

Implementierung eines genetischen Algorithmus zur automatisierten grafischen Darstellung von graphstrukturierten Daten

Heiko Gruner, Peter Lüders, Rolf Ernst

A design-tool for digital circuits by mixed specification (textual-graphical) was developed at the "Institut für Datenverarbeitungsanlagen". The textual input by the user is automatically transferred into a cyclic, hierarchical graph and displayed on the screen. This interactive system reacts on changes within the text and displays the changed graph.

A genetic algorithm was implemented as one part within the placing-routine. Its purpose is to place same sized elements in matrices in different hierarchies.

Genetic algorithms are stochastic optimisation algorithms that try to find solutions for an optimisation problem by simulating nature's evolution with selection and transmission.. Further development in the evaluated generations of individuals is influenced by cost functions that are inclined by the optimisation problem.

Two distinct methods of genetic algorithms are discussed in literature. The classical Method, first developed for function optimisation , has little knowledge about the optimisation problem. Directed evolution is the second method that works more methodical with more knowledge.

A implementation if both methods is presented. Results demonstrate that the directed evolution produces better layouts in acceptable computation time. So the implementation of a genetic algorithm within the interactive design tool 'circuit editor' seems to be useful.

Am Institut für Datenverarbeitungsanlagen wird ein Editor zur gemischt grafisch-textuellen Spezifikation von Digitalschaltungssystemen entwickelt. Hierbei wird die textuelle Eingabe automatisch in einen gerichteten, zyklischen, hierarchischen Graphen umgesetzt und auf dem Bildschirm grafisch dargestellt.

Das System soll interaktiv arbeiten und schnell auf Änderungen im Text reagieren. Im Rahmen einer Untersuchung von Layoutalgorithmen wurde ein genetischer Algorithmus für

ein Teilproblem des Layouts mit standardisierten Knotengrößen und mehreren Hierarchieebenen implementiert.

Genetische Algorithmen ahmen bei der Suche nach Lösungen eines Optimierungsproblems die Evolution in der Natur nach. Durch Selektion und Vererbung sollen sich über mehrere Generationen hinweg Individuen, die Lösungen des Optimierungsproblems darstellen, weiterentwickeln. Diese Weiterentwicklung erfolgt unter dem Einfluß einer Bewertung jedes Individuums durch eine Kostenfunktion.

Aus der Literatur sind zwei grundsätzliche Verfahren von genetischen Algorithmen für Plazierungsprobleme bekannt. Das klassische, ursprünglich zur Funktionsoptimierung entwickelte Verfahren ist weitgehend problemunabhängig. Das zweite, als gerichtete Evolution bekannte Verfahren geht mit mehr Wissen vom Optimierungsproblem zielgerichteter vor.

Es wird eine Umgebung vorgestellt, mit der beide Methoden untersucht und verglichen wurden. Die Ergebnisse zeigen, daß die zielgerichtete Methode bessere Ergebnisse in kürzerer Rechenzeit generiert und hiermit erst den Einsatz von genetischen Algorithmen in interaktiven Softwaresystemen ermöglicht.

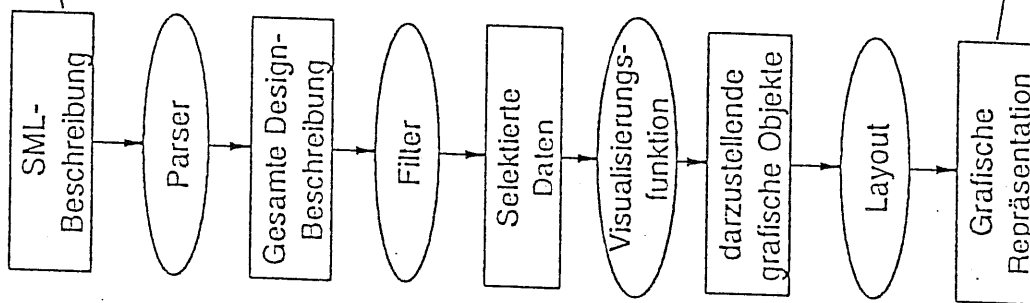
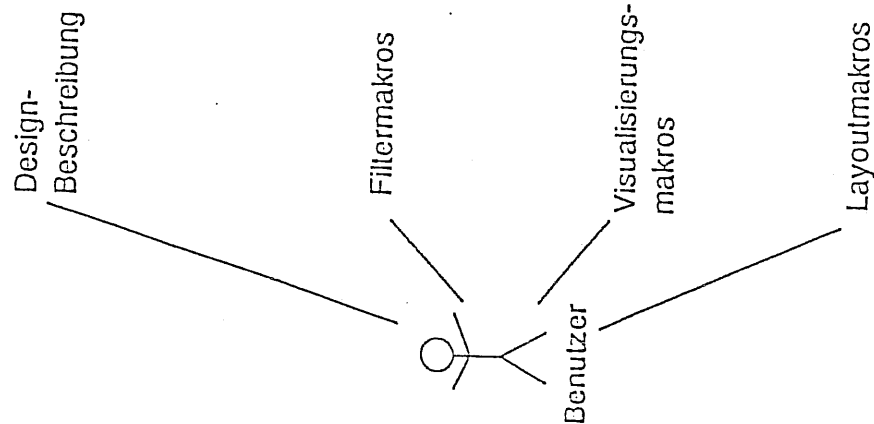
Implementierung eines genetischen Algorithmus zur automatisierten grafischen Darstellung von graphstrukturierten Daten

H. Gruner
P. Lüders
R. Ernst

Institut für Datenverarbeitungsanlagen
Technische Universität Braunschweig

- Aufgabenstellung des Algorithmus im Projekt
- Einführung in genetische Algorithmen
- charakteristische Punkte des Algorithmus
- Qualitäts-/ Laufzeitergebnisse
- Zusammenfassung / Ausblick



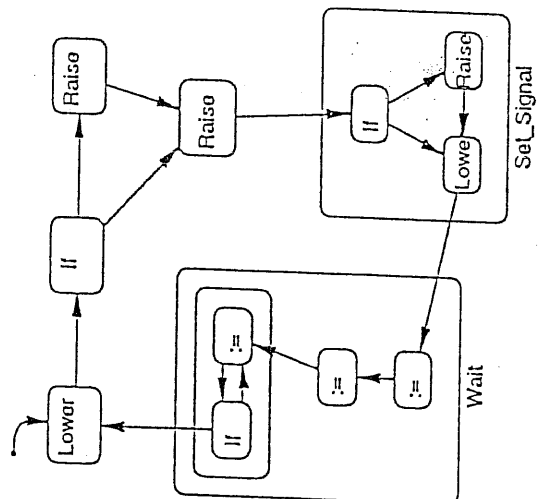


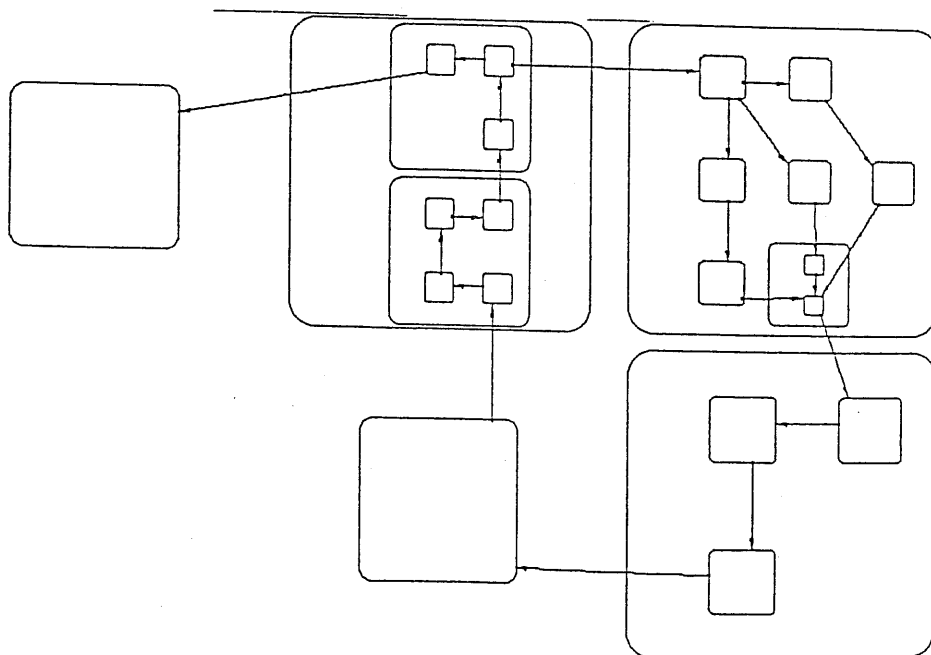
```

PROCEDURE WAIT
  SUM = 0
  MAX = 100
  B: IF (SUM>MAX) THEN SUM = SUM+1
    ELSE GOTO A: ENDIF
  A: ENDPROC

PROCEDURE SET_SIGNAL
  IF (SIG_A = '1') THEN RAISE (SIG_B) ENDIF
  LOWER (SIG_C)
ENDPROC

/* Hauptschleife */
LOOP
  LOWER (SIG_B)
  IF (TASTE = '1') THEN RAISE (ERR) ENDIF
  RAISE (SIG_A)
  SET_SIGNAL
  WAIT
ENDLOOP
  
```





Benutzeranforderungen:

schnelle Ergebnisse

übersichtliche Platzierung

Schwierigkeiten:

Hierarchieebenen

Lösungsansatz:

phasenweise Platzierung

hier:

globale Phase der Platzierung

Matrizenplatzierung innerhalb

sämtlicher Hierarchien

genetische Algorithmen:

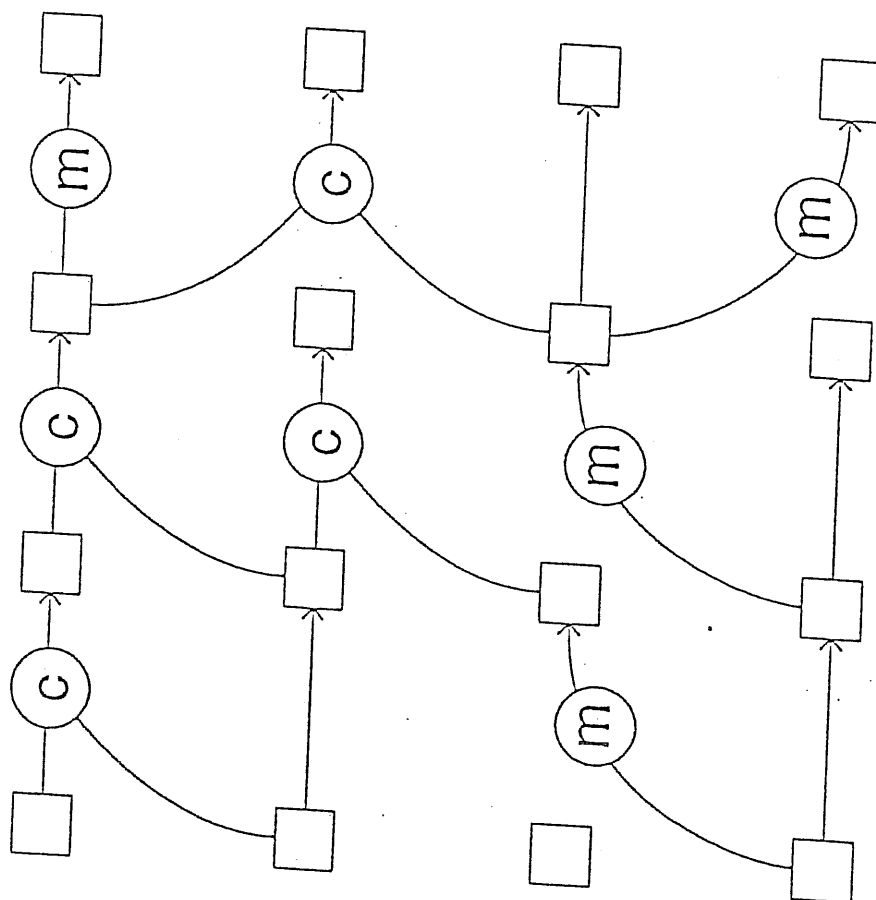
- stochastische Optimierungsverfahren
- ahmen die Evolution der Natur nach

Methoden:

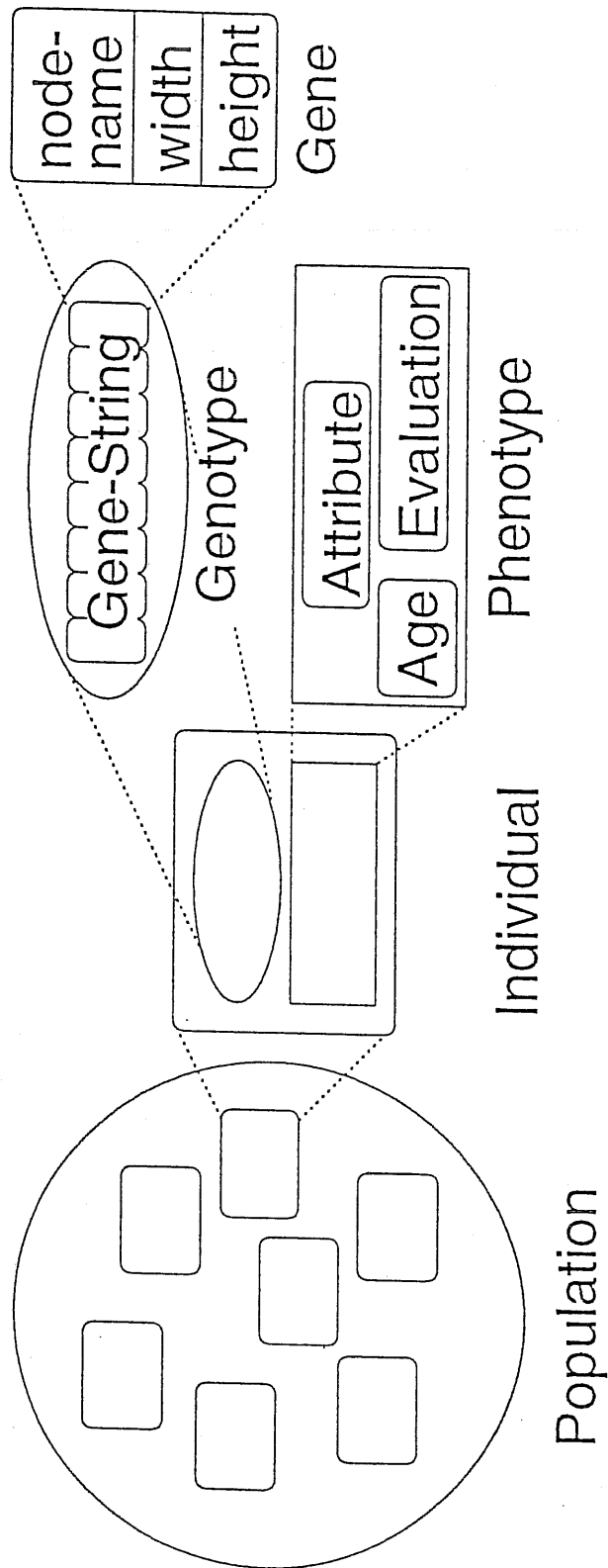
- Selektion
- Vererbung







Beispielablauf eines genetischen Algorithmus



Kostenfunktionen :

- Kantenlängen
- Kantenkreuzungen
- Überdeckungen
- Ähnlichkeit zwischen Individuen der Population
- Ähnlichkeit zur Vorplazierung
- Alter

Normierung :

- Verbindung mehrerer Kostenfunktionen
- zeitlich unterschiedliche Wichtungen möglich

Initialisierung:

gute Ausgangspunkte $\leftrightarrow$ hohe Varianz

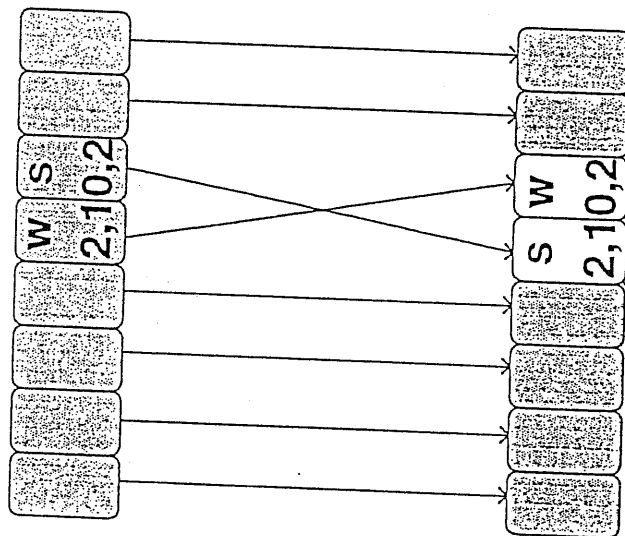
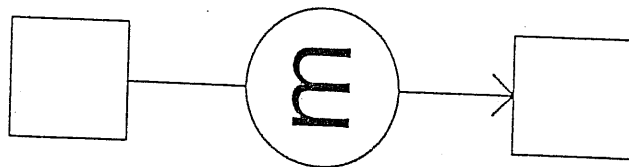
Auswahl der Überlebenden / Sterbenden:

schnelle Konvergenz $\leftrightarrow$ hohe Varianz

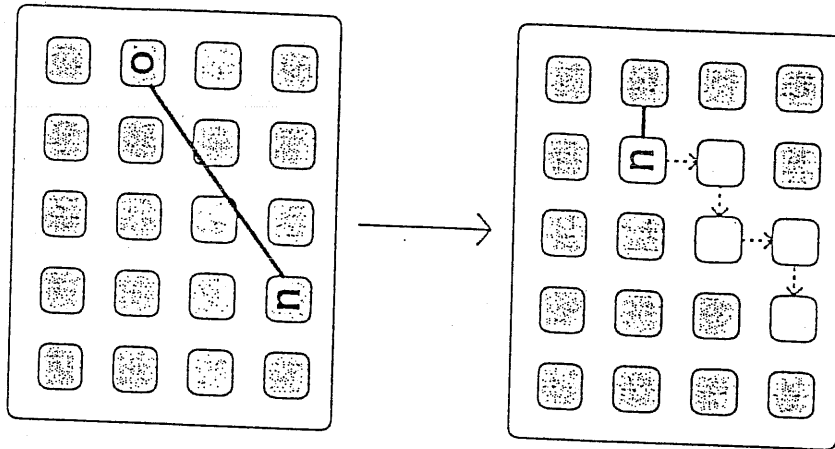
Auswahl der Elternteile und Operatoren:

schnelle Konvergenz $\leftrightarrow$ hohe Varianz





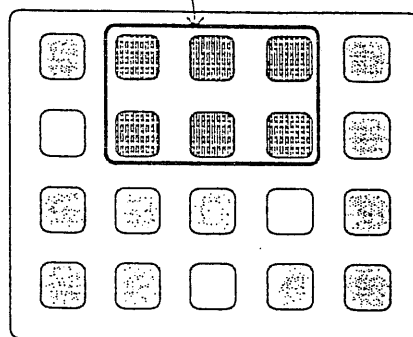
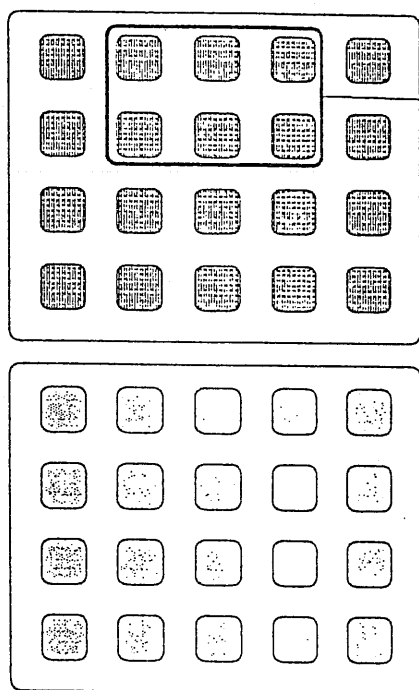
klassisch



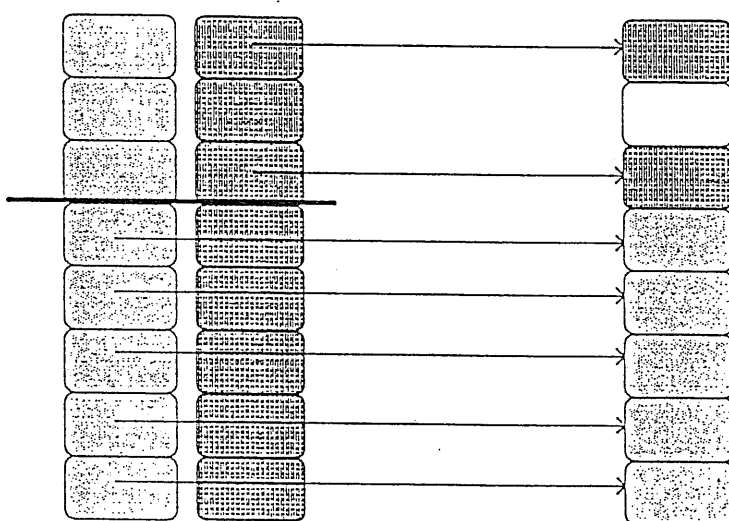
gerichtet

mutation-Operator

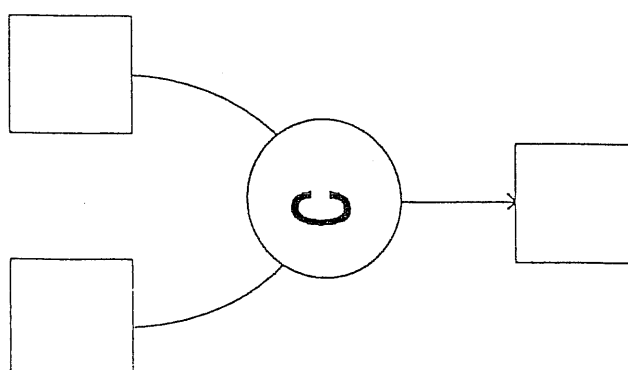




gerichtet

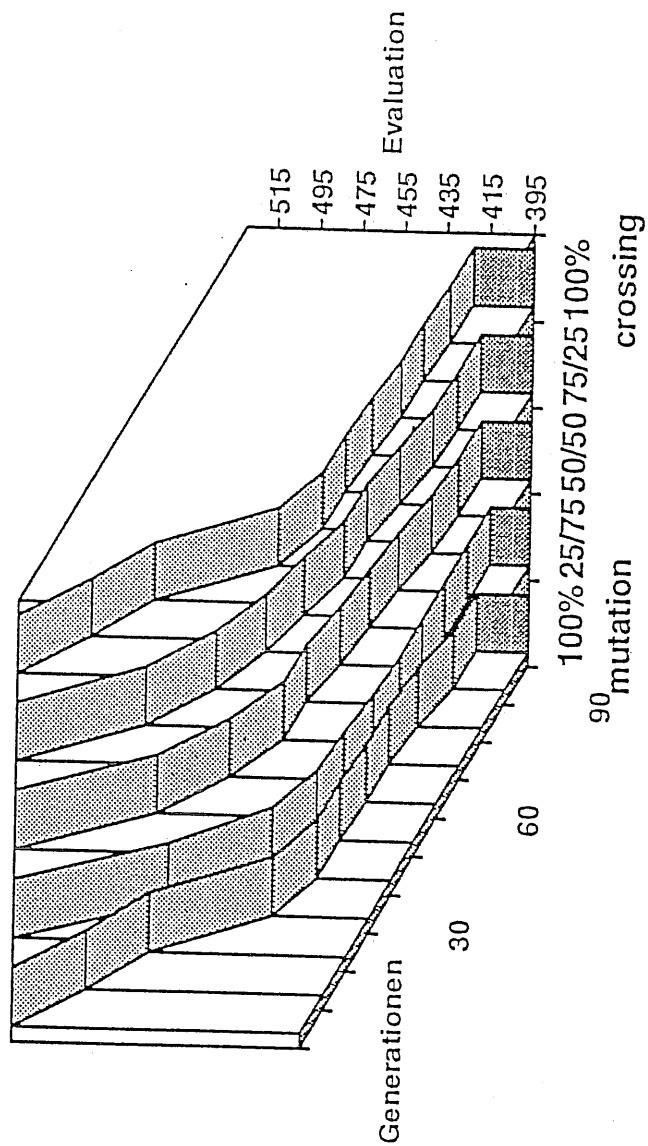


klassisch



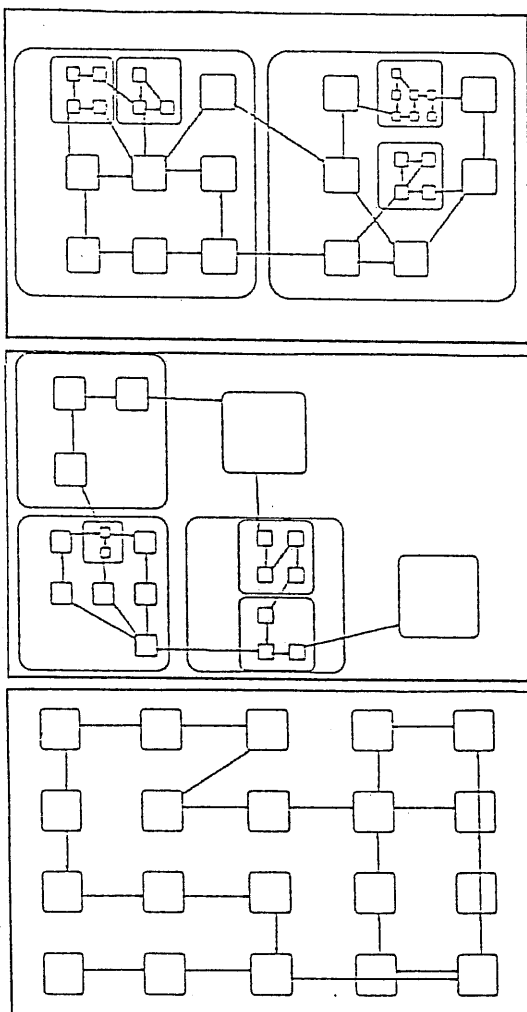
crossing-Operator





Qualitätsverbesserungen im Verlauf mehrerer Generationen





genetic	4.8	2.4	3.8	sec
	6000	3000	6000	Anzahl der Bewertungen
simulated annealing	3.6	2.0	3.4	sec
	6800	13100	20200	Anzahl der Bewertungen



Aufgabenstellung:

übersichtliche Platzierung hierarchisch strukturierter Graphen
in einem interaktiven Softwaresystem

genetischer Algorithmus:

stochastisches Optimierungsverfahren
Nachahmung der Evolution durch Selektion und Vererbung
klassische <-> gerichtete Methoden

Ergebnis:

Die Anwendung des genetischen Algorithmus mit gerichteten Methoden
liefert gute Platzierungen in einer akzeptablen Zeit.

Implementierung neuer gerichteter Operatoren

Kostenfunktionen nicht vollständig neu berechnen

Parallelisierbarkeit ausnutzen

Bildung von Bevölkerungsgruppen



Ausblick