Positionierung von Spannmitteln zur Werkstückfixierung mittels Evolutionärer Algorithmen

Thomas E. Koch, Ralph Müller, Franz Schneider, Andreas Zell

Wilhelm-Schickard-Institut für Informatik der Universität Tübingen

HOMAG Maschinenbau AG, Schopfloch

Dipl.-Inform. Thomas E. Koch ist seit 1998 wissenschaftlicher Mitarbeiter am Lehrstuhl Rechnerarchitektur bei Prof. Dr. Andreas Zell. Er studierte Informatik in Tübingen und Brisbane, Australien. Sein aktuelles Forschungsinteresse sind Evolutionäre Algorithmen.

Adresse:
koch@informatik.uni-tuebingen.de
www-ra.informatik.uni-tuebingen.de

Prof. Dr. Andreas Zell leitet seit 1995 den Lehrstuhl Rechnerarchitektur am Wilhelm-Schickard-Institut für Informatik der Universität Tübingen. Nach dem Studium der Informatik und Mathematik an der Universität Kaiserslautern und einem DAAD-Stipendium an der Stanford University promovierte er 1989 an der Universität Stuttgart und habilitierte sich dort 1994 über Simulation neuronaler Netze. Seine aktuellen Forschungsinteressen sind künstliche neuronale Netze, evolutionäre Algorithmen, mobile Roboter und Bioinformatik-Anwendungen. Er koordiniert den neuen Studiengang Bioinformatik an der Universität Tübingen.

Adresse:
WSI/RA
Köstlinstr.6
72072 Tübingen
zell@informatik.uni-tuebingen.de
www-ra.informatik.uni-tuebingen.de

Dipl.-Inform. Ralph Müller arbeitet seit 1996 bei der Fa. Homag AG, Schopfloch in der Softwareabteilung als Entwickler. Er studierte Informatik an der Universität Stuttgart. Bei Homag ist er verantwortlich für die Bereiche automatische Spannmitteloptimierung, Integration von Programmiersystemen in CAD-Systeme (CAD/CAM) und Projektierung komplexer Anbindungen.

Adresse:
tb10mra@dehomag.e-mail.com
www.homag.com

Dipl.-Ing. Franz Schneider leitet seit 1991 die Softwareabteilung der Fa. Homag AG, Schopfloch. Er studierte Maschinenbau an der Universität Stuttgart. Bei Homag arbeitet er seit 1989, zuerst in der Applikations-Entwicklung, heute ist er für die Entwicklung von Programmiersystemen für die Holzverarbeitungsmaschinen und deren datentechnische Intergration in ERP- und CAD-Umgebungen verantwortlich.

Adresse:
HOMAG Maschinenbau AG
Homagstrasse 3-5
72296 Schopfloch
Franz.Schneider@dehomag.e-mail.com
www.homag.com

Ein komplexes, geometrisches Optimierungsproblem aus der holzbearbeitenden Maschinenbaubranche wird mit Hilfe Evolutionärer Algorithmen gelöst. Dabei werden drei Ansätze aufgezeigt, getestet und bewertet: Evolutionsstrategien, Genetische Algorithmen und Co-Evolution in Verbindung mit Geometrie-Algorithmen. Die Ergebnisse überzeugen sowohl durch die Qualität der gefundenen Lösungen, als auch durch die Einfachheit der Formulierung der zugrunde liegenden Gütebewertung.

Optimal Positioning of Clamps for Workpiece Adjustment using Evolutionary Computation

A problem of geometric optimization known in woodworking industries is solved by Evolutionary Computation. Three approaches are shown, evaluated and discussed: Evolution Strategies, Genetic Algorithms and Co-Evolution in combination with Geometric Algorithms. The results satisfy by the quality of solutions found as well as by the simplicity of formulation of underlying quality measure.

1 Einleitung

Ein geometrisches Optimierungsproblem aus der holzbearbeitenden Maschinenbaubranche wird mit Hilfe Evolutionärer Algorithmen [KurSch97] [Nissen94] [Bäck96] gelöst. Es ist ein Gemeinschaftsprojekt zwischen dem Wilhelm-Schickard-Institut für Informatik der Universität Tübingen und der HOMAG Maschinenbau AG, Schopfloch.

Um Holzplatten automatisiert bearbeiten zu können, müssen sie zuerst auf einem Bearbeitungstisch fixiert werden. Zur Fixierung werden Spanner und/oder Sauger (engl. clamp) eingesetzt; diese halten einerseits (oben) das zu bearbeitende Holzwerkstück, andererseits (unten) sind sie selbst mit dem Bearbeitungstisch verbunden. Bild 1 zeigt ein HOMAG-Bearbeitungszentrum mit Flachtisch, einem zu bearbeitenden Holzstück und vier Vakuumsaugern als Spannmittel. Je nach Art des Spannmittels ist der Fixierungsmechanismus fest (Klemmen) oder durch ein Vakuum (Saugen) realisiert. Das Ziel ist nun die optimale Positionierung oder Plazierung dieser Spanner in beschränkter Zeit, so daß das dadurch fixier-

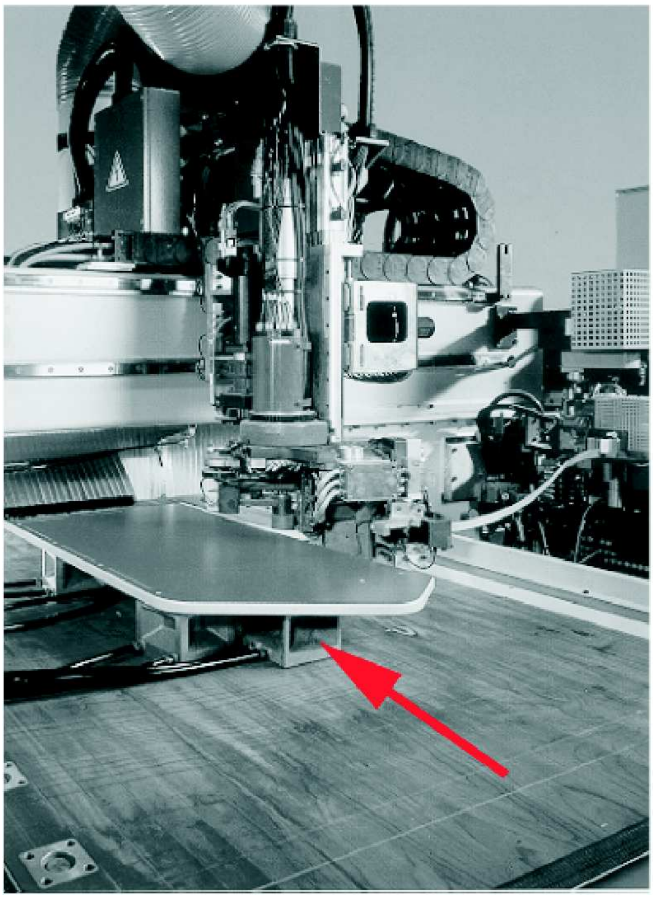

Bild 1: HOMAG-Holzbearbeitungszentrum mit Flachtisch und Vakuumsaugern als Spannmittel (roter Pfeil), die die zu bearbeitende Holzplatte fixieren.

te Holzstück während der Bearbeitung (Fräsen, Bohren, Leimen, etc.) nicht verrutscht oder zu flattern beginnt und die Spannmittel die Bearbeitung nicht behindern. Weitere Kriterien sind etwa, die Spannmittel möglichst nah an Bearbeitungswege zu setzen, um eine maximale Stabilität zu garantieren. Diese Optimierung soll bei Bearbeitungszentren eingesetzt werden, die nicht hohe Losgrössen (z.B. Regalböden) produzieren, sondern jedes Produkt individuell herstellen (Stückzahl 1). Dadurch ist für jedes Werkstück die Positionierung der Spannmittel neu zu berechnen, und zwar in beschränkter Zeit, um die Maschineneffizienz nicht zu vermindern.

Im praktischen Einsatz werden die Spannmittel bisher manuell programmiert; optional kann auch noch ein Vorschlag der proprietären HOMAG-Software erstellt werden, der einen einfachen Greedy-Algorithmus verwendet. Das praktische Ziel ist, die Spannmittel automatisiert zu setzen. Dazu ist eine Qualitätsbewertung der möglichen Lösungen aufzustellen und das globale Optimum dieser Funktion im Lösungsraum zu bestimmen. Bei z.B. 49 möglichen Positionen und 6 zu positionierenden Spannmitteln gibt es (6 aus 49) Konstellationen. Klassische, algorithmische Optimierungs-Verfahren (Gradienten, Simplex, ...) scheiden wegen des hochdimensionalen Parameterraumes, der Existenz vieler lokaler Optima sowie einer meist vorgegeben Suchstrategie aus. Evolutionäre Algorithmen hingegen versuchen entsprechend den Vorgaben aus der Biologie, die Qualität der aktuellen Lösungen durch quasi-paralleles „wandern“ im Lösungsraum zu verbessern.

Das potentielle Anwendungsspektrum Evolutionärer Algorithmen im Bereich der Optimierung ist groß. In vielen Fällen, in denen für ein Optimierungsproblem keine effizienten konventionellen Algorithmen bekannt sind, koönnen Evolutionäre Algorithmen erfolgreich eingesetzt werden. [Schöne94] und [Hafner98] stellen einige praktische Anwendungen vor. So ist auch die hier vorgestellte Spannmitteloptimierung zu komplex, um konventionelle Verfahren anwenden zu können.

2 Problemstellung

Bild 2 zeigt die Spannmittel-Optimierung mit Ein- und Ausgaben, Restriktionen und Güte-Kriterien.

2.1 Ein- und Ausgaben der Optimierung

Alle Eingaben für ein mögliches Berabeitungsszenario können über eine interaktive, graphische Benutzeroberfläche der HOMAG-Software WoodWOP [1] eingegeben und manipuliert werden; das heißt, der gesamte Bearbeitungsablauf der Maschine wird von WoodWOP zusammengestellt, daraus ein NC-Programm generiert, das diese dann abarbeitet. Die Formen der Bearbeitungsspuren der Maschine sowie die Formen des Werkstückes, des Maschinentisches und der Spannmittel liegen als Polygonzüge nach einer CNC-Simulation der NC-Daten vor. Außerdem steht eine unabhängige Testumgebung, die bestimmte, ausgesuchte Daten (aus WoodWOP generiert) benutzt, bereit. Zur Vereinfachung wird nur der zwei-dimensionale Raum betrachtet:

- Werkstückformen und Bearbeitungsspur: jeweils $1..n$ sich nicht schneidende Polygone; diese definieren „erlaubte“ und „verbotene“ Flächen.
- Bearbeitungstisch: Grundsätzlich gibt es zwei verschiedene Tischarten. Bei den Maschinentischen mit automatisch verfahrbaren Konsolen werden die Spannmittel auf den Konsolen plaziert; dadurch sind sie in ihrer Bewegungsfreiheit durch die Bewegung der Konsolen eingeschränkt ($\rightarrow$ weitere Restriktion). Bei Flachtischen (siehe Bild 1) kann jedes Spannmittel eine beliebige Position mit beliebiger Drehung einnehmen. Bisher wurden nur Flachtische betrachtet. Die Maße des Bearbeitungstisches sind durch Länge und Breite eindeutig bestimmt; diese stellen auch Schranken für den Wertebereich der gesuchten Spannmittelpositionen dar.
- Spannmittelform: 1 beliebiges konvexes Polygon. (Zusätzliche Schwierigkeiten wären die Unterscheidung von Ober- und Unterseite)
- Spannmittelanzahl: $3..n$ Spannmittel können positioniert werden. Die Anzahl kann im Laufe des Algorith-

[1] WoodWOP: WOP steht für Werkstatt-Orientiertes Programmieren

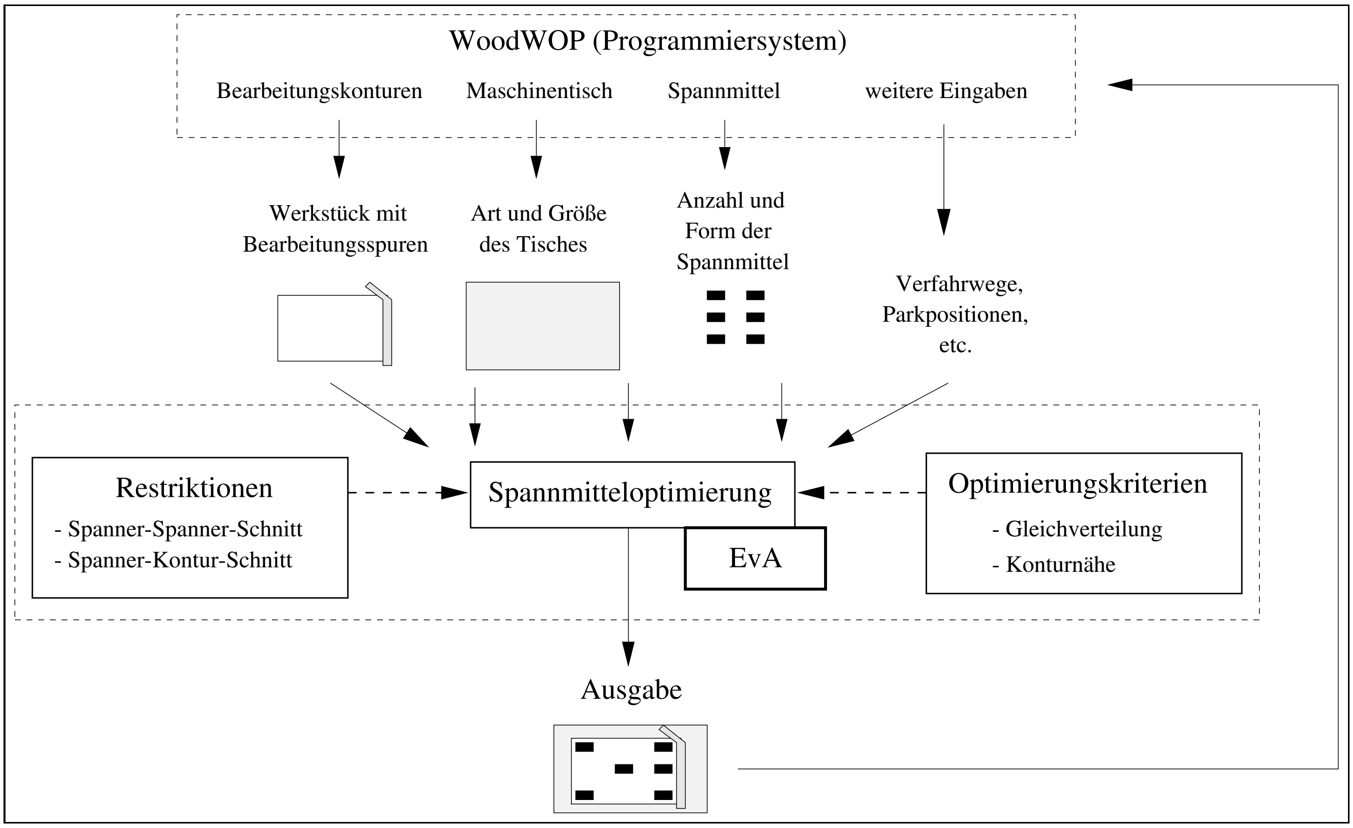

Bild 2: Spannmittel-Optimierungsproblem: die Saugerotpimierung, die das Programm-Paket EvA (Evolutionäre Algorithmen) [WakZell97] als „Black Box“ benutzt, verarbeitet Eingabedaten aus der Anwendung WoodWOP zu einer optimalen Positionierung der Spannmittel.

mus verringert werden, falls die gewünschte Anzahl keinen Platz findet.

- Weitere Eingaben wie Verfahrwege und Parkpositionen der Spannmittel sind noch nicht berücksichtigt.

Die Ausgabe beschreibt den Erfolg bzw. Mißerfolg des Laufes sowie die beste gefundene Lösung.

2.2 Restriktionen und Gütekriterien

Die Optimierung wird einerseits durch Restriktionen und andereseits durch Gütekriterien gesteuert. Folgende Restriktionen sind eizuhalten:

- Die Spannmittel sind innerhalb der Werkstückform zu plazieren.
- „Verbotene“ Flächen, definiert durch die Bearbeitungsspur, dürfen weder berührt noch geschnitten werden. Eventuell ist ein Sicherheitsabstand einzuhalten.
- Die Spannmittel dürfen sich nicht gegenseitig behindern.

Ein wichtiges Kriterium für die Praxistauglichkeit des Optimierungsverfahrens ist die Zeit. Die Berechnung darf auf einem Pentium-200-Rechner nicht länger als 5 Sekunden dauern, da ein Einsatz in Echtzeit geplant ist.

Folgende Gütekriterien bewerten gefundene Lösungen:

- Die Spannmittel sollen nah an der Bearbeitungsform sein.
- Die Werkstückform soll gleichmäßig von Spannmitteln überdeckt sein.
- Die Anzahl der Spannmittel soll ausreichend sein.

Die Gütekriterien stehen in der realen Anwendung in Konflikt miteinander.

3 Geometriealgorithmen

Die implementierten Optimierungen verwenden verschiedene Geometriealgorithmen [Foley95], um die Güte-Kriterien zu bestimmen und die Restriktionen einzuhalten. Wie in Abschnitt 2.1 beschrieben, arbeitet man vor allem mit Polygonen, die sehr komplex sein können (bis zu 200 und mehr Ecken). Insbesondere der Schnitt zweier Polygone muß wegen der Zeitrestriktion sehr effizient durchgeführt werden. Deswegen wird der Scan-Line-Algorithmus [Foley95] eingesetzt, der nicht immer exakte Ergebnisse liefert, jedoch sehr effizient ist.

Der Algorithmus wird einerseits für die Schnittberechnung des Spannmittels mit Werkstück-/Bearbeitungsform (Spanner-Kontur-Schnitt) sowie der Spannmittel untereinander (Spanner-Spanner-Schnitt) verwendet. Diese Berechnungen müssen für jede neue Lösung angestellt werden. Da die Werkstück- und Bearbeitungsformen sich nicht ändern, kann der Spanner-Kontur-Schnitt in einem Vorverarbeitungsschritt effizienter gemacht werden.

Für den Spanner-Kontur-Schnitt werden die Polygone der Werkstück- und Bearbeitungsform mittels einer Scanlinie rasterisiert. Dabei wird festgestellt, welche Rasterquadrate im gültigen bzw. ungültigen Bereich liegen und in einem booleschen ScanArray gespeichert. Gültige Bereiche sind innerhalb der Werkstück- und außerhalb der Bearbeitungsform. Die Auflösung des Rasters kann dabei ebenfalls stufenlos eingestellt werden: bei feinerer Granularität wird das Ergebnis zwar genauer, aber die Effizienz geringer. Der Spanner-Spanner-Schnitt wird entsprechend dynamisch für jedes Spanner-Paar -mit einem vorgeschalteten Abstandstest- berechnet.

Ein wesentlicher Nachteil der Scankonvertierung ist die Diskretisierung des Problems und die damit auftretenden Aliasing-Effekte, die bei einer Unterabtastung der Polygone entstehen: wird der Maßstab zu groß gewählt, gehen Formfeinheiten oder sogar kleine Polygone verloren [2].

Durch die Geometriealgorithmen werden folgende Terme berechnet:

- SK_i ist der Schnitt von Spanner i mit unerlaubten Flächen (Spanner-Kontur-Schnitt)
- SS_i ist die Summe der Schnittflächen von Spanner i mit anderen Spannern (Spanner-Spanner-Schnitt)

SK_i und SS_i sind reellwerige Größen mit $0 \leq SK_i \leq 1$ und $0 \leq SS_i \leq 1$. Sie geben jeweils die anteilig zur Gesamtfläche verletzte Fläche des Spanners i an. Daraus ergibt sich der erste Teilterm für das Güte-Kriterium einer Lösung:

$$I = \sum_{i=1}^{n} SS_i \tag{1}$$

wobei n die Anzahl der Spannmittel ist. Der Term I (engl. Intersection) beschreibt den Spanner-Spanner-Schnitt einer Lösung. Dieser Teilterm muß gleich 0 sein, da sich schneidende Spannmittel reell nicht plaziert werden können, und wird deshalb als negatives Güte-Kriterium (Strafterm) in der Fitnessfunktion gewertet. Entsprechend werden die SK_i als Strafterme bzw. Ausschlußkriterien gewertet.

4 Optimierungsalgorithmen

Zur Optimierung mit Evolutionären Algorithmen wurden drei Ansätze realisiert, die sich im wesentlichen durch die Restriktionen und die Darstellung der Individuen unterscheiden. Als Bibliothek wird das Programm-Paket Evolutionäre Algorithmen (EvA, ebenfalls am Lehrstuhl entwickelt) [WakZell97] eingesetzt. EvA enthält die Module VeGA (Verteilte Genetische Algorithmen) [BauZell96] und VeES (Verteilte Evolutionsstrategien) [Wakund95], die sowohl auf einem Einzelplatzrechner als auch mittels Message Passing Interface (MPI) verteilt lauffähig sind Verfügbare Plattformen sind fast alle Unixe und Windows NT. EvA ist in C/C++ implementiert und hat eine Tcl/Tk-Schnittstelle und eine graphische Benutzeroberfläche.

4.1 Gemeinsame Teilterme

Das Güte-Kriterium setzt sich aus mehreren Teiltermen zusammen, die bei den realisierten Ansätzen unterschiedlich berechnet und bewertet werden. Alle Ansätze benutzen folgende Teilterme.

4.1.1 Initialisierung: Auffinden gültiger Positionen

Alle drei realisierten Verfahren benötigen zur Initialisierung gültige Lösungen. Diese „sicheren“ Positionen werden in einem Vorverarbeitungsschritt gesucht. Dabei werden die gefundenen Spannmittelpositionen gleich in Eck-, Kanten- und Innenpositionen kategorisiert. Anhand der Eingabepolygone der Bearbeitungs- und Werkstückform werden Kanten- und Eckpositionen durch „Herantasten“ an Ecken und Kanten ermittelt. Dadurch erhält man maximal für jede Ecke eine gültige Position; die Kanten werden in n Teile gegliedert, um maximal n Positionen pro Kante zu erhalten (Die Länge eines Teilstückes ist gleich der halben Ausdehnung eines Spannmittels. Die Innenpositionen werden relativ einfach durch Auflegen eines gleichmäßigen Gitters über den Bearbeitungstisch bestimmt. Die Granularität des Gitters kann eingestellt werden (enger $\rightarrow$ mehr Innenpositionen). Falls nicht mindestens drei Positionen gefunden werden, wird dies als Abbruchkriterium interpretiert, da keine Spannmittel plaziert werden können.

4.1.2 Gleichverteilung der Spannmittel

Um das Güte-Kriterium einer möglichst großen Gleichverteilung zu erreichen, werden zuerst die Abstände aller Spannmittel einer Lösung berechnet und in der Menge M gespeichert:

$$M = \bigcup_{i<j} dist(S_i, S_j) = \{d_1, d_2, \ldots, d_{|M|}\}$$

$$|M| = \frac{n(n-1)}{2}$$

dabei n die Anzahl der Spannmittel,
S_i der Referenzpunkt des Spannmittels i, $0 \leq i < n$,
$dist(P_i, P_j)$ der Abstand der Punkte P_i und P_j,
M die Menge aller dieser Abstände,
$|M|$ die Anzahl der Elemente von M.
Außerdem sei M' die aufsteigend sortierte Menge M:

$$M' = \{d'_1, d'_2, d'_3, ..., d'_{|M|} | d'_i \leq d'_j \Leftrightarrow i < j\}$$

Ziel ist es nun, die Summe dieser Abstände zu maximieren. Dies entspricht durch die Wurzelbewertung einer Maximierung des Gleichverteilungsmaßes U (engl. Uniform Distribution). Liegt ein Spannmittel z.B. zwischen zwei anderen, dann ist U dann am Größten, wenn das Spannmittel in der Mitte liegt.

$$U = \sum_{i=1}^{k} \sqrt{d'_i} \qquad 0 \leq k < |M| \tag{2}$$

[2] Bei einem Maßstab von 5:1 (5mm in der Realität entsprechen 1 Rasterquadrat) geht ein Loch der Größe von $3mm \cdot 3mm$ verloren. Dieses muß dann mindestens als Loch der Größe $5mm \cdot 5mm = 1$ Rasterquadrat simuliert werden.

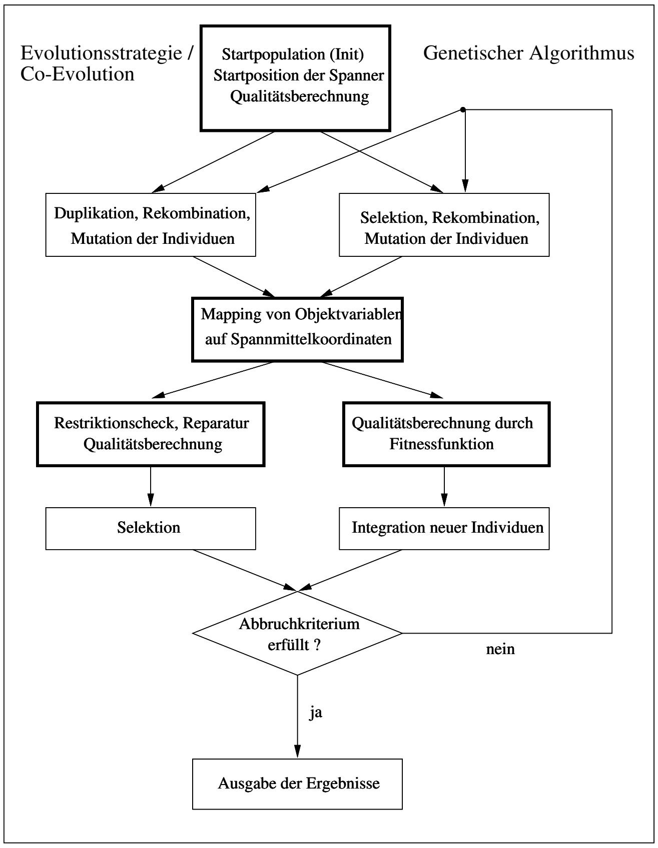

Bild 3: Ablauf der Evolutionären Algorithmen: links eingebettet in Evolutionsstrategien (ES, CoEv), rechts in Genetische Algorithmen (GA).

Die Variable k gibt hier an, daß nicht unbedingt alle Elemente der Menge M' aufsummiert werden, sondern nur die k kleinsten Abstände. Dies führt zu besseren Ergebnissen, als alle Abstände aufzuaddieren oder nur den nächsten Nachbarn zu betrachen, da sich eine Topologie bildet, in der einzelne, weiter entfernte Spannmittel zunächst nicht betrachtet werden. In den Implementierungen wurde $k = n$ gewählt.

4.1.3 Elimination von Spannmitteln

In der Praxis kann der Fall auftreten, daß die gewünschte Anzahl n von Spannmittel nicht plaziert werden kann, da die erlaubte Fläche nicht ausreicht. So ist es möglich, Spannmittel im Laufe der Optimierung „herauszunehmen“: zuerst wird die erlaubte Fläche durch die Fäche eines Spannmittels dividiert, um die maximale Anzahl zu setzender Spannmittel zu bestimmen; ist diese kleiner als n, wird n der neue Wert zugewiesen; im Laufe der Optimierung wird nun schrittweise ein Spannmittel herausgenommen, falls sich Spannmittel nach Term I (1) schneiden. Es ist jedoch darauf zu achten, dem Algorithmus genügend viele Generationen zwischen zwei Eliminationen zu geben, um sich zu „regenerieren“.

4.2 Ansatz mit Evolutionsstrategien

Das erste Verfahren benutzt Evolutionsstrategien [Rechen94][Bäck96]. Die hier verwendete Evolutionsstrategie definiert ein Individuum als n zu positionierende Spannmittel, die wiederum durch ihre drei Freiheitsgrade im zwei-dimensionalen Raum (x-, y-Position und Winkel) definiert sind (Beispiel: $3 \cdot 10$ Spannmittel $= 30$ Objektvariablen bilden ein Individuum). Ein Individuum ist somit eine Lösung des Problems. Der eigentliche Algorithmus - eingebettet in VeES - ist in Bild 3 links dargestellt.

Initialisiert wird die Startpopulation, bestehend aus meist 20 Individuen mit gültigen Positionen wie in Abschnitt 4.1.1 beschrieben. Die eigentlichen Evolutionsschritte wie Selektion, Mutation und Rekombination, die die aktuelle Population in die nächste Generation überführen, laufen in der black-box von VeES ab. Die Nachkommen einer Population, die Kind-Individuuen, werden danach einem Restriktionscheck unterzogen, daraus erkannte ungültige Lösungen verworfen und das Individuum auf eine alte Lösung zurückgesetzt (Evolutionsstrategie mit Reparatur). Daraufhin erfolgt die Qualitätsberechnung der Population und deren Individuuen durch die Gütebewertung. Das Abbruchkriterium ist entweder eine fest vorgegebene Anzahl von Iterationen (Generationen) oder die Konvergenz der Gütebewertung.

Um die Konturnähe der Spannmittel zu gewährleisten, wird ein „Checkpoint“-Term definiert: die Idee ist, Punkte (Checkpoints) auf den Werkstück- und Bearbeitungskonturen zu plazieren. Für jeden Checkpoint wird der quadratische Abstand zum nächsten Spannmittel berechnet; danach werden alle diese Abstände aufsummiert zum Konturnäheterm O_1 (engl. Outline):

$$O_1 = \sum_{i=1}^{m} \min_{0 \le j < n} [dist(CP_i, S_j)]^2 \qquad (3)$$

wobei m die Anzahl der Checkpoints CP_i, n die Anzahl der Spannmittel, S_j der Mittelpunkt von Spannmittel j und $dist(P_i, P_j)$ der Abstand der Punkte P_i und P_j. Es ist bei der Bestimmung der Checkpoints darauf zu achten, diese geschickt zu verteilen, um dadurch eine gute und gleichmäßige Verteilung der Spannmittel an den Konturen zu erhalten. Die Anzahl der Checkpoints kann variiert werden.

Die Qualität einer Lösung wird durch die gewichteten Teilterme Spanner-Spanner-Schnitt I (1) (kleiner ist besser), Gleichverteilung U (2) (größer ist besser) und die Kontur-(Checkpoint-) Nähe O_1 (3) (kleiner ist besser) bestimmt:

$$F_{ES} = \alpha \cdot I - \beta \cdot U + \gamma \cdot O_1 \qquad \rightarrow \text{Min!} \qquad (4)$$

wobei α, β und γ Gewichtungsfakoren der entsprechenden Teilterme sind. Diese sind durch Test und subjektive Auswertung zu bestimmen. Insgesamt soll der Fitnessterm F_{ES} minimiert werden, das heißt ein kleinerer Wert stellt eine bessere Güte dar.

4.3 Ansatz mit Co-Evolution

Dieser Ansatz verwendet ebenfalls Evolutionsstrategien mit dem Unterschied, daß ungültige Lösungen nicht repariert und zurückgesetzt, sondern bestraft werden. Der Algorithmus arbeitet hier auf zwei Populationen: einer Lösungspopulation wie im vorhergehenden Ansatz und einer Restriktionspopulation, die eine inverse Fitnessinteraktion verbindet. Diese Raubtier-Beute-Beziehung wird Co-Evolution genannt [Paredi94] [Michal96]. Die beiden Populationen sollen sich so gegenseitig hochschaukeln: entwickelt das Raubtier bessere Angriffsstrategien, reagiert die Beute mit besseren Verteidigungsstrategien.

Allgemein ist die Gewichtung der Verletzung der Restriktion eines Individuums folgendermaßen definiert:

$$g(i,r) \begin{cases} = 0 : \text{Individuum } i \text{ hält Restriktion } r \text{ ein} \\ > 0 : \text{sonst} \end{cases}$$

$$C_r = \text{Gewichtung der Verletzung der Restriktion } r \quad \text{mit } 0 \leq r < 2n$$

C_r wird auf einen initialen Wert gesetzt (relativ groß, um anfangs nur gültige Lösungen zuzulassen) und danach je nach Restriktionsverletzung oder -einhaltung erhöht bzw. erniedrigt. Die einzelnen Restriktionen des Spannmittelproblems bestehen aus den Teiltermen Spanner-Spanner-Schnitt *SS* und Spanner-Kontur-Schnitt *SK*:

$$g(i,r) = SS_r \quad \forall\, 0 \leq i < ind, \forall\, 0 \leq r < n$$

$$g(i,n+r) = SK_r \quad \forall\, 0 \leq i < ind, \forall\, 0 \leq r < n$$

wobei *ind* die Anzahl der Individuen einer Population, *n* die Anzahl der Spannmittel (eines Individuums) sind. Der Strafterm für die zu bestimmende Fitnessfunktion eines Lösungsindividuums ist die gewichtete Summe der Fitnesswerte der Restriktionen:

$$S = S_i = \sum_{j=1}^{2n} C_j \cdot g(i,j) \quad \forall\, 0 \leq i < ind$$

Die Gewichtung C_r wird nun nach folgender Strategie verändert: wird eine Restriktion häufig verletzt, wird die Gewichtung erhöht, die Verletzung wird also zunehmend bestraft und der Algorithmus wird versuchen, die Verletzung zu vermeiden. Bei der Co-Evolution wird im Vergleich zum Evolutionsstrategieansatz der Restriktionsterm I -erweitert um die Spanner-Kontur-Schnitte (SK_i)- durch den Strafterm S ersetzt.

$$F_{CoEv} = \alpha \cdot S - \beta \cdot U + \gamma \cdot O_1 \qquad \to \text{Min!} \tag{5}$$

Die Fitnessinteraktion zwischen Individuen und ihren Restriktionen führt zu einer konkurrierenden Co-Evolution. Sobald die Individuen auf einer Teilmenge der Restriktionen erfolgreich sind, erhalten sie eine geringere Fitness, die restlichen, noch nicht eingehaltenen Restriktionen jedoch eine stärkere Gewichtung. Dadurch werden die Individuen versuchen, nun diese einzuhalten etc.

4.4 Ansatz mit Genetischen Algorithmen

Genetische Algorithmen [Hollan93] werden hier eingesetzt, um aus den in Abschnitt 4.1.1 bestimmten diskreten Positionen die optimale Konstellation zu finden. Wie zum Beispiel beim Lotto-Spiel (6 aus 49) wäre hier das Ziel, die beste Konstellation von 6 Spannmitteln aus 49 möglichen Positionen zu finden.

Die gefundenen Positionen werden zwei-dimensional sortiert, um das Prinzip der starken Kausalität einzuhalten. Dieses besagt, daß eine kleine Änderung des Gen-Codes (etwa eine Mutation von Position $(42,15)$ auf $(43,15)$) auch eine kleine Änderung der Bewertung nach sich zieht (Position $(42,15)$ ist Nachbar von $(43,15)$). Dadurch ist eine bessere Konvergenz des Algorithmus gewährleistet.

Die Konturnähe der Spannmittel wird hier nicht durch das Checkpoint-Verfahren des ES-Ansatzes erreicht, sondern indem die einzelnen Kategorien der diskreten Positionen (Eck, Kante oder Innen) unterschiedlich gewichtet werden. Eine Unterscheidung zwischen Werkstück- und Bearbeitungsform wird ebefalls vorgenommen:

$$Gewicht(j) = \begin{cases} w_1 : j \text{ an Werkstückkante} \\ w_2 : j \text{ an Werkstückecke} \\ w_3 : j \text{ an Bearbeitungskante} \\ w_4 : j \text{ an Bearbeitungsecke} \\ w_5 : j \text{ innerhalb} \end{cases}$$

$$O_2 = \sum_{i=1}^{n} Gewicht(i) \tag{6}$$

Der Teilterm O_2 bewertet die Konturnähe eines Individuums (größer ist besser). Dadurch haben wir die Fitnessfunktion, die im Vergleich zur Fitnessfunktion des ES-Ansatzes (4) den Teilterm Kontur-(Checkpoint)-Nähe O_1 (3) durch den negativen, diskreten Konturnäheterm O_2 ersetzt:

$$F_{GA} = \alpha \cdot S - \beta \cdot E - \gamma \cdot O_2 \qquad \to \text{Min!} \tag{7}$$

Die Parameter w_k sind variabel einstellbar. Ist etwa eine Annäherung der Spannmittel an die Bearbeitungskonturen erwünscht, so sind $w_{1,2,5}$ auf 0 und $w_{3,4} > 0$ zu setzen.

5 Ergebnisse

Um ausgiebig testen zu können, wurde eine Testoberfläche in Tcl/Tk implementiert, aus der auch die folgenden Graphiken stammen. Die verwendeten Testfälle sind alle aus der Praxis des Industriepartners HOMAG oder dessen Kunden entnommen.

Insgesamt wurden umfangreiche Tests mit den im vorigen Abschnitt vorgestellten Ansätzen durchgeführt. Als Vergleich diente zusätzlich ein vollständiges Absuchen des gesamten Lösungsraumes, beschränkt durch die sicheren Positionen (siehe Abschnitt 4.1.1), um zumindest ein grobes Wissen über die Lage des globalen Optimums zu haben.

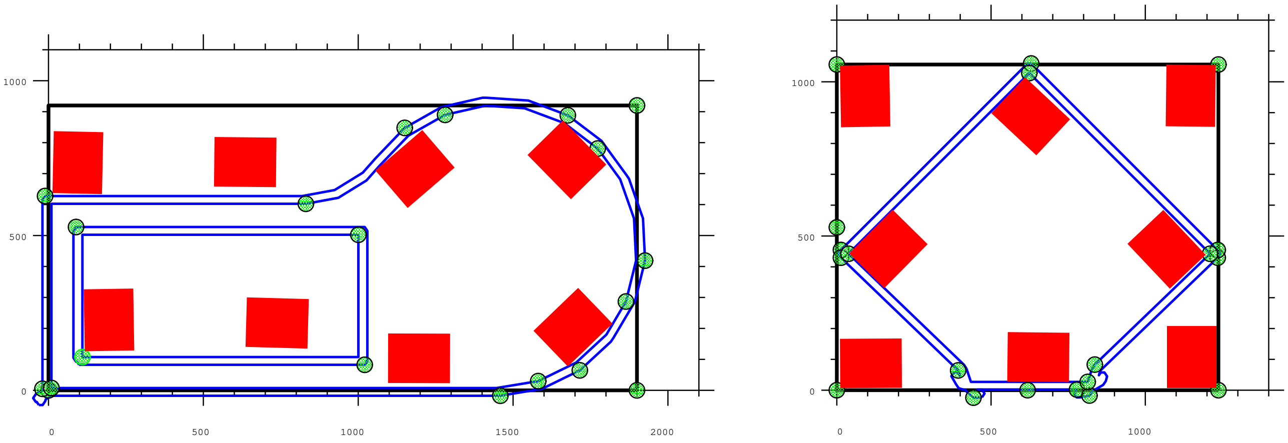

Bild 4: Ansatz mit Evolutionsstrategien: 8 Spannmittel; 400 Generationen; Ergebnis: sehr gut. Das schwarze Rechteck definiert die zu bearbeitende Rohholzplatte, die blauen Polygone die Bearbeitungskonturen und die rot gefüllten Polygone die Spannmittel. Die grünen Kreise zeigen die Checkpoints (Anzahl ist gleich $2.5 \cdot 8$ Spannmittel).

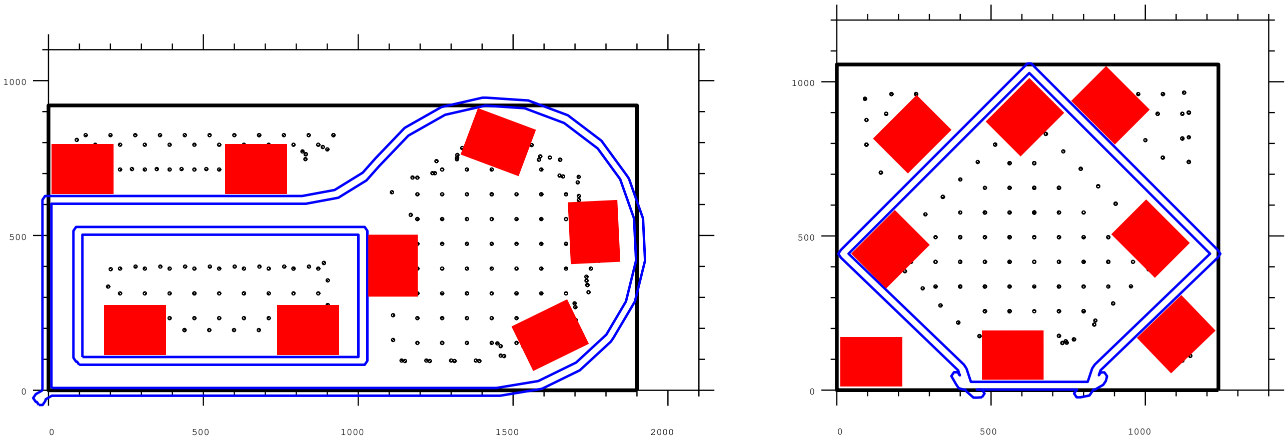

Bild 5: Ansatz mit Genetischen Algorithmen: 8 Spannmittel; 200 Generationen; Ergebnis: links sehr gut, rechts gut. Die schwarzen Punkte zeigen die sicheren Positionen (die Granularität des Gitters beträgt 80 mm).

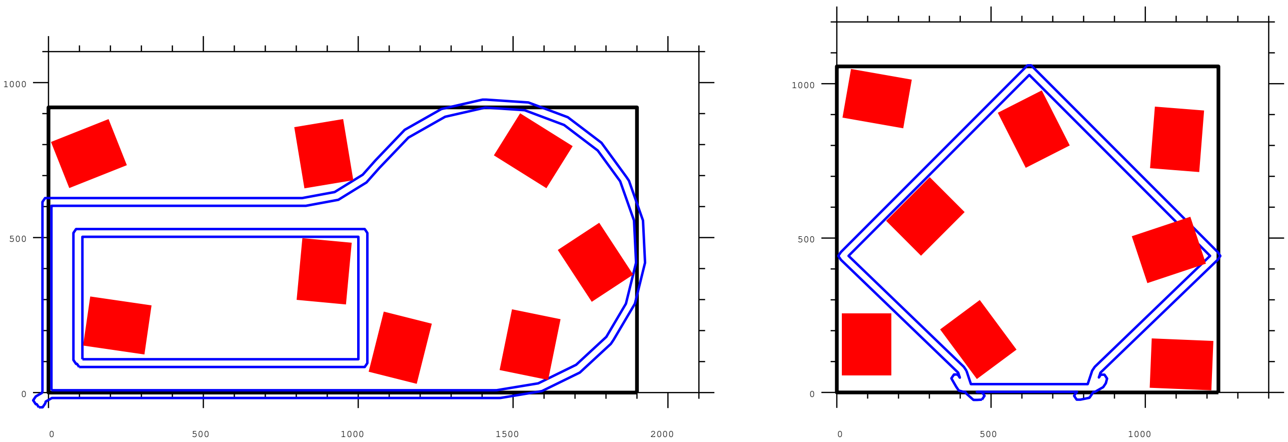

Bild 6: Ansatz mit Co-Evolution: 8 Spannmittel; 200 Generationen; Ergebnis: unbefriedigend.

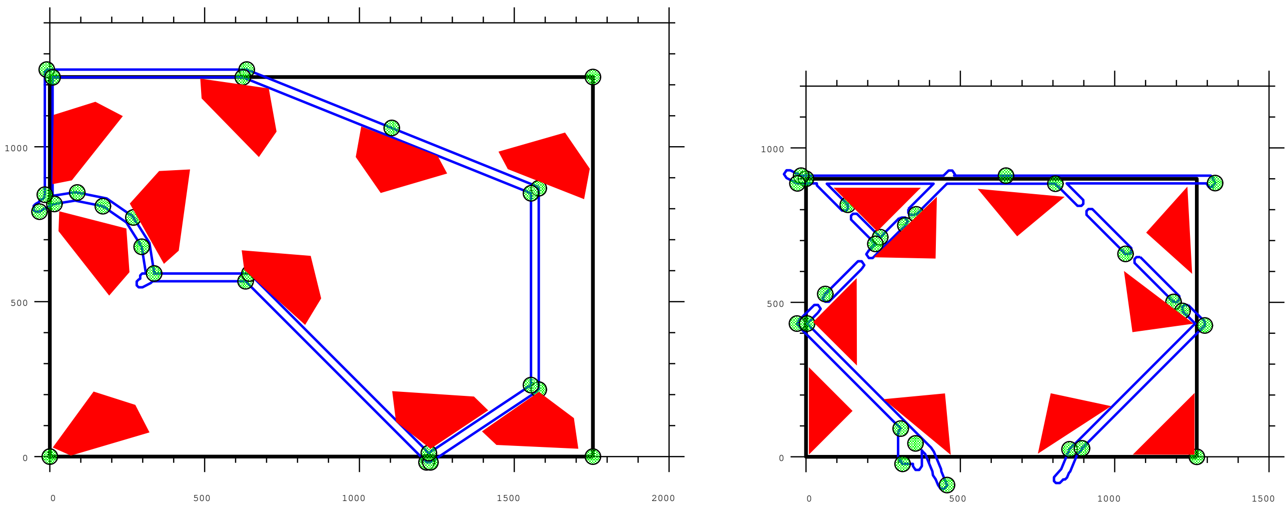

Bild 7: Ansatz mit Eolutionsstrategien: hier werden jeweils zehn Spannmittel eingesetzt, die aber eine andere Form aufweisen und sich sehr gut in die Konturen einpassen.

Jeder einzelne Ansatz ist mit unterschiedlichen Parametereinstellungen wie

- Anzahl der Spannmittel,
- Anzahl der Checkpoints,
- verschiedenen Bearbeitungs- und Werkstückformen und konkurrierender Gewichtung dieser,
- verschiedenen Spannmittelformen,
- Gewichtung der Gleichverteilung kontra der Konturnähe,
- mit und ohne Spannmittelelimination

getestet worden. Die Ansätze untereinander wurden mit gleichen Parametereinstellungen verglichen. Die Evolutionären Algorithmen des EvA Pakets wurden ebenfalls mit deren Parametervariationen getestet. So bietet etwa ES für die Schrittweitenadaption der Objektvariablen mehrere Verfahren an.

Die Tests und deren Beurteilung erwiesen sich sehr schwierig. Dies lag zum einen an den vielen Parametern sowie der Schwierigkeit, die Ergebnisse objektiv zu vergleichen; meist mußte deswegen subjektiv verglichen und beurteilt werden, um eine bessere Parametereinstellung für eine bestimmte Konstellation zu finden. Das Auffinden einer allgemeinen Parametereinstellung für alle verwendeten Testfälle ist nicht möglich.

Die Ansätze mit Evolutionsstrategien und Genetischen Algorithmen zeigen insgesamt ähnliche und gute bis sehr gute Ergebnisse. Durch diese zwei Ansätze ist es möglich, 95% aller Optimierungsprobleme in der gegebenen Zeit zu lösen. Der Vergleich durch Evaluierung aller möglichen Lösungen und deren Fitness zeigt, daß die hier eingesetzten Verfahren das globale Optimum nahezu erreichen. Der Ansatz mit Co-Evolution fällt jedoch deutlich ab. Dies liegt zum einen daran, daß ungültige Lösungen produziert werden und auch die Zeitrestriktion nicht eingehalten wird.

Das größte Problem ist die richtige Einstellung der Parameter der verwendeten black-box-Algorithmen sowie der Fitness-Funktion, um das globale Optimum so zu setzen, daß es für den subjektiven Betrachter auch eine vernünftige Lösung ergibt. So hat sich teilweise herausgestellt, daß die gefundene Lösung zwar im Sinne der Fitness-Funktion sehr gut ist, subjektiv aber nicht das gewünschte Ergebnis erbringt.

Die Bilder 4 bis 6 zeigen Ergebnisse der vorgestellten Ansätze. Zu sehen sind jeweils der skalierte Maschinentisch (schwarz), die Werkstückform (rechteckig, schwarz), die Bearbeitungsformen (blau) und die Spannmittel (rot ausgefüllt). Der jeweils verwandte Ansatz sowie besondere Einstellungen sind in den Bildbeschriftungen enthalten.

Die Beispiele in Bild 7 zeigen die Verwendung anderer Spannmittel, links ein unsymmetrisches, fünfeckiges Polygon und rechts ein symmetrisches Dreieck. Es ist schön zu sehen, wie sich die Spannmittel in ihre nächste Umgebung einpassen. Der Einsatz verschiedener Spannmittel gleichzeitig ist noch nicht realisiert.

Anhand des Beispieles in Bild 8 soll die Spannmittelelimination geprüft werden. Von zwölf ins Rennen ge-

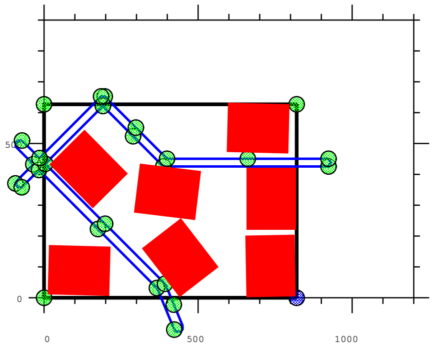

Bild 8: Ansatz mit Evolutionsstrategien und Spannmittelelimination: von den 12 gewünschten Spannmitteln wurden fünf im Laufe des Algorithmus entfernt.

schickten Spannmittel haben sich sieben behauptet. Die Tests der Spannmittelelimination haben gezeigt, daß hier bis zu Faktor 2 mehr Generationen nötig sind, um ein vernünftiges Ergebnis zu erhalten. Dies liegt daran, daß die Algorithmen nach einer Elimination wie in einen Initialzustand gelangen und sich erst wieder regenerieren müssen.

Der Ansatz mit Evolutionsstrategien und Checkpoints setzt auch die Vorgabe um, extreme Positionen wie spitze Ecken und schmale Stellen besonders zu gewichten. Dies erfolgt durch eine stärkere Plazierung von Spannmitteln an solchen Stellen und eine erhöhte Gewichtung des entsprechenden Teilterms in der Fitness-Funktion zur Gütebewertung.

Das gestellte Zeitkriterium von 5 Sekunden wurde mit den Ansätzen Evolutionsstrategie und Genetischen Algorithmen eingehalten. Mit jeweils 200 Generationen, durchschnittlich komplexen Eingabepolygonen und 8 Spannmitteln wurden auf einem Pentium-200 mit 128 MB RAM gute Ergebnisse erzielt. Durch die fortschreitende Hardwareentwicklung ist das Zeitkriterium in Zukunft etwas zu vernachlässigen.

6 Zusammenfassung

Der vorliegende Beitrag zeigt verschiedene Ansätze zur Lösung eines hochkomplexen geometrischen Optimierungsproblems aus der holzbearbeitenden Maschinenbaubranche. Dabei wurden neue Verfahren aus dem Bereich der Computational Intelligence angewendet.

Durch die beschriebenen Ansätze ist es möglich, in Echtzeit (5 Sekunden Restriktion) 95% der zu bearbeitenden Eingabekonstellationen automatisiert zu lösen und somit Spannmittel zur Werkstückfixierung zu positionieren. Die vorgestellten Heuristiken suchen dabei den Lösungsraum nicht nach festen Strategien, sondern quasi-parallel auf mehreren Lösungspfaden ab. Dadurch ist eine größere Effizienz und die Vermeidung eines lokalen Optimums gewährleistet.

Unter den vorgestellten Ansätzen sind Evolutionsstrategien und Genetische Algorithmen sehr gut einsetzbar, die Co-Evolution ist für den Praxis-Einsatz jedoch nicht zu verwenden. Die Hauptproblematik lag im Finden einer geeigneten Fitness-Funktion und der richtigen Einstellung ihrer Parameter. Insgesamt ist die Verwendung von Heuristiken der Computational Intelligence jedoch als positiv zu bewerten, falls keine anderen effizienten Algorithmen vorhanden sind.

Für den anstehenden Praxiseinsatz ist es sehr wichtig, gültige Lösungen zu produzieren. Dieses wird durch die beiden erfolgreichen Ansätze auf jeden Fall gewährleistet. Die fehlenden 5% stellen fast ausschließlich Konstellationen dar, bei denen überhaupt keine Spannmittel plaziert werden konnten.

Die hier vorgestellten Ansätze mit der Einbeziehung von Geometrie-Algorithmen können auch in andere Problembereiche wie die Verschnittoptimierung übernommen werden. Es erscheint durchaus möglich, mit diesen Verfahren in Zukunft auch komplexere Aufgabenstellungen wie sie sich z.B. beim Einsatz von anderen Maschinentischen (Konsolen) oder von dreidimensionalen Spannmitteln mit versetzter Ober- und Unterseite ergeben, zu lösen.

Danksagung

Wir danken der HOMAG Maschinenbau AG, Schopfloch für die Förderung.

Literatur

[Bäck96] Bäck, Th.: Evolutionary Algorithms in Theory and Practice. Oxford University Press, New York, 1996.

[BauZell96] Baumann, R.; Zell, A.: Distributed Genetic Algorithms on the Intel Paragon, a large MIMD Computer. In Proceedings of the 5th Annual Conference on Evolutionary Programming, San Diego (CA), 1996.

[Foley95] Foley, James D. et al.: Computer Graphics Principles and Practice, Second edition in C, Addison-Wesley, Reading (MA), 1995.

[Hafner98] Hafner, S. (Hrsg.): Industrielle Anwendung Evolutionärer Algorithmen. München: Oldenbourg Verlag, 1998.

[Hollan93] Holland, J.H.: Adaption in natural and artificial systems; an introduction analysis with applications to biology, control and artificial intelligence. MIT press, 1993.

[KurSch97] Kursave, F.;Schwefel, H.-P.: Optimierung mit Evolutionären Algorithmen. Automatisierungstechnische Praxis 39 (1997) 9, S. 10-17.

[Michal96] Michalewicz, Z.: Applications of evolutionary algorithms for constrained problems. In Parallel Problem Solving from Nature - PPSN IV, Proceedings, pp. 245-254. Berlin: Springer, 1996.

[Nissen94] Nissen, V.: Einführung in Evolutionäre Algorithmen. Vieweg-Verlag, 1997.

[Paredi94] Paredis, J.: Co-Evolutionary constraint staisfaction. In Parallel Problem Solving from Nature - PPSN III, Proceedings, pp. 46-55. Berlin: Springer, 1994.

[Rechen94] Rechenberg, I.: Evolutionsstrategie 94, Volume 1 von Werkstatt Bionik und Evolutionstechnik. Frommann-Holzboog, Stuttgart, 1994.

[Schöne94] Schöneburg, E.: Genetische Algorithmen und Evolutionsstrategien. Addison-Wesley, 1994.

[Wakund95] Wakunda, J.: Verteilte Evolutionsstrategien auf den Supercomputern Intel Paragon und Cray T3D. Diplomarbeit, Universität Stuttgart, 1995.

[WakZell97] Wakunda, J.; Zell, A.: EvA, A Tool for Optimization with Evolutionary Algorithms. Proceedings of the 23rd EUROMICRO Conference; Budapest, Hungary, 1.-4. September 1997, pp. 644-651.

Manuskripteingang: 14. Dezember 1999