ich auch meinen Eltern, die mich während des ganzen Studiums und speziell während der Zeit der Diplomarbeit sowohl persönlich, als auch finanziell, in jeder erdenklichen Weise unterstützt haben.

Kurzfassung

Die Kühlung von Produktionsanlagen und Gebäuden ist ein energieintensiver Prozess. Besonders in der Halbleiterindustrie müssen beim Herstellungsprozess enge Temperaturgrenzen eingehalten werden. Aufgrund des daraus resultierenden hohen Bedarfs an Kälteleistung ist der ressourcenschonende Betrieb von Kälteanlagen in diesem Industriezweig von besonderer Wichtigkeit. Hier setzt diese Arbeit an und beschäftigt sich mit dem Finden einer optimierten Betriebsstrategie für Kälteanlagen. Die Ermittlung einer solchen Strategie ist schwierig, da es sich um Kälteanlagen handelt, die aufgrund des hohen Kältebedarfs aus mehreren Kältemaschinen aufgebaut sind. Es gibt viele Einflussfaktoren, die bei der Festlegung der Betriebsstrategie berücksichtigt werden müssen.

Um eine optimierte Betriebsstrategie zu ermitteln wird ein simulationsbasierter Optimierungsansatz gewählt. Das komplexe System der energietechnischen Anlage wird bei der Modellbildung in seine Komponenten zerlegt. Dafür wird ein Ansatz verwendet, der eine Modularisierung des Systems mittels Cubes ermöglicht. Durch den modularen Aufbau des Simulationsmodells kann es leicht für verschiedene Anlagen und Anlagenkonfigurationen adaptiert werden.

Anschließend werden die Modelle in die Simulationssoftware AMESIM implementiert. Danach folgt die Implementierung eines Optimierers, der eingesetzt wird, um eine optimierte Betriebsstrategie zu ermittelt. Aus implementierungstechnischen Gründen wird die Optimierung in zwei Stufen durchgeführt: durch Simulation einer Permutationsliste und einer anschließenden Optimierung mit einem genetischen Algorithmus.

Mit dem erstellten Simulationsmodell und dem Optimierer wird die Optimierung der Betriebsstrategie einer bestehenden Kälteanlage durchgeführt. Es wird gezeigt, dass durch die optimierte Betriebsstrategie der Energiebedarf der Kälteanlage, abhängig vom Szenario, um 5% bis 21% reduziert werden kann. Die Ergebnisse lassen vermuten, dass durch den gewählten Ansatz Einsparungspotenzial beim Energiebedarf bestehender Kälteanlagen identifiziert und durch eine optimierte Betriebsstrategie eine Effizienzsteigerung beim Betrieb erreicht werden kann.

Abstract

The cooling of production plants and buildings is an energy-intensive process. Especially in the semiconductor industry, tight temperature limits must be maintained during the manufacturing process. Due to the resulting high cooling demand, the resource-saving operation of refrigeration systems in this industry is of importance. The subject of this work is to find an optimized operating strategy for refrigeration systems. The determination of such a strategy is difficult, because due to the high cooling demand, the refrigeration systems are composed of several chillers. There are many influencing factors that must be considered when defining the operating strategy.

To determine an optimized operating strategy, a simulation-based optimization approach is chosen. The complex system of the refrigeration system is divided into its components in the modelling process. For this an approach is used, which allows a modularization of the system by means of cubes. Due to the modular structure of the simulation model, it can be easily adapted for different systems and system configurations.

Then the models are implemented in the simulation software AMESIM. This is followed by the implementation of an optimizer, which is used to determine an optimized operating strategy. For reasons of implementation, the optimization is carried out in two stages: the first stage is the simulation of a permutation list and the second stage is the optimization with a genetic algorithm.

The simulation model and the optimizer are used to optimize the operating strategy of an existing refrigeration system. It is shown that the optimized operating strategy can reduce the energy demand of the refrigeration system by 5% to 21% depending on the scenario. The results suggest that the selected approach can identify potential for energy savings of existing refrigeration systems and increase the energy efficiency of operation through an optimized operating strategy.

Inhaltsverzeichnis

1 Einleitung 1
2 Grundlagen und Stand der Technik 5
2.1 Übersicht und Begriffsbestimmung 5
2.2 Simulation und Modellbildung 8
2.2.1 Konzeptmodell 10
2.2.2 Formale Modelle 13
2.2.3 Parametrisierung 18
2.3 Optimierung 22
2.3.1 Optimierungsansätze 22
2.3.2 Evolutionäre Algorithmen 25
2.3.3 Genetische Algorithmen 25
2.3.4 Partikel-Schwarm Algorithmen 27
2.3.5 Weitere Algorithmen 27
2.3.6 Optimieren mit Randbedingungen 28
3 Implementierung 32
3.1 Grundlagen 33
3.1.1 Variablentypen 33
3.1.2 Modellports 34
3.1.3 Aufbau des AMESIM-Templates 35
3.2 Ausführbare Modelle 37
3.2.1 Kältemaschinenmodell 37
3.2.2 Energienetz-Modell 40
3.2.3 Verteilfunktion 43
3.2.4 Zielfunktionsbaustein 48
3.2.5 Rückkühlwerk-Modell 52
3.3 Aufbau der lauffähigen Simulation 54
3.4 Optimierung 58
3.4.1 Implementierung des Prioritätsvektors 58
3.4.2 Implementierung der Permutationsliste 62
3.4.3 Anwendung des genetischen Algorithmus 62

4 Experimente und Ergebnisse 67
4.1 Simulationsszenario und Parameter 67
4.1.1 Eingangsgrößen 68
4.1.2 Parameter der Energienetze 70
4.1.3 Zielfunktionsparameter 71
4.1.4 Weitere Parameter 71
4.2 Optimierungsergebnis 72
4.3 Betrachtung wichtiger Zeitverläufe 74
4.4 Weitere Betrachtungen 79
4.5 Einsparungspotenzial 80
4.5.1 Elektrischer Energiebedarf 83
4.5.2 Gesamtenergiebedarf 84
4.5.3 Zusammenfassung der untersuchten Szenarien 85
5 Zusammenfassung und Ausblick 86
6 Anhang 88
6.1 Kältemaschinenparameter 88
6.2 Quellcode Kalkulationsfunktionen 94
6.2.1 Kältemaschinenmodell 94
6.2.2 Energienetz-Modell 96
6.2.3 Verteilfunktion 97
7 Literaturverzeichnis 106
8 Abbildungsverzeichnis 110
9 Formelverzeichnis 112
10 Tabellenverzeichnis 115
11 Abkürzungsverzeichnis 116

1 Einleitung

Viele gängige Herstellungsprozesse in der Industrie sind temperaturabhängig. Nur wenn ein bestimmtes Temperaturniveau bei der Fertigung eingehalten wird, entsprechen die gefertigten Produkte den Qualitätsanforderungen. Um dieses Temperaturniveau unabhängig von äußeren Bedingungen gewährleisten zu können, ist es notwendig die Produktionsanlagen zu klimatisieren. Für die Halbleiterindustrie ist die Klimatisierung der Produktion von besonderer Bedeutung. Halbleiter müssen verfahrensbedingt unter Einhaltung von engen Temperaturgrenzen gefertigt werden.

Daraus ergibt sich ein hoher Kältebedarf, der durch mehrere Kältemaschinen gedeckt wird. Die Kältemaschinen sind an zwei getrennte Wasserkreisläufe angeschlossen. Der erste ist der Kaltwasserkreislauf. Das Kaltwasser wird durch die Kältemaschinen gekühlt und an die Produktionsanlagen weitergeleitet. Dort wird es zur Klimatisierung eingesetzt. Die überschüssige Wärme der Kältemaschinen wird an den zweiten Kreislauf weitergegeben. Dieser wird als Kühlkreislauf bezeichnet. Der Kühlkreislauf gibt die Abwärme der Kältemaschinen über ein Rückkühlwerk an die Umgebung ab. Diese Komponenten bilden zusammen die Kälteanlage. Es ist aus ökologischer und ökonomischer Sicht sinnvoll die Kälteanlage den Anforderungen entsprechend, optimal zu betreiben. Aus dieser Forderung nach ressourcenschonendem Betrieb leitet sich die wesentliche Fragestellung ab, die mit dieser Arbeit beantwortet werden soll: Mit welchen Methoden kann eine optimierte Betriebsstrategie einer Kälteanlage ermittelt werden und wie können diese Methoden in das Softwaretool AMESIM implementiert werden?

Die Aufgabenstellung dieser Arbeit umfasst somit die Implementierung eines Simulationsmodells einer Kälteanlage in das vorgegebene Softwaretool AMESIM und in weiterer Folge das Finden und Implementieren einer geeigneten Methode zur Optimierung der Betriebsstrategie der Anlage. Weiters umfasst die Aufgabenstellung die Anwendung der Optimierung auf eine bestehende Kälteanlage mit Hilfe von AMESIM und die Bewertung des daraus resultierenden Einsparungspotenzials. Nachfolgend wird ein Überblick über die Literatur in diesem Themenbereich gegeben. Auf die genannten Quellen wird im Verlauf der Arbeit an geeigneter Stelle detailliert eingegangen.

In [1] werden unterschiedliche Strategien zur Energieeffizienzsteigerung von Kälteanlagen vorgestellt. Neben der Verwendung von neuen Technologien auf Komponentenebene stellt die Anwendung von intelligenten Betriebsstrategien eine Möglichkeit zur Effizienzsteigerung dar. Die Autoren beschreiben die optimale Wahl der Einschaltreihenfolge der Maschinen als eine solche Betriebsstrategie. Die relevante Fragestellung, die beantwortet werden muss, lautet: Wann wird welche

Kältemaschine betrieben? Besonders interessant ist diese Fragestellung aufgrund der Tatsache, dass die einzelnen Kältemaschinen unterschiedliche Spezifikationen aufweisen. Die Entscheidung ob eine Maschine in einem bestimmten Szenario betrieben werden soll ist nicht allein über den Wirkungsgrad zu treffen. Denn Maschinen mit Wärmerückgewinnungssystem weisen zwar einen niedrigeren Wirkungsgrad auf, als Maschinen ohne Wärmerückgewinnungssystem, jedoch kann die rückgewonnen Wärme in anderen Bereichen eingesetzt werden. Es können weitere Betrachtungen wie z.B. die Wartungskosten von einzelnen Maschinen oder die Belastung des Kühlsystems der Kältemaschinen miteinbezogen werden. Durch die Gewichtung der einzelnen Teilziele werden verschiedene Strategien verfolgt (z.B. Kosteneffizienz, Energieeffizienz). Strategische Managemententscheidungen können dadurch schnell in die operative Ebene umgesetzt werden. Die relative Wichtigkeit der Teilziele zueinander kann sich außerdem zeitlich verändern (z.B. durch schwankende Energiepreise). Durch die Anpassung der Gewichtung der Teilziele kann auf diese Änderungen reagiert werden.

In [2] werden Möglichkeiten zur verbesserten Maschinensteuerung, in Abhängigkeit von der Kältelast, dargestellt. Den Autoren gelingt es durch Modifikation des Kühlwassertemperatursollwertes und der Umschaltzeitpunkte der Maschinen Einsparungen in der Höhe von 5,3% des Energiebedarfs der Kälteanlage zu erzielen. Wird zusätzlich eine ideale Verteilung der Kältelast auf die Maschinen berücksichtigt, kann der Energiebedarf um insgesamt 5,6% reduziert werden. In [3] werden über eine optimierte Einschaltreihenfolge der Maschinen bis zu 4,3% des Energiebedarfs eingespart. Gleichzeitig wird ein Konzept zur vereinfachten Bestimmung einer optimierten Kältelastverteilung auf die Maschinen vorgestellt. Durch die Vereinfachung wird das Einsparungspotenzial reduziert (6,7% im Vergleich zu 10,3% bei klassischer Optimierung der Kältelastverteilung), aber dafür können unnötige Ein- und Ausschaltvorgänge der Maschinen verhindert werden und die Stabilität des Systems wird verbessert.

Ein weiterer Ansatz wird in [4] vorgestellt. Dabei wird die Komplexität des Optimierungsproblems reduziert, indem alle Kältemaschinen als eine einzige Maschine betrachtet werden. Zusätzlich wird die Möglichkeit zur Energiespeicherung über einen thermischen Speicher in die Betrachtung miteinbezogen. So können Lastspitzen abgefangen werden und die Kältemaschinen werden zu Zeiten mit niedrigen Energiekosten betrieben. Der Energiebedarf kann um 9,4% und die Kosten um 17,4% gesenkt werden.

In [5] werden die Korrelationen zwischen verschiedenen Einflussgrößen und der Leistungszahl COP (Coefficient of Performance) untersucht. COP ist das Verhältnis von abgegebener Kälteleistung zu elektrischer Leistungsaufnahme der Kältemaschine. Die Autoren stellen fest, dass die Kältelast und die Temperaturen der

angeschlossenen Wasserkreisläufe den größten Einfluss auf COP haben. Durch Anpassungen dieser Einflussgrößen können Energieeinsparungen von 5% beim elektrischen Energiebedarf realisiert werden.

In Literaturquelle [6] erfolgt die Optimierung mit einem generischen Modell, das die elektrische Leistungsaufnahme der Kältemaschinen mit einer Polynomfunktion vorhersagt. Die Koeffizienten der Polynomfunktion werden mittels Messdaten bestimmt. Durch den generischen Aufbau wird das Modell sowohl zur Optimierung einer Wasserförderanlage, wie auch zur Optimierung einer Kälteanlage eingesetzt. Im Falle der Kälteanlage kann der Energiebedarf um 11,95% gesenkt werden. Das entspricht einer Kostenreduktion um 16,44%.

Gute Ergebnisse mit variabler Kältelast und unter Berücksichtigung der Wetterbedingungen als zusätzliche Einflussgröße werden in [7] erzielt. Als Freiheitsgrade werden die Einschaltreihenfolge der Maschinen, Durchflussmenge der Wasserkreisläufe, Temperaturen der Wasserkreisläufe und der Luftdurchsatz in den Rückkühlwerken verwendet. Es werden verschiedene Anlagekonfigurationen untersucht und die Einsparungen beim Energiebedarf liegen bei bis zu 40% für mittelgroße Anlagen (2 bis 3 Kältemaschinen) und bei 20% für kleine Anlagen (1 Kältemaschine).

Simulationsbasierte Ansätze sind in der Literatur weit verbreitet. In [8] wird ein mehrstufiger genetischer Algorithmus eingesetzt, um eine Kälteanlage zu optimieren. Dabei werden einerseits die Kältelastverteilung auf die einzelnen Maschinen und andererseits die Einschaltreihenfolge der Maschinen optimiert. Durch den mehrstufigen genetischen Algorithmus werden bessere Ergebnisse, als mit anderen Standard-Algorithmen erzielt. In [9] wird ebenfalls ein genetischer Algorithmus zur Optimierung einer Kälteanlage eingesetzt. Im Vergleich zu einer ebenfalls untersuchten Referenzstrategie können zusätzlich 0,73%-2,55% des Energiebedarfs eingespart werden.

Neben genetischen Algorithmen werden auch verbesserte Partikel-Schwarm Algorithmen zur Lösung des Optimierungsproblems eingesetzt [10]. In [11] wird ein neuronales Netzwerk zur Modellbildung verwendet und die Optimierung mit einem Partikel-Schwarm Algorithmus durchgeführt. Bei einer aus 5 Maschinen bestehenden Kälteanlage können durch eine optimierte Kältelastverteilung bis zu 17,6% des elektrischen Energieverbrauchs eingespart werden. Ein weiterer Optimierungsversuch mit Partikel-Schwarm Algorithmen wird in [12] unternommen. Es werden vier Szenarien mit unterschiedlichen Freiheitsgraden für den Optimierer untersucht. Beim Szenario mit den meisten Freiheitsgraden (Temperatursollwerte von Kalt- und Kühlwasserkreislauf, Einschaltreihenfolge) können die höchsten Einsparungen von 13,6% des Gesamtenergiebedarfs erreicht werden. Werden Freiheitsgrade entzogen sinkt das Einsparungspotenzial. In [13] wird ein datenbasierter Ansatz gewählt und ein

genetischer Algorithmus in Kombination mit einem Partikel-Schwarm Algorithmus zur Lösung des Optimierungsproblems eingesetzt. Bei der aus 4 Kältemaschinen bestehenden Anlage können an zwei verschiedenen untersuchten Tagen 14% des Energiebedarfs eingespart werden.

Wie durch obige Quellen dargestellt gibt es in der Literatur bereits verschiedene Methoden die Energieeffizienz bestehender Kälteanlagen zu verbessern. Aufgrund des simulationsbasierten Optimierungsansatzes, der für diese Arbeit gewählt wird, werden im nachfolgenden Kapitel die theoretischen Grundlagen in Bezug auf Modellbildung, Simulation und Optimierungsalgorithmen betrachtet. Dann wird die Modellbildung mit einem modularen Ansatz beschrieben. Durch den modularen Aufbau des Simulationsmodells ist die Anpassung des Modells an verschiedene Anlagekonfigurationen möglich. Danach werden die in den Literaturquellen verwendeten Optimierungsansätze betrachtet. Außerdem werden die erwähnten Optimierungsalgorithmen erläutert und die Anwendbarkeit auf das Optimierungsproblem untersucht. Im anschließenden Kapitel erfolgt die Implementierung von Simulation und Optimierer in die Simulationssoftware AMESIM. Dabei wird auf die implementierungsspezifischen Limitierungen der verwendeten Software eingegangen. Dann werden Experimente am Simulationsmodell durchgeführt. Dabei wird eine optimierte Betriebsstrategie für eine bestehende Kälteanlage ermittelt und die Ergebnisse ausgewertet. Im abschließenden Kapitel werden die gewonnenen Erkenntnisse diskutiert und weiterführende Verbesserungsvorschläge erläutert.

2 Grundlagen und Stand der Technik

In diesem Kapitel wird eine Begriffsbestimmung durchgeführt und die theoretischen Grundlagen von Modellbildung, Simulation und Optimierung erläutert. Dann wird die Modellbildung vom Konzeptmodell zu den formalen Modellen beschrieben. Außerdem werden Möglichkeiten zur datenbasierten Parametrisierung von Modellen betrachtet und die Parametrisierung anhand von Messdaten beschrieben. Abschließend befasst sich das Kapitel mit zwei wesentlichen Optimierungsansätzen für Kälteanlagen nach dem aktuellen Stand der Technik. Außerdem werden verschiedene evolutionäre Optimierungsalgorithmen beschrieben und aktuelle Forschungsergebnisse erläutert.

2.1 Übersicht und Begriffsbestimmung

Um eine Optimierung durchführen zu können, muss das Systemverhalten durch eine Simulation abgebildet werden. Die beiden Komponenten (Optimierer und Simulation) und ihre Wechselwirkungen werden in Abbildung 1 dargestellt.

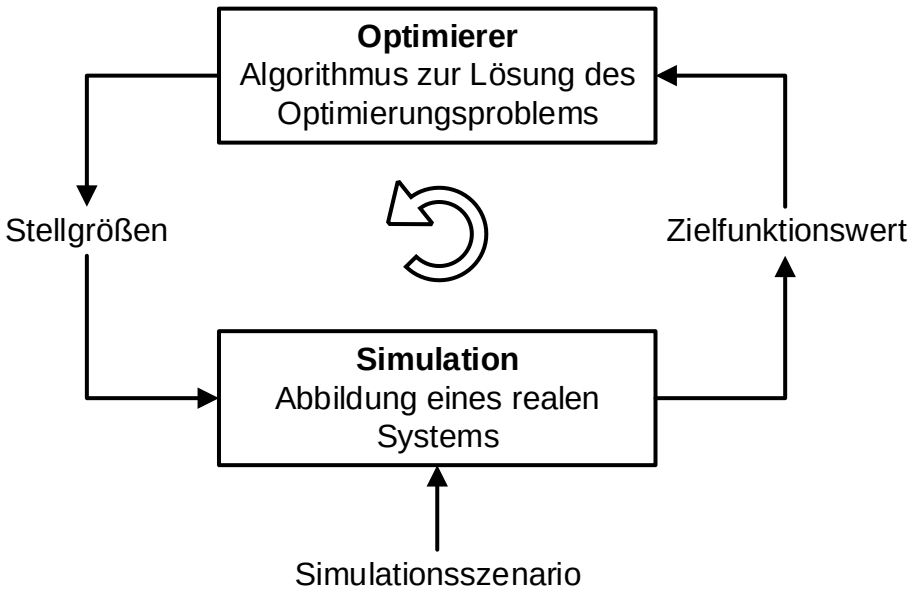

Abbildung 1: Prozess simulationsbasierte Optimierung in Anlehnung an [14]

Die Simulation bildet alle relevanten Wirkzusammenhänge eines realen Systems ab. Sie wird aus einem oder mehreren Simulationsmodellen aufgebaut. Simulationsmodelle sind abstrahierte Abbildungen der realen Komponenten des Systems. Diese Abstraktion ist notwendig um den Detailgrad des realen Systems auf ein beherrschbares Maß zu reduzieren. Grundsätzlich können Simulationen als physikalische Experimente auch ohne Computer durchgeführt werden [15]. In dieser Arbeit liegt der Fokus aber auf rechnergestützten Simulationen. Die nachfolgenden Einteilungen beziehen sich deshalb speziell auf Computersimulationen. Sie können grundsätzlich in folgende Kategorien eingeteilt werden [15]:

- statisch/dynamisch
- kontinuierlich/diskret
- deterministisch/stochastisch

Statische Simulationen bilden das System zu einem festen Zeitpunkt ab, oder die Zeit spielt in der Simulation keine Rolle. Im Gegensatz dazu bilden dynamische Simulationen das zeitliche Verhalten eines Systems ab [15]. Die nächste Unterteilung wird anhand der Systemzustände getroffen. Ändern sich die Systemzustände laufend, spricht man von einer kontinuierlichen Simulation. Kontinuierliche Systeme werden üblicherweise durch Differenzialgleichungssysteme beschrieben. Ändern sich die Systemzustände an festen Zeitpunkten, handelt es sich um eine diskrete Simulation [15]. Außerdem wird noch in deterministische und stochastische Simulationen unterteilt. Deterministische Simulationen weisen kein zufälliges Verhalten auf. Die Ausgangsgrößen sind festgelegt, sobald die Eingangsgrößen definiert werden. Die Bestimmung der Ausganggrößen kann trotzdem viel Rechenzeit in Anspruch nehmen [15]. In stochastischen Simulationen treten zufällige Ereignisse auf. Jeder Simulationsdurchlauf führt zu andere Ausgangsgrößen, selbst wenn die Eingangsgrößen konstant sind. Die Ausgangsgrößen sind somit selbst zufallsabhängig und müssen deshalb als Schätzung der wahren Modellcharakteristik betrachtet werden [15].

Die Eingangsgrößen einer Simulation werden über das Simulationsszenario festgelegt. Abhängig von den Eingangsgrößen berechnet die Simulation die Ausgangsgrößen. Für die Bewertung eines Simulationsdurchlaufs im Rahmen einer Optimierung müssen alle Ausgangsgrößen zu einem Zahlenwert zusammengefasst werden. Das geschieht über die Zielfunktion. Die Zielfunktion beinhaltet alle Zielgrößen und gewichtet diese über einen Gewichtungsfaktor. Der dadurch ermittelte Zahlenwert wird als Zielfunktionswert bezeichnet und an den Optimierer weitergeleitet (siehe Abbildung 1).

Der Optimierer variiert die Stellgrößen und beeinflusst damit die Eingangsgrößen der Simulation. Dadurch nimmt er Einfluss auf den Zielfunktionswert. Der Optimierer hat die Aufgabe die Stellgrößen so zu wählen, dass die Zielfunktion einen Extremwert annimmt (je nach Verfahren Minimum oder Maximum). Ist dieses Ziel erreicht, spricht man von einem Optimum. Der Prozess der Suche nach einem Optimum wird als Optimierung bezeichnet [14].

Im Optimierer arbeitet ein Optimierungsalgorithmus. Dieser Algorithmus definiert die Logik mit welcher der Optimierer die Stellgrößen wählt und sich so über mehrere Simulationsdurchläufe hinweg dem Optimum annähert. Eine wesentliche Anforderung an den Optimierer ist es, die Anzahl an notwendigen Simulationsdurchläufen für das Finden des Optimums gering zu halten, denn jeder Simulationsdurchlauf benötigt

Rechenzeit. Die Optimierungsverfahren können in folgende Kategorien eingeteilt werden [14]:

- deterministische Verfahren
- stochastische Verfahren

In Abbildung 2 wird der Verlauf des Zielfunktionswertes f(x), in Abhängigkeit von der Stellgröße x, dargestellt. Er weist zwei lokale Optima und ein globales Optimum auf. In diesem Fall ist das Optimum erreicht, wenn der Zielfunktionswert maximal wird. Jeder Punkt auf f(x) stellt eine mögliche Lösung des Optimierungsproblems dar.

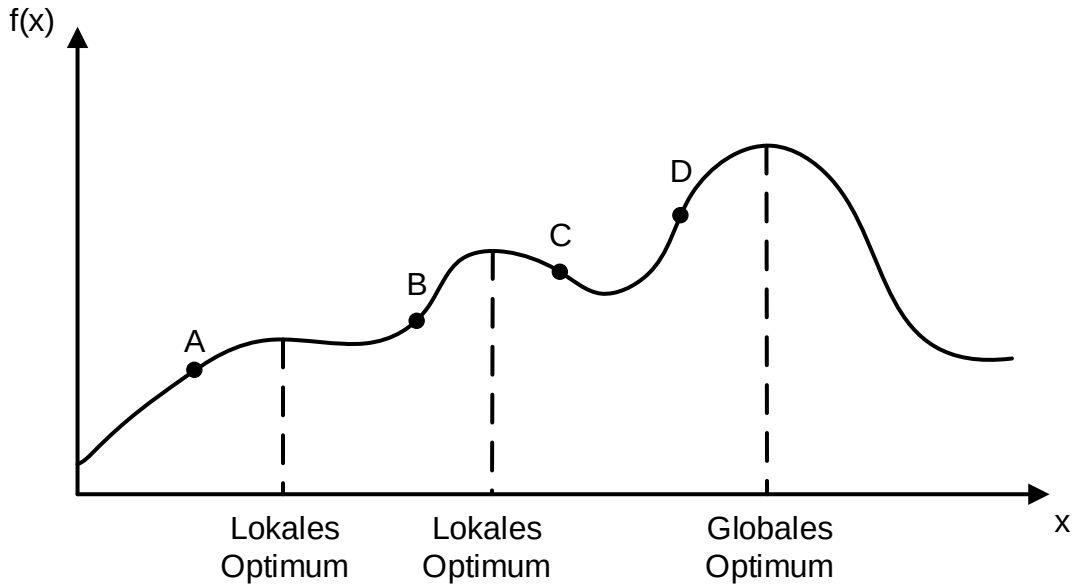

Abbildung 2: Lokale und globale Optima der Zielfunktion f(x) in Anlehnung an [14]

Deterministische Optimierungsverfahren wählen zu Beginn einen Startwert für die Stellgröße x und die Simulation ermittelt den zugehörigen Zielfunktionswert f(x). Dann wird die Stellgröße variiert und überprüft, ob sich der Zielfunktionswert erhöht oder vermindert hat. Der Wert der Stellgröße, der den Zielfunktionswert am stärksten in Richtung Optimum bewegt hat, wird als neuer Startwert verwendet [14]. So nähert sich der Zielfunktionswert langsam einem Optimum. Es lässt sich aber nicht feststellen ob es sich um das globale Optimum handelt, oder ob sich das Verfahren nur einem lokalen Optimum nähert. Wird in Abbildung 2 mit einem deterministischen Verfahren Startpunkt A, B oder C gewählt, kann das globale Optimum nicht erreicht werden. Das gelingt nur mit Startpunkt D. Ähnliche Effekte treten auf, wenn der Lösungsraum zerklüftet ist. Auf diese Problematik wird in Kapitel 2.3.6 näher eingegangen.

Im Gegensatz zu deterministischen Verfahren verteilen stochastische Verfahren die Startwerte der Stellgröße x zufällig. Die Startwerte werden durch die Zielfunktion bewertet. Die Werte der Stellgröße x, die die besten Zielfunktionswerte liefern, werden durch zufällige Wahl neuer Werte für x in diesem Bereich näher untersucht [14]. Der

Lösungsraum wird breiter durchsucht und die Gefahr, dass ein lokales Optimum als globales Optimum bestimmt wird, sinkt. Ausgeschlossen werden kann dieses Verhalten aber auch mit stochastischen Verfahren nicht. Eine Untergruppe der stochastischen Verfahren sind die evolutionären Verfahren. Diese Verfahren werden in Kapitel 2.3.2 näher betrachtet.

2.2 Simulation und Modellbildung

Die Simulation hat die Aufgabe anhand von Simulationsmodellen das Verhalten der realen Kälteanlage mit ausreichender Genauigkeit vorherzusagen. In Abbildung 3 wird die schrittweise Vorgehensweise bei der Erstellung einer Simulation von der Systemanalyse bis zu den Experimenten an der fertigen Simulation dargestellt. In diesem Kapitel werden die ersten zwei Punkte der Abbildung behandelt. Es wird die Strategie bei der Erstellung des Konzeptmodells beschrieben und eine Literaturrecherche zu bestehenden formalen Modellen für die einzelnen Simulationsbestandteile durchgeführt. Die Implementierung wird in Kapitel 3 und die Experimente werden in Kapitel 4 beschrieben.

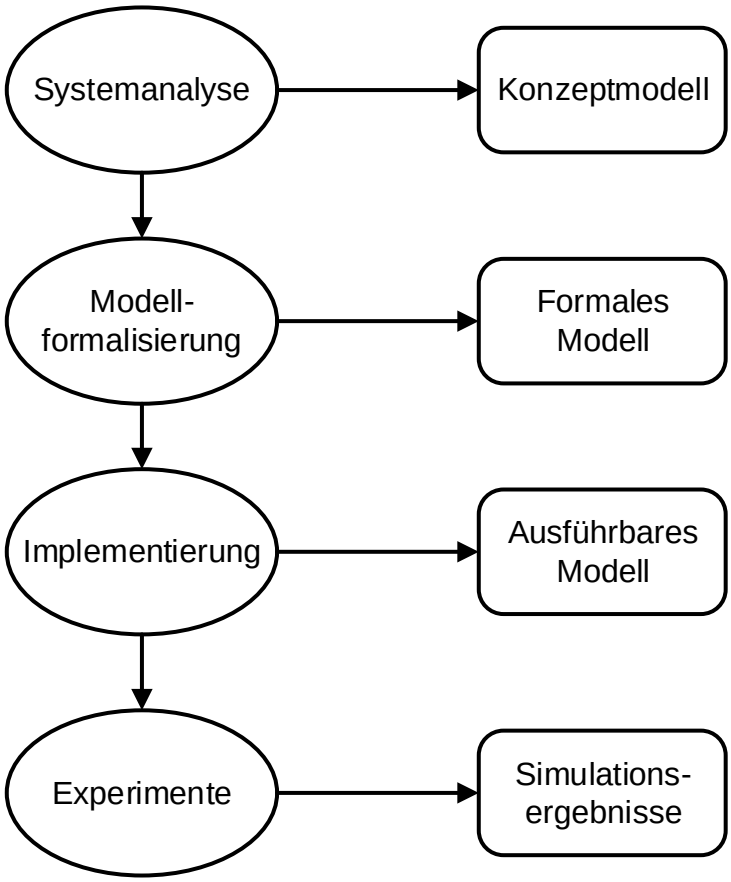

Abbildung 3: Vorgehensweise bei Erstellung der Simulation in Anlehnung an [16]

Die Modellbildung gliedert sich in die ersten 3 Phasen, der in Abbildung 3 dargestellten Vorgehensweise: Systemanalyse, Modellformalisierung und Implementierung. Das Konzeptmodell ist das Resultat einer Systemanalyse und stellt alle Zusammenhänge,

Funktionen und Wechselwirkungen des abzubildenden Systems dar [16]. Aufgrund der Komplexität des Gesamtsystems wird zur Erstellung des Konzeptmodells eine Methode benötigt, die eine Modularisierung des Systems in einzelne Subkomponenten erlaubt. Durch die Modularisierung wird die Anpassbarkeit des Simulationsmodells an unterschiedliche Kälteanlagen gewährleistet. Das Modell wird skalierbar.

In der Literatur existieren bereits Ansätze, die ein Produktionssystem modular umsetzen und in eine generische Simulationsumgebung implementieren, um so die Produktionsplanung und -steuerung zu optimieren [17]. Durch die Modularisierung kann die Simulation einfach angepasst werden. Die Autoren merken an, dass durch die generische Simulationsumgebung keine Bindung an eine bestimmte Simulationssoftware gegeben ist. Die Möglichkeiten der Simulation werden anhand von zwei Anwendungsfällen demonstriert. Im ersten Anwendungsfall kann, abhängig vom Szenario, der Energieverbrauch um bis 38% gesenkt werden. Dabei ist anzumerken, dass der Produktionsoutput ebenfalls gesenkt wird. Somit ergibt sich im besten Fall eine Energieeffizienz von 130% im Vergleich zum nicht optimierten Fall. Im zweiten Anwendungsfall kann eine Energieeffizienz von 135% im Vergleich zum nicht optimierten Fall erreicht werden.

Ein Ansatz zur Modularisierung mittels „Cubes“ wird in [18] beschrieben. Prinzipiell können beliebige Einheiten des realen Systems zu einem Cube zusammengefasst werden. Entsprechend existieren unterschiedliche Kategorien von Cubes:

- Maschine, Produktionsprozess
- Gebäude
- Energiesysteme
- Logistik

Für den Anwendungsfall der Kälteanlage sind Cubes der Kategorie „Energiesysteme“ besonders interessant. Wichtig bei einem Cube ist, dass er klar definierte Systemgrenzen aufweist. Über diese Systemgrenzen hinweg können Material (Werkstücke, Güter), Energien (elektrisch, thermisch, etc.) oder Informationen (Steuerungssignale, Messdaten) über Schnittstellen zu anderen Cubes fließen. In Abbildung 4 wird ein generischer Cube dargestellt.

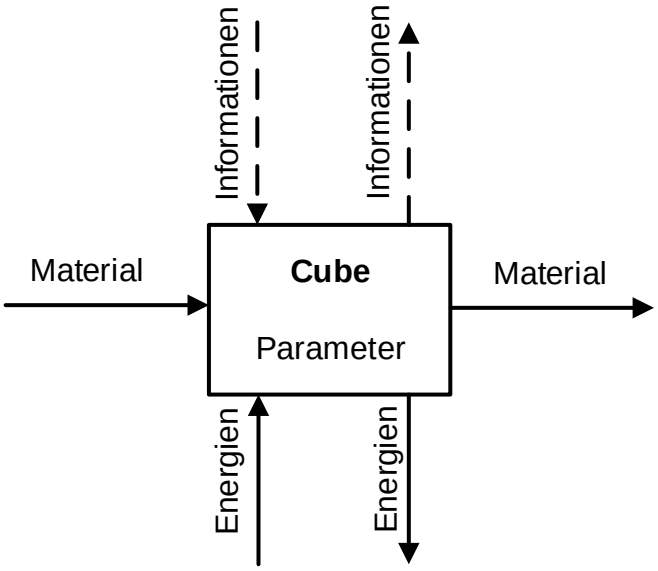

Abbildung 4: Generischer Cube in Anlehnung an [18]

Der Cube bildet das physikalisches Verhalten einer Systemkomponente des realen Systems ab, wobei durch die Schnittstellen Cubes mit unterschiedlichen Modellansätzen oder Abstraktionsgraden miteinander verbunden werden können [18]. Zur näheren Beschreibung des Verhaltens des Cube werden Parameter verwendet. Diese Parameter können sowohl physikalischen Größen entsprechen (z.B. Nennleistung, Nennkapazität, Temperaturen), wie auch Größen, die das Verhalten des Cube steuern (z.B. Art der Regelung, Wärmerückgewinnungssystem). Die Forderung nach leichter Anpassbarkeit des Modells an verschiedene Kälteanlagen wird mit dem Cube-Ansatz erfüllt. Im Konkreten bedeutet das, dass die Anzahl der Kältemaschinen und die Anzahl der angeschlossenen Kalt- und Kühlwasserkreisläufe leicht variiert werden können. Der Cube-Ansatz wird deshalb für das nachfolgende Konzeptmodell gewählt.

2.2.1 Konzeptmodell

Das Konzeptmodell der Kälteanlage wird in Abbildung 5 dargestellt. Der Austausch von Energien oder Informationen zwischen den Cubes wird als Fluss bezeichnet und durch Verbindungslinien symbolisiert, wobei Energieflüsse als Volllinie und Informationsflüsse als strichlierte Linie dargestellt werden. Nachfolgend werden die wichtigsten Zusammenhänge beschrieben.

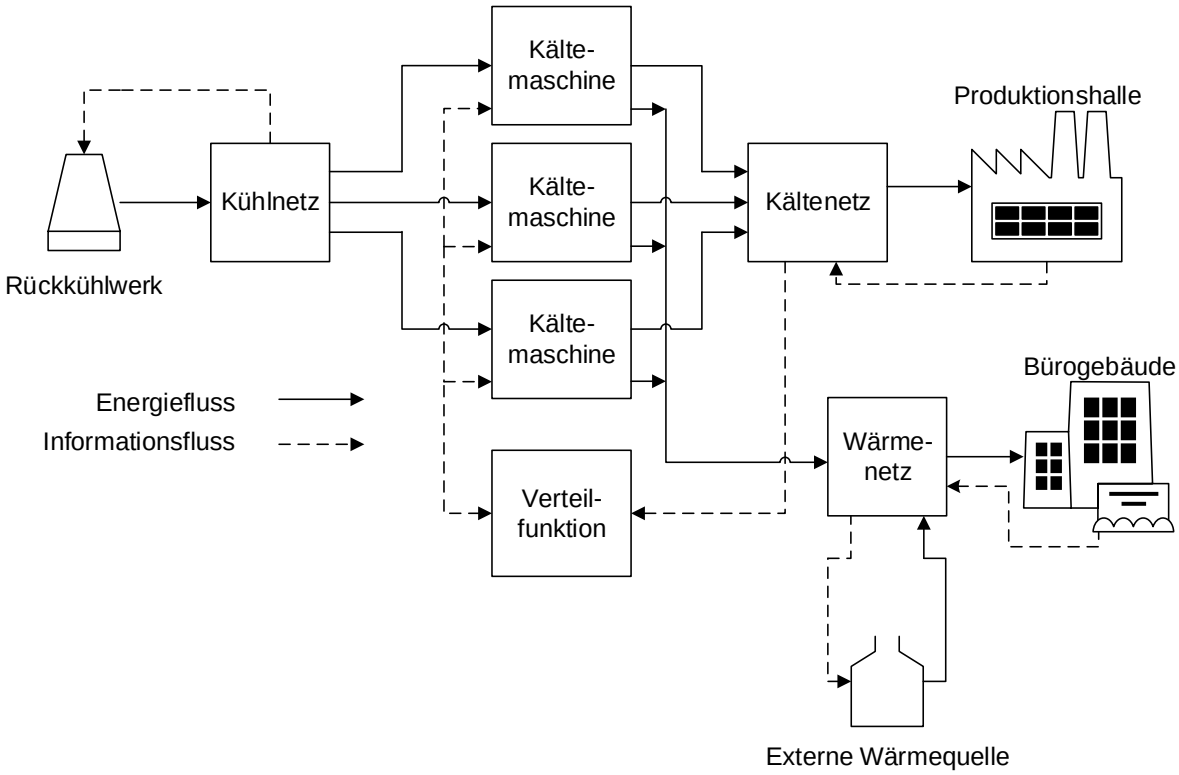

Abbildung 5: Konzeptmodell der Kälteanlage

Die erste Eingangsgröße für die Simulation ist der Kältebedarf der Produktionshalle. Diesen Bedarf zu decken ist die Hauptaufgabe der Kälteanlage. Die Produktionshalle fordert zur Deckung des Bedarfs die Kälteleistung $\dot{Q}_{k,b}$ vom Kältenetz an. Kälteleistungen sind im Modell negative Wärmeströme. Die angeforderte Kälteleistung $\dot{Q}_{k,b}$ ist als Informationsfluss von der Produktionshalle zum Kältenetz dargestellt. Das Kältenetz deckt diesen Bedarf, indem es die Kälteleistung $\dot{Q}_k$ an die Produktionshalle abgibt. Dies ist als Energiefluss dargestellt. Dadurch verringert sich der Energieinhalt E des Kältenetzes. Das Kältenetz reagiert darauf, indem es Kälteleistung von der Verteilfunktion anfordert.

Die Verteilfunktion hat die Aufgabe den Gesamtkältebedarf $\dot{Q}_{k,b}$ auf die einzelnen Kältemaschinen aufzuteilen. Diese Aufteilung erfolgt anhand eines Prioritätsvektors. Um eine korrekte Aufteilung auf die Maschinen durchführen zu können, senden die Kältemaschinen ihre aktuellen Kapazitäten an die Verteilfunktion. Die Kapazität einer Kältemaschine ist der maximale Wärmestrom $\dot{Q}_{max}$, der dem Fluid des Kältenetzes von einer Maschine entzogen werden kann. Es erfolgt also ein beidseitiger Informationsaustausch zwischen Verteilfunktion und Kältemaschinen. Sollte eine Maschine ausgefallen sein oder still stehen wird diese Information ebenfalls übertragen, indem eine Kapazität von 0 übermittelt wird. Die Kältemaschinen decken dann den von der Verteilfunktion vorgegeben Kältebedarf und ein Energiefluss von den Kältemaschinen an das Kältenetz entsteht.

Verfügt die Kältemaschine über ein Wärmerückgewinnungssystem, wird der dadurch rückgewonnene Wärmestrom $\dot{Q}_{wrg}$ an das Wärmenetz abgegeben. Das Wärmenetz deckt den Bedarf an Wärme, z.B. für die Heizung von Bürogebäuden. Der Bruttowärmebedarf $Q_{wb,brutto}$ wird über die Bürogebäude aufgebracht und ist, neben dem Kältebedarf, eine zweite Eingangsgröße für die Simulation. Reicht die Wärme aus der Wärmerückgewinnung der Kältemaschinen nicht für die Deckung des Bedarfs an Wärme aus, wird zusätzliche Wärme $Q_{wb,netto}$ von einer externen Wärmequelle (z.B. Fernwärmenetz) angefordert. Die externe Wärmequelle stellt unbegrenzte Wärme zur Verfügung, sodass der Bedarf sicher gedeckt werden kann.

In Abbildung 6 werden die Wärmeströme und die elektrische Leistungsaufnahme einer Kältemaschine abgebildet. Die Kältemaschine gibt die Kälteleistung $\dot{Q}_k$ an das Kältenetz ab, indem dem Kaltwasser auf Verdampferseite Wärme entzogen wird und bezieht die Kühlleistung $\dot{Q}_c$ vom Kühlnetz, indem die Wärme auf Kondensatorseite an das Kühlwasser abgegeben wird. Sie hat also einen Kühlleistungsbedarf $\dot{Q}_{c,b}$, der vom Kühlnetz gedeckt wird. Gleichzeitig bezieht sie elektrische Leistung P_{el} vom Stromnetz. Zur näheren Beschreibung der Funktionsweise siehe Kapitel 2.2.2.

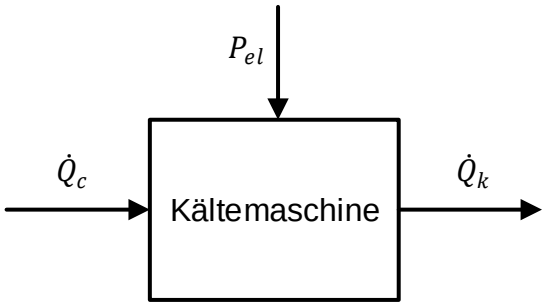

Abbildung 6: Wärmeströme der Kältemaschine in Anlehnung an [19]

Das Kühlnetz stellt die Kühlleistung $\dot{Q}_c$ für die Kältemaschinen zur Verfügung. Das Fluid des Kühlnetzes erwärmt sich dadurch. Um diese Temperaturerhöhung wieder auszugleichen, ist das Kühlnetz an ein Rückkühlwerk angeschlossen. Das Rückkühlwerk kann maximal den Wärmestrom $\dot{Q}_{c,max}$ an die Umgebung abgeben und besitzt somit begrenzte Kapazität. Sollten diese überschritten werden steigt die Temperatur im Kühlnetz und Wirkungsgrade und Kapazitäten der Kältemaschinen ändern sich.

Der gleiche Effekt tritt auf, wenn zu viel Kälteleistung vom Kältenetz angefordert wird. Die Kältemaschinen können den Bedarf nicht mehr decken und die Temperaturen im Kältenetz steigen. Die initialen Temperaturen von Kälte- und Kühlnetz sind ebenfalls Eingangsgrößen für die Simulation. Während der Simulation werden die Temperaturen von den Netzen über einen integrierten Regler konstant gehalten, solange der Bedarf an Kälte- und Kühlleistung gedeckt werden kann.

2.2.2 Formale Modelle

In diesem Kapitel werden die einzelnen Komponenten des Konzeptmodells formal beschrieben. Dabei wird zuerst ein Überblick über die Einteilung von Modellen in White Box, Grey Box und Black Box Modelle gegeben. Dann wird eine Literaturrecherche durchgeführt um bereits existierende Modelle zu finden. Diese werden dann detailliert beschrieben und die wichtigsten mathematischen Zusammenhänge erläutert. Die formalen Modelle werden im Implementierungsteil (Kapitel 3) in die Simulationssoftware AMESIM implementiert.

Kältemaschinenmodell

Über die Art der Verdichtung des Kühlmediums könne zwei Arten von Kältemaschinen unterschieden werden. Während bei Adsorptionskältemaschinen zur Verdichtung ein thermischer Verdichter eingesetzt wird, geschieht dies bei einer Kompressionskältemaschine über einen mechanischen Verdichter [20]. Die nachfolgenden Beschreibungen und Erläuterungen beziehen sich speziell auf Kompressionskältemaschinen mit mechanischem Verdichter.

In [21] werden existierende Modelle von Kältemaschinen in zwei verschiedene Kategorien eingeteilt. Der ersten Kategorie werden Modelle zugeordnet, die empirisch ermittelt werden (Black Box Modelling). Der zweiten Kategorie werden Modelle zugeordnet, die analytisch bestimmt werden (White Box Modelling). Ein Vorteil von empirische Modelle ist, dass sie kein tiefergehendes Wissen über die physikalischen Zusammenhänge im System benötigen. Der Nachteil besteht allerdings darin, dass die Modelle nur in jenem Bereich aussagekräftig sind, der mit Messdaten abgedeckt werden kann. Extrapolation außerhalb dieses Bereichs führt in der Regel zu großen Fehlern. Außerdem muss eine bestimmte Güte der Messdaten vorhanden sein, um genaue Modelle zu erhalten. Demgegenüber stehen analytische Modelle, die weitestgehend auf physikalischen Zusammenhängen beruhen. Dadurch werden weniger Messdaten als bei empirischen Modellen benötigt und eine Extrapolation außerhalb des durch Messdaten abgedeckten Bereichs ist möglich. Analytische Modelle benötigen ein gutes Verständnis der physikalischen Zusammenhänge im System. Dieses Wissen ist nicht immer verfügbar. An dieser Stelle soll eine weitere Kategorie eingeführt werden: Grey Box Modelling. Unter diesem Begriff versteht man die Verwendung von empirischen Daten in Kombination mit dem Wissen über physikalische Vorgänge in einem Modell [22].

In [22] wird argumentiert, dass umfassendes Prozesswissen bei komplexen, realen Vorgängen selten vorhanden ist. Es gibt immer Unsicherheitsfaktoren, die das System beeinflussen. Diese Unsicherheitsfaktoren können durch Experimente identifiziert und bei der Modellbildung berücksichtigt werden. Der Autor sieht deshalb Grey Box Modelle als gute Alternative zu rein empirischen oder analytischen Modellen.

Ein datenbasiertes Modell einer Kompressionskältemaschine wird in [23] vorgestellt. Es beruht auf 3 Polynomkurven, die empirisch bestimmt werden. Die Berechnung der elektrischen Leistungsaufnahme des Verdichters wird mit Größen durchgeführt, die eine physikalische Bedeutung haben, weshalb das Modell der Kategorie Grey Box Modelling zugeordnet werden kann [21]. Das Modell beruht auf der Annahme, dass die Durchflussraten des Kältemittels durch Verdampfer und Kondensator konstant sind. Für Maschinen mit stark variablen Durchflussraten eignet sich das Modell nicht. Das Modell errechnet seine Outputs aus unterschiedlichen stationären Zuständen. Für die Beschreibung dieser Zustände werden algebraische Gleichungen verwendet. Differenzialgleichungen sind nicht notwendig. Das bedeutet weniger Rechenaufwand für den Differenzialgleichungslöser in der späteren Softwareimplementierung. Außerdem werden in [23] verschiedene Möglichkeiten der Parametrisierung erläutert. Auf die Parametrisierung wird in Kapitel 2.2.3 näher eingegangen.

Da das gesamte Simulationsmodell aus einzelnen Cubes aufgebaut wird, muss ein Cube identifiziert werden, der dem Verhalten einer Kältemaschine entspricht. Der in [24] beschriebene Cube „Energiewandler“ erfüllt diese Anforderung. Dieser Cube wandelt unterschiedliche Energieformen ineinander um. Die Inputs des Cube sind die Leistung, die für die Umwandlung erforderlich ist, sowie die angeforderte Energie. Als Output gibt der Cube die umgewandelte Energie ab und ermittelt den dafür notwendigen Leistungsbedarf. Der Cube kommt ebenfalls ohne Differenzialgleichungen aus.

Das Verhalten hängt im Wesentlichen von 2 Funktionen ab: Der Kapazitätsfunktion und der Effizienzfunktion. Die Kapazitätsfunktion definiert die maximale Kapazität an umgewandelter Energie, die der Cube bereitstellen kann. Die Effizienzfunktion legt den Wirkungsgrad fest, mit dem die Energieumwandlung stattfindet. Beide Funktionen können von beliebig vielen Faktoren abhängig sein. Als Ansatz für die in [24] nicht näher bestimmten Funktionen werden wie in [23] Polynomfunktionen gewählt, die die oben beschriebene Kapazitäts- und Effizienzfunktion in Abhängigkeit von den Vor- und Rücklauftemperaturen der an der Maschine angeschlossenen Wasserkreisläufe, sowie der Kältelast der Maschine darstellen. Die Funktionen sind als quadratische Gleichungen aufgebaut und die Regressionskoeffizienten können mittels linearer Regression bestimmt werden. In [25] wird ein im Vergleich zu [23] weiterentwickeltes Modell verwendet, das aber ebenso auf 3 Polynomfunktionen beruht. Anstatt einer Referenzleistung P_{ref} wird in [25] eine Referenzleistungszahl COP_{ref} zur Berechnung der elektrischen Leistungsaufnahme verwendet. Außerdem wird die Funktion zur Berechnung der Teillastkennlinie $EIRFPLR$ erweitert (Verwendung von 7 Regressionskoeffizienten, anstatt von 3). Nachfolgend wird auf das in [25] beschriebene Modell näher eingegangen.

Für die nachfolgenden Abschnitte ist es notwendig, die relevanten Temperaturen der Wasserkreisläufe genau zu beschreiben. Zu diesem Zweck wird in Abbildung 7 der prinzipielle Aufbau einer Kompressionskältemaschine mit angeschlossenen Kreisläufen dargestellt. Nachfolgend wird kurz auf die Funktionsweise einer Kompressionskältemaschine eingegangen.

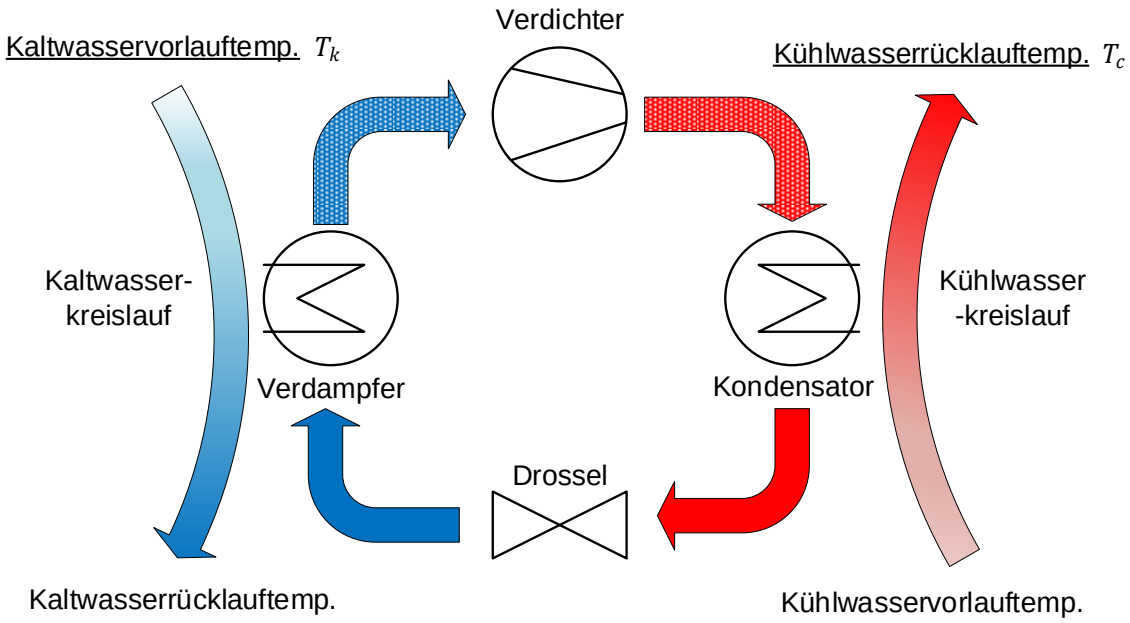

Abbildung 7: Prinzipdarstellung Kompressionskältemaschine in Anlehnung an [20]

Ein in der Kälteerzeugung häufig verwendetes Verfahren ist der Kompressionskreisprozess. Eine Kompressionskältemaschine nutzt diesen Prozess zur Kälteerzeugung [20]. Das Kühlmedium zirkuliert in der Maschine und wird abwechselnd verdampft und kondensiert. Dem Kaltwasser wird durch das Verdampfen des Kühlmediums im Verdampfer Wärme entzogen (Kälteleistung $\dot{Q}_k$). Es findet ein Phasenübergang des Kühlmediums von flüssig zu gasförmig statt. Die Kaltwasservorlauftemperatur ist deswegen höher als die Kaltwasserrücklauftemperatur. Anschließend wird der Druck des Mediums durch einen Verdichter erhöht. Der mechanische Verdichter bezieht beim Verdichtungsprozess elektrische Leistung P_{el}. Im Kondensator gibt das Medium beim Kondensieren die Wärme an das Kühlwasser wieder ab (Kühlleistung $\dot{Q}_c$). Das Kühlmedium liegt nun wieder in flüssiger Phase vor. Die Kühlwasservorlauftemperatur ist deswegen niedriger als die Kühlwasserrücklauftemperatur. Anschließend wird das Medium in einer Drossel entspannt und gelangt wieder zum Verdampfer [20]. Die für das Modell relevanten Temperaturen (Kaltwasservorlauftemperatur T_k und Kühlwasserrücklauftemperatur T_c) der beiden Wasserkreisläufe werden in Abbildung 7 unterstrichen dargestellt.

Das in [25] beschriebene Modell errechnet die erforderliche elektrische Leistung P_{el} des mechanischen Verdichters der Kältemaschine in Abhängigkeit von T_k, T_c und der Kälteleistung $\dot{Q}_k$. Die erforderliche elektrische Leistung des Verdichters P_{el} der Kältemaschine errechnet sich nach Gleichung (2.1) und Gleichung (2.2).

$$P_{el} = \frac{\dot{Q}_{max}}{COP_{ref}} * EIRFT * EIRFPLR \tag{2.1}$$

$$\dot{Q}_{max} = \dot{Q}_{ref} * CAPFT \tag{2.2}$$

COP_{ref} ist die Leistungszahl der Kältemaschine bei Referenzbedingungen. Die maximale Kälteleistung $\dot{Q}_{max}$ berechnet sich aus der verfügbaren Kühlkapazität bei Referenzbedingungen $\dot{Q}_{ref}$ und der Kapazitätskennlinie $CAPFT$. Nachfolgend sind die Gleichungen für die Kapazitätskennlinie $CAPFT$ (2.3), die Wirkungsgradkennlinie $EIRFT$ (2.4) und die Teillastkennlinie $EIRFPLR$ (2.5) aufgeführt:

$$CAPFT = a_0 + a_1 T_k + a_2 T_k^2 + a_3 T_c + a_4 T_c^2 + a_5 T_k T_c \tag{2.3}$$

$$EIRFT = b_0 + b_1 T_k + b_2 T_k^2 + b_3 T_c + b_4 T_c^2 + b_5 T_k T_c \tag{2.4}$$

$$EIRFPLR = c_0 + c_1 T_c + c_2 T_c^2 + c_3 PLR + c_4 PLR^2 + c_5 T_c PLR + c_6 PLR^3 \tag{2.5}$$

Die derzeitige Auslastung der Maschine wird in [25], analog zu [24], über PLR (Part Load Ratio) bestimmt. Sie errechnet sich nach Gleichung (2.6).

$$PLR = \frac{\dot{Q}_k}{\dot{Q}_{max}} \tag{2.6}$$

Die Koeffizienten a_i, b_i und c_i der Kennlinien (2.3) bis (2.5) können entweder durch Verwendung von Herstellerdaten, oder durch Verwendung von Messdaten bestimmt werden. Auf die Parametrisierung wird in Kapitel 2.2.3 eingegangen.

Manche Kältemaschinen verfügen über ein Wärmerückgewinnungssystem. Solche Maschinen werden in [25] nicht berücksichtigt. Das Modell muss deswegen um Gleichung (2.7) erweitert werden. Sie beschreibt den Zusammenhang zwischen dem Wärmestrom durch Wärmerückgewinnung $\dot{Q}_{wrg}$ und der Kälteleistung $\dot{Q}_k$ der Maschine. Die Bestimmung des Wärmerückgewinnungswirkungsgrad η_1 erfolgt mit der Methode der kleinsten Quadrate.

$$\dot{Q}_{wrg} = \eta_1 \dot{Q}_k \tag{2.7}$$

Energienetzmodell

Das Verhalten der Wasserkreisläufe wird durch das Energienetzmodell abgebildet. Somit werden die für das Kältemaschinenmodell relevante Temperaturen dynamisch durch die Simulation berechnet. Der in [24] erläuterte Cube „Energienetz“ dient als Grundlage. Dieser beschreibt ein Netz mit oder ohne Speicherkapazität, das Energie von einem oder mehreren Erzeugern anfordern und an einen oder mehrere Bezieher abgeben kann. Werden als Erzeuger die Kältemaschinen verwendet und als Bezieher die Produktionsanlagen, die durch das Kaltwasser gekühlt werden, bildet der Cube den Kaltwasserkreislauf ab. Beim Kühlwasserkreislauf wird derselbe Cube verwendet, allerdings sind in diesem Fall die Kältemaschinen die Bezieher und die Erzeuger sind Modelle, die mit der Umgebung interagiert und die Abwärme an die Umgebung abführen (Rückkühlwerk-Modell). Außerdem muss ein Regler in den Cube integriert werden, der die von Erzeugern angeforderte Leistung regelt und so den Speicherstand auf dem Niveau des Soll-Speicherstandes hält. Im Falle des Kaltwasserkreislaufs wird vom Kaltwasser durch die Kältemaschine ein Wärmestrom abgeführt. Auf der anderen Seite wird dem Kaltwasser von den Produktionsanlagen ein Wärmestrom zugeführt. Die Temperatur des Fluids ändert sich daraufhin. Das Verhalten eines einfachen Kalt- bzw. Warmspeichers kann durch eine Enthalpiebilanz bestimmt werden. Die zeitliche Änderung der Enthalpie aufgrund der Wärmeströme ist in Gleichung (2.8) gegeben [19].

$$\frac{dH}{dt} = \dot{Q}_{zu} - \dot{Q}_{ab} \tag{2.8}$$

Es wird angenommen, dass es sich um eine isobare Zustandsänderung handelt und somit gilt dp = 0. Die Enthalpie des Fluids ändert sich genau um die zugeführten und abgeführten Wärmeströme. Wird weiters von reibungsfreier Strömung ausgegangen ergibt sich unter der Annahme einer konstanten spezifischen Wärmekapazität c im Temperaturintervall der in Gleichung (2.9) dargestellte Zusammenhang [19].

$$\frac{dH}{dt} = mc\frac{dT}{dt} \tag{2.9}$$

Eventuell auftretende konvektive Verluste über die Oberfläche der Rohrleitungen werden nicht berücksichtigt. Da es sich bei den Speichern um Wasserkreisläufe handelt lässt sich die Speicherkapazität des Netzes nach Gleichung (2.10) aus der Masse des zirkulierenden Fluids, sowie der spezifischen Wärmekapazität des Fluids errechnen.

$$C = mc \tag{2.10}$$

Der Speicherstand des Netzes entspricht der Differenz der Enthalpie des Fluids im Kreislauf H_1 und der Enthalpie des Fluids bei Umgebungsbedingungen H_0 (siehe Gleichung (2.11)).

$$E = H_1 - H_0 \tag{2.11}$$

Werden die Gleichungen (2.10) und (2.11) in Gleichung (2.9) eingesetzt und auf die Temperatur T_1 umgeformt ergibt sich mit Gleichung (2.12) eine Bestimmungsgleichung für die Temperatur des Fluids.

$$T_1 = \frac{E}{C} + T_0 \tag{2.12}$$

Tritt keine Temperaturdifferenz des Fluids mit der Umgebung auf ist der Speicher leer und E = 0. Es steht keine nutzbare Temperaturdifferenz mehr zur Verfügung.

Weitere Modelle

Das gesamte Simulationsmodell ist aus weiteren Modellen aufgebaut. Einige dieser Modelle geben nicht direkt ein beobachtbares physikalisches Verhalten wieder, sondern sind durch die Implementierung in die Simulationssoftware oder durch die Anbindung der Simulation an den Optimierer notwendig geworden. Auf diese implementierungsspezifischen Modelle wird detailliert im Implementierungsteil eingegangen. An dieser Stelle wird auf Kapitel 3 verwiesen.

2.2.3 Parametrisierung

Durch die Parametrisierung werden die Modellparameter der Simulationsmodelle bestimmt. Die Kältemaschinenmodelle haben als zentrales Element der Simulation große Auswirkungen auf alle daran angeschlossenen Komponenten und somit auf die gesamte Simulation. Deshalb wird in diesem Kapitel speziell auf die Parametrisierung der Kältemaschinenmodelle eingegangen. Um eine genaue Vorhersage der elektrischen Leistungsaufnahme der Kältemaschinen über den mechanischen Verdichter zu gewährleisten, müssen die Regressionskoeffizienten a_i, b_i und c_i der Kennlinien möglichst genau für jede Maschine bestimmt werden.

In [23] werden zwei verschiedene Möglichkeiten der Parametrisierung mittels Herstellerdaten oder Kurzzeit-Messdaten erläutert. Die lineare Regressionsmethode wird der Referenzkurvenmethode gegenübergestellt. Die Referenzkurvenmethode setzt einen Katalog an Referenzkurven voraus. Es muss eine Vorauswahl getroffen werden welche Kurven am ehesten denen der zu parametrisierenden Maschine entsprechen. Für jede Referenzkurve werden die Referenzkapazität und die Referenzleistung berechnet. Dann wird der Fehler bei der Vorhersage der elektrischen Leistungsaufnahme für jede Kurve bestimmt. Die Referenzkurve mit dem geringsten Fehler wird dann verwendet.

Bei der linearen Regressionsmethode wird eine quadratische Gleichung als Ansatz gewählt und die Regressionskoeffizienten mit der Methode der kleinsten Quadrate bestimmt. Voraussetzung für diese Methode ist, dass Daten für jenen Betriebsbereich

der Maschine vorhanden sind, der später simuliert werden soll. Eine Extrapolation auf andere Betriebsbereich führt zu großen Fehlern.

Die lineare Regressionsmethode erweist sich als genauere der beiden Methoden, allerdings merkt der Autor an, dass die richtige Auswahl der Daten eine wichtige Grundvoraussetzung für eine genaue Parametrisierung ist. Außerdem sinken die Genauigkeiten bei Extrapolation auf durch Daten nicht abgedeckte Betriebsbereiche. Die lineare Regressionsmethode setzt einen entsprechend großen Datensatz voraus. Die Referenzkurvenmethode kann allerdings auch dann verwendet werden, wenn die vorhandenen Daten für die Bestimmung der Regressionskoeffizienten nicht ausreichend vorhanden sind.

Sollten Herstellerdaten nicht verfügbar sein muss die datenbasierte Parametrisierung anhand von Messdaten vorgenommen werden. In der Literatur werden vielversprechende Ergebnisse mit dieser Methode erzielt. In [25] wird eine datenbasierte Parametrisierung anhand von Messdaten mittels linearer Regression durchgeführt. Die Daten werden über einen Zeitraum von 28 Tagen alle 15 Minuten erfasst. Der relative Fehler bei der Vorhersage der elektrischen Leistungsaufnahme der Kältemaschinen liegt bei 0,6% für Kältemaschine 1 und bei 3,7% für Kältemaschine 2 bezogen auf den Validierungsdatensatz. Diese Methodik wird nachfolgend für die Parametrisierung von 12 Kältemaschinen verwendet.

Um die Regressionskoeffizienten a_i, b_i und c_i zu bestimmen werden die elektrische Leistungsaufnahme P_{el} und der Verdampferwärmestrom $\dot{Q}_k$ von 12 Kältemaschinen über eine Zeitperiode von mindestens einem Monat aufgezeichnet. Zusätzlich werden von 4 Maschinen die Eingangstemperaturen auf Verdampferseite T_k und die Ausgangstemperaturen auf Kondensatorseite T_c aufgezeichnet. Die Daten werden alle 30 Sekunden erfasst. Bevor mit der Bestimmung der Regressionskoeffizienten begonnen wird, werden die Daten analysiert und Datensätze entfernt, die die Genauigkeit negativ beeinflussen. Das sind einerseits alle Datensätze, die den Wert 0 haben und andererseits alle Datensätze, die Bedingung (2.13) oder (2.14) erfüllen, wobei y_i der aktuelle Messwert, y_{mean} der Mittelwert der Messwerte des Datensatzes und σ die Standardabweichung der Messwerte des Datensatzes ist.

$$y_i < (y_{mean} - 3\sigma) \tag{2.13}$$

$$y_i > (y_{mean} - 3\sigma) \tag{2.14}$$

Bei den Temperaturmessungen werden zusätzlich alle Datensätze entfernt, die außerhalb der in Tabelle 1 festgelegten Grenzen liegen, da Temperaturen außerhalb dieser Bereiche nicht plausibel sind.

	T_c [°C]	T_k [°C]
Obere Grenze	30	15
Untere Grenze	15	4

Tabelle 1: Plausibilitätsgrenzen für Temperaturmesswerte [26]

Durch die Filterung der Daten, reduziert sich Datensatzgröße zwischen 0,005% und 28,5% im Vergleich zu den ungefilterten Daten. In Tabelle 2 werden die aufgenommenen Messgrößen und die Dateigrößen der Datensätze vor und nach der Filterung aufgeführt.

Maschine	P_{el}	$\dot{Q}_k$	T_k	T_c	Datensatzgröße	Gefilterte Datensatzgröße
1	X	X			919865	919757
2	X	X			1124472	1124415
3	X	X			1125003	1124808
4	X	X			1125671	1125611
5	X	X			915289	915143
6	X	X			1134663	1134474
7	X	X			1127933	844476
8	X	X	X	X	678718	583815
9	X	X	X	X	679007	587077
10	X	X	X	X	679250	591193
11	X	X	X	X	679451	556556
12	X	X	X	X	103660	74159

Tabelle 2: Aufgenommene Messgrößen und Dateigrößen der Datensätze [26]

Die gefilterten Daten werden in Kalibrierungs- und Validierungsdatensätze aufgeteilt. Dabei werden 90% der Daten zur Kalibrierung und 10% der Daten zur Validierung verwendet. Die Aufteilung erfolgt zufällig. Die Regressionskoeffizienten der Maschinen, bei denen Daten zu P_{el}, $\dot{Q}_k$, T_k und T_c vorliegen, werden mit der in [25] angewendeten Methoden ermittelt. Für die Maschinen, bei denen keine Temperaturmesswerte vorliegen, wird ein vereinfachtes Modell verwendet, bei dem alle temperaturabhängigen Koeffizienten 0 gesetzt werden. Die ermittelten Parameter für alle 12 Maschinen sind im Anhang aufgeführt (Kapitel 6.1).

Um eine Aussage über die Modellgenauigkeit treffen zu können, werden die Vorhersangen des Modells mit den Messwerten aus dem Validierungsdatensatz verglichen. Hierfür wird $RMSE_{rel}$ (Root Mean Square Error) nach Gleichung (2.15) gebildet [26].

$$RMSE_{rel} = \frac{\sqrt{\sum_{i=1}^{n}(y_{i,predicted} - y_i)^2}}{\sum_{i=1}^{n} y_i} * 100 \quad (2.15)$$

Die Ergebnisse der Validierung werden in Tabelle 3 dargestellt. 6 von 12 Maschinen weisen bei der Vorhersage der elektrischen Leistungsaufnahme P_{el} einen kleineren Fehler als 20% auf. Bei 3 weiteren Maschinen liegt der Fehler im Bereich von 20% bis 40% und bei 3 Maschinen liegt der Fehler über 100%. Bei der Vorhersage der von den Maschinen benötigten Kühlleistung $\dot{Q}_c$ weisen 3 Maschinen einen kleineren Fehler als 10% auf bzw. eine weitere Maschine einen Fehler von 10%. Bei 2 weitere Maschinen liegt der Fehler zwischen 20% und 40%. Alle anderen Maschinen weisen größere Fehler als 40% auf. Dabei ist anzumerken, dass die Messdaten von im Einsatz befindlichen Maschinen aus einer real existierenden Kälteanlage stammen. Herstellerangaben standen nicht zur Verfügung. Bei Verwendung von Herstellerdaten oder bei Messungen unter Laborbedingungen können wesentlich höhere Genauigkeiten mit der verwendeten Methodik erreicht werden [23]. Die ermittelten Parameter werden trotzdem verwendet, weil in 3 von 4 mit den Parametern untersuchten Szenarien das Einsparungspotenzial den gewichteten Modellfehler übersteigt (siehe Kapitel 4.5).

	$\boldsymbol{RMSE_{rel}}$ [%]	
Maschine	$\boldsymbol{P_{el}}$	$\boldsymbol{\dot{Q}_c}$
1	13,59	65,09
2	17,23	4,46
3	4,74	10,04
4	26,89	21,18
5	38,49	48,61
6	178,29	722,20
7	126,07	136,16
8	28,78	69,10
9	10,25	3,90
10	4,42	8,42
11	11,79	29,66
12	179,51	432,17

Tabelle 3: Relativer RMSE für P_{el} und $\dot{Q}_c$ [26]

Für die Maschinen mit Wärmerückgewinnungssystem (Maschine 1 und 5) wird zusätzlich der Wärmerückgewinnungswirkungsgrad bestimmt. Dabei ist anzumerken, dass die Messwerte der rückgewonnenen Wärme nur als Tagesmittelwerte vorliegen. Um trotzdem den Wirkungsgrad bestimmen zu können, werden die 30-Sekunden-Mittelwerte der anderen Größen ebenfalls zu Tagesmittelwerten zusammengefasst.

Dann wird der Wärmerückgewinnungsgrad mit der Methode der kleinsten Quadrate bestimmt. Durch den sehr geringen Umfang an Messdaten entfällt die Aufteilung in Kalibrierungs- und Validierungsdatensatz. Der $RMSE_{rel}$ bei der Vorhersage der rückgewonnenen Wärme liegt für Maschine 1 bei 1,02% und für Maschine 5 bei 1,36%. Diese Werte sagen aber nur etwas darüber aus, wie gut die Tagesmittelwerte der Maschinen vorhergesagt werden können. Zur Genauigkeit der Vorhersage in kleineren Auflösungen kann aufgrund fehlender Daten keine Aussage gemacht werden.

2.3 Optimierung

In der Einleitung werden einige Ansätze zur Optimierung der Betriebsstrategie von Kälteanlagen kurz beschrieben und das Einsparungspotenzial erläutert. In diesem Kapitel werden diese Ansätze detailliert behandelt. Dann werden verschiedene Möglichkeiten zur Optimierung über einen Optimierungsalgorithmus betrachtet und die Anwendbarkeit auf das vorliegende Problem überprüft. Außerdem werden verschiedene Methoden für den Umgang mit Randbedingungen untersucht.

2.3.1 Optimierungsansätze

Eine wesentliche Fragestellung für die Optimierung ist es, welche Freiheitsgrade der Optimierer erhält und wie das Optimierungsergebnis in der Praxis umgesetzt werden kann. In vielen Quellen werden folgende Ansätze verfolgt [1] - [13], [26], [27], [28]:

- Die Optimierung durch die ideale Lastverteilung auf die Maschinen: bei diesem Konzept wird die Verteilung des Gesamtkältebedarfs auf die einzelnen Kältemaschinen modifiziert um Einsparungen zu erzielen. Dieses Verfahren wird als OCL (Optimal Chiller Loading) bezeichnet.
- Die Optimierung durch die ideale Einschaltreihenfolge der Maschinen: bei diesem Konzept wird die Einschaltreihenfolge bzw. die Anzahl eingeschalteter Maschinen modifiziert um Einsparungen zu erzielen. Dieses Verfahren wird als OCS (Optimal Chiller Sequencing) bezeichnet.

Bei Kältemaschinen ist die Leistungsziffer abhängig von der Auslastung. Dieses Konzept macht sich OCL zunutze. Die Auslastung einer Kältemaschine wird über PLR (Part Load Ratio) angegeben. Sie ist der Quotient aus der Kälteleistung $\dot{Q}_k$ und der unter den gegebenen Betriebsbedingungen maximal möglichen Kälteleistung $\dot{Q}_{max}$ der Kältemaschine (siehe Gleichung (2.6), Kapitel 2.2.2).

Die Leistungszahl COP einer Kältemaschine wird nach Gleichung (2.16) berechnet [19]. Sie bildet sich aus dem Verhältnis von Kälteleistung $\dot{Q}_k$ zur elektrischen Leistungsaufnahme P_{el} der Kältemaschine.

$$COP = \frac{\dot{Q}_k}{P_{el}} \tag{2.16}$$

Eine hohe PLR bedeutet aber nicht zwangsweise eine hohe Leistungszahl. In [2] wird bei 3 untersuchten Maschinen die höchste Leistungszahl bei einer PLR von 0,3 bis 0,5 festgestellt, also im Niedriglastbereich. Beim OCL Ansatz wird versucht für jede Maschine die PLR mit der höchsten Leistungszahl zu bestimmen und die Maschine in diesem Bereich zu betreiben.

In [2] merken die Autoren an, dass es in der Praxis schwierig ist die Maschinen über PLR in den optimalen Bereich zu regeln, da PLR weder direkt gemessen, noch direkt gesteuert werden kann. Die Messung ist nur indirekt über die Messung der Durchflussmenge am Verdampfer und die Vor- und Rücklauftemperaturen möglich. Die Autoren modifizieren den Ansatz und verteilen die Kältelast abhängig von den Nennkühlkapazitäten der Maschinen durch Modifikation der Umschaltzeitpunkte. Dadurch können Energieeinsparung von 0,5% realisiert werden. Die Optimierung durch eine optimale Anzahl an Maschinen in Betrieb und eine ideale Einschaltreihenfolge (OCS) lässt sich leichter umsetzen. Hier wird jede Maschine mit maximaler Auslastung betrieben und erst wenn die angeforderte Kälteleistung die abgegebene Kälteleistung der Maschinen übersteigt wird eine neue Maschine zugeschaltet.

Die unter den gegebenen Betriebsbedingungen zur Verfügung stehende Kühlkapazität (maximale Kälteleistung $\dot{Q}_{max}$) weicht von der Nennkühlkapazität ab. Sie wird insbesondere von den Temperaturen der angeschlossenen Kalt- und Kühlwasserkreisläufe beeinflusst. In [2] wird festgestellt, dass bei 3 untersuchten Kältemaschinen die Nennkühlkapazität bei guter Wahl der Temperaturen des Kühlwasserkreislaufs um bis zu 11% überschritten werden kann. Die Autoren modifizieren den Sollwert der Kühlwassertemperatur und erzielen so eine Steigerung der zur Verfügung stehende Kühlkapazität. Zusätzlich werden, wie im ersten Ansatz, die Umschaltzeitpunkte zwischen den Maschinen modifiziert. So kann der Kältebedarf in bestimmten Szenarien durch weniger Maschinen gedeckt werden und so Einsparungen erzielt werden. Mit dieser Vorgehensweise kann der Energiebedarf um 5,3% gesenkt werden. Ein dritter in [2] vorgestellter Ansatz kombiniert beide Optimierungsstrategien. Hier liegt das Einsparungspotenzial bei 5,6% des Gesamtenergiebedarfs. In [3] wird die Optimierung einer aus 3 Kältemaschinen bestehenden Kälteanlage ebenfalls mit OCS und OCL-Ansatz durchgeführt. Mit optimierter Einschaltreihenfolge (OCS) werden bis zu 4,3% des Energiebedarfs eingespart. Bei der Optimierung der Kältelastverteilung auf die Maschinen (OCL) wird folgende Vereinfachung vorgenommen: Die maximale Kältelast wird in N gleich große Abschnitte unterteilt (z.B. 10 Abschnitte zu 10% der maximalen Kältelast). Dann wird die Kältelastverteilung auf die Maschinen für jeden dieser Abschnitte optimiert. Im zeitlichen Verlauf eines 24 Stunden Szenarios wird, abhängig von der Kältelast, der

jeweils benötigte optimierte Teilabschnitt angewendet. Die Ergebnisse werden für N = 10 und N = 20 ermittelt und mit den Ergebnissen einer Optimierung mittels OCL-Ansatz ohne dieser Vereinfachung verglichen. Es wird gezeigt, dass das Einsparungspotenzial ohne Vereinfachung mit 10,3% am höchsten ist. Bei Anwendung der Strategie mit N = 10 können 6,2% und bei N = 20 6,7% des Energiebedarfs eingespart werden. Durch die Simplifizierung wird zwar das Einsparungspotenzial reduziert, aber das Optimum kann leichter gefunden werden und die Ein- und Ausschaltvorgänge pro Maschine werden reduziert. Das erhöht die Stabilität des Systems.

Eine weitere Methodik zur Optimierung der Operationsstrategie verschiedener Kälteanlage wird in [7] vorgestellt. Es werden Modelle für Kältemaschinen, Kühltürme, Pumpen und Ventilatoren verwendet und mittels eines Optimierungsalgorithmus eine optimale Betriebsstrategie für die Kälteanlage ermittelt. Ziel ist es, das thermische Verhalten aller Komponenten und ihre Wechselwirkungen, einschließlich des elektrischen Stromverbrauchs, zu erfassen. Als Simulationssoftware wird Matlab-Simulink verwendet. Die Freiheitsgrade sind die Einschaltreihenfolge der Maschinen, die Durchflussmengen der Kalt- und Kühlwasserkreisläufe, die Temperaturen der Kreisläufe und der Luftdurchsatz in den Rückkühlwerken. Als Ziel der Optimierung wird ein minimaler elektrischer Energieverbrauch gewählt. Es werden drei verschiedene Anlagekonfigurationen untersucht. Bei der ersten Konfiguration handelt es sich um eine Anlage, die aus zwei Kältemaschinen besteht. Dabei können Energieeinsparungen von 42,5% erreicht werden. Bei der zweiten aus drei Maschinen bestehenden Konfiguration liegen die Einsparungen bei 40,3%. Bei der dritten Konfiguration besteht die Anlage aus nur einer Maschine. Dort können 20% des Energiebedarfs eingespart werden.

In [4] wird eine Technik zur Lösung des OCL-Problems vorgestellt, bei der die Komplexität des Problems reduziert wird, indem alle Kältemaschinen zu einer einzigen Maschine zusammengefasst werden. Dadurch wird die Anzahl an Freiheitsgraden reduziert. Zusätzlich wird ein thermischer Energiespeicher in die Betrachtung miteinbezogen. So kann die Kältelast nicht nur zu den energieeffizientesten Maschinen verschoben werden, sondern diese können durch den Speicher auch zu kosteneffizienten Zeiten betrieben werden. Dabei wird davon ausgegangen, dass der Strompreis über den Tag schwankt, wobei sich der Strompreis in dem Szenario der Autoren zu Lastspitzenzeiten verdoppelt. Die Einsparungen beim Energieverbrauch liegen bei 9,4% und die Einsparungen bei den Kosten bei 17,4%.

Ein simulationsbasierter Ansatz zur Lösung solcher Optimierungsprobleme ist weit verbreitet. Da viele Arbeiten in diesem Bereich unterschiedliche Optimierungsalgorithmen anwenden wird nachfolgend eine Einordnung und Beschreibung von diesen Algorithmen vorgenommen.

2.3.2 Evolutionäre Algorithmen

Evolutionäre Algorithmen sind metaheuristische, stochastische Suchmethoden, die auf biologischen Evolutionsprozessen in der Natur beruhen [29]. Im Vergleich zu deterministischen Algorithmen (z.B. Simplex-Verfahren, Newton-Raphson-Verfahren), weisen stochastische Algorithmen immer einen gewissen Zufallsfaktor auf. Es gibt keine Garantie dafür, dass die gefundene Lösung auch dem Optimum entspricht. Für gewöhnlich nähern sich stochastische Algorithmen mit zunehmender Rechenzeit einer optimalen Lösung an [30].

Evolutionäre Algorithmen weisen im Vergleich zu konventionellen Optimierungsalgorithmen einige interessante Eigenschaften auf [31]:

- Es sind keine zeitlichen Ableitungen der Zielfunktion erforderlich.
- Es können Optimierungsprobleme mit diskreten und kontinuierlichen Parametern gelöst werden.
- Sie können leicht parallelisiert werden.

Nachfolgend werden verschiedene evolutionären Algorithmen beschrieben und anhand von Literaturquellen ihre Anwendbarkeit auf das Optimierungsproblem erläutert.

2.3.3 Genetische Algorithmen

Eine große Untergruppe der evolutionären Algorithmen stellen die genetischen Algorithmen dar. Sie basieren auf dem darwinistischen Prinzip des Überlebens des Stärkeren und dem natürlichen Prozess der Evolution durch Rekombination und Mutation, wobei bei genetischen Algorithmen neue Generationen hauptsächlich durch Rekombination erzeugt werden und die Mutation eine untergeordnete Rolle spielt [29]. Genetischen Algorithmen sind weit verbreitet und werden bei vielen komplexen Optimierungsproblemen in Industrie, Forschung und Wirtschaft eingesetzt (Optimierung von Konstruktionen z.B. in der Aerodynamik, Optimierung von Strategien zur Ressourcennutzung, Prognose von Zeitreihen, etc.) [30].

Der Optimierungsprozess mittels genetischem Algorithmus kann in folgende Schritte gegliedert werden [30]:

1. Anfangspopulation wird zufällig erzeugt
2. Berechnung des Zielfunktionswertes für jedes Individuum
3. Selektion der Elternpaare nach bestem Zielfunktionswert
4. Erzeugung von Nachkommen durch Rekombination
5. Eventuelle Mutation der Nachkommen
6. Reproduktion der neuen Population
7. Wiederholung der Schritte 2 bis 6 bis Abbruchkriterium erreicht wird

In [8] wird ein mehrstufiger genetischer Algorithmus zur Optimierung einer Kälteanlage eingesetzt. Als Softwaretoll wird Matlab-Simulink verwendet. Die Besonderheit bei diesem Ansatz liegt darin, dass mehrere genetische Algorithmen stufenweise nacheinander ausgeführt werden. Bei der 1. Stufe wird die Anfangspopulation vollständig zufällig erzeugt. Der Algorithmus führt die Optimierung aus bis die festgelegten Anzahl an Generationen erreicht wird und die besten Individuen stellen einen Teil der Anfangspopulation für die nächste Stufe dar. Der andere Teil der Anfangspopulation bildet sich aus zufällig neu generierten Individuen. So werden die Stufen nacheinander durchlaufen. Die Anzahl an Generationen wird gleichmäßig auf die Stufen aufgeteilt. Anstatt also einen einzigen Durchgang mit der maximalen Anzahl an Generationen durchzuführen, werden mehrere Durchläufe mit einer gleichmäßigen Aufteilung der Generationen auf mehrere Stufen ausgeführt. Durch den mehrstufigen genetischen Algorithmus können das OCL und das OCS Problem simultan gelöst werden. Die Autoren erzielen bessere Leistungen, als bei Anwendung eines Standardalgorithmus.

Ein weiterer Optimierungsansatz mit einem genetischen Algorithmus wird in [9] vorgestellt. Es werden vereinfachte lineare Modelle der wichtigsten Komponenten verwendet. Der Autor hebt die Wichtigkeit der Modellgenauigkeit hervor. Mit einem RLS (Recursive Least Squares) Algorithmus werden die Modellparameter ermittelt und laufen angepasst. Dieser Algorithmus basiert auf der Methode der kleinsten Quadrate. Um auf sich ändernde Betriebsbedingungen reagieren zu können werden laufend Messdaten erfasst und die Modellparameter aktualisiert. Dadurch wird sichergestellt, dass die Vorhersagen des Modells zuverlässig sind. Die Optimierung selbst wird mit dem genetischen Algorithmus durchgeführt. Es können Einsparungen von 0,73%-2,55% des Energiebedarfs, bezogen auf eine Referenzstrategie, erzielt werden.

In [13] wird eine Kombination aus genetischem Algorithmus und Partikel-Schwarm Algorithmus vorgestellt. Der resultierende Algorithmus besteht aus zwei Ebenen. In der 1. Ebene arbeitet der genetische Algorithmus, der die Einschaltreihenfolge der Maschinen über diskrete Variablen steuert. Nachdem der Zielfunktionswert für alle Individuen bestimmt wurde, greift der Partikel-Schwarm Algorithmus ein und steuert zwei kontinuierlichen Variablen: Kälteleistung und Kühlwassereintrittstemperatur auf Kondensatorseite. Der Eingriff erfolgt zwischen Schritt 2 und 3 in obiger Darstellung der schrittweisen Arbeitsweise des genetischen Algorithmus. Der Partikel-Schwarm Algorithmus verbessert durch Variation der kontinuierlichen Variablen den Zielfunktionswert und gibt die verbesserten Individuen an den genetischen Algorithmus zurück. Dann wird der Ablauf des genetischen Algorithmus fortgesetzt. Die erzielbaren Einsparungen durch die Optimierung liegen an zwei untersuchten Tagen bei 14% des Energiebedarfs.

2.3.4 Partikel-Schwarm Algorithmen

Die Partikel-Schwarm Algorithmen stellen eine weitere Untergruppe der evolutionären Algorithmen dar. Sie beruhen auf dem Prinzip der Schwarmintelligenz. Ein Partikel weist ähnliches Verhalten auf wie ein Individuum in einer Population beim genetischen Algorithmus. Jeder Partikel stellt eine mögliche Lösung dar. Nur werden keine neuen durch Rekombination erzeugt, sondern die Partikel tauschen untereinander Informationen aus. Vergleichbar mit einem Vogelschwarm, der aus einer Vielzahl einzelner Vögel (Partikel) besteht. Jeder Vogel sieht in eine spezifische Richtung und durch Kommunikation untereinander wird der Vogel mit der besten Position innerhalb des Schwarms identifiziert. Daraufhin passt jeder Vogel seine eigene Bewegungsrichtung, abhängig von seiner aktuellen Geschwindigkeit, an. Die fortschreitende Anpassung findet über das Verhalten der einzelnen Partikel statt und nicht, wie beim genetischen Algorithmus, durch genetische Reproduktion [29].

In [29] werden 5 evolutionäre Algorithmen miteinander verglichen. Es wird sowohl die Rechenzeit, wie auch die Güte der Lösung bewertet. Dabei werden sowohl kontinuierliche, wie auch diskrete Optimierungsprobleme betrachtet. Der Autor kommt zu dem Ergebnis, dass Partikel-Schwarm Algorithmen sowohl für diskrete, wie auch kontinuierliche Optimierungsprobleme die besten Lösungen ermitteln. Außerdem erreichen sie die zweitkürzesten Rechenzeiten.

Auch zur Optimierung von Kälteanlagen werden Partikel-Schwarm Algorithmen eingesetzt. In [10] wird ein verbesserter Bienenschwarm-Algorithmus (IRBSO = improved ripple bee swarm optimization) verwendet. Bei [27] wird ebenfalls ein Schwarm-Algorithmus verwendet. Hier wird das oben beschriebene OCL Problem erweitert. Die Optimierung soll für ein 24 Stunden Szenario erfolgen und das zu lösende Optimierungsproblem wird dementsprechend als DOCL (daily optimal chiller loading) Problem bezeichnet. Die Komplexität ist nochmals erhöht und als Ergebnis liefert der Optimierungsprozess ein optimales 24 Stunden Kältelastprofil für jede Kältemaschine. Zur Lösung des Optimierungsproblems werden zwei Varianten von Partikel-Schwarm Algorithmen eingesetzt: EPSO (elitism-based particle swarm optimization) und MA-PSO (multi-agent particle swarm optimization). Auch mit diesen Algorithmen können gute Ergebnisse erzielt werden, wobei der EPSO die beste Performance liefert.

2.3.5 Weitere Algorithmen

Ein weiterer Ansatz zur Lösung des OCL Problems wird in [28] vorgestellt. Dort wird das General Algebraic Modelling System (GAMS) verwendet, um die Optimierung durchzuführen. Die Resultate des GAMS werden mit anderen Optimierungsalgorithmen verglichen, die das gleiche Szenario optimiert haben und zeigen bessere Ergebnisse als etwa genetische Algorithmen oder Partikel-Schwarm

Algorithmen. Anzumerken ist, dass sich das System in einem stationären Zustand befindet und somit keine Bedarfsschwankungen der Kälteleistung auftreten was eine Vereinfachung des in [27] beschriebenen DOCL darstellt.

2.3.6 Optimieren mit Randbedingungen

Für ein Optimierungsproblem existiert eine endliche Anzahl möglicher Lösungskandidaten. Die Summe aller Lösungskandidaten bildet den Suchraum Ω [30]. Der Suchraum kann durch Randbedingungen eingeschränkt werden. Dabei wird zwischen weichen und harten Randbedingungen unterschieden. Bei weichen Randbedingungen ist die Einhaltung erwünscht, bei harten Randbedingung ist die Einhaltung eine Grundvoraussetzung. Werden die harten Randbedingungen von einem Lösungskandidaten nicht erfüllt ist er ungültig [30]. Nachfolgend werden die Randbedingungen beschrieben, die beim Finden einer optimalen Einschaltreihenfolge für die in Kapitel 4 simulierte und optimierte Kälteanlage berücksichtigt werden müssen.

Die Einschaltreihenfolge der Maschinen ist der Freiheitsgrad des Optimierers. Diese Reihenfolge wird über einen Prioritätsvektor festgelegt. Der Prioritätsvektor $\vec{p}$ besteht aus den Variablen v_1 bis v_{12} (Gleichung (2.17)).

$$\vec{p} = (v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}, v_{11}, v_{12}) \tag{2.17}$$

Da die abzubildende Kälteanlage aus 12 Maschinen aufgebaut ist handelt es sich um einen 12-dimensionalen Vektor, bestehend aus 12 Variablen. Diese Variablen sind nicht unabhängig voneinander, denn der resultierende Vektor muss bestimmten Regeln entsprechen:

- Jeder Eintrag ist eine natürliche Zahl im Wertebereich von 1 bis 12
- Jeder Eintrag ist einzigartig
- Die Summennorm des Vektors ist immer 78

Diese Regeln stellen die harten Randbedingungen des Optimierungsproblems dar. Prioritätsvektoren, die diesen Randbedingungen nicht entsprechen sind ungültig. Die Gültigkeit eines Prioritätsvektors wird von der Simulation vor jedem Simulationsstart überprüft. Wird versucht die Simulation mit einem ungültigen Prioritätsvektor zu starten, verhindert die Verteilfunktion den Start der Simulation (siehe Kapitel 3.2.3). Im Optimierungsprozess muss dieser Prioritätsvektor selbstständig von einem Optimierungsalgorithmus variiert werden und so automatisiert eine optimale Lösung gefunden werden.

Im Falle des Prioritätsvektors umfasst der Suchraum sämtliche gültigen Vektoren. Da es sich um ein Reihenfolgenproblem handelt, entspricht die Anzahl an gültigen Vektoren der Anzahl an Permutationen ohne Wiederholung des Vektors $\vec{p}$. Die Anzahl an Permutationen ohne Wiederholung kann über die Fakultät berechnet werden. Es gilt Gleichung (2.18), wobei P_n die Anzahl an Permutationen darstellt [32].

$$P_n = n! \tag{2.18}$$

Bei 12 Maschinen besteht der Suchraum Ω aus 12! (479 001 600) Vektoren. Diese Zahl steigt bei weiteren Maschinen sehr stark an. Auf der anderen Seite kann die Anzahl durch weniger Maschinen schnell reduziert werden.

Um die Güte eines Lösungskandidaten bestimmen zu können wird eine Zielfunktion f benötigt. Die Berechnung des Zielfunktionswertes eines Lösungskandidaten nimmt Rechenzeit in Anspruch. Deshalb besteht die wesentliche Aufgabe des Optimierungsalgorithmus darin, den Suchraum durch Bewertung möglichst weniger Lösungskandidaten nach einem optimalen Lösungskandidaten zu durchsuchen.

Grundsätzlich können zwei Methoden unterschieden werden, um mit Randbedingungen bei einer Optimierung mit einem evolutionären Algorithmus umzugehen [30]:

- Restriktive Methode: die Optimierung erfolgt auf einen unbeschränkten Suchraum. Das Vorkommen von ungültigen Lösungen in der Population wird durch zusätzliche Maßnahmen verhindert (z.B. genetisches Reparieren, Krippentod, Methode der gültigen Individuen).
- Tolerante Methode: ungültige Lösungskandidaten werden generiert und in der Population zugelassen. Bei der weiteren Evolution werden sie aber benachteiligt und so aus der Population mit zunehmender Anzahl an Generationen verdrängt.

Die Benachteiligung von ungültigen Lösungen kann bei toleranten Methoden an verschiedenen Stellen erfolgen (Selektion, Mutation, Bewertung), doch kann beim verwendeten Simulationstool AMESIM nicht so weit in die Arbeitsweise des Algorithmus eingegriffen werden, um Selektion und Mutation zu beeinflussen. Die Benachteiligung von ungültigen Lösungskandidaten muss also über Strafterme in der Zielfunktion bei der Bewertung erfolgen. Für die Bewertung durch die Zielfunktion ist ein vollständiger Simulationsdurchlauf notwendig. Werden mehreren Maschinen durch einen ungültigen Vektor die gleichen Prioritätszahlen zugeordnet kann es passieren, dass zu wenige Maschinen für den Belastungsfall in Betrieb sind. Bei einem längeren Simulationszeitraum überschreiten die Netztemperaturen sehr schnell physikalisch sinnvolle Werte. Die Kältemaschinenmodelle liefern in diesem Fall keine gültigen Vorhersagen mehr. Das Simulationsergebnis wird unbrauchbar und die Bewertung

durch die Zielfunktion ist nur mehr eingeschränkt möglich. Der Start der Simulation mit einem ungültigen Vektor wird deshalb von der Verteilfunktion verhindert. Für den Anwendungsfall des Prioritätsvektors sind tolerante Methoden somit ungeeignet. Es wird eine Methode benötigt, die ungültige Prioritätsvektoren gar nicht erst von der Simulation simulieren lässt.

Restriktive Methoden erfüllen diese Anforderung. Sie verfolgen den Ansatz, dass ungültige Lösungen gar nicht erst in der Population vorkommen, indem sie sofort nach der Generierung gelöscht oder modifiziert werden. Das Löschen direkt nach der Generierung wird als Krippentod bezeichnet. Dieses Verfahren liefert gute Ergebnisse, wenn einfache Randbedingungen vorhanden sind und der Lösungsraum nicht zerklüftet ist. Für einen zerklüfteten Lösungsraum (siehe Abbildung 8) ist diese Methode nicht geeignet. So kann es durchaus sinnvoll sein, ungültige Lösungen im Punkt B als Eltern zuzulassen, um durch Rekombination und Mutation auf Lösungen im Lösungsraum C zugreifen zu können. Außerdem kann bei komplexen Randbedingungen das Problem auftrete, dass sehr viele ungültige Kandidaten generiert und anschließend wieder eliminiert werden und der Algorithmus langsam wird.

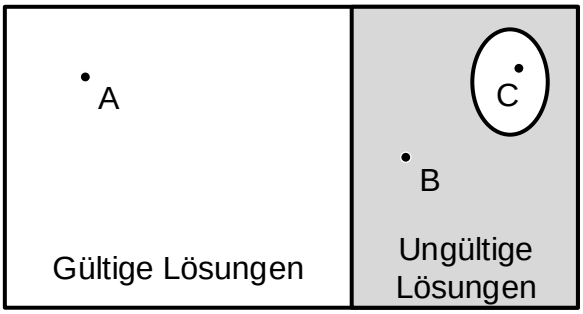

Abbildung 8: Zerklüfteter Lösungsraum in Anlehnung an [30]

Beim genetischen Reparieren wird ein Reparaturalgorithmus angewendet, um ungültigen Lösungskandidaten in gültige Lösungskandidaten umzuwandeln. Dabei werden die Kandidaten solange modifiziert, bis die Randbedingungen erfüllt werden. Im Gegensatz zum Krippentod bringt diese Methode für einen zerklüfteten Lösungsraum keine entscheidenden Vorteile. Auch hier ist Punkt C von Punkt A aus schwer erreichbar, wenn Kandidaten im Punkt B unzulässig sind.

Eine Lösung für diese Problematik stellt die Methode der gültigen Individuen dar. Dabei werden ungültige Lösungskandidaten gar nicht erst erzeugt. Um dies sicherzustellen wird einerseits die zufallsbasierte initiale Population nur mit gültigen Lösungen gebildet und die Rekombinationsoperationen so festgelegt, dass aus gültigen Lösungen wieder nur gültigen Lösungen hervorgehen können.

Die Methode der gültigen Individuen wird bei der Implementierung verwendet. Der Optimierungsalgorithmus erzeugt als initiale Population nur natürliche Zahlen im

Wertebereich von 1 bis 362 880. In den Rekombinationseigenschaften wird festgelegt, dass nachfolgende Generationen wiederrum nur aus natürlichen Zahlen im Wertebereich bestehen dürfen. Der Wertebereich ergibt sich aus der Anzahl an Permutationen ohne Wiederholung für 9 Maschinen. Über eine Tabelle werden diese Zahlen einem gültigen Prioritätsvektor zugeordnet. Der vom Optimierungsalgorithmus vorgegebene Zahlenwert entspricht einer Zeilennummer in der Tabelle. Diese Tabelle wird in weiterer Folge als Prioritätsliste bezeichnet.

Die Reduktion auf 9 Maschinen ist aufgrund der Implementierung notwendig, weil eine Prioritätsliste mit 362 880 Einträgen in der Simulationssoftware noch handhabbar ist. Um die Optimierung trotzdem mit 12 Maschinen durchführen zu können ist ein vorrangehender Schritt notwendig, der 3 Maschinen identifiziert, die unabhängig vom Optimierungsalgorithmus als Bestandteil des optimierten Vektors vorgegeben werden. Dies geschieht durch Simulation einer Permutationsliste. Nähere Ausführungen zum Optimierungsprozess in der Simulationsumgebung finden sich in Kapitel 3.4.

3 Implementierung

In diesem Kapitel wird die Implementierung der formalen Modelle in die Simulationssoftware AMESIM beschrieben. AMESIM (Advanced Modelling Environment for performing Simulations of engineering systems) ist eine grafische Simulationsplattform, die Werkzeuge zur Modellierung, Simulation und Analyse von multidisziplinären Systemen bereitstellt [33]. Die programmspezifischen Beschreibungen in diesem Kapitel sind der AMESIM Programmdokumentation entnommen [34].

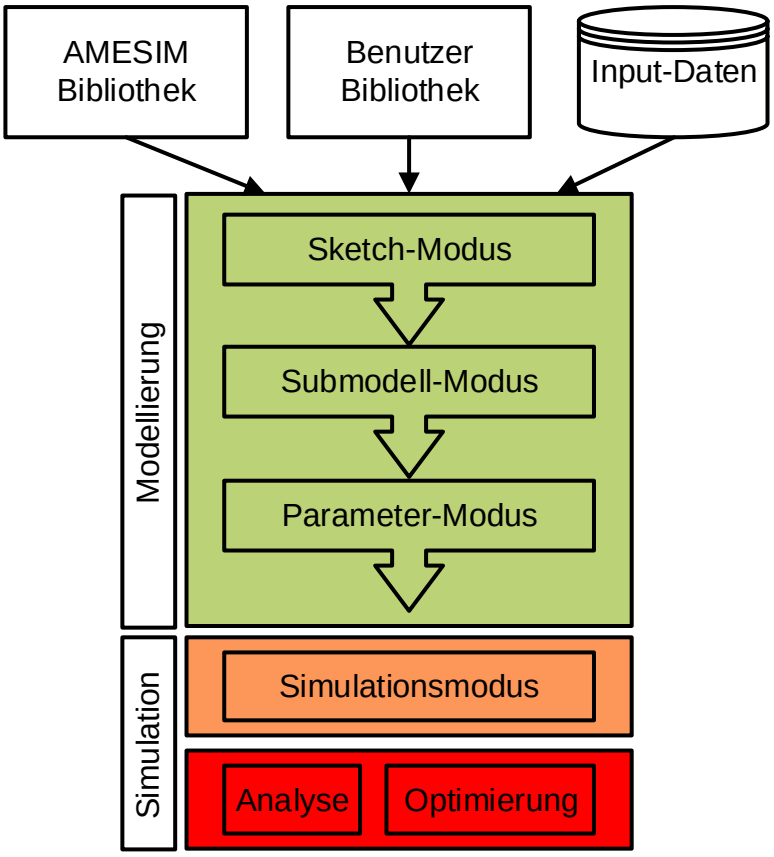

Abbildung 9: Aufbau der AMESIM Simulationssoftware in Anlehnung an [34]

Der grundlegende Aufbau der Simulationssoftware wird in Abbildung 9 dargestellt. Er gliedert sich in zwei Bestandteile: Modellierung und Simulation. Im Sketch-Modus ist die gesamte Simulation grafisch dargestellt und wird aus einzelnen Komponenten aufgebaut. Diese Komponenten stammen entweder aus den in AMESIM integrierten Bibliotheken oder aus Benutzerbibliotheken. Im Submodell-Modus werden den Komponenten die Submodelle zugewiesen. In den Submodellen sind die mathematischen Berechnungen der Komponente hinterlegt. Im Parameter-Modus werden die Parameter für alle Komponenten festgelegt. Das Simulationsmodell kann im Simulationsmodus ausgeführt werden. Außerdem stehen im Simulationsmodus Werkzeuge zur Analyse und Optimierung zur Verfügung.

Wenn eine Simulation mit AMESIM ausgeführt wird, überführt AMESIM das auf dem Sketch dargestellte System in computerlesbaren Code. Dieser Code wird ausgeführt. Im Kern von AMESIM arbeitet ein Integrier-Algorithmus. Dieser Algorithmus ruft die einzelnen Submodelle auf, welche an die Komponenten der Simulation gebunden sind und berechnet die Lösung der Gleichungen, abhängig von der Zeit.

AMESIM stellt Werkzeuge zur Analyse der Simulationsergebnisse zur Verfügung. Es kann jede Variable und ihr Zeitverlauf ausgegeben werden. Zusätzlich besteht die Möglichkeit die Variablen in Form von Animationen zu visualisieren. Außerdem stellt AMESIM einige Möglichkeiten zur Optimierung des Simulationsmodells bereit. Auf den genetischen Algorithmus, der von AMESIM als Optimierungsoption angeboten wird, wird in Kapitel 3.4 detailliert eingegangen.

Obwohl AMESIM-Bibliotheken für Kühlsysteme vorhanden sind, gibt es keine Komponenten, die ein datenbasiertes Kältemaschinenmodell abbilden können. Auch für das Energienetz sind keine vorgefertigten Komponenten vorhanden. Deswegen müssen alle formalen Modelle über die Benutzerbibliotheken in die Software implementiert werden. Das bedeutet, dass diese Modelle mit dem Werkzeug „Ameset“ vollständig benutzerdefiniert erstellt werden. Für die Übertragung von Informations- und Energieflüssen zwischen den Komponenten kann auf die Standardkomponenten der AMESIM-Bibliothek „Signal, Control“ zurückgegriffen werden.

3.1 Grundlagen

In diesem Kapitel soll der grundlegende Aufbau von Komponenten und Submodellen erläutert werden. Diese Grundlagen sind notwendig, um den Aufbau der benutzerdefinierten Modelle nachvollziehen zu können.

3.1.1 Variablentypen

Jede Größe, die während einer Simulation auftritt, wird durch eine Variable im Programmcode ausgedrückt. Im folgenden Abschnitt werden die Variablen, die bei der Implementierung verwendet werden erläutert. Sie können in folgende Kategorien eingeteilt werden:

- Externe Variablen
    - Inputvariablen
    - Outputvariablen
- Interne Variablen
- Parameter
    - Kontinuierliche Parameter
    - Diskrete Parameter

Variablen sind zeitabhängige Größen, die sich im Laufe der Simulation verändern. Sie können in externe und interne Variablen unterschieden werden. Externe Variablen stehen über die Modellports anderen Submodellen zur Verfügung und werden im Laufe der Simulation zwischen den Submodellen ausgetauscht. Sie werden entweder von anderen Submodell bereitgestellt (Inputvariablen), oder werden an andere Submodell weitergeleitet (Outputvariablen). Interne Variablen stehen ausschließlich dem entsprechenden Submodell zur Verfügung und dienen zum Ausführen der mathematischen Berechnungen, die im Submodell hinterlegt sind. Neben der Variable selbst kann auch ihre zeitliche Ableitung definiert werden. Die Ableitung kann sowohl explizit, als auch implizit implementiert werden. Diese Gleichungen werden nicht vom Submodell, sondern vom Integrator gelöst.

Parameter sind Größen, die während des gesamten Simulationsdurchlaufs konstant bleiben. Sie werden zur Berechnung der externen Variablen (kontinuierliche Parameter) oder zur Festlegung von modellspezifischen Einstellungen (diskrete Parameter) verwendet. Kontinuierliche Parameter sind, genauso wie externe und interne Variablen, reelle Zahlen, die im Programm mithilfe von Fließkommazahlen approximiert werden. Sie haben den Datentyp „double“. Im Gegensatz dazu legen diskrete Parameter Einstellungen zum Modellverhalten fest (z.B. Wärmerückgewinnung vorhanden/nicht vorhanden beim Kältemaschinenmodell). Sie können ausschließlich diskrete Zahlenwerte annehmen. Dafür reicht ein auf ganze Zahlen begrenzter Wertebereich aus. Sie haben deswegen den Datentyp „integer“.

3.1.2 Modellports

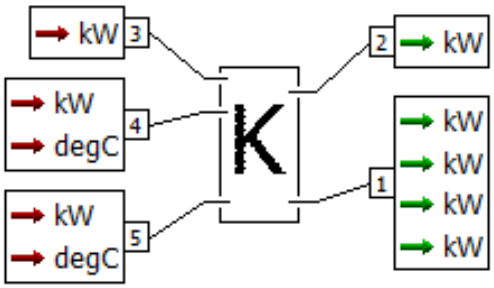

Abbildung 10: Ports und externe Variablen des Kältemaschinenmodells

Um den Austausch von Variablen zwischen den Komponenten zu ermöglichen, verfügt jede Komponente über einen oder mehrere Ports. Als Beispiel werden in Abbildung 10 die Ports des Kältemaschinenmodells dargestellt. Die Ports sind von 1 bis 5 durchnummeriert. Gleichzeitig werden die externen Variablen dargestellt, die an den Ports anliegen. Die Pfeilrichtung zeigt an, ob es sich um eine Inputvariable oder um eine Outputvariable handelt. Inputvariablen zeigen zum Icon hin, Outputvariablen vom Icon weg.

3.1.3 Aufbau des AMESIM-Templates

Nach der Definition aller Variablen und Ports eines Submodells, generiert AMESIM automatisch eine Vorlage (Template) in der Programmiersprache C. Ein solches Template wird in Abbildung 11 dargestellt.

```c
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include "ameutils.h"
#define _SUBMODELNAME_ "TEMP_001"

void temp_001in_(int *n, double rp[1], int ip[1])
{
   int loop, error;
   int a;
   double z;
   a           = ip[0];
   z           = rp[0];
   loop = 0;
   error = 0;

   if (a < 0 || a > 1)
   {
      amefprintf(stderr, "\nInteger Parameter a must be in range [0..1].\n");
      error = 2;
   }
   if(error == 1)
   {
      amefprintf(stderr, "\nWarning in %s instance %d.\n", _SUBMODELNAME_, *n);
   }
   else if(error == 2)
   {
      amefprintf(stderr, "\nFatal error in %s instance %d.\n", _SUBMODELNAME_,
    *n);
      amefprintf(stderr, "Terminating the program.\n");
      AmeExit(1);
   }
}

void temp_001_(int *n, double *in, double *out, double *x, double *y
      , double rp[1], int ip[1])
{
   int loop;
   int a;
   double z;
   a           = ip[0];
   z           = rp[0];
   loop = 0;
/*
   Set all submodel outputs below:
   *out         = ??;
   *x           = ??;
   *y           = ??;
*/
/* Calculation Function Executable Statements. */

/* End of Calculation Executable Statements. */
}
```

Abbildung 11: Von AMESIM generiertes Template

Das Template ist aus zwei Funktionen aufgebaut: Initialisierung (Zeile 7 bis 32) und Kalkulation (Zeile 34 bis 52). Die Initialisierung wird einmal am Beginn der Simulation ausgeführt. Die Kalkulation wird während der Simulation laufend ausgeführt.

In der Initialisierung können neben den von AMESIM generierten Variablendeklarationen weitere Funktionen oder Subroutinen deklariert werden. Außerdem können vom Benutzer eingegebene Parameter überprüft werden. Das ist eine wichtige Funktion, um das Ausführen der Simulation mit ungültigen Parametern zu verhindern. In den Zeilen 17 bis 21 wird bspw. überprüft, ob der diskrete Parameter a im richtigen Wertebereich liegt. Dieser Parameter wird zur Steuerung von Funktionen im Kalkulationsteil verwendet und kann nur die Werte 0 oder 1 annehmen. Zusätzlich könnte in der Initialisierung noch der Wertebereich des kontinuierlichen Parameters z überprüft werden.

Sollten Parameter außerhalb des gültigen Wertebereichs liegen wird die Variable error gesetzt (Zeile 20). Danach erfolgt die Ausgabe einer Fehlermeldung am Bildschirm (Zeile 19). Es gibt die Möglichkeit die Variable error auf die Werte 1 oder 2 zu setzen. Wird error = 1 gesetzt, erfolgt die Ausgabe des Fehlers lediglich als Warnung. Der Benutzer wird über die fehlerhafte Eingabe informiert, die Simulation aber trotzdem ausgeführt. Wird error = 2 gesetzt, bricht die Simulation sofort ab. Besonders wichtig ist diese Fehlerbehandlung bei der Verteilfunktion. Wird dort versucht die Simulation mit einem falschen Prioritätsvektor zu starten, wird error = 2 gesetzt und die Simulation bricht sofort ab. Die Fehlermeldung wird in Abbildung 12 dargestellt.

No		Message	Submodel
1	ℹ	Instantiating a system with 80 unknowns.	
2	ℹ	Initializing model for simulation...	
3	ℹ	Read 362880 rows and 10 columns from file ./Use_Case_1_1_v031_data/vector_list_9_9_1.data.	
4	ℹ	Read 6600 rows and 13 columns from file ./Use_Case_1_1_v031_data/vector_list_12_3_5.data.	
5	ℹ	Number of x-y pairs read from file ./Use_Case_1_1_v031_data/input_qkb_110515.data : 1440	
6	ℹ	Requesting amesim_runtime license ...	
7	ℹ	got the license for amesim_runtime, continuing	
8	⚠	ACHTUNG: Falsche Prioritätszuordnung (falsche Gesamtsumme)!	
9	⚠	ACHTUNG: 2 oder mehr Maschinen mit gleicher Priorität!	
10	⚠	Fatal error in VERTEILFKT_12M_019 instance 1.	VERTEILFKT_12M_019 instance 1
11	⚠	Terminating the program.	
12	ℹ	Total CPU time: 0 seconds.	
13	ℹ	Total CPU time: 0 seconds.	
14	⚠	Initialization failed.	
15	✖	LMS Amesim model Use_Case_1_1_v031 initialization failed (1).	

Abbildung 12: Fehlermeldung aufgrund eines falschen Prioritätsvektors

Die Überprüfung des Fehlers geschieht aber bei der Verteilfunktion nicht im Initialisierungsteil sondern erst im Kalkulationsteil, da die Prioritätszahlen keine Parameter der Zielfunktion sind, sondern Outputvariablen des Logikbausteins und von

diesem an die Verteilfunktion weitergeleitet werden. Sie sind somit Inputvariablen der Verteilfunktion. Inputvariablen können erst im Kalkulationsteil überprüft werden.

Im Kalkulationsteil werden die Berechnungen für die Outputs out, x und y hinterlegt (Zeile 50, Abbildung 11). AMESIM ruft die Submodelle unabhängig von der verwendeten Programmiersprache auf, d.h. dass die Outputs im Kalkulationsteil als Zeiger ausgeführt sind und deshalb mit einem „*“ gekennzeichnet werden (siehe Zeile 34, Abbildung 11). Zeiger enthalten die Adresse zu einer Variable, nicht aber den Wert der Variable selbst.

Im Anhang (Kapitel 6.2) sind die Kalkulationsfunktionen der im nachfolgenden Kapitel beschriebenen Implementierungsmodelle von Kältemaschine, Energienetz und Verteilfunktion aufgeführt.

3.2 Ausführbare Modelle

In diesem Kapitel werden aufbauend auf den formalen Modellen des Grundlagenteils (siehe Kapitel 2.2.2) die ausführbaren Modelle beschrieben. Diese werden im Vergleich zu den formalen Modellen um implementierungsspezifische Variablen, Funktionen und Parameter erweitert. Außerdem werden Inputs und Outputs an den Ports festgelegt. Die mathematischen Berechnungen der formalen Modelle und die implementierungsspezifischen Zusätze werden in Programmcode umgesetzt. Als Programmiersprache wird C verwendet. Nachfolgend wird auf jedes Modell eingegangen und die internen Berechnungen, die Parameter, sowie die an den Ports anliegenden Inputs und Outputs beschrieben. Außerdem wird die Funktionsweise jedes Modells erläutert.

3.2.1 Kältemaschinenmodell

Die Energie- und Informationsports des Kältemaschinenmodells werden in Abbildung 13 dargestellt.

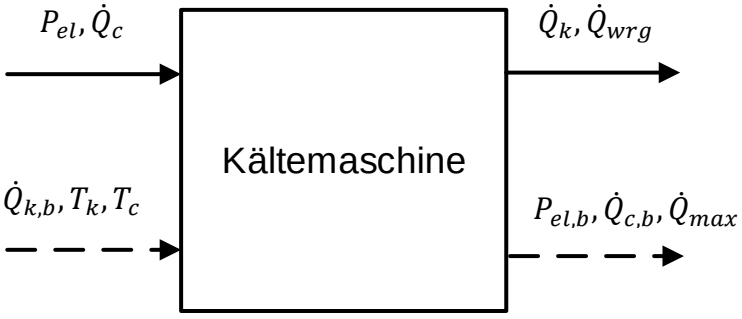

Abbildung 13: Energie- und Informationsports Kältemaschine

Das Kältemaschinenmodell bezieht elektrische Leistung und Kühlleistung und gibt Kälteleistung ab. Ist ein Wärmerückgewinnungssystem verbaut, kann die Maschine zusätzlich rückgewonnene Wärme abgeben. Die Kälteleistung wird mit negativem Vorzeichen angefordert und auch mit negativem Vorzeichen abgegeben. Die Referenzkälteleistung wird jedoch mit positivem Vorzeichen angegeben. Das Modell benötigt Informationen zum Kälteleistungsbedarf, sowie zu den Temperaturen in Kälte- und Kühlkreislauf. Außerdem gibt das Modell die Informationen zur benötigten elektrischen Leistung und der benötigten Kühlleistung aus. Alle externen Variablen, die an den Ports des Modells anliegen, werden in Tabelle 4 aufgeführt.

Variable	Bezeichnung	Einheit	Datentyp	Input/Output
P_{el}	Elektrische Leistung	kW	Double	Input
$\dot{Q}_c$	Kühlleistung	kW	Double	Input
$\dot{Q}_k$	Kälteleistung	kW	Double	Output
$\dot{Q}_{wrg}$	Wärmeleistung	kW	Double	Output
$\dot{Q}_{k,b}$	Bedarf Kälteleistung	kW	Double	Input
T_k	Kaltwassertemperatur	°C	Double	Input
T_c	Kühlwassertemperatur	°C	Double	Input
$P_{el,b}$	Bedarf elektr. Leistung	kW	Double	Output
$\dot{Q}_{c,b}$	Bedarf Kühlleistung	kW	Double	Output
$\dot{Q}_{max}$	Maximale Kälteleistung	kW	Double	Output

Tabelle 4: Externe Variablen des Kältemaschinenmodells

In Tabelle 5 werden die Parameter des Kältemaschinenmodells aufgeführt. Auf die Parametrisierung wird in Kapitel 2.2.3 eingegangen.

Variable	Bezeichnung	Einheit	Datentyp
$\dot{Q}_{ref}$	Referenzkälteleistung	kW	Double
η_0	Referenzwirkungsgrad Kühlleistung	-	Double
η_1	Referenzwirkungsgrad Wärmerückgewinnung	-	Double
COP_{ref}	Referenzleistungszahl	-	Double
a_i	Regressionskoeffizienten Teillastkennlinie	-	Double
b_i	Regressionskoeffizienten Wirkungsgradkennlinie	-	Double
c_i	Regressionskoeffizienten Kapazitätskennlinie	-	Double
WRG	Wärmerückgewinnung ein/aus	-	Integer

Tabelle 5: Parameter Kältemaschinenmodell

Tabelle 6 listet die internen Variablen des Kältemaschinenmodells auf.

Variable	Bezeichnung	Einheit	Datentyp
$\dot{Q}_{kb,temp}$	Temporäre Kälteleistung	kW	Double
PLR	Auslastung	-	Double
$EIRFT$	Wirkungsgradkennlinie	-	Double
$EIRFPLR$	Teillastkennlinie	-	Double
EER	Wirkungsgrad	-	Double

Tabelle 6: Interne Variablen Kältemaschinenmodell

Nachfolgend werden die Berechnungen des Modells anhand des Rechenweges erläutert. Es müssen nur algebraische Gleichungen gelöst werden. Es sind keine Differenzialgleichungen notwendig. Die Berechnung der elektrischen Leistungsaufnahme in Abhängigkeit von der angeforderten Kälteleistung wird analog zu [25] durchgeführt (siehe Kapitel 2.2.2), wobei im Unterschied zu [25] eine vereinfachte Funktion für die Teillastkennlinie $EIRFPLR$ verwendet wird. Die vereinfachte Funktion für die Teillastkennlinie ist [23] entnommen.

$$\dot{Q}_{max} = \dot{Q}_{ref}(c_0 + c_1 T_k + c_2 T_c + c_3 T_k^2 + c_4 T_k T_c + c_5 T_c^2) \tag{3.1}$$

$$EIRFT = b_0 + b_1 T_k + b_2 T_c + b_3 T_k^2 + b_4 T_k T_c + b_5 T_c^2 \tag{3.2}$$

Die Gleichungen (3.1) und (3.2) beschreiben die Berechnung der maximalen Kälteleistung $\dot{Q}_{max}$ und der Wirkungsgradkennlinie $EIRFT$ über die quadratischen Funktionen. Durch Anwenden von Gleichung (3.3) wird die Kälteleistung der Maschine mit $\dot{Q}_{max}$ gedeckelt, d.h. die Kältemaschine kann nicht mehr Kälteleistung abgeben, als durch die Kapazitätskennlinie festgelegt.

$$\dot{Q}_{kb,temp} = -\min(\dot{Q}_{k,b}, \dot{Q}_{max}) \tag{3.3}$$

Anschließend erfolgt die Berechnung der Maschinenauslastung PLR nach Gleichung (3.4). In Gleichung (3.5) wird mit PLR die Teillastkennlinie $EIRFPLR$ bestimmt.

$$PLR = \frac{|\dot{Q}_{kb,temp}|}{\dot{Q}_{max}} \tag{3.4}$$

$$EIRFPLR(\dot{Q}_{kb,temp}) = a_0 + a_1 PLR + a_2 PLR^2 \tag{3.5}$$

Mithilfe von Gleichung (3.6) und Gleichung (3.7) kann schließlich der elektrische Leistungsbedarf der Maschine in Abhängigkeit von der angeforderten Kälteleistung bestimmt werden.

$$EER(\dot{Q}_{kb,temp}) = EIRFT * EIRFPLR \tag{3.6}$$

$$P_{el,b}(\dot{Q}_{kb,temp}) = \frac{\dot{Q}_{max} EER(\dot{Q}_{kb,temp})}{COP_{ref}} \tag{3.7}$$

Wird der elektrische Leistungsbedarf durch das Stromnetz gedeckt gibt die Maschine die angeforderte Kälteleistung ab (Gleichung (3.8)).

$$\dot{Q}_k = \begin{cases} \dot{Q}_{k,btemp}, & P_{el} \geq P_{el,b} \\ 0, & P_{el} < P_{el,b} \end{cases} \tag{3.8}$$

Die Kühlleistung, die die Kältemaschine dabei benötigt, wird nach Gleichung (3.9) berechnet.

$$\dot{Q}_{c,b} = \eta_0(\dot{Q}_k + P_{el,b}) \tag{3.9}$$

Sollte die Maschine über ein Wärmrückgewinnungssystem verfügen wird der Parameter WRG auf 1 gesetzt, und der rückgewonnene Wärmestrom nach Gleichung (3.10) berechnet.

$$\dot{Q}_{wrg} = \begin{cases} \eta_1 \dot{Q}_k, & WRG = 1 \\ 0, & WRG = 0 \end{cases} \tag{3.10}$$

Sollte am Eingang der Kältemaschine keine elektrische Leistung anliegen ($P_{el} = 0$), ist die Maschine außer Betrieb. In diesem Fall setzt das Modell alle Outputvariablen gleich 0 ($\dot{Q}_k, P_{el,b}, \dot{Q}_{c,b}, \dot{Q}_{max}, \dot{Q}_{wrg}$). Die Verteilfunktion kann darauf reagieren und die Maschine bei der Verteilung der Kältelast überspringen.

3.2.2 Energienetz-Modell

Die Ports des Energienetz-Modells werden in Abbildung 14 dargestellt.

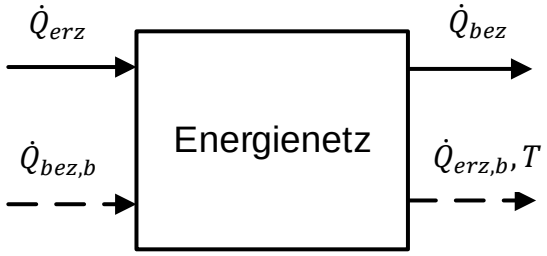

Abbildung 14: Energie- und Informationsports Energienetz

Das Energienetz-Modell wird verwendet, um die Kalt- und Kühlwasserkreisläufe in der Simulation abzubilden. Es kann die von den Erzeugern zur Verfügung gestellte Leistung aufnehmen, die Energie speichern und die Leistung an die angeschlossenen Bezieher abgeben. Gleichzeitig fordert das Netz, abhängig vom Speicherstand, Leistung von den Erzeugern an und hält damit die Temperatur des Kalt- bzw. Kühlwassers konstant.

Alle externen Variablen, die an den Ports des Modells anliegen, werden in Tabelle 7 aufgeführt.

Variable	Bezeichnung	Einheit	Datentyp	Input/Output
$\dot{Q}_{erz}$	Erzeugerleistung	kW	Double	Input
$\dot{Q}_{bez}$	Bezieherleistung	kW	Double	Output
$\dot{Q}_{bez,b}$	Bedarf Bezieherleistung	kW	Double	Input
$\dot{Q}_{erz,b}$	Bedarf Erzeugerleistung	kW	Double	Output
T	Netztemperatur	°C	Double	Output

Tabelle 7: Externe Variablen des Energienetzmodells

In Tabelle 8 werden die Parameter des Energienetzmodells aufgeführt.

Variable	Bezeichnung	Einheit	Datentyp
c	Spezifische Wärmekapazität	kJ/(kgK)	Double
m	Masse	kg	Double
T_{Ziel}	Zieltemperatur	°C	Double
T_0	Referenztemperatur	°C	Double
K_p	Verstärkung P-Anteil	-	Double
K_i	Verstärkung I-Anteil	-	Double

Tabelle 8: Parameter Energienetzmodell

Die vom Modell verwendeten internen Variablen werden in Tabelle 9 aufgelistet.

Variable	Bezeichnung	Einheit	Datentyp
E	Speicherstand	kJ	Double
$regeldiff$	Regeldifferenz	kJ	Double
C	Speicherkapazität	kJ/K	Double
E_{soll}	Soll-Speicherstand	kJ	Double

Tabelle 9: Interne Variablen Energienetzmodell

Die Kapazität C des Speichers berechnet sich nach Gleichung (3.11).

$$C = mc \tag{3.11}$$

Für die Temperatur des Fluids, abhängig vom Speicherstand, gilt Gleichung (3.12).

$$T = \frac{E}{C} + T_0 \tag{3.12}$$

Es handelt sich um einen seriellen Speicher, d.h. sämtliche Energie wird direkt aus dem Speicher bezogen. Der Speicherstand ändert sich nach Gleichung (3.13).

$$\frac{dE}{dt} = \dot{Q}_{erz} - \dot{Q}_{bez} \tag{3.13}$$

$$E(0) = E_{soll} \tag{3.14}$$

Der Speicherstand E wird zu Beginn der Simulation ($t = 0$) gleich der Führungsgröße E_{soll} gesetzt (Gleichung (3.14)). E_{soll} wird beim Start der Simulation aus der Zieltemperatur und der Referenztemperatur nach Gleichung (3.15) berechnet. Es ist sinnvoll die Referenztemperatur gleich der Umgebungstemperatur zu setzen. Wenn dann die Netztemperatur Umgebungstemperatur erreicht wird $E = 0$ und der Speicher ist leer.

$$E_{soll} = C(T_{Ziel} - T_0) \tag{3.15}$$

Damit sich der Speicher nicht leert muss Leistung von den Erzeugern $\dot{Q}_{erz,b}$ angefordert werden. Der Speicherstand E wird so konstant gehalten. $\dot{Q}_{erz,b}$ stellt somit die Stellgröße des Regelkreises dar. Die Regeldifferenz $regeldiff$ errechnet sich nach Gleichung (3.16).

$$regeldiff = E_{soll} - E \tag{3.16}$$

Zu Beginn der Simulation ist durch Gleichung (3.14) die Regeldifferenz gleich 0. Die Zieltemperatur T_{Ziel} ist die gemessene Temperatur am Verdampfer-Eingang bzw. Kondensator-Ausgang. Das Energienetz hält dann mithilfe des integrierten PI-Reglers den Speicherstand und somit die Temperatur konstant.

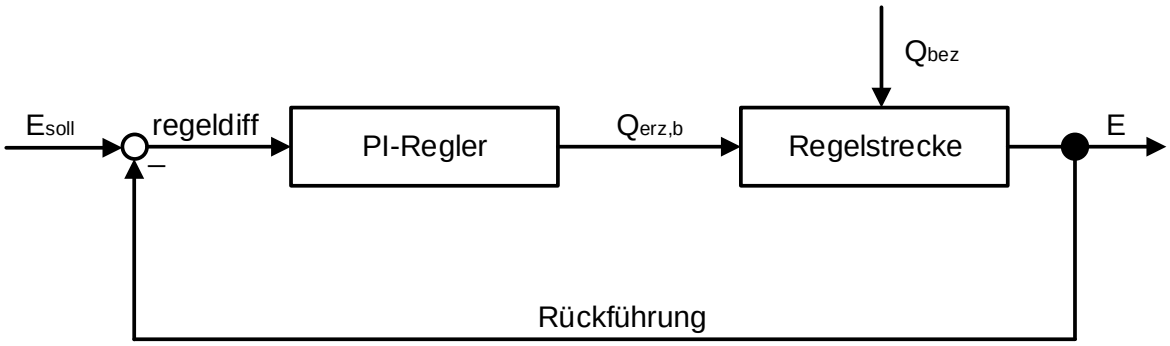

Abbildung 15: Regelkreis des Energienetzes in Anlehnung an [35]

In Abbildung 15 ist der Regelkreis ersichtlich. In Tabelle 10 werden die in Abbildung 15 dargestellten Variablen des Energienetz-Modells den Größen des Standardregelkreises zugeordnet.

Größe des Standardregelkreises	**Variable im Modell**
Führungsgröße w(t)	E_{soll}
Regelabweichung e(t)	$regeldiff$
Stellgröße u(t)	$\dot{Q}_{erz,b}$
Störgröße d(t)	$\dot{Q}_{bez}$
Regelgröße y(t)	E

Tabelle 10: Zuordnung der Größen des Standardregelkreises

Gleichung (3.17) ist die Integralgleichung des PI-Reglers [35].

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau \tag{3.17}$$

K_p entspricht der Verstärkung des P-Anteils des Reglers, K_i der Verstärkung des I-Anteils und e ist die Regelabweichung. Der Vorteil des PI-Reglers gegenüber eines reinen P-Reglers liegt darin, dass keine bleibende Regelabweichung auftritt. Für die Bestimmung der Reglerparameter K_p und K_i können unterschiedliche Verfahren angewendet werden. Beispielhaft zu nennen sind hier das Ziegler-Nichols Verfahren oder das Chien, Hrones, Reswick (CHR) Verfahren. Beide Verfahren setzen ein Aufzeichnen der Sprungantwort der Regelstrecke voraus [35]. Werden alle Größen des Regelkreises in die Integralgleichung des PI-Reglers eingesetzt, ergibt sich Gleichung (3.18).

$$\dot{Q}_{erz,b} = K_p * (regeldiff) + K_i \int_0^t (regeldiff) d\tau \tag{3.18}$$

3.2.3 Verteilfunktion

Die Verteilfunktion verteilt den Gesamtkältebedarf, der vom Energienetz angefordert wird, auf die einzelnen Kältemaschinen. In ihr ist die Schaltlogik implementiert, nach der die Aufteilung erfolgt. Sie weist ausschließlich Informationsports auf (siehe Abbildung 16). Die Verteilfunktion hat außerdem die Aufgabe, die Betriebszeiten der einzelnen Maschinen innerhalb eines Simulationsszenarios aufzuzeichnen und an den Zielfunktionsbaustein weiterzugeben. Im Zielfunktionsbaustein werden diese Informationen für die Berechnung des Zielfunktionswertes benötigt. Damit die Verteilfunktion die Aufteilung des Gesamtkältebedarfs durchführen kann, benötigt sie Informationen über die maximale Kälteleistung jeder einzelnen Maschine. Außerdem wird der Prioritätsvektor an die Verteilfunktion übermittelt und so die Einschaltreihenfolge der Kältemaschinen festgelegt. Außerdem verfügt das Modell über eine Fehlerbehandlung und überprüft die Gültigkeit des Prioritätsvektors. Ist dieser ungültig wird der Start der Simulation verhindert.

Abbildung 16: Informationsports Verteilfunktion

In Tabelle 11 werden die externen Variablen der Ports aufgelistet.

Variable	Bezeichnung	Einheit	Datentyp	Input/Output
$\dot{Q}_{k,b}$	Kälteleistungsbedarf	kW	Double	Input
$\dot{Q}_{max,i}$	Maximale Kälteleistung Maschine i	kW	Double	Input
$\dot{Q}_{kb,i}$	Kälteleistungsbedarf Maschine i	kW	Double	Output
B_i	Betriebszeit Maschine i	h	Double	Output
p_i	Prioritätszahl Maschine i	-	Integer	Input

Tabelle 11: Externe Variablen der Verteilfunktion

Die Parameter der Verteilfunktion werden in Tabelle 12 aufgeführt.

Variable	Bezeichnung	Einheit	Datentyp
$Hyst$	Hysterese	kW	Double

Tabelle 12: Parameter der Verteilfunktion

Die internen Variablen der Verteilfunktion werden in Tabelle 13 aufgelistet.

Variable	Bezeichnung	Einheit	Datentyp
m_{on}	Anzahl Maschinen eingeschaltet	-	Integer
$\dot{Q}_{k,ges}$	Gesamtkälteleistung der Maschinen	kW	Double

Tabelle 13: Interne Variablen der Verteilfunktion

Bei der Schaltung der Maschinen wird eine sequenzielle Strategie verfolgt, d.h. die Maschinen werden nacheinander eingeschaltet. Wenn der Kältebedarf durch die laufenden Maschinen nicht mehr gedeckt werden kann, wird die nächste Maschine eingeschaltet. Dadurch wird immer nur die zuletzt eingeschaltete Maschine unter Teillastbedingungen betrieben, während alle anderen Maschinen unter Volllast laufen.

Die Festlegung der Einschaltreihenfolge erfolgt anhand eines Prioritätsvektors. Dieser Vektor ist in einem Tabellenbaustein gespeichert und wird mit einem Logikbaustein in die Prioritätszahlen p_{01} bis p_{12} konvertiert. Jede Prioritätszahl ist einer Maschine fest zugeordnet (p_{01} legt die Priorität von Maschine 1 fest). Wird nun Kälte angefordert, werden die Maschinen der Reihe nach eingeschaltet, bis der Kältebedarf gedeckt werden kann. In Gleichung (3.19) wird ein Beispielprioritätsvektor $\vec{p}$ dargestellt. Der Vektor $\vec{p}$ ist wie folgt zu lesen: Maschine 1 startet zuerst, dann Maschine 4, dann Maschine 2, usw.

$$\vec{p} = (1{,}4{,}2{,}6{,}5{,}9{,}3{,}7{,}12{,}11{,}10{,}8) \tag{3.19}$$

Eine nähere Beschreibung zur Darstellung eines Vektors in AMESIM und der Konvertierung findet sich in Kapitel 3.4.1. Das Ergebnis der Konvertierung durch den Logikbaustein wird in Tabelle 14 dargestellt.

Prioritätszahl	Wert
p_{01}	1
p_{02}	3
p_{03}	7
p_{04}	2
p_{05}	5
p_{06}	4
p_{07}	8
p_{08}	12
p_{09}	6
p_{10}	11
p_{11}	10
p_{12}	9

Tabelle 14: Werte der Prioritätszahlen aufgrund des Beispielvektors $\vec{p}$

Die Verteilfunktion steuert die Anzahl der aktiven Maschinen durch Erhöhen oder Reduzieren von m_{on}. Diese Variable gibt die Anzahl der derzeit laufenden Maschinen an. Es werden zwei Werte permanent miteinander verglichen: die gesamte derzeit von den Kältemaschinen geleistete Kälteleistung $\dot{Q}_{k,ges}$ und die vom Energienetz angeforderte Kälteleistung $\dot{Q}_{k,b}$. Die gesamte von den Maschinen geleistete Kälteleistung $\dot{Q}_{k,ges}$ ist die Summe der Einzelleistungen der Kältemaschinen (Gleichung (3.20)).

$$\dot{Q}_{k,ges} = \sum_{i=1}^{12} \dot{Q}_{kb,i} \tag{3.20}$$

Weichen diese beiden Werte um einen höheren Betrag als $Hyst$ voneinander ab, wird eine Maschine zu- oder weggeschaltet. Gleichung (3.21) beschreibt das Verhalten von m_{on}. Um zu verhindern, dass m_{on} bei stark schwankender Last zu schnell zwischen den Werten hin- und herspringt, wird das Einschalten bzw. Ausschalten einer Maschine durch eine Hysterese $Hyst$ verzögert.

$$m_{on} = \begin{cases} m_{on} + 1, & \dot{Q}_{k,ges} - \dot{Q}_{k,b} > Hyst \\ m_{on} - 1, & \dot{Q}_{k,ges} - \dot{Q}_{k,b} < -Hyst \end{cases} \tag{3.21}$$

Jeder Maschine wird über den Prioritätsvektor $\vec{p}$ eine Prioritätszahl p_i zugeordnet (siehe Tabelle 14). Diese Prioritätszahl wird mit m_{on} verglichen, um festzustellen, ob an die Maschine eine Kältelast $\dot{Q}_{kb,i}$ weitergeleitet werden muss. Sobald eine Maschine mit Kältelast belastet wird gilt sie als eingeschaltet. Jede Masche kann bis zu ihrer derzeitigen maximalen Kälteleistung $\dot{Q}_{max,i}$ belastet werden. $\dot{Q}_{max,i}$ wird von

jeder Maschine an die Verteilfunktion weitergeleitet (siehe Gleichung (3.1) in Kapitel 3.2.1). Sollte ein Maschine nicht in Betrieb sein (z.B. aufgrund von Wartungspausen) sendet die Maschine $\dot{Q}_{max,i} = 0$ an die Verteilfunktion. Diese Maschine wird dann von der Verteilfunktion übersprungen. So kann die Verteilfunktion auf nicht in Betrieb befindliche Maschinen reagieren. Bei in Betrieb befindlichen Maschinen können die in Gleichung (3.22) dargestellten Zustände auftreten.

$$\dot{Q}_{kb,i} = \begin{cases} \dot{Q}_{k,b} - \dot{Q}_{k,ges}, & m_{on} = p_i \\ \dot{Q}_{max,i}, & m_{on} > p_i \\ 0, & m_{on} < p_i \end{cases} \tag{3.22}$$

Ist $m_{on} > p_i$, ist bereits die nächste Maschine auf dem Prioritätsvektor eingeschaltet worden und die betrachtete Maschine wird mit $\dot{Q}_{max,i}$ voll belastet. Ist $m_{on} < p_i$, wird der Kältebedarf von Maschinen mit niedriger Prioritätszahl gedeckt und die Maschine ist ausgeschaltet. Ist $m_{on} = p_i$, so muss die Maschine den Differenzbedarf von $\dot{Q}_{k,b}$ und $\dot{Q}_{k,ges}$ decken, d. h. die Maschine fängt die Bedarfsschwankungen ab. Aufgrund dieser Schaltlogik laufen immer $m_{on} - 1$ Maschinen unter Volllast und jene Maschine, für die gilt $m_{on} = p_i$, deckt den Differenzbedarf von $\dot{Q}_{k,b}$ und $\dot{Q}_{k,ges}$.

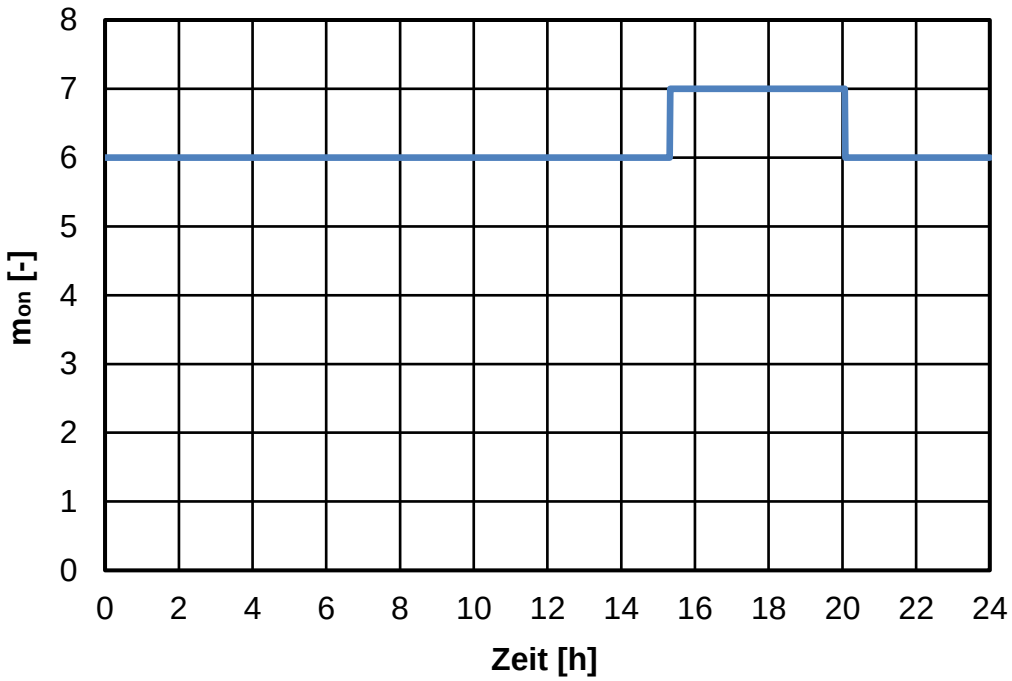

Abbildung 17: Zeitlicher Verlauf m_{on}

Um die Einschaltlogik besser zu veranschaulichen wird in Abbildung 17 der zeitliche Verlauf von m_{on} dargestellt. Es wird ein 24h Szenario simuliert. Es ist aus Abbildung 17 ersichtlich, dass für die ersten 15 Stunden des Szenarios 6 Maschinen ausreichen, um den Kältebedarf zu decken. Als Prioritätsvektor wird Beispielvektor $\vec{p}$ verwendet. D.h. bis Stunde 15 laufen die Maschinen 1, 4, 2, 6, 5 und 9. Die Maschinen 1, 4, 2, 6 und 5 laufen unter Volllast und Maschine 9 deckt den Differenzbedarf.

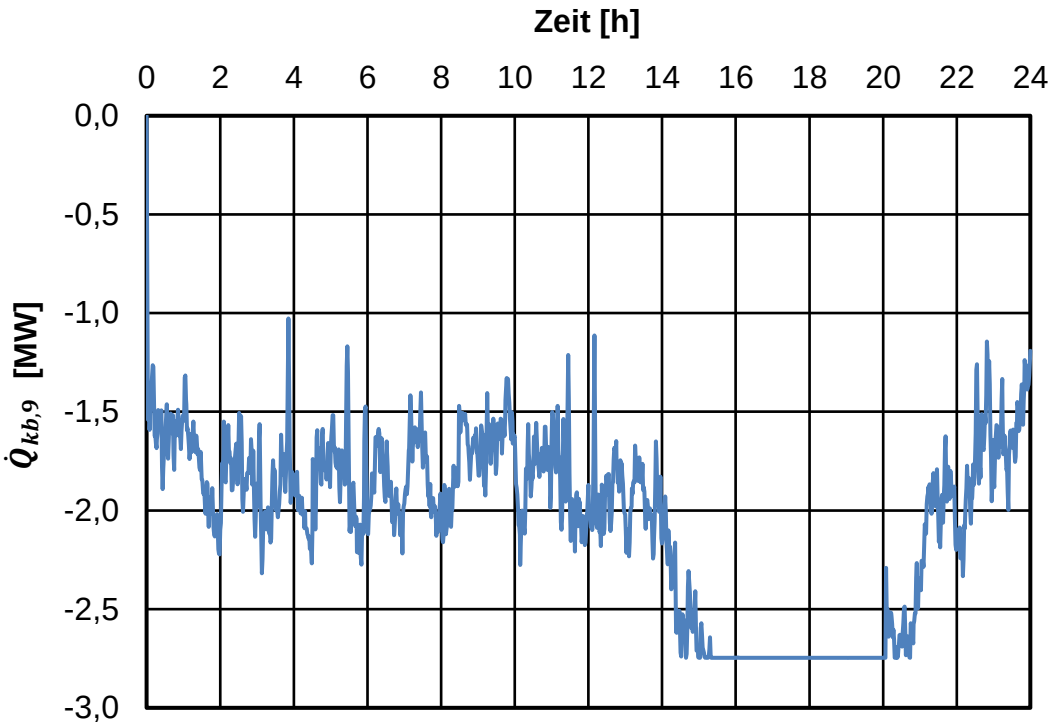

Abbildung 18: Zeitlicher Verlauf $\dot{Q}_{kb,9}$

In Abbildung 18 wird der Kältebedarf $\dot{Q}_{kb,9}$, der an Kältemaschine 9 geschickt wird abgebildet. Es ist deutlich ersichtlich, dass sich die Maschine bis Stunde 15 im Teillastbereich bewegt. Von 15h bis 20h ist die Maschine im Volllastbetrieb. Der Kältebedarf kann in diesem Zeitabschnitt nicht mehr durch 6 Maschinen gedeckt werden. Die Verteilfunktion reagiert darauf und erhöht m_{on} um 1 auf 7. So wird Maschine 3 ($p_{03} = 7$) eingeschaltet.

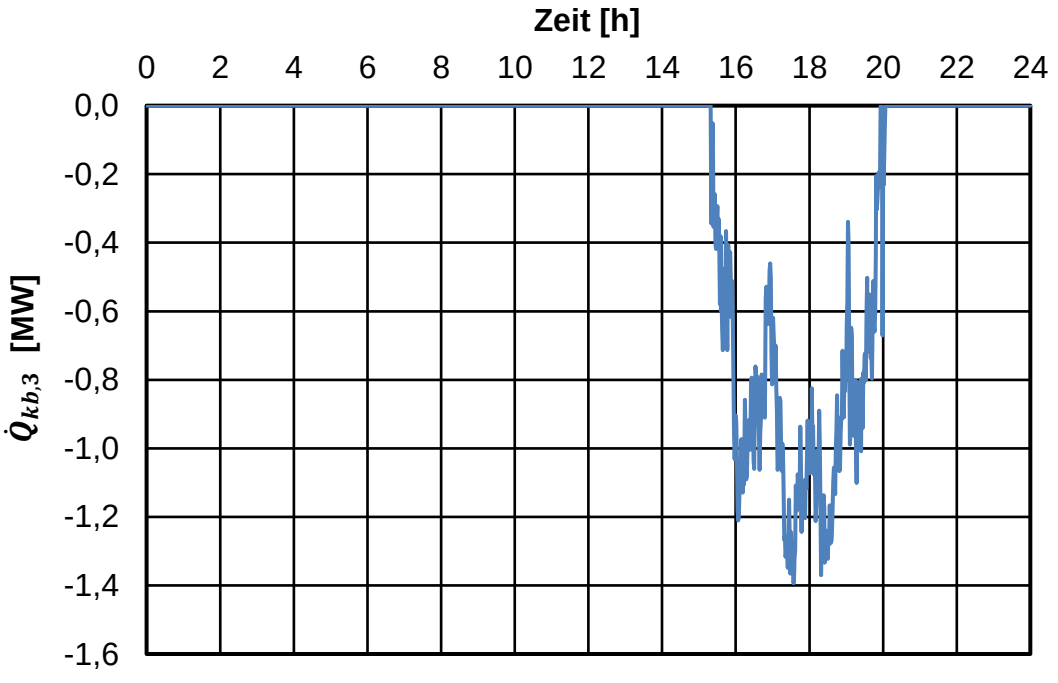

Abbildung 19: Zeitlicher Verlauf $\dot{Q}_{kb,3}$

In Abbildung 19 wird der zeitliche Verlauf von $\dot{Q}_{kb,3}$ abgebildet. Die Maschine ist bis Stunde 15 ausgeschaltet, wird dann von Stunde 15 bis Stunde 20 eingeschaltet und ab Stunde 20 wieder ausgeschaltet.

Um ein schnelles Ein- und Ausschalten aufgrund von Bedarfsschwankungen zu verhindern, wird für dieses Beispiel eine Hysterese $Hyst$ = 900 kW gewählt. Um die Unterschiede zu verdeutlichen wird derselbe Belastungsfall mit $Hyst$ = 100 kW und $Hyst$ = 500 kW ebenfalls simuliert. Die Ergebnisse werden in Abbildung 20 dargestellt. Aus der Abbildung ist ersichtlich, dass m_{on} stärker schwankt, je kleiner $Hyst$ gewählt wird. $Hyst$ sollte abhängig vom Lastfall so gewählt werden, dass nicht mehr als 4 Schaltvorgänge pro Maschine pro Stunde ausgeführt werden.

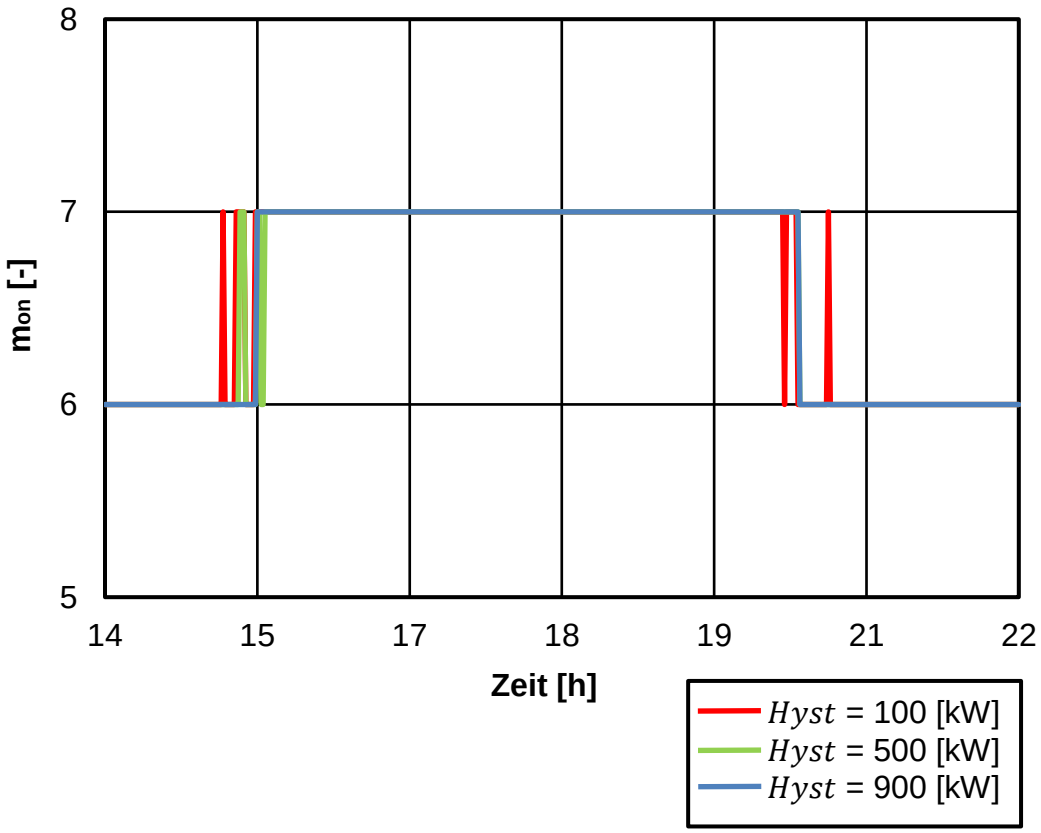

Abbildung 20: m_{on} in Abhängigkeit von $Hyst$

3.2.4 Zielfunktionsbaustein

Der Zielfunktionsbaustein hat die Aufgabe, alle relevanten Informationen aus der Simulation zu erfassen und daraus einen Zielfunktionswert zu errechnen. Der Optimierer versucht dann durch Variation der Einschaltreihenfolge den Zielfunktionswert zu minimieren. Das Minimum des Zielfunktionswertes stellt das Optimum dar. Der Zielfunktionsbaustein weißt ausschließlich Informationsports auf (siehe Abbildung 21).

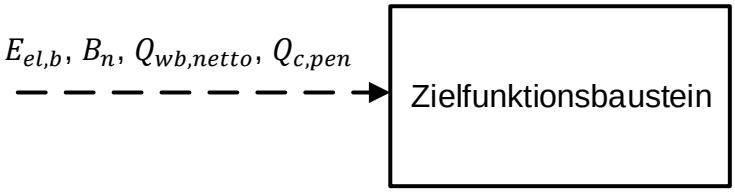

Abbildung 21: Informationsports Zielfunktionsbaustein

Dieser Baustein ist direkt auf dem Sketch implementiert, obwohl diese Rechenoperationen klassische Post-Processing Operationen sind. Aufgrund der eingeschränkten Möglichkeiten, die AMESIM bei Post-Processing Operationen bietet, ist die Implementierung direkt auf dem Sketch notwendig geworden.

Die externen Variablen der Ports werden in Tabelle 15 aufgelistet.

Variable	**Bezeichnung**	**Einheit**	**Datentyp**	**Input/Output**
$E_{el,b}$	Bedarf elektrische Energie	kWh	Double	Input
B_n	Betriebsstunden	h	Double	Input
$Q_{wb,netto}$	Nettowärmebedarf	kWh	Double	Input
$Q_{c,pen}$	Strafterm Kühlkapazität	kWh	Double	Input

Tabelle 15: Variablen der Ports des Zielfunktionsbausteins

Die Parameter des Zielfunktionsbausteins werden in Tabelle 16 aufgeführt.

Variable	**Bezeichnung**	**Einheit**	**Datentyp**
ω_n	Gewichtungsfaktor	-	Double
k_i	Kostenfaktor Betriebskosten Maschine i	-	Double
k_P	Kostenfaktor elektrische Energie	-	Double
k_Q	Kostenfaktor Wärme aus externen Quellen	-	Double
$B_{init,i}$	Initiale Betriebsstunden Maschine i	h	Double

Tabelle 16: Parameter Zielfunktionsbaustein

Die vom Zielfunktionsbaustein verwendeten internen Variablen werden in Tabelle 17 aufgelistet.

Variable	**Bezeichnung**	**Einheit**	**Datentyp**
f_n	Zielfunktionsbestandteil	-	Double
f	Zielfunktion	-	Double
$\bar{B}_{Gvor}$	Mittlere Betriebsstunden aller Maschinen vorher	h	Double
$B_{Gvor,i}$	Betriebsstunden Maschine i vorher	h	Double
$\bar{B}_{Gnach}$	Mittlere Betriebsstunden aller Maschinen nachher	h	Double
$B_{Gnach,i}$	Betriebsstunden Maschine i nachher	h	Double

Tabelle 17: Interne Variablen Zielfunktionsbaustein

Nachfolgend wird die im Zielfunktionsbaustein implementierte Zielfunktion erläutert. Sie besteht aus 5 Zielfunktionsbestandteilen. Der erste Zielfunktionsbestandteil ist in Gleichung (3.23) gegeben und bewertet den Einfluss der Kosten für elektrische Energie.

$$f_1 = \omega_1 E_{el,b} k_P \tag{3.23}$$

ω_1 ist der Gewichtungsfaktor für die elektrischen Energiekosten. Jeder Zielfunktionsbestandteil hat einen Gewichtungsfaktor. Die Gewichtungsfaktoren haben die Aufgabe, alle Bestandteile der Zielfunktion zu gewichten und auf einen Wertebereich zu bringen. Die Gewichtung wird je nach Anforderungen vorgenommen und kann je nach Optimierungsfall unterschiedlich ausfallen. Durch die Gewichtung können verschiedene Strategien verfolgt werden.

$E_{el,b}$ wird von der Simulation ermittelt und von der Verteilfunktion an den Zielfunktionsbaustein übertragen. Dazwischen ist noch ein Integrator-Glied geschaltet, das den Verlauf der elektrischen Leistung integriert und so den Energiebedarf in der Zeitperiode ermittelt (Gleichung (3.24)).

$$E_{el,b} = \int_{t_0}^{t_1} P_{el,b} dt \tag{3.24}$$

Der Kostenfaktor k_P dient dazu, das Verhältnis von elektrischen Energiekosten zu Energiekosten für Wärme aus externen Quellen abzubilden. Gleichung (3.25) zeigt den zweiten Zielfunktionsbestandteil. Dieser bewertet den Einfluss der Kosten für Wärme aus externen Quellen.

$$f_2 = \omega_2 Q_{wb,netto} k_Q \tag{3.25}$$

k_Q ist der Kostenfaktor für Wärme aus externen Quellen. $Q_{wb,netto}$ ist der Nettowärmebedarf im Simulationszeitraum. Verfügen die Kältemaschinen über ein Wärmerückgewinnungssystem wird die rückgewonnene Wärme Q_{wrg} ins Wärmenetz eingespeist. Sollte der Wärmebedarf $Q_{wb,brutto}$ nicht gedeckt werden können, wird der Differenzbedarf $Q_{wb,netto}$ durch die externe Wärmequelle gedeckt. Wird der Bruttowärmebedarf $Q_{wb,brutto}$ vollständig durch die Maschinen gedeckt, wird $Q_{wb,netto} = 0$ (Gleichung (3.26)).

$$Q_{wb,netto} = \begin{cases} Q_{wb,brutto} - Q_{wrg}, & Q_{wrg} < Q_{wb,brutto} \\ 0, & Q_{wrg} > Q_{wb,brutto} \end{cases} \tag{3.26}$$

Der dritte Zielfunktionsbestandteil ist in Gleichung (3.27) gegeben und berücksichtigt die Wartungskosten für die unterschiedlichen Maschinen.

$$f_3 = \omega_3 \sum B_i k_i \tag{3.27}$$

B_i stellt die Betriebszeit der Maschine i im Simulationsszenario dar. Die Betriebszeit pro Maschine wird von der Verteilfunktion ermittelt und an den Zielfunktionsbaustein weitergegeben. k_i ist der Betriebskostenfaktor für Maschine i.

Der 4. Zielfunktionsbestandteil (Gleichung (3.28)) bewertet die Belastung der Rückkühlwerke durch den Betrieb der Maschinen.

$$f_4 = \omega_4 Q_{c,pen} \tag{3.28}$$

Der Wärmestrom, der von den Rückkühlwerken an die Umgebung abgegeben werden kann ist mit $\dot{Q}_{c,max}$ limitiert. Sollte dieses Limit überschritten werden, wird aus dem Differenzwärmestrom $\dot{Q}_{c,pen}$ über einen Integrator die Differenzwärme $Q_{c,pen}$ gebildet und diese geht in den Zielfunktionswert mit ein und erhöht diesen. $Q_{c,pen}$ kann als Malus für die Überbeanspruchung der Rückkühlwerke verstanden werden.

Ein weiterer Effekt, der bei Überbeanspruchung der Rückkühlwerke auftritt ist eine Temperaturerhöhung der Kühlnetze, weil zu wenig Wärme an die Umgebung abgegeben werden kann. Als Folge daraus ändern sich die Wirkungsgrade und Kapazitäten der Kältemaschinen (siehe Kapitel 3.2.1).

Der 5. Zielfunktionsbestandteil berücksichtigt die Angleichung der Betriebsstunden aller Maschinen. Maschinen mit wenigen Betriebsstunden sollen länger betrieben werden, als Maschinen mit vielen Betriebsstunden. Um diesen Umstand mathematisch beschreiben zu können wird eine Kennzahl benötigt, die die Betriebsstundenabweichung aller Maschinen vom Mittelwert bewertet. Hierfür wird die mittlere absolute Abweichung vom arithmetischen Mittel d_A als Maßzahl für die Streuung gewählt (Gleichung 3.29) [36].

$$d_A = \frac{1}{n} * \sum_{i=1}^{n} |x_i - \bar{x}| \tag{3.29}$$

Die mittlere absolute Abweichung wird vor ($B_{Gvor,i}$, $\bar{B}_{Gvor}$) und nach ($B_{Gnach,i}$, $\bar{B}_{Gnach}$) dem Simulationsdurchlauf miteinander verglichen. Somit kann bewertet werden, ob die Maschinen so betrieben wurden, dass die Betriebsstunden sich angeglichen oder weiter voneinander entfernt haben. Über den Zielfunktionsbestandteil f_5 geht die mittlere absolute Abweichung der Betriebsstunden vor und nach dem Simulationsdurchlauf in den Zielfunktionswert mit ein (Gleichung (3.30)).

$$f_5 = \omega_5 \frac{1}{n} \sum (\left|B_{Gnach,i} - \bar{B}_{Gnach}\right| - \left|B_{Gvor,i} - \bar{B}_{Gvor}\right|) \tag{3.30}$$

Sollten sie sich angeglichen haben, wird f_5 negativ und reduziert den Zielfunktionswert. Dadurch verschiebt sich der gesamte Zielfunktionswert weiter in Richtung Optimum.

Die gesamte Zielfunktion f berechnet sich dann nach Gleichung (3.31) als Summe der Einzelkomponenten.

$$f = f_1 + f_2 + f_3 + f_4 + f_5 \tag{3.31}$$

Die Wahl der Gewichtungsfaktoren hat wesentlichen Einfluss auf das Optimierungsergebnis. Es können unterschiedliche Fragestellungen durch Variation der Gewichtungsfaktoren beantwortet werden. Nachfolgend wird auf einige Punkte eingegangen.

Bei einer Optimierung der gesamten Zielfunktion mit allen Zielfunktionsbestandteilen werden die Gewichtungsfaktoren so gewählt, dass ein ausgewogenes Verhältnis zwischen den einzelnen Bestandteilen besteht. Eine solche Wahl der Gewichtungsfaktoren wird in Kapitel 4.1.3 vorgenommen. Werden einzelne Gewichtungsfaktoren 0 gesetzt können spezifischere Fragestellungen beantwortet werden.

Werden der Gewichtungsfaktor ω_1 = 1 und die übrigen Gewichtungsfaktoren $\omega_2 = \omega_3 = \omega_4 = \omega_5 = 0$, sowie der Kostenfaktor $k_P = 1$ gesetzt, entspricht der Zielfunktionswert direkt dem elektrischen Energieverbrauch und der Optimierer minimiert diesen. Werden $\omega_1 = 1$ und $\omega_2 = 1$, sowie die Kostenfaktoren $k_P = 1$ und $k_Q = 1$ gesetzt so entspricht der Zielfunktionswert dem Gesamtenergieverbrauch (elektrische Energie und Wärme). Diese Wahl der Gewichtungsfaktoren wird zur Beurteilung des Einsparungspotenzials verwendet (siehe Kapitel 4.5).

Werden die Kostenfaktoren den aktuellen Strompreisen und Preisen für Wärme aus externen Quellen entsprechend gesetzt, entspricht der Zielfunktionswert direkt den Energiekosten und der Optimierer minimiert die Kosten. Es ist auch denkbar die Gewichtungsfaktoren entsprechend der CO_2-Emission von 1 kWh Strom und 1 kWh Wärme zu setzen und somit die CO_2-Emission vom Optimierer minimieren zu lassen.

3.2.5 Rückkühlwerk-Modell

Das Rückkühlwerk hat die Aufgabe, den Bedarf des Kühlnetzes an Kühlleistung zu decken. Dieser Bedarf entsteht aufgrund der Kältemaschinen, die an das Kühlnetz angeschlossen sind und dem Kaltwasser Wärme entziehen und dabei Wärme an das Kühlwasser abgeben. Das Rückkühlwerk führt somit die überschüssige Wärme des durch die Kältemaschinen erwärmten Kühlwassers an die Umgebung ab. In Abbildung 22 werden die Energie- und Informationsports des Rückkühlwerk-Modells abgebildet.

Abbildung 22: Ports des Rückkühlwerks

Die externen Variablen der Ports werden in Tabelle 18 dargestellt.

Variable	**Bezeichnung**	**Einheit**	**Datentyp**	**Input/Output**
$\dot{Q}_{c,b}$	Bedarf Kühlleistung	kW	Double	Input
$\dot{Q}_{c,erz}$	Abgegebene Kühlleistung	kW	Double	Output
$\dot{Q}_{c,pen}$	Kühlleistungsmalus	kW	Double	Output

Tabelle 18: Variablen der Ports des Rückkühlwerkmodells

Die Parameter des Rückkühlwerks werden in Tabelle 19 aufgeführt.

Variable	**Bezeichnung**	**Einheit**	**Datentyp**
$\dot{Q}_{c,max}$	Maximale Kühlleistung	kW	Double

Tabelle 19: Parameter Rückkühlwerk

Das Rückkühlwerk verwendet keine internen Variablen.

Der Wärmestrom $\dot{Q}_{c,erz}$, der vom Rückkühlwerk an die Umgebung abgegeben werden kann ist mit $\dot{Q}_{c,max}$ limitiert (Gleichung (3.32)). Wird das Maximum überschritten, wird die Differenz aus Bedarf $\dot{Q}_{c,b}$ und Maximum $\dot{Q}_{c,max}$ als Kühlleistungsmalus $\dot{Q}_{c,pen}$ über einen Integrator an den Zielfunktionsbaustein weitergeleitet (Gleichung 3.33)). Außerdem erhöht sich die Temperatur im Energienetz, weil nicht genügend Wärme über das Rückkühlwerk an die Umgebung abgegeben werden kann (siehe Kapitel 3.2.2).

$$\dot{Q}_{c,erz} = \begin{cases} \dot{Q}_{c,b}, & \dot{Q}_{c,b} < \dot{Q}_{c,max} \\ \dot{Q}_{c,max}, & \dot{Q}_{c,b} > \dot{Q}_{c,max} \end{cases} \tag{3.32}$$

$$\dot{Q}_{c,pen} = \begin{cases} \dot{Q}_{c,b} - \dot{Q}_{c,max}, & \dot{Q}_{c,b} > \dot{Q}_{c,max} \\ 0, & \dot{Q}_{c,b} < \dot{Q}_{c,max} \end{cases} \tag{3.33}$$

3.3 Aufbau der lauffähigen Simulation

Die im vorigen Kapitel beschriebenen ausführbaren Einzelmodelle werden in diesem Kapitel auf Grundlage des Konzeptmodells (siehe Kapitel 2.2.1) zu einer funktionierenden Simulation verschalten. Alle Outputs eines Cube werden zu den entsprechenden Inputs eines anderen Cube geleitet. Diese Signalweiterleitung erfolgt in AMESIM über Verbindungslinien.

In Abbildung 25 wird die Sketch-Ansicht aller verschalteten Komponenten dargestellt. Die rot dargestellten Bausteine sind Standardkomponenten und stammen aus der AMESIM-Bibliothek „Signal, Control“, die schwarz dargestellten Bausteine sind hingegen benutzerdefinierte Komponenten. Eine Übersicht über alle verwendeten Standardkomponenten wird in Abbildung 23 gegeben. Alle benutzerdefinierten Komponenten werden in Abbildung 24 dargestellt. Es handelt sich um das Simulationsmodell einer Kälteanlage, die aus 12 Kältemaschinen (Abbildung 24, j) aufgebaut ist. Diese 12 Kältemaschinen sind an einen Kaltwasserkreislauf angeschlossen, der über das Kältenetz 1 in der Simulation abgebildet wird. Auf Kondensatorseite werden die Kältemaschinen über 2 Kühlkreisläufe mit Kühlleistung versorgt. Dabei versorgt Kühlnetz 1 6 Maschinen und Kühlnetz 2 6 Maschinen. Die Kälte- und Kühlnetze werden mit dem Energienetz-Baustein dargestellt (Abbildung 24, a). Die Kühlnetze sind an jeweils ein Rückkühlwerk (Abbildung 24, e) angeschlossen.

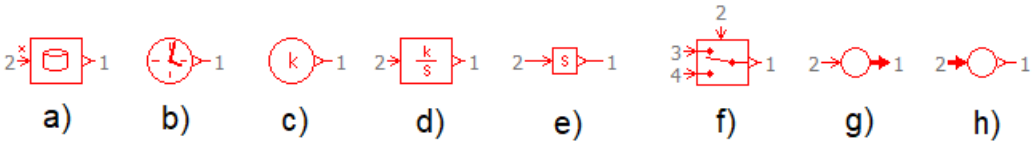

Abbildung 23: Standardkomponenten aus AMESIM-Bibliothek

Der Kälte- und Wärmebedarf wird über die Bausteine im mit „Produktionsanlagen/Bürogebäude“ bezeichneten Rechteck aufgebracht. Der Kältebedarf ist eine Inputgröße, die sich im Zeitverlauf ändert. Der Tabellenbaustein (Abbildung 23, a) ist deswegen an einen als Uhr dargestellten Baustein angebunden (Abbildung 23, b). Dieser gibt die Simulationszeit t in Sekunden aus. Der Tabellenbaustein liest dann aus einer Tabelle den zugehörigen Kältebedarf aus. Dabei ist es unerheblich, wie hoch der Kältebedarf in der Tabelle aufgelöst ist. Fehlende Werte werden vom Tabellenbaustein interpoliert. Der Kältebedarf ist als Simulationsinput immer negativ. Deshalb wird mit einem Kontrollbaustein (Abbildung 24, f) das Vorzeichen der vom Tabellenbaustein ausgegebenen Werte kontrolliert. Im Bedarfsfall dreht der Kontrollbaustein das Vorzeichen auf negativ. Der Wärmebedarf ist nicht zeitabhängig und somit konstant. Dieser wird über einen Konstanten-Baustein aufgebracht (Abbildung 23, c). Im Gegensatz zum Kältebedarf ist der Wärmebedarf als Simulationsinput immer positiv. Hier wird ebenfalls ein Kontrollbaustein verwendet, der im Bedarfsfall das Vorzeichen auf positiv ändert. Durch die beiden

Kontrollbausteine wird verhindert, dass Kältebedarf und Wärmebedarf mit falschem Vorzeichen in die Simulation eingebracht werden.

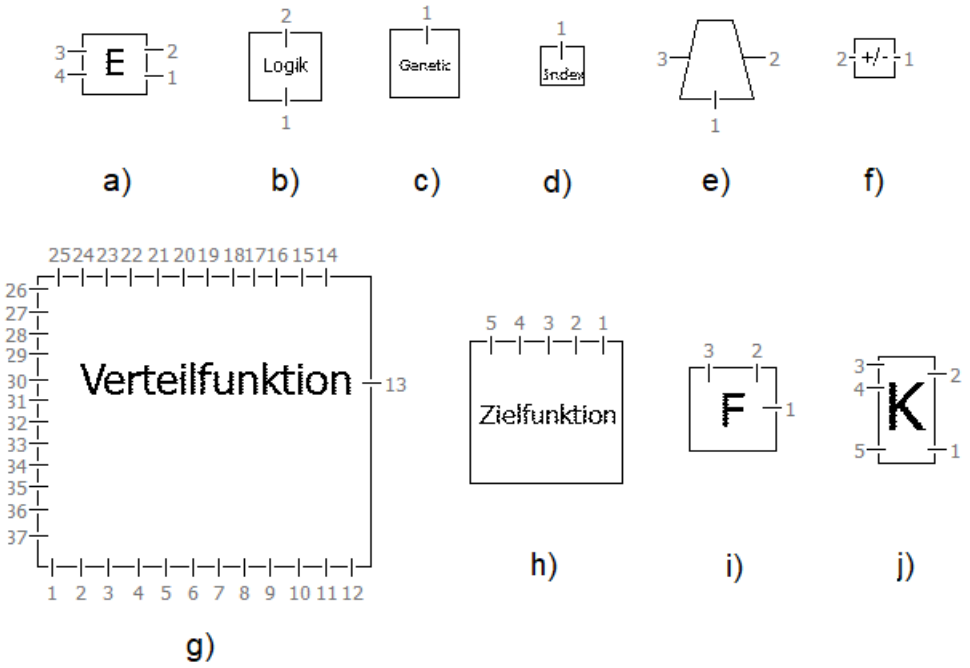

Abbildung 24: Benutzerdefinierte Komponenten

Vom Kältenetz 1 führt die Verbindungslinie hin zur Verteilfunktion (Abbildung 24, g). Über diese Linie wird der Gesamtkältebedarf an die Verteilfunktion übertragen. Die Verteilfunktion teilt diesen Bedarf entsprechend der Schaltlogik auf die Maschinen auf. Gleichzeitig bekommt sie von jeder einzelnen Maschine die Rückmeldung über die aktuell zur Verfügung stehende Kapazität, den durch das Wärmerückgewinnungssystem gewonnene Wärmestrom und den elektrischen Leistungsbedarf. Der elektrische Leistungsbedarf wird für alle Maschinen aufsummiert und an den Zielfunktionsbaustein (Abbildung 24, h) weitergeleitet. Dazwischen sitzt noch ein Integrator-Baustein (Abbildung 23, d), der das Signal nach der Zeit integriert und so aus der elektrischen Leistung die elektrische Energie ermittelt. Der rückgewonnene Wärmestrom wird an den Baustein „externe Wärmequelle“ (Abbildung 24, i) weitergegeben. Sollte der Wärmebedarf mit dem rückgewonnenen Wärmestrom nicht gedeckt werden können wird der Differenzwärmestrom von der externen Wärmequelle bereitgestellt. Aus diesem Differenzwärmestrom wird durch Integration nach der Zeit der Nettowärmebedarf gebildet. Dieser wird ebenfalls an den Zielfunktionsbaustein weitergeleitet. Außerdem leitet die Verteilfunktion noch die Betriebszeiten der Maschinen im Simulationszeitraum an den Zielfunktionsbaustein weiter. Die Temperaturen der Kühlnetze werden ohne Verbindungslinien über einen Sender (Abbildung 23, g) und mehrere Empfänger (Abbildung 23, h) an die Kältemaschinenbausteine übertragen. Diese Möglichkeit der Übertragung wird gewählt, da Verbindungslinien der Übersichtlichkeit schaden würden. Die Zugehörigkeit eines Empfängers zu seinem Sender wird in Abbildung 25 über einen

Buchstaben symbolisiert. Kühlnetz 1 sendet seine Temperatur an alle mit Buchstaben a gekennzeichneten Empfänger, Kühlnetz 2 an alle Empfänger mit Buchstabe b. Sender und Empfänger mit Buchstabe c werden verwendet, um eine Überschreitung der Kühlkapazitäten von Rückkühlwerk 1 oder 2 an den Zielfunktionsbaustein zu übertragen. Die Temperatur von Kältenetz 1 wird mit den Verbindungslinien, die den Kältebedarf übertragen, an die Kältemaschinen mitgeschickt. Hier entfallen Sender und Empfänger.

Der Prioritätsvektor wird über die mit „INPUT PRIORITY“ gekennzeichneten Bausteine an die Verteilfunktion weitergegeben. Je nachdem, ob gerade die Permutationsliste durchläuft oder die Optimierung durch den genetischen Algorithmus stattfindet, wird der Parameter des Switch-Bausteins (Abbildung 23, f) auf 0 oder 1 gestellt und somit die Outputs vom richtigen Tabellenbaustein verwendet. Die an den Tabellenbaustein angeschlossenen Bausteine „Genetic“ (Abbildung 24, c) und „Index“ (Abbildung 24, d) haben als Output jene Variable, die vom Optimierer variiert wird. So wird eine Zeile im Tabellenbaustein angewählt. Die „Logik“-Bausteine (Abbildung 24, b) wandeln die Tabellenzeile in 12 Prioritätszahlen um, die an die Verteilfunktion weitergeleitet werden. Die Logik, die bei der Konvertierung des Tabelleneintrags in die von der Verteilfunktion verwendeten Prioritätszahlen zur Anwendung kommt, wird in Kapitel 3.4.1 beschrieben.

Zwischen Energienetz und Kältemaschine tritt eine Rückkopplungsschleife auf, weil der angeforderte Wärmestrom $\dot{Q}_{erz,b}$ direkt vom Energiestand E des Netzes abhängig ist, und dieser wiederrum von der Differenz aus zugeführtem $\dot{Q}_{erz}$ und abgegebenem Wärmestrom $\dot{Q}_{bez}$. Weil das Kältemaschinenmodell bei ausreichender Stromversorgung genau so viel Kälte liefert, wie angefordert wird, kann $\dot{Q}_{erz,b} = \dot{Q}_{erz}$ angenommen werden. Die Variable $\dot{Q}_{erz,b}$ ist somit von sich selbst abhängig. Solche Rückkopplungen treten in dynamischen Systemen häufig auf und werden als algebraische Schleifen bezeichnet [37]. Sie können entweder durch Einführen einer Hilfsvariable oder durch eine Zeitverzögerung aufgelöst werden. In der Simulation wird diese Problematik durch eine Zeitverzögerung gelöst. In Abbildung 25 wird diese Zeitverzögerung auf das Signal aufgebracht, indem ein Integrator und ein Differentiator (Abbildung 23, e) direkt hintereinandergeschaltet werden. Das Signal wird integriert und unter Verwendung einer minimalen Zeitverzögerung τ wieder abgeleitet. Deshalb liefert die Kältemaschine erst um die Zeit τ verzögert die angeforderte Kälteleistung. Eine Alternative wäre die Verwendung eines dezidierten Zeitverzögerungsbausteins, aber Versuche haben gezeigt, dass sich die Rechenzeiten dadurch nicht verbessern. Deshalb wird diese Variante bevorzugt. Physikalisch kann diese Zeitverzögerung ebenfalls erklärt werden. Im realen System wird eine Kältemaschine nie sofort auf einen geänderten Kältebedarf reagieren, wodurch im realen System hier ebenfalls immer eine Zeitverzögerung auftritt.

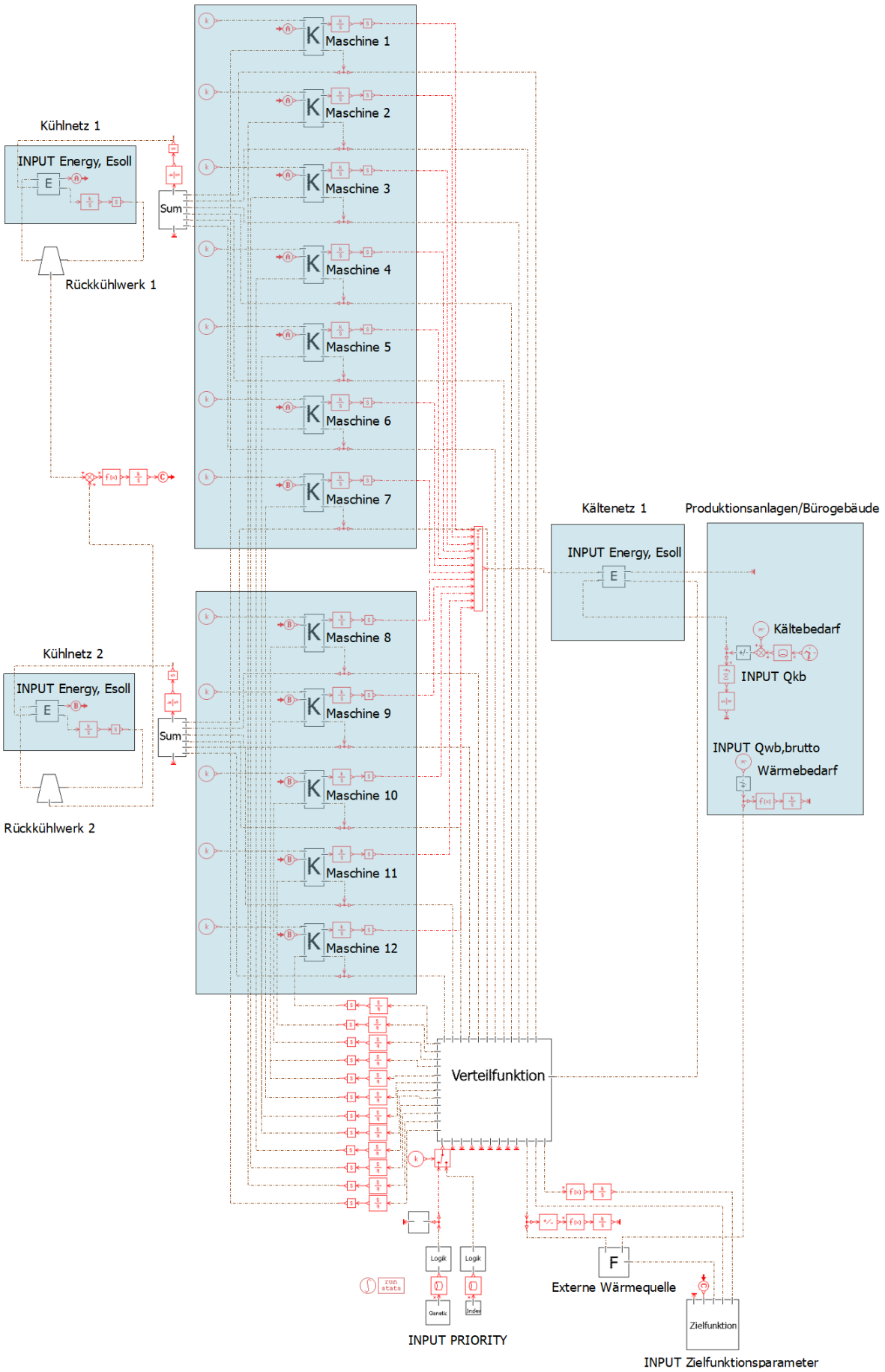

Abbildung 25: Sketch-Ansicht der verschalteten Komponenten

3.4 Optimierung

In diesem Kapitel wird die Implementierung des Optimierers in AMESIM beschrieben. Die Implementierung des Optimierers stellt aufgrund programmspezifischer Limitierungen bei der Definition von Randbedingungen für den Optimierungsalgorithmus eine besondere Herausforderung dar. Der Freiheitsgrad des Optimierers ist die Einschaltreihenfolge der Maschinen. Die daraus resultierenden Randbedingungen werden in Kapitel 2.3.6 erläutert.

Der gewählte Ansatz ist eine Kombination verschiedener Verfahren. Die Optimierung erfolgt in zwei Schritten: im ersten Schritt wird eine fest vorgegebene Liste an Prioritätsvektoren durchsimuliert (Permutationsliste). Die Ergebnisse werden ausgewertet und damit die ersten 3 Stellen des optimalen Prioritätsvektors bestimmt. Die restlichen 9 Stellen werden mit einem, in AMESIM bereits integrierten, genetischen Algorithmus ermittelt.

Nachfolgend wird auf die Vektordarstellung in AMESIM eingegangen, sowie die Implementierung der Permutationsliste und die Anwendung des genetischen Algorithmus über eine Prioritätsliste beschrieben.

3.4.1 Implementierung des Prioritätsvektors

Ein Vektor muss in AMESIM als Variablenset dargestellt werden. Gleichung (3.34) beschreibt diesen Zusammenhang. Der Vektor $\vec{p}$ besteht aus den Variablen v_1 bis v_{12}.

$$\vec{p} = (v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}, v_{11}, v_{12}) \tag{3.34}$$

Der Vektor muss die harten Randbedingungen des Optimierungsproblems erfüllen (siehe Kapitel 2.3.6). Prioritätsvektoren, die diese Randbedingungen nicht erfüllen sind ungültig. Wird versucht die Simulation mit einem ungültigen Prioritätsvektor zu starten, bricht die Verteilfunktion die Simulation ab. Versucht also AMESIM den Vektor zufallsbasiert aufzubauen und jeden Vektoreintrag unabhängig von den anderen zufällig zu wählen, ist die Wahrscheinlichkeit hoch, dass ein ungültiger Prioritätsvektor verwendet wird. Der Aufbau eines gültigen Prioritätsvektors stellt für AMESIM deshalb ein großes Hindernis dar. Um dieses Problem zu lösen, müssen dem Programm vorberechnete, gültige Vektoren vorgegeben werden. Diese Vektoren werden als Tabelle gespeichert und vor Beginn der Simulation ausgelesen.

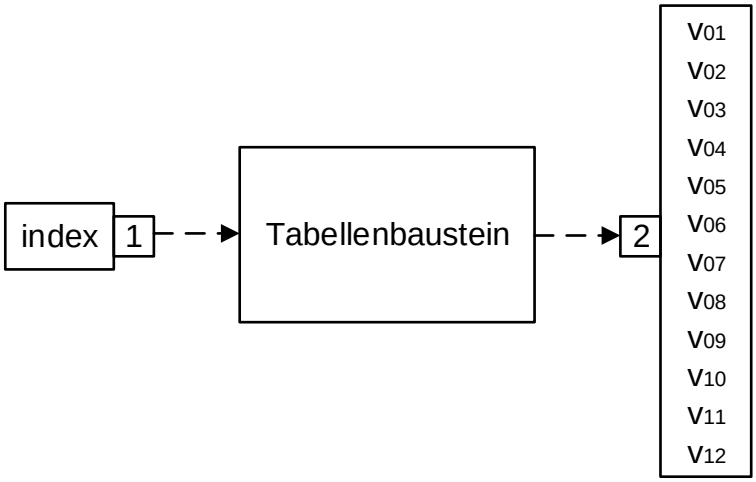

Abbildung 26: Input und Outputs Tabellenbaustein

Realisiert wird das mit einem Tabellenbaustein. Dieser Baustein hat als Input die entsprechende Zeilennummer als Variable $index$ und gibt den Prioritätsvektor in Form eines Variablensets aus (siehe Abbildung 26). Diese index-Variable wird vom Optimierungsalgorithmus variiert. So können vom Algorithmus unterschiedliche, gültige Vektoren angewählt werden. In Tabelle 20 wird ein Beispiel eines Tabelleneintrags dargestellt. Diese Zeile kann angewählt werden, wenn die Variable $index$ auf 54 gesetzt wird.

$index$	v_{01}	v_{02}	v_{03}	v_{04}	v_{05}	v_{06}	v_{07}	v_{08}	v_{09}	v_{10}	v_{11}	v_{12}
54	8	6	9	3	5	12	7	10	1	11	2	4

Tabelle 20: Beispiel für Tabelleneintrag

Die durch die Tabellenzeile festgelegte Einschaltreihenfolge der Maschinen lautet: Maschine 8, Maschine 6, Maschine 9, Maschine 3, usw.

Die Verteilfunktion schaltet die Maschinen durch Erhöhen oder Senken der Variable m_{on}. m_{on} wird mit der der Maschine i zugeordneten Prioritätszahl p_i verglichen, um festzustellen, ob von der Maschine Kälte angefordert werden muss. Die Prioritätszahlen sind fest mit der Maschine verbunden (p_{01} legt die Priorität von Maschine 1 fest). Um also die dargestellte Einschaltreihenfolge zu verwirklichen muss Maschine 8 die Prioritätszahl 1 ($p_{08} = 1$) zugeordnet werden, Maschine 6 die Prioritätszahl 2 ($p_{06} = 2$), usw.

In Abbildung 27 wird die Konvertierung vom Tabellenoutput v_i in die Prioritätszahlen p_i grafisch veranschaulicht. Mit diesen Prioritätszahlen starten die Maschinen in der richtigen Reihenfolge. Wenn die Verteilfunktion mit $m_{on} = 1$ beginnt, gilt $p_{08} \geq 1$ und an Maschine 8 wird eine Kältelast weitergeleitet, bei $m_{on} = 2$ gilt $p_{08} \geq 2$ und $p_{06} \geq 2$ und an Maschine 8 und 6 wird Kältelast weitergeleitet usw.

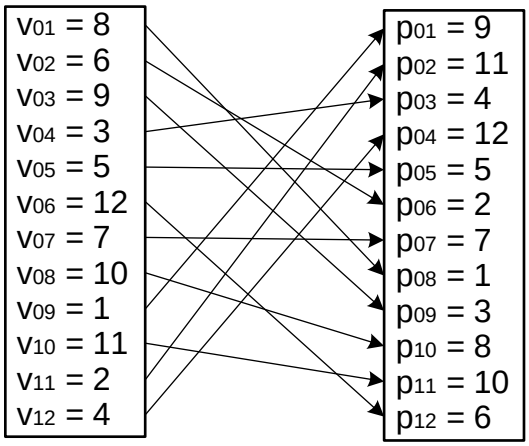

Abbildung 27: Konvertieren des Tabellenoutputs v_i in den Prioritätszahlen p_i

Diese Konvertierung übernimmt der Logik-Baustein. In Abbildung 28 wird die Umsetzung in Programmcode dargestellt. Es werden eine while-Schleife, zwei Laufvariablen und das Array p verwendet. Arrays sind Datenspeicher, die eine bestimmt Menge an Elementen aufnehmen können. Über die if-Bedingungen werden die Inputs des Logikbausteins v_{01} bis v_{12} (die vom Tabellenbaustein stammen) mit der Variable i verglichen. Vor dem ersten Durchlauf der Schleife werden i = 1 und j = 0 (Zeile 1 und 2, Abbildung 28) gesetzt. Die Schleife wird das erste Mal durchlaufen und dem obigen Beispiel folgend wird die if-Bedingung bei v_{09} erfüllt (Zeile 37, Abbildung 28). Zeile 39 wird ausgeführt und dem ersten Element des Arrays wird der Wert 9 zugewiesen (p[0] = 9). Danach werden die Laufvariablen i und j um jeweils 1 erhöht (Zeile 53 und 54, Abbildung 28). Der nächste Schleifendurchlauf beginnt. Das wird so lange ausgeführt bis die Bedingung j < 12 nicht mehr erfüllt ist. Zum Schluss werden die Elemente des Arrays p in die Variablen p_{01} bis p_{12} geschrieben und diese Variablen an die Verteilfunktion weitergeleitet. Dann schaltet die Verteilfunktion dem Tabelleneintrag (siehe Tabelle 20) entsprechend die Maschinen.

```
i = 1;
j = 0;
while (j < 12)
{
    if (*v01 == i)
        {
            p[j] = 1;
        }
    if (*v02 == i)
        {
            p[j] = 2;
        }
    if (*v03 == i)
        {
            p[j] = 3;
        }
    if (*v04 == i)
        {
            p[j] = 4;
        }
    if (*v05 == i)
        {
            p[j] = 5;
        }
    if (*v06 == i)
        {
            p[j] = 6;
        }
    if (*v07 == i)
        {
            p[j] = 7;
        }
    if (*v08 == i)
        {
            p[j] = 8;
        }
    if (*v09 == i)
        {
            p[j] = 9;
        }
    if (*v10 == i)
        {
            p[j] = 10;
        }
    if (*v11 == i)
        {
            p[j] = 11;
        }
    if (*v12 == i)
        {
            p[j] = 12;
        }
    i = i + 1;
    j = j + 1;
}
```

Abbildung 28: Konvertierung von v_i auf p_i in Programmcode umgesetzt

3.4.2 Implementierung der Permutationsliste

Bevor die Optimierung mittels genetischem Algorithmus beginnt ist es notwendig, eine Permutationsliste zu simulieren. Dieser Schritt bietet implementierungstechnische Vorteile, da sich die Größe der für den genetischen Algorithmus notwendigen Vektortabelle von 12! auf 9! reduziert. Ziel der Permutation ist es, die ersten 3 Stellen des finalen Prioritätsvektors zu bestimmen und so den Suchraum für den genetischen Algorithmus von 12! auf 9! zu reduzieren. Die Maschinen, die nach der Auswertung der Permutation den niedrigsten Zielfunktionswert erzielen, werden auch im finalen Prioritätsvektor ganz nach vorne gereiht. Diese Maschinen werden dann als erstes eingeschaltet und laufen die längste Zeit.

Die Permutationsliste ist aus 6600 Vektoren aufgebaut, wobei die ersten 3 Stellen jedes Prioritätsvektors permutiert werden. Die restlichen 9 Stellen des Vektors werden mit Zufallswerten aufgefüllt. Jede Permutation ist mit 5 Vektoren in der Liste vertreten. So soll der Zufallseinfluss der aufgefüllten Stellen reduziert werden. Daraus ergibt sich der Gesamtumfang von 6600 Vektoren.

$$\vec{p}_1 = (1,2,3,8,6,7,5,12,11,10,9,4) \tag{3.35}$$

Gleichung (3.35) ist ein Beispielvektor der Permutationsliste. Der grün markierte Teil wird permutiert und der gelb markierte Teil per Zufall bestimmt.

Nachdem die Permutationsliste durchsimuliert wurde und für jeden Eintrag ein entsprechender Zielfunktionswert vorliegt, werden die Ergebnisse mithilfe einer externen Software (Excel oder Matlab) ausgewertet, da AMESIM die dafür notwendigen Funktionalitäten nicht bereitstellt. Die Permutationsliste wird nach Zielfunktionswerten sortiert. Vektoren, die in den permutierten Stellen geeignete Maschinen für das Szenario haben werden dadurch nach oben gereiht. Es ist möglich, dass für das Szenario geeignete Maschinen durch die Zufallsstellen nach unten gereiht werden, aber dadurch, dass jede Permutation mit 5 verschiedenen Vektoren vertreten ist, wird der Einfluss des Zufalls reduziert. Anschließend wird ermittelt, wie oft jede Maschine in den ersten 3 Stellen (Permutationsstellen, grün markiert) der 100 Vektoren mit dem niedrigsten Zielfunktionswerten vorkommt. Die 3 Maschinen, die am häufigsten vorkommen sind am besten für dieses Simulationsszenario geeignet und dienen als Vorgabewert für den genetischen Algorithmus.

3.4.3 Anwendung des genetischen Algorithmus

AMESIM bietet neben anderen Optimierungsverfahren auch einen genetischen Algorithmus als Optimierungsmöglichkeit an. Die Einstellungsmöglichkeiten für die Randbedingungen sind allerdings begrenzt. Es können zwar Ober- und Untergrenzen für einzelne Variablen festgelegt werden, aber es besteht keine Möglichkeit die in

Kapitel 2.3.6 beschriebenen Randbedingungen einzustellen. Deshalb müssen dem Programm, wie in vorangegangenen Kapiteln beschrieben, gültige Vektoren über eine Tabelle vorgegeben werden. AMESIM variiert die Variable $index$, die der Zeilennummer der Tabelle entspricht und wählt so Prioritätsvektoren für die Simulation aus. Da im vorangegangenen Schritt die ersten 3 Stellen des finalen Prioritätsvektors bereits bestimmt wurden, reduziert sich der Suchraum für den Algorithmus von 12! auf 9! Vektoren.

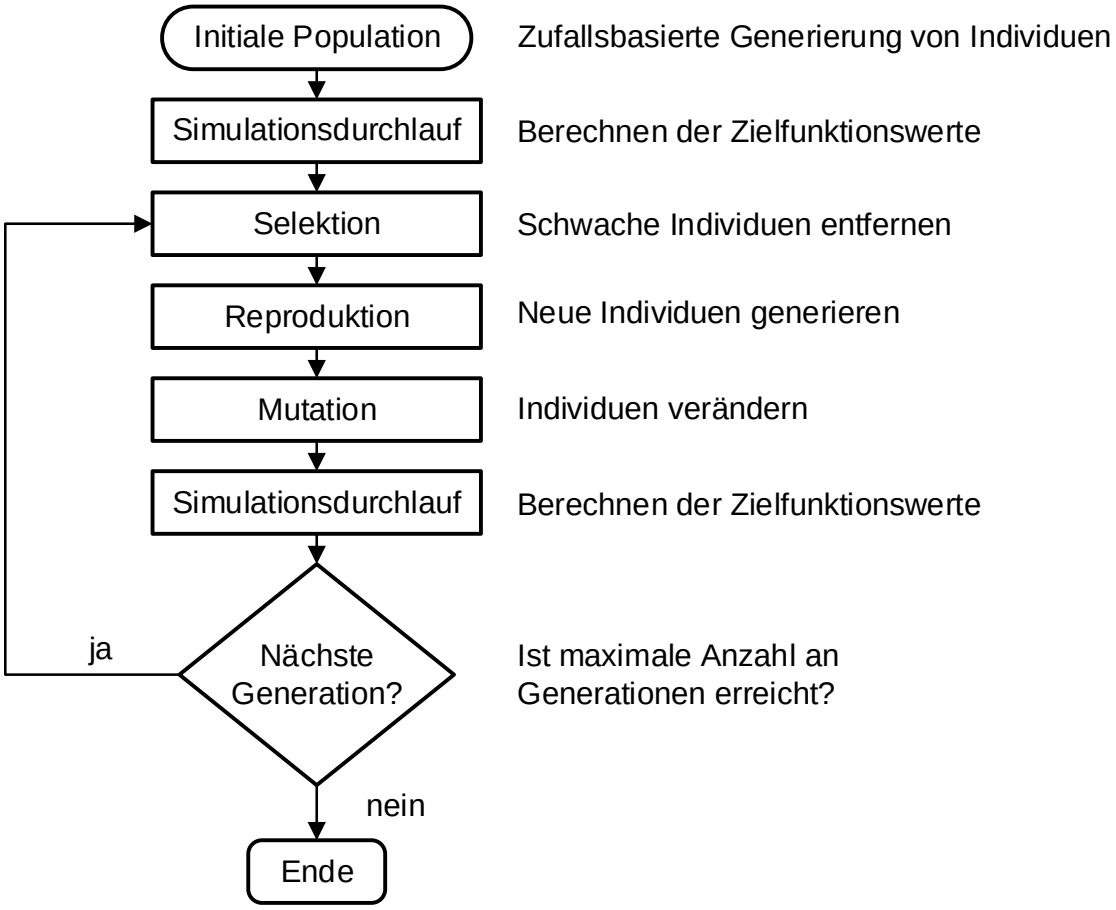

Abbildung 29: Arbeitsweise des GA in AMESIM in Anlehnung an [34]

Die Arbeitsweise des genetischen Algorithmus in AMESIM wird in Abbildung 29 dargestellt. Zu Beginn wird eine zufällige initiale Population an Individuen erstellt. Für diese Individuen wird der Zielfunktionswert ermittelt. Dann erfolgt eine Selektion der besten Individuen und schwache Individuen werden entfernt. Die Reproduktion erfolgt durch Rekombination der Elternindividuen und füllt die Generation um die entfernten Individuen wieder auf. Mit einer bestimmten Wahrscheinlichkeit werden die neuen Individuen durch Mutation verändert. Dann werden wieder die Zielfunktionswerte für alle Individuen errechnet. Wenn die maximale Anzahl an Generationen erreicht wurde, bricht die Optimierung ab und gibt das Ergebnis aus. Ist die maximale Anzahl noch nicht erreicht worden, wird eine neue Generation durch die Selektion gestartet.

AMESIM bietet folgende Einstellungsmöglichkeiten für den genetischen Algorithmus:

- Populationsgröße N [-]: Die Anzahl an unterschiedlichen Individuen in einer Population.
- Reproduktionsrate r [%]: Gibt an, wie viele Prozent einer Population bei jedem Reproduktionszyklus durch neue Individuen ersetzt werden.
- Maximale Anzahl Generationen G [-]: Die Anzahl an Reproduktionszyklen, die durchgeführt werden.
- Mutationswahrscheinlichkeit P [%]: Gibt die Wahrscheinlichkeit für jedes Individuum einer Population an, durch Mutation verändert zu werden.
- Mutationsamplitude α [-]: Ist eine reelle Zahl im Wertebereich von 0 bis 1 und beeinflusst die Standardabweichung der Mutationen eines Individuums. Wird dieser Wert nahe 0 gesetzt, konvergiert der Algorithmus schneller zu einer bestimmten Lösung, aber das Risiko steigt, dass es sich um ein lokales und kein globales Optimum handelt. Wird der Wert nahe 1 gesetzt, so wird der Suchraum breiter abgesucht, allerdings reduziert sich die Geschwindigkeit mit der der Algorithmus gegen eine Lösung konvergiert.

Die Anzahl an notwendigen Simulationsdurchläufen R (auch als Anzahl an Iterationen bezeichnet) mit den gewählten Einstellungen lässt sich mit Gleichung (3.36) abschätzen.

$$R \approx N + \frac{rN(G-1)}{100} \tag{3.36}$$

Die Parameter P und α beeinflussen die Anzahl der Iterationen nicht. Sie werden entsprechend der Empfehlung der Programmdokumentation mit P = 10% und α = 0,2 gewählt.

Die Parameter N, r und G beeinflussen direkt die Anzahl der Iterationen und somit die Rechenzeit. Um eine gute Balance zwischen Güte der gefundenen Lösung und Rechenzeit zu finden werden 4 verschiedene Parametersets getestet. Es wird ein Beispielszenario mit diesen Parametersets optimiert und der Verlauf des Zielfunktionswertes mit zunehmender Anzahl an Iterationen betrachtet. In Tabelle 21 werden die getesteten Parametersets aufgeführt. Die Anzahl an Generationen G wird so gewählt, dass etwa 1000 Iterationen erreicht werden.

Parameterset	N **[-]**	r **[%]**	G **[-]**	R **[-]**
1	100	80	13	1060
2	50	80	25	1010
3	100	60	16	1000
4	50	60	33	1010

Tabelle 21: Getestete Parametersets für genetischen Algorithmus

Die initiale Population wird zufällig bestimmt. Die Zielfunktionswerte der 1. Generation streuen deshalb sehr stark. Erst durch Selektion der stärksten Individuen und anschließender Rekombination nimmt die Streuung langsam ab. D.h. der Algorithmus nähert sich mit jeder Generation weiter dem Optimum an. Aufgrund der großen Streuung ist eine direkte Betrachtung der Zielfunktionswerte nicht zweckmäßig. Wichtiger ist es den Verlauf der Zielfunktionsmittelwerte über die Iterationsschritte ablesen zu können. So kann bestimmt werden, wie schnell sich der Algorithmus mit den verschiedenen Parametersets dem Optimum nähert. Zur Glättung wird der gleitende Mittelwert mit Periode 100 verwendet.

Das bedeutet, dass aus den ersten 100 Iterationsschritten das arithmetische Mittel gebildet wird. Dieser Punkt wird aufgetragen. Im nächsten Schritt wird aus den Schritten 1 bis 101 das Mittel gebildet und dieser Punkt aufgetragen. So ergeben sich die in Abbildung 30 dargestellten gleitenden Zielfunktionsmittelwerte der Parametersets.

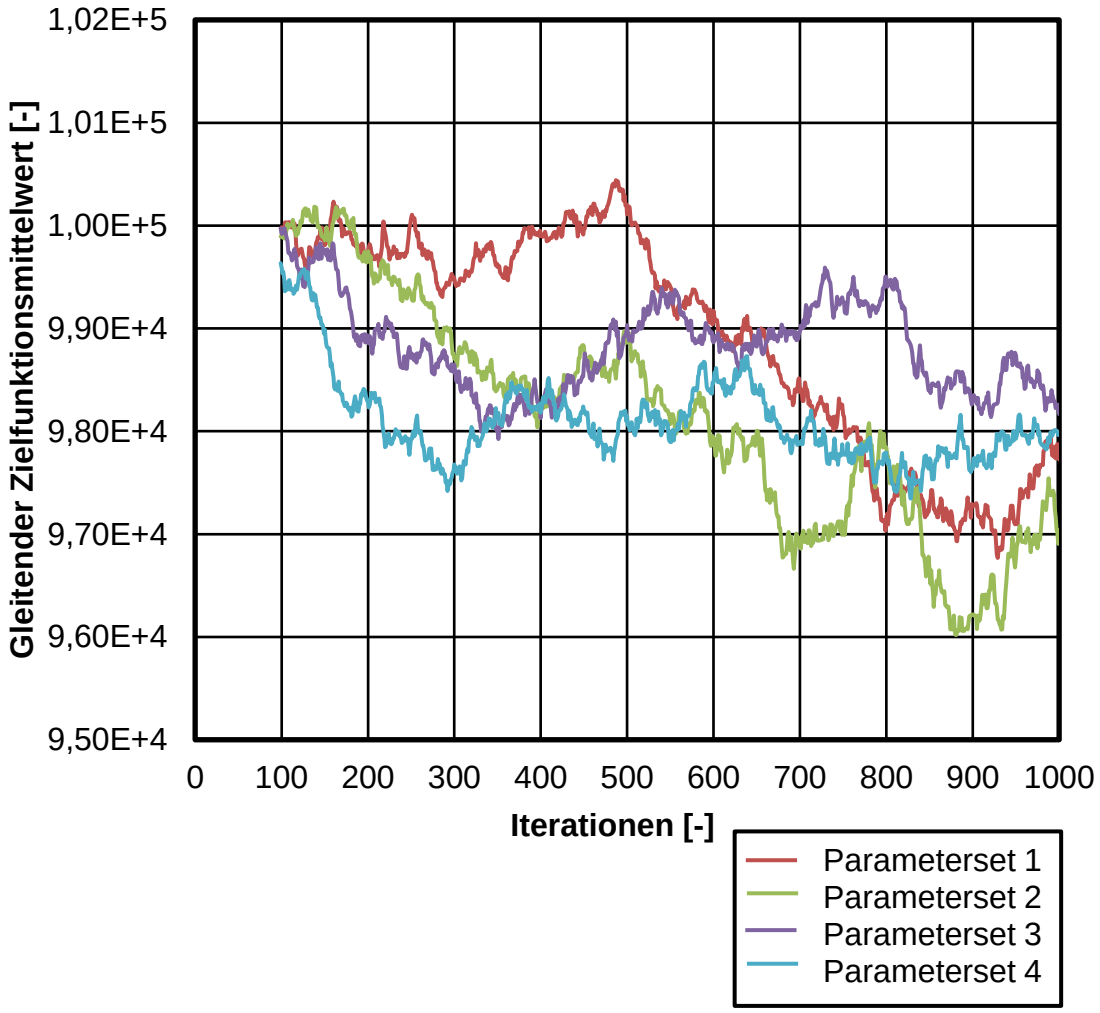

Abbildung 30: Gleitende Zielfunktionsmittelwerte

Aus Abbildung 30 ist ersichtlich, dass Parametersets mit niedriger Reproduktionsrate (Parameterset 3 und 4) insgesamt eine schlechtere Performance aufweisen, als Parametersets mit höherer Reproduktionsrate (Parameterset 1 und 2). Sie senken den Zielfunktionswert zwar schnell ab, verbleiben dann aber auf einem höheren Niveau. Parameterset 2 liefert die besten Ergebnisse. Dabei handelt es sich um ein Parameterset, das eine hohe Reproduktionsrate mit einer niedrigen Anzahl an Individuen pro Population kombiniert. Dieses Parameterset wird zur Optimierung der Simulationsszenarien in Kapitel 4 verwendet. Die Anzahl an Generationen wird auf 45 erhöht. In Tabelle 22 sind die verwendeten Parameter aufgeführt.

N [-]	r [%]	G [-]	R [-]
50	80	45	1810

Tabelle 22: Verwendete Parameter für genetischen Algorithmus

Es ergeben sich 1810 Simulationsdurchläufe um die Optimierung mittels genetischem Algorithmus durchzuführen. Für eine vollständige Optimierung sind somit insgesamt 8410 Simulationsdurchläufe notwendig, wobei angemerkt werden muss, dass die Simulationsdauer pro Durchlauf bei der Permutationsliste niedriger ist, als beim genetischen Algorithmus mit der Prioritätsliste. Dieses Verhalten ist der Tatsache geschuldet, dass beim genetischen Algorithmus die Prioritätsliste mit allen 362 880 möglichen Vektoren verwendet werden muss, während bei der Permutation die Permutationsliste mit einem Umfang von nur 6600 fest vorgegebenen Vektoren verwendet werden kann. Dadurch ist beim genetischen Algorithmus der Speicherbedarf wesentlich höher, was sich negativ auf die Geschwindigkeit auswirkt. Außerdem ist eine Ausführung der Simulation in 64-bit notwendig, um AMESIM den erforderlichen Arbeitsspeicherzugriff zu ermöglichen.

4 Experimente und Ergebnisse

In diesem Kapitel werden die Experimente am Simulationsmodell durchgeführt und die Ergebnisse ausgewertet. Es werden 24 Stunden Szenarien mit variablem Kältebedarf simuliert. Bei der Kälteanlage handelt es sich um eine bestehende und in Betrieb befindliche Anlage. Sie ist aus 12 Kältemaschinen aufgebaut. In den Kapiteln 4.1 bis 4.4 wird die Simulation und die zweistufige Optimierung mit allen 12 Maschinen demonstriert. Bei der Bewertung des Einsparungspotenzials (Kapitel 4.5) muss der gewichtete Modellfehler der Maschinen berücksichtigt werden. Aus diesem Grund werden in Kapitel 4.5 die 3 Maschinen mit dem höchsten Modellfehler entfernt. Dadurch reduziert sich der gewichtete Modellfehler. Die Optimierung wird mit den 9 verbleibenden Maschinen durchgeführt wodurch in Kapitel 4.5 der Schritt der Permutation entfällt.

4.1 Simulationsszenario und Parameter

In diesem Kapitel wird das Simulationsszenario für die Experimente am Simulationsmodell festgelegt. In Abbildung 31 wird der Aufbau der simulierten Kälteanlage dargestellt. 2 Maschinen besitzen ein Wärmerückgewinnungssystem (Maschine 1 und 5). Jeweils 6 Maschinen werden von einem Kühlnetz mit Kühlleistung versorgt, welches wiederrum die Wärme über jeweils ein Rückkühlwerk an die Umgebung abgibt. Alle Maschinen sind an ein Kältenetz angeschlossen, welches die Produktionsanlagen mit Kaltwasser versorgt.

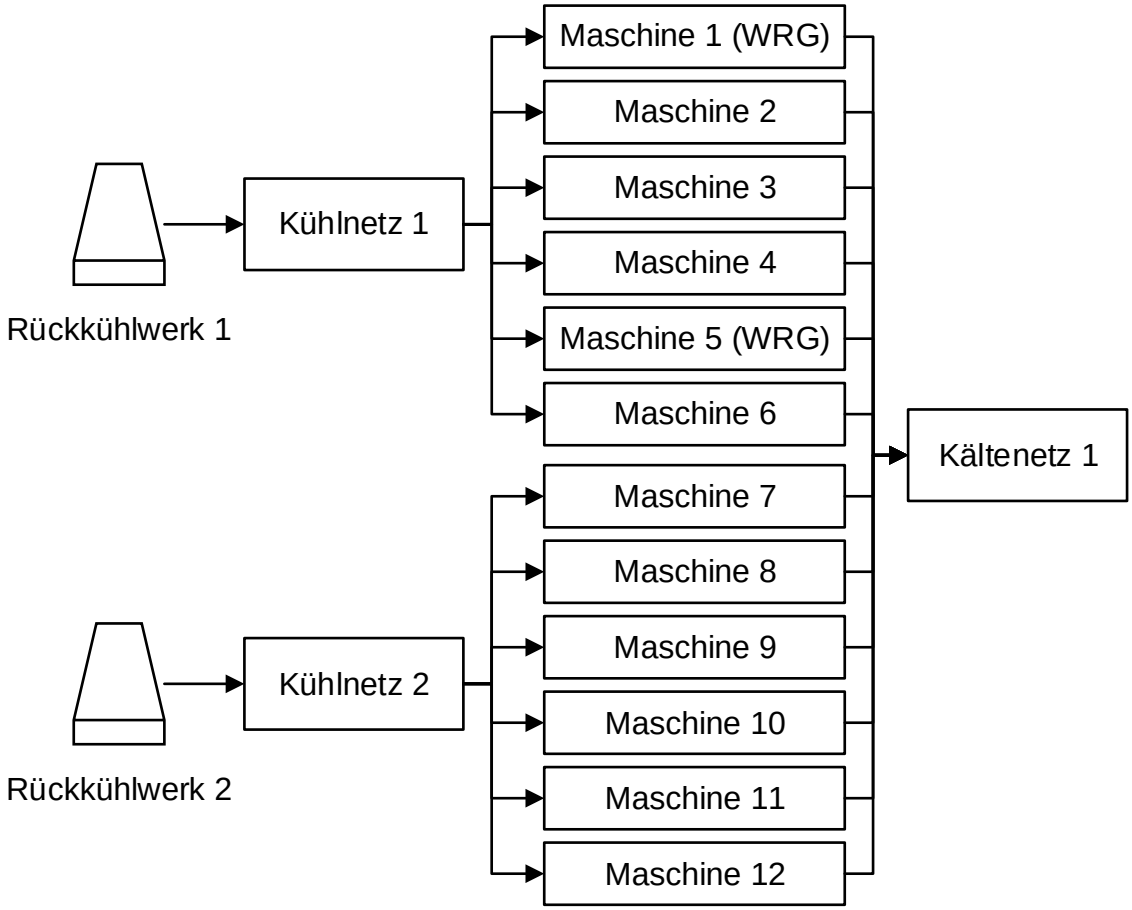

Abbildung 31: Aufbau der Kälteanlage

4.1.1 Eingangsgrößen

Die Eingangsgrößen für die Simulation sind der variable Kältebedarf in Form des negativen Wärmestroms $\dot{Q}_{k,b}$ und der Wärmebedarf in Form des positiven Wärmestroms $\dot{Q}_{wb,brutto}$, sowie die Netztemperaturen der Kreisläufe zu Beginn der Simulation und die Umgebungstemperatur. Der zeitliche Verlauf von $\dot{Q}_{k,b}$ über das 24 Stunden Szenario wird in Abbildung 32 dargestellt. Es ist eine ausgeprägte Lastspitze zwischen Stunde 14 und Stunde 22 zu erkennen.

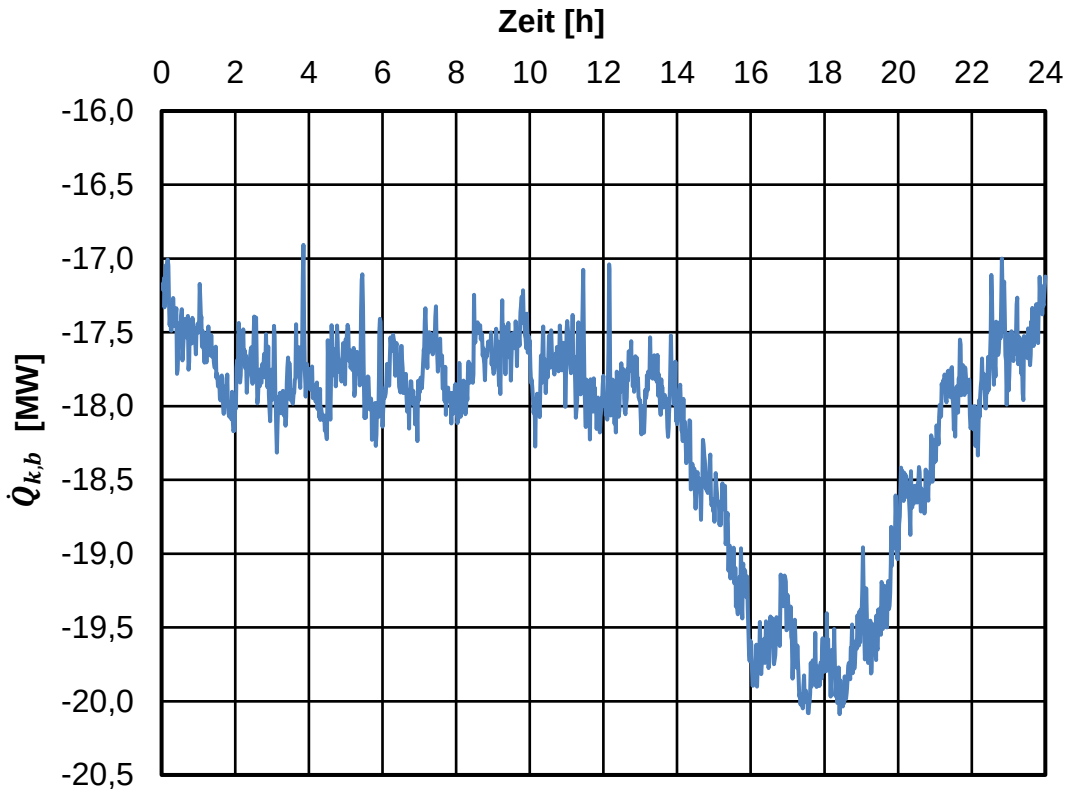

Abbildung 32: $\dot{Q}_{k,b}$ im 24h-Verlauf

Die Gesamtenergie, die in diesem Zeitraum in Form von Kälte angefordert wird, errechnet sich unter Verwendung von Gleichung (4.1). Der zeitliche Verlauf des negativen Wärmestroms $\dot{Q}_{k,b}$ wird nach der Zeit integriert um die Gesamtenergie des Zeitraums $Q_{k,b}$ zu erhalten. In der Simulation geschieht dies durch Verwendung eines Integrator-Bausteins.

$$Q_{k,b} = \int_0^t \dot{Q}_{k,b} dt \tag{4.1}$$

$$Q_{wb,brutto} = \int_0^t \dot{Q}_{wb,brutto} dt \tag{4.2}$$

Gleiches gilt für die gesamte im Simulationszeitraum benötigte Wärmeenergie $Q_{wb,brutto}$ (siehe Gleichung (4.2)). Der Wärmestrom $\dot{Q}_{wb,brutto}$ ist jedoch im Vergleich zu $\dot{Q}_{kb}$ im ganzen Simulationszeitraum konstant.

In Tabelle 23 sind der Kälteenergiebedarf $Q_{k,b}$ und der Wärmeenergiebedarf $Q_{wb,brutto}$, sowie die initialen Temperaturen $T_k$und T_c der Kühl- und Kältenetze aufgeführt. Die Referenztemperatur wird anhand der angenommenen maximalen Tagestemperatur von 30°C gewählt. Im Verlauf der Simulation werden die Temperaturen der Kühl- und Kaltwasserkreisläufe dynamisch vom Energienetzmodell berechnet. Die initialen Temperaturen sind gleichzeitig auch die Führungsgrößen für das Energienetzmodell, d.h. das Energienetz versucht diese Temperaturen unter der schwankenden Belastung (Abbildung 32) konstant zu halten (siehe Kapitel 3.2.2).

$Q_{k,b}$ [kWh]	$Q_{wb,brutto}$ [kWh]	T_k [°C]	T_c [°C]	T_0 [°C]
-436209	153504	7	25	30

Tabelle 23: Eingangsgrößen Simulationsszenario

Nachfolgend werden die verwendeten Parameter der einzelnen Komponenten beschreiben. Die Parameter der Kältemaschinen sind hingegen aus Platzgründen im Anhang (Kapitelt 6.1) zu finden.

4.1.2 Parameter der Energienetze

Der Kaltwasserkreislauf fasst 250 m^3 Wasser. Die Kühlwasserkreisläufe sind etwas kleiner und fassen 175 m^3 Wasser. Bei einer Temperatur von 20°C und einem Druck von 1 bar liegt die Dichte ρ des Wassers bei 998,21 kg/m^3 [38]. Somit liegt die Masse des zirkulierenden Wassers für den Kaltwasserkreislauf bei 249 553 kg und für den Kühlwasserkreislauf bei 174 687 kg. Für die spezifische Wärmekapazität des Wassers wird ebenfalls die Temperatur von 20°C und der Druck von 1 bar angenommen. Sie liegt bei 4,185 kJ/kgK [38]. Dieser Wert wird für alle Energienetze verwendet. Alle Parameter des Kältenetzes werden in Tabelle 24 aufgeführt.

Parameter	Bezeichnung	Einheit	Einheit
c	Spezifische Wärmekapazität	4,185	kJ/(kgK)
m	Masse	249553	kg
T_{Ziel}	Zieltemperatur	7	°C
T_0	Referenztemperatur	30	°C
K_p	Verstärkung P-Anteil	0,05	-
K_i	Verstärkung I-Anteil	$5*10^{-5}$	-

Tabelle 24: Parameter Kältenetz

Die Parameter der Kühlnetze sind in Tabelle 25 aufgeführt.

Parameter	Bezeichnung	Einheit	Einheit
c	Spezifische Wärmekapazität	4,185	kJ/(kgK)
m	Masse	174687	kg
T_{Ziel}	Zieltemperatur	25	°C
T_0	Referenztemperatur	30	°C
K_p	Verstärkung P-Anteil	0,05	-
K_i	Verstärkung I-Anteil	$5*10^{-5}$	-

Tabelle 25: Parameter Kühlnetz

4.1.3 Zielfunktionsparameter

Die Gewichtungsfaktoren werden so gewählt, dass die Kosten für Strom etwa 40%, die Kosten für Wärme auch etwa 40%, die Betriebskosten 10% und die gleichmäßige Auslastung 10% des Zielfunktionswertes ausmachen. Die gewählten Gewichtungsfaktoren werden in Tabelle 26 aufgeführt.

ω_1 [-]	ω_2 [-]	ω_3 [-]	ω_4 [-]	ω_5 [-]
0,2	0,2	100	1	500

Tabelle 26: Gewichtungsfaktoren der Zielfunktion

Weiters müssen in den Parametern des Zielfunktionsbausteins die Kostenfaktoren und die initialen Betriebsstunden der Maschinen festgelegt werden. In Tabelle 27 werden die verwendeten Werte aufgeführt. Außerdem ist ersichtlich, welche Maschinen über ein Wärmerückgewinnungssystem verfügen.

Maschine	**Kostenfaktor [-]**	**Initiale Betriebsstunden [h]**	**WRG**
1	0,89	95327	X
2	0,38	135497	
3	0,38	70368	
4	0,38	45932	
5	0,99	169784	X
6	0,38	38678	
7	0,38	6489	
8	0,38	13287	
9	0,38	10689	
10	0,38	12568	
11	0,38	9784	
12	0,38	4579	

Tabelle 27: Kostenfaktoren und initiale Betriebsstunden

Durch das Wärmerückgewinnungssystem sind die Betriebskostenfaktoren von Maschine 1 und 5 deutlich erhöht.

4.1.4 Weitere Parameter

Die Rückkühlwerke besitzen als einzigen Parameter die maximale Kühlleistung $\dot{Q}_{c,max}$. Es wird $\dot{Q}_{c,max}$ = 15 MW für das Szenario gewählt. Die Hysterese der Verteilfunktion wird auf $Hyst$ = 500 kW festgelegt.

4.2 Optimierungsergebnis

Die Optimierung wird in 2 Schritten ausgeführt: im ersten Schritt wird die Permutationsliste durchlaufen und die ersten 3 Einträge des finalen Prioritätsvektors ermittelt. Im zweiten Schritt wird der genetische Algorithmus angewendet, um die weiteren 9 Einträge zu ermitteln. Die 6600 Einträge der Permutationsliste sind nach 2:26 h Rechenzeit durchlaufen. Die Auswertung der Ergebnisse wird in Tabelle 28 dargestellt.

Maschine	Anzahl in ersten 100 Vektoren
1	47
5	45
9	36
7	32
11	30
6	27
4	25
10	24
8	14
12	13
2	5
3	2

Tabelle 28: Auswertung Permutationsliste

Die Maschinen 1, 5 und 9 erreichen die besten Werte und dienen für den genetischen Algorithmus als Vorgabewert. Der Bedarf an Wärme war im Simulationsszenario hoch, weshalb die beiden Maschinen mit Wärmerückgewinnung den Zielfunktionswert am weitesten gesenkt haben und entsprechend oft in den Vektoren mit niedrigstem Zielfunktionswert vorkommen.

Danach folgt die Bestimmung der restlichen Vektoren mittels genetischem Algorithmus. Nach 1:55 h Rechenzeit ist ein optimaler Prioritätsvektor $\vec{p}_{opt}$ gefunden (Gleichung (4.3)).

$$\vec{p}_{opt} = (1{,}5{,}9{,}7{,}10{,}6{,}4{,}11{,}12{,}2{,}3{,}8) \tag{4.3}$$

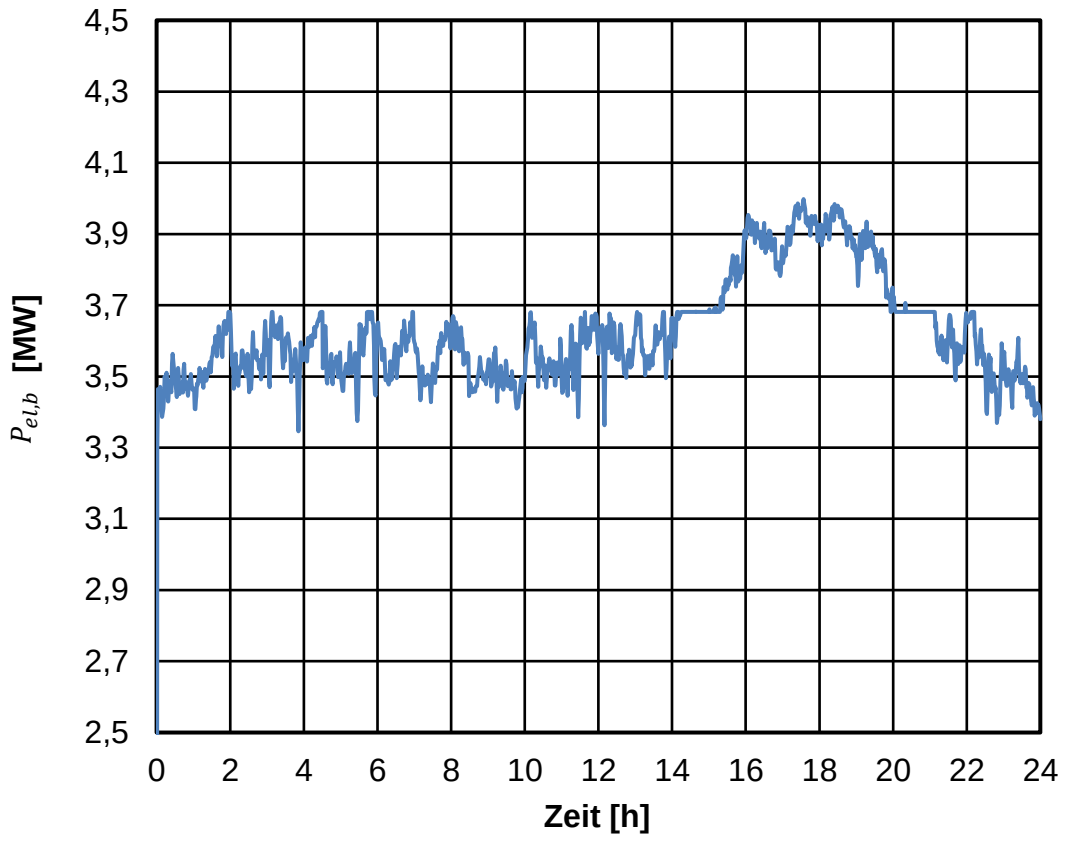

Abbildung 33: Elektrischer Leistungsbedarf $P_{el,b}$ der Kältemaschinen

Der elektrische Leistungsbedarf $P_{el,b}$ der Kältemaschinen im Simulationszeitraum ist in Abbildung 33 dargestellt. Es ist deutlich die erhöhte Leistungsaufnahme zur Lastspitze zwischen Stunde 14 und Stunde 20 zu erkennen. Im Simulationszeitraum werden 86 993 kWh elektrische Energie, sowie 44 588 kWh Wärmeenergie aus externen Quellen benötigt. Der Bedarf an Wärme kann nicht voll gedeckt werden, obwohl die Maschinen mit Wärmerückgewinnung unter Volllast betrieben werden und einen konstanten Wärmestrom von 4 538 kW liefern. Das entspricht über einen Zeitraum von 24 Stunden einer Wärmeenergie von 108 920 kWh. Die Summe aus elektrischem Energiebedarf und dem Bedarf an Wärme aus externen Quellen ergibt den Gesamtenergiebedarf von 131 581 kWh.

In Tabelle 29 werden der elektrische Energiebedarf, sowie der Bedarf an Wärmeenergie und der daraus resultierende Gesamtenergiebedarf dargestellt. Zum Vergleich ist der Messwert im selben Betrachtungszeitraum aufgeführt. In diesem Szenario liegt die Auslastung bei etwa 62% (siehe Kapitel 4.3). Bei höheren Auslastung sinkt das Optimierungspotenzial entsprechend.

	$E_{el,b}$ [kWh]	$Q_{wb,netto}$[kWh]	Gesamt [kWh]	Differenz [%]
Optimierung	86993	44588	131581	-23,54%
Messwert	98683	73405	172088	0

Tabelle 29: Einsparung durch optimierten Prioritätsvektor

Durch einen optimierten Prioritätsvektor können in diesem Szenario 40 507 kWh elektrische und thermische Energie eingespart werden. Das entspricht einer Einsparung von 23,54%. Zusätzlich werden die Maschinen so betrieben, dass sich die mittlere absolute Abweichung der Betriebsstunden der Maschinen reduziert. Die Gewichtungsfaktoren der Zielfunktion wurden so gewählt, dass alle Zielfunktionsbestandteile berücksichtigt werden. Wählt man diese Gewichtungsfaktoren anders, können die Einsparungen an Energie noch höher ausfallen. Allerdings müssen die Modellfehler der Kältemaschinen für die Betrachtung des Einsparungspotenzials berücksichtigt werden. Außerdem sind die Einsparungen auf die Messwerte bezogen. Diese Messwerte sind von der zum Aufnahmezeitpunkt der Messwerte angewendeten Betriebsstrategie abhängig. Eine detaillierte Bewertung des Einsparungspotenzials durch die Simulation wird in Kapitel 4.5 durchgeführt.

4.3 Betrachtung wichtiger Zeitverläufe

Neben der Optimierung lassen sich über die zeitlichen Verläufe wichtiger Variablen weitere Informationen über die Kälteanlage gewinnen. Für die Auslegung der Kälteanlage ist wichtig zu wissen, wie viel Kälteleistung noch erbracht werden kann, bevor die Rückkühlwerke überlastet sind oder die eingesetzten Kältemaschinen für die Deckung des Kältebedarfs nicht mehr ausreichen. Diese Fragestellungen werden nachfolgend erläutert.

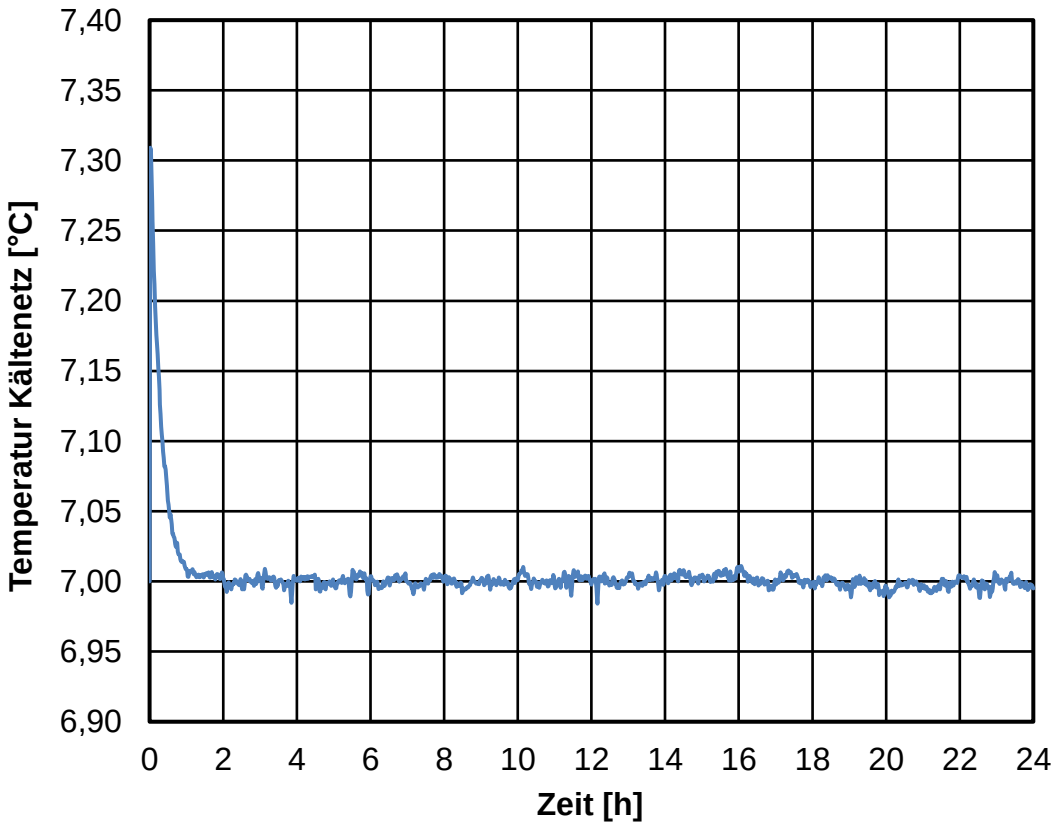

Abbildung 34: Temperaturverlauf Kältenetz

Der Temperaturverlauf im Kältenetz wird in Abbildung 34 dargestellt. Zu Beginn der Simulation ist ein Temperaturanstieg erkennbar, der dadurch hervorgerufen wird, dass am Anfang keine Kältemaschine in Betrieb ist und die Kältemaschinen erst nach und nach durch die Verteilfunktion eingeschaltet werden. Nach etwa einer Stunde ist die vorgegebene Netztemperatur erreicht und wird dann über die restliche Dauer des Szenarios konstant gehalten. Auch im Lastspitzenbereich zwischen Stunde 15 und Stunde 20 kann die Temperatur gehalten werden, d.h. es tritt keine Überlast der Kälteanlage auf. Eine weitere Möglichkeit die Auslastung der Kälteanlage zu ermitteln ist die Betrachtung der maximal möglichen Kälteleistung der Kältemaschinen im Zeitverlauf (siehe Abbildung 35).

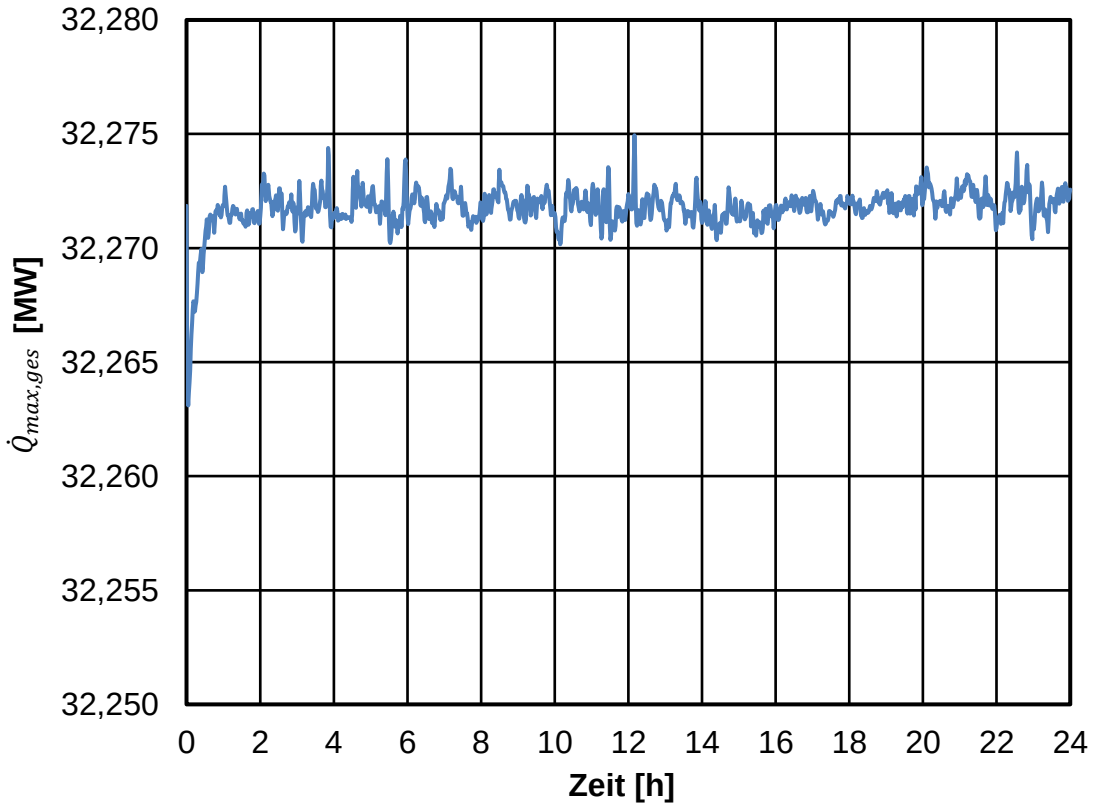

Abbildung 35: Zeitverlauf der Gesamtkältemaschinenkapazität $\dot{Q}_{max,ges}$

Die Gesamtkapazität der Kältemaschinen $\dot{Q}_{max,ges}$ liegt im Verlauf des Szenarios relativ konstant bei 32,27 MW. Die Lastspitze des Kältebedarfs (Abbildung 32) tritt nach 18,41 h auf und beträgt 20,08 MW. Die Auslastung der Kältemaschinen beträgt bei der Lastspitze ca. 62%.

Maschine	Betriebszeit [h]
1	24
2	0
3	0
4	7,148
5	23,999
6	23,992
7	23,997
8	0
9	23,998
10	23,995
11	0
12	0

Tabelle 30: Betriebszeiten der Maschinen

Bei Betrachtung der Betriebszeiten der Maschinen (siehe Tabelle 30) zeigt sich, dass etwa 18 Stunden lang 6 Maschinen für die Deckung des Bedarfs ausgereicht haben und nur im Lastspitzenbereich Maschine 4 als 7. Maschine zugeschaltet werden musste. Das gleiche Bild ergibt sich bei Betrachtung des zeitlichen Verlaufs von m_{on} in Abbildung 36.

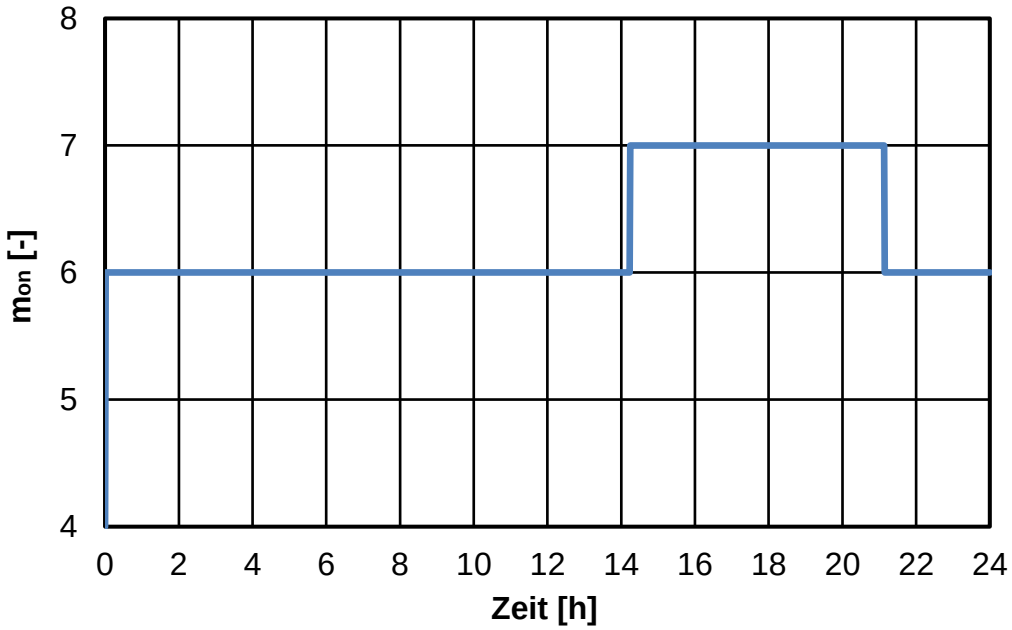

Abbildung 36: Zeitlicher Verlauf von m_{on}

m_{on} ist bis Stunde 14 gleich 6, d.h. die ersten 6 Maschinen auf dem Prioritätsvektor laufen. Zur Lastspitze wird die 7. Maschine zugeschaltet.

Die Kapazität der Kältemaschinen ist jedoch nicht der einzige limitierende Faktor. Die Belastung der Rückkühlwerke muss ebenfalls berücksichtigt werden. Dazu werden nachfolgend zuerst die Temperaturen der Kühlnetze betrachtet.

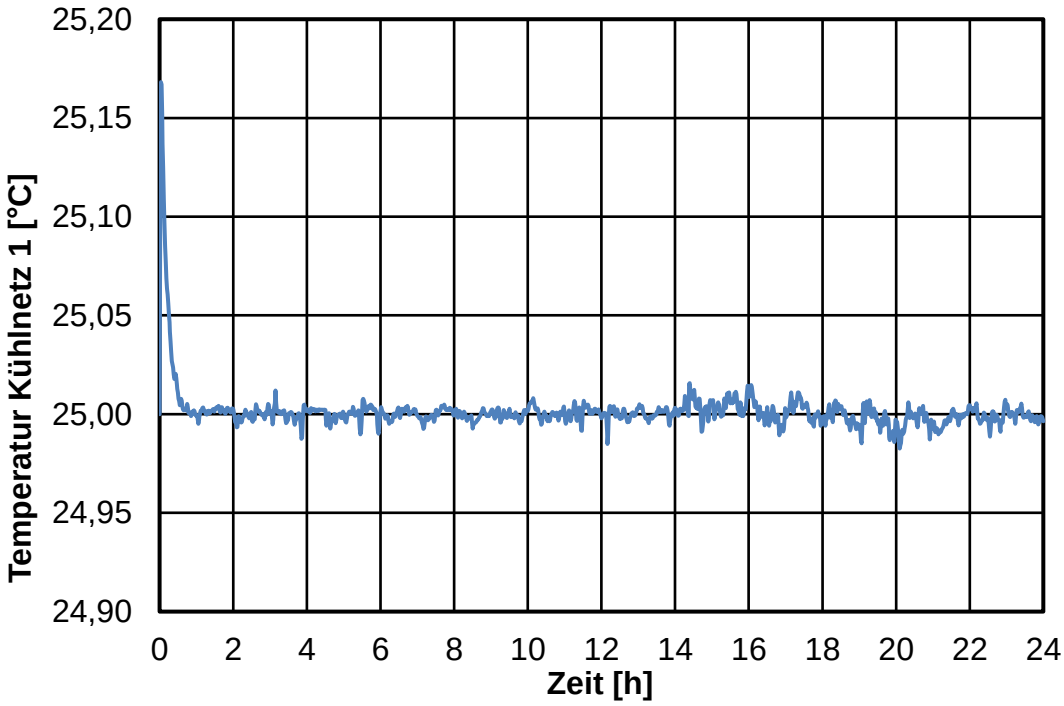

Abbildung 37: Temperaturverlauf Kühlnetz 1

Der Temperaturverlauf von Kühlnetz 1 wird in Abbildung 37 dargestellt. Im Gegensatz zu Kühlnetz 2 muss hier der schwankende Bedarf laufend durch die Regelung des Energienetzes angepasst werden, weil eine hier angeschlossene Maschine nicht im Vollbetrieb läuft. Aufgrund der Schaltlogik der Verteilfunktion laufen immer $m_{on} - 1$ Maschinen unter Volllast und eine Maschine im Teillastbereich (siehe Kapitel 3.2.3).

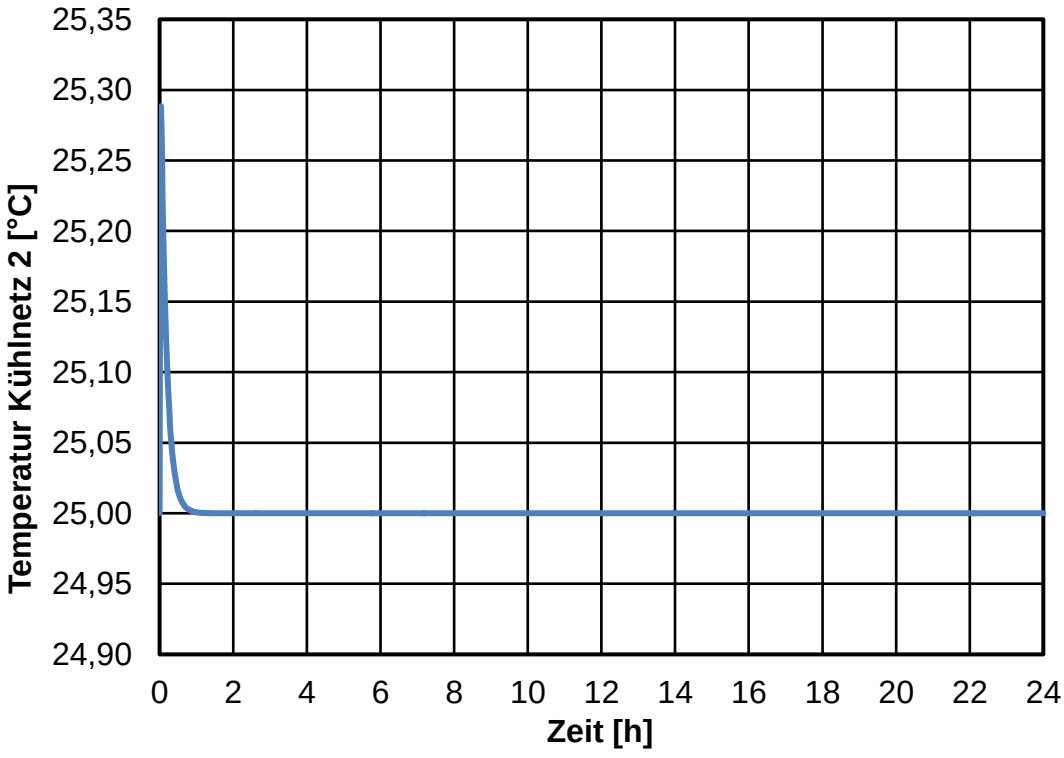

Abbildung 38: Temperaturverlauf Kühlnetz 2

Der Temperaturverlauf von Kühlnetz 2 wird in Abbildung 38 dargestellt. Auch hier ist wie im Kältenetz ein sprunghafter Temperaturanstieg am Beginn zu erkennen. Die Zieltemperatur von 25°C wird nach etwa einer Stunde erreicht und dann konstant gehalten. Die Belastung des Rückkühlwerks schwankt ab diesem Zeitpunkt nicht mehr, da die angeschlossenen Maschinen alle unter Volllast laufen und der Bedarf an Kühlung somit konstant ist.

In Abbildung 39 wird die von RKW 1 abgegebene Kühlleistung $\dot{Q}_{c,erz}$ dargestellt. Die Maximalkapazität von 15 MW wird deutlich unterschritten. Die Maximalbelastung liegt bei ca. 6 MW.

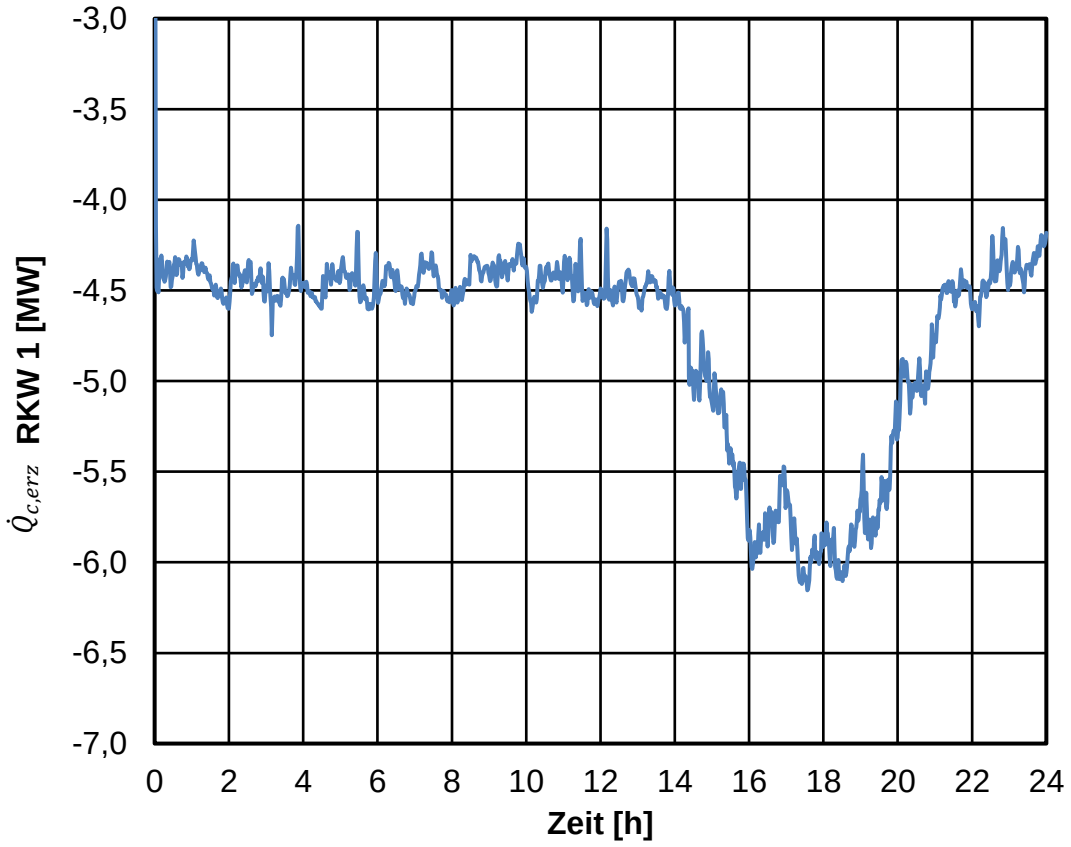

Abbildung 39: Von RKW 1 abgegebene Kühlleistung $\dot{Q}_{c,erz}$ im zeitlichen Verlauf

In Abbildung 40 wird die von RKW 2 abgegebene Kühlleistung dargestellt. Die Maximalkapazität von 15 MW wird ebenfalls unterschritten. Die Belastung ist aber insgesamt höher, als bei RKW 1. Wäre eine Überschreitung bei einem der 2 RKWs aufgetreten, hätten die RKW-Modelle $\dot{Q}_{c,pen}$ an die Zielfunktion weitergeleitet. Das hätte den Zielfunktionswert erhöht und der Optimierer hätte die Maschinen so betrieben, dass die Belastung gleichmäßig auf beide RKWs aufgeteilt wird.

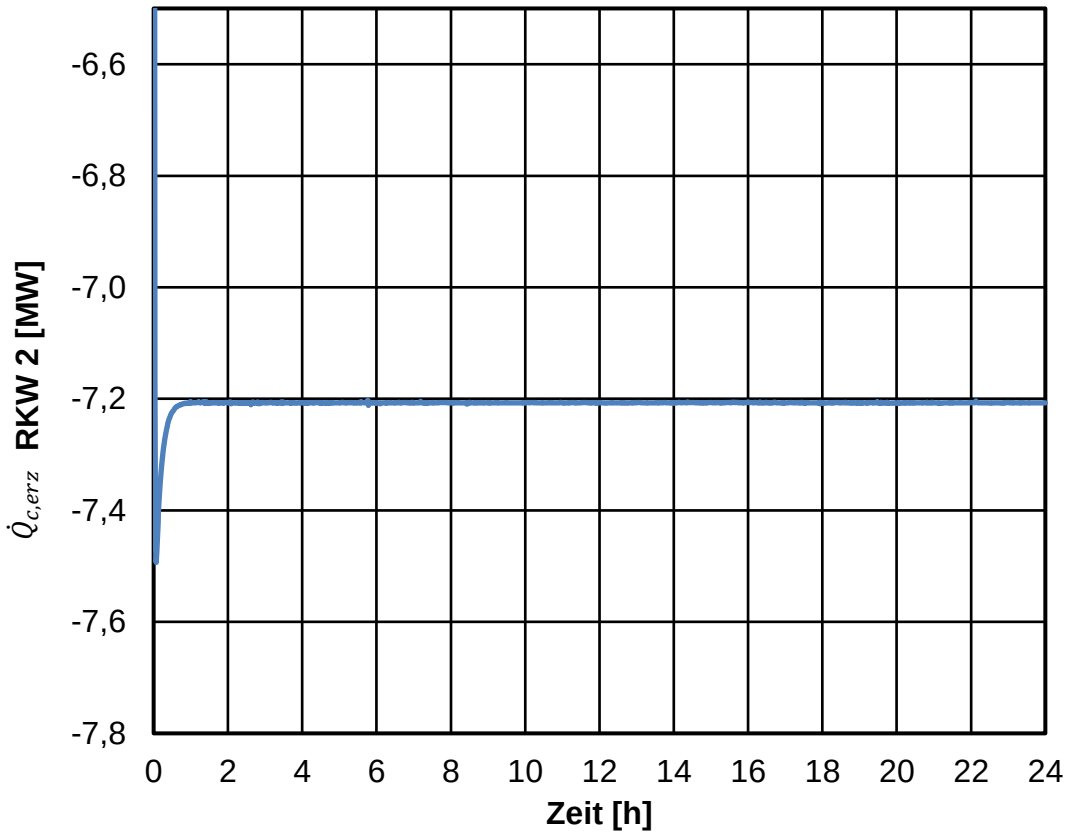

Abbildung 40: Von RKW 2 abgegebene Kühlleistung $\dot{Q}_{c,erz}$ im zeitlichen Verlauf

4.4 Weitere Betrachtungen

Anhand der Simulation können weitere Aussagen über das System getroffen werden. Beispielsweise kann simuliert werden, wie sich die Temperaturen in den Netzen ändern, wenn Maschinen ausfallen. Wenn nur 6 Maschinen für das in Kapitel 4.1.1 beschriebene Szenario zur Verfügung stehen und alle anderen zum Ausgleich der Lastspitze nicht verwendet werden, ergibt sich der in Abbildung 41 dargestellte Temperaturverlauf im Kältenetz. Bis Stunde 14 reicht die Kälteleistung der Maschinen noch aus, um den Bedarf zu decken. Dann steigt die Temperatur des Kältenetzes aufgrund der Unterversorgung mit Kälteleistung stark an und überschreitet nach 15 Stunden und 42 Minuten die 10°C Marke. Eine Kühlung der Produktionsanlagen ist nur noch eingeschränkt möglich.

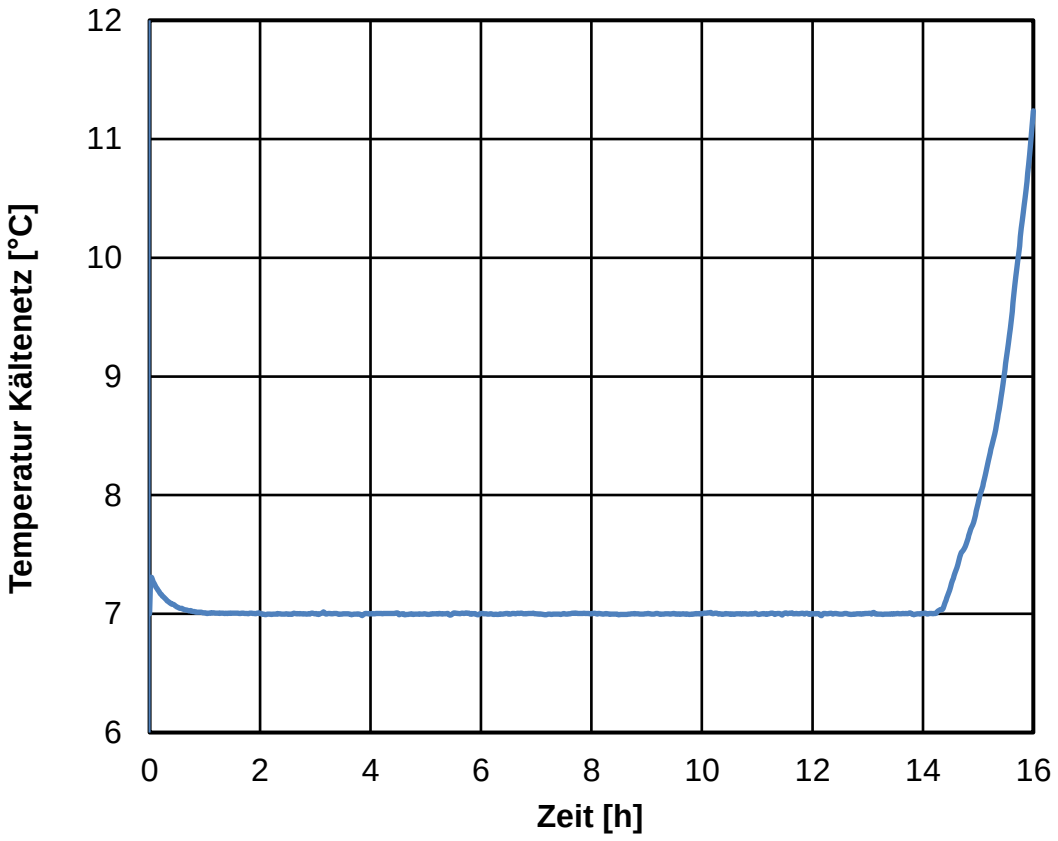

Abbildung 41: Temperaturverlauf Kältenetz mit nur 6 Kältemaschinen

4.5 Einsparungspotenzial

Das Einsparungspotenzial, das sich aufgrund der Optimierung ergibt, kann durch verschiedene Größen bewertet werden. In diesem Kapitel werden die Größen elektrischer Energiebedarf und Gesamtenergiebedarf betrachtet und eine Aussage über das Einsparungspotenzial bei diesen Größen getroffen. Für die Betrachtung des Einsparungspotenzials ist der gewichtete arithmetische Mittelwert des Modellfehlers zu berücksichtigen. Dieser errechnet sich aus dem $RMSE_{rel,i}$ der einzelnen Maschinen (siehe Kapitel 2.2.3, Tabelle 3) und der zugehörigen Referenzkälteleistung $\dot{Q}_{ref,i}$ nach Gleichung (4.4) [36].

$$\overline{RMSE}_{rel} = \frac{\sum_i \dot{Q}_{ref,i} RMSE_{rel,i}}{\sum_i \dot{Q}_{ref,i}} \tag{4.4}$$

Der mittlere Modellfehler aller Maschinen bei der Vorhersage der elektrischen Leistungsaufnahme liegt bei 59%. Dieser hohe Fehler ergibt sich aufgrund der Tatsache, dass 3 Maschinen einen Modellfehler von über 100% aufweisen. Aus diesem Grund werden jene 3 Maschinen (6, 7 und 12) für die Ausführungen in diesem Kapitel entfernt, wodurch sich der mittlere Modellfehler auf 17,3% reduziert. Durch die Reduktion auf 9 Maschinen entfällt die Simulation der Permutationsliste und die Optimierung wird nur mit dem genetischen Algorithmus durchgeführt. Wenn bei

niedriger Auslastung nicht alle Maschinen in einem Szenario in Betrieb sind, werden für die Bewertung des Modellfehlers nur die betriebenen Maschinen herangezogen.

Ein weiterer wichtiger Faktor, der das Optimierungspotenzial beeinflusst ist die Gesamtauslastung der Kälteanlage. Bei hoher Auslastung laufen beinahe alle Maschinen unter Volllast und die Einschaltreihenfolge hat keinen Einfluss mehr auf den Energieverbrauch. Bei niedriger Auslastung hat der Prioritätsvektor einen höheren Einfluss, da nicht alle Maschinen in Betrieb sind und die Einschaltreihenfolge stärker variiert werden kann. Deshalb werden für die Beurteilung des Optimierungspotenzials immer zwei Szenarien verwendet: ein Szenario mit hoher Auslastung und ein Szenario mit niedriger Auslastung. Die Auslastung wird nach Gleichung (4.5) bestimmt, wobei die Durchschnittswerte des Kältebedarfs $\bar{\dot{Q}}_{k,b}$ und der maximalen Maschinenkapazität $\bar{\dot{Q}}_{max,ges}$ im Simulationszeitraum verwendet werden.

$$Auslastung = \frac{\bar{\dot{Q}}_{k,b}}{\bar{\dot{Q}}_{max,ges}} * 100 \tag{4.5}$$

Die verwendeten Eingangsgrößen der beiden Szenarien werden in Tabelle 31 aufgeführt. Wie in Kapitel 4.1.1 erläutert handelt es sich um zeitabhängige Größen. Die in der Tabelle aufgeführten Werte werden durch Integration der zeitlichen Verläufe von $\dot{Q}_{k,b}$ und $\dot{Q}_{wb,brutto}$ nach der Zeit bestimmt.

$Q_{k,b}$ [kWh]	Auslastung [%]	$Q_{wb,brutto}$ [kWh]	T_k [°C]	T_c [°C]
-469969,4	84,4	61375	7,8	25,1
-200088,5	35,9	86722	7,2	24,3

Tabelle 31: Simulationsszenarien für die Bewertung des Einsparungspotenzials

Diese beiden Szenarien werden in Kapitel 4.5.1 auf minimalen elektrischen Energiebedarf und in Kapitel 4.5.2 auf minimalen Gesamtenergiebedarf optimiert. In Kapitel 4.5.3 werden die Ergebnisse zusammengefasst.

Der Verlauf von $\dot{Q}_{k,b}$ für das Szenario mit niedriger Auslastung wird in Abbildung 42 und der Verlauf von $\dot{Q}_{k,b}$ für das Szenario mit hoher Auslastung in Abbildung 43 dargestellt.

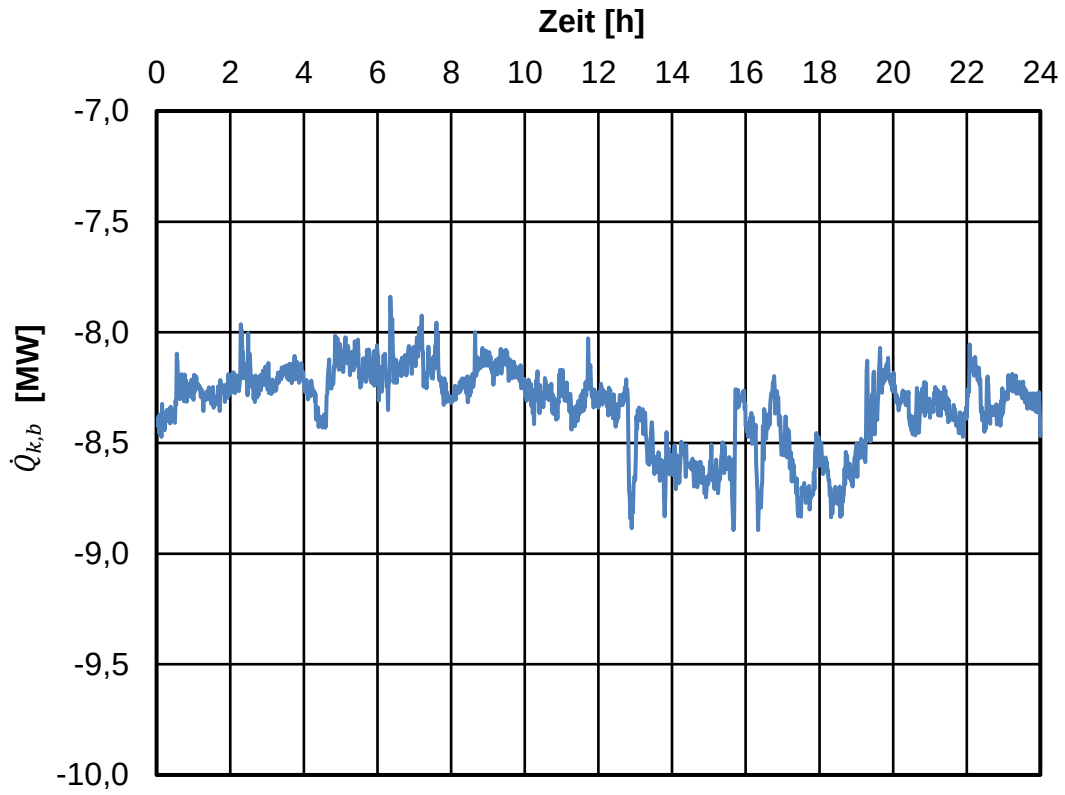

Abbildung 42: Verlauf von $\dot{Q}_{k,b}$ für niedrige Auslastung

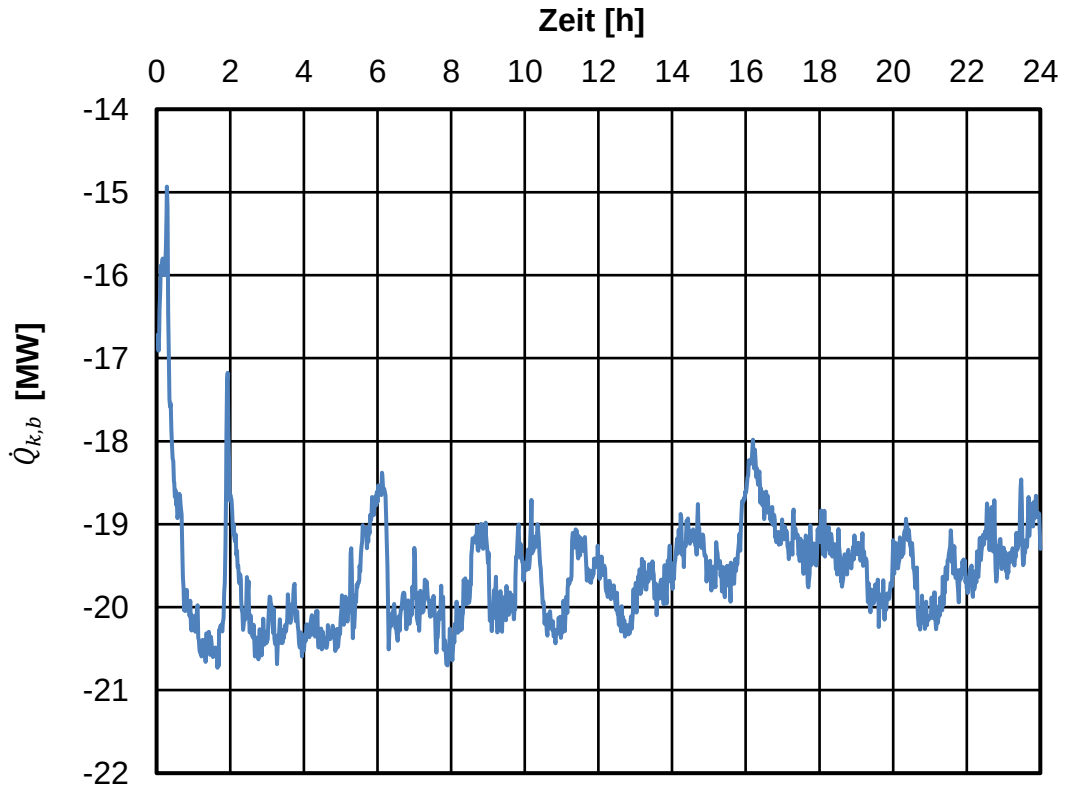

Abbildung 43: Verlauf von $\dot{Q}_{k,b}$ für hohe Auslastung

4.5.1 Elektrischer Energiebedarf

Zur Bewertung des Einsparungspotenzials bei der elektrischen Energie werden die Gewichtungsfaktoren aller Zielfunktionsbestandteile, außer dem für den elektrischen Energieverbrauch, 0 gesetzt (siehe Tabelle 32). Die Kostenfaktoren werden 1 gesetzt. Somit entspricht der Zielfunktionswert genau dem elektrischen Energieverbrauch. Mit dieser Konfiguration minimiert der Optimierer direkt den elektrischen Energieverbrauch. Das Wärmerückgewinnungssystem der Maschinen wird in dieser Konfiguration nicht berücksichtigt.

ω_1 [-]	ω_2 [-]	ω_3 [-]	ω_4 [-]	ω_5 [-]	k_P [-]	k_Q [-]
1	0	0	0	0	1	1

Tabelle 32: Gewichtungsfaktoren der Zielfunktion

Die Ergebnisse des Optimierungsprozesses werden in Tabelle 33 dargestellt. Der starke Zusammenhang zwischen Auslastung der Kälteanlage und Einsparungspotenzial ist klar erkennbar. Bei hoher Auslastung liegt das Einsparungspotenzial bei 14,87% gegenüber dem Messwert und bei niedriger Auslastung steigt das Einsparungspotenzial auf 34,96% an. Wird die Anlage in einem Auslastungsbereich zwischen diesen beiden Extremwerten betrieben bewegt sich die erzielbare Einsparung ebenfalls zwischen diesen Extremwerten.

$E_{el,b}$ Messwert [kWh]	$E_{el,b}$ optimiert [kWh]	Auslastung [%]	Einsparung [%]
104592	89038,2	84,4	14,87
53957,9	35093,7	35,9	34,96

Tabelle 33: Ergebnisse Optimierung elektrischer Energiebedarf

Die Ungenauigkeit des Kältemaschinenmodells bei der Vorhersage der elektrischen Leistungsaufnahme muss bei der Betrachtung des Einsparungspotenzials berücksichtigt werden. Beim Szenario mit minimaler Auslastung sind die Maschinen 10, 9, 11 und 4 gelaufen. Der Mittelwert des Modellfehlers liegt für die Maschinen 10, 9, 11 und 4 bei 13,45%. Nun wird dieser Fehler vom Einsparungspotenzial abgezogen. Nach Abzug bleibt ein Einsparungspotenzial von 21,5% beim Szenario mit niedriger Auslastung.

Beim Szenario mit hoher Auslastung sind alle 9 Maschinen in Betrieb. Es muss der gewichtete Durchschnitt des $\overline{RMSE}_{rel}$ von allen zur Bewertung des Einsparungspotenzials herangezogenen Maschinen gebildet werden (alle Maschinen außer 6, 7 und 12). Dieser liegt bei 17,27%. Der Fehler übersteigt in diesem Fall die Einsparungen von 14,87%. Für den Fall einer sehr hohen Auslastung können vermutlich keine Einsparungen erzielt werden.

4.5.2 Gesamtenergiebedarf

In diesem Kapitel wird das Einsparungspotenzial beim Gesamtenergiebedarf (elektrische Energie und Wärme) betrachtet. Hierfür werden die Gewichtungsfaktoren der Zielfunktion neu gewählt (siehe Tabelle 34). Die Zielfunktionsbestandteile für den elektrischen Energieverbrauch und den Bedarf an Wärmeenergie aus externen Quellen werden gleich hoch gewichtet. Der Zielfunktionswert entspricht in dieser Konfiguration dem Gesamtenergieverbrauch. Das Wärmerückgewinnungssystem der Maschinen wird im Zielfunktionswert berücksichtigt.

ω_1 [-]	ω_2 [-]	ω_3 [-]	ω_4 [-]	ω_5 [-]	k_P [-]	k_Q [-]
1	1	0	0	0	1	1

Tabelle 34: Gewichtungsfaktoren der Zielfunktion

Die Ergebnisse der Optimierung werden in Tabelle 35 aufgelistet. Das Einsparungspotenzial ist beim Gesamtenergiebedarf noch höher als bei reiner Betrachtung des elektrischen Energiebedarfs. Der Optimierer hat durch die zusätzlichen Möglichkeiten die Wärmerückgewinnung der Maschinen einzusetzen zusätzliche Freiheitsgrade und erzielt dadurch höhere Einsparungen.

$E_{el,b}$ + $Q_{wb,netto}$Messwert [kWh]	$E_{el,b}$ + $Q_{wb,netto}$ optimiert [kWh]	Auslastung [%]	Einsparung [%]
115672	90294,8	84,4	21,94
70351	41472,6	35,9	41,05

Tabelle 35: Ergebnisse Optimierung Gesamtenergiebedarf

Auch hier muss zur Bewertung des endgültigen Einsparungspotenzials der gewichtete mittlere Fehler $\overline{RMSE}_{rel}$ betrachtet werden. Im Szenario mit niedriger Auslastung sind die Maschinen 1, 4, 5 und 10 gelaufen. Der mittlere gewichtete Fehler für diese Maschinen liegt bei 22,15%. Wird dieser Fehler von der Einsparung abgezogen bleibt ein Einsparungspotenzial von 18,9%.

Beim Szenario mit hoher Auslastung sind alle Maschinen in Betrieb. Der mittlere Fehler liegt bei 17,27%. Es bleibt ein Einsparungspotenzial von 4,67%. Anders als bei der Optimierung auf den elektrischen Energieverbrauch, können bei der Optimierung auf den Gesamtenergieverbrauch auch beim Szenario mit hoher Auslastung Einsparungen erzielt werden.

4.5.3 Zusammenfassung der untersuchten Szenarien

In Tabelle 36 werden die untersuchten Szenarien noch einmal übersichtlich dargestellt. In der letzten Spalte sind die erzielbaren Einsparungen nach Abzug des gewichteten mittleren Modellfehlers aufgeführt. Für 3 von 4 Szenarien können Einsparungen erzielt werden. Bei der Optimierung auf minimalen elektrischen Energiebedarf mit hoher Auslastung (Szenario 1) können vermutlich keine Einsparungen realisiert werden.

Szenario	**Optimiert**	**Auslastung [%]**	$\overline{RMSE_{rel}}$ **[%]**	**Einsparung [%]**
1	$E_{el,b}$	84,4	17,27	-
2	$E_{el,b}$	35,9	13,45	21,15
3	$E_{el,b}$ + $Q_{wb,netto}$	84,4	17,27	4,67
4	$E_{el,b}$ + $Q_{wb,netto}$	35,9	22,15	18,9

Tabelle 36: Szenarien und Einsparungen abzüglich des mittleren Fehlers

5 Zusammenfassung und Ausblick

In dieser Arbeit wird demonstriert, wie die Optimierung einer bestehenden Kälteanlage durchgeführt werden kann. Die Ergebnisse zeigen, dass durch eine Optimierung der Einschaltreihenfolge der Maschinen, Einsparungen beim Energiebedarf bestehender Kälteanlagen erzielt werden können. Außerdem wird beschrieben, wie die Implementierung des Simulationsmodells und des Optimierers in AMESIM methodisch durchführbar ist. Es wird gezeigt, dass die Modellbildung auf Grundlage von Cubes eine gute Möglichkeit darstellt, eine Kälteanlage in einem Simulationsmodell abzubilden. Eine Anpassung des Simulationsmodells an andere Anlagenkonfigurationen ist aufgrund der Cube-Struktur des Modells einfach möglich.

Die in der Literatur erzielten Modellgenauigkeiten können bei der Parametrisierung teilweise nicht erreicht werden. Die Datenerhebung wird an im Einsatz befindlichen Maschinen durchgeführt und nicht unter Laborbedingungen. Trotzdem haben die angewendeten Methoden der Parametrisierung gute Ergebnisse geliefert, allerdings ist die Parametrisierung bei 3 Maschinen aufgrund des großen Fehlers bei der Vorhersage des elektrischen Energiebedarfs fehlgeschlagen. Die restlichen Maschinen bewegen sich in einem vertretbaren Bereich. 6 von 12 Maschinen weisen bei der Vorhersage des elektrischen Leistungsbedarfs einen kleineren Fehler als 20% auf. Bei 3 weiteren Maschinen liegt der Fehler im Bereich von 20% bis 40%.

Die Implementierung datenbasierter Modelle in AMESIM stellt eine Herausforderung dar, da vorgefertigte Bausteine fehlen. Aus diesem Grund ist es notwendig, auf benutzerdefiniert Modelle zurückzugreifen. AMESIM bietet hierfür ein Tool, mit dem sich benutzerdefinierte Modelle in die Simulation einbauen lassen. Durch eine Kombination aus vorgefertigten Bausteinen und benutzerdefinierten Modellen lässt sich die Simulation implementieren. AMESIM bietet durch die einfache Darstellung von zeitlichen Verläufen einzelner Variablen gute Möglichkeiten zusätzliche Informationen über das Systemverhalten zu gewinnen.

Die Implementierung der Optimierung hat gezeigt, dass der genetische Algorithmus, den AMSIM zur Verfügung stellt, nicht optimal für diesen Anwendungsfall geeignet ist. Es fehlen Möglichkeiten komplexe Randbedingungen festzulegen. Dieses Problem wird durch die Vorgabe von gültigen Prioritätsvektoren über eine Tabelle gelöst. So können die beschränkten Einstellungsmöglichkeiten umgangen und die Optimierung implementiert werden. Allerdings ist ein zweiter Schritt notwendig, der der Optimierung durch den genetischen Algorithmus vorangestellt wird. Dieser Schritt besteht aus der Simulation einer Permutationsliste. Dadurch reduziert sich der Suchraum für den genetischen Algorithmus von 12! auf 9! Vektoren. Diese Anzahl an möglichen Prioritätsvektoren kann über eine Tabelle vorgegeben werden. Nach der Implementierung liefert die zweistufige Optimierung gute Ergebnisse.

Für die Bewertung des Einsparungspotenzials wird der Modellfehler in die Betrachtung miteinbezogen. Aufgrund des hohen Modellfehlers von 3 Maschinen werden bei der Betrachtung des Einsparungspotenzials nur 9 Maschinen berücksichtigt. Das erzielbare Einsparungspotenzial hängt von der Auslastung der Anlage ab und schwankt bei den untersuchten Szenarien zwischen 5% und 21%. Die Einsparungen liegen somit in Bereichen, die von anderen Autoren mit vergleichbaren Optimierungsansätzen ebenfalls erreicht werden. Bei einem von vier Szenarien können vermutlich keine Einsparungen erzielt werden.

Ein Schwerpunkt sollte in Zukunft auf die Parametrisierung der Modelle gelegt werden. Wenn die Modellgenauigkeiten noch gesteigert werden können, ist die Freilegung von weiterem Optimierungspotenzial möglich. Die verwendete Software AMESIM kann nicht alle Anforderungen zur Genüge erfüllen. Für diesen Anwendungsfall ist AMESIM aufgrund der beschriebenen Implementierungshürden nicht die beste Wahl. Für die reine Simulation bietet AMESIM zwar alle benötigten Funktionalitäten, die Optimierungstools können aber nicht alle Anforderungen erfüllen. Eine Auslagerung des Optimierers wäre eine Möglichkeit hier Abhilfe zu schaffen. Dann kann der Schritt der Permutation entfernt und die Optimierung über den genetischen Algorithmus alleine durchgeführt werden. Dadurch würde sich die benötigte Rechenzeit erheblich verkürzen.

6 Anhang

6.1 Kältemaschinenparameter

Kältemaschine 1	
a_0	0,16192893
a_1	1,08055901
a_2	0,08887312
b_0	1
b_1	0
b_2	0
b_3	0
b_4	0
b_5	0
c_0	1
c_1	0
c_2	0
c_3	0
c_4	0
c_5	0
η_0	0,502349099
η_1	1,01710955
$\dot{Q}_{ref}$	1979,99033361
EER_0	4,32970712

Kältemaschine 2	
a_0	1,48496083
a_1	-2,91800842
a_2	2,70025178
b_0	1
b_1	0
b_2	0
b_3	0
b_4	0
b_5	0
c_0	1
c_1	0
c_2	0
c_3	0
c_4	0
c_5	0
η_0	0,60703117
$\dot{Q}_{ref}$	3258,18620041
EER_0	6,44004665

Kältemaschine 3	
a_0	0,01465603
a_1	1,00695082
a_2	-0,00293085
b_0	1
b_1	0
b_2	0
b_3	0
b_4	0
b_5	0
c_0	1
c_1	0
c_2	0
c_3	0
c_4	0
c_5	0
η_0	1,17002104
$\dot{Q}_{ref}$	2999,6951291
EER_0	5,69556465

Kältemaschine 4	
a_0	-0,29461871
a_1	1,38229433
a_2	-0,00812307
b_0	1
b_1	0
b_2	0
b_3	0
b_4	0
b_5	0
c_0	1
c_1	0
c_2	0
c_3	0
c_4	0
c_5	0
η_0	0,96121603
$\dot{Q}_{ref}$	2690,09360917
EER_0	5,96421417

Kältemaschine 5	
a_0	1,48496083
a_1	-2,91800842
a_2	2,70025178
b_0	1
b_1	0
b_2	0
b_3	0
b_4	0
b_5	0
c_0	1
c_1	0
c_2	0
c_3	0
c_4	0
c_5	0
η_0	0,60703117
η_1	0,79204726
$\dot{Q}_{ref}$	3119,94065072
EER_0	6,44004665

Kältemaschine 6	
a_0	1,48496083
a_1	-2,91800842
a_2	2,70025178
b_0	1
b_1	0
b_2	0
b_3	0
b_4	0
b_5	0
c_0	1
c_1	0
c_2	0
c_3	0
c_4	0
c_5	0
η_0	0,60703117
$\dot{Q}_{ref}$	4897,22779906
EER_0	6,44004665

Kältemaschine 7	
a_0	-0,29461871
a_1	1,38229433
a_2	-0,00812307
b_0	1
b_1	0
b_2	0
b_3	0
b_4	0
b_5	0
c_0	1
c_1	0
c_2	0
c_3	0
c_4	0
c_5	0
η_0	0,96121603
$\dot{Q}_{ref}$	2827,83186358
EER_0	5,96421417

Kältemaschine 8	
a_0	0,87562579
a_1	0,18809551
a_2	0,01791803
b_0	-4,39856083
b_1	0,85375095
b_2	0,24234091
b_3	-0,02780042
b_4	-0,02100584
b_5	-0,00178190
c_0	1,06743755
c_1	-0,11117324
c_2	0,02627123
c_3	0,00065935
c_4	0,00458971
c_5	-0,00152463
η_0	0,4864
$\dot{Q}_{ref}$	1893,5
EER_0	4,333

Kältemaschine 9	
a_0	0,24402461
a_1	0,63642630
a_2	0,12927342
b_0	3,43048333
b_1	-0,13017944
b_2	-0,16908284
b_3	0,00145813
b_4	0,00525377
b_5	0,00281831
c_0	0,56153048
c_1	-0,02495248
c_2	0,03765596
c_3	0,00183740
c_4	-0,00072339
c_5	-0,00057294
η_0	1,14818894
$\dot{Q}_{ref}$	2870,21449467
EER_0	5,66786801

Kältemaschine 10	
a_0	-0,00427420
a_1	1,00462277
a_2	-0,00220303
b_0	2,38480336
b_1	-0,04140645
b_2	-0,11528039
b_3	0,00075129
b_4	0,00136655
b_5	0,00251421
c_0	0,97479732
c_1	-0,01307403
c_2	0,00366460
c_3	-0,00003881
c_4	0,00057429
c_5	-0,00030103
η_0	1,14975665
$\dot{Q}_{ref}$	2980,18539924
EER_0	5,76479102

Kältemaschine 11	
a_0	-0,13559233
a_1	1,14567154
a_2	-0,00846885
b_0	1,79190957
b_1	0,01394175
b_2	-0,05088924
b_3	-0,00070986
b_4	-0,00019001
b_5	0,00131045
c_0	1,09209299
c_1	-0,02104849
c_2	-0,00335911
c_3	0,00040356
c_4	0,00065058
c_5	-0,00018636
η_0	0,83417309
$\dot{Q}_{ref}$	2581,19110310
EER_0	7,12373688

Kältemaschine 12	
a_0	0,87562579
a_1	0,18809551
a_2	0,01791803
b_0	-4,39856083
b_1	0,85375095
b_2	0,24234091
b_3	-0,02780042
b_4	-0,02100584
b_5	-0,00178190
c_0	1,06743755
c_1	-0,11117324
c_2	0,02627123
c_3	0,00065935
c_4	0,00458971
c_5	-0,00152463
η_0	0,4864
$\dot{Q}_{ref}$	1613,20725358
EER_0	4,333

6.2 Quellcode Kalkulationsfunktionen

6.2.1 Kältemaschinenmodell

```
void kaelte_5port_012_(int *n, double *Pelb, double *Qcb, double *Cap
      , double *Qwrg, double *Qk, double *Pel, double *Qc, double *Tcr
      , double *Qkb, double *Tkr, double *EERFT, double *Pelmin
      , double *PLR, double *EERFPLR, double *Qkbtemp, double *EER
      , double *Pelba, double rp[19], int ip[1])
{
   int loop;
   int WRG;
   double Cap0, n0, n1, EER0, a0, a1, a2, b0, b1, b2, b3, b4, b5, d0,
      d1, d2, d3, d4, d5;
   WRG         = ip[0];
   Cap0        = rp[0];
   n0          = rp[1];
   n1          = rp[2];
   EER0        = rp[3];
   a0          = rp[4];
   a1          = rp[5];
   a2          = rp[6];
   b0          = rp[7];
   b1          = rp[8];
   b2          = rp[9];
   b3          = rp[10];
   b4          = rp[11];
   b5          = rp[12];
   d0          = rp[13];
   d1          = rp[14];
   d2          = rp[15];
   d3          = rp[16];
   d4          = rp[17];
   d5          = rp[18];
   loop = 0;

*Cap = Cap0*(d0+d1**Tkr+d2**Tcr+d3**Tkr**Tkr+d4**Tkr**Tcr+d5**Tcr**Tcr);
*EERFT = b0+b1**Tkr+b2**Tcr+b3**Tkr**Tkr+b4**Tkr**Tcr+b5**Tcr**Tcr;

if (*Cap < sqrt(*Qkb**Qkb))
   {
      *Qkbtemp=-*Cap;
   }
   else
   {
      *Qkbtemp=-sqrt(*Qkb**Qkb);
   }

*PLR = (sqrt(*Qkbtemp**Qkbtemp))/(*Cap);
*EERFPLR = a0+a1**PLR+a2**PLR**PLR;
*EER = *EERFT**EERFPLR;

if (*Qkb < 0)
   {
   *Pelba = (*Cap**EER)/(EER0);
   if (*Pelba > 0)
      {
      *Pelb = *Pelba;
      }
      else
      {
      *Pelb = 0;
      }
   }
```

```
61      else
62      {
63      *Pelb = 0;
64      }
65
66  *Pelmin=((*Cap)/(EER0))*(*EERFT*a0);
67
68  if (*Pel > *Pelb)
69      {
70          *Qk = *Qkbtemp;
71      }
72      else
73      {
74          *Qk = 0;
75      }
76
77  *Qcb = (n0*(*Qk+*Pelb));
78
79  if (WRG == 1)
80      {
81          *Qwrg = n1**Qk;
82      }
83      else
84      {
85          *Qwrg = 0;
86      }
87  if (*Pel == 0)
88      {
89          *Qk = 0;
90          *Pelb = 0;
91          *Qcb = 0;
92          *Cap = 0;
93          *Qwrg = 0;
94      }
95  }
```

6.2.2 Energienetz-Modell

```
void enetz_mit_speicher_010_(int *n, double *Qerzb, double *Temperatur
      , double *Pbez, double *Qerz, double *Pbezb, double *Energy
      , double *dEnergy, double *h, double *h1, double *regeldiff
      , double *dregeldiff, double *Capacity, double *Esoll
      , double rp[9], double *t)
{
   int loop;
   double cp, m, T0, Zieltemp, piregler, Hyst, Ph, Kp, Ki;
   cp          = rp[0];
   m           = rp[1];
   T0          = rp[2];
   Zieltemp    = rp[3];
   piregler    = rp[4];
   Hyst        = rp[5];
   Ph          = rp[6];
   Kp          = rp[7];
   Ki          = rp[8];
   loop = 0;
*Pbez = *Pbezb;
*Capacity = m * cp;
*Esoll = (Zieltemp - T0) * *Capacity;
if (*t == 0)
   {
      *Energy = *Esoll;
   }
/*Regelung Anfang*/
if (piregler == 0)
   {
      if ((*Esoll-*Energy)>Hyst)
         {
            *h=0;
         }
         else if ((*Esoll-*Energy)<-Hyst)
         {
            *h=1;
         }
         else
         {
            *h1=0;
         }
      if (*h==1)
         {
            *Qerzb = Ph;
         }
         else
         {
            *Qerzb = 0;
         }
   }
  else
   {
      *dregeldiff = *Esoll-*Energy;
      *Qerzb = Kp*(*Esoll-*Energy) + Ki**regeldiff;
   }
/*Regelung Ende*/
*dEnergy = *Qerz - *Pbez;
*Temperatur = (*Energy / *Capacity) + T0;
}
```

6.2.3 Verteilfunktion

```
void verteilfkt_12m_019_(int *n, double *p01, double *p02, double *p03
      , double *p04, double *p05, double *p06, double *p07
      , double *p08, double *p09, double *p10, double *p11
      , double *p12, double *f02, double *f03, double *f04
      , double *f05, double *f06, double *f07, double *f08
      , double *f09, double *Qwrg, double *B01, double *dB01
      , double *B02, double *dB02, double *B03, double *dB03
      , double *B04, double *dB04, double *B05, double *dB05
      , double *B06, double *dB06, double *B07, double *dB07
      , double *B08, double *dB08, double *B09, double *dB09
      , double *B10, double *dB10, double *B11, double *dB11
      , double *B12, double *dB12, double *Pelbges, double *aQkb
      , double *Temp, double *Pelb01, double *Qcb01, double *Cap01
      , double *Qwrg01, double *Pelb02, double *Qcb02, double *Cap02
      , double *Qwrg02, double *Pelb03, double *Qcb03, double *Cap03
      , double *Qwrg03, double *Pelb04, double *Qcb04, double *Cap04
      , double *Qwrg04, double *Pelb05, double *Qcb05, double *Cap05
      , double *Qwrg05, double *Pelb06, double *Qcb06, double *Cap06
      , double *Qwrg06, double *Pelb07, double *Qcb07, double *Cap07
      , double *Qwrg07, double *Pelb08, double *Qcb08, double *Cap08
      , double *Qwrg08, double *Pelb09, double *Qcb09, double *Cap09
      , double *Qwrg09, double *Pelb10, double *Qcb10, double *Cap10
      , double *Qwrg10, double *Pelb11, double *Qcb11, double *Cap11
      , double *Qwrg11, double *Pelb12, double *Qcb12, double *Cap12
      , double *Qwrg12, double *Qkb12, double *Qkb11, double *Qkb10
      , double *Qkb09, double *Qkb08, double *Qkb07, double *Qkb06
      , double *Qkb05, double *Qkb04, double *Qkb03, double *Qkb02
      , double *Qkb01, double *mon, double *Qkbges, double *abw
      , double *aQkb01, double *aQkb02, double *aQkb03, double *aQkb04
      , double *aQkb05, double *aQkb06, double *aQkb07, double *aQkb08
      , double *aQkb09, double *aQkb10, double *aQkb11, double *aQkb12
      , double *Qkb, double rp[1])
{
   int loop;
   int error;
   int x;
   double Hyst;
   Hyst       = rp[0];
   loop = 0;
/*Fehlerbehandlung*/
error = 0;
if (*p01 <= 0 || *p02 <= 0 || *p03 <= 0 || *p04 <= 0 || *p05 <= 0 || *p06 <= 0
    || *p07 <= 0 || *p08 <= 0 || *p09 <= 0 || *p10 <= 0 || *p11 <= 0 || *p12 <=
    0)
{
amefprintf(stderr, "ACHTUNG: Prioritaetszahl groesser 0 eingegeben!");
error = 2;
}
if (*p01 >= 13 || *p02 >= 13 || *p03 >= 13 || *p04 >= 13 || *p05 >= 13 || *p06
    >= 13 || *p07 >= 13 || *p08 >= 13 || *p09 >= 13 || *p10 >= 13 || *p11 >= 13
    || *p12 >= 13)
{
amefprintf(stderr, "ACHTUNG: Prioritaetszahl kleiner 13 eingegeben!");
error = 2;
}
if (*p01 + *p02 + *p03 + *p04 + *p05 + *p06 + *p07 + *p08 + *p09 + *p10 + *p11
    +*p12 != 78)
{
amefprintf(stderr, "ACHTUNG: Falsche Prioritaetszuordnung (falsche Gesamtsumme)
    !");
```

```
55 error = 2;
56 }
57 if (*p01 == *p02 || *p01 == *p03 || *p01 == *p04 || *p01 == *p05 || *p01 == *
       p06 || *p01 == *p07 || *p01 == *p08 || *p01 == *p09 || *p01 == *p10 || *p01
       == *p11 || *p01 == *p12)
58 {
59 amefprintf(stderr, "ACHTUNG: 2 oder mehr Maschinen mit gleicher Prioritaet!");
60 error = 2;
61 }
62 if (*p02 == *p03 || *p02 == *p04 || *p02 == *p05 || *p02 == *p06 || *p02 == *
       p07 || *p02 == *p08 || *p02 == *p09 || *p02 == *p10 || *p02 == *p11 || *p02
       == *p12)
63 {
64
65 amefprintf(stderr, "ACHTUNG: 2 oder mehr Maschinen mit gleicher Prioritaet!");
66
67 error = 2;
68 }
69 if (*p03 == *p04 || *p03 == *p05 || *p03 == *p06 || *p03 == *p07 || *p03 == *
       p08 || *p03 == *p09 || *p03 == *p10 || *p03 == *p11 || *p03 == *p12)
70 {
71 amefprintf(stderr, "ACHTUNG: 2 oder mehr Maschinen mit gleicher Prioritaet!");
72 error = 2;
73 }
74 if (*p04 == *p05 || *p04 == *p06 || *p04 == *p07 || *p04 == *p08 || *p04 == *
       p09 || *p04 == *p10 || *p04 == *p11 || *p04 == *p12)
75 {
76 amefprintf(stderr, "ACHTUNG: 2 oder mehr Maschinen mit gleicher Prioritaet!");
77 error = 2;
78 }
79 if (*p05 == *p06 || *p05 == *p07 || *p05 == *p08 || *p05 == *p09 || *p05 == *
       p10 || *p05 == *p11 || *p05 == *p12)
80 {
81 amefprintf(stderr, "ACHTUNG: 2 oder mehr Maschinen mit gleicher Prioritaet!");
82 error = 2;
83 }
84 if (*p06 == *p07 || *p06 == *p08 || *p06 == *p09 || *p06 == *p10 || *p06 == *
       p11 || *p06 == *p12)
85 {
86 amefprintf(stderr, "ACHTUNG: 2 oder mehr Maschinen mit gleicher Prioritaet!");
87 error = 2;
88 }
89 if (*p07 == *p08 || *p07 == *p09 || *p07 == *p10 || *p07 == *p11 || *p07 == *
       p12)
90 {
91 amefprintf(stderr, "ACHTUNG: 2 oder mehr Maschinen mit gleicher Prioritaet!");
92 error = 2;
93 }
94 if (*p08 == *p09 || *p08 == *p10 || *p08 == *p11 || *p08 == *p12)
95 {
96 amefprintf(stderr, "ACHTUNG: 2 oder mehr Maschinen mit gleicher Prioritaet!");
97 error = 2;
98 }
99 if (*p09 == *p10 || *p09 == *p11 || *p09 == *p12)
100 {
101 amefprintf(stderr, "ACHTUNG: 2 oder mehr Maschinen mit gleicher Prioritaet!");
102 error = 2;
103 }
104 if (*p10 == *p11 || *p10 == *p12)
105 {
```

```
106 amefprintf(stderr, "ACHTUNG: 2 oder mehr Maschinen mit gleicher Prioritaet!");
107 error = 2;
108 }
109 if (*p11 == *p12)
110 {
111 amefprintf(stderr, "ACHTUNG: 2 oder mehr Maschinen mit gleicher Prioritaet!");
112 error = 2;
113 }
114  if(error == 1)
115    {
116       amefprintf(stderr, "\nWarning in %s instance %d.\n", _SUBMODELNAME_, *n);
117    }
118    else if(error == 2)
119    {
120       amefprintf(stderr, "\nFatal error in %s instance %d.\n", _SUBMODELNAME_,
     *n);
121       amefprintf(stderr, "Terminating the program.\n");
122       AmeExit(1);
123    }
124 /*Fehlerbehandlung*/
125 *Pelbges = *Pelb01+*Pelb02+*Pelb03+*Pelb04+*Pelb05+*Pelb06+*Pelb07+*Pelb08+*
     Pelb09+*Pelb10+*Pelb11+*Pelb12;
126 *Qkbges = *Qkb01+*Qkb02+*Qkb03+*Qkb04+*Qkb05+*Qkb06+*Qkb07+*Qkb08+*Qkb09+*Qkb10
     +*Qkb11+*Qkb12;
127 *abw = *Qkbges - *Qkb;
128 /*Outputs unused*/
129 *f02 = 0;
130 *f03 = 0;
131 *f04 = 0;
132 *f05 = 0;
133 *f06 = 0;
134 *f07 = 0;
135 *f08 = 0;
136 *f09 = 0;
137 /*Outputs unused*/
138 *Qwrg = *Qwrg01+*Qwrg02+*Qwrg03+*Qwrg04+*Qwrg05+*Qwrg06+*Qwrg07+*Qwrg08+*Qwrg09
     +*Qwrg10+*Qwrg11+*Qwrg12;
139 if (*aQkb > 0)
140    {
141    *Qkb = 0;
142    }
143    else
144    {
145    *Qkb = *aQkb;
146    }
147
148 /*Steuerung mon*/
149 if ((*Qkbges-*Qkb) > Hyst && *mon <= 12)
150    {
151    x = 0;
152    }
153    else if ((*Qkbges-*Qkb) < -Hyst && *mon >= 0)
154    {
155    x = 1;
156    }
157    else
158    {
159    x = 2;
160    }
161
```

```
162 if (x == 0)
163    {
164    *mon = *mon+1;
165    }
166
167 if (x == 1)
168    {
169    *mon = *mon-1;
170    }
171 /*Steuerung mon*/
172
173 if (*mon == *p01)
174    {
175    *dB01 = 0.000277777778;
176    *aQkb01=(*Qkb-(*Qkb02+*Qkb03+*Qkb04+*Qkb05+*Qkb06+*Qkb07+*Qkb08+*Qkb09+*
       Qkb10+*Qkb11+*Qkb12));
177    if (*aQkb01 < -*Cap01 && *aQkb01 < 0)
178       {
179       *Qkb01 = -*Cap01;
180       }
181       else if (*aQkb01 > -*Cap01 && *aQkb01 < 0 && *Cap01 !=0)
182       {
183       *Qkb01 = *aQkb01;
184       }
185       else
186       {
187       *Qkb01 = 0;
188       }
189    }
190 if (*mon > *p01)
191    {
192    *dB01 = 0.000277777778;
193    *Qkb01 = -*Cap01;
194    }
195 if (*mon < *p01)
196    {
197    *dB01 = 0;
198    *Qkb01 = 0;
199    }
200
201 if (*mon == *p02)
202    {
203    *dB02 = 0.000277777778;
204    *aQkb02=(*Qkb-(*Qkb01+*Qkb03+*Qkb04+*Qkb05+*Qkb06+*Qkb07+*Qkb08+*Qkb09+*
       Qkb10+*Qkb11+*Qkb12));
205    if (*aQkb02 < -*Cap02 && *aQkb02 < 0)
206       {
207       *Qkb02 = -*Cap02;
208       }
209       else if (*aQkb02 > -*Cap02 && *aQkb02 < 0 && *Cap02 !=0)
210       {
211       *Qkb02 = *aQkb02;
212       }
213       else
214       {
215       *Qkb02 = 0;
216       }
217    }
218 if (*mon > *p02)
219    {
```

```
220     *dB02 = 0.000277777778;
221     *Qkb02 = -*Cap02;
222     }
223 if (*mon < *p02)
224     {
225     *dB02 = 0;
226     *Qkb02 = 0;
227     }
228
229 if (*mon == *p03)
230     {
231     *dB03 = 0.000277777778;
232     *aQkb03=(*Qkb-(*Qkb01+*Qkb02+*Qkb04+*Qkb05+*Qkb06+*Qkb07+*Qkb08+*Qkb09+*
        Qkb10+*Qkb11+*Qkb12));
233     if (*aQkb03 < -*Cap03 && *aQkb03 < 0)
234        {
235        *Qkb03 = -*Cap03;
236        }
237        else if (*aQkb03 > -*Cap03 && *aQkb03 < 0 && *Cap03 !=0)
238        {
239        *Qkb03 = *aQkb03;
240        }
241        else
242        {
243        *Qkb03 = 0;
244        }
245     }
246 if (*mon > *p03)
247     {
248     *dB03 = 0.000277777778;
249     *Qkb03 = -*Cap03;
250     }
251 if (*mon < *p03)
252     {
253     *dB03 = 0;
254     *Qkb03 = 0;
255     }
256
257 if (*mon == *p04)
258     {
259     *dB04 = 0.000277777778;
260     *aQkb04=(*Qkb-(*Qkb01+*Qkb02+*Qkb03+*Qkb05+*Qkb06+*Qkb07+*Qkb08+*Qkb09+*
        Qkb10+*Qkb11+*Qkb12));
261     if (*aQkb04 < -*Cap04 && *aQkb04 < 0)
262        {
263        *Qkb04 = -*Cap04;
264        }
265        else if (*aQkb04 > -*Cap04 && *aQkb04 < 0 && *Cap04 !=0)
266        {
267        *Qkb04 = *aQkb04;
268        }
269        else
270        {
271        *Qkb04 = 0;
272        }
273     }
274 if (*mon > *p04)
275     {
276     *dB04 = 0.000277777778;
277     *Qkb04 = -*Cap04;
```

```
278     }
279 if (*mon < *p04)
280     {
281     *dB04 = 0;
282     *Qkb04 = 0;
283     }
284 
285 if (*mon == *p05)
286     {
287     *dB05 = 0.000277777778;
288     *aQkb05=(*Qkb-(*Qkb01+*Qkb02+*Qkb03+*Qkb04+*Qkb06+*Qkb07+*Qkb08+*Qkb09+*
        Qkb10+*Qkb11+*Qkb12));
289     if (*aQkb05 < -*Cap05 && *aQkb05 < 0)
290        {
291        *Qkb05 = -*Cap05;
292        }
293        else if (*aQkb05 > -*Cap05 && *aQkb05 < 0 && *Cap05 !=0)
294        {
295        *Qkb05 = *aQkb05;
296        }
297        else
298        {
299        *Qkb05 = 0;
300        }
301     }
302 if (*mon > *p05)
303     {
304     *dB05 = 0.000277777778;
305     *Qkb05 = -*Cap05;
306     }
307 if (*mon < *p05)
308     {
309     *dB05 = 0;
310     *Qkb05 = 0;
311     }
312 
313 if (*mon == *p06)
314     {
315     *dB06 = 0.000277777778;
316     *aQkb06=(*Qkb-(*Qkb01+*Qkb02+*Qkb03+*Qkb04+*Qkb05+*Qkb07+*Qkb08+*Qkb09+*
        Qkb10+*Qkb11+*Qkb12));
317     if (*aQkb06 < -*Cap06 && *aQkb06 < 0)
318        {
319        *Qkb06 = -*Cap06;
320        }
321        else if (*aQkb06 > -*Cap06 && *aQkb06 < 0 && *Cap06 !=0)
322        {
323        *Qkb06 = *aQkb06;
324        }
325        else
326        {
327        *Qkb06 = 0;
328        }
329     }
330 if (*mon > *p06)
331     {
332     *dB06 = 0.000277777778;
333     *Qkb06 = -*Cap06;
334     }
335 if (*mon < *p06)
```

```
336     {
337     *dB06 = 0;
338     *Qkb06 = 0;
339     }
340
341 if (*mon == *p07)
342     {
343     *dB07 = 0.000277777778;
344     *aQkb07=(*Qkb-(*Qkb01+*Qkb02+*Qkb03+*Qkb04+*Qkb05+*Qkb06+*Qkb08+*Qkb09+*
        Qkb10+*Qkb11+*Qkb12));
345     if (*aQkb07 < -*Cap07 && *aQkb07 < 0)
346        {
347        *Qkb07 = -*Cap07;
348        }
349        else if (*aQkb07 > -*Cap07 && *aQkb07 < 0 && *Cap07 !=0)
350        {
351        *Qkb07 = *aQkb07;
352        }
353        else
354        {
355        *Qkb07 = 0;
356        }
357     }
358 if (*mon > *p07)
359     {
360     *dB07 = 0.000277777778;
361     *Qkb07 = -*Cap07;
362     }
363 if (*mon < *p07)
364     {
365     *dB07 = 0;
366     *Qkb07 = 0;
367     }
368
369 if (*mon == *p08)
370     {
371     *dB08 = 0.000277777778;
372     *aQkb08=(*Qkb-(*Qkb01+*Qkb02+*Qkb03+*Qkb04+*Qkb05+*Qkb06+*Qkb07+*Qkb09+*
        Qkb10+*Qkb11+*Qkb12));
373     if (*aQkb08 < -*Cap08 && *aQkb08 < 0)
374        {
375        *Qkb08 = -*Cap08;
376        }
377        else if (*aQkb08 > -*Cap08 && *aQkb08 < 0 && *Cap08 !=0)
378        {
379        *Qkb08 = *aQkb08;
380        }
381        else
382        {
383        *Qkb08 = 0;
384        }
385     }
386 if (*mon > *p08)
387     {
388     *dB08 = 0.000277777778;
389     *Qkb08 = -*Cap08;
390     }
391 if (*mon < *p08)
392     {
393     *dB08 = 0;
```

```
394     *Qkb08 = 0;
395     }
396
397 if (*mon == *p09)
398     {
399     *dB09 = 0.000277777778;
400     *aQkb09=(*Qkb-(*Qkb01+*Qkb02+*Qkb03+*Qkb04+*Qkb05+*Qkb06+*Qkb07+*Qkb08+*
        Qkb10+*Qkb11+*Qkb12));
401     if (*aQkb09 < -*Cap09 && *aQkb09 < 0)
402         {
403         *Qkb09 = -*Cap09;
404         }
405         else if (*aQkb09 > -*Cap09 && *aQkb09 < 0 && *Cap09 !=0)
406         {
407         *Qkb09 = *aQkb09;
408         }
409         else
410         {
411         *Qkb09 = 0;
412         }
413     }
414 if (*mon > *p09)
415     {
416     *dB09 = 0.000277777778;
417     *Qkb09 = -*Cap09;
418     }
419 if (*mon < *p09)
420     {
421     *dB09 = 0;
422     *Qkb09 = 0;
423     }
424
425 if (*mon == *p10)
426     {
427     *dB10 = 0.000277777778;
428     *aQkb10=(*Qkb-(*Qkb01+*Qkb02+*Qkb03+*Qkb04+*Qkb05+*Qkb06+*Qkb07+*Qkb08+*
        Qkb09+*Qkb11+*Qkb12));
429     if (*aQkb10 < -*Cap10 && *aQkb10 < 0)
430         {
431         *Qkb10 = -*Cap10;
432         }
433         else if (*aQkb10 > -*Cap10 && *aQkb10 < 0 && *Cap10 !=0)
434         {
435         *Qkb10 = *aQkb10;
436         }
437         else
438         {
439         *Qkb10 = 0;
440         }
441     }
442 if (*mon > *p10)
443     {
444     *dB10 = 0.000277777778;
445     *Qkb10 = -*Cap10;
446     }
447 if (*mon < *p10)
448     {
449     *dB10 = 0;
450     *Qkb10 = 0;
451     }
```

```
452
453 if (*mon == *p11)
454    {
455    *dB11 = 0.000277777778;
456    *aQkb11=(*Qkb-(*Qkb01+*Qkb02+*Qkb03+*Qkb04+*Qkb05+*Qkb06+*Qkb07+*Qkb08+*
        Qkb09+*Qkb10+*Qkb12));
457    if (*aQkb11 < -*Cap11 && *aQkb11 < 0)
458       {
459       *Qkb11 = -*Cap11;
460       }
461       else if (*aQkb11 > -*Cap11 && *aQkb11 < 0 && *Cap11 !=0)
462       {
463       *Qkb11 = *aQkb11;
464       }
465       else
466       {
467       *Qkb11 = 0;
468       }
469    }
470 if (*mon > *p11)
471    {
472    *dB11 = 0.000277777778;
473    *Qkb11 = -*Cap11;
474    }
475 if (*mon < *p11)
476    {
477    *dB11 = 0;
478    *Qkb11 = 0;
479    }
480
481 if (*mon == *p12)
482    {
483    *dB12 = 0.000277777778;
484    *aQkb12=(*Qkb-(*Qkb01+*Qkb02+*Qkb03+*Qkb04+*Qkb05+*Qkb06+*Qkb07+*Qkb08+*
        Qkb09+*Qkb10+*Qkb11));
485    if (*aQkb12 < -*Cap12)
486       {
487       *Qkb12 = -*Cap12;
488       }
489       else if (*aQkb12 > -*Cap12 && *aQkb12 < 0 && *Cap12 !=0)
490       {
491       *Qkb12 = *aQkb12;
492       }
493       else
494       {
495       *Qkb12 = 0;
496       }
497    }
498 if (*mon > *p12)
499    {
500    *dB12 = 0.000277777778;
501    *Qkb12 = -*Cap12;
502    }
503 if (*mon < *p12)
504    {
505    *dB12 = 0;
506    *Qkb12 = 0;
507    }
508 }
```

7 Literaturverzeichnis

[1] K. J. Chua, S. K. Chou, W. M. Yang und J. Yan, „Achieving better energy-efficient air conditioning - A review of technologies and strategies," *Applied Energy 104,* pp. 87-104, 2013.

[2] S. Hunag, W. Zuo und M. D. Sohn, „Amelioration of the cooling load based chiller sequencing control," *Applied Energy 168,* pp. 204-215, 2016.

[3] B. Fan, X. Jin und Z. Du, „Optimal control strategies for multi-chiller system based on probability density distribution of cooling load ratio," *Energy and Buildings 43,* pp. 2813-2821, 2011.

[4] K. M. Powell, W. J. Cole, U. F. Ekarika und T. F. Edgar, „Optimal chiller loading in a district cooling system with thermal energy storage," *Energy 50,* pp. 445-453, 2013.

[5] F. W. Yu und K. T. Chan, „Improved energy management of chiller systems by mulitvariate and data envelopment analyses," *Applied Energy 92,* pp. 168-174, 2012.

[6] J. F. Ordonez, V. Gutierrez, C. A. Canizares, K. Bhattacharya und S. Paudyal, „Applications of optimal industrial load management modeling in smart grids," in *IEEE Manchester PowerTech*, Manchester, 2017.

[7] S. R. Thangavelu, A. Myat und A. Khambadkone, „Energy optimization methodology of multi-chiller plant in commercial buildings," *Energy 123,* pp. 64-76, 2017.

[8] A. Beghi, L. Cecchinato und M. Rampazzo, „A mulit-phase genetic algortihm for the efficient management of multi-chiller systems," *Energy Conversion and Management 52,* pp. 1650-1661, 2011.

[9] Z. Ma und S. Wang, „Supervisory and optimal control of central chiller plants using simplified adaptive models and genetic algorithm," *Applied Energy 88,* pp. 198-211, 2011.

[10] C.-C. Lo, S.-H. Tsai und B.-S. Lin, „Economic dispatch of chiller plant by improved ripple bee swarm optimization algorithm for saving energy," *Applied Thermal Engineering 100,* pp. 1140-1148, 2016.

[11] C.-L. Chen, Y.-C. Chang und T.-S. Chan, „Applying smart models for energy saving in optimal chiller loading," *Energy and Buildings 68,* pp. 364-371, 2014.

[12] M. Karami und L. Wang, „Particle Swarm optimization for control operation of an all-variable speed water-cooled chiller plant," *Applied Thermal Engineering 130,* pp. 962-978, 2018.

[13] X. Wei, G. Xu und A. Kusiak, „Modeling and optimization of a chiller plant," *Energy 73,* pp. 898-907, 2014.

[14] L. März, W. Krug, O. Rose und G. Weigert, Simulation und Optimierung in Produktion und Logistik, Berlin Heidelberg: Springer-Verlag, 2011.

[15] A. M. Law und W. D. Kelton, Simulation Modeling & Analysis, New York: McGraw-Hill, 1991, 2. Auflage.

[16] S. Wenzel, M. Weiß, S. Collisi_Böhmer, H. Pitsch und O. Rose, Qualitätskriterien für die Simulation in Produktion und Logistik, Berlin Heidelberg: Springer-Verlag, 2008.

[17] C. Hermann, S. Thiede, S. Kara und J. Hesselbach, „Energy oriented simulation of manufacturing systems - Concept and application," *CIRP Annals - Manufacturing Technology 60,* pp. 45-48, 2011.

[18] I. Leobner, P. Smolek, B. Heinzl, I. Kovacic, F. Dür, K. Ponweiser und W. Kastner, „Balanced Manufacturing – a Methodology for Energy Efficient Production Plant Operation," in *Proceedings of the 10th Conference on Sustainable Development of Energy*, 2015.

[19] J. Dohmann, Thermodynamik der Kälteanalgen und Wärmepumpen - Grundlagen und Anwendungen der Kältetechnik, Berlin Heidelberg: Springer-Verlag, 2016.

[20] H. D. Baehr und S. Kabelac, Thermodynamik - Grundlagen und technische Anwendungen, Berlin Heidelberg: Springer Verlag, 2016, 16. Auflage.

[21] S. Priya, „Evaluation of chiller modeling approaches and their usability for fault detection," Department of Mechanical Engineering, University of California, Berkeley, Kalifornien, 2001.

[22] P. Lindskog, „Fuzzy Identification from a Grey Box Modeling Point of View," in *Fuzzy model identification*, Berlin Heidelberg, Springer Berlin Heidelberg, 1997, pp. 3-50.

[23] M. Hydeman und K. L. Gillespie, „Tools and Techniques to Calibrate Electric Chiller Component Models," in *ASHRAE Transactions*, 2002.

[24] P. Smolek, I. Leobner, G. Gourlis, B. Mörzinger, B. Heinzl und K. Ponweiser, „Hybrid Building Performance Simulation Models for Industrial Energy Efficiency Applications," in *Proceedings of the 11th Conference on Sustainable Development of Energy, Water, and Environment Systems*, Lissabon, 2016.

[25] D. Monfet und R. Zmeureanu, „Identification of the electric chiller model for the energyplus program using monitored data in an existing cooling plant," in *12th Conference of International Building Performance Simulation Association*, Sydney, 2011.

[26] B. Mörzinger, C. Loschan, F. Kloibhofer und F. Bleicher, „A modular, holistic optimization approach for industrial appliances," in *10th CIRP Conference on Industrial Product-Service Systems*, Linköping, 2018.

[27] A. Askarzadeh und L. d. S. Coelho, „Using two improved particle swarm optimization variants for optimization of daily electrical power consumption in multi-chiller systems," *Applied Thermal Engineering 89,* pp. 640-646, 2015.

[28] E. Salari und A. Askarzadeh, „A new solution for loading optimization of mulit-chiller systems by general algebraic modeling system," *Applied Thermal Engineering 84,* pp. 429-436, 2015.

[29] E. Elbeltagi, T. Hegazy und D. Grierson, „Comparison among five evolutionary-based optimization algorithms," *Advanced Engineering Informatics 19,* pp. 43-53, 2005.

[30] K. Weicker, Evolutionäre Algorithmen, Wiesbaden: Springer Vieweg, 2015, 3. Auflage.

[31] Y. Jin, M. Olhofer und B. Sendhoff, „On evolutionary optimization with approximate fitness function," in *Proceedings of the 2nd Annual Conference on Genetic and Evolutionary Computation*, Nevada, 2000.

[32] P. Tittmann, Einführung in die Kombinatorik, Berlin Heidelberg: Springer-Verlag, 2014, 2. Auflage.

[33] „LMS Imagine.Lab AMESIM Webseite," Siemens AG, [Online]. Available: https://www.plm.automation.siemens.com/de_at/products/lms/imagine-lab/amesim/. [Zugriff am 16 1 2018].

[34] Siemens AG, „AMESIM Manual,“ Siemens PLM Software, 2017.

[35] T. Beier und P. Wurl, Regelungstechnik - Basiswissen, Grundlagen, Anwendungsbeispiele, München: Carl Hanser Verlag GmbH & Co. KG, 2013.

[36] I. Rößler und A. Ungerer, Statistik für Wirtschaftswissenschaftler, Heidelberg: Physica-Verlag, 2008.

[37] H. Bossel, Systeme, Dynamik, Simulation: Modellbildung, Analyse und Simulation komplexer Systeme, Norderstedt: Books on Demand GmbH, 2004.

[38] Verein Deutscher Ingenieure, VDI-Wärmeatlas, Berlin Heidelberg: Springer-Verlag, 2013, 11. bearbeitete und erweiterte Auflage.

8 Abbildungsverzeichnis

Abbildung 1: Prozess simulationsbasierte Optimierung in Anlehnung an [14].....5
Abbildung 2: Lokale und globale Optima der Zielfunktion f(x) in Anlehnung an [14]....7
Abbildung 3: Vorgehensweise bei Erstellung der Simulation in Anlehnung an [16].....8
Abbildung 4: Generischer Cube in Anlehnung an [18].....10
Abbildung 5: Konzeptmodell der Kälteanlage.....11
Abbildung 6: Wärmeströme der Kältemaschine in Anlehnung an [19].....12
Abbildung 7: Prinzipdarstellung Kompressionskältemaschine in Anlehnung an [20].15
Abbildung 8: Zerklüfteter Lösungsraum in Anlehnung an [30].....30
Abbildung 9: Aufbau der AMESIM Simulationssoftware in Anlehnung an [34] 32
Abbildung 10: Ports und externe Variablen des Kältemaschinenmodells.....34
Abbildung 11: Von AMESIM generiertes Template 35
Abbildung 12: Fehlermeldung aufgrund eines falschen Prioritätsvektors 36
Abbildung 13: Energie- und Informationsports Kältemaschine.....37
Abbildung 14: Energie- und Informationsports Energienetz 40
Abbildung 15: Regelkreis des Energienetzes in Anlehnung an [35] 42
Abbildung 16: Informationsports Verteilfunktion 43
Abbildung 17: Zeitlicher Verlauf m_{on}.....46
Abbildung 18: Zeitlicher Verlauf $Qkb,9$.....47
Abbildung 19: Zeitlicher Verlauf $Qkb,3$.....47
Abbildung 20: mon in Abhängigkeit von $Hyst$ 48
Abbildung 21: Informationsports Zielfunktionsbaustein 49
Abbildung 22: Ports des Rückkühlwerks 53
Abbildung 23: Standardkomponenten aus AMESIM-Bibliothek.....54
Abbildung 24: Benutzerdefinierte Komponenten 55
Abbildung 25: Sketch-Ansicht der verschalteten Komponenten.....57
Abbildung 26: Input und Outputs Tabellenbaustein.....59
Abbildung 27: Konvertieren des Tabellenoutputs v_i in den Prioritätszahlen p_i.....60
Abbildung 28: Konvertierung von v_i auf p_i in Programmcode umgesetzt.....61
Abbildung 29: Arbeitsweise des GA in AMESIM in Anlehnung an [34].....63
Abbildung 30: Gleitende Zielfunktionsmittelwerte.....65
Abbildung 31: Aufbau der Kälteanlage 68
Abbildung 32: Qk,b im 24h-Verlauf 69
Abbildung 33: Elektrischer Leistungsbedarf Pel,b der Kältemaschinen 73
Abbildung 34: Temperaturverlauf Kältenetz 74
Abbildung 35: Zeitverlauf der Gesamtkältemaschinenkapazität $Qmax,ges$ 75
Abbildung 36: Zeitlicher Verlauf von m_{on} 76
Abbildung 37: Temperaturverlauf Kühlnetz 1 77
Abbildung 38: Temperaturverlauf Kühlnetz 2 77
Abbildung 39: Von RKW 1 abgegebene Kühlleistung Qc,erz im zeitlichen Verlauf...78

Abbildung 40: Von RKW 2 abgegebene Kühlleistung Qc, erz im zeitlichen Verlauf...79
Abbildung 41: Temperaturverlauf Kältenetz mit nur 6 Kältemaschinen.....80
Abbildung 42: Verlauf von Qk, b für niedrige Auslastung.....82
Abbildung 43: Verlauf von Qk, b für hohe Auslastung.....82

9 Formelverzeichnis

Formelzeichen	Bedeutung	Einheit
$\dot{Q}_{ab}$	Abgeführter Wärmestrom	kW
$\dot{Q}_{zu}$	Zugeführter Wärmestrom	kW
$\dot{Q}_{bez}$	Bezieherwärmestrom	kW
$\dot{Q}_{bez,b}$	Bedarf Bezieherwärmestrom	kW
$\dot{Q}_{erz}$	Ezeugerwärmestrom	kW
$\dot{Q}_{erz,b}$	Bedarf Erzeugerwärmestrom	kW
$\dot{Q}_{k}$	Kälteleistung	kW
$\dot{Q}_{k,b}$	Bedarf Kälteleistung	kW
$\dot{Q}_{c}$	Kühlleistung	kW
$\dot{Q}_{c,b}$	Bedarf Kühlleistung	kW
$\dot{Q}_{wb,brutto}$	Bruttowärmestrom	kW
$\dot{Q}_{wb,netto}$	Nettowärmestrom	kW
P_{el}	Elektrische Leistung	kW
$P_{el,b}$	Elektrischer Leistungsbedarf	kW
$\dot{Q}_{k,ges}$	Gesamtkälteleistung	kW
$Hyst$	Hysterese	kW
$\dot{Q}_{max,ges}$	Maximale Gesamtkälteleistung	kW
$\dot{Q}_{max}$	Maximale Kälteleistung	kW
$\dot{Q}_{c,max}$	Maximale Kühlleistung	kW
$\bar{\dot{Q}}_{max,ges}$	Mittlere maximale Gesamtkälteleistung	kW
$\bar{\dot{Q}}_{k,b}$	Mittlerer Kälteleistungsbedarf	kW
$\dot{Q}_{ref}$	Referenzkälteleistung	kW
$\dot{Q}_{kb,temp}$	Temporäre Kälteleistung	kW
$\dot{Q}_{wrg}$	Wärmestrom Wärmerückgewinnung	kW
$Q_{k,b}$	Kältebedarf	kWh
$Q_{wb,brutto}$	Bruttowärmebedarf	kWh
$Q_{wb,netto}$	Nettowärmebedarf	kWh
$E_{el,b}$	Elektrischer Energiebedarf	kWh
$Q_{c,pen}$	Strafterm für Überschreitung von $\dot{Q}_{c,max}$	kWh
c	Spezifische Wärmekapazität	kJ/kgK
C	Speicherkapazität	kJ/K
H	Enthalpie	kJ
$regeldiff$	Regelabweichung	kJ

E_{soll}	Soll-Speicherstand	kJ
E	Speicherstand	kJ
m	Masse	kg
ρ	Dichte	kg/m^3
B_i	Betriebsstunden Maschine i	h
$B_{Gnach,i}$	Betriebsstunden Maschine i nach Durchlauf	h
$B_{Gvor,i}$	Betriebsstunden Maschine i vor Durchlauf	h
$B_{init,i}$	Initiale Betriebsstunden Maschine i	h
$\bar{B}_{Gnach}$	Mittlere Betriebsstunden nach Durchlauf	h
$\bar{B}_{Gvor}$	Mittlere Betriebsstunden vor Durchlauf	h
t	Simulationszeit	s
T_k	Kaltwasservorlauftemperatur	°C
T_c	Kühlwasserrücklauftemperatur	°C
T	Netztemperatur	°C
T_0	Referenztemperatur	°C
T_{Ziel}	Zieltemperatur	°C
$RMSE_{rel}$	Relativer Root Mean Square Error	%
$\overline{RMSE}_{rel}$	Mittlerer relativer Root Mean Square Error	%
P	Mutationswahrscheinlichkeit	%
r	Reproduktionsrate	%
G	Anzahl Generationen	-
N	Anzahl Individuen pro Population	-
m_{on}	Anzahl laufender Maschinen	-
P_n	Anzahl Permutationen	-
R	Anzahl Simulationsdurchläufe	-
k_i	Betriebskostenfaktor Maschine i	-
$index$	Ganzzahlige Variable	-
WRG	Ganzzahliger Parameter WRG	-
ω_n	Gewichtungsfaktoren	-
$CAPFT$	Kapazitätskennlinie	-
c_i	Koeffizienten Kapazitätskennlinie	-
a_i	Koeffizienten Teillastkennlinie	-
b_i	Koeffizienten Wirkungsgradkennlinie	-
k_P	Kostenfaktor elektrische Energie	-
k_Q	Kostenfaktor thermische Energie	-

COP	Leistungszahl	-
COP_{ref}	Referenzleistungszahl	-
d_A	Mittlere absolute Abweichung	-
α	Mutationsamplitude	-
PLR	Part Load Ratio	-
$\vec{p}$	Prioritätsvektor	-
p	Prioritätszahl	-
η_0	Referenzwirkungsgrad Kühlleistung	-
η_1	Referenzwirkungsgrad WRG	-
σ	Standardabweichung	-
$EIRFPLR$	Teillastkennlinie	-
v	Vektoreintrag	-
K_i	Verstärkung I-Anteil	-
K_p	Verstärkung P-Anteil	-
EER	Wirkungsgrad	-
$EIRFT$	Wirkungsgradkennlinie	-
f_i	Zielfunktionsbestandteil	-

10 Tabellenverzeichnis

Tabelle 1: Plausibilitätsgrenzen für Temperaturmesswerte [26] 20
Tabelle 2: Aufgenommene Messgrößen und Dateigrößen der Datensätze [26] 20
Tabelle 3: Relativer RMSE für Pel und Qc [26] 21
Tabelle 4: Externe Variablen des Kältemaschinenmodells 38
Tabelle 5: Parameter Kältemaschinenmodell 38
Tabelle 6: Interne Variablen Kältemaschinenmodell 39
Tabelle 7: Externe Variablen des Energienetzmodells 41
Tabelle 8: Parameter Energienetzmodell 41
Tabelle 9: Interne Variablen Energienetzmodell 41
Tabelle 10: Zuordnung der Größen des Standardregelkreises 42
Tabelle 11: Externe Variablen der Verteilfunktion 44
Tabelle 12: Parameter der Verteilfunktion 44
Tabelle 13: Interne Variablen der Verteilfunktion 44
Tabelle 14: Werte der Prioritätszahlen aufgrund des Beispielvektors p 45
Tabelle 15: Variablen der Ports des Zielfunktionsbausteins 49
Tabelle 16: Parameter Zielfunktionsbaustein 49
Tabelle 17: Interne Variablen Zielfunktionsbaustein 49
Tabelle 18: Variablen der Ports des Rückkühlwerkmodells 53
Tabelle 19: Parameter Rückkühlwerk 53
Tabelle 20: Beispiel für Tabelleneintrag 59
Tabelle 21: Getestete Parametersets für genetischen Algorithmus 64
Tabelle 22: Verwendete Parameter für genetischen Algorithmus 66
Tabelle 23: Eingangsgrößen Simulationsszenario 70
Tabelle 24: Parameter Kältenetz 70
Tabelle 25: Parameter Kühlnetz 70
Tabelle 26: Gewichtungsfaktoren der Zielfunktion 71
Tabelle 27: Kostenfaktoren und initiale Betriebsstunden 71
Tabelle 28: Auswertung Permutationsliste 72
Tabelle 29: Einsparung durch optimierten Prioritätsvektor 73
Tabelle 30: Betriebszeiten der Maschinen 76
Tabelle 31: Simulationsszenarien für die Bewertung des Einsparungspotenzials 81
Tabelle 32: Gewichtungsfaktoren der Zielfunktion 83
Tabelle 33: Ergebnisse Optimierung elektrischer Energiebedarf 83
Tabelle 34: Gewichtungsfaktoren der Zielfunktion 84
Tabelle 35: Ergebnisse Optimierung Gesamtenergiebedarf 84
Tabelle 36: Szenarien und Einsparungen abzüglich des mittleren Fehlers 85

11 Abkürzungsverzeichnis

bzw.	beziehungsweise
ca.	circa
d.h.	das heißt
etc.	et cetera
z.B.	zum Beispiel
bspw.	beispielsweise
Fa.	Firma
WRG	Wärmerückgewinnung
RMSE	Root Mean Square Error
RLS	Recursive Least Squares
OCL	Optimal Chiller Loading
DOCL	Daily Optimal Chiller Loading
OCS	Optimal Chiller Sequencing
PLR	Part Load Ratio
COP	Coefficient of Performance
GA	Genetischer Algorithmus
GAMS	General Algebraic Modelling System
IRBSO	Improved Ripple Bee Swarm Optimization
EPSO	Elitism-Based Particle Swarm Optimization
MA-PSO	Multi-Agent Particle Swarm Optimization