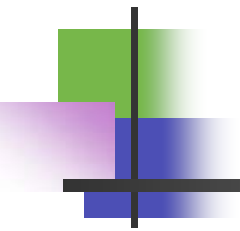


Using Genetic Algorithms to Explore Pattern Recognition in the Immune System

von

Burkhard Menges

Inhalt

- Einführung in genetische Algorithmen
- Allgemeine Problemstellung / Modellbeschreibung
- Gemeinsame Schemata erkennen
- Aufrechterhaltung der Unterschiedlichkeit
- Zusammenfassung
- Fazit

Grundlagen genetischer Algorithmen

- Gehören zu den evolutionären Algorithmen
- EA arbeitet auf einer Population von Individuen von denen jedes einen Punkt im Suchraum der möglichen Lösungen für ein gegebenes Problem darstellt
- GA arbeitet auf Gen-Population
- Population entwickelt sich durch Reproduktion, genetische Variation und Selektion
- Individuen der Population entwickeln sich in vordefinierter Umgebung

Ein Standard-evolutionärer Algorithmus

(1) *Initialisierung:* Erzeuge eine initiale Population von Individuen (Eltern)

(2) *Evaluation:* Evaluiere die Qualität des Verhaltens des Systems für alle Individuen der Population (Fitness-Test)

(3) *Selektion:* Bestimme mittels Auswahloperatoren, welche Individuen mit welcher Häufigkeit ausgewählt werden

(4) *Reproduktion und genetische Variation:* Erzeuge neue Population durch Reproduzieren der Ausgewählten und Variieren ihrer Nachkommen (Crossover, Mutation)

(5) *Schleife:* Wiederhole Schritt (2) und (4) bis eine vordefinierte Lösung erreicht oder eine bestimmte Anzahl von Generationen durchlaufen ist

Ein Standard-evolutionärer Algorithmus (2)

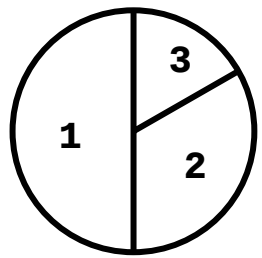
- Jedes Mal wenn Schritte (2) bis (4) durchlaufen sind, ist eine neue Generation entstanden
- Dieser Prozess der simulierten Evolution entsteht durch sukzessive Anwendung eines natürlichen Auswahlmechanismus
- Überlebenswahrscheinlichkeit für besonders fitte Individuen wird über Generationen hinweg erhöht

Genetische Algorithmen konkret

- Stochastischer Algorithmus dessen Suchmethode genetische Vererbung und natürliche Auslese zu modellieren versucht
- Individuen werden durch „genetische Strings“ fester Länge repräsentiert
- Benutzt eine Kostenfunktion (Fitness, Ziel, Anpassungsfähigkeit)
- Anwendung probabilistischer Übergangsregeln anstelle deterministischer

Repräsentation und Reproduktion

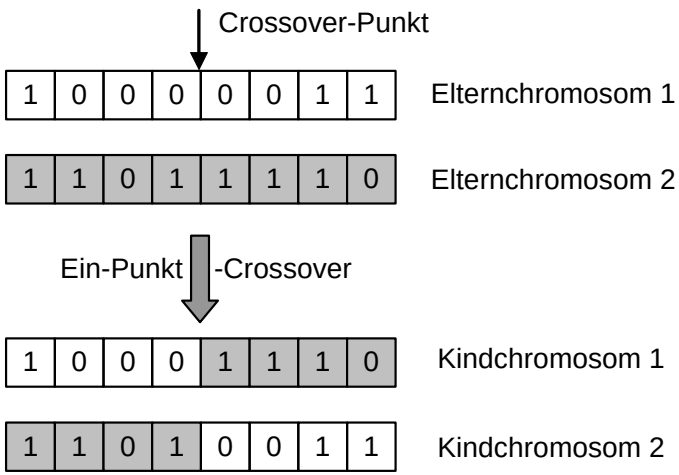
- Darstellung als Vektor im n-dimensionalen Suchraum
- Auswahl der Eltern z.B. durch Rouletterad Methode



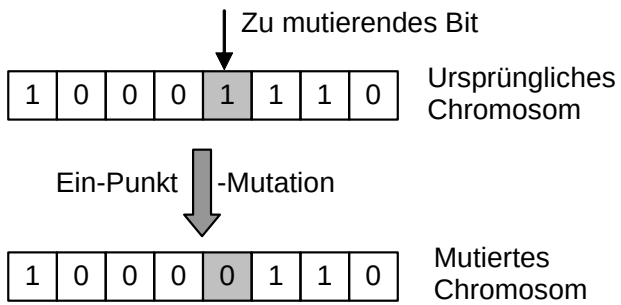
- Reproduktion mittels Crossover oder Mutation
- Beim Crossover werden die Eltern-Vektoren (Strings) an bestimmten Stellen aufgetrennt und überkreuzt wieder zusammengesetzt
- Bei Mutation werden ausgewählte Positionen im Vektor zufällig verändert

Crossover und Mutation

Crossover



Mutation



Kurze Erinnerung

- Immunsystem kann etwa 10^{16} verschiedene Fremdkörper erkennen
- Selbst im Labor gezüchtete Körperfremde Moleküle werden erkannt
- Erkennung erfolgt durch B- bzw. T Lymphozyten
  - Rezeptoren haben einzigartige Oberflächenbeschaffenheit
- (Prinzip der klonalen Selektion)

Allgemeine Problemstellung

- Immunsystem ist in der Lage verschiede Arten von Mustererkennungsproblemen zu lösen
- Frage: Sind diese Probleme auch mit genetischen Algorithmen zu lösen
- Ziel der Recherche:
  - Immunsystem vom Informationsverarbeitenden Standpunkt aus verstehen
  - Ideen nutzen um neue effiziente parallele Algorithmen zu konstruieren

Binäres Immunsystem

- Antikörper (Rezeptoren)und Antigene werden durch binäre Strings repräsentiert (Hamming Space)
- Modell benutzt Strings der Länge 64 Bit
  - $\Rightarrow 2^{64} = 10^{19} \, Möglichkeiten$
- Modellierung ist extreme Vereinfachung
  - Direkte Mutation der Bitstrings
  - Zufällige Selektion und Mutation

Binäres Immunsystem (2)

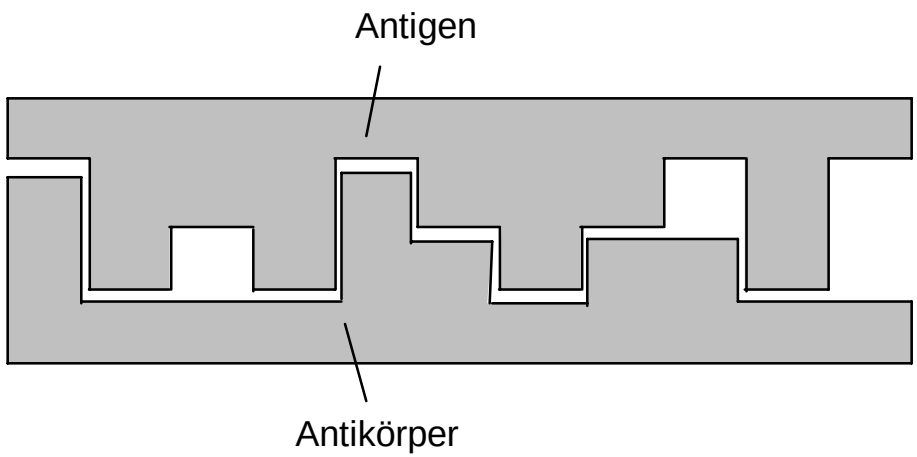
- Dennoch angemessen
  - Vielfalt mind. genauso groß wie in der Biologie
  - Immunsystem hat einige Mechanismen zur Randomisierung der Antikörper (wenn auch komplexer)
- Modell behandelt nur die Erkennung der Antigene durch Antikörper nicht deren Vernichtung
- Matching zwischen Rezeptor und Antigen wenn ihre Bitstrings komplementär sind

Matching

Binäres Modell:

Antibody:	11001001000100100001001010101010
Antigen:	01111100111001011110110101110100
Complement:	10110101111101111111111111011110

Allgemeines Modell:



Matching (2)

- Perfektes bitweises Matching nicht nötig, da jeder Antikörper zu mehreren Antigenen passen muss
- Grad des Matchings wird durch Matching-Funktion bestimmt
  - $M : Antigen \times Antibody \rightarrow \Re$
- Verschiedene Arten von Matching Regeln denkbar
  - Aufsummieren der komplementären Bits
  - Länge von komplementären Regionen bestimmen und lange Regionen mehr belohnen als kurze

Testumgebung

- Benutzte Match Funktion: $M = \sum_i l_i$
- Ziel: Überdeckung mit Fremdkörperpopulation
  - Kann trivial sein wenn Match Bedingung zu schwach
- Forderung:
  - Jedes Antigen wird von mindestens einem Antikörper erkannt
  - Kein Körpereigenes Molekül wird fälschlicherweise als Körperfremd erkannt
- Um das zu erreichen fordere höhere Fitness wenn viele Matchings auftreten

Gemeinsame Schemata entdecken

- Motivation: Immunsystem muss Bakterien und andere Pathogene erkennen
  - Viele Bakterien haben Eigenschaften die im Körper so nicht auftauchen
  - z.B. Zellwände aus Polymeren die bei menschlichen Zellen nicht vorkommen
- Testen der Fähigkeit von GA's gemeinsame Muster (Schemata) zu erkennen
- Idee: Gemeinsame Bitmuster repräsentieren Eigenschaften wie oben beschrieben

Problemstellung

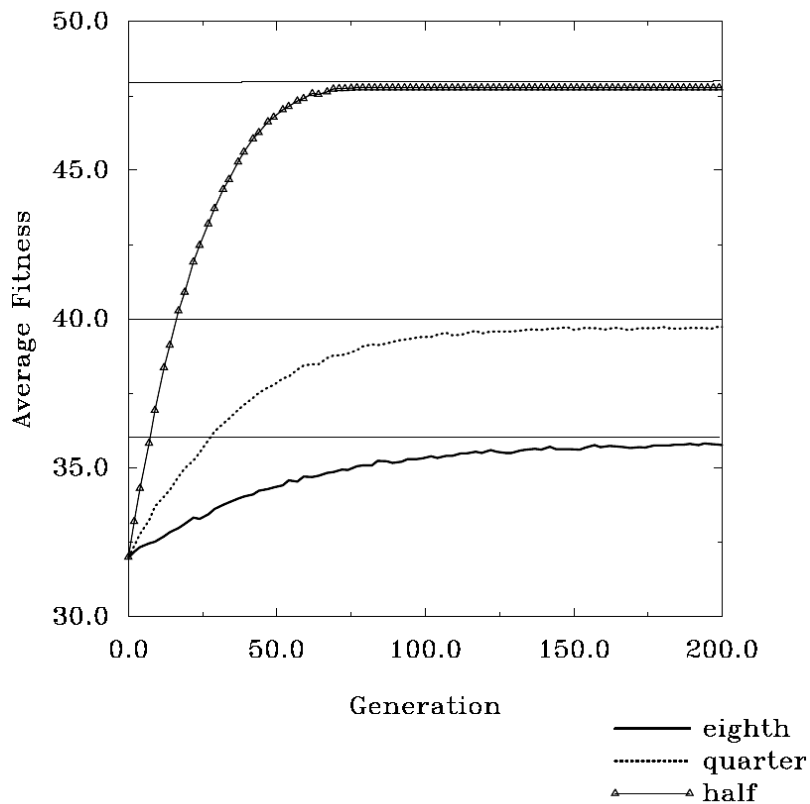
- Half-length schema:
  - 50% 11 … 11** … **
  - 50% ** … **11 … 11
- Quarter-length schema:
  - 25% 11****** 25% ****11**
  - 25% **11**** 25% ******11
- Fitnessberechnung für Problemstellung:
  - Zufällige Auswahl von p Antigenen aus fester Population
  - Für jedes Antigen k und jeden Antikörper j berechne M(k,j)
  - Fitness eines Antikörpers j ist durchschnittliche Matching Punktzahl über den p Antigenen

Problemstellung (2)

- Optimaler Antikörper für beide Muster: 000 … 000
- Durchschnittliche Fitness des Antikörpers sinkt von half- zu quarter-length (weniger sichere Matchings)
- Durchschnittliche Fitness errechnet sich zu:
  - $[(s/l)(1)+(1-s/l)(1/2)]l = \frac{l+s}{2}$ s=Schemalänge, l=Stringlänge
- Lösungen können Generalisten oder Spezialisten sein
- Spezialisten für gegebene Problemstellung ungeeignet

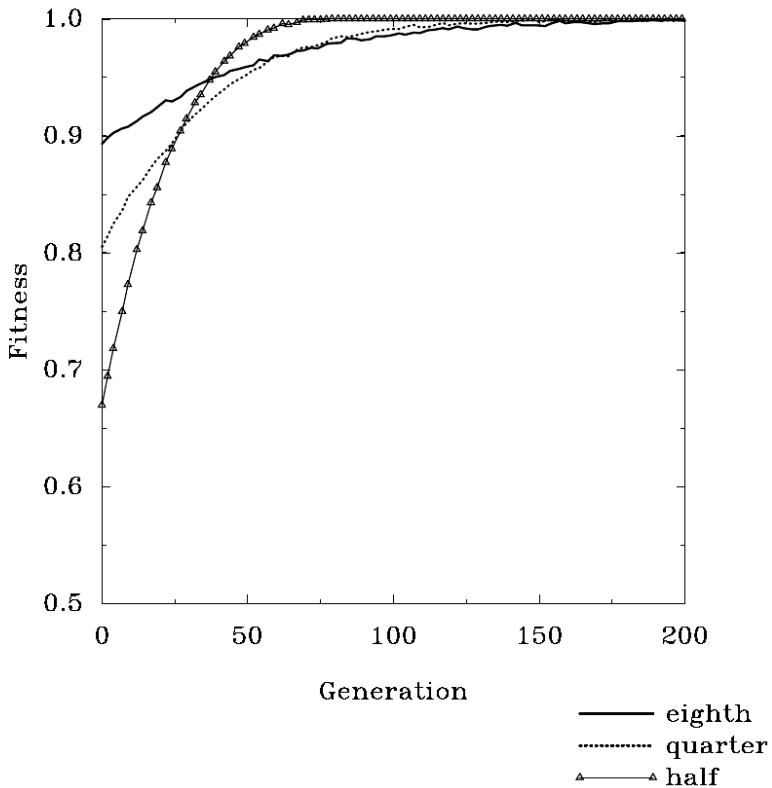
Wie gut erkennt ein GA gemeinsame Schemata?

- Versuch mit 200 zufällig ausgewählten Antikörpern (Länge 64 Bit), p=5
- Durchschnittliche Fitness steigt monoton bis Maximum
- Fitnessasymptote wie errechnet



Wie gut erkennt ein GA gemeinsame Schemata?(2)

- Bei längerem Schema wird Maximum schneller erreicht (weniger Rauschen)
- Vermutlich können nur begrenzt kleine Schemata erkannt werden



Aufrechterhalten der Unterschiedlichkeit

- Betrachte Antigen Populationen die nicht von einem Generalisten erkannt werden können
- 50% 000 … 000
- 50% 111 … 111
- Zur Lösung dieses Problems müsste GA zwei verschiedene Lösungen gleichzeitig aufrechterhalten (Multiple Peaks Problem)
- Schwierig für GA, typischerweise konvergieren Algorithmen wahllos gegen einen der Peaks

Aufrechterhalten der Unterschiedlichkeit (2)

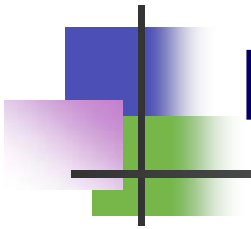
- Crossover eher ungeeignet
  - Somatische Mutation nutzt kein Crossover
- Zum Testen der Aufrechterhaltung von Unterschiedlichkeit kann Lösung vorgegeben werden
- Für dieses Problem muss Fitnessberechnung angepasst werden

Fitnessberechnung

- Eigenschaften des Immunsystems bei Fitnessberechnung:
  - Antigene treten sequentiell auf
  - Immunsystem reagiert nur mit einer Teilmenge seiner Lymphozyten
  - Es gibt einen „Wettkampf“ um die Antigene. Antikörper mit größerer Affinität vermehren sich schneller
  - Antikörper entwickeln sich durch Punkt Mutation (Somatische Mutation) => GA muss mit vermehrt mit Mutation anstelle von Crossover arbeiten

Fitnessberechnung (2)

- Neues Fitness Schema:
  - Wähle ein Antigen zufällig aus
  - Eine Menge der Größe σ von Antikörpern wird zufällig gewählt
  - Jeder dieser Antikörper wird mit dem Antigen gematcht
  - Antikörper mit größter Match Punktzahl bekommt diese auf seine aktuelle Fitness addiert
  - Wiederhole Vorgang für weitere Antigene (typischerweise 3*σ)
- Schema entspricht einer „best-match" Strategie von Klassifizierungssystemen

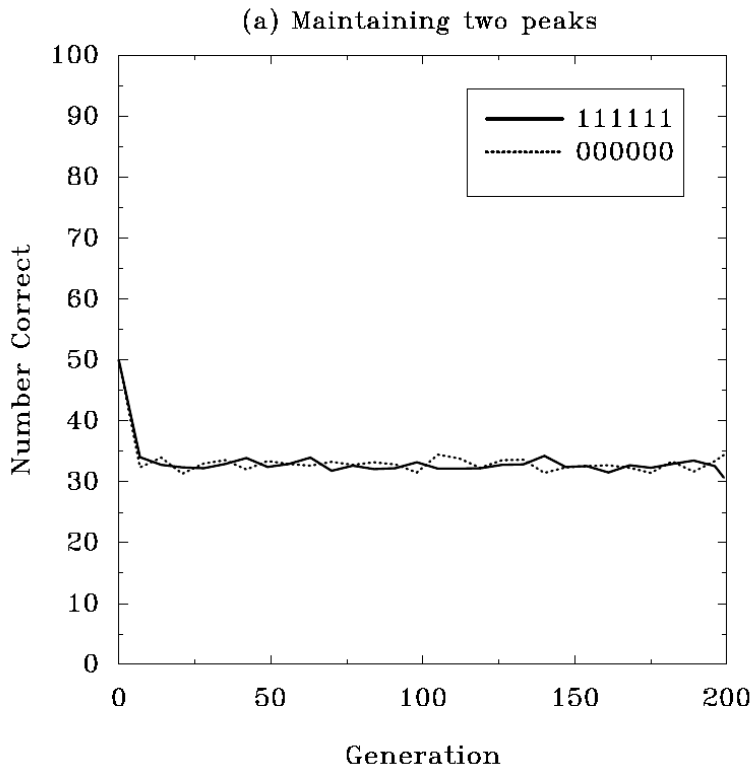


Fragen

- Kann ein konventioneller GA die verschiedenen Antikörper aufrechterhalten oder konvergiert er zu einer Lösung?
- Wie viele verschiedene Peaks kann der GA erhalten?
- Findet der GA die Peaks auch ohne vorgegebene Lösung?
- Spielt die Entfernung der Peaks voneinander eine Rolle (Hamming Distanz)
- Entwickeln sich Spezialisten oder Generalisten und unter welchen Umständen?

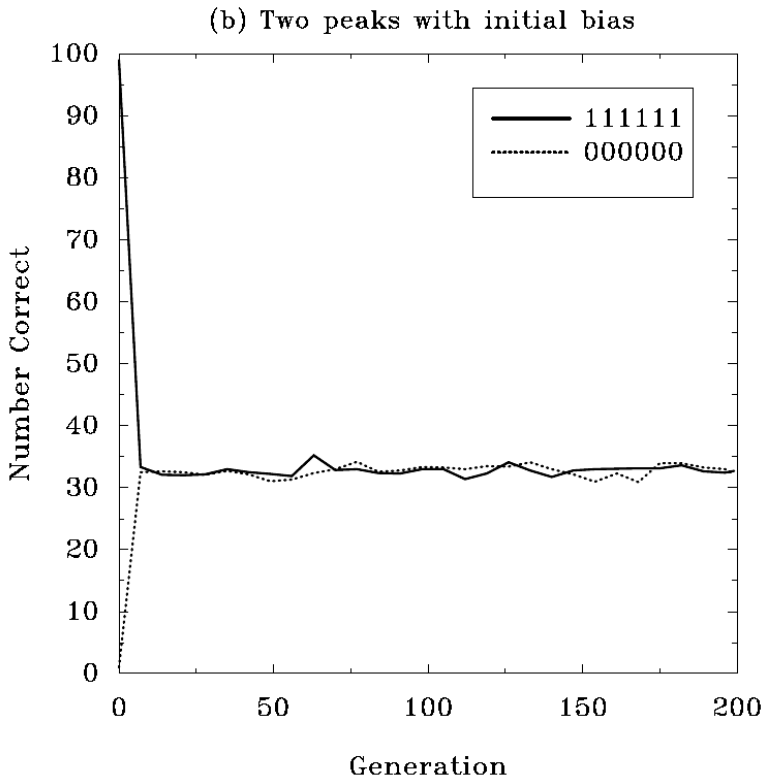
Kann Verschiedenheit erhalten werden?

- Teste GA mit 2 Antigenklassen der Länge 64 Bit und σ=30
  - 50% 111 … 111
  - 50% 000 … 000
- Wenn GA mit korrekten Lösungen initialisiert wird, kann er sie aufrechterhalten
- Durch das Crossover gehen einige korrekte Antikörper verloren aber 50-50 Verteilung bleibt erhalten



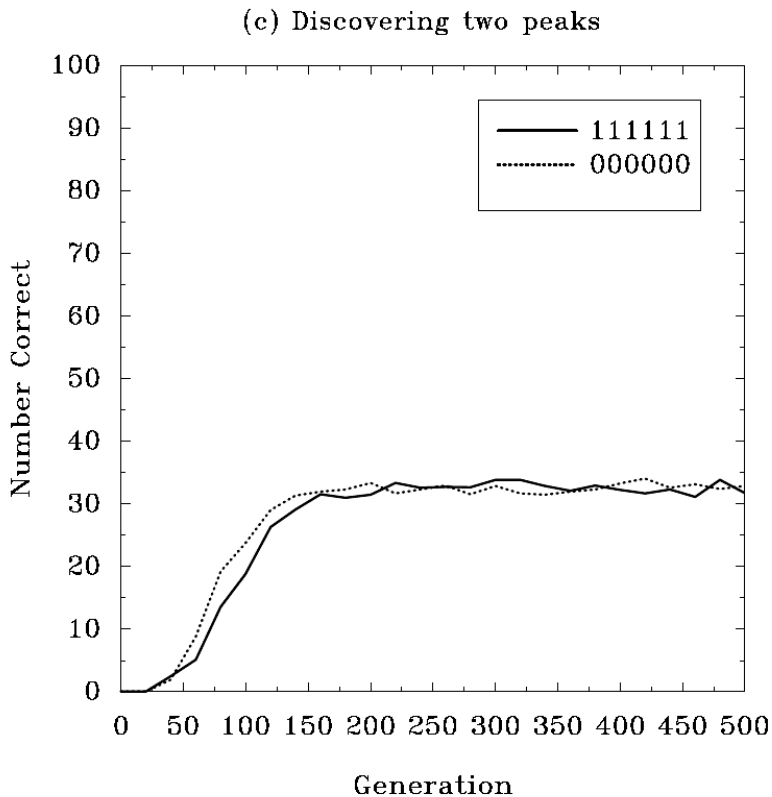
Kann Verschiedenheit erhalten werden? (2)

- Was passiert bei einer Initialisierung der AB‘s mit einer anderen Mengenverteilung als die der AG‘s
- 50-50 Verhältnis wird in weniger als 10 Generationen wieder hergestellt
- Starke Tendenz zur Selbst – Regulierung => sollte stabil gegen Störungen (Rauschen etc.) sein



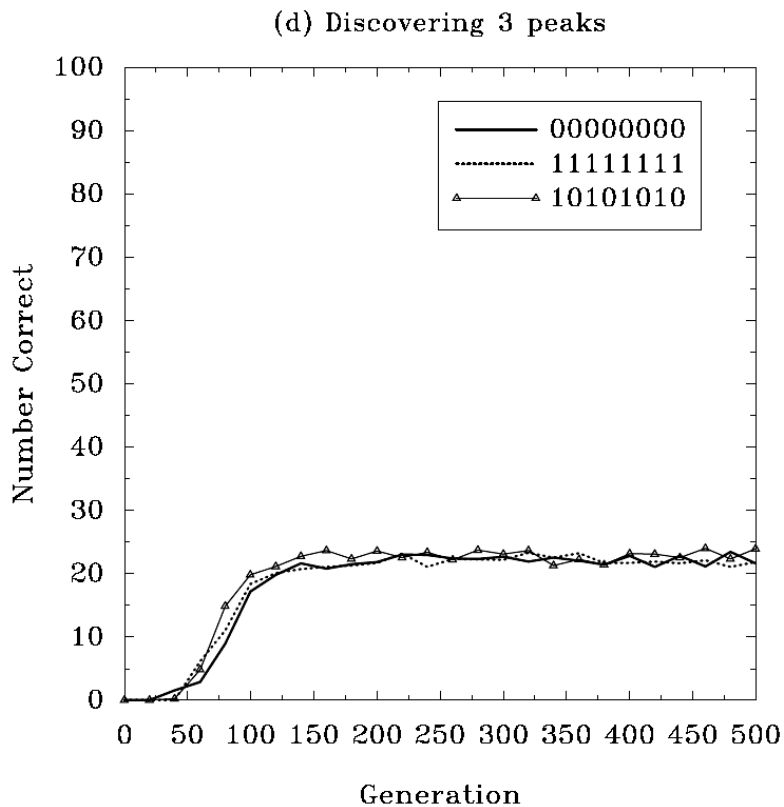
Werden verschiedene Peaks selbstständig gefunden?

- Gleiche Vorraussetzungen wie zuvor
- Antikörper werden zufällig aus Population entnommen
- Auch hier werden Peaks erkannt und aufrechterhalten



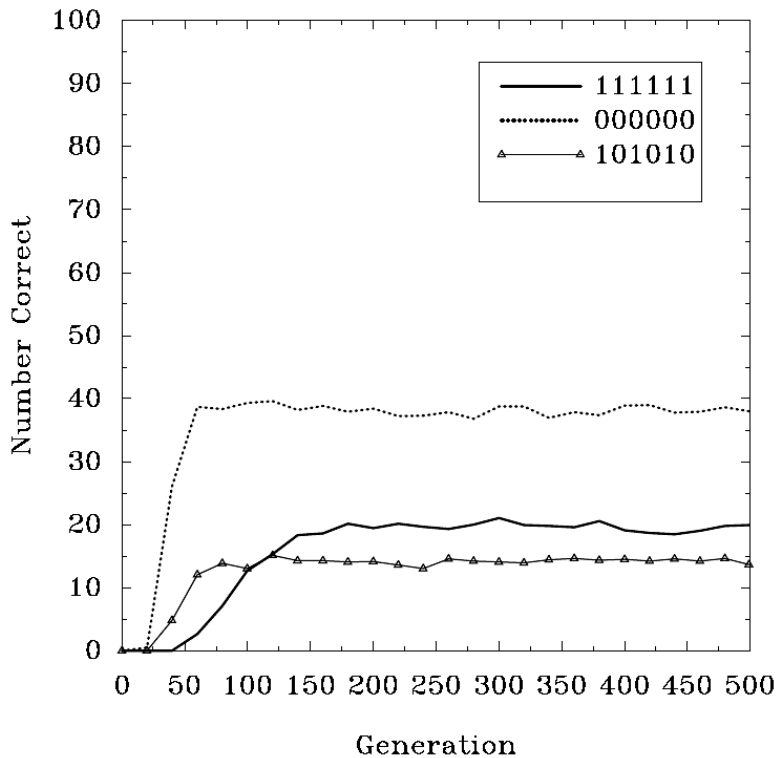
Werden verschiedene Peaks selbstständig gefunden? (2)

- Auch 3 gleichverteilte Peaks können erkannt und aufrechterhalten werden
- σ ist wichtiger Kontrollparameter
- Bei σ=5 wird nur ein Peak gefunden
- Also existiert für jede Population ein Maximum von auffindbaren Peaks



Unterschiedliche Größen der Peaks

- Antigen Population:
  - 50% 000 … 000
  - 30% 111 … 111
  - 20% 101 … 010
- Restliche Vorraussetzungen wie zuvor
- Peaks werden gefunden
- Anzahl der Antikörper korrespondiert mit der Zahl der zugehörigen Antigene



Wie wirkt sich σ aus

- Wie viele Antikörper werden benötigt um ein Antigen zu erkennen?
- Auch Biologisch interessant => Erkennungs Kapazität des Immunsystems
- Etwa 15 Antikörper pro Antigen werden benötigt
- 10-15% der Antikörper matchen mit keinem Peak

Antigen Types	Population Size		
	50	100	200
2	21.6 (3.2)	46.6 (2.6)	93.0 (4.6)
3	14.1 (7.2)	30.5 (2.3)	60.7 (5.3)
4	0.0	23.0 (0.9)	45.4 (4.6)
5		18.0 (0.6)	36.1 (3.3)
6		0.0	30.0 (3.0)
7		0.0	25.3 (3.3)
8		0.0	22.6 (0.4)
9		0.0	20.0 (0.4)
10		0.0	17.6 (0.7)
11		0.0	15.4 (0.7)
12		0.0	0.0

σ=15% Population Size

σ im realen Immunsystem

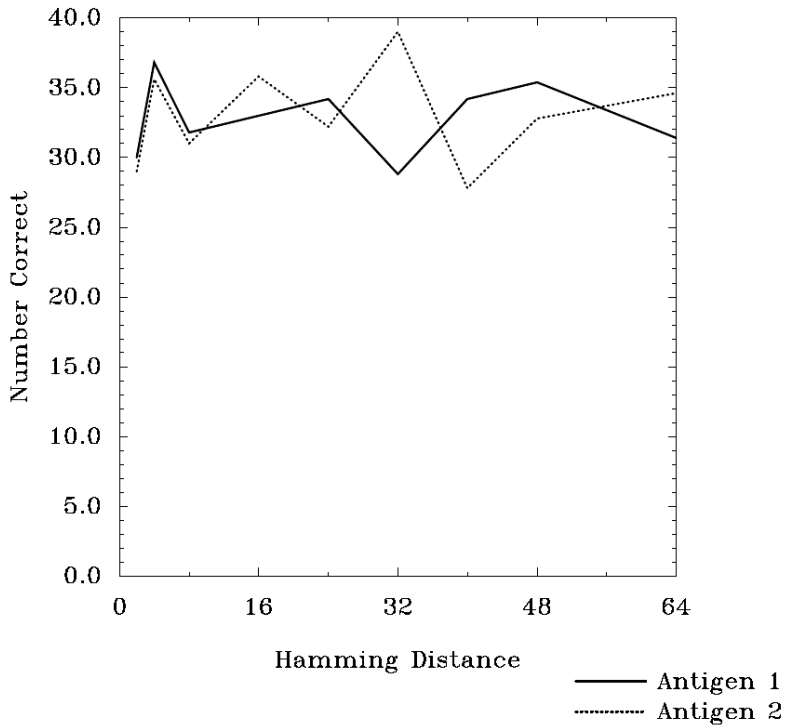
- Bei einem Tier reicht ein Antikörper (eigentlich B-Zelle) um ein Antigen zu erkennen
- => kleine Mutationen im Antikörper oder Antigen könnten die Erkennung verhindern
- IS nutzt verschiede Antikörper um Antigene auf unterschiedliche Art zu erkennen
  - Zahl bewegt sich im 2-3-stelligen Bereich
  - Durchschnittlich etwa 100

Ähnlichkeit von Peaks

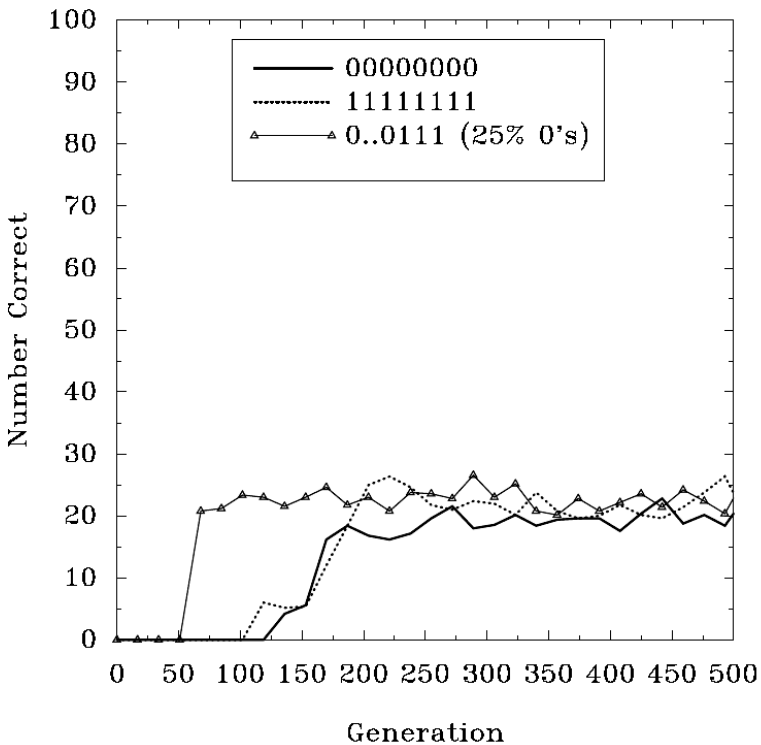
- Antigen-Peaks gleicher Größe mit variierender Hamming Distanz von 64 bis 1 werden untersucht
- Antikörper werden zufällig ausgewählt
- $\sigma=15$
- Peaks müssen über 500 Generationen aufrecht erhalten werden

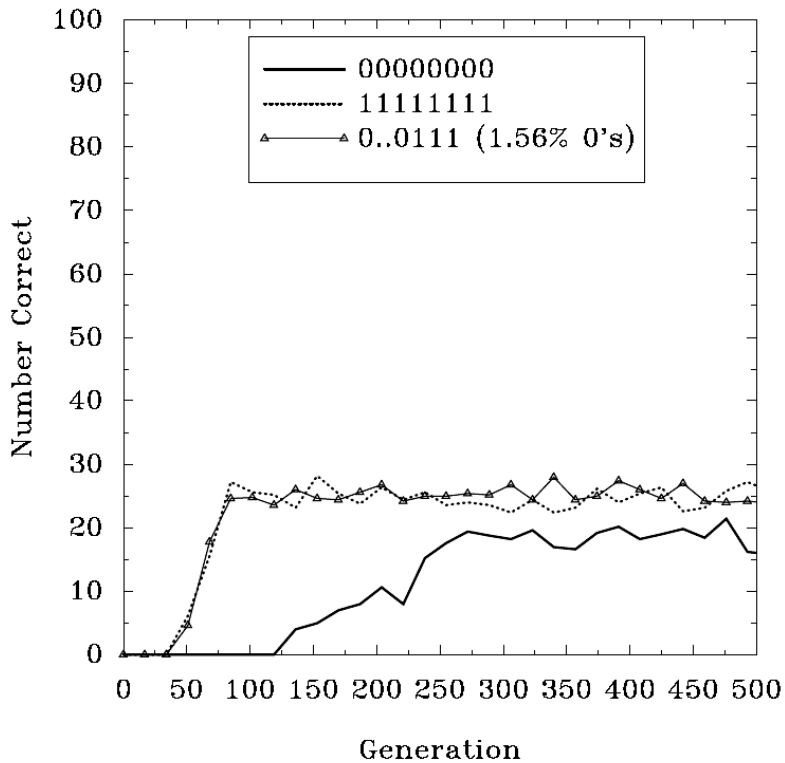
Ähnlichkeit von Peaks (2)

- Überraschenderweise hat die Hamming Distanz keinen Einfluss auf das Auffinden der Peaks
- Probleme nur bei zu klein gewähltem σ



Ähnlichkeit von Peaks (3)

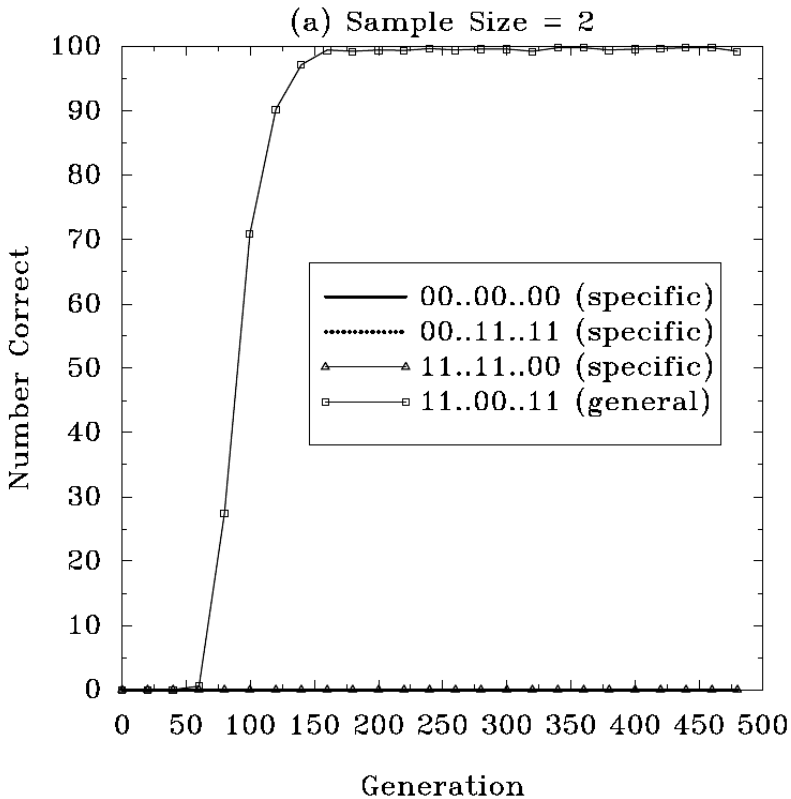


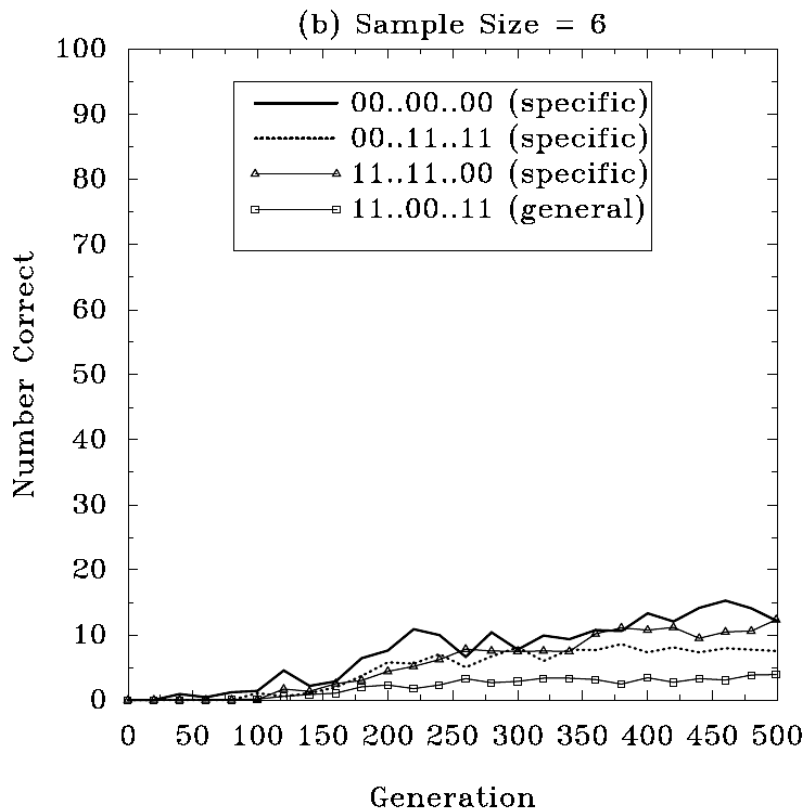


Generalisierung

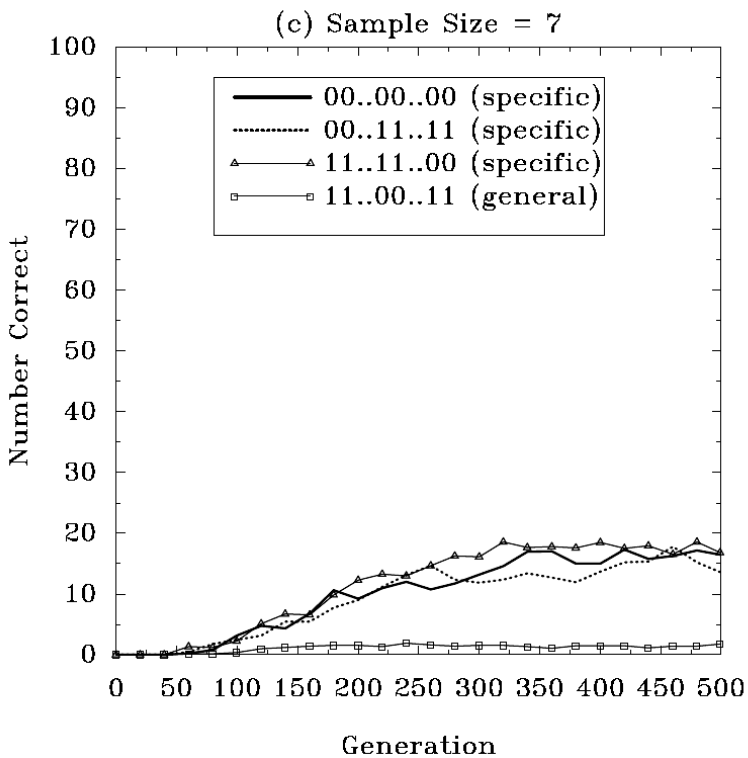
- Durch variieren von σ Generalisierung der Antikörper untersuchen
- Idee: Generiere 3 Antigen-Peaks so, dass ein Antikörper alle 3 zu einem gewissen Grad matcht
- Beispiel: AG‘s: 000, 011, 110
  passender AB: 101
- Da im Immunsystem weniger Antikörper als mögliche Antigene existieren muss es viele gemeinsame Muster der Antigene erkennen

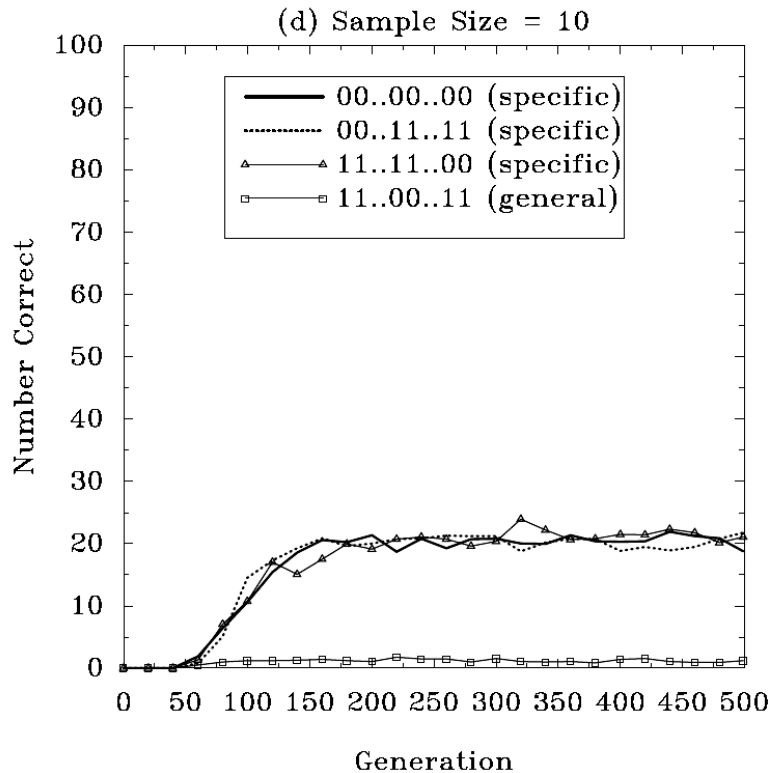
Generalisierung (2)





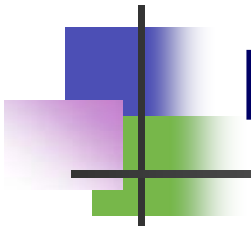
Generalisierung (3)





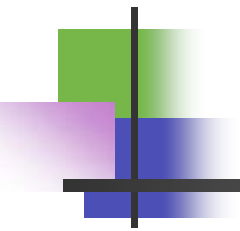
Zusammenfassung

- Binäres Immunsystem Modell mit Matching Funktion die spezifische Lösungen mehr als Unspezifische belohnt
  - Vermeiden von „Self-recognition“
- Fähigkeit eines GA zur Erkennung gemeinsamer Muster und Aufrechterhaltung von Unterschiedlichkeiten
  - Analog zur Bakterienerkennung erkennt auch der GA gemeinsame Muster
  - GA in der Lage verschieden Peaks von AG‘s zu erkennen und passende AB‘s aufrecht zu erhalten
  - Ähnlichkeiten von Peaks beeinflusst die Erkennung nicht
  - Etwa 15 AB‘s werden pro Peak benötigt um stabil zu bleiben



Fazit

- Immunsystem Algorithmus ist gute Approximation des realen Immunsystems
- Biologische Abläufe werden nachvollziehbarer
- Allerdings ist vorgestellter Algorithmus nicht der einzige der verschiedene Peaks erkennen und halten kann (z.B. Fitness Sharing Algorithmus)



Vielen Dank für die Aufmerksamkeit

Fragen?