

Feinkörnig parallele Varianten des Insel-Modells Genetischer Algorithmen

Dirk Eichberg, Udo Kohlmorgen, Hartmut Schmeck
Institut für Angewandte Informatik und Formale Beschreibungsverfahren
Universität Karlsruhe, D-76128 Karlsruhe

E-mail: {eichberg, kohlmorgen, schmeck}@aifb.uni-karlsruhe.de

Zusammenfassung

In diesem Beitrag berichten wir über erste Ergebnisse einer systematischen Untersuchung verschiedener Möglichkeiten, Genetische Algorithmen auf dem massiv parallelen Rechner MasPar1 zu implementieren. Durch Aufteilung des Prozessorfeldes in Teilfelder läßt sich das Insel-Modell Genetischer Algorithmen simulieren mit Nachbarschaften zwischen Inseln, die durch die Verbindungsstruktur des Feldes in natürlicher Weise gegeben sind. Dadurch läßt sich sowohl der Einfluß von Parametern wie Migrationsrate, Migrationsintervall und Migrationsrichtung untersuchen als auch Möglichkeiten der parallelen Ausführung genetischer Operatoren auf (Teil-)Populationen. Die Ergebnisse zeigen, daß diese feinkörnige Parallelisierung das Optimierungsverhalten Genetischer Algorithmen erheblich verbessern kann.

1 Einleitung

Genetische Algorithmen sind auf dem Prinzip der natürlichen Evolution beruhende Heuristiken, die zur Lösung von Such- und Optimierungsproblemen und zur Lösung anderer schwerer Probleme verwendet werden. Ihre universelle Anwendbarkeit sowie das für viele Anwendungen nachgewiesene gute Optimierungsverhalten haben zu einem starken Interesse an diesem Algorithmientyp geführt (vgl. z.B. [5] und [8]). Wird die Evolution großer Populationen potentieller Lösungen über viele Generationen betrachtet, entsteht ein sehr hoher Rechenaufwand. Um trotzdem in vertretbarer Zeit zu guten Ergebnissen zu kommen, liegt es nahe, Parallelrechner einzusetzen.

Während bei Parallelisierungen üblicherweise das funktionale Verhalten des sequentiellen Algorithmus erhalten bleiben soll, führen die bekanntesten Ansätze der Parallelisierung Genetischer Algorithmen auch zu neuen algorithmischen Strukturen. So werden im *Insel-Modell* Genetische Algorithmen auf mehreren unabhängigen (Teil-)Populationen gleichzeitig ausgeführt, verbunden mit der Möglichkeit, regelmäßig

gute Individuen zwischen 'benachbarten' Inseln auszutauschen (vgl. [1], [6], [10] und [11]).

Neben diesem grobkörnigen parallelen Ansatz gibt es mit dem *Nachbarschafts-Modell* auch eine feinkörnig parallelisierte Variante Genetischer Algorithmen, bei der die Population auf die Prozessoren eines großen (in der Regel gitterverbundenen) Prozessorfeldes verteilt wird. Die dadurch entstehende räumliche Anordnung der Population erlaubt die Verwendung von 'Nachbarschaften' bei der Selektion von Individuen zur Erzeugung von Nachkommen für die nächste Generation (vgl. [2], [7] und [9]).

In diesem Beitrag wird über einen neuen Parallelisierungsansatz berichtet, bei dem das Insel-Modell Genetischer Algorithmen auf dem massiv parallelen Rechner MasPar1 implementiert wird. Die Inseln und ihre Nachbarschaften ergeben sich dabei in natürlicher Weise durch Aufteilung des Prozessorfeldes in Teilfelder, die dann zur feinkörnig parallelen Ausführung der genetischen Operatoren genutzt werden.

Anhand mehrerer Anwendungsprobleme (u.a. das Traveling Salesperson Problem sowie kapazitätsbeschränkte Projektplanungsprobleme) wird das Optimierungsverhalten derartig parallelisierter Genetischer Algorithmen systematisch untersucht, insbesondere interessiert der Einfluß von Zahl und Größe der Teilpopulationen sowie von Migrationsrate, -intervall und -richtung auf den Verlauf der Evolution und auf die Qualität der erzeugten Lösungen.

Nach einer Darstellung des klassischen Genetischen Algorithmus und der Parallelisierungsstrategien in den nächsten beiden Abschnitten wird in Abschnitt 4 die von uns gewählte Implementierung auf dem MasPar-Rechner genauer beschrieben. Abschnitt 5 enthält erste Ergebnisse der durchgeführten systematischen Untersuchungen.

2 Klassischer Genetischer Algorithmus

In diesem Abschnitt wird die Struktur des klassischen Genetischen Algorithmus erläutert, wie er z.B. von Goldberg in [5] beschrieben wird.

Generell operiert ein Genetischer Algorithmus auf einer Menge von *Individuen*, auch *Population* genannt. Jedes Individuum der Population repräsentiert durch eine binäre Folge fester Länge eine potentielle Lösung des betrachteten Problems, die *Güte* des Individuums entspricht der Güte dieser Lösung bezüglich der zu optimierenden Zielfunktion.

Aus der Population werden *Eltern* selektiert, um *Nachkommen* für die folgende Generation zu erzeugen. Das klassische *Selektionsverfahren* entspricht der Verwendung eines Rouletterades, das für jedes Individuum einen Sektor enthält, dessen Größe gemäß der relativen Güte des Individuums innerhalb der Population gewählt wird. Dadurch bestimmt die relative Güte der Individuen die Wahrscheinlichkeit, mit der sie als Eltern selektiert werden. Es gibt eine Vielzahl anderer Selektionsverfahren, bei denen beispielsweise nicht die relative Güte, sondern der Rang eines Individuums

innerhalb der Population die Selektionswahrscheinlichkeit bestimmt oder bei denen nur die besten x Prozent der Individuen gewählt werden.

Selektierte Eltern werden zufällig gepaart und produzieren durch *Crossover* ein bis zwei Nachkommen. Durch die Kombination von Teilfolgen beider Elternteile erhalten die Nachkommen Eigenschaften beider Eltern. Die Generation der Nachkommen wird anschließend einer *Mutation* unterworfen, die allerdings meist nur mit einer geringen Wahrscheinlichkeit genetische Informationen verändert.

Nach Ermittlung der Güte der Individuen der so entstandenen neuen Population wird dieser Prozeß solange wiederholt, bis sich keine Verbesserungen bzw. Veränderungen mehr ergeben, oder bis ein anderes Abbruchkriterium erfüllt ist. Der klassische Genetische Algorithmus hat also folgende Struktur:

Genetischer Algorithmus

Erzeuge Anfangspopulation

BEWERTUNG der Individuen

REPEAT

 SELEKTION von Eltern

 CROSSOVER zur Erzeugung von Nachkommen

 MUTATION der Nachkommen

 BEWERTUNG der neuen Population

UNTIL Abbruchbedingung erfüllt

3 Parallelisierung Genetischer Algorithmen

Betrachtet man den im vorigen Abschnitt beschriebenen Genetischen Algorithmus, so fällt auf, daß einige Komponenten leicht zu parallelisieren sind. Dazu gehören der Crossover, die Mutation und die Bewertung der Individuen. Bei genügend hohem Parallelitätsgrad lassen sich diese Schritte in konstanter Zeit ausführen, d.h. unabhängig von der Größe der Population. Dies gilt jedoch nicht für die Selektion, da die Bestimmung der relativen Güte einen globalen Informationsaustausch voraussetzt.

Beim Insel-Modell Genetischer Algorithmen (vgl. [10] und [11]) geschieht die Parallelisierung durch Aufteilung der Population auf mehrere Teilpopulationen, auf denen Genetische Algorithmen weitgehend unabhängig voneinander operieren. Dabei wird jeder Teilpopulation (Insel) ein Prozessor zugeordnet. Bei dieser grobkörnigen Parallelisierung entsteht also ein neuer Genetischer Algorithmus ohne globalen Informationsaustausch während der Selektion. Auf eine Parallelisierung der für große Populationen inhärent parallelen genetischen Operationen wird dabei verzichtet. Nach einer bestimmten Anzahl von Generationen, dem Migrationsintervall, wandern ausgewählte Individuen von Insel zu Insel, gemäß der Migrationsrate und festgelegter

Nachbarschaftsbeziehungen. Dadurch wird das genetische Material in allen Teilpopulationen aufgefrischt. Da die getrennte Entwicklung der Teilpopulationen zwischen den Migrationsvorgängen zu einer Nischenbildung führt, wird der Suchraum an verschiedenen Stellen intensiver durchsucht. Dies verringert das Problem der vorzeitigen Konvergenz der Gesamtpopulation und man erreicht eine größere Vielfalt des genetischen Materials.

Bei der anfangs beschriebenen feinkörnigen Parallelisierung bietet es sich an, die Verbindungsstruktur des verwendeten Parallelrechners auszunutzen. Wählt man auf einem gitterverbundenen Prozessorfeld die Eltern für die Erzeugung der Nachkommen nur aus benachbarten Prozessoren, so entsteht ein neuer Selektionsoperator mit rein lokalem Informationsaustausch. Dieses Nachbarschafts- oder auch Diffusions-Modell Genetischer Algorithmen wird in dieser Arbeit aber nicht weiter betrachtet.

Bei der Verwendung eines gitterverbundenen Prozessorfeldes für die feinkörnige Parallelisierung liegt es außerdem nahe, Eltern nicht zufällig zu paaren, sondern ein *dialleles Paarungsschema* anzuwenden, d.h. in einem $n \times n$ Prozessorfeld werden n Individuen selektiert und dann durch Verteilung über Zeilen und Spalten vollständig miteinander kombiniert. Dies führt zu einer wesentlich stärkeren Ausnutzung des genetischen Materials als im klassischen Genetischen Algorithmus.

4 Implementierung des Insel-Modells auf einem feinkörnigen Parallelrechner

Bei der von uns vorgenommenen feinkörnigen Parallelisierung Genetischer Algorithmen wird jedem Prozesselement des verwendeten MasPar1-Rechners mit 16 k Prozessoren ein Individuum zugeordnet, d.h. die Prozessorzahl bestimmt die Größe der Gesamtpopulation. Inselstrukturen ergeben sich dann durch Aufteilung des Prozessorfeldes in Teilfelder. Bei einer festen Populationsgröße von 16384 Individuen werden 1024, 256, 64, 16, 4 oder 1 Teilpopulation(en) mit 16, 64, 256, 1024, 4096 und 16384 Individuen betrachtet. In Abb. 1 sind 4 Inseln mit jeweils 4096 Individuen skizziert. Nachbarschaftsbeziehungen zwischen den Inseln ergeben sich automatisch durch die Verbindungsstruktur des Prozessorfeldes, dabei wurden die auf dem MasPar-Rechner vorhandenen Diagonalverbindungen allerdings nicht benutzt. Jede Insel hat daher vier Nachbarinseln. Die Migration findet nur zwischen benachbarten Inseln statt, d.h. eine durch die Migrationsrate bestimmte Zahl von Prozesselementen an den Rändern der Inseln schickt Individuen an benachbarte Inseln. Um den Einfluß dieser Migration genauer untersuchen zu können, lassen sich die erlaubten Migrationsrichtungen festlegen, z.B. nur nach Norden oder Norden und Osten oder in alle vier Richtungen. Auch das Migrationsintervall kann frei gewählt werden. Bei den von uns vorgenommenen Untersuchungen verwendeten wir eine Migrationsrate, die von der Größe der Teilpopulationen abhängt, und zwar werden jeweils pro Migrationsrichtung ca. 10% der Randprozessoren ausgesucht.

Durch die Aufteilung des Prozessorfeldes steht jeder Teilpopulation ein ihrer

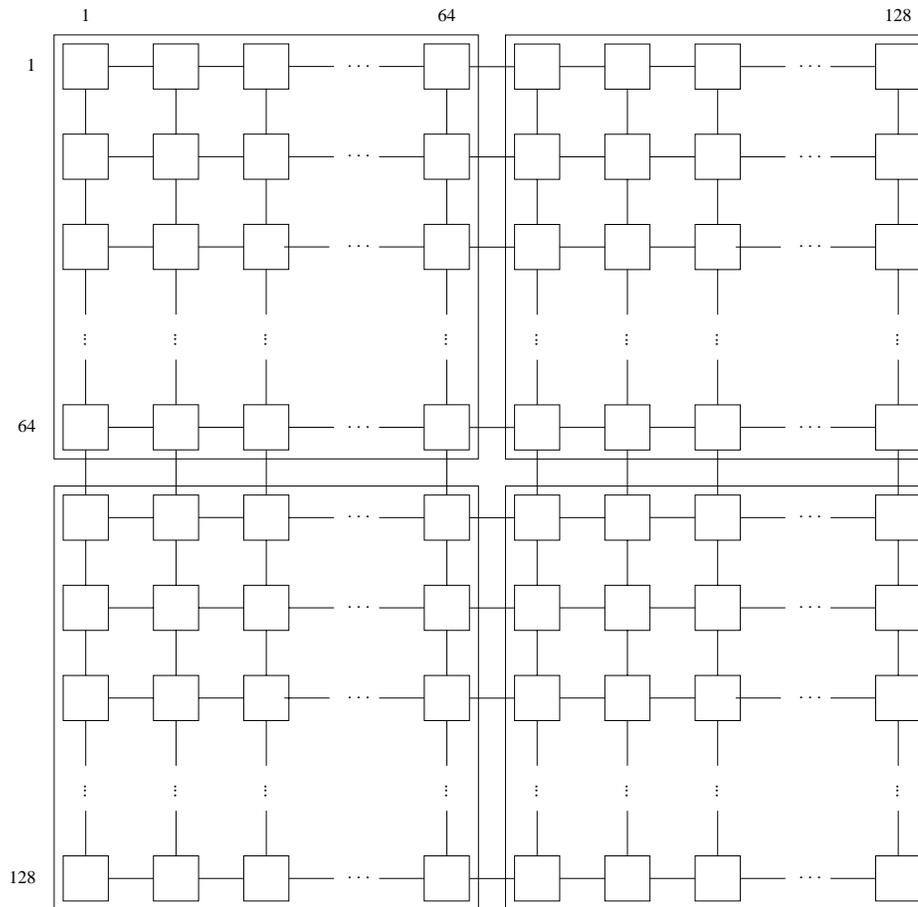


Abbildung 1: 4 Inseln mit je 4096 Individuen

Größe entsprechendes Teilfeld für die Ausführung der genetischen Operationen zur Verfügung. Als Paarungsschema wird das vollständige Diallel verwendet. Als Eltern werden entweder die n besten Individuen oder n nach dem gütebasierten Rouletteradverfahren (RRG) bestimmte Individuen selektiert.

5 Ergebnisse

In diesem Abschnitt berichten wir über einige Ergebnisse, der Anwendung des parallelisierten Genetischen Algorithmus auf das Traveling Salesperson Problem (TSP) und auf die kapazitätsbeschränkte Projektplanung (Ressource Constraint Project Scheduling Problem = RCPSP). Für das TSP gibt es viele Vergleichswerte und das RCPSP ist ein Problem, das im Operations Research wegen seiner vielfältigen praktischen Anwendungen auch mit großem Interesse betrachtet wird (siehe z.B. [3]).

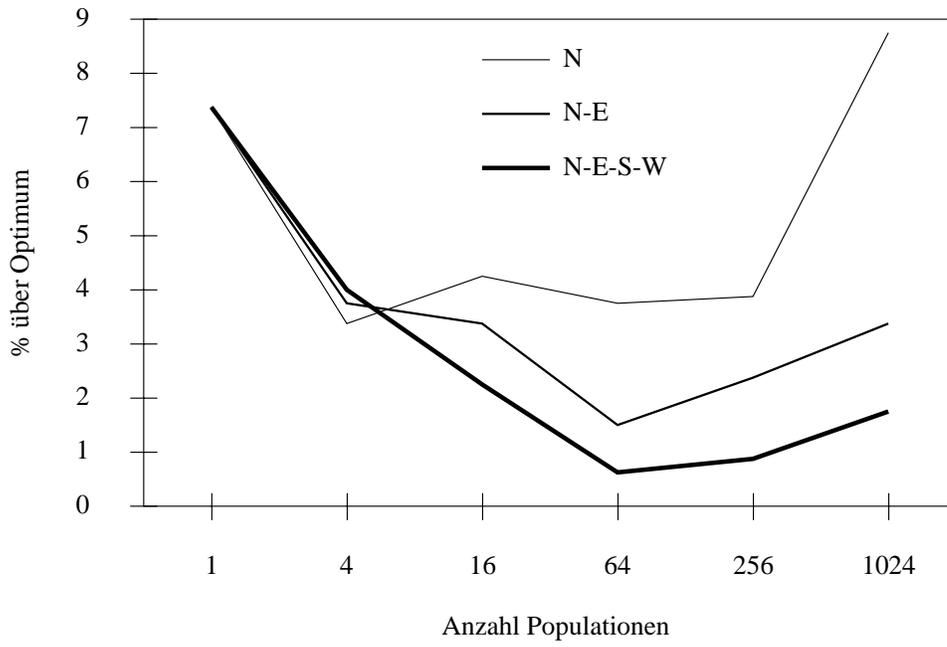


Abbildung 2: Einfluß der Migrationsrichtung

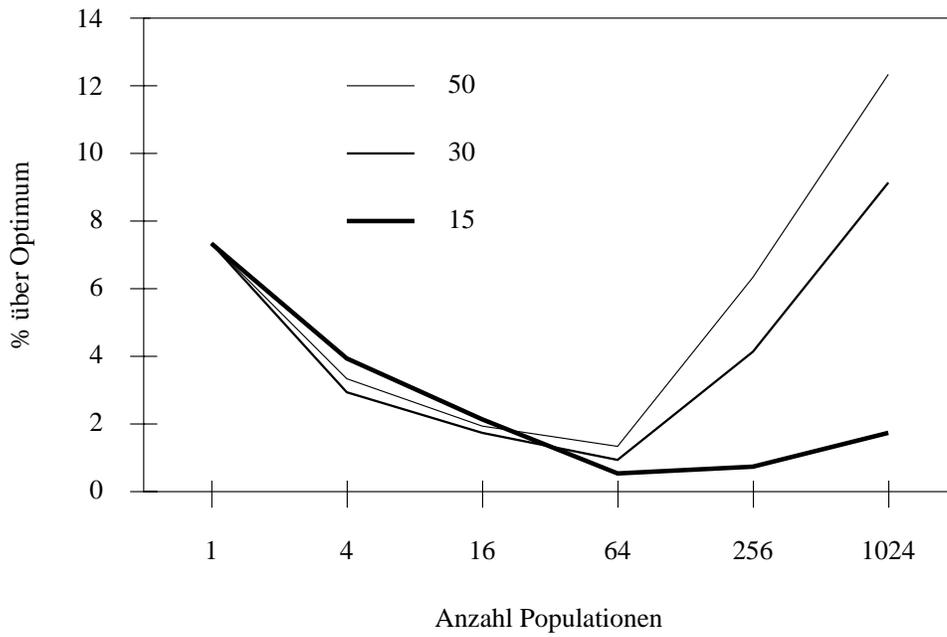


Abbildung 3: Einfluß des Migrationsintervalls

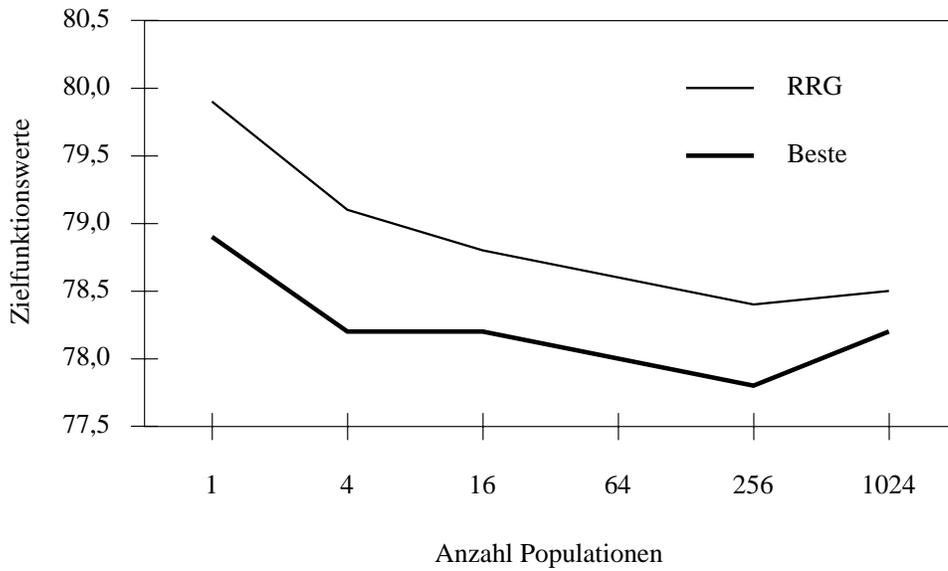


Abbildung 4: Vergleich Selektionsstrategien: Beste RRG

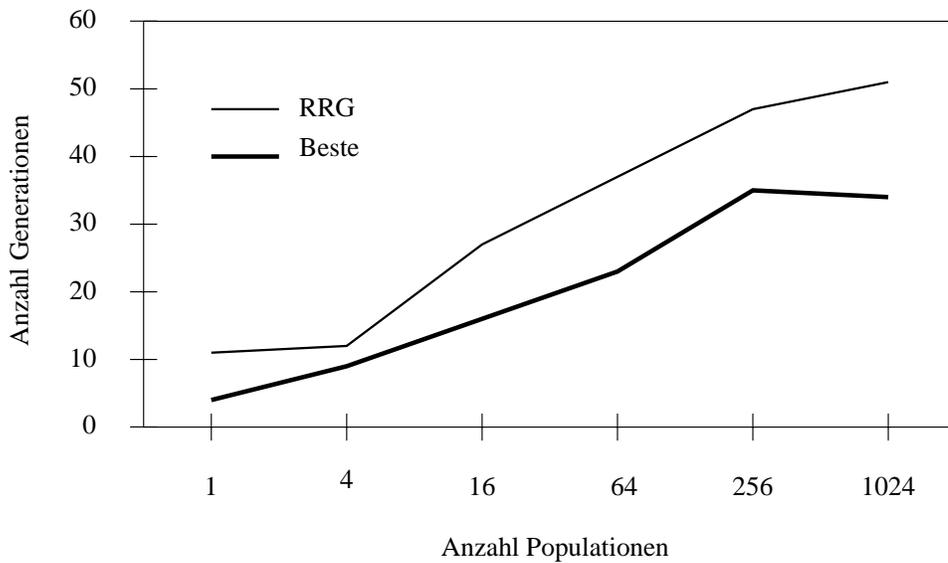


Abbildung 5: Vergleich Selektionsstrategien: Beste RRG

Für eine genaue Beschreibung der für diese Probleme verwendeten genetischen Repräsentierung sei auf [4] verwiesen. Abbildungen 2 bis 5 enthalten einen Teil der Ergebnisse unserer empirischen Untersuchungen. Dabei beziehen sich Abb. 2 und 3 auf ein TSP mit 51 Städten und Abb. 4 und 5 auf ein RCPSP mit 60 Jobs auf 4 Ma-

schinen. Für beide Probleme wurden die Daten Benchmarkbibliotheken entnommen. Die den Abbildungen zugrundeliegenden Zahlen sind Durchschnittswerte, die sich auf jeweils 4 unabhängige Durchläufe des Genetischen Algorithmus beziehen. Für die Migrationsrichtung hat sich gezeigt, daß die besten Ergebnisse erreicht werden, wenn Individuen in alle 4 Richtungen wandern (siehe Abb. 2). Ferner erkennt man, daß es günstig ist, die Gesamtpopulation in 64 oder 256 Teilpopulationen aufzuteilen. Ohne Aufteilung in Teilpopulationen erhält man in allen Fällen deutlich schlechtere Ergebnisse. Abb. 3 zeigt darüberhinaus, daß ein kleines Migrationsintervall einen positiven Einfluß auf die Ergebnisse hat, wobei die Aufteilung auf 64 Teilpopulationen wiederum die am besten geeignete ist. Dabei ist allerdings zu berücksichtigen, daß für 256 und 1024 Teilpopulationen der Genetische Algorithmus nach 600 Generationen abgebrochen wurde während in den anderen Fällen ein Abbruch aufgrund von Konvergenz erfolgte. Der Vergleich der beiden Selektionsstrategien (Abb. 4) ergibt einen deutlichen Vorteil der Auswahl der besten gegenüber dem Rouletteradverfahren. Auch hier erweist sich die Aufteilung der Gesamtpopulation auf 64 bzw. 256 Teilpopulationen als günstig. In Abb. 5 ist allerdings auch zu erkennen, daß viele Teilpopulationen mehr Generationen benötigen, um gute Ergebnisse zu erzielen.

6 Ausblick

Die bisherigen Ergebnisse haben gezeigt, daß mit der Implementierung des Insel-Modells Genetischer Algorithmen auf einem feinkörnigen Parallelrechner nicht nur ein hoher Zeitgewinn erzielt werden kann, sondern daß darüberhinaus der parallelisierte Genetische Algorithmus der klassischen Variante im Optimierungsverhalten überlegen ist, zumindest für die hier betrachteten Probleme. Durch Anwendung des Verfahrens für andere Problemgrößen und für Probleme anderen Typs ist zu untersuchen, ob die Aussagen über die günstige Zahl und Größe von Teilpopulationen, über Migrationsstrategien und über die Eigenschaften der verschiedenen Selektionsverfahren auch allgemeine Gültigkeit haben. Ferner liegt es nahe, das hier beschriebene feinkörnig parallelisierte Insel-Modell in seiner Leistungsfähigkeit mit dem Nachbarschafts-Modell zu vergleichen.

Literatur

- [1] J. P. Cohoon, S. U. Hedge, W. N. Martin, and D. Richards. Punctuated equilibria: A Parallel Genetic Algorithm. In J. J. Grefenstette(Ed.), Second International Conference on Genetic Algorithms. Lawrence Erlbaum Associates, (1987), 148-154.
- [2] R. J. Collins and D. R. Jefferson. Selection in massively parallel genetic algorithms. In R. K. Belew and L. B. Booker (Eds.), Fourth International Conference on Genetic Algorithms, Morgan Kaufmann, San Diego, CA, (1991), 244-248.

- [3] W. Domschke und A. Drexl. Einführung in Operations Research. Springer-Verlag, Berlin, Heidelberg, 1991.
- [4] D. Eichberg. Untersuchung des Insel-Modells Genetischer Algorithmen auf einem massiv parallelen Rechner. Diplomarbeit, Universität Karlsruhe, Institut für Angewandte Informatik und Formale Beschreibungsverfahren, in Vorbereitung.
- [5] D.E. Goldberg. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley Publishing Company, 1989.
- [6] B. Kröger, P. Schwenderling, and O. Vornberger. Parallel Genetic Packing on Transputers. In Stender, J.(Ed.), Parallel Genetic Algorithms: Theory & Applications, IOS Press, Amsterdam (1993), 151-185.
- [7] B. Manderick and P. Spiessens. Fine-grained parallel Genetic Algorithms. In J. D. Schaffer (Ed.), Third International Conference on Genetic Algorithms, Morgan Kaufmann, San Mateo, CA (1989), 428-433.
- [8] Z. Michalewicz. Genetic Algorithms + Data Structures = Evolution Programs. Springer Verlag, 1994.
- [9] M. Schwehm, Th. Oparterny, and K.-H. Kirsch. Platzierung von Makrozellen durch genetische Algorithmen auf verteilten und massiv parallelen Rechnern. Mitteilungen - Gesellschaft für Informatik e.V., Parallel-Algorithmen und Rechnerstrukturen, Workshop 1994, (1994), 69 -74.
- [10] R. Tanese. Distributed Genetic Algorithms. In J. D. Schaffer (Ed.), Third International Conference on Genetic Algorithms, Morgan Kaufmann, San Mateo, CA, (1989), 434-439.
- [11] L. D. Whitley and T. Starkweather. Genitor II: a Distributed Genetic Algorithm. Expt. Theor. Artif. Intell., Vol. 2, (1990), 189-214.