

**UNIVERSITY OF DORTMUND**

---

REIHE COMPUTATIONAL INTELLIGENCE

---

COLLABORATIVE RESEARCH CENTER 531

---

Design and Management of Complex Technical Processes  
and Systems by means of Computational Intelligence Methods

---

Multikriterielle evolutionäre  
Aufstellungsoptimierung von Chemie-Anlagen  
unter Beachtung gewichteter Designregeln

Ingo Mierswa und Thorsten Geisbe

No. CI-188/04

Technical Report

ISSN 1433-3325

Dezember 2004

Secretary of the SFB 531 · University of Dortmund · Dept. of Computer Science/XI  
44221 Dortmund · Germany

---

This work is a product of the Collaborative Research Center 531, "Computational Intelligence," at the University of Dortmund and was printed with financial support of the Deutsche Forschungsgemeinschaft.

# Multikriterielle evolutionäre Aufstellungsoptimierung von Chemie-Anlagen unter Beachtung gewichteter Designregeln

Ingo Mierswa und Thorsten Geisbe

Universität Dortmund

Fachbereich Informatik, Lehrstuhl für künstliche Intelligenz

E-Mail: [mierswa@ls8.cs.uni-dortmund.de](mailto:mierswa@ls8.cs.uni-dortmund.de), [th.geisbe@gmx.de](mailto:th.geisbe@gmx.de)

## Zusammenfassung

Die Planung und Aufstellung technischer Anlagen ist ein komplexer Prozess. Bereits für das einfache *Facility Placement Problem* konnte nachgewiesen werden, dass die Optimierung NP-hart ist. Ziel der Optimierung ist eine Anlage, die einer Vielzahl von Bewertungskriterien gleichzeitig genügen muss. Wir betrachten eine Variante dieses Problems mit unregelmäßigen Grundflächen, mehreren Dimensionen durch Verwendung eines Stahlbaus, Zonen ohne Bebauung sowie vordefinierten Wegen. Bisherige Ansätze zur Bearbeitung des einfachen Facility Placement Problems sind zudem nicht in der Lage, die multikriterielle Optimierung von Kosten und die Erfüllung gewichteter und unscharfer Designregeln durchzuführen. In dieser Arbeit verbinden wir einen multikriteriellen evolutionären Algorithmus zur Konstruktion von Pareto-optimalen Anlagenentwürfen mit Techniken der Fuzzy-Logik. Die geschickte Indexierung der Regeln und die Zuordnung der Konklusionen zu Komponenten erlaubt die effiziente Überprüfung des Regelerfüllungsgrads als Teil der Fitnessfunktion. Für eine real existierende Anlage zur Gasbehandlung werden auf diese Weise Anlagen konstruiert, die in allen Bewertungskriterien bessere Werte als das Original besitzen.

## 1 Einleitung

Die Planung einer verfahrenstechnischen Anlage ist eine komplexe Aufgabe, welche den gesamten Lebenszyklus der Anlage berücksichtigen muss. Die Kosten für die Planung und Realisation sowie der spätere Betrieb müssen möglichst gering sein. Die immer kürzer werdenden *time-to-market* Zyklen erfordern eine hohe Flexibilität und den massiven Einsatz von Rechnern. Es hat sich gezeigt, dass die Kosten für die Anlage in wesentlichen Teilen während der frühen Phasen des *basic-* und *extended basic engineering* beeinflusst werden können. Fehlentscheidungen in diesen Phasen können hohe Kosten während des Aufbaus und

Betriebs der Anlage erzeugen. Im *basic engineering* werden die chemischen Verfahren und die benötigten Komponenten ausgewählt bzw. optimiert. Für das anschließende *extended basic engineering* sind die wesentlichen Komponenten mit ihren Haupteigenschaften bekannt und müssen im Rahmen der Aufstellungsplanung in einem vorgegebenen oder zu entwickelnden Stahlbau möglichst optimal platziert werden. Alle folgenden Phasen des *detail engineering* nutzen diese Platzierung und erhöhen den Detailgrad der Planung und Darstellung (siehe Abbildung 1). Seit Jahren sind Programme für die 3D-Planung im *detail engineering* im Einsatz, aber die Aufstellungsplanung im *extended basic engineering* wird manuell bewältigt oder von nicht für diesen Einsatz geeigneten Programmen aus dem *detail engineering* unterstützt. Ziel ist die Unterstützung der Aufstellungsplanung im *extended basic engineering* durch geeignete Software. Dies ist notwendig, um:

- frühzeitig verlässliche Kostenabschätzungen zu bekommen und günstige Angebote erstellen zu können,
- mit standardisierten Abläufen bei der Planung von Anlagen das Expertenwissen in den Prozess reproduzierbar einfließen zu lassen, um es bei anderen Anlagen wiederzuverwenden,
- Arbeitsabläufe rationeller zu gestalten,
- Fehlplanungen zu vermeiden und insgesamt Kosten zu reduzieren.

## 1.1 Anforderungen an die Aufstellungsplanung

Aufgabe der Aufstellungsplanung ist die Anordnungsbestimmung sämtlicher für die Anlage relevanter und verfahrenstechnisch notwendiger Komponenten innerhalb der definierten Anlagengrenzen [23]. Für Chemieanlagen bedeutet dies die Positionierung von Maschinen und Apparaten innerhalb eines Stahlgerüsts und vorgegebener Bauflächen. In der Layoutplanung müssen Anforderungen aus verschiedenen Fachgebieten erfüllt werden:

*Verfahrenstechnik:* Anforderungen aus der Verfahrenstechnik hängen vom Prozess ab. Bei Pumpen muss ein geeigneter Vordruck vorhanden sein, der durch Höhenunterschiede eingestellt werden kann.

*Sicherheitstechnik:* Von der Sicherheitstechnik werden Anforderungen an den gefahrlosen Betrieb der Anlage sowie an den Schutz der Mitarbeiter gestellt.

*Apparatetechnik:* Aus der Apparatetechnik ergeben sich Lageanforderungen einzelner Komponenten, z. B. die Aufstellung schwerer Komponenten ins Erdgeschoss.

*Rohrleitungstechnik:* Alle wesentlichen Rohrleitungen sollten bei der Planung beachtet werden. Dies gilt insbesondere für Leitungen mit hohem Durchmesser oder aus teuren Materialien.

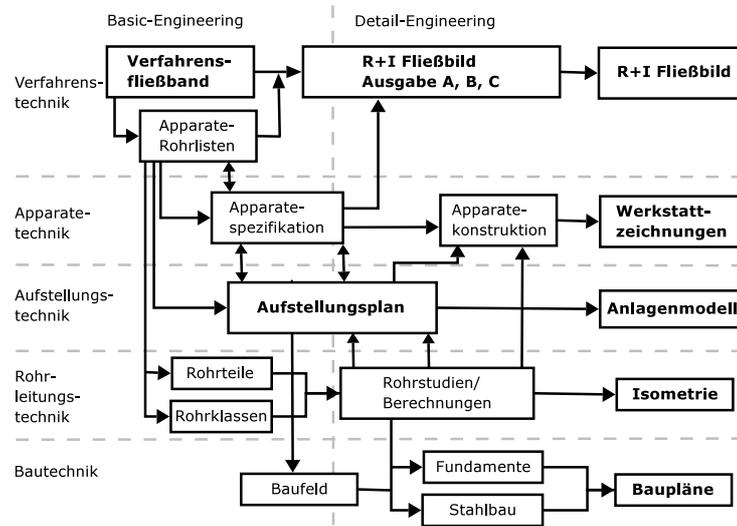


Abbildung 1: Der Planungsprozess für Chemieanlagen wird in *basic engineering* (links) und *detail engineering* (rechts) unterteilt. Die Aufstellungsplanung liegt an der Schnittstelle dieser Designphasen.

*Montage:* Anforderungen aus dem Bereich der Montage sollen den kostengünstigen Aufbau der Anlage sicherstellen.

*Bedienung:* Manuell zu bedienende Komponenten sollten zugänglich aufgestellt werden.

*Wartung:* Wartungsbedürftige Komponenten müssen für nötigen Hilfsmittel zugänglich sein.

Diese Anforderungen können als Regeln definiert werden, wodurch Erfahrungswissen konserviert wird [2]. Ein Anlagenentwurf ist genau dann optimal, wenn alle Regeln erfüllt sind und sonstige z.B. durch die Rohrlänge und Montage entstehenden Kosten minimiert sind. Die Schwierigkeit der Suche nach einer optimalen Aufstellung ergibt sich aus der Identifizierung und Beschreibung aller wichtigen Teile der Anlage und Räume, die Identifizierung und Beschreibung aller wichtigen Beziehungen zwischen den Teilen der Anlage und der Bewältigung der kombinatorischen Komplexität bei der Berechnung optimaler Layouts. Darüberhinaus sind die Anforderungen oftmals unscharf formuliert und widersprechen einander teilweise. Eine optimale Anlage kann also in der Praxis häufig nicht erstellt werden, da die oben genannten Kriterien konkurrieren. Für die automatisierte Suche nach einer optimalen Aufstellung ist eine Gewichtung dieser konkurrierenden Kriterien notwendig, die je nach Anlage variieren kann.

## 1.2 Rechnergestützte Layoutoptimierung

Die hier vorgestellte Layoutoptimierung einer Chemieranlage ist eine Abwandlung und Erweiterung bekannter Probleme. Aus dem VLSI Design ist das *Placement Problem* bekannt, dabei müssen beliebige rechteckige Baugruppen mit dem Ziel platziert werden, die Gesamtfläche zu minimieren. Dieses Problem ist NP-hart [8], sodass für praktische Anwendungen nur heuristische Verfahren anwendbar sind. Dies schließen insbesondere auch evolutionäre Verfahren ein [11, 7, 10, 19, 31]. Lösungen aus dem *Computer Aided Engineering* (CAE) schließlich ermöglichen ausschließlich die manuelle Platzierung der Komponenten mittels graphischer Oberflächen. Die verwendeten Routingalgorithmen liefern Abschätzungen der benötigten Rohrleitungslängen. Eine wissensbasierte Entwurfsunterstützung existiert nicht.

Das Finden einer optimalen Aufstellung ist also ein schwieriges Problem, welches bereits mit einer Vielzahl von Optimierungsmethoden bearbeitet wurde. Klassische Methoden zur Lösung des Facility Placement Problems umfassen die Auffassung als *quadratic assignment problem*, als Problem der linearen Programmierung und auch der gemischt-ganzzahligen linearen Programmierung [21]. Ein Manko dieser Entwürfe ist die Reduktion des Optimierungsproblems auf eine einzelne oder nur wenige Klassen von Kosten. Amorese [1] betrachtet bei seiner Optimierung nur die Rohrleitungskosten, während Malingriaux [25] weitere Kostenfaktoren (z.B. Betriebskosten, Bodenkosten, Montagekosten, etc.) hinzuzieht. Die exakte Bestimmung einer allgemein gültigen Kostenfunktion ist äußerst aufwändig [28].

Andere Ansätze schränken das Problem weiter ein. Der Ansatz von Georgiadis [15] verwendet ein festes Raster für die möglichen Aufstellungsorte. Die Unterschiedlichkeit der Komponenten erlaubt in diesem Fall offensichtlich keine optimale Aufstellung. Erst wenn eine praktisch freie Aufstellung der Komponenten gegeben ist, kann eine optimale Aufstellung erreicht werden.

Penteado und Ciric [27] betrachten bei der Optimierung zusätzliche Sicherheitsaspekte und werden damit der Komplexität des Problems des Aufstellungsentwurfs in Ansätzen gerecht. Die Laufzeit des Algorithmus ist aber für praktische Anwendungen nicht ausreichend.

Castell [3] betrachtet die gewichteten Kosten für Fläche, Rohrleitungen (Manhattan-Distanz) und Sicherheit. Die Gewichtung, die Kosten für jede Rohrleitung und der Sicherheitsabstand sind vom Benutzer vorzugeben. Als Repräsentation hat Castell einen Graphen gewählt, wobei die Knoten die Komponenten und die Kanten die Rohre modellieren. Als Selektion werden „Turnier-Selektion“ und „Roulette-Wheel“ verwendet. Einzelne Komponenten dürfen sich nicht überschneiden. Das Optimierungsziel minimale Fläche bei minimaler Rohrlänge widerspricht i.a. den Sicherheitsanforderungen. Durch Variation der Gewichtung der Sicherheitsanforderungen kann der Benutzer das Maß an Sicherheit der Anlage bestimmen. Das Programm besitzt eine einfache graphische Oberfläche mit welcher die Optimierung verfolgt und kontrolliert werden kann. Nicht berücksichtigt wird die Dreidimensionalität des Problems sowie weitere gewünschte *Constraints*. Andere evolutionäre Algorithmen zur Lösung des Aufstellungspro-

blems berücksichtigen nicht die in dieser Arbeit geforderten Kriterien [5, 33].

Von einem Programm für das Layoutproblem technischer Anlagen muss neben einer Optimierung der Kosten auch eine Einbeziehung aller Anforderungen aus 1.1 verlangt werden, was das Problem zum *Constraint Placement Problem* erweitert. Der dreidimensionale Suchraum muss beachtet werden, wobei sich eine zu grobe Rasterung verbietet. Externe Vorgaben (z.B. der Stahlbau, Platzierung bestimmter Komponenten, etc.) müssen vor der Optimierung möglich sein. Das sich ergebende multikriterielle Optimierungsproblem muss effizient gelöst werden.

Im nächsten Abschnitt formalisieren wir das in dieser Arbeit betrachtete Entwurfsproblem als Erweiterung des klassischen Facility Layout Problems. Zusätzlich wird eine Indexierung der Regelbasis für Designanforderungen vorgestellt, die eine effiziente Überprüfung der Erfüllungsgrade der (unscharfen) Anforderungen erlaubt. In Abschnitt 3 beschreiben wir einen genetischen Algorithmus auf dieser Repräsentation, der als Fitnessfunktion die vorgestellte Regelerfüllung benutzt. Die *Pareto dominant selection* unterstützt die Suche nach einer Pareto-Menge von Anlagenentwürfen. Abschnitt 4 stellt Ergebnisse des Verfahrens in Form von Anlagenentwürfen für eine real existierende Anlage vor.

## 2 Erweiterung des Anlagenentwurfsproblems

Unter dem Anlagenentwurfsproblem versteht man die Positionierung rechteckiger Anlagenteile innerhalb einer gegebenen Fläche [30]. Dabei sollen die durch die Flussmatrix  $M = [m_{ij}]_{i,j=1,\dots,n}$  induzierten Kosten minimiert werden. Diese Matrix definiert die Verbindungskosten zwischen zwei Komponenten  $i$  und  $j$ . Das Ziel ist eine nichtüberlappende Anordnung aller Komponenten mit minimalen Flusskosten  $\sum_{i,j} d_{ij} \cdot m_{ij}$  mit  $d_{ij}$  als Distanz zwischen den betreffenden Komponenten (siehe Abschnitt 2.1). Diese Formalisierung schließt sowohl das Problem des Anlagenentwurfs als auch des Design von Schaltungen (VLSI Design) ein. Es ist ein Spezialfall des *quadratic assignment problem*, welches sich als NP-hart erwiesen hat [14].

Wir betrachten eine Variante dieses Problems mit unregelmäßigen Grundflächen, mehreren Dimensionen durch Verwendung eines Stahlbaus, Zonen ohne Bebauung sowie vordefinierten Wegen, die ebenfalls nicht bebaut werden dürfen. Ähnliche Varianten wurden bereits in [21, 26] untersucht. Um diese zusätzlichen und weitere Anforderungen erfüllen zu können, wird eine Chemieanlage wie in Abbildung 2 beschrieben. Alle regelrelevanten Bestandteile stellen Komponenten der Anlage dar. Komponenten besitzen Ausdehnungen in allen Raumdimensionen und einen Ort  $(x,y,ebene)$ . Dabei sind sowohl Wege als auch blockierte Flächen besondere unbewegliche Komponenten, welche zur Definition der zu erfüllenden Regeln benötigt werden. Komponenten können darüberhinaus um ihr Zentroid rotiert werden. Sofern nicht anders erwähnt, bezeichnen wir im Folgenden mit Komponenten alle beweglichen Anlagenteile (Maschinen u.a.) einer Chemieanlage. Dieses Komponentenkonzept, die optionale Möglichkeit zur Stahlbau- und Wegedefinition und die freien Grundflächen in mehreren Ebenen

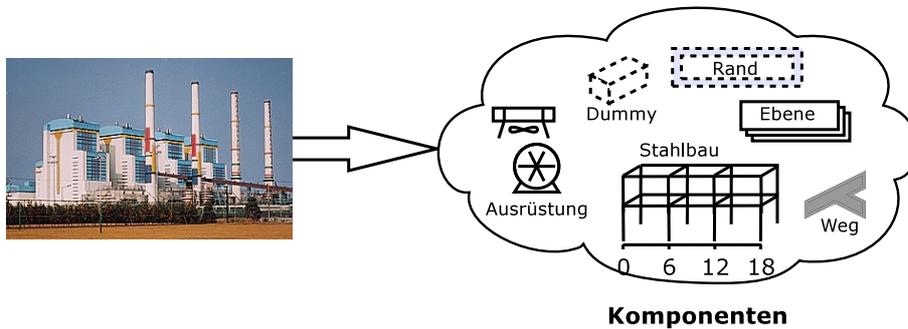


Abbildung 2: Aufbau einer technischen Anlage am Beispiel einer Chemiefabrik. Eine solche Anlage besteht aus mehreren Ebenen, Platzhaltern (dummy), Rändern, Wegen und einem Stahlbau. Zusätzlich werden auch sämtliche Ausrüstungsteile als Komponenten aufgefasst.

schließen sowohl die hier vorliegende komplexe Aufgabenstellung zur Chemieanlagenaufstellung als auch einfachere Anlagenentwurfsprobleme ein.

## 2.1 Routing und Distanzen

Zur Bestimmung der Verbindungskosten muss die Summe  $\sum_{i,j} d_{ij} \cdot m_{ij}$  über die Flussmatrix  $M$  gebildet werden. Hierzu müssen die Distanzen  $d_{ij}$  zwischen den Komponenten effizient berechnet werden. Das Routing Problem wird i.a. mit Variationen des Algorithmus von Lee [22] gelöst. Der Algorithmus teilt die Grundfläche in  $P \times Q$  Felder auf und besitzt eine Laufzeit und einen Platzbedarf von  $O(PQ)$  pro Verbindung. Schnellere Algorithmen garantieren keine kürzesten Wege mehr [4]. Einen performanteren Routing Algorithmus für dreidimensionales VLSI Routing wird in [35, 34] vorgestellt. Schon für zwei Schichten ist das 3D-Routing Problem jedoch NP-hart [34]. Dieses Problem wurde ebenfalls mit evolutionären Algorithmen bearbeitet [13, 12].

Weiter verschärft wird dieses Problem durch die Tatsache, dass bei jeder Positionsänderung einer Komponente alle Verbindungen zu den Komponenten neu berechnet werden müssen. Die Integration dieser Positionsänderungen in allgemeine Routing Algorithmen ist aufwändig, da bei Routing Algorithmen die Anschlusspunkte im allgemeinen fest sind. Damit die Optimierungszeit nicht zu stark vom Routing abhängt, muss eine einfache Approximation zur Längenabschätzung während der Aufstellungsoptimierung verwendet werden.

Beim VLSI Design wird am Häufigsten die *Semi-Perimeter*-Methode als Approximation des Routing Problems verwendet, da sie gute Ergebnisse liefert [32]. Dabei wird der halbe Umfang der minimalen *Bounding Box* berechnet, welche alle zu verbindenden Anschlusspunkte enthält. Bei Anwendung der Manhattan-Metrik ist die *Semi-Perimeter* Methode äquivalent zur Manhattan-Distanz, wel-

che das theoretische Minimum für die Rohrleitungslänge ist. Daher verwenden wir die Manhattan-Distanz als Annäherung an die tatsächliche Rohrlänge. Die Manhattan-Distanz ist die kürzeste orthogonale Verbindung zweier Punkte im Raum. Dabei werden Kollisionen nicht berücksichtigt. Durch die Dreiecksungleichung im euklidischen Raum entspricht eine Minimierung der Manhattan-Distanz immer auch einer Minimierung des euklidischen Abstands. Darüberhinaus werden durch die Manhattan-Distanz die real zu erwartenden Kosten besser approximiert, da Rohrleitungen im Anlagenbau fast ausschließlich orthogonal verlegt werden.

## 2.2 Regeln zum Anlagenentwurf

Die Grundlagen des Optimierungsproblems wurden bereits in Abschnitt 1.1 dargestellt. In diesem Abschnitt wird beschrieben, wie die unterschiedlichen betrieblichen Anforderungen formalisiert und bewertet werden. Am Lehrstuhl für Anlagentechnik der Universität Dortmund wurde für das Programm CAPD die Regelbasis zum Layoutentwurf einer Chemieanlage entwickelt [23]. Die Regeln lassen sich unabhängig von ihrer Herkunft in verschiedene Gruppen einteilen:

*Absolute Positionierungsanforderungen:* Absolute Regeln beziehen sich auf eine bewegliche Komponente und auf eine der Randbedingungen der Anlage wie z.B. Wege. Für einen Luftkühler gibt es aufgrund seiner Funktion die Anforderung, dass er in der obersten Etage stehen muss, wobei die absolute Position nicht von Bedeutung ist. Ähnliche Regeln gibt es für sehr große Komponenten, die in den Stahlbau eingehängt werden müssen. Die Anforderung ergibt sich hier aber nicht aufgrund der Funktion, sondern der Größe. Diese Komponenten sollten nahe des Randes aufgestellt werden, um einen Einbau mit einem mobilen Kran zu ermöglichen.

*Relative Positionierungsanforderungen:* Relative Regeln setzen zwei verbundene Komponenten der Anlage in Beziehung, ohne eine absolute Position zu bestimmen. So sollte z.B. ein Behälter, von dem aus eine Rohrleitung zu einer Pumpe führt, oberhalb der entsprechenden Pumpe stehen, damit an der Pumpe der erforderliche Druck anliegt.

Durch das komponentenbasierte Vorgehen und die Betrachtung aller Randbedingungen wie Stahlbauten, Wege und Ränder als Komponenten ist es möglich, beide Arten von Anforderungen innerhalb einer Funktionsklasse zu betrachten. Absolute Positionsanforderungen wie *neben dem Rand* oder *neben einem Weg* können auf diese Weise ebenfalls als relative Anforderungen zwischen einer Komponente und den Komponentenklassen *Rand* oder *Weg* betrachtet werden. Für beide Anforderungsgruppen können Regeln formuliert werden, die von den Schlüsselwörtern *neben*, *nahe*, *über*, *innerhalb*, *ausserhalb*, *in\_ebene* Gebrauch machen.

Die sich aus den Anforderungen ergebenden Regeln, haben nicht alle die gleiche Wichtigkeit. Es gibt Regeln, welche für das Funktionieren der Anlage oder

```

<rule name="L071" weight="muss" constraint="Border">
  <method name="getType" type="=" value="kolonne"/>
  <method name="getMinLength" type=">" value="500"/>
  <method name="getHeightDivByMinLength" type="<" value="15"/>
</rule>
<rule name="L080" weight="soll" constraint="Ground">
  <method name="getType" type="=" value="leitstand"/>
</rule>
<rule name="L005" weight="soll" constraint="Way">
  <method name="getMaintDev" type="=" value="gabelhubwagen"/>
  <method name="getMaxLength" type="<" value="5000"/>
</rule>
...

```

Abbildung 3: Ausschnitt aus einer XML Regelbasis für absolute Positionierungsanforderungen. Border ist z.B. eine Abkürzung für die Anforderung, neben einer beliebigen Randkomponente zu stehen, d.h. das Maximum der Erfüllungsgrade für alle Randkomponenten.

aus Sicherheitsgründen eingehalten werden müssen, und solche, welche nur untergeordnete Wünsche (z.B. an das Design oder die Kosten) realisieren. Die Gewichtung einer Regel wird über die Schlüsselwörter *muss*, *soll*, *sollte* ausgedrückt, wobei *muss* die maximale Wichtigkeit ausdrückt und *sollte* eine untergeordnete Wichtigkeit beschreibt. In Abschnitt 2.2.2 und 2.2.3 beschreiben wir zwei unterschiedliche Ansätze, welche die Erfüllung solcher Anforderungen unter Berücksichtigung der Gewichtungen bewerten. Die Abbildungen 3 und 4 zeigen einige Beispiele für Regeln zum Anlagenentwurf einer Chemieanlage, welche in einem XML Format abgelegt wurden.

Es ist nicht immer sinnvoll, eine Komponente automatisch zu platzieren. Einige Komponenten müssen aufgrund der Gegebenheiten vor Ort an festen Plätzen aufgestellt werden. Die Lage dieser Komponenten wird während des Anlagenentwurfs nicht geändert. In ähnlicher Weise sind auch Wege vorgegeben und ihre Lage ist nicht Gegenstand der Optimierung. Wie oben bereits angedeutet wurde, sind Wege Bezugspunkt für einige Regeln. So wird z.B. versucht alle mit einem Gabelhubwagen zu bedienenden Komponenten an einem Weg zu platzieren. In gleicher Weise können sich Regeln auf den Stahlbau oder den Rand einer Anlage beziehen.

### 2.2.1 Vorverarbeitung der Regelmenge und Indexierung für eine gegebene Anlage

Die Regelbasis ist in Form von Datenbanktabellen gegeben, aus denen die oben gezeigten Regeln im XML-Format generiert werden. Diese allgemeinen Regeln, die grundsätzlich auf den Bau von Chemieanlagen anwendbar sind, müssen jedoch nicht alle für eine konkret gegebene Anlage zutreffen. Der naive Ansatz,

```

<rule name="L040" weight="sollte" constraint="Above">
  <first>
    <method name="getType" type="" value="behaelter"/>
    <method name="getSpecs" type="" value="waschbehaelter"/>
  </first>
  <second>
    <method name="getType" type="" value="zentrifuge"/>
  </second>
</rule>
<rule name="L002/L003/L008" weight="muss" constraint="CloseTo">
  <connection>
    <method name="getDiameter" type="" value="150"/>
    <method name="getMaterial" type="" value="carbon_steel"/>
  </connection>
</rule>
...

```

Abbildung 4: Ausschnitt aus einer XML Regelbasis für relative Positionierungsanforderungen. Es können beliebige Kombinationen von Prämissen für die beiden Verbindungspartner als auch für die Verbindung selbst angegeben werden.

während der Optimierung zur Evaluierung der Regelerfüllung durch alle Regeln zu iterieren und ihre Anwendbarkeit für alle Komponenten und Verbindungen zu prüfen, verbietet sich aus Laufzeitgründen. Wir wenden daher vor der Optimierung einen Vorverarbeitungsschritt an, der einmal die Prämissen der Regeln für alle Komponenten und Verbindungen prüft und die Konklusionen aller Regeln, deren Prämissen erfüllt sind, den betreffenden Komponenten bzw. Verbindungen zuordnet. Durch diese Indexierung der Regelbasis für eine konkret gegebene Anlage kann der Aufwand der Regelüberprüfung auf die Berechnung des Erfüllungsgrades der überhaupt nur in Frage kommenden Konklusionen reduziert werden. Damit braucht nur eine Teilmenge der allgemeinen Regelbasis überprüft werden, deren Prämissen zudem bereits *vor* der Optimierung ausgewertet wurden.

Wir illustrieren dieses Vorgehen an einem Beispiel. Die allgemeine Regelbasis verwendet Komponententypen (z.B. Pumpe, Wärmetauscher oder Behälter), Funktionen, Spezifikationen der Komponenten sowie Bedien- und Wartungshilfsmittel zum Ausdrücken von Anforderungen. Wir betrachten eine Regel der Regelbasis:

Wenn eine Verbindung existiert  
 von einem Wärmetauscher zu einer Pumpe,  
 dann sollte der Wärmetauscher über der Pumpe stehen.

Wenn in der zu planenden Anlage keine Wärmetauscher und Pumpen existieren, braucht die Erfüllung dieser Regel im Rahmen der automatisierten Optimierung

niemals überprüft zu werden. Existieren jedoch in der zu planenden Anlage sowohl Wärmetauscher als auch Pumpen und sind diese in entsprechender Richtung miteinander verbunden, so braucht die Erfüllung der Regel nur für diese Verbindungen überprüft zu werden. Wenn also zwischen einem Wärmetauscher  $W$  und einer Pumpe  $P$  in der Anlage eine Rohrverbindung existiert, so wird folgende konkrete Regel (Anforderung) abgeleitet:

Der Wärmetauscher  $W$  sollte über der Pumpe  $P$  stehen.

Für Verbindungen andersartiger Komponenten braucht diese Regel ebenfalls nie überprüft zu werden. Durch die Indexierung der Regeln und der Ableitung der Anforderungen aus der allgemeinen Regelbasis wird der Aufwand der Regelüberprüfung also auf das Nötigste reduziert.

### 2.2.2 Scharfe Regelerfüllung durch Bereichsquader um Komponenten

Der Test, ob eine Regel erfüllt ist, lässt sich für alle Regeln auf einen einfachen Kollisionstest im Raum reduzieren. Dabei werden alle Komponenten, auch die Wege und Ränder, als Volumenkörper modelliert. Für bewegliche Komponenten wird entsprechend der Relation *nahe*, *neben* und *über* ein individuell erweitertes Volumenmodell betrachtet. Eine Bedingung wie z.B. *neben dem Rand* ist erfüllt, wenn das erweiterte Volumenmodell einer Komponente mit dem Volumenmodell des Randes kollidiert. Dies ist ein sehr performantes Vorgehen. Während der Generationenfolge des genetischen Algorithmus kann der erweiterte Volumenkörper einer Komponente verkleinert werden, um eine höhere Qualität der Optimierung zu erreichen.

### 2.2.3 Erfüllungsgrad unscharfer Regeln durch Fuzzy-Konklusionen

Alternativ zu einer festen Grenzenangabe, bei der beim Überschreiten eines gegebenen Schwellwertes eine Regel sprunghaft nicht mehr erfüllt ist, bieten sich zum Erfassen des Erfüllungsgrades der Regelmenge auch Techniken der Fuzzy-Logik an [9]. Um im Rahmen der Fitnessbestimmung für evolutionäre Algorithmen eine gleichmäßige Bewegung in Richtung eines Optimums zu erlauben, sind stetige Fitnessfunktionen zudem besonders wünschenswert. Die unscharfen Erfüllungsgrade tragen dazu bei, die Fitnessfunktion zu glätten. In Anlehnung an eine linguistische Variable können zweistellige Funktionen definiert werden, die auf einen Zugehörigkeitsgrad zu einem bestimmten Konzept wie Nachbarschaft oder vertikale Anordnung abbilden. Wir definieren diese Abbildung eines Komponentenpaares auf ein Zugehörigkeitsintervall als:

$$f : K \times K \rightarrow [0, 1]$$

$K$  ist hierbei die Menge der Komponenten mit  $|K| = n$ . Diese *Fuzzy-Bedingungen* können dann als Konklusion von Regeln verwendet werden. Ein weiterer Vorteil dieser Fuzzy-Bedingungsfunktionen ist die intuitiv mögliche Gewichtung durch Skalierung mit dem Regelgewicht. Die Zugehörigkeit der Komponenten  $k_i$  und

$k_j$  zu der Konklusion  $f_r$  bzw. der korrespondierende Erfüllungsgrad  $e$  berechnet sich für eine Regel  $r$  mit Gewicht  $w_r \in \mathbb{R}$  als:

$$e_r(k_i, k_j) = w_r \cdot f_r(k_i, k_j)$$

Um auch absolute Ortspositionierungen, wie z.B. die Forderung der Nachbarschaft einer Komponente zu einem beliebigen Weg, erfüllen zu können, lassen wir als Variante dieser Abbildungen jeweils auch  $f : K \times K^p \rightarrow [0, 1]$  zu. Hierbei ist diese Abbildung als Maximum der Zugehörigkeitsgrade der Komponente zu einer der Komponenten in der gegebenen Menge zu berechnen.

Im Folgenden beschreiben wir die Definition der Konklusionsbedingungen im Detail. Aufgrund verfahrenstechnischer Anforderungen ist häufig eine vertikale Anordnung über einer Komponente gefordert. Da es hierbei nicht auf eine genau senkrechte Ausrichtung ankommt, bildet die Fuzzy-Bedingung über in Abhängigkeit vom durch die Komponenten eingeschlossenen Winkel auf das Intervall  $[0, 1]$  ab. Es sei  $d_{ij}$  der Abstand und  $h_{ij}$  der Höhenunterschied zwischen der Komponente  $i$  und Komponente  $j$ . Dieser Wert ist negativ, falls Komponente  $i$  unter Komponente  $j$  liegt. Es ist der Wert der Fuzzy-Bedingung über:

$$ueber(i, j) = \begin{cases} 0 & \text{falls } h_{ij} < 0 \\ \max\left(0, 1 - \frac{\arctan \frac{d_{ij}}{h_{ij}}}{\alpha_{max}}\right) & \text{sonst} \end{cases}$$

Für alle Winkel größer  $\alpha_{max}$  liefert die Funktion ebenfalls 0.

Für eine geforderte Nebeneinanderanordnung (*neben*) zweier Komponenten in der gleichen Ebene kann der Abstand  $d_{ij}$  der Komponenten mit der Flächendiagonale  $d_2$  der entsprechenden Ebene normiert werden:

$$neben(i, j) = \begin{cases} 0 & \text{falls } h_{ij} \neq 0 \\ \frac{d_{ij}}{d_2} & \text{sonst} \end{cases}$$

In ähnlicher Weise liefert die Definition von *nahe* die Ausprägung der Nachbarschaft in allen drei Raumdimensionen für die Raumdiagonale  $d_3$ :

$$nahe(i, j) = \frac{d_{ij}}{d_3}$$

Die nachfolgenden Fuzzy-Bedingungen finden ausschließlich für absolute Positionierungsanforderungen im Rahmen der von uns betrachteten Chemieanlagen Anwendung. In ähnlicher Weise können jedoch auch weitere Konklusionserfüllungsfunktionen definiert werden. Um eine Einordnung von Komponenten in bestimmte Ebenen oder den Stahlbau zu fordern bzw. zu verhindern, wurden die folgenden Fuzzy-Bedingungen konstruiert:

$$innerhalb(i, j) = o(i, j) / \min(a(i), a(j))$$

$$ausserhalb(i, j) = 1 - innerhalb(i, j)$$

Dabei berechnet  $a(i)$  die Grundfläche der Komponente  $i$  und  $o(i, j)$  die Größe der überlappenden Flächen der Komponenten. Der Wertebereich liegt zwischen 0 und dem Minimum der Flächen der beiden Komponenten.

Für einige Komponenten existieren Regeln, die eine Anordnung in einer bestimmten Ebene verlangen. Da Ebenen ebenfalls Komponenten sind, kann mit Hilfe der Höhe der Anlage  $h$  die folgende Fuzzy-Bedingung `in_ebene` definiert werden:

$$\text{in\_ebene}(i, j) = 1 - \frac{h_{ij}}{h}$$

#### 2.2.4 Zahl der Regeln pro Anlage und Aufwand der Regelüberprüfung

Bei  $n$  Komponenten ist im schlimmsten Fall davon auszugehen, dass jede Komponente mit jeder anderen verbunden ist, also  $O(n^2)$  Verbindungen existieren. Da die Überprüfung der Prämissen und die Zuordnung zu den Verbindungen im Vorfeld durchgeführt wird, ist eine Berechnung des Erfüllungsgrades in  $O(1)$  möglich. Weiterhin existiert nur eine konstante Zahl  $k$  möglicher Konklusionen für jede Verbindung, in unserem Fall die sechs Konklusionen `neben`, `nahe`, `über`, `innerhalb`, `außerhalb`, `in_ebene`. Zu guter Letzt ergibt sich noch ein weiterer additiver Faktor  $n$  durch die Prüfung der Ortsbedingungen, d.h. den relativen Bedingungen zu unbeweglichen Komponenten. Im *worst case* ergibt sich also eine Laufzeit von  $O(kn^2 + n)$ . In der Praxis zeigt sich jedoch, dass nicht alle Komponenten untereinander verbunden sind, sondern jede Komponente nur eine kleine Anzahl von Verbindungen besitzt. Bei durchschnittlich  $\hat{m}$  Verbindungen pro Komponente ergibt sich für die *average case* Analyse die lineare Laufzeit  $O(k(\hat{m} + 1)n)$  in der Zahl der Komponenten. Für die in Abschnitt 4 betrachtete Anlage zur Gasbehandlung beträgt die durchschnittliche Zahl der Verbindungen und damit der lineare Faktor  $\hat{m} \approx 3,9$ .

### 2.3 Anlagenentwurf als multikriterielles Optimierungsproblem

Die im vorigen Abschnitt beschriebenen Regeln decken bereits einen großen Teil der in Abschnitt 1.1 beschriebenen betrieblichen Anforderungen ab. Darüberhinaus sind jedoch noch andere Kriterien definierbar, die in dieser oder ähnlicher Form für die Konstruktion von Anlagen eine wesentliche Rolle spielen. Allen betrachteten Bewertungsmaßen gemein ist, dass sie in maximal polynomieller Zeit bestimmt werden können. Wir definieren als Bewertungskriterien für den Anlagenentwurf:

$f_a$ : der scharf oder weich berechnete gewichtete Erfüllungsgrad der absoluten Positionsanforderungen. Die Berechnung ist im Durchschnitt in linearer Zeit in der Zahl der Komponenten möglich.

$f_r$ : der scharf oder weich berechnete gewichtete Erfüllungsgrad der relativen Positionsanforderungen. Die Berechnung ist im Durchschnitt in linearer Zeit in der Zahl der Verbindungen möglich.

- $f_e$ : der Grad der Ebenenfüllung. Dieser sollte für eine verbesserte Wartbarkeit und Zugänglichkeit im Intervall zwischen 30% und 70% liegen. Tatsächliche Belegungsmaße sind vom Typ einer Anlage abhängig [20]. Da alle Komponenten einmal betrachtet werden müssen, beträgt die Laufzeit  $O(n)$ .
- $f_c$ : Verbindungskosten mittels Manhattan-Distanz und Flussmatrix (siehe Abschnitt 2.1). Die quadratische Laufzeit im schlimmsten Fall wird für reale Anlagen praktisch nie erreicht. Im Durchschnitt ist der Aufwand ebenso hoch wie die Bestimmung der Regelerfüllung.
- $f_u$ : der Grad der Überlappung. Eine Relaxierung erleichtert das Problem und erlaubt auch kontinuierliche Verschiebungen in andere Zusammenhangskomponenten des Anlagengraphen über verbotene Flächen oder Wege hinaus. Da die Überlappung jeder Komponente zu allen anderen Komponenten überprüft werden muss, ist die Laufzeit  $O(n^2)$ .

Diese fünf Funktionen können auf einen Wertebereich  $[0, 1]$  normiert werden. Es ist leicht einzusehen, dass die Kriterien im allgemeinen konkurrieren. Eine Anlage, die sämtliche Regeln erfüllt wird Kompromisse in den Verbindungskosten eingehen müssen. Eine stark überlappende Anlage würde eben diese Verbindungskosten minimieren. Ein erfahrener Konstrukteur ist unter Umständen dazu in der Lage, die Kriterien gegeneinander abzuwiegen und für eine konkrete Anlage und ihrer Anforderungen Gewichte  $w_a, w_r, w_u, w_c$  und  $w_e$  zu definieren. Auf diese Weise könnte die Zielfunktion

$$Z = w_a f_a + w_r f_r + w_u f_u + w_c f_c + w_e f_e$$

maximiert werden. Die Anlage, welche diese Funktion maximiert, ist optimal. Erfahrungsgemäß gelingt es jedoch nicht, eine Gewichtung dieser Bewertungskriterien für eine Anlage anzugeben. Darüberhinaus ergeben sich im allgemeinen unterschiedliche Gewichtungen für unterschiedliche Anlagen und Anforderungen. Ein solches Optimierungsproblem mit mehreren konkurrierenden Bewertungskriterien heißt *multikriterielles Optimierungsproblem*.

Die vorliegende Problemstellung ist also derart, dass bezüglich mehrerer in Konflikt stehender Zielgrößen eine Optimierung durchgeführt werden soll. Solche Problemstellungen können mit normalen evolutionären Verfahren nicht gelöst werden. Da der Anwender nur eine vage Vorstellung von der Gewichtung der verschiedenen Gütekriterien zueinander hat, versucht man bei der multikriteriellen Optimierung, alle Lösungen, die bei irgendeiner denkbaren Gewichtung der Kriterien optimal wären, zu finden. Diese nennt man *Pareto-optimale* Lösungen. Diese Lösungsmenge zeichnet sich dadurch aus, dass sie Lösungen enthält, bei denen die Verbesserung eines Zielkriteriums nur durch eine Verschlechterung eines anderen Zielkriteriums zu erreichen ist. In Abschnitt 3.4 beschreiben wir die Suche nach solchen Pareto-optimalen Anlagenentwürfen mit Hilfe eines evolutionären Algorithmus.

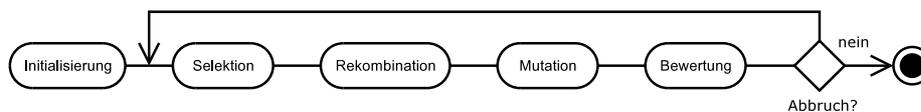


Abbildung 5: Allgemeiner Ablauf eines evolutionären Algorithmus.

### 3 Genetischer Algorithmus zum Finden Pareto- optimaler Anlagenentwürfe

Evolutionäre Algorithmen (EA) sind eine Teilklasse heuristischer Suchverfahren. Ihr Einsatz hat sich für komplexe Zielfunktion bewährt, für die die Entwicklung eines speziellen Suchverfahrens zu schwierig ist oder nicht genug Wissen über die Optimierung vorhanden ist. Diese allgemeinen Suchverfahren sind auch bei der sogenannten *Black-Box Optimierung* anwendbar, bei der die Zielfunktion durch ein Experiment bestimmt werden muss. EAs bestehen aus fünf wesentlichen Schritten, die in Abbildung 5 dargestellt werden [16, 17].

Im Folgenden wenden wir EAs zur Konstruktion einer Chemieanlage an. Wir legen dabei die oben beschriebene Repräsentation und Evaluierungsfunktion zugrunde. In den nächsten Abschnitten beschreiben wir die einzelnen Schritte des EAs im Detail.

#### 3.1 Initialisierung und Startpopulation

In der Initialisierungsphase werden  $N$  zufällige Lösungen erzeugt (Startpopulation).  $N$  heißt Populationsgröße, jedes Element dieser Menge heißt Individuum. Für die Erzeugung einer Startpopulation von Chemieanlagen ist zu bedenken, dass eine zufällig erzeugte Anordnung der Komponenten mit hoher Wahrscheinlichkeit Überlappung aufweist. Um die Anzahl der Startindividuen, die wegen Überlappung verworfen werden müssen, zu reduzieren, erweist sich auch hier die oben beschriebene Überlappungsrelaxation als sinnvoll. Ansonsten sollten die Komponenten mit abnehmender Größe zufällig angeordnet werden.

#### 3.2 Rekombination und Mutationen

Bei der Rekombination werden durch Mischen aus mehreren (i.a. zwei) Individuen neue Individuen zufällig erzeugt (Kreuzung). Im Falle der Chemieanlagen wird eine Kreuzung durch den Austausch der Koordinaten und der Ausrichtung der beweglichen Komponenten realisiert. Hierzu wird für jede Komponente zufällig entschieden, ob ein Austausch mit der betreffenden Komponente des anderen Elter erfolgen soll (*uniform crossover*).

Bei der Mutation wird ein Individuum mindestens einmal zufällig verändert. Die Wahrscheinlichkeit für kleine Änderungen soll dabei größer sein als die Wahrscheinlichkeit für starke Änderungen des Individuums. Auf der gewählten Repräsentation der Chemieanlagen wird zu diesem Zweck bei Individuen mit

$k$  beweglichen Komponenten für jede der Komponenten mit Wahrscheinlichkeit  $1/k$  eine der folgenden Mutationen durchgeführt:

*Verschiebung*: Die betreffende Komponente wird in eine zufällige Richtung mit Standardabweichung  $\sigma$  verschoben. Diese wird im Verlauf der Evolution mittels der 1/5-Regel adaptiert [29].

*Sprung innerhalb der Ebene*: Die betreffende Komponente wird an eine zufällige Stelle innerhalb der gleichen Ebene versetzt (nicht in Kombination mit Verschiebung und Crossover verwendet).

*Ebenenwechsel*: Die betreffende Komponente wird an eine zufällig gewählte Position einer anderen Ebene des Stahlbaus versetzt.

*Rotation*: Für die betreffende Komponente wird eine Rotation um  $90^\circ$ ,  $180^\circ$  oder  $270^\circ$  durchgeführt.

### 3.3 Evaluierung der Fitness und Selektion

Auf die Individuen der aktuellen Population wird die Zielfunktion angewendet. Diese wird auch als Fitnessfunktion bezeichnet.

In der Selektionsphase werden  $M$  Individuen aus der Population ausgewählt. Ist eine Gewichtung der Evaluierungskriterien bekannt, so können die Selektionswahrscheinlichkeiten der Individuen proportional zu ihrer Fitness gebildet werden (*Roulette Wheel*). Wenn  $Z$  die Bewertungsfunktion ist, dann wird ein Individuum  $c_i$  mit Wahrscheinlichkeit

$$Z(c_i) \cdot \left( \sum_{j=1}^N Z(c_j) \right)^{-1}$$

aus der Population mit  $N$  Elementen ausgewählt. Eine weit verbreitete Variante der Selektion ist die Turnier-Selektion. Dabei werden die besten  $m$  Individuen aus einer zufälligen Auswahl von  $t$  Individuen selektiert ( $t \geq m$ ), bis die resultierende Population die Größe  $M$  erreicht. Eine Abwandlung dieser Turnierselktion ist die hier verwendete *Pareto dominant selection*, welche im nächsten Abschnitt beschrieben wird.

Im Allgemeinen enthält die Nachfolgeneration erneut  $N$  Individuen. Sollte die neue Population das optimale Ergebnis enthalten, der Optimierungsschritt nicht mehr ausreichend sein, oder eine vorgegebene Anzahl von Generationen überschritten sein, wird der Algorithmus abgebrochen. Ansonsten beginnt der Algorithmus erneut mit der Selektionsphase.

### 3.4 Pareto dominierende Selektion

Die Selektion der Individuen innerhalb einer Population geschieht anhand eines skalaren Fitnesswertes, bezüglich dessen optimiert wird. Multikriterielle evolutionäre Algorithmen bieten durch Einführung spezieller Selektionsoperatoren

die Möglichkeit, mehrere Zielgrößen gleichzeitig zu optimieren [6, 36]. Durch die populationsbasierte Vorgehensweise wird innerhalb eines Durchlaufs eine breite Auswahl Pareto-optimaler Lösungen gefunden, aus der der Anwender anschließend wählen kann.

Bei der einkriteriellen Optimierung ist das Ziel die Konvergenz zu einem einzigen globalen Optimum. Randomisierte heuristische Verfahren wie evolutionäre Algorithmen versuchen dem Steckenbleiben in lokalen Optima durch geschicktes Auswiegen von Exploration neuer Suchgebiete und dem Weiterverfolgen bisher gegangener Suchpfade entgegenzusteuern. Wenn das Ziel nicht nur ein globales sondern mehrere verschiedene lokale Optima ist, muss die totale Konvergenz aller betrachteten Suchpunkte zu dem einen globalen Optimum verhindert werden. Stattdessen ist eine Diversität der Suchpunkte bezüglich verschiedener Peaks in der Peaklandschaft der Gütekriterien gewünscht. Im Bereich der evolutionären Algorithmen spricht man von *Nischenbildung* [24].

Für den vorliegenden Fall des Anlagenentwurfs mittels multikriterieller evolutionärer Optimierung haben wir den Selektionsansatz *Pareto domination tournament* adaptiert [18]. Dieser Selektionsansatz versucht basierend auf der Ähnlichkeit der Individuen eine möglichst große Diversität der Population aufrecht zu erhalten. Zwei Konstruktionsentwürfe (Individuen) werden zufällig aus der aktuellen Population ausgewählt. Diese werden mit einer Vergleichsmenge bestehend aus  $k$  anderen Individuen aus der Population verglichen. Wird eines der beiden gezogenen Individuen von dieser Vergleichsmenge dominiert und das andere nicht, so gewinnt das letztere dieses Turnier und wird der nächsten Population hinzugefügt. Dominiert wird eine Anlage von einer Menge anderer Anlagen genau dann, wenn in dieser Menge eine Anlage existiert, die in Bezug auf alle Bewertungskriterien besser ist. Dieses Vorgehen wird iteriert bis die gewünschte Populationsgröße wiederhergestellt ist.

Interessant ist der Fall, in dem entweder beide Anlagen dominiert werden oder beide Anlagen nicht dominiert werden. In diesem Fall ist das Ziel nicht eine Bewegung in Richtung der Pareto-Front sondern entlang der Front, d.h. eine Streuung der Individuen. Um dieses Ziel zu erreichen, wird die Fitness einer Anlage in stark bevölkerten Nischen reduziert, d.h. die Fitness einer Anlage mit vielen Nachbarn wird reduziert.

Sei  $P$  die aktuelle Population. Eine Abschätzung der Bevölkerungsdichte im Umkreis des  $i$ -ten Individuums kann berechnet werden mittels

$$m_i = \sum_{j \in P} s(d(i, j))$$

Dabei ist  $d(i, j)$  eine Abstandsfunktion zwischen zwei Anlagen  $i$  und  $j$ . Sie wird berechnet als durchschnittlicher Abstand der Komponenten der Anlagen. Die Abbildung  $s : R \rightarrow [0, 1]$  ist die sogenannte *sharing* Funktion, welche den Abstand zu einer anderen Anlage auf einen Grad von Nachbarschaftlichkeit abbildet. Sie muss monoton fallend sein und  $s(0) = 1$  erfüllen. In unserem Fall ist

$$s(d) = \begin{cases} 1 - d/\sigma_{share} & \text{für } d < \sigma_{share} \\ 0 & \text{sonst} \end{cases}$$

Zur Reduktion der Fitness in Nischen wird im Falle eines Gleichstands bei der Dominanzüberprüfung mit der Vergleichsmenge die Anlage gewählt, die einen kleineren Wert für  $m_i$  besitzt und sich damit in einer weniger stark bevölkerten Nische befindet.

Der Parameter  $\sigma_{share}$  definiert die Größe der Nische und kann geschätzt werden aus der minimal geforderten Separierung der Anlagen in der Pareto-Menge. Die Größe der Vergleichsmenge  $k$  definiert den Selektionsdruck. Je größer der Wert von  $k$  ist, desto wahrscheinlicher wird die Selektion der besten Anlage in jeder Population und auch die Konvergenz. Allerdings steigt damit auch das Risiko, in lokalen Extrema zu verweilen.

### 3.5 Größenabschätzung des Suchraums

Im Folgenden soll die Größe des Suchraums abgeschätzt werden. Idealerweise können Komponenten beliebig platziert werden, d.h. der Suchraum ist unendlich. Wir wollen im Folgenden die Auswirkungen einer Rasterung auf die Größe des Suchraums analysieren. Sei  $d$  die Kantenlänge zwischen zwei Rasterpunkten, dann wird das gegebene Baufeld der Fläche  $A$  in  $R = \frac{A}{d^2}$  Rasterpunkte aufgeteilt. Ferner sei  $n$  die Anzahl der Komponenten. Wird davon ausgegangen, dass Überschneidungen von Komponenten möglich sind (aber bei der Optimierung bestraft werden), ergibt sich die Anzahl  $k$  der möglichen Anlagen als:

$$k = \binom{\frac{A}{d^2}}{n} = \frac{\frac{A}{d^2}!}{n! (\frac{A}{d^2} - n)!} \quad \text{mit } \frac{A}{d^2} \geq n$$

Für eine kleine Anlage mit  $50m^2$  Fläche, einem Raster von  $1cm^2$  und 30 Komponenten ergibt sich als Größe des Suchraums:

$$k = \binom{25.000.000}{30} \approx 3,27^{189}$$

Die Erfahrung zeigt, dass sich aus der Anwendung der Regeln deutlich weniger Aufstellungsorte für eine Komponente ergeben. Sei  $\hat{r}$  die mittlere Anzahl der Aufstellungsorte aller Komponenten. Dann gibt es  $\hat{r}^n$  mögliche Aufstellungsorte. Schon für  $\hat{r} = 5$  existieren ungefähr  $10^{20}$  mögliche Anlagen. Dies zeigt, dass trotz des Einsatzes einer Rasterung der Suchraum zu groß ist und eine Aufzählung aller Lösungen nicht gelingen kann.

## 4 Experimente und Ergebnisse

Die hier vorgestellten Experimente wurden auf Basis einer real existierenden Anlage zur Gasbehandlung durchgeführt, die im Folgenden mit *original* bezeichnet wird. Auf diese Weise ist ein Vergleich mit dem Vorgehen eines Anlagendesigners möglich. Tabelle 1 fasst die Kennzahlen dieser Anlage zusammen. Die 66 Designregeln hätten für jede Komponente (absolute Positionierungsanforderung) und für jede Verbindung (relative Positionierungsanforderung) überprüft werden

Anzahl der Komponenten	28
vorgegebene Platzierungen	5
freie Platzierungen	23
Anzahl der Verbindungen	108
Ebenen	4
Stahlbau	ja
verbotene Zonen	ja
Wege	ja
Regeln insgesamt	66 (8976)
zutreffende Regeln	141

Tabelle 1: Die Kennzahlen der Originalanlage, die mit dem hier vorgestellten Verfahren ebenfalls geplant wurde. Von den  $66 \cdot (28 + 108) = 8976$  Regeln brauchten durch die Regelindexierung für jedes Individuum nur 141 Konklusionserfüllungsgrade überprüft zu werden.

müssen, insgesamt existieren also für die Originalanlage  $(28 + 108) \cdot 66 = 8976$  Regeln. Durch die vorherige Indexierung der Regelbasis und Prüfung der Prämissen konnte die Anzahl der Konklusionen auf 141 reduziert werden (1,5%).

Tabelle 2 zeigt die Erfüllungsgrade der einzelnen Kriterien für die real gebaute Anlage. Diese wurden für die Originalanlage in exakt der gleichen Weise bestimmt wie die Berechnung während der Optimierung. Aus diesen Erfüllungsgraden wurden Gewichte bestimmt für eine gewichtete Optimierung durch Bilden des Anteils eines Kriteriums an der Summe der Kriterien. Diese Gewichte wurden später im Rahmen der Optimierung mittels des GA für die Berechnung des gewichteten Durchschnitts angewendet.

#### 4.1 Evolutionäre Optimierung einer Anlage zur Gasbehandlung

Für die Startpopulation wurden  $k$  Anlagen zufällig generiert, indem die Positionen und die Ebenen der beweglichen Komponenten gleichverteilt bestimmt wurden. Die Aufstellungsplanung wurde dann mit dem in Abschnitt 3 vorgestellten Verfahren optimiert. Vier verschiedene Experimente wurden durchgeführt:

1. genetischer Algorithmus mit Populationsgröße 1 und ohne Kreuzungen, Kriterien gleich stark gewichtet
2. genetischer Algorithmus mit Populationsgröße 1 und ohne Kreuzungen, Kriterien gewichtet nach Originalanlage
3. genetischer Algorithmus mit Populationsgröße 10 und Kreuzungen, Kriterien gewichtet nach Originalanlage
4. multikriterieller genetischer Algorithmus mit Populationsgröße 30 und Kreuzungen

Anforderung	Erfüllungsgrad	Gewicht
absolute Positionsanforderungen	0.67	0.21
relative Positionsanforderungen	0.77	0.22
Grad der Ebenenfüllung	0.50	0.18
Verbindungslänge	0.79	0.23
Grad der Überlappung	0.32	0.16

Tabelle 2: Erfüllungsgrade der betrachteten Kriterien für die betrachtete Originalanlage. Die Gewichte entsprechen dem Quotienten aus dem Wert des Kriteriums und der Summe aller Kriterien.

	Original	p=1/w=1	p=1/w=o	p=10/w=o	p=30
abs. Pos.	0.67	0.69 (8)	0.68 (9)	<b>0.73</b> (4)	0.72 (5)
rel. Pos.	0.77	0.79 (0)	0.79 (0)	<b>0.82</b> (1)	0.80 (1)
Ebenen	0.50	0.55	<b>0.58</b>	0.58	0.57
Länge	0.79	0.84	<b>0.84</b>	0.84	0.83
Überl.	0.32	<b>0.80</b>	0.78	0.69	0.79
Mittel	0.63	0.74	<b>0.75</b>	0.74	0.75
Zeit	—	1024 s	983 s	11245 s	32132 s

Tabelle 3: Ergebnisse der Optimierung für einzelne Kriterien im Rahmen verschiedener Experimente. Die Angabe  $p$  beschreibt die Populationsgröße. Bei einer Größe von 1 wurde auf Kreuzungen verzichtet. Die Gewichte sind entweder alle 1 oder aus der Bewertung der Originalanlage bestimmt ( $w = o$ ). Für die multikriterielle Optimierung ( $p = 30$ ) wurde eine der Anlagen ausgesucht, die die Originalanlage dominieren. Die besten Werte jeder Zeile sind fett hervorgehoben, die Zahlen in Klammern geben die Anzahl der *Muss*-Regeln an, die vollständig verletzt wurden.

Sämtliche Läufe wurden mit 10000 Generationen durchgeführt. Tabelle 3 zeigt die Ergebnisse für die Experimente auf Basis eines skalaren Fitnesswertes. Mit allen Ansätzen ist es gelungen, Anlagen zu konstruieren, die die Originalanlage dominieren. Für die Experimente mit gewichteter skalarer Fitness konnte beobachtet werden, dass eine Verwendung von Populationen und Kreuzungen trotz erhöhter Laufzeit keinen Zugewinn in der Güte der Kriterien erbracht hat. Im multikriteriellen Fall wurden die Werte der Kriterien für eine die Originalanlage dominierende Anlage aufgezeichnet. Eine Überprüfung der Pareto-Menge und die Auswahl der optimalen Anlage kann durch einen Experten erfolgen. Das Konvergenzverhalten der Optimierung ist in Abbildung 6 dargestellt. Es weist den typischen exponentiellen Kurvenverlauf auf.

Abbildung 7 zeigt links eine 3D-Darstellung der Anlage wie sie im Original gebaut wurde. Der rechte Teil der Abbildung zeigt den Konstruktionsvorschlag unseres Ansatzes. Diese Anlage entstand in einem Optimierungslauf mit einem Individuum und 10000 Generationen. Ästhetische Aspekte, wie die Übereinan-

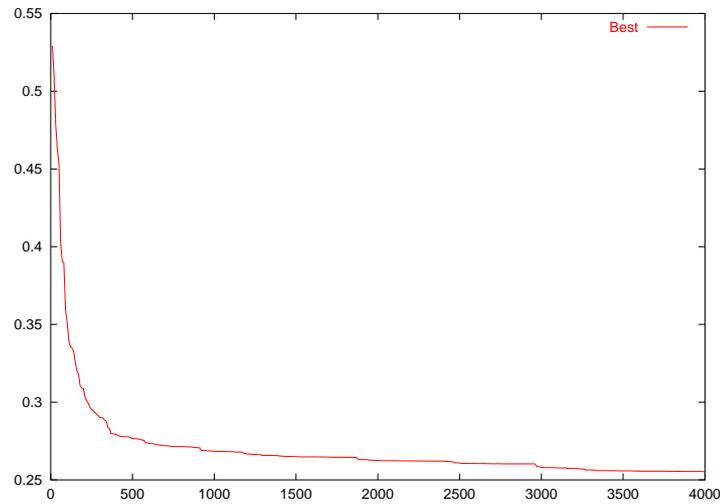


Abbildung 6: Konvergenzkurve der skalaren Fitnessfunktion gegen die Zahl der Generationen im Bereich von 0 bis 4000.

derordnung der beiden länglichen Behälter in den oberen Etagen weichen der optimierten Rohrlänge und der besseren Erfüllbarkeit der Designregeln durch eine versetzte Anordnung in einer tieferen Ebene. Abbildung 8 zeigt den gleichen Konstruktionsvorschlag in einer zweidimensionalen Darstellung, die auch während der Optimierung beobachtet werden kann.

Obwohl bereits die Optimierung auf Basis einer gewichteten Summe der Einzelkriterien Lösungen produzierte, die die Originalanlage in allen Kriterien dominierte, sollte nun überprüft werden, in wie weit auch die multikriterielle Optimierung in der Lage war, die Pareto-optimalen Anlagen zu finden. Hierzu wurde eine Population bestehend aus 30 Individuen gewählt. Die Ergebnismenge bestand zu einem großen Teil aus Anlagen, die die Originalanlage dominierten. Auch eine Aufrechterhaltung der Divergenz der Population konnte beobachtet werden. Eine genaue Inspektion der Lösungsmenge und ein fachliches Urteil über die Nützlichkeit einer solchen Pareto-optimalen Menge von Konstruktionsvorschlägen steht jedoch noch aus.

## 5 Zusammenfassung und Ausblick

Wir haben einen multikriteriellen evolutionären Algorithmus zur Konstruktion von Pareto-optimalen Anlagenentwürfen vorgestellt und entsprechende Mutationsoperatoren entwickelt. Dieser ist in der Lage, auch Varianten des NP-harten *Facility Placement Problems* zu bearbeiten, welches unregelmäßige Grundflächen, mehrere Dimensionen und Zonen ohne Bebauung besitzt. Ziel der Optimierung ist eine Menge von Anlagen, die mehreren Bewertungskriterien gleichzei-

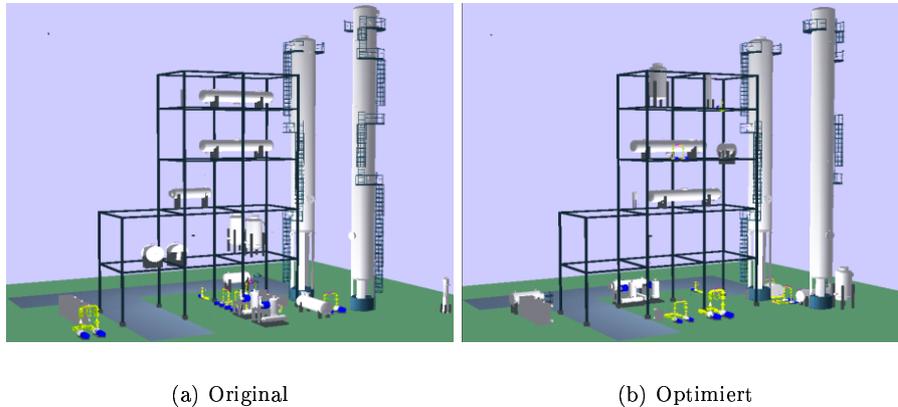


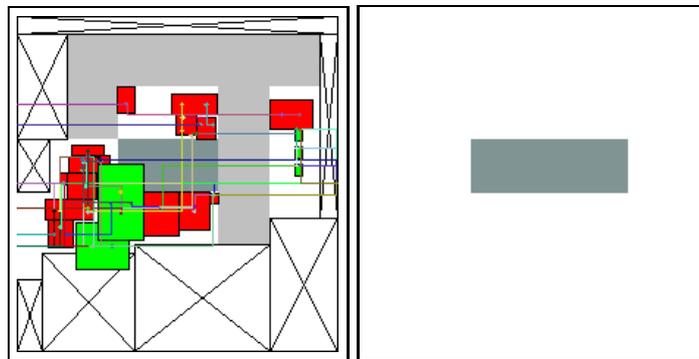
Abbildung 7: Dreidimensionale Darstellung der Originalanlage (links) und einer optimierten Version (rechts). Auffallend ist die Verletzung ästhetischer Gesichtspunkte wie die Übereinanderordnung der länglichen Behälter in den oberen Ebenen zugunsten verfahrenstechnischer Bewertungskriterien.

tig genügen müssen. Darüberhinaus wurde die Berechnung der Fitnessfunktion mittels Techniken der Fuzzy-Logik durchgeführt. Es stellte sich heraus, dass die Behandlung der Unschärfe, die Gewichtung der Designregeln und die Glättung der Fitnessfunktion durch stetige Erfüllbarkeitsgrade zu einer Verbesserung des Optimierungsprozesses führte. Die geschickte Indexierung der Regeln und die Zuordnung der Konklusionen zu Komponenten erlaubte die effiziente Überprüfung des Regelerfüllungsgrads als Teil der Fitnessfunktion. Für eine real existierende Anlage zur Gasbehandlung wurden auf diese Weise Anlagen konstruiert, die in allen Bewertungskriterien bessere Werte als das Original besitzen.

Eine abschließende Beurteilung der Pareto-Menge durch Experten steht jedoch noch aus. Eine weitere Erleichterung der Auswahl des zu bauenden Entwurfs kann durch Clustering der Pareto-Menge und Vorstellung prototypischer Anlagen erzielt werden. In diesem Rahmen können auch weitere verfahrens- oder sicherheitstechnische Anforderungen berücksichtigt werden. Die Konstruktion technischer Anlagen unter Berücksichtigung gewichteter Designregeln ist bisher noch nicht untersucht worden. Eine Anwendung auf standardisierte Benchmarkdatensätze steht folglich noch aus.

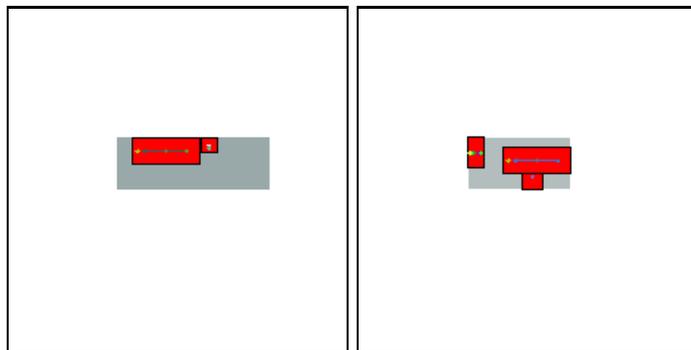
## 6 Danksagungen

Diese Arbeit wurde unterstützt durch die Deutsche Forschungsgemeinschaft (DFG) im Rahmen des Sonderforschungsbereichs 531 „Design und Management komplexer technischer Prozesse und Systeme mit Methoden der Computational Intelligence“.



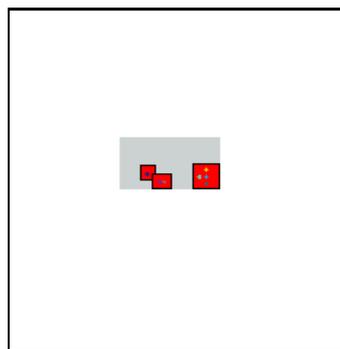
(a) Ebene 0

(b) Ebene 1



(c) Ebene 2

(d) Ebene 3



(e) Ebene 4

Abbildung 8: Zweidimensionale Darstellung der Ebenen der konstruierten Anlage. Die grauen Flächen repräsentieren Wege und den vorgegebenen Stahlbau der Anlage.

## Literatur

- [1] AMORESE, L., V. CENA und C. MUSTACCHI: *A Heuristic For the Compact Location of Process Components*. Chemical Engineering Science, 2(32):119–124, 1977.
- [2] BAASEL, W. D.: *Preliminary Chemical Engineering Plant Design*. Elsevier, New York, Oxford, Amsterdam, 1974.
- [3] CASTELL, C. M. L., R. LAKSHMANAN, J. M. SKILLING, and R. BAÑARES-ALCÁNTARA: *Optimisation of process plant layout using genetic algorithms*. Computers and Chemical Engineering, (22):S993–S996, 1998. Supplement 1.
- [4] CHATTOPADHYAY, S., D. W. BOULDIN, and P. H. DEHKORDI: *An overview of placement and routing algorithms for multi-chip modules*. 1996.
- [5] CHUNG, P. W. H., G. LOVEGROVE, and M. ALI (eds.): *A Quantitative Approach to the Plant Layout Problem Using Genetic Algorithms*, Yverdon, Switzerland, 1993. Gordon and Breach.
- [6] COELLO, C. A. C.: *A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques*. Knowledge and Information Systems, 1(3):129–156, 1999.
- [7] COHOON, J. P. and W. D. PARIS: *Genetic placement*. Proceedings of the IEEE International Conference On Computer-Aided Design, pp. 422–425, 1986.
- [8] DONATH, W. E.: *Complexity theory and design automation*. In *Proceedings of the 17th Conference On Design Automation*, pp. 412 – 419, 1980.
- [9] DUBOIS, D. and H. PRADE: *An introduction to possibilistic and fuzzy logics*. In SHAFER, G. and J. PEARL (eds.): *Readings in Uncertain Reasoning*, chap. IX, pp. 742–762. Morgan Kaufmann, San Mateo, CA, 1990.
- [10] EGUCHI, K., J. SUZUKI, S. YAMANE, and K. OSHIMA: *An application of genetic algorithms to floorplanning of vlsi*. No. 1424 in *Lecture Notes in Computer Science*, p. 263, Heidelberg, 1998. Springer-Verlag.
- [11] ESBENSEN, H.: *A genetic algorithm for macro cell placement*. In *Proceedings of the European Design Automation Conference*, pp. 52–57, 9 1992.
- [12] ESBENSEN, H.: *Computing near-optimal solutions to the steiner problem in a graph using a genetic algorithm*. 1994.
- [13] ESBENSEN, H.: *A macro-cell global router based on two genetic algorithms*. In *Proceedings of the Conference On European Design Automation*, pp. 428–433, Los Alamitos, CA, USA, 1994. IEEE Computer Society Press.

- [14] GAREY, M. R. and D. S. JOHNSON: *Computers and Intractability – A Guide to NP-Completeness*. International Computer Science Series. Freeman, New York, 1979.
- [15] GEORGIADIS, M. C., G. SCHILLING, G. E. ROTSTEIN, and S. MACCHIETTO: *A general mathematical programming approach for process plant layout*. *Computers and Chemical Engineering*, 7(23):823–840.
- [16] GOLDBERG, D. E.: *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, MA., 1989.
- [17] HOLLAND, J. H.: *Adaptation in Natural and Artificial Systems*. MIT Press, Cambridge, MA., 1992.
- [18] HORN, J., N. NAFPLIOTIS, and D. E. GOLDBERG: *A Niche Pareto Genetic Algorithm for Multiobjective Optimization*. In *Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence*, pp. 82–87, Piscataway, New Jersey, 1994. IEEE Service Center.
- [19] KLING, R.-M. and P. BANERJEE: *Optimization by simulated evolution with applications to standard cell placement*. In *Proceedings of the 27th ACM/IEEE Design Automation Conference*, pp. 20–25, 1990. Orlando, FL, 24.-28. June 1990. IEEE, New York.
- [20] KÖSTER, D.: *Ein Assistenzsystem zur methodischen Unterstützung der Aufstellungsplanung von Chemieanlagen*. Shaker Verlag, 1998.
- [21] KUZIAK, A. and S. S. HERAGU: *The facility layout problem*. *European Journal of Operational Research*, 29:229–251, 1987.
- [22] LEE, C. Y.: *An algorithm for path connections and its applications*. *IEEE Transactions On Electronic Computers*, 10(2):346–365, 1961.
- [23] LEUDERS, P.: *Rechnergestützte Optimierung der Layoutplanung von Chemieanlagen*. Shaker Verlag, 2002.
- [24] MAHFOUD, S. W.: *Niching methods*. In BACK, T., D. B. FOGEL, and T. MICHALEWICZ (eds.): *Evolutionary Computation 2: Advanced Algorithms and Operators*, pp. 87–92. Institute of Physics Publishing, 2000.
- [25] MALINGRIAUX, R., K.-R. HILBRING und L. SCHUART: *Zur Optimalen Anordnung der Elemente in Anlagen der stoffumwandelnden Industrie*. *Wissenschaftliche Zeitung der Technischen Hochschule Otto von Guericke*, 8(14), 1970.
- [26] MELLER, R. und K. GAU: *The Facility Layout Problem: Recent and Emerging Trends and Perspectives*. *Journal of Manufacturing Systems*, 15:351–366, 1996.

- [27] PENTEADO, F. D. und A. R. CIRIC: *An MINLP Approach For Safe Process Plant Layout*. Industrial and Engineering Chemistry Research, 4(35):1354–1361, 1996.
- [28] PETERS, M. S., K. D. TIMMERHAUS, and R. E. WEST: *Plant Design and Economics For Chemical Engineers*. McGraw-Hill, New York, 2003.
- [29] RECHENBERG, I.: *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Frommann-Holzboog, Stuttgart, 1973.
- [30] SCHNECKE, V. and O. VORNBERGER: *Hybrid Genetic Algorithms for Constrained Placement Problems*. IEEE Transactions on Evolutionary Computation, 1(4):266–277, 1997.
- [31] SHAHOOKAR, K. and P. MAZUMDER: *Gasp - a genetic algorithm for standard cell placement*. In *Proceedings of the European Design Automation Conference*, pp. 660–664, 1990. Glasgow (UK), 12.-15. March 1990. IEEE Computer Society Press, Los Alamitos, CA.
- [32] SHAHOOKAR, K. and P. MAZUMDER: *Vlsi cell placement techniques*. ACM Computing Surveys, 2(23):143–220, 6 1991.
- [33] YAN, W. X., C. L. NING, Y. J. W. X. YAN, C. L. NING, and Y. JIAN: *Application of genetic algorithms in plant layout*. Industrial Engineering and Management, 4(7):30–34, 8 2002.
- [34] YU, Q., S. BADIDA, and N. SHERWANI: *Algorithmic aspects of three dimensional mcm routing*. In *31st ACM/IEEE Design Automation Conference*, 1994.
- [35] YU, Q., J. L. LAI, S. BADIDA, and N. SHERWANI: *A performance driven three dimensional approach to mcm routing*. Technical report, 9 1993.
- [36] ZITZLER, E. and L. THIELE: *Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach*. IEEE Transactions on Evolutionary Computation, 3(4):257–271, 1999.