Evolutionäre Algorithmen für die Positionsoptimierung von Vakuumspannern zur Holzwerkstückfixierung

Hartmut Ott, Jürgen Wakunda, Andreas Zell
Wilhelm-Schickard-Institut für Informatik, Abt. Rechnerarchitektur
Universität Tübingen
Köstlinstraße 6, D-72074 Tübingen
{ott,wakunda,zell}@informatik.uni-tuebingen.de

Zusammenfassung

Ein geometrisches Optimierungsproblem, das in der Holzindustrie auftaucht, wird versucht mittels Evolutionärer Algorithmen zu lösen. Dabei werden Genetische Algorithmen und Evolutionsstrategien eingesetzt und es wird eine neuartige Methode der Co-Evolution getestet. Die Ergebnisse überzeugen sowohl durch die Qualität der Ausgabe als auch durch die Einfachheit der Formulierung einer Fitneßfunktion, die die Güte der gefundenen Lösung beschreibt.

Stichwörter: Genetische Algorithmen, Evolutionsstrategien, Co-Evolution, Plazierungsproblem.

1 Einleitung

Ziel dieser Arbeit ist es, Evolutionäre Algorithmen [Nis97] auf ein geometrisches Optimierungsproblem anzuwenden, das von der Firma Homag Maschinenbau AG, Schopfloch, gestellt wurde. Um Holzplatten automatisiert zu bearbeiten, müssen sie zuerst auf einen Bearbeitungstisch gelegt und fixiert werden. Zur Fixierung werden Vakuumspanner eingesetzt, die sich mit der einen Seite an das Werkstück und der anderen an den Tisch festsaugen. Ziel ist eine optimale Plazierung dieser Spanner, sodaß sie während der Bearbeitung (Fräsen, Sägen, Bohren, Kante anleimen, etc.) nicht im Weg sind und das Werkstück möglichst nicht verrutscht oder zu flattern beginnt. Abb. 1 zeigt HOMAG-Holzbearbeitungsmaschinen.

Bisher wurden entweder algorithmische Plazierungsverfahren eingesetzt, oder man positionierte die Vakuumspanner einfach manuell. Algorithmische Verfahren haben den Nachteil, daß ihnen eine bestimmte Strategie der Lösungssuche zugrunde liegt. Aufgrund der enormen Vielfalt der Eingabekonfigurationen kann solch eine feste Strategie nicht immer akzeptable Lösungen liefern.

Optimierungen denen eine Qualitätsfunktion zugrunde liegt, bewerten lediglich Lösungsinstanzen. Dadurch ergibt sich ein Qualitätsgebirge, in dem man nun nach dem globalen Extremum sucht. Es ist keine bestimmte Strategie der Suche nach der besten Lösung notwendig. Man versucht lediglich im Qualitätsgebirge abwärts zu „wandern“.

Es gibt eine Reihe von Optimierungsverfahren, die anhand einer Qualitätsfunktion nach globalen Extrema suchen. Wegen des hochdimensionalen Parameterraumes und der Existenz von vielen lokalen Extrema im Lösungsraum scheiden klassische Optimierungsverfahren (Gradientenverfahren, Simplex-Algorithmus, etc.) aus. Dies ist der Grund, warum Evolutionäre Algorithmen eingesetzt wurden.

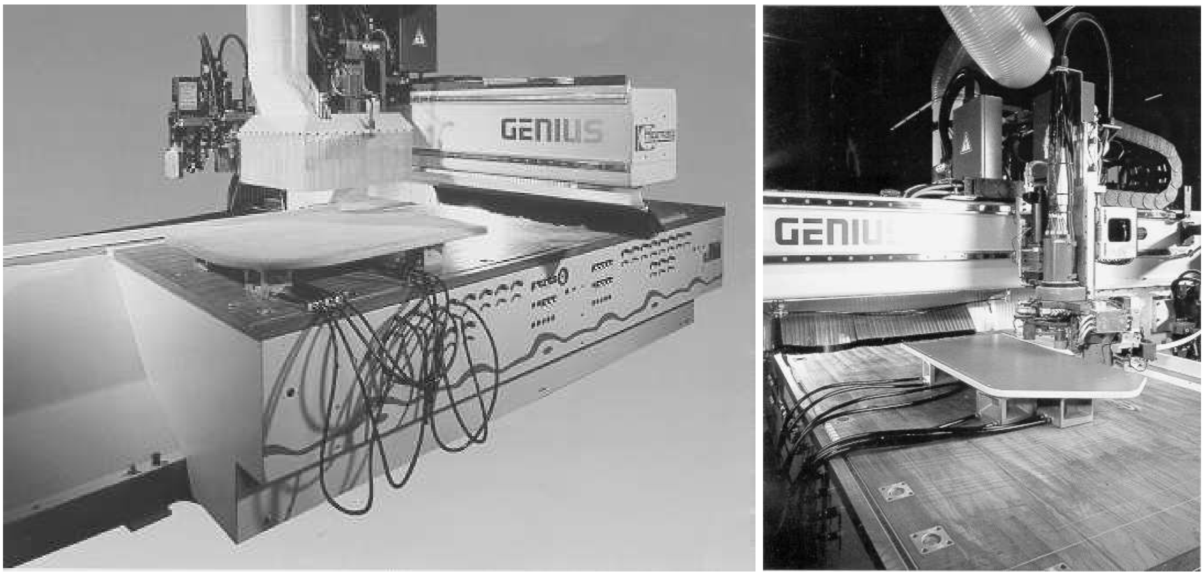

Abbildung 1: HOMAG-Holzbearbeitungsmaschinen mit Flachtisch

Das potentielle Anwendungsspektrum Evolutionärer Algorithmen im Bereich der Optimierung ist fast unbegrenzt. Immer dort, wo für ein Optimierungsproblem keine effizienten konventionellen Algorithmen bekannt sind, werden Evolutionäre Algorithmen erfolgreich eingesetzt. So ist auch die Spanneroptimierung zu komplex, um algorithmische Verfahren anwenden zu können.

Die Palette der bereits erfolgreich implementierten Optimierungen mittels Evolutionärer Algorithmen geht weit über das hier vorgestellte Problem hinaus. Sie reicht vom einfachen Travelling Salesman Problem über Verkehrsleitprogramme bis hin zur Produktionsplanung. In [Sch94] werden einige Anwendungen von Evolutionären Algorithmen vorgestellt.

2 Problemspezifikation

Um den Umfang der Arbeit einzuschränken, wurde die reale Problemstellung etwas vereinfacht. Dabei werden weniger wichtige Details der Optimierung vorerst weggelassen. Abb. 2 zeigt das Optimierungsproblem mit seinen Eingaben und Kriterien. Die Werkstückform kann interaktiv mit Hilfe der graphischen Oberfläche WoodWOP [Sch97] (WOP steht für werkstattorientiertes Programmieren) generiert werden. Dabei werden auch die Bearbeitungen festgelegt, die am Werkstück nötig sind. Daraus wird ein NC-Programm erzeugt, das für die Steuerung der Maschine verantwortlich ist. Durch eine CNC-Simulation werden die Bearbeitungsspurformen der Maschine ermittelt, welche dann als Eingabe für die Spannmitteloptimierung dienen.

Weitere Eingaben der Optimierung sind die Beschreibungen des Maschinentisches und der Vakuumspanner. Zusätzliche Angaben wie die Verfahrwege oder die Parkpositionen der Spanner sind ebenfalls denkbar, werden aber hier nicht betrachtet. Parkpositionen sind die Positionen der Spanner, die sie bei Nichtbenutzung einnehmen. Verfahrwege der Spanner treten meist nur bei Konsolentischen auf und schränken die Bewegungsmöglichkeit des Spanners ein.

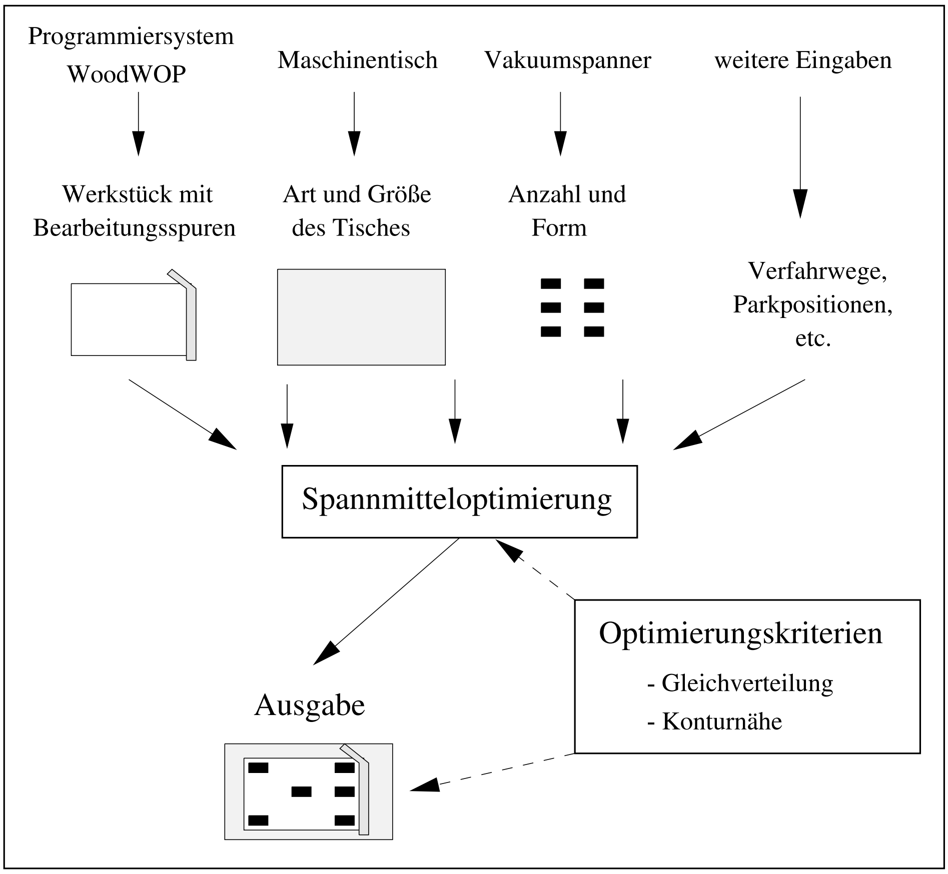

Abbildung 2: Optimierungsproblem

2.1 Eingaben der Optimierung

Der Evolutionäre Algorithmus erhält folgende Eingaben zur Optimierung:

- Bearbeitungstisch: Grundsätzlich gibt es zwei verschiedene Tischarten. Bei Maschinentischen mit automatisch verfahrbaren Konsolen werden die Vakuumspanner auf den Konsolen plaziert. Dabei sind die Spanner in ihrer Bewegungsfreiheit durch die Bewegung der Konsolen eingeschränkt. Auf sogenannten Flachtischen kann der Spanner auf beliebigen Positionen und mit beliebiger Drehung plaziert werden. Im folgenden sollen nur Flachtische betrachtet werden. Die Form des Tisches ist rechteckig und somit durch eine Länge und Breite (angegeben in mm) eindeutig bestimmt.
- Spannmittelform: Die Form der Spannmittel wird durch ein beliebiges konvexes Polygon angegeben.
- Werkstückform: Eine beliebige Anzahl von sich nicht schneidenden Polygonen.
- Bearbeitungsspurform: Eine beliebige Anzahl von sich nicht schneidenden Polygonen.
- Spannmittelanzahl: Es dürfen bis zu 12 Vakuumspanner in der Optimierung verwendet werden. Alle Vakuumspanner eines Maschinentisches haben die gleiche Form.

Bei Flachtischen muß die tatsächliche Positionierung der Spanner vom Maschinenführer per Hand geleistet werden. Um den Maschinenführer zu leiten und ihm die Spannerposition und -drehung anzuzeigen, wird ein Projektionslaser an der Hallendecke positioniert. Die Spannerkonturen werden dabei mittels dieses Lasers auf das Werkstück projiziert.

Die Optimierung wird gesteuert durch eine bestimmte Menge von Gütekriterien und Restriktionen. Die Restriktionen dürfen auf keinen Fall verletzt werden, während gleichzeitig die Güte der Lösungen möglichst optimal sein soll.

Restriktionen sind:

- Die Vakuumspanner dürfen die Bearbeitungskonturen nicht berühren oder schneiden und müssen innerhalb der Werkstückkontur plaziert werden. Dabei soll eventuell ein Sicherheitsabstand zum Rand eingehalten werden.
- Die Vakuumspanner dürfen sich nicht gegenseitig überlappen.

Folgende Gütekriterien wurden definiert:

- Die Vakuumspanner sollten nah an der Bearbeitungskontur sein.
- Die Anzahl der Vakuumspanner im Innengebiet soll ausreichend sein.
- Die Teilefläche soll gleichmäßig von den Spannern überdeckt werden.
- Die Vakuumspanner sollten nah an schmalen Stellen des Teils sein.

Außerdem ist durch die Anwendung eine Zeitbeschränkung vorgegeben, wonach die Berechnung der optimalen Spannerpositionierung auf einem Pentium 200-Rechner nicht länger als 5 Sekunden dauern sollte.

3 Geometrie-Algorithmen

Bei den implementierten Optimierungen kommen verschiedene Geometrie-Algorithmen vor allem aus der Computergrafik [Dam90] zum Einsatz. Insbesondere muß die Berechnung des Schnitts zweier beliebiger Polygone effizient durchgeführt werden. Es ist dabei zu beachten, daß die Rechenzeit ein entscheidender Faktor ist, weil es unter Umständen sehr komplexe Eingabepolygone geben kann. Außerdem müssen solche Schnittberechnungen im Laufe der Evolution unter Umständen sehr oft wiederholt werden. Deswegen wurde hier ein Ansatz gewählt, der zwar nicht immer exakte Ergebnisse liefert, der jedoch sehr effizient ist.

Die Schnittberechnung von Polygonen wird sowohl für den Vakuumspanner-Kontur-Schnitt als auch für den Schnitt der Konturen zweier Vakuumspanner (Kollisionsdetektion) benötigt. Grundlegender Unterschied dieser beiden Schnittberechnungen ist, daß sich die Konturen in ihrer Position und Rotation im Laufe der Evolution nicht ändern. Somit kann durch einen Vorverarbeitungsschritt der Vakuumspanner-Kontur-Schnitt effizienter gemacht werden.

Die Vorgehensweise zur Berechnung des Vakuumspanner-Kontur-Schnitts ist aus der Computergraphik bekannt. Sie benutzt einen Polygonfüllalgorithmus (Scankonvertierung). Werkstück- und Bearbeitungspolygone werden mithilfe einer Scanline rasterisiert. Dabei wird festgestellt, welche Rasterquadrate im gültigen bzw. ungültigen Bereich liegen. Ein Rasterquadrat ist im gültigen Bereich, falls es sich innerhalb des Werkstückes und außerhalb der Bearbeitungskontur befindet.

Die so erzielte Aufteilung in gültige und ungültige Rasterquadrate wird in einem ScanArray gespeichert. Um nun die Schnittberechnung durchzuführen, rasterisiert man den Vakuumspanner ebenfalls und zählt einfach die Rasterquadrate innerhalb des Vakuumspanners, die ungültig sind. Bei dieser Berechnung ist darauf zu achten, daß ein vorher angegebener Sicherheitsabstand zu der Kontur einzuhalten ist (etwa zwischen 0.5 und $2cm$).

Mit einer ähnlichen Methode kann auch der Schnitt der Konturen zweier Vakuumspanner berechnet werden. Die Berechnung ist hier aufwendiger, da für jede Spanner-Spanner Kombination getrennt ein Test durchgeführt werden muß. Allerdings kann man durch einen vorgeschalteten Abstandtest dem zeitaufwendigen Test auf Schnitt aus dem Weg gehen.

Der Scankonvertierungsalgorithmus funktioniert für eine beliebige Anzahl von Polygonen. Allerdings besteht die Restriktion, daß sich die Kanten der Polygone nicht schneiden dürfen.

Es wäre ebenfalls eine exakte (nicht diskretisierende) Schnittberechnung vorstellbar. Ebenso könnte man den Abstand von einem Spanner zur nächsten Konturkante in die Fitneßfunktion mit einfließen lassen. Die Arbeit beschränkte sich jedoch auf die Implementierung der oben beschriebenen Rasteralgorithmen.

Ein weiterer Vorteil der implementierten Rasterisierung neben der Schnelligkeit ist folgender: Die Grobheit der Rasterisierung kann stufenlos eingestellt werden. Stellt man fest, daß die Berechnung zu zeitaufwendig ist, kann durch diesen Parameter die Rechenzeit auf Kosten der Genauigkeit verringert werden.

Nachteil der Scankonvertierung ist die Diskretisierung des Problems und die damit einhergehenden Aliasing-Effekte, die vor allem dann auftreten, wenn man sehr kleine Konturen hat. Aliasing-Effekte sind Effekte, die durch eine Unterabtastung des Originalpolygons entstehen. Gewisse Formfeinheiten werden dann nicht mehr erkannt. Wählt man zum Beispiel eine Auflösung von $5mm$, so werden bei Polygonzügen kleinerer Größenordnung (z.B. ein $3mm$ Bohrloch) Aliasing-Probleme auftreten. Das Bohrloch würde dann als ein $5x5mm$ Quadrat, also als ein Rasterquadrat, angenähert werden.

Folgende Schnittwerte werden durch die Geometrie-Algorithmen berechnet:

- $SK(i)$ ist der Schnitt von Spanner i mit unerlaubten Konturen,
- $SS(i)$ ist der Schnitt von Spanner i mit anderen Spannern.

wobei $0 \leq SS(i), SK(i) \leq 1$ und die $SS(i)$ bzw. $SK(I)$ reellwertig sind. Ist $SS(i) = 0.5$ so bedeutet das, daß die Hälfte der Fläche des Spanners i von anderen Spannern überdeckt ist. Ebenso bedeutet $SK(i) = 0.5$, daß Spanner i zur Hälfte auf einer unerlaubten Kontur positioniert ist.

Daraus ergibt sich dann:

$$I = \sum_{i=1}^{n} SS(i) \tag{1}$$

wobei n ist die Anzahl der Vakuumspanner ist. $SK(i)$ wird verwendet, um für ein Individuum zu testen, ob die einzelnen Vakuumspanner auf einer gültigen Position plaziert sind. I wird in den Optimierungsverfahren als Teilterm verwendet, damit die Spanner eine Tendenz haben, sich voneinander zu entfernen.

4 Optimierungsverfahren

Es wurde mit drei verschiedenen Verfahren optimiert, die jeweils Evolutionäre Algorithmen einsetzen. Diese Verfahren unterscheiden sich im wesentlichen in der Behandlung der Restriktionen und der Repräsentation der Individuen. Der benutzte Fitneßterm setzt sich jeweils aus Teiltermen zusammen. Ein Teilterm sichert die Gleichverteilung, der zweite die Konturnähe und der dritte addiert einen Strafterm, wenn Restriktionsverletzungen auftreten. Die Teilterme wichtet man mit Faktoren, die heuristisch bestimmt werden.

4.1 Gemeinsame Teilterme der Optimierungen

Für jedes der drei vorgestellten Verfahren wurde ein eigener Fitneßterm verwendet, der auf die speziellen Eigenheiten der Optimierung abgestimmt ist. Es werden bestimmte Teilterme bei mehr als einem Verfahren benutzt. Diese sollen hier vorgestellt werden.

4.1.1 Gleichverteilung der Vakuumspanner

Das Gütekriterium der Gleichverteilung wird bei allen drei Verfahren durch den gleichen Teilterm erreicht. Um diesen Teilterm zu berechnen werden zuerst alle Spanner-Spanner-Abstände berechnet und in der Menge M gespeichert:

$$M = \bigcup_{i<j} d(S_i, S_j) = \{d_1, d_2, d_3, ..., d_{\#M}\} \tag{2}$$

Hierbei bezeichnen n die Anzahl der Vakuumspanner, S_i den Referenzpunkt des Vakuumspanners i, $d(P_i, P_j)$ den Abstand von Punkt i zu Punkt j, M die Menge aller Abstände, $\#M$ die Anzahl der verschiedenen Abstände und M' die sortierte Menge aller Abstände.

$$\#M = \frac{n \cdot (n-1)}{2} \tag{3}$$

Diese $\#M$ Abstände werden in aufsteigender Reihenfolge sortiert und in M' gespeichert:

$$M' = \{d'_1, d'_2, d'_3, ..., d'_{\#M}\} \tag{4}$$

Ziel des Gleichverteilungs-Kriteriums ist es, das Abstandsmaß E zu maximieren:

$$E = \sum_{i=1}^{k} \sqrt{d'_i} \tag{5}$$

Bei der Berechnung der Gleichverteilungsformel (5) werden nur die kleinsten Abstände herangezogen und aufaddiert. Dies führt zu besseren Ergebnissen, als alle Abstände aufzuaddieren. Es bildet sich eine Topologie, in der einzelne, weit entfernte Vakuumspanner zunächst nicht betrachet werden. Somit können Spanner, die sich durch eine ungünstige Anfangskonfiguration gruppiert haben, sich sehr schnell voneinander entfernen.

Zusätzlich wird von dem eigentlichen Abstand noch eine Wurzelberechnung durchgeführt. Dies führt dazu, daß z.B. ein Vakuumspanner, der in der Mitte zweier anderer Spanner liegt, den besten Fitneßwert erhält, wenn er genau zwischen den beiden plaziert wird. Die Anzahl der Abstände k, die man mit in die Berechnung einfliessen läßt, kann variiert werden. In der Implementierung wurde $k = n$ gewählt.

4.1.2 Spannerelimination

In der Praxis tritt sehr häufig der Fall auf, daß nicht alle Spanner positioniert werden können. Dabei ist entweder das Werkstück zu klein, die Spannerform zu groß, oder es sind zu viele Bearbeitungen bei der Positionierung im Weg.

Wenn die freie Fläche zu klein für die Positionierung aller Spanner ist, so wird eine schrittweise Elimination der Spanner eingesetzt. Beginnend mit allen Spannern werden nach und nach Spanner eliminiert, die sich noch mit anderen Spannern schneiden. Diese Elimination kann sehr gut in die Evolution eingebettet werden. Man definiert die Eliminationspunkte einfach als Prozentzahl der Generationen. Zwischen zwei Eliminationspunkten müssen genügend viele Generationen dazwischen liegen, um entstandene Lücken aufzufüllen und die Spanner wieder neu zu formieren.

Damit die Spanner eine Tendenz haben, einer Überlappung mit anderen Spannern aus dem Weg zu gehen, wird für jeden Spanner auf die Fitneßfunktion ein Strafterm aufaddiert, der proportional zu I von Gleichung 1 ist.

4.1.3 Nähe zu Konturen

Der Checkpoint-Optimierungsterm soll gewährleisten, daß die Vakuumspanner möglichst nahe an Konturen sind. Idee dabei ist, Punkte auf den Konturen (Werkstück und Bearbeitungen) zu plazieren die im folgenden Checkpoints genannt werden. Aus den Abständen der Vakuumspanner zu den Checkpoints wird die Qualitätsfunktion berechnet. Für einen Checkpoint wird der quadratische Abstand zum nächstgelegenen Spanner berechnet, und die Abstände werden über alle Checkpoints aufsummiert. Dieser Teilterm P wird dann zur Qualitätsfunktion addiert. Durch geschickte Verteilung der Checkpoints kann man erreichen, daß sich die Spannmittel gut und gleichmäßig an die Werkstück- bzw. Bearbeitungskonturen annähern. Dabei wird die Checkpointdichte abhängig gemacht von der Konturlänge und den Winkeln von Konturecken.

Der Checkpoint-Optimierungsterm:

$$P = \sum_{i=1}^{m} \min_{1 \le j \le n} [d(C(i), S(j)]^2 \tag{6}$$

Hierbei ist n die Anzahl der Spannmittel, m die Anzahl der Checkpoints, $S(i)$ der Mittelpunkt des Vakuumspanners i, $C(i)$ die Position des i-ten Checkpoints und $d(P_i, P_j)$ der Abstand von Punkt P_i zu Punkt P_j.

4.2 Verfahren mit Evolutionsstrategien

Das erste Verfahren benutzt Evolutionsstrategien [Rec73] [Rec94] [Sch95] [Bäc92] und codiert jeden Vakuumspanner mit seinen 3 Freiheitsgraden x-Position, y-Position und Winkel. Lösungen, die die Restriktion „Spanner schneidet unerlaubte Fläche“ verletzen, werden nicht zugelassen. Dabei werden diese nichtgültigen Individuen durch Rücksetzen der Spanner auf eine gültige Position repariert. Der Restriktion „Spanner schneidet andere Spanner“ wird durch das Spannereliminationsverfahren Genüge getan.

Um eine Effizienzsteigerung zu erhalten, wird die eigentliche Mutation erweitert. Dabei werden mit einer bestimmten Wahrscheinlichkeit Vakuumspanner von einer Position auf eine zufällig gewählte andere Position gesetzt. Ziel dieser problemspezifischen Erweiterung ist es, damit anfangs eine Art Groboptimierung durchzuführen. Nach einer bestimmten Anzahl von Generationen wird der Anteil dieser erweiterten Mutation an der Gesamtmutation erniedrigt, und die Mutation durch die MSR (adaptive Mutationsschrittweitenregelung) dominiert wieder.

Die Individuen bei dieser Optimierung dürfen nur Spannerpositionen beinhalten, die gültig sind. Deswegen wird die Individuenpopulation anfangs geeignet initialisiert. Dabei werden die Spanner von Individuen nur auf erlaubte Startpositionen gesetzt. Um solche erlaubten Startpositionen für die Spanner zu finden, wird der Evolution ein Verarbeitungsschritt vorgeschaltet, der nach diskreten erlaubten Spannerpositionen sucht. Ein weiterer Grund dieser Vorgehensweise ist, daß durch eine gute Anfangskonfiguration das Verfahren schneller konvergiert.

Aus den Gleichungen (6), (5) und (1) ergibt sich die Fitneßfunktion für das Verfahren mit Evolutionsstrategien. Dabei wurde festgelegt, daß ein kleinerer Wert eine bessere Fitneß F bedeutet (smaller is better):

$$F = \alpha \cdot P - \beta \cdot E + \gamma \cdot I \tag{7}$$

Hierbei ist α die Wichtung des Checkpoint-Terms, P der Checkpoint-Term von Formel 6, β die Wichtung des Gleichverteilungsterms, E der Gleichverteilungs-Term von Formel 5, γ die Wichtung des Schnitt-Terms und I der Schnitt-Term von Formel 1.

Die Wahl der drei Parameter α,β und γ ist durch Testen verschiedener Konfigurationen und durch subjektive Auswertung zu bestimmen.

4.3 Verfahren mit Genetischen Algorithmen

Genetische Algorithmen [Hol75] [Hol93] [Gol89] werden in dieser Optimierung eingesetzt. Das Restriktionsproblem wird hier dadurch umgangen, daß in einem Vorverarbeitungsschritt eine bestimmte Menge an Positionen entlang der Konturkanten, der Konturecken und in der Mitte von Werkstücken berechnet werden. Dieser Verarbeitungsschritt ist analog dem Suchen nach Startpositionen bei dem Verfahren mit Evolutionsstrategien. Der Genetische Algorithmus soll nun die beste Kombination dieser Positionen ermitteln.

Das Suchen nach diskreten Positionen erfolgt nach einfachen heuristischen Verfahren. Um beispielsweise gute Kantenpositionen herauszufinden, wird der Spanner einfach so rotiert, daß seine längste Seite entlang der Kante ausgerichtet ist. Nun wird er so nah wie möglich in orthogonaler Richtung zur Kante an sie herangeschoben.

Bei der Codierung der diskreten Positionen in einen Bitstring sollte beachtet werden, daß das Prinzip der starken Kausalität eingehalten wird. Verändert sich die Lösung nur ein wenig, so sollte sich auch die Bewertung des Individuums das die Lösung codiert, nur wenig verändern. Wird dieses Prinzip eingehalten, konvergiert das Verfahren schneller.

Es wird das gleiche Spannereliminationsverfahren eingesetzt wie bei der Optimierung mit Evolutionsstrategien. Die Restriktion „Spanner schneidet unerlaubte Fläche“ wird wie oben angedeutet dadurch umgangen, daß nur eine bestimmte Menge an diskreten gültigen Spannerpositionen zugelassen sind. Bei diesem Optimierungsverfahren wurde, um die Konturnähe der Spanner zu garantieren, ein anderer Teilterm eingesetzt als beim Verfahren mit Evolutionsstrategien (Checkpointterm).

Es wurde dabei ausgenutzt, daß die Suche nach Spannerpositionen an sich schon in Innen- , Kanten- und Eckpositionen kategorisiert. Die diskreten Spannerpositionen bekommen zusätzlich eine Wichtung, die ausdrückt, wie diese verschiedenen Positionen belohnt werden. Der Konturterm K kann dann anhand dieser Wichtungen berechnet werden.

Es gilt dabei:

$$weight(j) = \begin{cases} w_1 & : \text{ falls Spannerposition j an Werkstückkante} \\ w_2 & : \text{ falls Spannerposition j an Werkstückecke} \\ w_3 & : \text{ falls Spannerposition j an Bearbeitungskante} \\ w_4 & : \text{ falls Spannerposition j an Bearbeitungsecke} \\ w_5 & : \text{ falls Spannerposition j innerhalb des Werkstückes} \end{cases} \qquad \forall 1 \leq j \leq d \quad (8)$$

$$K = \sum_{i=1}^{n} weight(spanner(i)) \quad (9)$$

Hierbei ist n die Anzahl der Spanner, d die Menge von diskreten Spannerpositionen, $weight(j)$ die Wichtung für Spannerposition j und $spanner(i)$ die diskrete Spannerposition, auf der Spanner i plaziert ist.

Die Parameter w_1,w_2,w_3 und w_4 sind dabei variabel einstellbar. Wählt man dabei w_1 und w_2 groß gegenüber w_3 und w_4, so werden sich die Spanner bevorzugt an den Werkstückkonturen festsetzen. Umgekehrt gilt bei der Wahl von großen w_3 und w_4, daß Bearbeitungskonturen stärker unterstützt werden. w_5 wurde bei den Optimierungen auf 0 gesetzt. Als Gesamtfitneßterm ergibt sich wie bei Verfahren 1:

$$F = \alpha \cdot K - \beta \cdot E + \gamma \cdot I \quad (10)$$

4.4 Verfahren mittels Co-Evolution

In diesem Verfahren werden wieder Evolutionsstrategien eingesetzt. Dabei werden aber im Gegensatz zum 1. Verfahren mit Evolutionsstrategien die Individuen nicht repariert. Bei einer Restriktionsverletzung wird zur Fitneß ein Strafterm aufaddiert. Dieser Strafterm ist allerdings nicht statisch, sondern Faktoren im Strafterm werden während der Evolution angepaßt. Man bezeichnet das Verfahren auch als Co-Evolution [Mic96] [Par94].

Bei der Co-Evolution existieren zwei Populationen, die Lösungspopulation und die Restriktionspopulation. Ähnlich wie in der Natur bei einer Raubtier-Beute Beziehung besteht eine inverse Fitneßinteraktion der Populationen. Entwickelt das Raubtier bessere Angriffsstrategien, so sollte die Beute bessere Verteidungsstrategien ausbilden. Die Restriktionspopulation besteht dabei immer aus denselben Individuen, den Restriktionen des Problems.

Restriktionen eines Optimierungsproblems bzw. deren Fitneß sind folgendermaßen definiert:

$$g(i,j) \begin{cases} = 0 & , \text{ falls Individuum } i \text{ die Restriktion } j \text{ einhält} \\ > 0 & , \text{ falls das Individuum } i \text{ die Restriktion } j \text{ verletzt} \end{cases} \quad (11)$$

$$C(j) = \text{Fitneßwert von Restriktion } j \quad (12)$$

Für das spezielle Problem der Spanneroptimierung gibt es folgende Restriktionen:

$$g(i,j) = SS(j) \qquad \forall 1 \leq i \leq ind, \forall 1 \leq j \leq n \quad (13)$$

$$g(i, j+n) = SK(j) \qquad \forall 1 \leq i \leq ind, \forall 1 \leq j \leq n \tag{14}$$

Dabei ist ind die Anzahl der Individuen, c die Anzahl der Restriktionen und n die Anzahl der Vakuumspanner.

Anfangs werden die Werte $C(j)$ auf einen initialen Wert gesetzt. Dieser initiale Wert wird sehr groß relativ zu den Teiltermen der Fitneßfunktion gewählt. Damit wird erreicht, daß anfangs nur gültige Lösungen zugelassen werden. Durch die inverse Fitneßinteraktion geschieht folgendes: wird eine Restriktion häufig verletzt, so wird ihr Fitneßwert erhöht. Existieren sehr wenig Individuen, die eine bestimmte Restriktion verletzen, so wird der Fitneßwert erniedrigt.

Genau umgekehrt verfährt man mit der Lösungspopulation. Auf die Lösungen mit sehr viel Restriktionsverletzungen wird ein Strafterm proportional zu dem Maß der Verletzungen aufaddiert. Dabei wichtet man die Restriktionsverletzung mit dem Fitneßwert der Restriktion:

$$S(i) = \sum_{j=1}^{c} C(j) \cdot g(i, j) \qquad \forall 1 \leq i \leq ind \tag{15}$$

$$F_{CoEv}(i) = F(i) + S(i) \qquad \forall 1 \leq i \leq ind \tag{16}$$

Die Fitneßinteraktion zwischen den Individuen und den Restriktionen führt zu einer konkurrierenden Co-Evolution. Sobald die Individuen auf einer bestimmten Menge der Restriktionen erfolgreich sind, bekommen diese Restriktionen eine geringere Fitneß (Wichtung). Als Konsequenz bewegen sich andere, noch nicht eingehaltenen Restriktionen, in der Restriktionshierarchie nach oben. Die Individuen versuchen dann mehr diese schwieriger zu erfüllenden Einschränkungen einzuhalten.

5 Implementierung

Die Implementierung erfolgte mit dem Programmpaket EvA, das die Software-Tools VEGA (Verteilte Genetische Algorithmen) [Bau95] und VEES (Verteilte Evolutionsstrategien) [Wak95] enthält. Besonderheit aller Teilsysteme von EvA ist eine Batch-Schnittstelle durch die Skriptsprache Tcl.

Verfahren 1 wurde in VEES eingebettet, wobei haupstächlich die Fitneßfunktion implementiert werden mußte. Jeder Spanner wurde mit drei Werten codiert, der x-Position, der y-Position und der Drehung. Dabei wurde eine Option Drehung/keine Drehung mit eingebaut.

Verfahren 2 wurde in VEGA eingebettet, wobei dort ebenfalls haupsächlich die Fitneßfunktion implementiert wurde. Einem Spanner war hier zuerst durch eine Integer-Zahl eine bestimmte diskrete Spannerposition zugewiesen. Ein zusätzliches Mapping dieser Integer-Zahl auf ein zweidimensionales Feld sollte die starke Kausalität für Einzelspanner gewährleisten.

Verfahren 3 (Co-Evolution) wurde ebenfalls mittels VEES getestet, wobei VEES dabei durch die Co-Evolution erweitert werden mußte.

Um die Ergebnisse der Evolutionsläufe zu visualisieren und verschiedene Testläufe starten zu können, wurde ein Viewer in Tcl/Tk geschrieben. Er stellt den Maschinentisch, das Werkstück, die Bearbeitungen und die Spanner graphisch dar. Mit ihm wurden auch die folgenden Testergebnisse visualisiert.

Als Werkstück- und Bearbeitungspolygone wurden die Polgyonausgaben aus der CNC-Simulation verwendet. Diese werden in einer Polygondatei gespeichert, die direkt Eingabe der Spannmitteloptimierung ist. Außerdem sind auch eigene beliebige Formen als Eingabe der Optimierung erlaubt.

6 Testreihe

Der Test und Vergleich der implementierten Verfahren mit unterschiedlichen Parameterwerten stellte sich als sehr schwierig heraus. Dies war zum einen auf die Vielzahl an Parametern, die unabhängig voneinander verändert werden können, zurückzuführen. Zum anderen ist die Bewertung der Ergebnisse schwer objektivierbar. Eine optimale Einstellung der Parameter kann für viele Testeingaben hervorragende Endresultate liefern und für andere wiederum nur mäßige. Nahezu unmöglich erscheint das Problem, eine Fitneßfunktion zu finden, die eine Vergleichbarkeit der Ergebnisse bei verschiedenen Polygoneingaben ermöglicht.

Aus den obigen Gründen beschränkte sich die Testreihe darauf, die wesentlichen Vor- und Nachteile der Optimierungen anhand von Beispielen aufzuzeigen.

Folgende Testparameter wurden näher untersucht:

- Verfahren VEES,VEGA und Co-Evolution,
- Verfahren der Spannmittelelimination,
- Verschiedene Werkstück- und Bearbeitungskonturformen,
- Methoden der Mutationsschrittweitenadaption bei VEES,
- Vakuumspannerformen,
- algorithmische Parameter von VEES und VEGA,
- Rotation/Nichtrotation der Vakuumspanner bei VEES-Optimierung,
- Gleichverteilung/Bearbeitungskonturnähe,
- Anzahl der Checkpoints,

Die Abbildungen 3, 4 und 5 unten zeigen Testergebnisse, die mit Hilfe der VEES-, VEGA- und Co-Evolutionsverfahren erzielt worden sind. Zu sehen ist jeweils der skalierte Maschinentisch, das Rohholzstück (schwarz), die Bearbeitungskonturen (grau) und die evolvierten Spannerpositionen und -drehungen.

Für die drei Verfahren wurden jeweils gleiche Testbeispiele ausgewählt. Es wurde versucht 8 Spanner der Größe 200x160mm zu plazieren. Der Maschinentisch hatte eine Größe von 2000x1500mm. Man erkennt bei den VEES- und VEGA-Optimierungen eine sehr gute Drehrichtung der Spanner, die sich entlang von Konturkanten ausrichten. Die erwünschte Gleichverteilung der Spanner über das Werkstück und eine Annäherung an Konturen wird ebenfalls erreicht.

Alle Optimierungen benutzten eine Populationsgröße von 20 Individuen. Bei den VEES-Optimierungen reichten 200 Generationen aus, um gute Ergebnisse zu erhalten. Die VEGA-Optimierungen fanden schon nach 100 bis 150 Generationen ansprechende Spannerkonfigurationen. Dagegen benötigte die Co-Evolution immer sehr viel mehr an Generationen als die anderen zwei Verfahren.

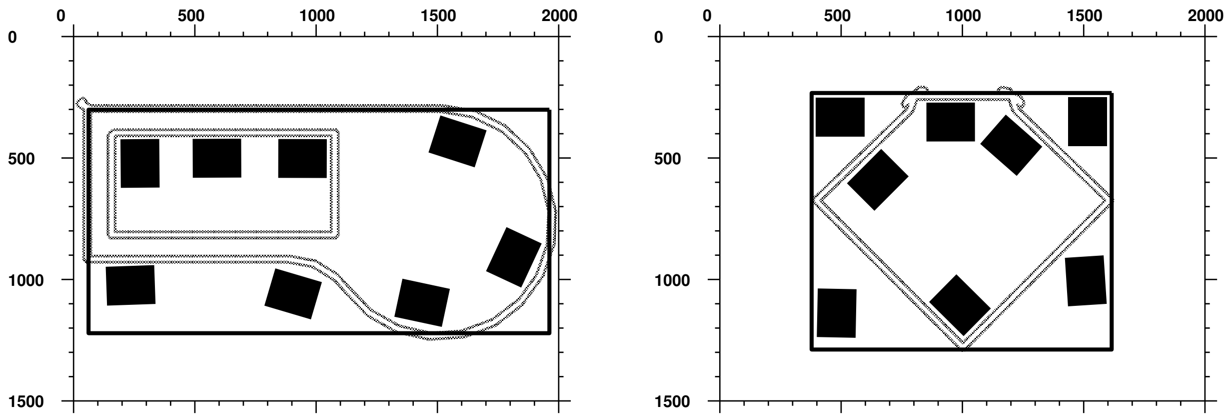

Abbildung 3: VEES-Optimierung

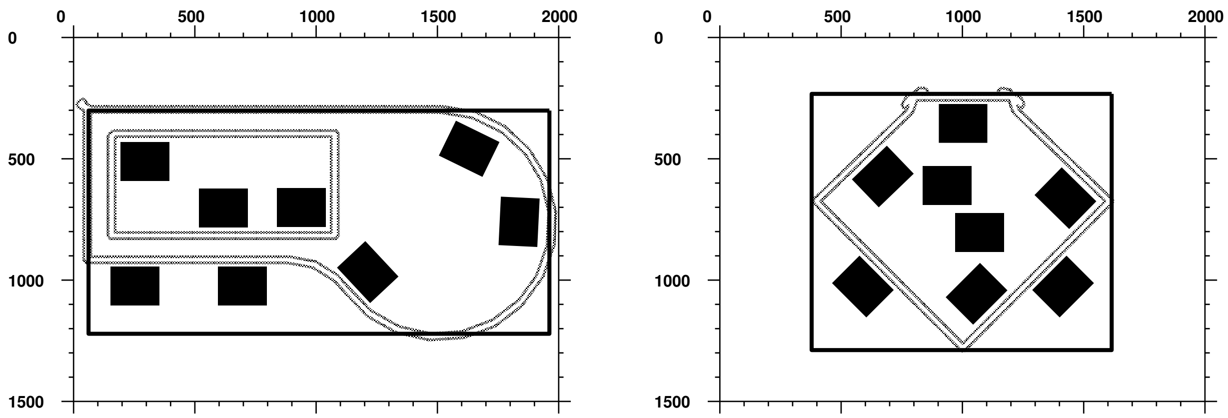

Abbildung 4: VEGA-Optimierung

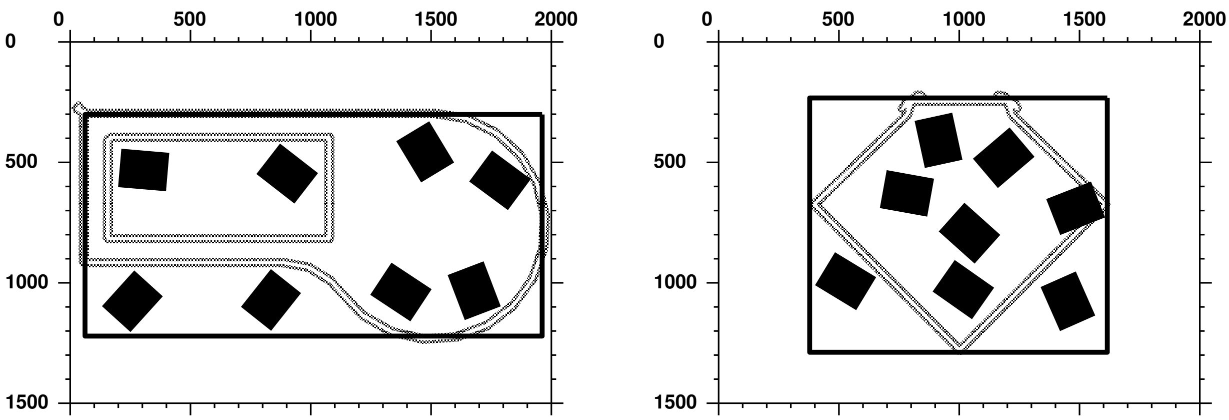

Abbildung 5: VEES-CoEvolution

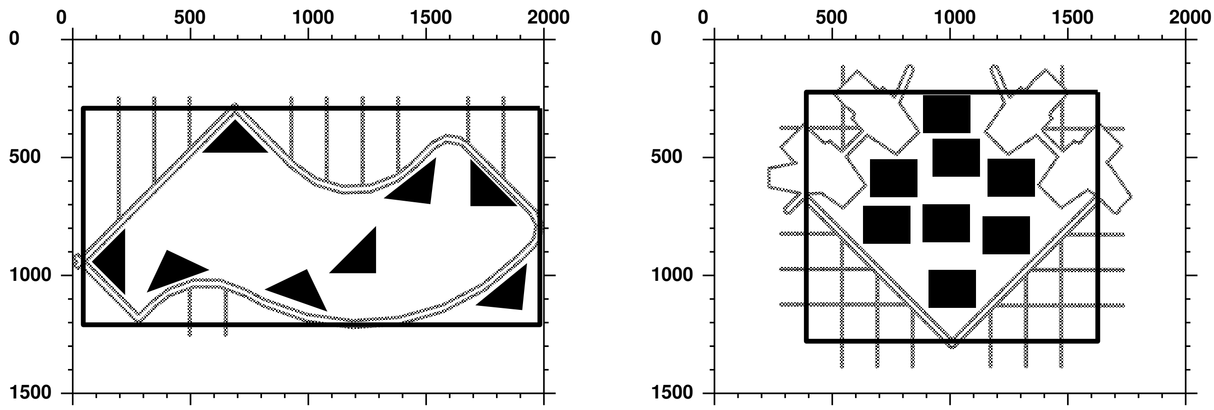

Abbildung 6: VEGA-Optimierung mit dreieckigen Spannern (links) und VEES-Optimierung ohne Rotation der Spanner (rechts)

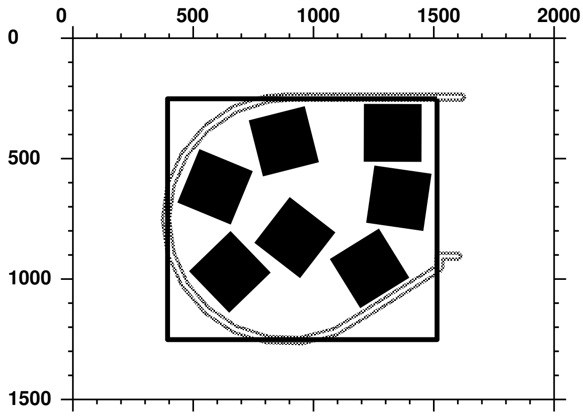

Abbildung 7: VEES-Optimierung mit Spannerelimination

Das Testbeispiel in Abb. 6 links soll zeigen, daß die Spanneroptimierung nicht auf rechteckförmige Spanner beschränkt ist. Grundsätzlich ist jedes konvexe Polygon für einen Spanner denkbar. Hier wurde das VEGA-Verfahren angewendet. Dabei fallen vor allem die guten Positionen und die optimale Drehrichtung der Spanner in den Ecken auf. Dies ist zurückzuführen auf die Heuristik, die beim Suchen nach den diskreten Spannerpositionen verwendet wurde.

Beim Testbeispiel in Abb. 6 rechts nach dem VEES-Verfahren wurden wieder rechteckförmige Spanner der Größe 200x160mm eingesetzt. Hier wurde im Gegensatz zu den anderen Tests die Rotation der Spanner abgeschaltet. Jeder Spanner ist dann nur noch durch 2 Freiheitsgrade (x- und y-Position) bestimmt und die Optimierung reduziert sich in der Dimension.

Sinn des Optimierungstests in Abb. 7 war es, die Spannerelimination zu überprüfen. Aus diesem Grund wurden hier relativ große (240x240mm) und sehr viele (12) Spanner eingesetzt. Dabei war dagegen die erlaubte Werkstückfläche relativ klein. Man erkennt, daß insgesamt 7 Spanner plaziert werden konnten.

7 Schlußfolgerung und Ausblick

Die Testreihe hat als Ergebnis gezeigt, daß die bisher implementierte Co-Evolution noch nicht ausgereift ist, um in der Realität eingesetzt werden zu können. Sie verletzt zum Teil noch K.O.-Kriterien und produziert damit nicht gültige Lösungen. Zwar kann sie bei wenig Spannern und somit kleiner Dimension des Problems durchaus mit der VEES- und VEGA- Optimierung mithalten, allerdings gibt es bei vielen Spannern noch Probleme. Das Reparieren nichtgültiger Individuen bei der VEES-Optimierung hat sich bewährt und liefert gute bis sehr gute Ergebnisse. Die VEGA-Optimierung läßt durch die Vorverarbeitung und Codierung gar keine Spanner auf ungültigen Positionen zu und liefert ebenfalls gute Testergebnisse.

Zwischen VEES- und VEGA-Optimierung lassen sich anhand der Testergebnisse relativ wenige Unterschiede erkennen. Bei wenig Spannern benötigt die VEGA-Optimierung weniger Generationen als die VEES-Optimierung um zu sehr guten Lösungen zu kommen. Dagegen scheint die VEES-Optimierung bei mehr Spannern bessere Resultate zu liefern. Durch die Diskretisierung in der VEGA-Optimierung ergeben sich bei der Spannerelimination erhebliche Vorteile für die VEES-Optimierung. In der Feinpositionierung an Konturkanten oder -ecken liefert wiederum die VEGA-Optimierung die besseren Resultate. Es wurde bei den Verfahren darauf Wert gelegt, daß die eingesetzte Fitneßfunktion sehr einfach ist, aber gleichzeitig auch alle Optimierungskriterien beinhaltet. Die VEES- und VEGA-Optimierungen liefern in jedem Fall gültige Lösungen. Dies ist sehr wichtig für den praktischen Einsatz. Bei einer Verteilungsoptimierung mit 12 Spannern (keine Plazierungsprobleme) und einem mittelkomplexen Eingabepolygon kann die Laufzeit von 5s für die VEES- und VEGA-Optimierung auf einem Pentium P-200 Rechner mit 64MByte RAM eingehalten werden.

Die vorgestellte Spezifikation des Problems vereinfachte die reale Problemstellung. Es wurden einige Aspekte der Spannmitteloptimierung nicht einbezogen. Zum Beispiel sollte genaugenommen die Form des Spanners nicht nur zweidimensional betrachtet werden. Außerdem wurden einige Einschränkungen hinsichtlich der Eingaben gemacht. Konsolentische, die in der Realität auch auftreten können, wurden beispielsweise noch nicht betrachtet.

Es hat sich gezeigt, daß Evolutionäre Algorithmen sehr gut auf das Spanneroptimierungsproblem angewendet werden können. Aufgrund der Tatsache, daß keine konventionelle Methode bekannt ist, dieses Optimierungsproblem zu lösen, scheint die Wahl einer evolutionären Strategie sinnvoll. Die Erweiterung der Arbeit und die Miteinbeziehung der noch nicht behandelten Aspekte der Spannmitteloptimierung ist notwendig. Allerdings scheint dies mit den gewonnenen Erkentnissen nun zukünftig möglich zu sein.

Zusammenfassend läßt sich sagen: Diese Arbeit sollte vor allem andeuten, daß bei vielen Optimierungsproblemen der Einsatz von Evolutionären Algorithmen eine echte Alternative zu anderen konventionellen Verfahren darstellt. Oft sind sie sogar den algorithmischen Verfahren überlegen. Speziell auf dem Gebiet der geometrischen Optimierungsprobleme können die hier gewonnenen Kenntnisse und Verfahren nützlich sein und eventuell teilweise übernommen werden. Beispielhaft soll hier die Verschnittoptimierung als weiteres industrielles Anwendungsgebiet genannt werden.

Literatur

[Bäc92] Thomas Bäck. A User's Guide to GENEsYs 1.0. Technical report, University of Dortmund, Department of Computer Science, System Analysis Research Group, 1992.

[Bau95] Roland Baumann. Verteilte Genetische Algorithmen auf dem MIMD-Parallelrechner Intel Paragon. Master's thesis, Universität Stuttgart, 1995.

[BZ96] Roland Baumann and Andreas Zell. Distributed Genetic Algorithms on the Intel Paragon, a large MIMD Computer. In *Proc. of the 5th Ann. Conf. on Evolutionary Programming, San Diego, CA*, 1996.

[Dam90] Foley/van Dam. *Computer Graphics Principles and Practice*. Addison Wesley, 1990.

[Gol89] David E. Goldberg. *Genetic algorithms in search, optimization and machine learning.* Addison-Wesley, 1989.

[Hol75] John H. Holland. *Adaption in Neural and Artificial Systems*. The University of Michigan Press, 1975.

[Hol93] John H. Holland. *Adaption in natural and artificial systems; an introduction analysis with applications to biology, control and artificial intelligence*. MIT Press, 1993.

[Mic96] Z. Michalewicz. Applications of evolutionary algorithms for constrained problems. In *Parallel Problem Solving from Nature - PPSN IV, Proceedings*, pages 245–254. Berlin: Springer, 1996.

[Nis97] Volker Nissen. *Einführung in Evolutionäre Algorithmen*. Vieweg-Verlag, 1997.

[Ous95] John K. Ousterhout. *TCL und TK*. Addison-Wesley, zweite unveränderte Auflage, 1995.

[Par94] Jan Paredis. Co-evolutionary constraint satisfaction. In *Parallel Problem Solving from Nature - PPSN III, Proceedings*, pages 46–55. Berlin: Springer, 1994.

[Rec73] Ingo Rechenberg. *Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. PhD thesis, TU Berlin, Fakultät für Maschinenwesen, Oktober 1973.

[Rec94] Ingo Rechenberg. *Evolutionsstrategie 1994, Band 1 von Werkstatt Bionik und Evolutionstechnik.* Frommann-holzboog-Verlag, 1994.

[Sch94] Eberhard Schöneburg. *Genetische Algorithmen und Evolutionsstrategien*. Addison-Wesley, 1994.

[Sch95] Hans-Paul Schwefel. *Evolution and optimum seeking*. Sixth-Generation Computer Technology Series. John Wiley & Sons, Inc. , 1995.

[Sch97] HOMAG Maschinenbau A.G. Schopfloch. *WoodWOP Betriebsanleitung*, v6.4 edition, 1997.

[Wak95] Jürgen Wakunda. Verteilte Evolutionsstrategien auf dem Supercomputer Intel Paragon. Master's thesis, Universität Stuttgart, 1995.

[WZ96] Jürgen Wakunda and Andreas Zell. Vorstudie über die Anwendung von Evolutionären Algorithmen bei der Saugerpositionierung für Homag-Bearbeitungszentren. Technical report, WSI für Informatik Tübingen, Lehrstuhl Rechnerarchitektur, November 1996.

[WZ97] Jürgen Wakunda and Andreas Zell. EvA - A Tool for Optimization with Evolutionary Algorithms. In *Proceedings of the 23rd EUROMICRO Conference; Budapest, Hungary*, 1997.