

Moderne Suchalgorithmen: Trends und Potential

Evolutionäre Suchalgorithmen für Sortier- und Kürzeste-Wege-Probleme

Dominik Schmid

18. Oktober 2014

Zusammenfassung

Die Analyse von Evolutionären Algorithmen stellt ein noch relativ junges Forschungsgebiet der Informatik dar, dennoch wird ihnen in der Zukunft ein sehr großes Potenzial, vor allem im Bereich des maschinellen Lernens, vorausgesagt. Mit diesem Beitrag soll hauptsächlich auf Basis des von Scharnow, Tinnefeld und Wegener verfassten Papers *The Analysis of Evolutionary Algorithms on Sorting and Shortest Paths Problems* eine Übersicht über Evolutionäre Algorithmen, insbesondere für Sortier- und Kürzeste-Wege-Probleme, gegeben werden [PM95].

Inhaltsverzeichnis

1	Einleitung	3
2	Evolutionäre Algorithmen	3
2.1	Konzept eines Evolutionären Algorithmus	3
2.2	Evolutionäre Algorithmen als Sortierverfahren	3
3	Analyse: Evolutionäre Algorithmen für Sortierprobleme	4
3.1	Untere Grenze für Fitnessfunktionen INV, HAM, EXC, LAS, RUN	4
3.2	Obere Grenze für Fitnessfunktionen INV, HAM, EXC, LAS	5
3.3	Erwartete Laufzeit der Fitnessfunktionen HAM, EXC und LAS	5
4	Analyse: Evolutionäre Algorithmen für Kürzeste-Wege-Problem	8
5	Fazit	9

1 Einleitung

Sortier- und Kürzeste-Wege-Probleme stellen in der Informatik alltägliche, gut und schnell mit speziellen Algorithmen lösbare Aufgaben dar. Einfache vergleichsbasierte Sortieralgorithmen wie der *Bubblesort-Algorithmus*, aber auch weitere, noch effizientere Verfahren wie *Mergesort* oder *Quicksort* mit einer erwarteten Laufzeit von $\Theta(n \log n)$ gehören schon früh zum Portfolio eines jeden Informatikstudenten. Eine besonders einfache Lösung des Kürzeste-Wege-Problems unter der Verwendung ausschließlich positiver Kantengewichte, liefert der *Algorithmus von Dijkstra*, der sich durch eine Laufzeit in $\mathcal{O}(n^2)$ bei n Knoten auszeichnet. Nichtsdestotrotz hat auch die Untersuchung dieser Probleme mit Hilfe von Evolutionären Algorithmen ihre Berechtigung. Gerade für eine sogenannte *Blackbox-Untersuchung*, bei der nur wenige Informationen über die zu untersuchenden Instanzen bekannt sind, haben Evolutionäre Algorithmen klare Vorteile.

In Kapitel 2 wird zunächst ein Überblick über die Bestandteile Evolutionärer Algorithmen und deren Ausgestaltung für Sortierprobleme gegeben. Die hierbei vorgestellten Messverfahren werden in Kapitel 3 hinsichtlich ihrer erwarteten Laufzeit im Zusammenspiel mit dem sogenannten $(1 + 1)$ EA genauer untersucht. In Kapitel 4 wird ein Ansatz zur Lösung des Kürzeste-Wege-Problems mithilfe Evolutionärer Algorithmen präsentiert. Abschließend soll die Arbeit mit einem Fazit abgerundet werden.

2 Evolutionäre Algorithmen

2.1 Konzept eines Evolutionären Algorithmus

Wie der Name „Evolutionärer Algorithmus“ bereits vermuten lässt, ist das Ziel eines solchen Verfahrens die Nachahmung des aus der Biologie bekannten Prinzips der Evolution. Rein formal wird ein Evolutionärer Algorithmus wie folgt definiert:

Definition 2.1. *Evolutionäre Algorithmen stellen eine Klasse von stochastischen Optimierungsverfahren dar.*

Der Ablauf sowie die für die Beschreibung des Algorithmus notwendigen Begriffe sind hingegen aus der Biologie entnommen:

Aus einem oder mehreren gegebenen Anfangsindividuen wird durch Mutations- und Crossoveroperationen eine Population entwickelt. Mittels einer vorher festgelegten Funktion wird der Fitnesswert eines jeden Individuums entwickelt und im Nachfolgenden entschieden, ob die Linie des Individuums verworfen oder weiterverfolgt wird. Sobald eine vorgegebene Güte durch ein Individuum erreicht wurde, bricht der Algorithmus ab. [SJI01]

2.2 Evolutionäre Algorithmen als Sortierverfahren

Für Evolutionäre Algorithmen, die speziell für Sortierverfahren modifiziert wurden, fassen wir jedes Individuum als ein Permutation π der Elemente 1 bis n auf. Ziel ist hierbei den maximalen Grad der Sortiertheit bezüglich einer gegebenen Permutation π_{opt} zu erlangen. Konkret heißt dies, dass wir eine gegebene Ausgangspermutation π_1 so lange verändern bis wir die optimale Permutation π_{opt} erhalten.

Um dieses Ziel zu erreichen stehen uns die folgenden beiden Mutationsoperatoren zur Verfügung:

1. `jump(i,j)`: Entnehme das Element an Position i und füge es anschließend vor dem j -tem Element der Permutation wieder ein
2. `exchange(i,j)`: Vertausche die beiden Elemente an Positionen i und j

Zur Auswertung einer Permutation hinsichtlich ihrer Fitness greifen wir auf die fünf am häufigsten betrachteten Fitnessfunktionen zurück:

- INV(π): $\# (i, j) : \pi(i) < \pi(j)$, wobei $1 \leq i < j \leq n$
- HAM(π): $\# i : \pi(i) = i$, $1 \leq i \leq n$
- EXC(π): $\min(\# \text{exchange}(i, j) : \pi(i) = id)$
- LAS(π): $\max(k : \pi(i_1) < \dots < \pi(i_k))$, wobei $i_1 < \dots < i_k$
- RUN(π): $1 + (\# i : \pi(i+1) < \pi(i))$

Alle diese Fitnessfunktionen haben eine Initialisierungslaufzeit in $\mathcal{O}(n^2)$ und eine Updatezeit von $\mathcal{O}(n \log n)$. Auch sei bemerkt, dass aufgrund der Symmetrie in der Menge aller Permutationen jede Fitnessfunktion $f_{\pi_{opt}}(\pi)$ zu $f(\pi) := f_{id}(\pi)$ vereinfacht werden kann [SJI04]

Mit diesen Erkenntnissen lässt sich nun der sogenannte *(1+1) Evolutionary Algorithm*, kurz (1+1) EA beschreiben:

- *Initialisiere zufällige Permutation π*
- *Wiederhole:*
 1. *Erstelle π' durch Mutation von π*
 2. *Ersetze π durch π' falls $f(\pi) \leq f(\pi')$ bei Maximierungsproblem und falls $f(\pi) \geq f(\pi')$ bei Minimierung*
 3. *Falls $\pi = \pi_{opt}$ Schleife abbrechen*

Für die Auswahl einer neuen Mutation wird der folgende Mutationsoperator verwendet:

Wähle k gemäß der Poissonverteilung $Poi_1(k) = \frac{1}{k!}e^{-1}$ aus und führe sequentiell $k + 1$ jump/exchange Operationen unter zufällig ausgewählten Paaren $(i, j) : 1 \leq i, j \leq n$ und $i \neq j$ durch.

3 Analyse: Evolutionäre Algorithmen für Sortierprobleme

Nachdem im vorherigen Kapitel der (1+1) EA mit verschiedenen Fitnessfunktionen eingeführt wurde, werden wir jetzt versuchen die Laufzeit der jeweiligen Algorithmen möglichst genau zu bestimmen. Es macht in diesem Zusammenhang nur Sinn die erwartete Laufzeit zu betrachten, da alle Algorithmen lediglich auf zufälligen Mutationen beruhen.

3.1 Untere Grenze für Fitnessfunktionen INV, HAM, EXC, LAS, RUN

Zunächst beweisen wir eine einfache untere Schranke, die für den (1+1) EA mit allen fünf beschriebenen Fitnessfunktionen gilt unter der Annahme, dass diese Funktionen jeweils nur ein globales Maximum beziehungsweise Minimum besitzen.

Theorem 1. *Die erwartete Laufzeit des (1+1) EA mit Fitnessfunktionen INV(π), HAM(π), RUN(π), LAS(π), EXC(π) liegt in $\Omega(n^2)$*

Beweis. Die Wahrscheinlichkeit mit einer optimalen Permutation zu beginnen liegt bei $\frac{1}{n!}$. Wir betrachten jetzt nur den letzten Mutationsschritt: Es existieren maximal zwei passende *jump*- und zwei passende *exchange*-Operationen, sodass für die Wahrscheinlichkeit, die optimale Permutation π_{opt} zu treffen, gilt:

$$Pr_{opt} \geq \frac{1}{2} \cdot \frac{2}{n(n-1)} + \frac{1}{2} \cdot \frac{2}{n(n-1)} = \frac{2}{n(n-1)}$$

Folglich ist die erwartete Laufzeit nach unten durch $\frac{n(n-1)}{2} = \Omega(n^2)$ beschränkt \square

3.2 Obere Grenze für Fitnessfunktionen INV, HAM, EXC, LAS

Nachdem wir nun eine untere Grenze gefunden haben, beweisen wir im nächsten Schritt, dass die folgende obere Schranke gilt.

Theorem 2. Erwartete Laufzeit des (1+1) EA mit Fitnessfunktionen $INV(\pi)$, $HAM(\pi)$, $LAS(\pi)$, $EXC(\pi)$ liegt in $\mathcal{O}(n^2 \log n)$

Beweis. Wir zeigen dieses Theorem zunächst nur für die Fitnessfunktion $INV(\pi)$

Sei (i, j) ein „inkorrektes“ Paar ($\pi(j) > \pi(i)$, wobei $i < j$). Definiere im Folgenden

- $a := \#k : \pi(k) < \pi(j)$ für $k \in \{i+1, \dots, j-1\}$
- $b := \#k : \pi(j) < \pi(k) < \pi(i)$ für $k \in \{i+1, \dots, j-1\}$
- $c := \#k : \pi(i) < \pi(k)$ für $k \in \{i+1, \dots, j-1\}$

Für $exchange(i, j) = exchange(j, i)$ verbessert sich die Fitness um $2b + 1$ und für $jump(i, j)$ beziehungsweise $jump(j, i)$ um $\pm(a - c) + b + 1$. Folglich existieren mindestens 3 mögliche Operatoren, die zu einer Fitnessverbesserung führen. Unter der Annahme, dass nur eine Mutation ausgeführt wird (Wahrscheinlichkeit $\frac{1}{e}$), gilt für die Wahrscheinlichkeit einer Fitnesserhöhung

$$Pr_{increase} \geq \frac{3m}{2en(n-1)}, \text{ wobei } m = \# \text{ inkorrekte Paare}$$

Insgesamt folgt für die erwartete Laufzeit von INV als obere Schranke:

$$\frac{2}{3}en^2 \sum_{1 \leq m \leq n(n-1)/2} 1/m = \frac{2}{3}en^2 H\left(\frac{n(n-1)}{2}\right) = \mathcal{O}(n^2 \log n)$$

Analog folgen die verbleibenden Beweise, da für $HAM(\pi) = k$ und $EXC(\pi) = k$ mindestens k passende *exchange* und für $LAS(\pi) = k$ genau k passende *jump* Operationen existieren [SJI04]. \square

3.3 Erwartete Laufzeit der Fitnessfunktionen HAM, EXC und LAS

Nachdem in den beiden Kapiteln 3.1 und 3.2 bereits schon relativ präzise eine untere und obere Schranke für vier der fünf ausgewählten Fitnessfunktionen bewiesen wurden, zeigen wir nun für drei dieser Funktionen, dass jeweils die obere Schranke auch der erwarteten Laufzeit entspricht. Dabei ist jedoch zu beachten, dass wir uns für die beiden Fitnessfunktionen HAM und EXC jeweils auf *exchange*-Operationen einschränken müssen, um die folgende Beweistechnik anwenden zu können:

Lemma 3.1. *Gilt für eine Fitnessfunktion f mit $k \in \Omega(n^\epsilon)$*

1. *Jede lokale Operation ändert den Fitnesswert maximal um Konstante c*
2. *$\exists k_0(n) : \forall k \leq k_0(n) : \text{Wahrscheinlichkeit für eine Fitnesserhöhung liegt in } \mathcal{O}(\frac{k}{n^2})$*

dann folgt als untere Grenze $\Omega(n^2 \log n)$

Beweis. Betrachte zunächst nur eine Fitnessänderung pro Mutationsschritt. Sei c' die Konstante der $\mathcal{O}(\frac{k}{n^2})$ Grenze und $f(\pi) \geq f(\pi_{opt}) - j$, so gilt für die erwartete Laufzeit einer Fitnesserhöhung $\frac{n^2}{c'j}$. Unter der Annahme es werde immer die maximale Fitnesserhöhung c ausgeführt, folgt als untere Grenze für die erwartete Laufzeit:

$$\frac{n^2}{c'} \cdot \left(\frac{1}{c} + \frac{1}{2c} + \dots + \frac{1}{\lfloor k/c \rfloor c} \right) = \Omega(n^2 \log n)$$

Mit den *Chernoff*-Grenzen folgt, dass lediglich eine exponentiell kleine Wahrscheinlichkeit für mehr als $3t$ Operationen in t Schritten besteht. Daher folgt unter Berücksichtigung des Vorfaktors $\frac{1}{3}(1 - o(1))$ analog die untere Grenze $\Omega(n^2 \log n)$. \square

Somit haben wir nun alle nötigen Vorkenntnisse beisammen, um das nun folgende Theorem zu beweisen:

Theorem 3. *Die erwartete Laufzeit des (1+1) EA nur mit exchange-Operationen liegt für Fitnessfunktionen $HAM(\pi)$ und $EXC(\pi)$ in $\Theta(n^2 \log n)$*

Beweis. Für den Beweis der oberen Grenze $\mathcal{O}(n^2 \log n)$ siehe *Theorem 2*. Nach [SJI04] beträgt der erwartete Fitnesswert von HAM nahezu 1. Jede *exchange*-Operation verbessert die Fitness um maximal 2. Für $HAM(\pi) = k$ folgt somit als Wahrscheinlichkeit für eine Fitnessverbesserung $\frac{k}{n} \cdot \frac{2}{(n-1)}$. Damit sind alle Voraussetzungen für *Lemma 3.1* erfüllt und es folgt als Größenordnung für die erwartete Laufzeit von HAM $\Theta(n^2 \log n)$. Um ein ähnliches Resultat für die Fitnessfunktion EXC folgern zu können benötigen wir das folgende Lemma, in welchem der Zusammenhang der Fitnessfunktionen HAM und EXC deutlich wird:

Lemma 3.2. *Wenn $EXC(\pi) = k > 0$, dann folgt $n - 2k \leq HAM(\pi) \leq n - k - 1$*

Beweis. Die Permutation π besteht aus $n - k < k$ Zyklen, wobei maximal $n - k - 1$ Zyklen die Länge 1 besitzen. Hieraus folgt direkt $HAM(\pi) \leq n - k - 1$. Für $k \geq \frac{n}{2}$ ist die untere Grenze trivial, daher o.E. $n - k > \frac{n}{2}$. Die Summe der Längen der $n - k$ Zyklen beträgt n . HAM wird genau dann minimal, wenn alle Zyklen, die keine Fixpunkte sind Länge 2 haben. Daraus folgt die untere Grenze $HAM(\pi) \geq n - 2k$. \square

Da die erwartete Fitness von HAM bei Initialisierung für $n\epsilon$ exponentiell klein ist, folgt mit *Lemma 3.2*, dass für EXC bei Initialisierung mit großer Wahrscheinlichkeit gilt:

$$EXC(\pi) \geq (n - n\epsilon)/2 = \Omega(n)$$

Wiederum mit *Lemma 3.2* folgt, dass $HAM(\pi)$ den „Fortschritt“ des (1+1) EA mit $EXC(\pi)$ angibt. Analog zu HAM ergibt sich daher eine erwartete Laufzeit von $\Theta(n^2 \log n)$. \square

Nachdem wir jetzt erfolgreich für HAM und EXC als erwartete Laufzeit $\Theta(n^2 \log n)$ gezeigt haben, beweisen wir im Folgenden, dass diese Grenze auch für die Fitnessfunktion LAS gilt.

Theorem 4. *Die erwartete Laufzeit des (1+1) EA mit Fitnessfunktionen $LAS(\pi)$ liegt in $\Theta(n^2 \log n)$*

Beweis. Für den Beweis der oberen Grenze $\mathcal{O}(n^2 \log n)$ siehe erneut *Theorem 2*. Wir wenden wiederum das Vorgehen aus *Lemma 3.1* an, um eine untere Grenze von $\Omega(n^2 \log n)$ zu zeigen. Die erste Voraussetzung ist offensichtlich erfüllt, da sowohl *exchange* und *jump*-Operationen die Fitness nur um jeweils maximal 2 erhöhen. Da die zweite Voraussetzung jedoch im Allgemeinen nicht gilt, ist es unser Ziel in geringer Zeit eine Permutation zu finden, die $\mathcal{O}(\frac{k}{n^2})$ erfüllt. Dazu benötigen wir zunächst die zwei folgenden Definitionen sowie die Tatsache, dass die Position eines jeden Elementes in einer längsten aufsteigenden Sequenz (kurz: LAS) eindeutig ist:

Definition 3.3. Sei $\text{LAS}(\pi) = n - k$ und bezeichne $\text{pos}(i, \pi)$ die eindeutige Position eines Elements i in einer beliebigen LAS. Offensichtlich existieren $n - k$ Funktionswerte ungleich 0 und $n - 2k$ Elemente mit nur einmal vergebenen Werten. Bezeichne die übrigen $2k$ Elemente als *outsiders*.

Definition 3.4. Seien a_1, \dots, a_m die Elemente, die in jeder LAS vorkommen. Die Elemente zwischen a_i und a_{i+1} werden *spaces* genannt.

Da nur die $2k$ *outsiders* durch *exchange* oder *jump*-Operationen die Fitness erhöhen können ist es Ziel zu zeigen, dass jeder *space* maximal $\mathcal{O}(1)$ *outsiders* enthält.

Lemma 3.5. Nach $\Theta(n \log^2 n)$ Schritten befindet sich mit großer Wahrscheinlichkeit höchstens ein *outsider* pro *space*, so dass nach $\Theta(n^2 \log n)$ Schritten mit Wahrscheinlichkeit $1 - o(1)$ maximal 6 *outsiders* pro *space* enthalten sind.

Beweis. Sei $n - \text{LAS}(\pi) = k \leq k_0 = n^\epsilon$ mit $0 < \epsilon < \frac{1}{3}$. Das *Sammelbildproblem* besagt, dass mit Wahrscheinlichkeit $1 - o(n)$ für jedes Element mindestens eine Operation ausgeführt wird, wobei wir nur die letzte dieser Operationen betrachten. Die Wahrscheinlichkeit für einen *jump* in einen *space* der Länge ≥ 1 liegt in $\mathcal{O}(n^{2\epsilon-1})$, woraus folgt, dass die Wahrscheinlichkeit mit $\mathcal{O}(n^{4\epsilon-2})$ für einen *space* der Länge ≥ 2 mit $\mathcal{O}(n^{2\epsilon-1} \log^2 n)$ verschwindend gering ist. Die Wahrscheinlichkeit für eine Längenerhöhung eines *space* liegt in $\Omega(\frac{1}{n})$ und für eine Verringerung in $\mathcal{O}(n^{\epsilon-2})$. Daher sind bei $\mathcal{O}(n^2 \log n)$ Schritten 2 oder mehr Verlängerungen mit Wahrscheinlichkeit $\mathcal{O}(n^{3\epsilon-2} \log n) = o(1)$ ist zu vernachlässigen. Haben wir nun für einen *space* eine Operation vorliegen, so ist diese mit Wahrscheinlichkeit $p := \mathcal{O}(n^{\epsilon-1})$ eine Längenerhöhung. Um zu klären mit welcher Wahrscheinlichkeit ein *space* zuerst die Länge 6 erreicht bevor er leer wird formulieren wir diesen Sachverhalt als ein *Gambler's ruin problem* mit $t = \frac{1-p}{p} = \Omega(n^{1-\epsilon})$ und damit

$$q = \frac{(t-1)}{t^6-1} = \mathcal{O}(n^{5\epsilon-5})$$

Bei $\mathcal{O}(n^2 \log n)$ Schritten ist die Wahrscheinlichkeit somit nach oben durch $\mathcal{O}(n^{6\epsilon-3} \log n) = o(1)$ beschränkt und die Behauptung gezeigt. [SJI04] \square

Damit ist nun auch die zweite Voraussetzung von *Lemma 3.1* erfüllt und somit $\Theta(n^2 \log n)$ als Größenordnung der erwarteten Laufzeit bewiesen. \square

Für die Fitnessfunktion RUN kann gezeigt werden, dass sie unter bestimmten Voraussetzungen eine exponentielle Laufzeit aufweist und damit als Fitnessfunktion für den (1+1) EA eher ungeeignet ist. Die genauen Voraussetzungen und der dazugehörige Beweis werden in [SJI04] sehr ausführlich dargestellt. Abschließend sei zu diesem Kapitel bemerkt, dass für die Fitnessfunktion INV sowie die Funktionen HAM und EXC keine genaueren Laufzeitgrenzen als die in Kapitel 3.1 und 3.2 bewiesenen Schranken von $\Omega(n^2)$ und $\mathcal{O}(n^2 \log n)$ bekannt sind.

4 Analyse: Evolutionäre Algorithmen für Kürzeste-Wege-Problem

Im Folgenden soll versucht werden den (1+1) EA auf das Kürzeste-Wege-Problem in diskreten Graphen anzuwenden. Hierzu fassen wir die Distanzwerte eines Graphen einer in Distanzmatrix $D = (d_{ij})_{1 \leq i, j \leq n}$ mit $d_{ij} \in \mathbb{N} \cup \{\infty\}$ auf

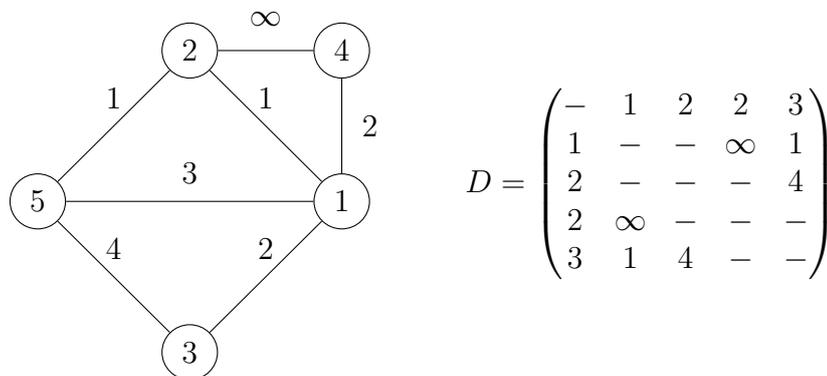


Abbildung 1: Diskreter Graph G mit dazugehöriger Distanzmatrix D

Als Suchraum betrachten wir die Vektoren $v = (v_1, \dots, v_{n-1}) \in \{1, \dots, n\}^{n-1}$ mit $v_i \neq i$. Wir setzen den Knoten n als Wurzel fest und nehmen v_i jeweils als Vorgänger von i im Kürzeste-Wege-Baum an. Als lokale Operation verstehen wir die Änderung einer Komponente v_i des Vektors v in $v'_i \in \{1, \dots, n\} - \{i, v_i\}$, sodass sich der Mutationsoperator des (1+1) EA wie folgt auffassen lässt:

Wähle k gemäß der Poissonverteilung mit $Poi_1(k) = \frac{1}{k!}e^{-1}$ und führe sequentiell $k + 1$ lokale Operationen aus.

Analysen in [SJI04] zeigen, dass es für diese Art von Problem effizienter ist einen *multi-objective*-basierten Optimierungsansatz zu wählen. Definiere eine partielle Ordnung auf \mathbb{R}^{n-1} :

$$f(v) \preceq f(v') \Leftrightarrow f_i(v) \leq f_i(v') \quad \forall i \in [n-1]$$

Ein mit dieser Ordnung minimaler Vektor wird auch *Pareto optimal* genannt. Lege nun den *multi-objective* (1+1) EA für das Kürzeste-Wege-Problem wie folgt fest:

- Initialisiere zufälligen Vektor $v \in \{1, \dots, n\}^{n-1}$
- Wiederhole:
 1. Erstelle v' durch Mutation von v
 2. Ersetze v durch v' falls $f(v') \preceq f(v)$
 3. Falls v einen *Pareto optimalen* Gewichtsvektor beschreibt, dann Schleife abbrechen

Analog zu *Theorem 1* lässt sich eine untere Schranke der erwarteten Laufzeit von $\Omega(n^2)$ zeigen. Betrachte hierzu wiederum nur die Wahrscheinlichkeit der letzten ausgeführten lokalen Operation. Ausführlich zeigen wir im Folgenden eine gültige obere Schranke:

Theorem 5. Die erwartete Laufzeit des *multi-objective* (1+1) EA liegt in $\mathcal{O}(n^3)$.

Beweis. Seien $t_i :=$ Minimum an Kanten im kürzesten $s - i$ -Weg, $m_j := \#\{i \mid t_i = j\}$ und $T := \max(j \mid m_j > 0)$. Wir zeigen als obere Grenze

$$en^2 \sum_{1 \leq j \leq T} (\ln m_j + 1)$$

die maximal wird für $m_1 = \dots = m_{n-1} = 1$. Wir nehmen an, dass $f_i(v) = l_i^* \forall i$ mit $t_i < t$ bereits gilt und betrachten nun die erwartete Laufzeit bis $f_i(v) = l_i^* \forall i$ mit $t_i \leq t$ ebenfalls erfüllt ist. Für alle r Positionen i mit $t_i = t$ und $f_i(v) > l_i^*$ existiert jeweils ein passender Vorgänger j mit $t_j = t - 1$. Somit besteht eine Wahrscheinlichkeit von $\frac{r}{e^{(n-1)(n-2)}}$ für eine „gute“ Mutationen. Insgesamt ist die erwartete Laufzeit für eine solche Phase nach oben durch $en^2(\ln m_t + 1)$ beschränkt. Da $1 \leq t \leq T$ folgt durch Summation über alle t die obere Grenze. \square

5 Fazit

Wir haben nun erfolgreich die Anwendung evolutionärer Algorithmen auf Sortier- und Kürzeste-Wege-Probleme untersucht. Die Analysen haben jedoch gezeigt, dass wir in beiden Fällen mit der erwarteten Laufzeit des (1+1) EA noch etwa einen Faktor n von der worst case Laufzeit eines problemspezifischen Algorithmus entfernt sind. In der Praxis sind daher evolutionäre Algorithmen nicht den traditionellen Verfahren vorzuziehen. Man sollte allerdings bedenken, dass dies auch nicht das Ziel unserer Untersuchungen war. Vielmehr sollen Grundlagen auf dem noch jungen Gebiet der evolutionären Algorithmen geschaffen werden, die weiterführenden Analysen dienlich sind.

Bibliographie

- [PM95] O. Petersson and A. Moffat. A framework for adaptive sorting. *Discrete Appl. Math.*, 59:153–179, 1995.
- [SJI01] Tinnefeld K. Scharnow J. and Wegener I. Theoretical aspects of evolutionary algorithms. *Int. Colloq. on Automata, Languages, and Programming – ICALP*, 2076:64–78, 2001.
- [SJI04] Tinnefeld K. Scharnow J. and Wegener I. Journal of Mathematical Modelling and Algorithms. *Journal of Combinatorial Theory-Series B*, 67:349–366, 2004.