

Design und Analyse randomisierter Suchheuristiken Populationen und Cliques

Dissertation
zur Erlangung des Grades eines
Doktors der Naturwissenschaften
der Universität Dortmund
am Fachbereich Informatik
von

Tobias Storch

Dortmund
2007

Tag der mündlichen Prüfung: 8. Februar 2007
Dekan: Prof. Dr. Peter Buchholz
Gutachter: Prof. Dr. Ingo Wegener, Juniorprof. Dr. Thomas Jansen

DANKSAGUNG

Diese Dissertation wäre ohne die finanzielle Unterstützung

der Deutschen Forschungsgemeinschaft im Rahmen des Projekts „Theoretische Analyse der Robustheit stochastischer Optimierungsprozesse“,

der Deutschen Forschungsgemeinschaft im Rahmen des Sonderforschungsbereichs „Design und Management komplexer technischer Systeme mit Methoden der Computational Intelligence“ und

der German-Israeli Foundation for Scientific Research and Development im Rahmen des Projekts „Robustness Aspects of Algorithms“

nicht möglich gewesen.

Danken möchte ich an dieser Stelle

Stefanie Storch,

Ingo Wegener,

Stefan Droste,

Jens Jägersküpper,

Margot Storch,

Uwe Storch und

allen anderen, die mich auf vielfältige Art und Weise unterstützt haben.

Dortmund, Februar 2007

Tobias Storch

INHALTSVERZEICHNIS

1	Einblick	1
1.A	Randomisierte Suchheuristiken	2
1.B	Analyse	5
1.C	Veröffentlichungen und Eigenanteil	9
2	Nachkommenpopulationen in evolutionären Algorithmen	11
2.A	Kleine Nachkommenpopulationen	12
2.B	Große Nachkommenpopulationen	15
2.C	Typische Zielfunktion	19
2.D	Erweiterungen der Methode der f -basierten Partition	19
2.E	Weder kleine noch große Nachkommenpopulationen	23
3	Elternpopulationen in evolutionären Algorithmen	29
3.A	Typische Zielfunktion	31
3.B	Erweiterungen der Methode der f -basierten Partition	38
3.C	Hierarchieresultat	42
4	Transformationen der Zielfunktionen und randomisierte Suchheuristiken	47
4.A	Alle randomisierten Suchheuristiken	47
4.B	Black-Box-Algorithmen und -Komplexität	50
4.C	Rangbasierte randomisierte Suchheuristiken	53
4.D	Nicht funktionswertbasierte randomisierte Suchheuristiken	58
4.E	Nicht rangbasierte randomisierte Suchheuristiken	59
5	Cliquenproblem und randomisierte Suchheuristiken	63
5.A	Approximierung für alle Graphen	66
5.B	Optimierung für spärliche Graphen	71

5.C	Optimierung für spärliche semizufällige Graphen	73
5.C.1	Kanten entfernen	73
5.C.2	Kanten einfügen	79
5.D	Optimierung für planare und zufällige planare Graphen	84
5.D.1	Weitere randomisierte Suchheuristiken	91
5.D.2	Weitere semizufällige Graphen	96
5.E	Black-Box-Algorithmen und -Komplexität	98
6	Ausblick	101
7	Notationen, Gleichungen und Ungleichungen	103
7.A	Notationen	103
7.B	Gleichungen und Ungleichungen	103
	Literaturverzeichnis	105

1 EINBLICK

Das Design von Algorithmen und ihre Analyse gehören zu den fundamentalen Aufgaben der Informatik. Dabei ist das Berechnen der Optima einer (Ziel)funktion (engl. objective function) $f : \mathcal{A} \rightarrow \mathcal{B}$ eines der wichtigsten Probleme – in Theorie wie Praxis.

Der Suchbereich (engl. decision space) \mathcal{A} kann kontinuierlich, diskret oder sogar endlich sein, der Wertebereich (engl. objective space) \mathcal{B} partiell oder sogar vollständig geordnet. Der (extreme) Suchbereich $\mathcal{A} = \{0, 1\}^n$, $n \in \mathbb{N}$, hat eine herausragende Bedeutung, da die Suchbereiche vieler Entscheidungs- und Optimierungsprobleme kanonischerweise durch eine endliche Auswahl von Objekten modelliert werden. Zu diesen algorithmischen Problemen gehören etwa die Aufgaben, einen minimalen Spannbaum (deterministisch in polynomieller Zeit möglich) oder eine größte Clique (vermutlich nicht deterministisch in polynomieller Zeit möglich) in einem Graphen zu berechnen (Wegener, 2005). Außerdem kann jeder endliche Suchbereich (im Wesentlichen) in einen der Suchbereiche $\mathcal{A} = \{0, 1\}^n$, $n \in \mathbb{N}$, transformiert werden. Der (extreme) Wertebereich $\mathcal{B} = \{0, 1\}$ resultiert aus Entscheidungsproblemen. Bei Optimierungsproblemen ist – zumindest im monokriteriellen (engl. single-objective) Fall – irgendein Element entweder aus $\arg \max\{f(x) \mid x \in \mathcal{A}\} \subseteq \mathcal{A}$ oder $\arg \min\{f(x) \mid x \in \mathcal{A}\} \subseteq \mathcal{A}$ zu bestimmen. Ohne Vorkenntnisse über die Funktion muss zur sicheren Bestimmung eines optimalen Suchpunkts der Funktionswert jedes Elements von \mathcal{A} einmal bestimmt werden. Dies ist mit 2^n Funktionsauswertungen für $\mathcal{A} = \{0, 1\}^n$, $n \in \mathbb{N}$, sehr aufwändig und für unendliche Suchbereiche \mathcal{A} unmöglich. Bei Kenntnis der Funktion (engl. one-shot scenario) bestimmt ein optimaler Algorithmus mit der ersten Funktionsauswertung einen optimalen Suchpunkt – wobei das Finden dieses Algorithmus aufwendig sein kann. Optimierungs- oder Approximationsalgorithmen werden im Allgemeinen für eine Klasse von Funktionen entworfen und analysiert.

Beim Design von Algorithmen für eine Klasse von Funktionen sind *spezielle* und *allgemeine* Suchverfahren zu unterscheiden. Spezielle Suchverfahren nutzen Vorkenntnisse über die zu optimierenden Funktionen. Sowohl die Klasse von Funktionen muss bekannt sein (engl. function-class scenario) als auch die Kapazität zum Entwurf eines Suchverfahrens verfügbar. Aussagen über die Qualität der berechneten Lösung und die Laufzeit eines solchen Algorithmus können dann typischerweise getroffen werden. Verwenden diese Algorithmen den Zufall werden sie als randomisierte Algorithmen bezeichnet. Für eine Reihe von Anwendungen sind beide Anforderungen nicht erfüllt – Informationen über die undurchschaubare Funktion sind nur durch zeitintensive Simulationen oder Experimente zu erlangen und Zeit oder Wissen zum Entwurf eines Suchverfahrens sind nur knapp. Allgemeine Suchverfahren nutzen möglichst wenig Vorwissen über die zu optimierenden Funktionen. Weder die Klasse

von Funktionen muss bekannt sein (engl. black-box scenario) noch die Kapazität zum Entwurf eines Suchverfahrens verfügbar. Aussagen über die Qualität der berechneten Lösung oder die Laufzeit einer solchen *Suchheuristik* können dann typischerweise nicht getroffen werden. Für eine Reihe von Anwendungen sind Suchheuristiken jedoch bereits erfolgreich eingesetzt worden. Im Fall der Verfügbarkeit spezieller Suchverfahren können diese immer besser als allgemeine Suchverfahren sein. Verwenden diese Heuristiken den Zufall werden sie als *randomisierte Suchheuristiken* bezeichnet.

1.A Randomisierte Suchheuristiken

Im Folgenden sei \mathcal{A} endlich und \mathcal{B} endlich und vollständig geordnet. Jede randomisierte Suchheuristik zur Optimierung einer Funktion $f : \mathcal{A} \rightarrow \mathcal{B}$ aus einer Klasse \mathcal{F} solcher Funktionen kann als Spezifizierung des folgenden Black-Box-Algorithmus aufgefasst werden (vergleiche Wegener, 2001).

Black-Box-Algorithmus

1. Setze $t := 0$.
2. Setze $t := t + 1$. Abhängig von $(x_1, f(x_1)), \dots, (x_{t-1}, f(x_{t-1}))$ bestimme eine Wahrscheinlichkeitsverteilung auf \mathcal{A} , wähle x_t bezüglich dieser Verteilung und bestimme $f(x_t)$.
3. Gehe zu 2.

Die (Art der) Wahl der Wahrscheinlichkeitsverteilungen in jedem Schritt $t \geq 1$ und für jede (mögliche) Historie $(x_1, f(x_1)), \dots, (x_{t-1}, f(x_{t-1}))$ beschreibt jede randomisierte Suchheuristik (eindeutig). Konzentriert sich jede dieser Wahrscheinlichkeitsverteilungen auf ein Element des Suchbereichs, liegt ein deterministischer, sonst ein randomisierter Black-Box-Algorithmus vor. Jeder randomisierte Black-Box-Algorithmus kann auch als eine Wahrscheinlichkeitsverteilung über deterministische Black-Box-Algorithmen angesehen werden.

Für eine randomisierte Suchheuristik A und eine Funktion $f : \mathcal{A} \rightarrow \mathcal{B}$ sei $T_{f,A}$ die diskrete Zufallsvariable, welche die Anzahl Funktionsauswertungen bis zur (erstmaligen) Auswertung eines Optimums von f beschreibt. Eine solche Suchheuristik wird als unendlicher stochastischer Prozess betrachtet. Sie enthält daher (in der Regel) kein Stoppkriterium, etwa eine maximale Anzahl durchzuführender Funktionsauswertungen, wie es in konkreten Implementationen notwendig ist.

Die *durchschnittliche erwartete Laufzeit* einer randomisierten Suchheuristik A zur Optimierung von Funktionen $f : \mathcal{A} \rightarrow \mathcal{B}$ einer Klasse \mathcal{F} beträgt $E_{\mathcal{F},A}^{\emptyset} := \sum_{f \in \mathcal{F}} E[T_{f,A}] / |\mathcal{F}|$. Somit beträgt die *durchschnittliche Black-Box-Komplexität* für \mathcal{F} (mit der Betrachtung aller Black-Box-Algorithmen) $E_{\mathcal{F}}^{\emptyset} := \min\{E_{\mathcal{F},A}^{\emptyset} \mid \text{Black-Box-Algorithmus } A\}$. Für die Klasse $\mathcal{F}_{\mathcal{A},\mathcal{B}}$ aller Funktionen $f : \mathcal{A} \rightarrow \mathcal{B}$ (engl. no-free-lunch scenario) konnten Wolpert und Macready (1997) beweisen, dass für zwei beliebige, nur unterschiedliche Elemente auswertende randomisierte Suchheuristiken A und B gilt: $E_{\mathcal{F}_{\mathcal{A},\mathcal{B}},A}^{\emptyset} = E_{\mathcal{F}_{\mathcal{A},\mathcal{B}},B}^{\emptyset}$.

Die *erwartete Laufzeit* einer randomisierten Suchheuristik A zur Optimierung von Funktionen einer Klasse \mathcal{F} beträgt (analog zur Algorithmik)

$$E_{\mathcal{F},A} := \max\{E[T_{f,A}] \mid f \in \mathcal{F}\}.$$

Somit beträgt die *Black-Box-Komplexität* für \mathcal{F} (mit der Betrachtung aller Black-Box-Algorithmen) (analog zur Komplexität der Algorithmik)

$$E_{\mathcal{F}} := \min\{E_{\mathcal{F},A} \mid \text{Black-Box-Algorithmus } A\}.$$

Droste et al. (2006) konnten einerseits konstruktiv nachweisen, dass es randomisierte Black-Box-Algorithmen gibt, die in einer erwarteten Laufzeit von höchstens $(|\mathcal{A}| + 1)/2$ jede Klasse von Funktionen optimieren und sie konnten andererseits konstruktiv nachweisen, dass die Black-Box-Komplexität einer speziellen Klasse von Funktionen $(|\mathcal{A}| + 1)/2$ beträgt. Für die Klasse NEEDLE aller Funktionen $\text{NEEDLE}_a : \mathcal{A} \rightarrow \mathcal{B}$, $a \in \mathcal{A}$, mit

$$\text{NEEDLE}_a(x) := \begin{cases} v_1 & , \text{ falls } x = a, \\ v_0 & , \text{ falls } x \neq a, \end{cases}$$

wobei $v_1, v_0 \in \mathcal{B}$ feste Elemente mit $v_1 > v_0$ sind, gilt beispielsweise:

Theorem 1.A.1 Die *Black-Box-Komplexität* von NEEDLE beträgt $(|\mathcal{A}| + 1)/2$.

Jeder deterministische Black-Box-Algorithmus benötigt eine (erwartete) Laufzeit von (mindestens) $|\mathcal{A}|$ zur Optimierung von NEEDLE.

Nach diesen allgemeinen Betrachtungen von randomisierten Suchheuristiken kommen wir zu Betrachtungen von speziellen randomisierten Suchheuristiken. Zu den vermutlich bekanntesten randomisierten Suchheuristiken gehören die *evolutionären Algorithmen* (engl. *evolutionary algorithms, evolutionary computation*) (EA), deren Entwurf von der natürlichen Evolution inspiriert ist. Sie benutzen den Zufall, um Konzepte der Biologie nachzubilden (vergleiche Schwefel, 1995). Aber auch die randomisierten lokalen Suchen (RLS) und die Metropolis-Algorithmen (MA) sind weit verbreitete randomisierte Suchheuristiken (vergleiche Aarts und Lenstra, 2003).

In der Regel hält eine randomisierte Suchheuristik einen Vorrat von Elementen des Suchbereichs, den so genannten Individuen, zusammen mit ihren Funktionswerten vor. Dieser Vorrat wird als *Elternpopulation* (engl. *parent population*) oder *Population* (engl. *population*) bezeichnet und ist eine Teilmenge der Historie. Mittels dieser Individuen und des Zufalls werden in der *Suche* (engl. *search*) einige neue Individuen erzeugt. Dieser Vorrat wird als *Nachkommenpopulation* (engl. *offspring population*) bezeichnet und ist eine Teilmenge der Historie. Aus der Eltern- und der Nachkommenpopulation wird in der *Selektion* (engl. *selection*) eine neue Elternpopulation bestimmt. Die nach $t \geq 1$ solchen Schritten vorliegende Population wird als t -te *Generation* (engl. *generation*) bezeichnet.

Im Folgenden sei $\mathcal{A} = \{0, 1\}^n$, $n \in \mathbb{N}$, und $\mathcal{B} = \mathbb{Z}$. Wir betrachten also die Maximierung pseudoboolescher Funktionen $f : \{0, 1\}^n \rightarrow \mathbb{Z}$. (Unsere Resultate sind sofort auf $\mathcal{B} = \mathbb{R}$ übertragbar.)

Zu den wohl populärsten und einfachsten evolutionären Algorithmen gehört der (1+1)-EA für pseudoboole Funktionen. Es bezeichne $\text{mutate}_{1/n}(x)$ für $x = x_1 \cdots x_n \in \{0, 1\}^n$ den globalen Suchoperator, der jedes Bit x_i , $1 \leq i \leq n$, in x unabhängig mit Wahrscheinlichkeit $1/n$ flippt (vergleiche Droste et al., 2002).

(1+1)-EA

1. Setze $t := 1$ und wähle $x_t \in \{0, 1\}^n$ zufällig gleichverteilt.
2. Setze $t := t + 1$ und $y_t := \text{mutate}_{1/n}(x_{t-1})$.
3. Falls $f(y_t) \geq f(x_{t-1})$, dann setze $x_t := y_t$ und sonst $x_t := x_{t-1}$.
4. Gehe zu 2.

Allgemeiner gibt die Notation $(\mu+\lambda)$ -EA (vergleiche Bäck et al., 1997), $\mu, \lambda \geq 1$, die Verwendung einer Elternpopulationsgröße μ und einer Nachkommenpopulationsgröße λ an. Diese Werte werden während der Optimierung nicht verändert. Das Plus „+“ deutet an, dass sich die nachfolgende Elternpopulation sowohl aus Elementen der Eltern- als auch der Nachkommenpopulation zusammensetzen kann. Ein Komma „,” gäbe dagegen an, dass die nachfolgende Elternpopulation nur Elemente der Nachkommenpopulation enthält. In beiden Fällen kommt in der Regel die *Elite-Selektion* zur Anwendung, welche die Elemente mit dem größten Funktionswert für die nachfolgende Elternpopulation auswählt. Dagegen wählt die *Metropolis-Selektion* mit einer gewissen Wahrscheinlichkeit, abhängig von der Differenz der Funktionswerte und einer „Temperatur“, auch Elemente mit geringerem Funktionswert für die nachfolgende Elternpopulation aus. In der Regel werden Eltern- und Nachkommenpopulationen von randomisierten Suchheuristiken durch eine Menge oder allgemeiner Multimenge von Elementen des Suchbereichs zusammen mit ihren Funktionswerten beschrieben. Bei den Elternpopulationen kommt der *Diversität*, ermittelt etwa durch die Summe der paarweisen Hammingabstände ihrer Individuen, eine besondere Bedeutung zu. Wir präzisieren dies an geeigneten Stellen. Die Individuen der ersten Elternpopulation werden typischerweise zufällig gleichverteilt gewählt.

Betrachten wir den Suchoperator $\text{mutate}_{1/n}(x)$ genauer. Dieser ist eine *Mutation* (engl. *mutation*), da sein Nachkomme von einem Elter abhängt. Hängt er von mehr als einem Elter ab, ist er eine *Kreuzung* (engl. *crossover*). Die Verwendung einer Mutation als wesentlichen Suchoperator anstatt einer Kreuzung stellt ein entscheidendes Merkmal von Evolutionsstrategien (engl. *evolution(ary) strategies*) (Schwefel, 1995) gegenüber genetischen Algorithmen (engl. *genetic algorithms*) (Holland, 1975) dar. Beide Ansätze entstanden unabhängig voneinander um 1968 in Deutschland und Amerika. Die Wahrscheinlichkeit von $1/n$ für das Flippen jedes Bits ist in Theorie wie Praxis der Standard. Für viele typische Funktionen und typische Klassen von Funktionen ist dies die oder eine optimale Wahl (vergleiche Droste et al., 2002). Im Erwartungswert flippt $\text{mutate}_{1/n}(x)$ genau ein Bit von x , aber jedes Element des Suchbereichs wird mit positiver Wahrscheinlichkeit erzeugt. Diese Eigenschaft charakterisiert einen globalen Suchoperator. Der Suchoperator, der in jedem Schritt ein zufällig gleichverteilt gewähltes Bit flippt, ist ein lokaler Suchoperator. Weiter werden mit $\text{mutate}_{1/n}(x)$ Elemente mit gleichem Hammingabstand zu x mit gleicher Wahrschein-

lichkeit generiert und Elemente mit geringerem Hammingabstand mit höherer Wahrscheinlichkeit als solche mit größerem Hammingabstand. Ein Duplikat von x wird mit Wahrscheinlichkeit $(1 - 1/n)^n$ erzeugt, wobei $6/17 \leq (1 - 1/n)^n \leq 1/e$ für $n \geq 11$ gilt. Eine spezielle genau k Bits flippende Mutation hat Wahrscheinlichkeit $(1/n)^k \cdot (1 - 1/n)^{n-k}$, wobei $1/n^k \leq (1/n)^k \cdot (1 - 1/n)^{n-k} \leq 1/(en^k)$ für $k \geq 1$ gilt. Eine mindestens k (irgendwelche) Bits flippende Mutation hat eine Wahrscheinlichkeit von höchstens $\binom{n}{k} \cdot \frac{1}{n^k} \leq 1/(k!)$ für $k \geq 0$ und mindestens $\binom{n}{k} \cdot \frac{1}{en^k} \geq 1/(ek!)$ für $1 \leq k \leq n$. Jansen und Wegener (2000) konnten jedoch nachweisen, dass die Wahl einer Mutationswahrscheinlichkeit von $1/n$ keineswegs optimal sein muss.

Mit dem Suchoperator geht ein Selektionsoperator einher, der Individuen aus der Population \mathcal{P} auswählt (vergleiche Bäck et al., 1997). Bei einer Populationsgröße von eins ist dies ein einfaches Unterfangen. Allgemein ist die *uniforme Selektion*, welche jedes Element $x \in \mathcal{P}$ zufällig gleichverteilt auswählt, der gewiss bekannteste Selektionsoperator für die Suche. Er benutzt die Funktionswerte der Individuen nicht. Ein weiterer bedeutender Selektionsoperator ist die *Roulette-Selektion*, welche jedes Element $x \in \mathcal{P}$ mit Wahrscheinlichkeit $f(x) / \sum_{x' \in \mathcal{P}} f(x')$ auswählt. Geeignet ist dieser Selektionsoperator nur für Funktionen $f : \{0, 1\}^n \rightarrow \mathbb{Z}^+$, er lässt sich jedoch kanonisch für die Anwendbarkeit auf beliebige Funktionen $f : \{0, 1\}^n \rightarrow \mathbb{Z}$ erweitern. Er benutzt den Funktionswerte der Individuen.

1.B Analyse

Schon kurz nach den ersten konkreten Implementierungen und Anwendungen von randomisierten Suchheuristiken, etwa den evolutionären Algorithmen, wurden erste Berechnungen über ihr Verhalten publiziert (vergleiche Holland, 1975; Schwefel, 1995). Diese sollten Anhaltspunkte für die Wahl ihrer Komponenten liefern. Wie die ersten beruhten auch die meisten folgenden Untersuchungen auf Modellannahmen (vergleiche Beyer et al., 2002; Droste et al., 2003). Einige Betrachtungen konzentrierten sich auf das lokale, andere auf das globale Verhalten. Beide Verfahren lassen jedoch keine konkreten Aussagen über die (erwartete) Laufzeit und damit die Effizienz einer randomisierten Suchheuristik zur Optimierung einer Funktion (aus einer Klasse von Funktionen) zu.

Wir interessieren uns hier im Besonderen für den folgenden Effizienzbegriff (analog zur Algorithmik).

Definition (Effizienz und totale Ineffizienz) *Eine Suchheuristik A heißt effizient auf Funktionen $f_n : \{0, 1\}^n \rightarrow \mathbb{Z}$, $n \in \mathbb{N}$, falls es ein Polynom $\text{poly}(n)$ gibt, so dass $E[T_{f_n, A}] \leq \text{poly}(n)$. Eine Suchheuristik A heißt total ineffizient auf Funktionen $f_n : \{0, 1\}^n \rightarrow \mathbb{Z}$, $n \in \mathbb{N}$, falls es eine Konstante $\varepsilon > 0$ gibt, so dass $\Pr[T_{f_n, A} \leq 2^{n^\varepsilon}] \leq 2^{-n^{\Omega(1)}}$.*

Falls eine Suchheuristik A total ineffizient ist, so gilt dies auch für die – häufig ebenfalls betrachtete – Suchheuristik, welche (unabhängige) (parallele) Multistarts von A durchführt.

Erste Laufzeitanalysen betrachteten (und verglichen) einfache evolutionäre Algorithmen auf einfachen (Klassen von) Funktionen. Die sicherlich prominenteste Funktion ist ONEMAX : $\{0, 1\}^n \rightarrow \mathbb{Z}$ mit $x = x_1 \cdots x_n \in \{0, 1\}^n$ und

$$\text{ONEMAX}(x) := |x| = \sum_{i=1}^n x_i.$$

Mühlenbein (1992) konnte eine Laufzeit von $\mathcal{O}(n \ln n)$ für den (1+1)-EA zur Optimierung von ONEMAX beweisen. Mit Estimation-of-Distribution- und Ant-Colony-Optimization-Algorithmen untersuchten Droste (2006) sowie Neumann und Witt (2006) das Verhalten neuerer Varianten randomisierter Suchheuristiken zur Optimierung von ONEMAX.

Droste et al. (2002) haben das Ergebnis für den (1+1)-EA zur Optimierung von ONEMAX auf die Klasse aller linearen Funktionen mit positiven Gewichten erweitert und eine Laufzeit von $\Theta(n \ln n)$ bewiesen. Mit der Klasse der monotonen Polynome untersuchten Wegener und Witt (2005) das Verhalten des (1+1)-EA zur Optimierung allgemeinerer Klassen von Funktionen.

Neben der Betrachtung einfacher evolutionärer Algorithmen und typischer (Klassen von) Funktionen wurden auch atypische Funktionen untersucht, um das Verhalten einfacher evolutionärer Algorithmen in bestimmten Situationen weiter zu illustrieren oder die – auch wechselseitigen – Effekte einer oder mehrerer Komponenten solcher Heuristiken. Die Effekte des (1+1)-EA auf kleinen Plateaus analysierten Jansen und Wegener (2001a), wohingegen Garnier et al. (1999) diese auf großen Plateaus erklärten. Plateaus sind (benachbarte) Elemente des Suchbereichs mit gleichem Funktionswert. Jansen et al. (2005) und Witt (2006) betrachteten nicht nur die Laufzeiten bei unterschiedlichen Nachkommen- und Elternpopulationsgrößen für ONEMAX sondern sie gaben auch Funktionen mit der Eigenschaft an, dass die wesentliche Veränderung der Populationsgröße von einer total ineffizienten zu einer effizienten Suchheuristik führt. Solche schwachen Hierarchieresultate für die Populationsgröße zeigten bereits Jansen und Wegener (2001b) und in einer erweiterten Fassung auch Witt (2003) für einfache populationsbasierte evolutionäre Algorithmen. Jansen und Wegener (2005) haben eine wesentliche Steigerung der Effizienz durch den Einsatz einer Kreuzung neben einer Mutation bei Verwendung einer großen Population gegenüber der alleinigen Verwendung einer Mutation nachgewiesen. Storch und Wegener (2004) haben dieses Resultat für die kleinstmögliche (eine Kreuzung erlaubende) Elternpopulationsgröße von zwei erweitert.

Wir analysieren und vergleichen in Kapitel 2 die Effekte der Wahl der Plus- und der Komma-Selektion bei unterschiedlichen Nachkommenpopulationsgrößen sowohl auf typischen als auch auf atypischen Funktionen. Anders als die Plus-Selektion führt die Komma-Selektion bei kleiner Nachkommenpopulation schon für typische Zielfunktionen zur total ineffizienten Optimierung (vergleiche Theorem 2.A.1 und Theorem 2.C.1). Bei großen Nachkommenpopulationen dagegen wird typischerweise kein Unterschied zwischen dem Verhalten der Plus- und der Komma-Selektion festgestellt (vergleiche Lemma 2.B.2 und Theorem 2.C.1). Für Nachkommenpopulationen, die weder klein noch groß sind, wird exemplarisch für eine atypische Funktion gezeigt, dass die Komma-Selektion wesentlich effizienter als die Plus-Selektion sein kann (vergleiche Theorem 2.E.1).

Wir untersuchen in Kapitel 3 die Effekte der Wahl der Elternpopulationsgröße. Für typische Funktionen und einfache randomisierte Suchheuristiken vergleichen wir dabei insbesondere die Auswirkungen der Diversität auf die Effizienz der Optimierung (vergleiche Theorem 3.A.1 und Theorem 3.A.7). Weiterhin wird ein starkes Hierarchieresultat für die Elternpopulationsgröße gezeigt. Bereits die Vergrößerung der Population um eins kann von einer total ineffizienten randomisierten Suchheuristik zu einer effizienten führen (vergleiche Theorem 3.C.1).

In Kapitel 4 betrachten wir detailliert die Effekte von Transformationen der Zielfunktionen sowohl für die unterschiedlichsten (Komponenten und Klassen von) randomisierten Suchheuristiken als auch für die Black-Box-Komplexität (vergleiche Abschnitt 4.B). Diesbezüglich wird eine Klassifizierung aller randomisierter Suchheuristiken (und Transformationen) (vergleiche Abschnitt 4.A) zu den rangbasierten, nicht funktionswertbasierten und nicht rangbasierten Algorithmen vorgenommen sowie für einfache Suchheuristiken konkretisiert (vergleiche Abschnitt 4.C, Abschnitt 4.D und Abschnitt 4.E).

Neuere Laufzeitanalysen betrachteten (und verglichen) einfache evolutionäre Algorithmen auf typischen kombinatorischen Optimierungsproblemen. Neumann und Wegener (2004) zeigen etwa, dass der (1+1)-EA effizient die Aufgabe löst, einen minimalen Spannbaum in einem gewichteten Graphen zu berechnen. Witt (2005) analysiert, inwiefern der (1+1)-EA die Aufgabe löst, eine Partitionierung von Gewichten in zwei gleich große Mengen zu approximieren.

Wir analysieren und vergleichen in Kapitel 5 einfache randomisierte Suchheuristiken zur Approximierung und Optimierung des Cliquenproblems. Dies geschieht nicht nur für beliebige Graphen erschöpfend (vergleiche Theorem 5.A.2 und Theorem 5.A.5), sondern auch für spezielle Klassen von Graphen. Den semizufälligen Graphen kommt eine besondere Bedeutung zu. Hier gelingt es schließlich in einer sehr allgemeinen Form, die wesentliche Steigerung der Effizienz einer randomisierten Suchheuristik durch den Einsatz einer (großen) Population nachzuweisen (vergleiche Theorem 5.C.5 und Theorem 5.C.6). Für planare und zufällige planare Graphen wird abschließend durch die Betrachtung der Black-Box-Komplexität und der durchschnittlichen Black-Box-Komplexität sogar die (asymptotische) Optimalität einer speziellen einfachen randomisierten Suchheuristik zur Berechnung einer größten Clique bewiesen (vergleiche Theoreme 5.E.1 und 5.E.2 für die Black-Box-Komplexität sowie Theoreme 5.D.9 und 5.D.10 für die Suchheuristik).

Nicht nur im Zuge dieser Untersuchungen sind eine Reihe von Methoden zur Analyse randomisierter Suchheuristiken entwickelt worden. Zu den vermutlich bekanntesten Techniken gehört die *Methode der f -basierten Partition* nach Wegener (2002). Diese kann helfen, die erwartete Laufzeit des (1+1)-EA zur Optimierung von Funktionen $f : \{0, 1\}^n \rightarrow \mathbb{Z}$ nach oben zu beschränken. Für $A, B \subseteq \{0, 1\}^n$ und $A, B \neq \emptyset$ gilt die Relation $A <_f B$ genau dann, wenn $f(x) < f(y)$ ist für alle $x \in A$ und $y \in B$. Wir nennen A_0, \dots, A_m eine *f -basierte Partition* genau dann, wenn $A_0 \cup \dots \cup A_m = \{0, 1\}^n$, $A_0 <_f \dots <_f A_m$ gilt und weiterhin A_m nur Optima von f enthält, also für alle $x \in A_m$ gilt $x \in \arg \max\{f(y) \mid y \in \{0, 1\}^n\}$. Sei $p(x)$, $x \in A_i$ für $i < m$, die Wahrscheinlichkeit, dass eine Mutation von x ein Element $y \in A_{i+1} \cup \dots \cup A_m$ erzeugt. Dann stellt $p(A_i) := \min\{p(x) \mid x \in A_i\}$, $i < m$, eine untere Schranke für die Wahrschein-

lichkeit dar, A_i zu verlassen. Da der (1+1)-EA keine Verschlechterungen in der Population akzeptiert, muss jedes A_i , $0 \leq i < m$, höchstens einmal verlassen werden. Für A_i wird dazu eine erwartete Anzahl von höchstens $1/p(A_i)$ Funktionsauswertungen benötigt. Zusammen mit der ersten Funktionsauswertung ergibt sich das folgende Lemma.

Lemma 1.B.1 *Sei A_0, \dots, A_m eine f -basierte Partition für $f : \{0, 1\}^n \rightarrow \mathbb{Z}$. Der (1+1)-EA (mit beliebiger Verteilung von x_1) erzeugt ein Optimum von f in einer erwarteten Laufzeit von höchstens*

$$1 + \frac{1}{p(A_0)} + \dots + \frac{1}{p(A_{m-1})}.$$

Betrachten wir exemplarisch die erwartete Laufzeit des (1+1)-EA zur Optimierung von ONEMAX und dazu die f -basierte Partition $A_i := \{x \mid |x| = i\}$ für $0 \leq i \leq n$. Für jedes $x \in A_i$ mit $0 \leq i < n$ existieren $n - i$ spezielle ein Bit flippende Mutationen, die ein Element aus $A_{i+1} \cup \dots \cup A_n$ erzeugen. Es gilt $p(A_i) \geq (n - i)/(en)$ und somit resultiert eine erwartete Laufzeit von höchstens $1 + \frac{en}{n} + \dots + \frac{en}{1} \leq 1 + en \cdot (\ln n + 1) = \mathcal{O}(n \ln n)$, da für die n -te harmonische Zahl gilt $\frac{1}{1} + \dots + \frac{1}{n} \leq \ln n + 1$.

Diese Methode der f -basierten Partition werden wir in den Kapiteln 2 und 3 für Nachkommen- und Elternpopulationen erweitern (vergleiche Lemmata 2.D.1 und 2.D.2 für die Nachkommenpopulationen sowie Lemmata 3.B.1 und 3.B.2 für die Elternpopulationen).

Eine Reihe weiterer bekannter Beweistechniken, etwa die Methode des typischen Laufs oder die Methode der Wiederholung unabhängiger Phasen, werden wir während unserer Analysen anwenden und neue Beweistechniken entwickeln.

1.C Veröffentlichungen und Eigenanteil

Diese Dissertation basiert im Wesentlichen auf den folgenden Veröffentlichungen:

- Jägersküpper, Jens und Storch, Tobias (2006): When the plus strategy outperforms the comma strategy—and when not. *Technischer Bericht CI-219/06*. Universität Dortmund. (zur Veröffentlichung in den *Proceedings of the First Symposium on Foundations of Computational Intelligence – FOCI 2007* angenommen)

Kapitel 2

Aus der Zusammenarbeit mit Jens Jägersküpper ging auch die folgende, thematisch verwandte Veröffentlichung hervor:

Jägersküpper, Jens und Storch, Tobias (2006): How comma selection helps with the escape from local optima. In: Runarsson et al., Hg., *Proceedings of the International Conference on Parallel Problem Solving From Nature IX – PPSN 2006*, Bd. 4193 von *Lecture Notes in Computer Science*, S. 52–61. Springer.

An diesen gemeinsamen Arbeiten war der Verfasser zur Hälfte beteiligt.

- Storch, Tobias (2004): On the choice of the population size. In: Deb et al., Hg., *Proceedings of the Genetic and Evolutionary Computation Conference – GECCO 2004*, Bd. 3102 von *Lecture Notes in Computer Science*, S. 748–760. Springer. Ausgezeichnet mit einem *Best Paper Award*.

Kapitel 3

- Storch, Tobias (2006): On the impact of objective function transformations on evolutionary and black-box algorithms. *Genetic Programming and Evolvable Machines*, 7(2):171–193. Erweiterte Fassung von

Kapitel 4

Storch, Tobias (2005): On the impact of objective function transformations on evolutionary and black-box algorithms. In: Beyer et al., Hg., *Proceedings of the Genetic and Evolutionary Computation Conference – GECCO 2005*, Bd. 1, S. 833–840. ACM Press. Ausgezeichnet mit einem *Best Paper Award*.

Kapitel 4

- Storch, Tobias (2006): Finding large cliques in sparse semi-random graphs by simple randomized search heuristics. *Technischer Bericht CI-211/06*. Universität Dortmund.

Kapitel 5

- Storch, Tobias (2006): How randomized search heuristics find maximum cliques in planar graphs. In: Keijzer et al., Hg., *Proceedings of the Genetic and Evolutionary Computation Conference – GECCO 2006*, Bd. 1, S. 567–574. ACM Press.

Kapitel 5

2 NACHKOMMENPOPULATIONEN IN EVOLUTIONÄREN ALGORITHMEN

Seit der Einführung der Plus- und der Komma-Selektion in evolutionären Algorithmen gibt es viele Debatten darüber, welche dieser Selektionen (unter welchen Umständen) zu favorisieren ist (vergleiche Schwefel, 1995). Hierbei wird häufig behauptet, wie in Jansen et al. (2005, Seite 415), dass *der Unterschied zwischen einem elitären $(1+\lambda)$ -EA und einem nicht-elitären $(1,\lambda)$ -EA im $\{0, 1\}^n$ weniger wichtig ist*¹. Wir werden für die im Folgenden definierten, einfachen $(1+\lambda)$ -EA und $(1,\lambda)$ -EA nachweisen, für welche Nachkommenpopulationsgrößen λ dies zutrifft – und für welche nicht.

$(1+\lambda)$ -EA

1. Setze $t := 1$ und wähle $x_t \in \{0, 1\}^n$ zufällig gleichverteilt.
2. Setze $t := t + 1$ und $y_{t,1} := \text{mutate}_{1/n}(x_{t-1}), \dots, y_{t,\lambda} := \text{mutate}_{1/n}(x_{t-1})$.
3. Wähle $y_t \in \arg \max\{f(y_{t,1}), \dots, f(y_{t,\lambda})\}$ zufällig gleichverteilt.
 - $(1+\lambda)$ -EA: Falls $f(y_t) \geq f(x_{t-1})$, dann setze $x_t := y_t$ und sonst $x_t := x_{t-1}$.
 - $(1,\lambda)$ -EA: Setze $x_t := y_t$.
4. Gehe zu 2.

Wir betrachten die Maximierung und bemerken, dass der $(1+\lambda)$ -EA mit $\lambda = 1$ dem $(1+1)$ -EA aus Abschnitt 1.A entspricht. Für $\lambda = \infty$ sind der $(1+\lambda)$ -EA und der $(1,\lambda)$ -EA die gleichen Optimierer, da Anweisung 3 in endlich vielen Schritten nicht durchgeführt wird.

Für die weiteren Untersuchungen dieses Kapitels ist es unerheblich, welches der besten Individuen der Nachkommenpopulation beim $(1,\lambda)$ -EA und der Nachkommenpopulation zusammen mit dem Elter beim $(1+\lambda)$ -EA den nächsten Elter definiert. Für keine Funktion f erzeugt der $(1+\lambda)$ -EA ein x_t , $t \geq 2$, mit $f(x_{t-1}) > f(x_t)$. Der $(1,\lambda)$ -EA hingegen generiert ein solches Element in jedem Schritt mit positiver Wahrscheinlichkeit, solange $x_{t-1} \notin \arg \min\{f(x) \mid x \in \{0, 1\}^n\}$ ist. Dies kann die schnelle Stagnation in lokalen Optima verhindern, jedoch auch zu einer schleppenden Bewegung zu globalen Optima führen.

Diese zwei evolutionären Algorithmen erlauben eine Analyse der Effekte der Wahl von λ sowie beider Selektionsverfahren und vermeiden zugleich Einflüsse weiterer Komponenten solcher Heuristiken. Jansen et al. (2005) haben den $(1+\lambda)$ -EA ausführlich analysiert. Wir gehen an geeigneten Stellen näher darauf ein.

¹... the difference between an elitist $(1+\lambda)$ EA and a non-elitist $(1,\lambda)$ EA is less important in $\{0, 1\}^n$.

2.A Kleine Nachkommenpopulationen

Wir beginnen unsere Betrachtungen mit einem Vergleich der Plus- und der Komma-Selektion bei kleinstmöglicher Nachkommenpopulation. Einerseits optimiert der (1+1)-EA in einer erwarteten Laufzeit von $\mathcal{O}(n^n)$ jede Funktion und es sind auch Funktionen bekannt, zu deren Optimierung der (1+1)-EA eine erwartete Laufzeit von $\Theta(n^n)$ benötigt. Andererseits optimiert der (1+1)-EA in einer erwarteten Laufzeit von $\mathcal{O}(n \ln n)$ eine beliebige lineare Funktion, und besitzt diese zusätzlich ein eindeutiges Optimum, so braucht er auch eine erwartete Laufzeit von $\Theta(n \ln n)$ (Droste et al., 2002). Der (1,1)-EA hingegen optimiert zwar jede Funktion in einer erwarteten Laufzeit von $\mathcal{O}(2^n)$, jedoch benötigt er auch eine erwartete Laufzeit von $\Theta(2^n)$ für jede Funktion mit eindeutigem Optimum (Garnier et al., 1999). Je näher sich der Suchpunkt x_t des (1,1)-EA am eindeutigen Optimum befindet, desto größer ist die Wahrscheinlichkeit, dass der Suchpunkt x_{t+1} weiter als x_t vom Optimum entfernt ist. Diese Drift weg vom Optimum ist so stark, dass der (1,1)-EA eine solche Funktion nicht effizient optimiert. Wir weisen mit folgendem Theorem sogar für den (1, λ)-EA mit $\lambda \leq \ln(n)/14$ eine Drift weg vom Optimum nach, welche stark genug ist, eine effiziente Optimierung einer beliebigen Funktion mit eindeutigem Optimum zu verhindern.

Theorem 2.A.1 *Sei $f : \{0, 1\}^n \rightarrow \mathbb{Z}$ eine beliebige Funktion mit eindeutigem Optimum und sei $\lambda \leq \varepsilon(n) \ln(n)/7$ mit $\varepsilon(n) \in [7/\ln n, 1/2]$. Mit Wahrscheinlichkeit $1 - 2^{-\Omega(n^{1-\varepsilon(n)})}$ benötigt der (1, λ)-EA eine Laufzeit von mehr als $2^{n^{1-\varepsilon(n)}}$ zur Optimierung von f .*

Mit $\varepsilon(n) = 7/\ln n$ erhalten wir den (1,1)-EA und eine untere Schranke seiner (erwarteten) Laufzeit zur Optimierung einer beliebigen Funktion mit eindeutigem Optimum, die jene von Garnier et al. (1999) mindestens nachvollzieht. Mit $\varepsilon(n) = 1/2$ wird des Weiteren die totale Ineffizienz des (1, $\lfloor \ln(n)/14 \rfloor$)-EA zur Optimierung einer solchen Funktion nachgewiesen. In den nächsten Abschnitten werden wir feststellen, dass bereits für unwesentlich größere Nachkommenpopulationen als $\lfloor \ln(n)/14 \rfloor$ ein wesentlich anderes Verhalten des (1, λ)-EA erwartet werden darf.

Wir führen ein Ergebnis über Markoffprozesse ein, bevor wir das gewünschte Resultat beweisen. Ein *Markoffprozess* (engl. *markov process*) \mathcal{M} mit $m < \infty$ Zuständen $0, \dots, m-1$ ist durch eine stochastische $m \times m$ -Matrix P der *Übergangswahrscheinlichkeiten* (der Eintrag $p_{i,j}$, $0 \leq i, j \leq m-1$, entspricht der Wahrscheinlichkeit, im t -ten Schritt, $t \geq 1$, von Zustand i in Zustand j zu wechseln) und einen stochastischen m -(Zeilen-)Vektor p der *Initialisierungswahrscheinlichkeiten* (der Eintrag p_i , $0 \leq i \leq m-1$, entspricht der Wahrscheinlichkeit nach dem 1-ten Schritt in Zustand i zu sein) beschrieben. Der i -te Eintrag des m -Vektors pP^{t-1} gibt die Wahrscheinlichkeit an, dass der Markoffprozess \mathcal{M} sich nach dem t -ten Schritt in Zustand i befindet. (Eine ausführliche Betrachtung von Markoffprozessen findet sich in Mitzenmacher und Upfal (2005) sowie in Motwani und Raghavan (1995).) Das folgende Resultat ergibt sich analog zu den Betrachtungen von Giel und Wegener (2004), welche wesentlich auf Ergebnissen von He und Yao (2001) beruhen.

Lemma 2.A.2 *Sei \mathcal{M} ein Markoffprozess mit m Zuständen und $\ell \in \{0, \dots, m-1\}$. Des Weiteren seien $\alpha(\ell), \beta(\ell), \gamma(\ell) > 0$ und es gelte:*

1. $\sum_{j=0}^{m-1} p_{i,j} \cdot e^{-\alpha(\ell) \cdot (j-i)} \leq 1 - 1/\beta(\ell)$ für alle i mit $1 \leq i < \ell$
2. $\sum_{j=0}^{m-1} p_{i,j} \cdot e^{-\alpha(\ell) \cdot (j-\ell)} \leq 1 + \gamma(\ell)$ für alle i mit $\ell \leq i \leq m-1$

Der 0-te Eintrag des m -Vektors pP^{t-1} beträgt höchstens $t \cdot e^{-\alpha(\ell) \cdot \ell} \beta(\ell)(1 + \gamma(\ell)) + \sum_{j=0}^{\ell-1} p_j$.

Beweis (Theorem 2.A.1). Im Folgenden sei n groß genug, was für den Beweis der Aussage angenommen werden darf. Sei $x^* \in \{0,1\}^n$ das eindeutige Optimum von f . Die Anzahl durchgeführter Funktionsauswertungen des $(1,\lambda)$ -EA bis zum einschließlich $t(n)$ -ten Schritt mit $t(n) \geq 1$ beträgt $1 + \lambda \cdot (t(n) - 1)$. Der $(1,\lambda)$ -EA benötigt eine Laufzeit von mehr als $1 + \lambda \cdot (t(n) - 1) \geq t(n)$ zur Optimierung von f , wenn er bis zum $t(n)$ -ten Schritt nicht x^* gefunden hat. Wir nehmen an, dass nachdem der $(1,\lambda)$ -EA erstmals das Optimum x^* erzeugt hat, er nur noch x^* generiert. Und wir sind am Ereignis $x_{\lfloor 2^{n^{1-\varepsilon(n)}} \rfloor} = x^*$ interessiert. Denn ist die Wahrscheinlichkeit seines Eintretens $2^{-\Omega(n^{1-\varepsilon(n)})}$, erhalten wir das gewünschte Resultat für die Laufzeit des $(1,\lambda)$ -EA zur Optimierung von f .

Wir beschreiben einen Markoffprozess \mathcal{M} mit $n+1$ Zuständen, der nach $t(n)$ Schritten mit mindestens gleicher Wahrscheinlichkeit einen Zustand i, \dots, n erreicht wie der $(1,\lambda)$ -EA einen Suchpunkt $x_{t(n)}$ mit Hammingabstand $H[x_{t(n)}, x^*] \geq i$ zum Optimum x^* . Dies wird für alle i mit $0 \leq i \leq n$ gelten. Der $(1,\lambda)$ -EA generiert also höchstens mit gleicher Wahrscheinlichkeit das Optimum, wie \mathcal{M} den Zustand 0 erreicht. Setzen wir $p_i := \binom{n}{i}/2^n$, hat \mathcal{M} die gewünschte Eigenschaft für $t(n) = 1$ – sogar mit Gleichheit. Befindet sich \mathcal{M} nach $t(n)$ Schritten im Zustand i , gilt für das Element $x_{t(n)}$ des $(1,\lambda)$ -EA mit zumindest gleicher Wahrscheinlichkeit $H[x_{t(n)}, x^*] \geq i$. Nehmen wir an, die Wahrscheinlichkeit beträgt höchstens $p_{i,\leq j}$, dass ein solches $x_{t(n)}$ als nächsten Elter $x_{t(n)+1}$ mit $H[x_{t(n)+1}, x^*] \leq j$ erzeugt. Damit \mathcal{M} die gewünschte Eigenschaft auch im nachfolgenden Schritt erfüllt, genügt es, die Ungleichung $p_{i,0} + \dots + p_{i,j} \geq p_{i,\leq j}$ sicherzustellen. Setzen wir $p_{i,j}$ für $j < i$ zumindest auf das Maximum der Wahrscheinlichkeiten, dass ein solches Individuum $x_{t(n)}$ ein $x_{t(n)+1}$ mit $H[x_{t(n)+1}, x^*] = j$ generiert, gilt die Ungleichung für $j < i$. Setzen wir $p_{0,0} := 1$, gilt die Ungleichung für $i = 0$. Setzen wir $p_{i,i+1}$ für $i \geq 1$ höchstens auf das Minimum der Wahrscheinlichkeiten, dass ein solcher Suchpunkt $x_{t(n)}$ ein $x_{t(n)+1}$ mit $H[x_{t(n)+1}, x^*] \geq i+1$ erzeugt und $p_{i,j} := 0$ für $j \geq i+2$ sowie $p_{i,i} := 1 - p_{i,i+1} - \sum_{j=0}^{i-1} p_{i,j}$, gilt die Ungleichung für $j \geq i$ und $i \geq 1$.

Wir wollen Lemma 2.A.2 für \mathcal{M} mit $\ell = \lfloor n^{1-\varepsilon(n)} \rfloor$, $\alpha(\ell) = 6/5$, $\beta(\ell) = 32\ell$ und $\gamma(\ell) = 1$ anwenden. Dazu muss das Vorliegen folgender zwei Voraussetzungen nachgewiesen werden.

1. $\sum_{j=0}^n p_{i,j} e^{-(6/5) \cdot (j-i)} \leq 1 - 1/(32 \lfloor n^{1-\varepsilon(n)} \rfloor) \leq 1 - n^{-(1-\varepsilon(n))}/32$ für alle i mit $1 \leq i < \lfloor n^{1-\varepsilon(n)} \rfloor$.

Wir betrachten zunächst $j < i$. Es gilt

$$p_{i,j} \leq \max\{\lambda \cdot \binom{i+k}{i+k-j} / n^{i+k-j} \mid 0 \leq k \leq n-i\},$$

da es zur Reduzierung des Hammingabstands zum Optimum auf j für jedes Element x mit $H[x, x^*] = i+k$, $0 \leq k \leq n-i$, notwendig ist, dass zumindest bei einem der λ Nachkommen $i+k-j$ von $i+k$ speziellen Bits flippen. Weiter ist

$$\max\{\lambda \cdot \binom{i+k}{i+k-j} / n^{i+k-j} \mid 0 \leq k \leq n-i\} \leq \lambda \cdot \binom{i}{i-j} / n^{i-j},$$

da $\binom{i+k}{i+k-j} = \binom{i}{i-j} \cdot \frac{(i+k)\cdots(i+1)}{(j+k)\cdots(j+1)} \leq \binom{i}{i-j} \cdot n^k$ gilt. Mit $\binom{i}{i-j} \leq i^{i-j}$ ist

$$\lambda \cdot \binom{i}{i-j} / n^{i-j} \leq \lambda \cdot i^{i-j} / n^{i-j} \leq \lambda \cdot n^{(1-\varepsilon(n)) \cdot (i-j)} / n^{i-j} = \lambda \cdot n^{-\varepsilon(n)(i-j)}.$$

Wir betrachten nun $j = i + 1$ für $i \geq 1$. Es gilt

$$p_{i,i+1} \geq \min\left\{\binom{n-i}{1} \cdot (1/n) \cdot (1-1/n)^{n-1}, (1-1/n)^n\right\}^\lambda,$$

da zur Erhöhung des Hammingabstands zum Optimum auf wenigstens $i + 1$ für jeden Suchpunkt x

- mit $H[x, x^*] = i$ es ausreicht, dass jedes der λ Nachkommen einen Hammingabstand von $i + 1$ zum Optimum hat; dazu muss bei jedem Nachkommen genau eines von $n - i$ vielen Bits flippen, während
- mit $H[x, x^*] > i$ es ausreicht, dass jedes der λ Nachkommen ein Duplikat von x darstellt.

Weiter ist

$$\min\left\{\binom{n-i}{1} \cdot (1/n) \cdot (1-1/n)^{n-1}, (1-1/n)^n\right\}^\lambda \geq (6/17)^\lambda,$$

da sowohl $(1-1/n)^n \geq 6/17$ als auch $((n-i)/n) \cdot (1-1/n)^{n-1} \geq (n-n^{1-7/\ln n})/(en) \geq 6/17$ gilt. Es ist

$$(6/17)^\lambda \geq (6/17)^{\varepsilon(n)\ln(n)/7} \geq n^{-\varepsilon(n)/6},$$

wobei die letzte Ungleichung $\ln(6/17) \cdot 1/7 \geq -1/6$ ausnutzt.

$\sum_{j=0}^{i-1} \lambda n^{-\varepsilon(n)(i-j)} e^{(6/5) \cdot (i-j)} + (1 - n^{-\varepsilon(n)/6} - \sum_{j=0}^{i-1} \lambda n^{-\varepsilon(n)(i-j)} e^{(6/5) \cdot 0} + n^{-\varepsilon(n)/6} e^{(6/5) \cdot (-1)}) \leq 1 - n^{-(1-\varepsilon(n))/32}$ bleibt für alle i mit $1 \leq i < \lfloor n^{1-\varepsilon(n)} \rfloor$ zu zeigen. Mit einer Indextransformation und der Konvergenzeigenschaft unendlicher geometrischer Reihen gilt

$$\sum_{j=1}^i \lambda n^{-\varepsilon(n)j} e^{6j/5} \leq \lambda \sum_{j=1}^{\infty} (n^{-\varepsilon(n)} e^{6/5})^j = \lambda / (n^{\varepsilon(n)} e^{-6/5} - 1).$$

Weiter ist mit $\lambda \leq \varepsilon(n)\ln(n)/7$

$$\lambda / (n^{\varepsilon(n)} e^{-6/5} - 1) \leq \varepsilon(n)\ln(n) / (7n^{\varepsilon(n)} e^{-6/5} - 7) \leq 2n^{-\varepsilon(n)/6} / 3,$$

wobei die letzte Ungleichung ausnutzt, dass für $\varepsilon(n) \geq 7/\ln n$ und mit $e^{5\varepsilon(n)\ln(n)/6} \geq 1 + 5\varepsilon(n)\ln(n)/6$ gilt

$$\begin{aligned} 0 &\leq \varepsilon(n)\ln(n) \cdot (14/3 \cdot 5/6 \cdot e^{-6/5} - 1) + 14/3(e^{-6/5} - e^{-7/6}) \\ &\leq 14/3 \cdot e^{-6/5} \cdot n^{5\varepsilon(n)/6} - 14/3 \cdot n^{-\varepsilon(n)/6} - \varepsilon(n)\ln(n) \\ &= (2n^{-\varepsilon(n)/6} / 3) \cdot (7n^{\varepsilon(n)} e^{-6/5} - 7) - \varepsilon(n)\ln(n). \end{aligned}$$

Da des Weiteren gilt

$$1 - n^{-\varepsilon(n)/6} - \sum_{j=1}^{i-1} \lambda n^{-\varepsilon(n)j} \leq 1 - n^{-\varepsilon(n)/6},$$

ist die obige Bedingung mit $2n^{-\varepsilon(n)/6}/3 + (1 - n^{-\varepsilon(n)/6}) + n^{-\varepsilon(n)/6}e^{-6/5} \leq 1 - n^{-\varepsilon(n)/6}/32 \leq 1 - n^{-(1-\varepsilon(n))/6}/32$ erfüllt, wobei die letzte Ungleichung $\varepsilon(n) \leq 1/2$ ausnutzt.

2. $\sum_{j=0}^n p_{i,j} e^{-(6/5) \cdot (j - \lfloor n^{1-\varepsilon(n)} \rfloor)} \leq 2$ für alle i mit $\lfloor n^{1-\varepsilon(n)} \rfloor \leq i \leq n$.

Wie im Nachweis des Vorliegens der ersten Voraussetzung gilt $p_{i,j} \leq \lambda n^{-\varepsilon(n)(\lfloor n^{1-\varepsilon(n)} \rfloor - j)}$ für $j < i$. Ebenfalls wie zuvor gezeigt, ist für $\lfloor n^{1-\varepsilon(n)} \rfloor \leq i \leq n$ die obige Bedingung nun mit $\sum_{j=0}^{\lfloor n^{1-\varepsilon(n)} \rfloor - 1} \lambda n^{-\varepsilon(n)(\lfloor n^{1-\varepsilon(n)} \rfloor - j)} e^{(6/5) \cdot (\lfloor n^{1-\varepsilon(n)} \rfloor - j)} + (1 - \sum_{j=0}^{\lfloor n^{1-\varepsilon(n)} \rfloor - 1} \lambda n^{-\varepsilon(n)(\lfloor n^{1-\varepsilon(n)} \rfloor - j)}) \cdot e^{(6/5) \cdot 0} \leq 2n^{-\varepsilon(n)/6}/3 + 1 \leq 2$ erfüllt.

Um Lemma 2.A.2 anzuwenden, bleibt $\sum_{j=0}^{\ell-1} p_j$ nach oben zu beschränken. Da $\ell \leq \lfloor n^{1-\varepsilon(n)} \rfloor \leq \lfloor n/e^7 \rfloor$ mit $\varepsilon(n) \geq 7/\ln n$ ist, gilt

$$\sum_{j=0}^{\ell-1} \binom{n}{j} / 2^n \leq \sum_{j=0}^{\lfloor n/e^7 \rfloor - 1} \binom{n}{j} / 2^n \leq n \binom{n}{\lfloor n/e^7 \rfloor} / 2^n \leq n \left(\frac{en}{n/e^7} \right)^{n/e^7} / 2^n = e^{\ln n + 8n/e^7 - n \ln 2} \leq e^{-n/3}.$$

Wie oben dargelegt, liefert eine Anwendung des Lemmas 2.A.2 für \mathcal{M} mit $t = \lfloor 2^{n^{1-\varepsilon(n)}} \rfloor$ eine Wahrscheinlichkeit von höchstens $\lfloor 2^{n^{1-\varepsilon(n)}} \rfloor \cdot e^{-(6/5) \cdot \lfloor n^{1-\varepsilon(n)} \rfloor} \cdot 32 \lfloor n^{1-\varepsilon(n)} \rfloor \cdot (1+1) + e^{-n/3} = 2^{-\Omega(n^{1-\varepsilon(n)})}$, in $\lfloor 2^{n^{1-\varepsilon(n)}} \rfloor$ Schritten den Zustand 0 zu erreichen. \square

2.B Große Nachkommenpopulationen

Jansen et al. (2005, Seite 415) behaupten, dass bei einer genügend großen Nachkommenpopulation sich die Art und Weise nicht wesentlich unterscheidet, wie der $(1+\lambda)$ -EA und der $(1,\lambda)$ -EA den $\{0,1\}^n$ durchsuchen¹. In dieser Situation nämlich enthält die Nachkommenpopulation üblicherweise mindestens eine exakte Kopie des Elters². Wir präzisieren dies im Folgenden.

Im Weiteren sei $\star \in \{, +, ,, \{ \}$ und $\bar{\star}$ das andere der beiden Symbole. Für $t(n) \geq 1$ sei

$$s_{t(n)} := x_1, y_{2,1}, \dots, y_{2,\lambda}, y_2, \quad x_2, y_{3,1}, \dots, y_{3,\lambda}, y_3, \quad \dots, \quad x_{t(n)}, y_{t(n)+1,1}, \dots, y_{t(n)+1,\lambda}, y_{t(n)+1}$$

eine beliebige Folge von $(\lambda + 2) \cdot t(n)$ Suchpunkten des $\{0,1\}^n$ und sei $f : \{0,1\}^n \rightarrow \mathbb{Z}$ eine beliebige Funktion. Der $(1\star\lambda)$ -EA beobachtet $s_{t(n)}$ bei Optimierung von f , wenn mit positiver Wahrscheinlichkeit sowohl $x_1, \dots, x_{t(n)}$ als seine ersten $t(n)$ Eltern auftreten als auch dann y_t für $t = 2, \dots, t(n) + 1$ der ausgewählte Nachkomme aus den jeweils λ Nachkommen $y_{t,1}, \dots, y_{t,\lambda}$ von x_{t-1} ist. Wir betrachten eine Folge $s_{t(n)}$, die beim $(1\star\lambda)$ -EA beobachtet wird. Der $(1\star\lambda)$ -EA und der $(1\bar{\star}\lambda)$ -EA unterscheiden sich lediglich in Anweisung 3.

¹With an offspring population size λ of any appreciable size, the $(1+\lambda)$ EA and the $(1,\lambda)$ EA will not differ significantly in the way they search $\{0,1\}^n$.

²... the offspring population will almost surely contain at least one exact copy of the parent.

Gilt $f(y_t) \geq f(x_{t-1})$ für alle t mit $2 \leq t \leq t(n)$, ist die Bedingung in Anweisung 3 immer erfüllt. Da der $(1\star\lambda)$ -EA und der $(1\bar{\star}\lambda)$ -EA mit gleicher Wahrscheinlichkeit sowohl ein identisches x_1 betrachten als auch bei gleichem Elter x_{t-1} die gleichen Nachfolger $y_{t,1}, \dots, y_{t,\lambda}$ erzeugen und ebenso den ausgewählten Nachkommen y_t identisch bestimmen, beobachten der $(1\star\lambda)$ -EA und der $(1\bar{\star}\lambda)$ -EA während der Optimierung von f die Folge $s_{t(n)}$ mit gleicher Wahrscheinlichkeit. Die Menge dieser Folgen sei mit $S_{\dagger,f,t(n)}$ bezeichnet.

Gilt $f(y_t) < f(x_{t-1})$ für (mindestens) ein t mit $2 \leq t \leq t(n)$, ist die Bedingung in Anweisung 3 wenigstens in Schritt t nicht erfüllt. Da mit $f(y_t) < f(x_{t-1})$ auch $y_t \neq x_{t-1}$ ist, wählen der $(1\star\lambda)$ -EA und der $(1\bar{\star}\lambda)$ -EA sicher unterschiedliche Elemente als nächsten Elter x_t . Die Folge $s_{t(n)}$ wird nicht beim $(1\bar{\star}\lambda)$ -EA während der Optimierung von f beobachtet. Die Menge dieser Folgen sei mit $S_{\star,f,t(n)}$ bezeichnet.

Wir beschränken die Wahrscheinlichkeit nach unten, eine beliebige Folge aus $S_{\dagger,f,t(n)}$ beim $(1+\lambda)$ -EA und beim $(1,\lambda)$ -EA zu beobachten. Damit $f(y_t) \geq f(x_{t-1})$ ist, genügt es, dass mindestens ein Nachkomme $y_{t,1}, \dots, y_{t,\lambda}$ von x_{t-1} ein Duplikat seines Elters ist. Die Wahrscheinlichkeit dafür beträgt mindestens $1 - (1 - (1 - 1/n)^n)^\lambda \geq 1 - (1 - 6/17)^\lambda = 1 - (11/17)^\lambda$ für n groß genug. Höchstens mit Wahrscheinlichkeit $(t(n) - 1) \cdot (11/17)^\lambda$ tritt dieses Ereignis für ein t mit $2 \leq t \leq t(n)$ nicht ein.

Lemma 2.B.1 *Sei $f : \{0, 1\}^n \rightarrow \mathbb{Z}$ eine beliebige Funktion und n groß genug. Mindestens mit Wahrscheinlichkeit $1 - (t(n) - 1) \cdot (11/17)^\lambda$ wird beim $(1+\lambda)$ -EA und beim $(1,\lambda)$ -EA (mit beliebiger, aber identischer Verteilung von x_1) eine Folge aus $S_{\dagger,f,t(n)}$, $t(n) \geq 1$, beobachtet.*

Mit Hilfe des Lemmas 2.B.1 lassen sich Erfolgswahrscheinlichkeiten und sogar Erwartungswerte vom $(1+\lambda)$ -EA auf den $(1,\lambda)$ -EA übertragen und umgekehrt. Dies gelingt insbesondere, wenn die Nachkommenpopulation groß ist – in Abhängigkeit des Betrachtungszeitraums. Die Anzahl durchgeführter Funktionsauswertungen bis zum einschließlich $t(n)$ -ten Schritt mit $t(n) \geq 1$ beträgt $1 + \lambda \cdot (t(n) - 1)$.

Betrachten wir $\ell(n)$ Funktionsauswertungen mit $1 \leq \ell(n) \leq 1 + \lambda$ – also höchstens zwei Schritte. Für jede Funktion f und beliebiges λ ist die Wahrscheinlichkeit gleich, dass der $(1\star\lambda)$ -EA und der $(1\bar{\star}\lambda)$ -EA nach $\ell(n)$ Funktionsauswertungen f nicht optimiert haben.

Betrachten wir $\ell(n)$ Funktionsauswertungen mit $2 + \lambda \cdot (t(n) - 1) \leq \ell(n) \leq 1 + \lambda \cdot t(n)$ und $t(n) \geq 2$ – also höchstens $t(n) + 1$ Schritte. Sei \mathcal{E}^* das Ereignis, dass der $(1\star\lambda)$ -EA nach $\ell(n)$ Funktionsauswertungen die Funktion f nicht optimiert hat. Dieses Ereignis tritt genau dann auf, wenn beim $(1\star\lambda)$ -EA eine Folge $s_{t(n)}$ beobachtet wird, deren angefragte Suchpunkte $x_1, y_{2,1}, \dots, y_{2,\lambda}, y_{3,1}, \dots, y_{3,\lambda}, \dots, y_{t(n)+1,1}, \dots, y_{t(n)+1,\ell(n)-\lambda \cdot (t(n)-1)-1}$ nicht optimal sind. Wir spalten dieses Ereignis auf in die zwei disjunkten Ereignisse \mathcal{E}_1^* , $s_{t(n)} \in S_{\dagger,f,t(n)}$, und \mathcal{E}_2^* , $s_{t(n)} \notin S_{\dagger,f,t(n)}$ und somit $s_{t(n)} \in S_{\star,f,t(n)}$. Nach vorheriger Überlegung, dass jede Folge aus $S_{\dagger,f,t(n)}$ mit gleicher Wahrscheinlichkeit beim $(1\star\lambda)$ -EA und beim $(1\bar{\star}\lambda)$ -EA beobachtet wird, ist $\Pr[\mathcal{E}_1^*] = \Pr[\mathcal{E}_1^{\bar{\star}}]$ und damit $\Pr[\mathcal{E}^{\bar{\star}}] = \Pr[\mathcal{E}_2^{\bar{\star}}] + \Pr[\mathcal{E}_1^{\bar{\star}}] = \Pr[\mathcal{E}_2^{\bar{\star}}] + \Pr[\mathcal{E}_1^*] = \Pr[\mathcal{E}_2^{\bar{\star}}] + \Pr[\mathcal{E}^*] - \Pr[\mathcal{E}_2^*]$.

Die Wahrscheinlichkeit, beim $(1\star\lambda)$ -EA und beim $(1\bar{\star}\lambda)$ -EA eine Folge aus $S_{\dagger,f,t(n)}$ zu beobachten, beträgt mindestens $1 - (t(n) - 1) \cdot (11/17)^\lambda$ nach Lemma 2.B.1. Nehmen wir an, dass

$\lambda \geq (5/2) \cdot (1 + c(n)) \cdot \ln t(n)$ ist mit $c(n) \geq 0$. Dann erhalten wir $1 - (t(n) - 1) \cdot (11/17)^\lambda \geq 1 - t(n) \cdot (11/17)^{(5/2) \cdot (1 + c(n)) \cdot \ln t(n)} \geq 1 - t(n) \cdot 1/t(n)^{1 + c(n)} = 1 - 1/t(n)^{c(n)}$, wobei die vorletzte Ungleichung $\ln(11/17) \cdot (5/2) \leq -1$ ausnutzt. Eine Folge aus $S_{*,f,t(n)}$ wird höchstens mit Wahrscheinlichkeit $1/t(n)^{c(n)}$ beim $(1 \star \lambda)$ -EA beobachtet. Da Ereignis \mathcal{E}_2^* voraussetzt, beim $(1 \star \lambda)$ -EA eine Folge aus $S_{*,f,t(n)}$ zu beobachten, ist $0 \leq \Pr[\mathcal{E}_2^*], \Pr[\mathcal{E}_2^*] \leq 1/t(n)^{c(n)}$ und damit $-1/t(n)^{c(n)} \leq \Pr[\mathcal{E}_2^*] - \Pr[\mathcal{E}_2^*] \leq 1/t(n)^{c(n)}$.

Lemma 2.B.2 Sei $f : \{0, 1\}^n \rightarrow \mathbb{Z}$ eine beliebige Funktion, n groß genug und sei $x_{(1)} \in \{0, 1\}^n$.

a) Für $0 \leq \ell(n) \leq 1 + \lambda$ gilt die Gleichung:

$$\Pr[T_{f,(1 \bar{\star} \lambda)\text{-EA}} > \ell(n) \mid x_1 = x_{(1)}] = \Pr[T_{f,(1 \star \lambda)\text{-EA}} > \ell(n) \mid x_1 = x_{(1)}]$$

b) Für $2 + \lambda \cdot (t(n) - 1) \leq \ell(n) \leq 1 + \lambda \cdot t(n)$, $t(n) \geq 2$, und $\lambda \geq (5/2) \cdot (1 + c(n)) \cdot \ln t(n)$, $c(n) \geq 0$, gelten die Ungleichungen:

$$\Pr[T_{f,(1 \bar{\star} \lambda)\text{-EA}} > \ell(n) \mid x_1 = x_{(1)}] \leq \Pr[T_{f,(1 \star \lambda)\text{-EA}} > \ell(n) \mid x_1 = x_{(1)}] + 1/t(n)^{c(n)}$$

$$\Pr[T_{f,(1 \bar{\star} \lambda)\text{-EA}} > \ell(n) \mid x_1 = x_{(1)}] \geq \Pr[T_{f,(1 \star \lambda)\text{-EA}} > \ell(n) \mid x_1 = x_{(1)}] - 1/t(n)^{c(n)}$$

Betrachten wir eine beliebige Funktion f mit eindeutigem Optimum x^* und den $(1 \star \lambda)$ -EA. Im Beweis des Theorems 2.E.1.a wird festgestellt werden, dass $H[x_1, x^*] > \lfloor n/3 \rfloor$ ist mit Wahrscheinlichkeit $1 - 2^{-\Omega(n)}$. Weiter beträgt die Wahrscheinlichkeit mindestens $1 - \lfloor n^{n/4} \rfloor \cdot 1/n^{\lfloor n/3 \rfloor} = 1 - 2^{-\Omega(n \ln n)}$, dass keiner der $\min\{\lambda, \lfloor n^{n/4} \rfloor\}$ (ersten) Nachkommen von x_1 das Optimum x^* ist. Dazu müssen mehr als $\lfloor n/3 \rfloor$ spezielle Bits in x_1 flippen. Also benötigt der $(1 \star \lambda)$ -EA zur Optimierung von f eine Laufzeit von mehr als $\min\{\lambda, n^{n/4}\}$ mit Wahrscheinlichkeit $1 - 2^{-\Omega(n)}$ und damit auch eine erwartete Laufzeit von $\Omega(\min\{\lambda, n^{n/4}\})$. Lediglich Nachkommenpopulationsgrößen $\lambda \leq \text{poly}(n)$ erlauben die mögliche effiziente Optimierung von f .

Nehmen wir an, der $(1 \star \lambda)$ -EA mit $\lambda = \omega(\ln n)$ optimiert die Funktion f effizient. Ihm genügt eine erwartete Laufzeit von $E[T_{f,(1 \star \lambda)\text{-EA}}] = \mathcal{O}(n^{c_1})$ zur Optimierung von f für eine Konstante $c_1 \geq 0$. Eine Anwendung der Markoffungleichung zeigt, dass für jede beliebige Konstante $c_2 \geq 0$ gilt $\Pr[T_{f,(1 \star \lambda)\text{-EA}} > \Theta(n^{c_1 + c_2})] \leq E[T_{f,(1 \star \lambda)\text{-EA}}] / \Theta(n^{c_1 + c_2}) = \mathcal{O}(n^{-c_2})$. Nach vorheriger Betrachtung kann $\lambda = \mathcal{O}(n^{c_1})$ angenommen werden. Es ist $\Pr[T_{f,(1 \bar{\star} \lambda)\text{-EA}} > \Theta(n^{c_1 + c_2})] \leq \Pr[T_{f,(1 \star \lambda)\text{-EA}} > \Theta(n^{c_1 + c_2})] + \mathcal{O}(n^{-c_2}) = \mathcal{O}(n^{-c_2})$ nach Lemma 2.B.2.b mit $c(n) = 1$, da $t(n) \leq (\Theta(n^{c_1 + c_2}) - 1) / \lambda = \Omega(n^{c_2})$ und $\lambda \geq 5 \ln t(n) = \mathcal{O}(\ln n)$ ist. Dem $(1 \bar{\star} \lambda)$ -EA genügt mit hoher Wahrscheinlichkeit $1 - o(1)$ eine polynomielle Laufzeit zur Optimierung von f , wenn dem $(1 \star \lambda)$ -EA dazu ebenfalls eine mit hoher Wahrscheinlichkeit $1 - o(1)$ oder eine erwartete polynomielle Laufzeit reicht.

Nach Betrachtung der Erfolgswahrscheinlichkeiten werden wir diese Resultate verwenden, um die Übertragbarkeit der Erwartungswerte genauer zu analysieren. Häufig gelingt es, eine obere Schranke für die erwartete Laufzeit einer Heuristik zur Optimierung einer Funktion herzuleiten, die sogar für jeden beliebigen ersten Suchpunkt gilt. Dies werden wir im nachfolgenden Lemma ausnutzen.

Lemma 2.B.3 Sei $f : \{0, 1\}^n \rightarrow \mathbb{Z}$ eine beliebige Funktion und n groß genug. Mit $t_{f,(1 \star \lambda)\text{-EA}}^{\max} := \max\{E[T_{f,(1 \star \lambda)\text{-EA}} \mid x_1 = x_{(1)}] \mid x_{(1)} \in \{0, 1\}^n\}$ gilt $t_{f,(1 \bar{\star} \lambda)\text{-EA}}^{\max} \leq 43 t_{f,(1 \star \lambda)\text{-EA}}^{\max}$ falls $\ln t_{f,(1 \star \lambda)\text{-EA}}^{\max} \geq 20$ und $\lambda \geq 11(\ln t_{f,(1 \star \lambda)\text{-EA}}^{\max})/3$ ist.

Beweis. Wir betrachten zunächst die ersten $\ell(n)$ mit $0 \leq \ell(n) \leq 1 + \lambda$ Funktionsauswertungen des $(1\bar{\lambda})$ -EA mit einem beliebigen Individuum $x_1 = x_{(1)} \in \{0, 1\}^n$. Nach Lemma 2.B.2.a gilt $\Pr[T_{f,(1\bar{\lambda})\text{-EA}} > \ell(n) | x_1 = x_{(1)}] = \Pr[T_{f,(1\star\lambda)\text{-EA}} > \ell(n) | x_1 = x_{(1)}]$. Damit beträgt die erwartete Laufzeit des $(1\bar{\lambda})$ -EA, bis entweder ein Optimum von f gefunden oder der zweite Schritt beendet wurde, $E[\min\{T_{f,(1\bar{\lambda})\text{-EA}}, 1 + \lambda\} | x_1 = x_{(1)}] = \sum_{\ell=0}^{\lambda} \Pr[T_{f,(1\bar{\lambda})\text{-EA}} > \ell | x_1 = x_{(1)}] = \sum_{\ell=0}^{\lambda} \Pr[T_{f,(1\star\lambda)\text{-EA}} > \ell | x_1 = x_{(1)}] \leq \min\{t_{f,(1\star\lambda)\text{-EA}}^{\max}, 1 + \lambda\}$. Die Wahrscheinlichkeit, dass der $(1\bar{\lambda})$ -EA mehr als zwei Schritte zur Optimierung von f benötigt, ist mit der Markoffungleichung $\Pr[T_{f,(1\bar{\lambda})\text{-EA}} > 1 + \lambda | x_1 = x_{(1)}] = \Pr[T_{f,(1\star\lambda)\text{-EA}} > 1 + \lambda | x_1 = x_{(1)}] \leq \min\{t_{f,(1\star\lambda)\text{-EA}}^{\max}/(1 + \lambda), 1\}$.

Wir betrachten nun die i -te Phase, $i \geq 0$, des $(1\bar{\lambda})$ -EA. Diese umfasst $7\lceil(t_{f,(1\star\lambda)\text{-EA}}^{\max} - 1)/\lambda\rceil =: t(n) \geq 7$ aufeinanderfolgende Schritte und beginnt, nachdem der $(2 + i \cdot t(n))$ -te Schritt durchgeführt wurde. Nehmen wir an, dass bis zu diesem Schritt kein Optimum gefunden wurde, also $x_1, \dots, x_{2+i \cdot t(n)} \notin \arg \max\{f(x) | x \in \{0, 1\}^n\}$ gilt. Dann fallen höchstens $\lambda \cdot t(n)$ weitere Funktionsauswertungen an – bis ein Optimum gefunden wurde oder die $(i + 1)$ -te Phase beginnt. Da zum einen die Bestimmung der Nachfolger des ausgewählten Nachfolgers und des nächsten Elters nicht von der Anzahl vorangegangener Schritte abhängen und zum anderen der Funktionswert von $x_{2+i \cdot t(n)}$ bereits bekannt ist, gilt für jedes beliebige $x_{2+i \cdot t(n)} = x \in \{0, 1\}^n$

$$\Pr[T_{f,(1\bar{\lambda})\text{-EA}} > 1 + (i+1) \cdot \lambda \cdot t(n) + \lambda \cdot t(n) | x_{2+i \cdot t(n)} = x] = \Pr[T_{f,(1\bar{\lambda})\text{-EA}} - 1 > \lambda \cdot t(n) | x_1 = x].$$

Nach Lemma 2.B.2.b mit $c(n) = 1/3$ gilt für n groß genug weiter, da $t(n) \leq 7t_{f,(1\star\lambda)\text{-EA}}^{\max}$ und $10(\ln 7t_{f,(1\star\lambda)\text{-EA}}^{\max})/3 \leq 11(\ln t_{f,(1\star\lambda)\text{-EA}}^{\max})/3 \leq \lambda$ für $\ln t_{f,(1\star\lambda)\text{-EA}}^{\max} \geq 20$ ist,

$$\begin{aligned} \Pr[T_{f,(1\bar{\lambda})\text{-EA}} - 1 > \lambda \cdot t(n) | x_1 = x] &\leq \Pr[T_{f,(1\star\lambda)\text{-EA}} - 1 > \lambda \cdot t(n) | x_1 = x] + 1/t(n)^{1/3} \\ &\leq \frac{t_{f,(1\star\lambda)\text{-EA}}^{\max} - 1}{\lambda \cdot 7\lceil(t_{f,(1\star\lambda)\text{-EA}}^{\max} - 1)/\lambda\rceil} + \frac{1}{t(n)^{1/3}} \leq \frac{1}{7} + \frac{1}{7^{1/3}} \leq \frac{2}{3}, \end{aligned}$$

wobei für die zweite Ungleichung abermals die Markoffungleichung angewendet wurde. Die Wahrscheinlichkeit, vor Beginn der i -ten Phase kein Optimum erzeugt zu haben, beträgt höchstens $\min\{t_{f,(1\star\lambda)\text{-EA}}^{\max}/(1 + \lambda), 1\} \cdot (2/3)^i$. Insgesamt erhalten wir eine erwartete Laufzeit des $(1\bar{\lambda})$ -EA mit beliebigem x_1 von höchstens

$$\begin{aligned} &t_{f,(1\bar{\lambda})\text{-EA}}^{\max} \\ &\leq \min\{t_{f,(1\star\lambda)\text{-EA}}^{\max}, 1 + \lambda\} + \sum_{i=0}^{\infty} \min\{t_{f,(1\star\lambda)\text{-EA}}^{\max}/(1 + \lambda), 1\} \cdot \left(\frac{2}{3}\right)^i \cdot (\lambda \cdot t(n)) \\ &\leq \min\{t_{f,(1\star\lambda)\text{-EA}}^{\max} + (t_{f,(1\star\lambda)\text{-EA}}^{\max}/(1 + \lambda)) \cdot 3 \cdot (\lambda \cdot 7), (1 + \lambda) + 1 \cdot 3 \cdot (\lambda \cdot 7 \lceil \frac{t_{f,(1\star\lambda)\text{-EA}}^{\max} - 1}{\lambda} \rceil)\} \\ &\leq \min\{t_{f,(1\star\lambda)\text{-EA}}^{\max} + 21t_{f,(1\star\lambda)\text{-EA}}^{\max}, t_{f,(1\star\lambda)\text{-EA}}^{\max} + 21t_{f,(1\star\lambda)\text{-EA}}^{\max} + 21t_{f,(1\star\lambda)\text{-EA}}^{\max}\} \leq 43t_{f,(1\star\lambda)\text{-EA}}^{\max}, \end{aligned}$$

wobei die Fälle $\lambda \geq t_{f,(1\star\lambda)\text{-EA}}^{\max} - 1$ und $\lambda < t_{f,(1\star\lambda)\text{-EA}}^{\max} - 1$ unterschieden und die Konvergenzeigenschaft unendlicher geometrischer Reihen ausgenutzt wurde. \square

Wir wenden dieses Resultat exemplarisch im nächsten Abschnitt an.

2.C Typische Zielfunktion

Betrachten wir die typische Funktion ONEMAX. Jansen et al. (2005) haben die Laufzeit des $(1+\lambda)$ -EA zur Optimierung der (unimodalen) Funktion ONEMAX analysiert. Aussage a) des folgenden Theorems fasst dies zusammen. Wir übertragen und erweitern dieses Resultat mit Hilfe der in den letzten Abschnitten gewonnenen Erkenntnisse auf den $(1,\lambda)$ -EA. Da Aussage a) sogar für beliebiges x_1 gilt, kann Lemma 2.B.3 angewendet werden. Mit $\lambda \geq 4 \ln n$ beträgt die erwartete Laufzeit des $(1+\lambda)$ -EA mit beliebigem x_1 zur Optimierung von ONEMAX höchstens $c\lambda n$ für eine genügend große Konstante c . Für $\lambda \geq 4 \ln n$ gilt $11 \ln(c\lambda n)/3 \leq \lambda$ für n groß genug, da $11 \ln(n)/3 \leq 11\lambda/12$ und $11 \ln(c\lambda)/3 \leq \lambda/12$ ist. Mit Lemma 2.B.3 ergibt sich Aussage b) des folgenden Theorems. Aussage c) folgt direkt nach Theorem 2.A.1, da ONEMAX ein eindeutiges Optimum hat.

Theorem 2.C.1

- Der $(1+\lambda)$ -EA erzeugt das Optimum von ONEMAX in einer erwarteten Laufzeit von
 - $\mathcal{O}(n \ln n)$, falls $\lambda = \mathcal{O}\left(\frac{(\ln n)(\ln \ln n)}{\ln \ln \ln n}\right)$ ist, und von
 - $\mathcal{O}(\lambda n)$, falls $\lambda = \Omega(\ln n)$ ist.
- Sei $\lambda \geq 4 \ln n$. Der $(1,\lambda)$ -EA erzeugt das Optimum von ONEMAX in einer erwarteten Laufzeit von
 - $\mathcal{O}(n \ln n)$, falls $\lambda = \mathcal{O}\left(\frac{(\ln n)(\ln \ln n)}{\ln \ln \ln n}\right)$ ist, und von
 - $\mathcal{O}(\lambda n)$.
- Sei $\lambda \leq \varepsilon(n) \ln(n)/7$ mit $\varepsilon(n) \in [7/\ln n, 1/2]$. Mit Wahrscheinlichkeit $1 - 2^{-\Omega(n^{1-\varepsilon(n)})}$ benötigt der $(1,\lambda)$ -EA eine Laufzeit von mehr als $2^{n^{1-\varepsilon(n)}}$ zur Optimierung von ONEMAX.

2.D Erweiterungen der Methode der f -basierten Partition

Wir erweitern die Methode der f -basierten Partition für den $(1+1)$ -EA nach Lemma 1.B.1 auf den $(1+\lambda)$ -EA. Dabei wird ausgenutzt, dass zum Wechseln der Partition von A_i nach A_{i+1}, \dots, A_m es beim $(1+\lambda)$ -EA ausreicht, mindestens einen seiner λ Nachkommen aus $A_{i+1} \cup \dots \cup A_m$ zu erzeugen, während es beim $(1+1)$ -EA notwendig ist, seinen einzigen Nachkommen aus $A_{i+1} \cup \dots \cup A_m$ zu generieren.

Lemma 2.D.1 Für eine beliebige Funktion $f : \{0, 1\}^n \rightarrow \mathbb{Z}$ sei A_0, \dots, A_m eine f -basierte Partition und sei

$$p_i^+ := 1 - (1 - p(A_i))^\lambda \text{ für } 0 \leq i \leq m.$$

Der $(1+\lambda)$ -EA (mit beliebiger Verteilung von x_1) erzeugt ein Optimum von f in einer erwarteten Laufzeit von höchstens

$$1 + \lambda \left(\frac{1}{p_0^+} + \dots + \frac{1}{p_{m-1}^+} \right).$$

Mit $\lambda = 1$ erhalten wir das gleiche Resultat wie Lemma 1.B.1 und Wegener (2002) für den $(1+1)$ -EA.

Beweis (Lemma 2.D.1). Wir beschreiben einen Markoffprozess \mathcal{M} mit $m + 1$ Zuständen, der nach $t(n)$ Schritten mit höchstens gleicher Wahrscheinlichkeit einen Zustand i, \dots, m erreicht wie der $(1+\lambda)$ -EA einen Suchpunkt $x_{t(n)}$ mit $x_{t(n)} \in A_i \cup \dots \cup A_m$. Dies wird für alle i mit $0 \leq i \leq m$ gelten. Der $(1+\lambda)$ -EA generiert also mindestens mit gleicher Wahrscheinlichkeit ein Optimum, wie \mathcal{M} den Zustand m erreicht. Setzen wir $p_0 := 1$ und $p_i := 0$, $i \geq 1$, hat \mathcal{M} die gewünschte Eigenschaft für $t(n) = 1$ und beliebiges x_1 . Befindet sich \mathcal{M} nach $t(n)$ Schritten im Zustand i , gilt für das Element $x_{t(n)}$ des $(1+\lambda)$ -EA mit zumindest gleicher Wahrscheinlichkeit $x_{t(n)} \in A_i \cup \dots \cup A_m$. Es ist unmöglich, ein $x_{t(n)+1}$ mit $x_{t(n)+1} \in A_0 \cup \dots \cup A_{i-1}$ als nächsten Elter zu erzeugen. Weiterhin ist p_i^+ eine untere Schranke für die Wahrscheinlichkeit, dass der $(1+\lambda)$ -EA ein $x_{t(n)+1}$ mit $x_{t(n)+1} \in A_{i+1} \cup \dots \cup A_m$ generiert. Dazu genügt es, einen der λ Nachkommen von $x_{t(n)}$ aus dieser Menge zu erzeugen. Damit \mathcal{M} die gewünschte Eigenschaft auch im nachfolgenden Schritt erfüllt, genügt es, $p_{i,j} := 0$ für $0 \leq j < i \leq m$ und $i + 2 \leq j \leq m$, $p_{i,i+1} := p_i^+$ und $p_{i,i} := 1 - p_i^+$ für $0 \leq i < m$ sowie $p_{m,m} := 1$ zu setzen.

Die erwartete Anzahl Schritte E_i , um vom Zustand i zum Zustand m zu gelangen, beträgt $E_i = 1 + p_i^+ \cdot E_{i+1} + (1 - p_i^+) \cdot E_i$ für $0 \leq i < m$ und $E_m = 0$. Für $0 \leq i < m$ ist dies äquivalent zu $E_i = 1/p_i^+ + E_{i+1}$ und damit gilt $E_0 = 1/p_0^+ + \dots + 1/p_{m-1}^+$. Mit der Betrachtung für den ersten Schritt und da in jedem weiteren Schritt λ Funktionsauswertungen anfallen, beträgt die erwartete Laufzeit des $(1+\lambda)$ -EA zur Optimierung von f höchstens $1 + \lambda \cdot E_0$. \square

Wir wenden diese Methode exemplarisch auf ONEMAX an. Betrachten wir die Partitionierung in A_i , $0 \leq i \leq n$, mit $A_i := \{x \mid |x| = i\}$, gilt (vergleiche Jansen et al., 2005)

$$p_i^+ \geq 1 - \left(1 - \frac{n-i}{en}\right)^\lambda \geq 1 - e^{-\lambda(n-i)/(en)} \geq 1 - \frac{1}{1 + \lambda(n-i)/(en)} = \frac{\lambda(n-i)}{en + \lambda(n-i)}.$$

Nach Lemma 2.D.1 beträgt die erwartete Laufzeit des $(1+\lambda)$ -EA zur Optimierung von ONEMAX höchstens (vergleiche Jansen et al., 2005)

$$1 + \lambda \sum_{i=0}^{n-1} \frac{en + \lambda(n-i)}{\lambda(n-i)} \leq 1 + \lambda n + en \sum_{i=1}^n \frac{1}{i} \leq 1 + \lambda n + en(\ln n + 1) = \mathcal{O}(n \ln n + \lambda n).$$

Dies zeigt bereits einen wesentlichen Teil von Aussage a) des Theorems 2.C.1.

Wir betrachten den $(1, \lambda)$ -EA. Mit Lemma 2.B.3 kann das Lemma 2.D.1 für den $(1+\lambda)$ -EA leicht auf den $(1, \lambda)$ -EA übertragen werden. Bei Schritten des $(1, \lambda)$ -EA, die kein Duplikat erzeugen, wird damit pessimistisch angenommen, dass ein beliebig schlechter Suchpunkt entsteht. Wir können jedoch auch optimistisch annehmen, dass es von Vorteil ist, gelegentlich einen schlechteren Suchpunkt zu akzeptieren. Im Folgenden werden wir die Eigenschaft der f -basierten Partition dahingehend abschwächen.

Sei A_0, \dots, A_m eine – nicht notwendigerweise f -basierte – Partition von $\{0, 1\}^n$, aber A_m enthalte nur Optima von f . Weiterhin sei

- $p^-(x)$, $x \in A_i$, die Wahrscheinlichkeit, dass eine Mutation von x ein Element x' mit $x' \in A_0 \cup \dots \cup A_{i-1}$ und $f(x') \geq f(x)$ erzeugt, und

- $p^+(x)$, $x \in A_i$, die Wahrscheinlichkeit, dass eine Mutation von x ein Element x' mit $x' \in A_{i+1} \cup \dots \cup A_m$ und $f(x') > f(x)$ erzeugt.

Somit ist $p^-(A_i) := \max\{p^-(x) \mid x \in A_i\}$ eine obere Schranke für die Wahrscheinlichkeit, dass für den Nachkommen $y_{t,j}$, $1 \leq j \leq \lambda$, von $x_{t-1} \in A_i$ die Ungleichung $f(y_{t,j}) \geq f(x_{t-1})$ gilt sowie $y_{t,j} \in A_0 \cup \dots \cup A_{i-1}$ und damit möglicherweise $x_t = y_{t,j}$ ist – selbst wenn ein Duplikat von x_{t-1} entsteht. Dies entspricht dem (möglichen) Wechsel in eine Partition mit geringerem Index. Ebenso ist $p^+(A_i) := \min\{p^+(x) \mid x \in A_i\}$ eine untere Schranke für die Wahrscheinlichkeit, dass für den Nachkommen $y_{t,j}$, $1 \leq j \leq \lambda$, von $x_{t-1} \in A_i$ die Ungleichung $f(y_{t,j}) > f(x_{t-1})$ gilt sowie $y_{t,j} \in A_{i+1} \cup \dots \cup A_m$ und damit möglicherweise $x_t = y_{t,j}$ ist – selbst wenn ein Duplikat von x_{t-1} entsteht. Dies entspricht dem (möglichen) Wechsel in eine Partition mit höherem Index.

Sicher gilt $p(A_i) \geq p^+(A_i)$. Falls $p^-(A_i) = 0$ für alle i mit $0 \leq i \leq m$ ist und daher für alle $z \in A_{i-1}$ sowie $x \in A_i$, $1 \leq i \leq m$, die Ungleichung $f(z) < f(x)$ gilt, stellt A_0, \dots, A_m sogar eine f -basierte Partition dar und dann ist $p^+(A_i) = p(A_i)$ für alle i mit $0 \leq i \leq m$.

Lemma 2.D.2 Für eine beliebige Funktion $f : \{0, 1\}^n \rightarrow \mathbb{Z}$ sei A_0, \dots, A_m eine Partition von $\{0, 1\}^n$ mit $\emptyset \neq A_m \subseteq \arg \max\{f(x) \mid x \in \{0, 1\}^n\}$. Sei

$$\begin{aligned} p_i^+ &:= \max\{(1 - p^-(A_i))^\lambda - (1 - p^-(A_i) - p^+(A_i))^\lambda; p(A_i)^\lambda\} \text{ für } 0 \leq i \leq m, \\ p_i^- &:= 1 - (1 - p^-(A_i))^\lambda + (11/17 - p^-(A_i) - p^+(A_i))^\lambda \text{ für } 0 \leq i \leq m \end{aligned}$$

und n groß genug. Der $(1, \lambda)$ -EA (mit beliebiger Verteilung von x_1) erzeugt ein Optimum von f in einer erwarteten Laufzeit von höchstens

$$1 + \lambda \left(\frac{1}{p_0^+} + \frac{1}{p_1^+ + p_1^-} + \dots + \frac{1}{p_{m-1}^+ + p_{m-1}^-} \right) \cdot \frac{p_1^+ + p_1^-}{p_1^+} \dots \frac{p_{m-1}^+ + p_{m-1}^-}{p_{m-1}^+}.$$

Liegt sogar eine f -basierte Partition vor, so ist $p_i^+ \geq 1 - (1 - p(A_i))^\lambda$ und $p_i^- \leq (11/17)^\lambda$ nach den obigen Betrachtungen. Sei $\lambda \geq 5 \ln(m \cdot \max\{1/p_i^+ \mid 1 \leq i \leq m-1\})/2$. Da dann $(11/17)^\lambda \leq 1/(m \cdot \max\{1/p_i^+ \mid 1 \leq i \leq m-1\}) \leq p_i^+/m$ ist für alle i mit $1 \leq i \leq m-1$, gilt insbesondere $\prod_{i=1}^{m-1} (1 + (11/17)^\lambda/p_i^+) \leq (1 + 1/m)^{m-1} \leq e$. Mit $1/(p_i^+ + p_i^-) \leq 1/p_i^+$ erhalten wir das folgende Resultat. Falls $\lambda \geq 5 \ln(m \cdot \max\{1/p_i^+ \mid 1 \leq i \leq m-1\})/2$ ist, erzeugt der $(1, \lambda)$ -EA (mit beliebiger Verteilung von x_1) ein Optimum von f in einer erwarteten Laufzeit von höchstens

$$e \cdot \left(1 + \lambda \left(\frac{1}{p_0^+} + \dots + \frac{1}{p_{m-1}^+} \right) \right).$$

Wir bemerken, dass diese erwartete Laufzeit nur um einen Faktor e größer ist als jene für den $(1+\lambda)$ -EA und die gleiche Situation nach Lemma 2.D.1.

Beweis (Lemma 2.D.2). Es ist p_i^+ eine untere Schranke für die Wahrscheinlichkeit, dass der $(1, \lambda)$ -EA mit dem Individuum $x_{t(n)} \in A_i$ als nächsten Elter $x_{t(n)+1} \in A_{i+1} \cup \dots \cup A_m$ generiert, da es ausreicht – damit dieses Ereignis eintritt –, dass entweder

- mindestens ein Nachkomme aus $A_{i+1} \cup \dots \cup A_m$ erzeugt wird, der einen größeren Funktionswert hat als jedes Element aus A_i , aber keine Nachkommen aus $A_1 \cup \dots \cup A_{i-1}$, die einen mindestens ebenso großen Funktionswert haben wie jedes Element aus A_i ; dies hat mindestens Wahrscheinlichkeit $\sum_{j=1}^{\lambda} \binom{\lambda}{j} p^+(A_i)^j \cdot (1 - p^-(A_i) - p^+(A_i))^{\lambda-j} = (1 - p^-(A_i))^{\lambda} - (1 - p^-(A_i) - p^+(A_i))^{\lambda}$; oder
- nur Nachkommen aus $A_{i+1} \cup \dots \cup A_m$ erzeugt werden; dies hat mindestens Wahrscheinlichkeit $p(A_i)^{\lambda}$.

Ebenso ist p_i^- eine obere Schranke für die Wahrscheinlichkeit, dass der $(1, \lambda)$ -EA mit dem Individuum $x_{t(n)} \in A_i$ als nächsten Elter $x_{t(n)+1} \in A_1 \cup \dots \cup A_{i-1}$ generiert, da es ausreicht – damit dieses Ereignis nicht eintritt –, dass

- mindestens ein Nachkomme aus $A_i \cup A_{i+1} \cup \dots \cup A_m$ erzeugt wird, der einen mindestens so großen Funktionswert hat wie jedes Element aus A_i , aber keine Nachkommen aus $A_1 \cup \dots \cup A_{i-1}$, die einen mindestens ebenso großen Funktionswert haben wie ein beliebiges Element aus A_i . Dies hat für n groß genug mindestens Wahrscheinlichkeit $\sum_{j=1}^{\lambda} \binom{\lambda}{j} (6/17 + p^+(A_i))^j \cdot (1 - p^-(A_i) - (6/17 + p^+(A_i)))^{\lambda-j} = (1 - p^-(A_i))^{\lambda} - (11/17 - p^-(A_i) - p^+(A_i))^{\lambda}$, da die Wahrscheinlichkeit eines Duplikates $(1 - 1/n)^n \geq 6/17$ beträgt.

Sei $p_{x,x'}$ die Wahrscheinlichkeit, dass der $(1, \lambda)$ -EA mit dem Individuum x den nächsten Elter x' generiert, und sei T_x die erwartete Anzahl Schritte des $(1, \lambda)$ -EA, um mit dem Elter x erstmals einem Suchpunkt aus A_m zu erreichen. Es gilt $T_x = 0$, falls $x \in A_m$ ist, und sonst mit $x \in A_i$ gilt

$$T_x = 1 + \sum_{x' \in A_0 \cup \dots \cup A_{i-1}} p_{x,x'} T_{x'} + \sum_{x' \in A_i} p_{x,x'} T_{x'} + \sum_{x' \in A_{i+1} \cup \dots \cup A_m} p_{x,x'} T_{x'}.$$

Da $T_x \leq \max\{T_x \mid x \in A_i \cup \dots \cup A_m\} =: T_i$ und somit $T_i \geq T_{i+1}$ ist, gilt

$$T_x \leq 1 + \sum_{x' \in A_0 \cup \dots \cup A_{i-1}} p_{x,x'} T_0 + \sum_{x' \in A_i} p_{x,x'} T_i + \sum_{x' \in A_{i+1} \cup \dots \cup A_m} p_{x,x'} T_{i+1}.$$

Wie zuvor festgestellt, ist $\sum_{x' \in A_{i+1} \cup \dots \cup A_m} p_{x,x'} \geq p_i^+$ sowie $\sum_{x' \in A_0 \cup \dots \cup A_{i-1}} p_{x,x'} \leq p_i^-$ und daher gilt für jedes $x \in A_i$

$$T_x \leq 1 + p_i^- T_0 + (1 - p_i^- - p_i^+) T_i + p_i^+ T_{i+1}.$$

Somit ist $\max\{T_x \mid x \in A_i\} \leq 1 + p_i^- T_0 + (1 - p_i^- - p_i^+) T_i + p_i^+ T_{i+1}$ und mit $T_{i+1} \leq 1 + p_i^- T_0 + (1 - p_i^- - p_i^+) T_i + p_i^+ T_{i+1}$ gilt

$$T_i = \max\{\max\{T_x \mid x \in A_i\}, T_{i+1}\} \leq 1 + p_i^- T_0 + (1 - p_i^- - p_i^+) T_i + p_i^+ T_{i+1}.$$

Da $T_x \leq T_0$ ist für alle $x \in \{0, 1\}^n$, sind wir an einer oberen Schranke für T_0 interessiert.

Wir betrachten den folgenden Markoffprozess \mathcal{M} mit $m + 1$ Zuständen. Für $0 < i < m$ sei $p_{i,0} := p_i^-$, $p_{i,i+1} := p_i^+$, $p_{i,i} := 1 - p_i^- - p_i^+$ sowie $p_{0,0} := 1 - p_0^+$, $p_{0,1} := p_0^+$, $p_{m,0} := p_m^-$, $p_{m,m} := 1 - p_m^-$. Weiterhin sei $p_{i,j} := 0$ für $1 \leq j \leq i - 1$ und $i + 2 \leq j \leq m$ mit $0 \leq i \leq m$.

Für die erwartete Anzahl Schritte E_i , um vom Zustand i zum Zustand m zu gelangen, gilt $E_i \geq T_i$. Wir weisen induktiv über i nach, dass

$$E_0 \leq \left(\frac{1}{p_0^+} + \frac{1}{p_1^+ + p_1^-} + \cdots + \frac{1}{p_{i-1}^+ + p_{i-1}^-} \right) \cdot \frac{p_1^+ + p_1^-}{p_1^+} \cdots \frac{p_{i-1}^+ + p_{i-1}^-}{p_{i-1}^+} + E_i$$

ist für alle i mit $0 \leq i \leq m$. Mit der Betrachtung für den ersten Schritt und $i = m$ beträgt die erwartete Laufzeit des $(1+\lambda)$ -EA zur Optimierung von f höchstens $1 + \lambda \cdot E_0$, da $E_m = 0$ gilt und in jedem Schritt – bis auf den ersten – λ Funktionsauswertungen anfallen. Es ist

$$E_0 = E_0,$$

was die Behauptung für $i = 0$ zeigt, und

$$E_0 = 1 + (1 - p_0^+)E_0 + p_0^+E_1 \Leftrightarrow E_0 = 1/p_0^+ + E_1,$$

was die Behauptung für $i = 1$ zeigt. Ebenso gilt

$$E_i = 1 + p_i^-E_0 + (1 - p_i^- - p_i^+)E_i + p_i^+E_{i+1} \Leftrightarrow E_i = \frac{1 + p_i^-E_0 + p_i^+E_{i+1}}{p_i^+ + p_i^-}$$

und mit $\frac{1}{p_i^+ + p_i^-} \leq \frac{1}{p_i^+ + p_i^-} \cdot \frac{p_1^+ + p_1^-}{p_1^+} \cdots \frac{p_{i-1}^+ + p_{i-1}^-}{p_{i-1}^+}$ sowie der Voraussetzung folgt

$$\begin{aligned} & E_0 \cdot \left(1 - \frac{p_i^-}{p_i^+ + p_i^-}\right) \\ & \leq \left(\frac{1}{p_0^+} + \frac{1}{p_1^+ + p_1^-} + \cdots + \frac{1}{p_{i-1}^+ + p_{i-1}^-} + \frac{1}{p_i^+ + p_i^-} \right) \cdot \frac{p_1^+ + p_1^-}{p_1^+} \cdots \frac{p_{i-1}^+ + p_{i-1}^-}{p_{i-1}^+} + \frac{p_i^+E_{i+1}}{p_i^+ + p_i^-}. \end{aligned}$$

Schlussendlich ist $\left(1 - \frac{p_i^-}{p_i^+ + p_i^-}\right) \cdot \frac{p_i^+ + p_i^-}{p_i^+} = 1$ und es gilt

$$E_0 \leq \left(\frac{1}{p_0^+} + \frac{1}{p_1^+ + p_1^-} + \cdots + \frac{1}{p_{i-1}^+ + p_{i-1}^-} + \frac{1}{p_i^+ + p_i^-} \right) \cdot \frac{p_1^+ + p_1^-}{p_1^+} \cdots \frac{p_{i-1}^+ + p_{i-1}^-}{p_{i-1}^+} \cdot \frac{p_i^+ + p_i^-}{p_i^+} + E_{i+1}.$$

Das gewünschte Resultat ist somit gezeigt. \square

Wir wenden diese Methode exemplarisch im nächsten Abschnitt an.

2.E Weder kleine noch große Nachkommenpopulationen

Aufschlussreich ist ein Vergleich der Effizienz des $(1+\lambda)$ -EA und des $(1,\lambda)$ -EA zur Optimierung einer Funktion f mit eindeutigem Optimum insbesondere für $\lambda = \Theta(\ln n)$. Abschnitt 2.A hat gezeigt, dass der $(1,\lambda)$ -EA mit $\lambda \leq \ln(n)/14$ auf f notwendigerweise total ineffizient ist und Abschnitt 2.B hat nachgewiesen, dass der $(1+\lambda)$ -EA mit $\lambda = \omega(\ln n)$ nicht

effizient sein kann, während der $(1, \lambda)$ -EA total ineffizient ist und umgekehrt. Höchstens bei Nachkommenpopulationsgrößen $\lambda = \Theta(\ln n)$ darf auf ein wesentlich verschiedenes Verhalten des $(1+\lambda)$ -EA und des $(1, \lambda)$ -EA gehofft werden.

Wir weisen nun exemplarisch für die (multimodale) Funktion $\text{CLIFF} : \{0, 1\}^n \rightarrow \mathbb{Z}$ mit dem eindeutigen Optimum 1^n Folgendes nach. Der $(1, \lambda)$ -EA mit logarithmisch in n großer Nachkommenpopulation λ optimiert CLIFF effizient, während der $(1+\lambda)$ -EA sogar mit beliebiger Nachkommenpopulationsgröße λ total ineffizient ist. Sei

$$\text{CLIFF}(x) := \begin{cases} |x| - \lfloor n/3 \rfloor & , \text{ falls } |x| \geq n - \lfloor n/3 \rfloor, \\ |x| & , \text{ falls } |x| < n - \lfloor n/3 \rfloor. \end{cases}$$

Der $(1+\lambda)$ -EA wartet typischerweise lange auf dem *Kliff* (engl. *cliff*) – das umfasst alle Individuen x mit $|x| < n - \lfloor n/3 \rfloor$ – während der $(1, \lambda)$ -EA es schnell überwindet und auch nicht auf das Kliff zurückkehrt.

Theorem 2.E.1

- Mit Wahrscheinlichkeit $1 - 2^{-\Omega(n)}$ benötigt der $(1+\lambda)$ -EA eine Laufzeit von mehr als $n^{n/4}$ zur Optimierung von CLIFF .
- Sei $\lambda \geq 5 \ln n$. Der $(1, \lambda)$ -EA erzeugt das Optimum von CLIFF in einer erwarteten Laufzeit von $\mathcal{O}(e^{5\lambda})$.
- Sei n groß genug. Der $(1, \lambda)$ -EA benötigt eine erwartete Laufzeit von mehr als $\min\{n^{n/4}, e^{\lambda/4}\}/3$ zur Optimierung von CLIFF .
- Sei $\lambda \leq \varepsilon(n) \ln(n)/7$ mit $\varepsilon(n) \in [7/\ln n, 1/2]$. Mit Wahrscheinlichkeit $1 - 2^{-\Omega(n^{1-\varepsilon(n)})}$ benötigt der $(1, \lambda)$ -EA eine Laufzeit von mehr als $2^{n^{1-\varepsilon(n)}}$ zur Optimierung von CLIFF .

Beweis. Im Folgenden sei n groß genug, was für die Beweise der Aussagen angenommen werden darf.

- Wir bemerken, dass der $(1+\lambda)$ -EA mit $|x_{t(n)}| < n - \lfloor n/3 \rfloor$ höchstens Suchpunkte $x_{t(n)+1}$ mit $|x_{t(n)+1}| < n - \lfloor n/3 \rfloor$ (Fall 1) oder $|x_{t(n)+1}| \geq |x_{t(n)}| + \lfloor n/3 \rfloor$ (Fall 2) erzeugt. Damit Fall 2 eintritt, müssen mindestens $\lfloor n/3 \rfloor$ (irgendwelche) Bit flippen. Die Wahrscheinlichkeit, dass dies wenigstens einmal bei $\lfloor n^{n/4} \rfloor$ Mutationen passiert, beträgt höchstens $\lfloor n^{n/4} \rfloor \cdot 1/(\lfloor n/3 \rfloor!) \leq n^{n/4} \cdot (e/\lfloor n/3 \rfloor)^{\lfloor n/3 \rfloor} = 2^{-\Omega(n \ln n)}$. Solange Fall 1 eintritt, wird das Optimum nicht gefunden. Die Wahrscheinlichkeit, dass $|x_1| \geq n - \lfloor n/3 \rfloor$ ist, beträgt höchstens

$$\sum_{i=n-\lfloor n/3 \rfloor}^n \binom{n}{i} / 2^n \leq n \cdot \binom{n}{\lfloor n/3 \rfloor} / 2^n.$$

Weiterhin ist $n \cdot \binom{n}{\lfloor n/3 \rfloor} \leq n \cdot \frac{n \cdots (n-\lfloor n/6 \rfloor + 1) \cdots (n-\lfloor n/6 \rfloor) \cdots (n-\lfloor n/3 \rfloor + 1)}{(1/n) \cdot \lfloor n/3 + 1 \rfloor!} \leq n^2 \cdot \frac{n^{\lfloor n/6 \rfloor} \cdot (5n/6)^{\lfloor n/3 \rfloor - \lfloor n/6 \rfloor}}{2\sqrt{n/3} \cdot (n/(3e))^{\lfloor n/3 \rfloor}} \leq n^2 \cdot 2^{\lfloor n/3 \rfloor \cdot (\log_2(5/6)/2 + \log_2(3e))} \leq 2^{28n/29}$ mit der Stirlingungleichung. Es gilt

$$n \cdot \binom{n}{\lfloor n/3 \rfloor} / 2^n \leq 2^{28n/29} / 2^n = 2^{-\Omega(n)}.$$

Mit Wahrscheinlichkeit $1 - 2^{-\Omega(n \ln n)} - 2^{-\Omega(n)}$ ist $|x_1| < n - \lfloor n/3 \rfloor$ und bei den ersten $\lfloor n^{n/4} \rfloor$ Mutationen tritt Fall 2 nicht ein. Das Optimum von CLIFF wird dann nicht erzeugt.

- b) Für $\lambda > n \ln n$ folgt die Aussage direkt, da das Optimum bei jeder Mutation zumindest mit Wahrscheinlichkeit $1/n^n$ erzeugt wird. Sei $\lambda \leq n \ln n$, $\ell := \lfloor \ln \lambda / \ln \ln \lambda \rfloor$ und sei bemerkt, dass $(11/17)^\lambda \leq (11/17)^{5 \ln n} \leq 1/n^2$ ist. Wir wollen Lemma 2.D.2 anwenden. Dazu unterscheiden wir drei Typen von Elementen A_i der Partition und bestimmen jeweils p_i^+ und p_i^- .

1. $A_i := \{x \mid |x| = i\}$, $0 \leq i \leq n - \lfloor n/3 \rfloor - 1$. Da $p^-(A_i) = 0$ ist, gilt

$$p_i^- \leq 1/n^2 \text{ für } 0 \leq i \leq n - \lfloor n/3 \rfloor - 1.$$

Weiterhin ist $(1 - p^-(A_i))^\lambda - (1 - p^-(A_i) - p^+(A_i))^\lambda \geq 1 - (1 - p^+(A_i)) \geq \lfloor n/3 \rfloor \cdot (1/n)(1 - 1/n)^{n-1} \geq 1/9$ und somit (Bewegung zum Rand des Kliffs)

$$p_i^+ \geq 1/9 \text{ für } 0 \leq i < n - \lfloor n/3 \rfloor - 1 \text{ sowie } p_{n - \lfloor n/3 \rfloor - 1}^+ \geq 9^{-\lambda}$$

(Überwindung des Kliffs), da $p(A_{n - \lfloor n/3 \rfloor - 1})^\lambda \geq (\lfloor n/3 \rfloor \cdot (1/n)(1 - 1/n)^{n-1})^\lambda \geq (1/9)^\lambda$ ist.

2. $A_{n - \lfloor n/3 \rfloor + i} := \{x \mid n - \lfloor n/3 \rfloor + i\ell \leq |x| < n - \lfloor n/3 \rfloor + (i+1)\ell\}$, $0 \leq i < \lfloor n/(12\ell) \rfloor$. Es gilt

$$p^+(A_{n - \lfloor n/3 \rfloor + i}) \geq \frac{\binom{\lfloor n/3 \rfloor - (i+1)\ell}{\ell}}{en^\ell},$$

da es ausreicht, genau ℓ von $n - |x| \geq \lfloor n/3 \rfloor - (i+1)\ell \geq n/5$ speziellen Bits in $x \in A_{n - \lfloor n/3 \rfloor + i}$ zu flippen. Somit ist weiter

$$\frac{\binom{\lfloor n/3 \rfloor - (i+1)\ell}{\ell}}{en^\ell} \geq \frac{\binom{n/5}{\ell}}{en^\ell} = e^{-\ell \ln(\ell \cdot 5) - 1} \geq e^{-\frac{\ln \lambda}{\ln \ln \lambda} \cdot (\ln \ln \lambda - \ln 5) - 1} \geq \frac{1}{\lambda},$$

wobei die letzte Ungleichung $(\ln \lambda / \ln \ln \lambda) \cdot (\ln \ln \lambda - \ln 5) - 1 \geq 0$ ausnutzt. Wir erhalten neben (Zurückkehren zum Kliff wahrscheinlich)

$$p_{n - \lfloor n/3 \rfloor + i}^- \leq 1 - p_{n - \lfloor n/3 \rfloor + i}^+ \text{ für } 0 \leq i \leq 1$$

auch (Entfernen vom Kliff unwahrscheinlich)

$$\begin{aligned} p_{n - \lfloor n/3 \rfloor + i}^+ &\geq (1 - p^-(A_{n - \lfloor n/3 \rfloor + 1}))^\lambda - (1 - p^-(A_{n - \lfloor n/3 \rfloor + 1}) - p^+(A_{n - \lfloor n/3 \rfloor + 1}))^\lambda \\ &= \sum_{j=1}^{\lambda} \binom{\lambda}{j} p^+(A_{n - \lfloor n/3 \rfloor + i})^j \cdot (1 - p^-(A_{n - \lfloor n/3 \rfloor + i}) - p^+(A_{n - \lfloor n/3 \rfloor + i}))^{\lambda - j} \\ &\geq p^+(A_{n - \lfloor n/3 \rfloor + i}) \cdot (1 - p^-(A_{n - \lfloor n/3 \rfloor + i}) - p^+(A_{n - \lfloor n/3 \rfloor + i}))^{\lambda - 1} \\ &\geq (6/17)^\lambda / \lambda \text{ für } 0 \leq i \leq 1, \end{aligned}$$

wobei die letzte Ungleichung ausnutzt, dass sowohl $p^+(A_{n - \lfloor n/3 \rfloor + i}) \geq 1/\lambda$ als auch $p^-(A_{n - \lfloor n/3 \rfloor + i}) + p^+(A_{n - \lfloor n/3 \rfloor + i}) \leq 11/17$ ist. Betrachten wir $2 \leq i < \lfloor n/(12\ell) \rfloor$, dann gilt

$$p^-(A_{n - \lfloor n/3 \rfloor + i}) \leq 1/(\ell i)! \leq (e/(\ell i))^{\ell i} \leq e^{-(1 + \frac{3 \ln \ln \lambda}{4}) \cdot (\frac{\ln \lambda}{\ln \ln \lambda} - 1) \cdot i} \leq 1/\lambda^{3i/4},$$

da es notwendig ist, mindestens ℓi (irgendwelche) Bits in $x \in A_{n-\lfloor n/3 \rfloor+i}$ zu flippen, und wobei die vorletzte Ungleichung $e/(\ell i) \leq 1/(e \ln^{3/4} \lambda)$ sowie die letzte Ungleichung

$$-\frac{\ln \lambda}{\ln \ln \lambda} + 1 - \frac{3 \ln \lambda}{4} + \frac{3 \ln \ln \lambda}{4} \leq \frac{-3 \ln \lambda}{4}$$

ausnutzt. Somit ist

$$(1 - p^-(A_{n-\lfloor n/3 \rfloor+i}))^\lambda \geq (1 - 1/\lambda^{3i/4})^\lambda \geq 1 - \lambda^{1-3i/4}$$

und wir erhalten neben (Zurückkehren zum Kliff unwahrscheinlich)

$$p_{n-\lfloor n/3 \rfloor+i}^- \leq \lambda^{1-3i/4} + 1/n^2 \text{ für } 2 \leq i < \lfloor n/(12\ell) \rfloor$$

auch (Entfernen vom Kliff wahrscheinlich)

$$p_{n-\lfloor n/3 \rfloor+i}^+ \geq (1 - \lambda^{1-3i/4}) - (1 - 1/\lambda)^\lambda \geq 1/2 \text{ für } 2 \leq i < \lfloor n/(12\ell) \rfloor.$$

3. $A_{n-\lfloor n/3 \rfloor+\lfloor n/(12\ell) \rfloor+i} := \{x \mid x = n - \lfloor n/3 \rfloor + \lfloor n/(12\ell) \rfloor \ell + i\}$, $0 \leq i \leq \lfloor n/3 \rfloor - \lfloor n/(12\ell) \rfloor \ell$.
Es gilt

$$\begin{aligned} (1 - p^-(A_{n-\lfloor n/3 \rfloor+\lfloor n/(12\ell) \rfloor+i}))^\lambda &\geq \left(1 - \frac{1}{\lfloor n/13 \rfloor!}\right)^\lambda \geq 1 - \lambda \cdot \left(\frac{e}{\lfloor n/13 \rfloor}\right)^{\lfloor n/13 \rfloor} \\ &\geq 1 - e^{-n}, \end{aligned}$$

da mindestens $\lfloor n/(12\ell) \rfloor \ell \geq \lfloor n/13 \rfloor$ (irgendwelche) Bits flippen müssen. Wir erhalten (Zurückkehren zum Kliff unwahrscheinlich)

$$p_{n-\lfloor n/3 \rfloor+\lfloor n/(12\ell) \rfloor+i}^- \leq e^{-n} + 1/n^2 \text{ für } 0 \leq i \leq \lfloor n/3 \rfloor - \lfloor n/(12\ell) \rfloor \ell$$

und mit $p^+(A_{n-\lfloor n/3 \rfloor+\lfloor n/(12\ell) \rfloor+i}) \geq (1/n)(1 - 1/n)^{n-1} \geq 1/(en)$ für $i < \lfloor n/3 \rfloor - \lfloor n/(12\ell) \rfloor \ell$ auch (Entfernen vom Kliff wahrscheinlich)

$$\begin{aligned} p_{n-\lfloor n/3 \rfloor+\lfloor n/(12\ell) \rfloor+i}^+ &\geq (1 - p^-(A_{n-\lfloor n/3 \rfloor+\lfloor n/(12\ell) \rfloor+i}))^\lambda - (1 - p^+(A_{n-\lfloor n/3 \rfloor+\lfloor n/(12\ell) \rfloor+i}))^\lambda \\ &\geq (1 - e^{-n}) - (1 - (1/n)(1 - 1/n)^{n-1})^\lambda \\ &\geq 1/(3n) \text{ für } 0 \leq i < \lfloor n/3 \rfloor - \lfloor n/(12\ell) \rfloor \ell. \end{aligned}$$

Wie oben dargelegt, liefert eine Anwendung des Lemmas 2.D.2 eine erwartete Laufzeit des $(1, \lambda)$ -EA zur Optimierung von CLIFF von höchstens

$$\begin{aligned} &\left(1 + \lambda \cdot \left(\sum_{i=0}^{n-\lfloor n/3 \rfloor-2} 9\right) + n^2 + 1 + 1 + \left(\sum_{i=2}^{\lfloor n/(12\ell) \rfloor-1} 2\right) + \left(\sum_{i=0}^{\lfloor n/3 \rfloor - \lfloor n/(12\ell) \rfloor \ell} 3n\right)\right) \\ &\cdot \prod_{i=0}^{n-\lfloor n/3 \rfloor-2} \left(\frac{\frac{1}{9} + \frac{1}{n^2}}{\frac{1}{9}}\right) \cdot \frac{\frac{1}{9^\lambda} + \frac{1}{n^2}}{\frac{1}{9^\lambda}} \end{aligned} \quad (1)$$

$$\cdot \frac{1}{\frac{(6/17)^\lambda}{\lambda}} \cdot \frac{1}{\frac{(6/17)^\lambda}{\lambda}} \cdot \prod_{i=2}^{\lfloor n/(12\ell) \rfloor-1} \left(\frac{\frac{1}{2} + (\lambda^{1-3i/4} + \frac{1}{n^2})}{\frac{1}{2}}\right) \quad (2)$$

$$\cdot \prod_{i=0}^{\lfloor n/3 \rfloor - \lfloor n/(12\ell) \rfloor \ell} \left(\frac{\frac{1}{3n} + (\frac{1}{e^n} + \frac{1}{n^2})}{\frac{1}{3n}}\right). \quad (3)$$

Die Nummer zur Rechten identifiziert den betrachteten Typ von Element der Partition. Der Ausdruck in der ersten Zeile ist $\mathcal{O}(n^2\lambda)$. Der Ausdruck (1) ist $\mathcal{O}(1) \cdot \mathcal{O}(9^\lambda/n^2) = \mathcal{O}(e^{9\lambda/4}/n^2)$, da $\ln 9 \leq 9/4$ ist, und der Ausdruck (2) ist $(17/6)^\lambda \lambda \cdot (17/6)^\lambda \lambda \cdot \mathcal{O}(e^{\lambda/4}/\lambda^3) = \mathcal{O}(e^{11\lambda/4}/\lambda)$, da $\ln(17/6) \leq 5/4$ gilt und

$$\begin{aligned} \prod_{i=2}^{\lfloor n/(12\ell) \rfloor - 1} (1 + 2\lambda^{1-3i/4} + 2/n^2) &\leq \prod_{i=2}^{\lfloor \ln n \rfloor} 2 \cdot \prod_{i=\lfloor \ln n \rfloor + 1}^{\lfloor n/(12\ell) \rfloor - 1} (1 + 3/n^2) \\ &\leq n \cdot 2 = \mathcal{O}(e^{\lambda/4}/\lambda^3). \end{aligned}$$

Der Ausdruck (3) ist $\mathcal{O}(1)$. Insgesamt genügt dem $(1, \lambda)$ -EA eine erwartete Laufzeit von $\mathcal{O}(n^2\lambda) \cdot \mathcal{O}(e^{9\lambda/4}/n^2) \cdot \mathcal{O}(e^{11\lambda/4}/\lambda) \cdot \mathcal{O}(1) = \mathcal{O}(e^{5\lambda})$ zur Optimierung von CLIFF.

- c) Nach Aussage a) beträgt die Wahrscheinlichkeit $1 - 2^{-\Omega(n)}$, dass der $(1+\lambda)$ -EA mehr als $\min\{n^{n/4}, e^{\lambda/4}\}$ Funktionsauswertungen zur Optimierung von CLIFF benötigt.

Falls $\min\{n^{n/4}, e^{\lambda/4}\} \leq 1 + \lambda$ ist, ist die Wahrscheinlichkeit nach Lemma 2.B.2.a ebenfalls $1 - 2^{-\Omega(n)} \geq 1/3$, dass der $(1, \lambda)$ -EA eine Laufzeit von mehr als $\min\{n^{n/4}, e^{\lambda/4}\}$ zur Optimierung von CLIFF benötigt.

Falls $\min\{n^{n/4}, e^{\lambda/4}\} > 1 + \lambda$ ist, ist die Wahrscheinlichkeit nach Lemma 2.B.2.b mit $c(n) = 3/5$ mindestens $(1 - 2^{-\Omega(n)}) - 1/2^{3/5} \geq 1/3$, dass der $(1, \lambda)$ -EA eine Laufzeit von mehr als $\min\{n^{n/4}, e^{\lambda/4}\}$ zur Optimierung von CLIFF benötigt, da $2 \leq t(n) \leq e^{\lambda/4}$ und $\lambda \geq (5/2) \cdot (1 + 3/5) \ln t(n)$ ist.

Insgesamt benötigt der $(1, \lambda)$ -EA eine erwartete Laufzeit von mehr als $\min\{n^{n/4}, e^{\lambda/4}\}/3$ zur Optimierung von CLIFF.

- d) Diese Aussage folgt direkt nach Theorem 2.A.1, da CLIFF ein eindeutiges Optimum hat.

Das gewünschte Resultat ist somit gezeigt. \square

3 ELTERNPOPULATIONEN IN EVOLUTIONÄREN ALGORITHMEN

Seit der Einführung populationsbasierter evolutionärer Algorithmen gibt es viele Debatten darüber, welche Elternpopulationsgröße und – damit verbunden – welche Methode zur Diversitätserhaltung (unter welchen Umständen) zu favorisieren ist (vergleiche Schwefel, 1995). Falls die Population oder die Diversität zu klein ist, neigt ein evolutionärer Algorithmus zur schnellen Stagnation in lokalen Optima. Andererseits neigt er zur schleppenden Bewegung zu globalen Optima, falls die Population oder die Diversität zu groß ist. Viele Strategien zur Bewältigung dieser Schwierigkeiten wurden entwickelt. Dazu gehört die Nischenbildung und der Multistart, aber auch viele weitere Techniken. Wir werden den im Folgenden definierten einfachen evolutionären Algorithmus betrachten, der eine natürliche und schwache Methode der Diversitätserhaltung verwendet. Er vermeidet Duplikate in der Population, welche somit durch eine Menge von Suchpunkten beschrieben wird. Eine spezielle Art der Nischenbildung findet also Anwendung.

$(\mu+1)$ -EA

1. Setze $t := 1$ und wähle $x_{t,1}, \dots, x_{t,\mu} \in \{0, 1\}^n$ zufällig gleichverteilt, aber paarweise verschieden. Setze $\mathcal{P}_t := \{x_{t,1}, \dots, x_{t,\mu}\}$.
2. Setze $t := t + 1$, wähle $x_{t-1} \in \mathcal{P}_{t-1}$ zufällig gleichverteilt und setze $y_t := \text{mutate}_{1/n}(x_{t-1})$.
3. Wähle $z_t \in \arg \min\{f(x) \mid x \in \mathcal{P}_{t-1} \cup \{y_t\}\}$ zufällig gleichverteilt. Falls $y_t \notin \mathcal{P}_{t-1}$, dann setze $\mathcal{P}_t := (\mathcal{P}_{t-1} \cup \{y_t\}) \setminus \{z_t\}$ und sonst $\mathcal{P}_t := \mathcal{P}_{t-1}$.
4. Gehe zu 2.

Wir betrachten die Maximierung und bemerken, dass der $(\mu+1)$ -EA mit $\mu = 1$ nicht dem $(1+1)$ -EA aus Abschnitt 1.A entspricht. Dieser $(1+1)$ -EA setzt den Suchpunkt x_t mit Wahrscheinlichkeit 1 auf y_t , wenn $f(x_{t-1}) = f(y_t)$ ist, während der $(\mu+1)$ -EA mit $\mu = 1$ dies höchstens mit Wahrscheinlichkeit $1/2$ tut.

Für die weiteren Untersuchungen dieses Kapitels ist es unerheblich, welcher der schlechtesten Suchpunkte aus der Population – zusammen mit dem Nachkommen – entfernt wird. Ein Individuum der Population wird somit nie durch ein Element mit geringerem Funktionswert ersetzt. Nur Elternpopulationsgrößen $\mu \leq 2^n$ sind möglich.

Dieser evolutionäre Algorithmus erlaubt eine Analyse der Effekte der Wahl von μ sowie der Diversitätserhaltung und vermeidet zugleich Einflüsse weiterer Komponenten solcher Heuristiken.

Wir beginnen unsere Betrachtungen mit einem Vergleich des $(\mu+1)$ -EA bei kleinst- und größtmöglicher Population. Einerseits optimiert der $(1+1)$ -EA in einer erwarteten Laufzeit von $\mathcal{O}(n^n)$ jede Funktion und es sind auch Funktionen bekannt, zu deren Optimierung der $(1+1)$ -EA eine erwartete Laufzeit von $\Theta(n^n)$ benötigt. Andererseits optimiert der $(1+1)$ -EA in einer erwarteten Laufzeit von $\mathcal{O}(n \ln n)$ eine beliebige lineare Funktion, und besitzt diese ein eindeutiges Optimum, so braucht er auch eine erwartete Laufzeit von $\Theta(n \ln n)$. Der (2^n+1) -EA hingegen optimiert zwar jede Funktion in einer erwarteten Laufzeit von $\mathcal{O}(2^n)$, jedoch benötigt er auch eine erwartete Laufzeit von $\Theta(2^n)$ für jede Funktion mit eindeutigem Optimum. Betrachten wir eine beliebige Funktion f mit eindeutigem Optimum, so benötigt der $(\mu+1)$ -EA zu ihrer Optimierung mit Wahrscheinlichkeit $1 - \binom{2^n-1}{\mu-1} / \binom{2^n}{\mu} = 1 - \mu/2^n$ mehr als μ Schritte. Also erlauben nur Elternpopulationsgrößen $\mu \leq \text{poly}(n)$ die mögliche effiziente Optimierung von f .

Wir betrachten zwei weitere einfache populationsbasierte evolutionäre Algorithmen.

Um die Effekte des Vermeidens von Duplikaten in der Population zu verdeutlichen, vergleichen wir unsere Resultate für den $(\mu+1)$ -EA mit denen des $(\mu+1)$ -EA*, der Duplikate in der Population zulässt. Seine Population wird somit durch eine Multimenge von Suchpunkten beschrieben. Witt (2006) hat den $(\mu+1)$ -EA* ausführlich analysiert, worauf wir an geeigneten Stellen näher eingehen.

$(\mu+1)$ -EA*

1. Setze $t := 1$ und wähle $x_{t,1}, \dots, x_{t,\mu} \in \{0, 1\}^n$ zufällig gleichverteilt. Setze $\mathcal{P}_t := \{x_{t,1}, \dots, x_{t,\mu}\}$.
2. Setze $t := t + 1$, wähle $x_{t-1} \in \mathcal{P}_{t-1}$ zufällig gleichverteilt und setze $y_t := \text{mutate}_{1/n}(x_{t-1})$.
3. Wähle $z_t \in \arg \min\{f(x) \mid x \in \mathcal{P}_{t-1} \cup \{y_t\}\}$ zufällig gleichverteilt. Setze $\mathcal{P}_t := (\mathcal{P}_{t-1} \cup \{y_t\}) \setminus \{z_t\}$.
4. Gehe zu 2.

Um die Effekte des Vermeidens von Duplikaten in der Population weiter zu verdeutlichen, vergleichen wir unsere Resultate für den $(\mu+1)$ -EA mit denen des $\mu \times (1+1)$ -EA, der mit dem μ -maligen Multistart des $(\mu+1)$ -EA mit $\mu = 1$ eine andere Technik zur Diversitätserhaltung verwendet. Seine Population wird durch eine Multimenge von Suchpunkten beschrieben. Droste et al. (2002) sowie Jansen und Wegener (2001a) haben den $\mu \times (1+1)$ -EA ausführlich analysiert, worauf wir an geeigneten Stellen näher eingehen.

$\mu \times (1+1)$ -EA

1. Setze $t := 1$ und wähle $x_{t,1}, \dots, x_{t,\mu} \in \{0, 1\}^n$ zufällig gleichverteilt. Setze $\mathcal{P}_t := \{x_{t,1}, \dots, x_{t,\mu}\}$.
2. Setze $t := t + 1$ und $y_{t,1} := \text{mutate}_{1/n}(x_{t-1,1}), \dots, y_{t,\mu} := \text{mutate}_{1/n}(x_{t-1,\mu})$.
3. Wähle $z_{t,1} \in \arg \min\{f(x_{t-1,1}), f(y_{t,1})\}, \dots, z_{t,\mu} \in \arg \min\{f(x_{t-1,\mu}), f(y_{t,\mu})\}$ jeweils zufällig gleichverteilt. Setze $\mathcal{P}_t := (\mathcal{P}_{t-1} \cup \{y_{t,1}, \dots, y_{t,\mu}\}) \setminus \{z_{t,1}, \dots, z_{t,\mu}\}$.
4. Gehe zu 2.

Für $\mu = 1$ sind der $(\mu+1)$ -EA, der $(\mu+1)$ -EA* und der $\mu \times (1+1)$ -EA die gleichen Optimierer.

3.A Typische Zielfunktion

Betrachten wir die typische Funktion ONEMAX. Analog zu den Analysen von Droste et al. (2002) folgt, dass auch der $(\mu+1)$ -EA mit $\mu = 1$ typischerweise eine Laufzeit von $\Theta(n \ln n)$ zur Optimierung der Funktion ONEMAX benötigt. Genauer resultiert eine erwartete Laufzeit von $\Theta(\mu n \ln n)$ des $\mu \times (1+1)$ -EA zur Optimierung von ONEMAX, falls $\mu \leq \text{poly}(n)$ ist. Witt (2006) hat eine erwartete Laufzeit von $\Theta(n \ln n + \mu n)$ des $(\mu+1)$ -EA* zur Optimierung von ONEMAX nachgewiesen, falls $\mu \leq \text{poly}(n)$ ist. Durch das Erzeugen von Duplikaten existieren schnell viele beste Suchpunkte (Kopien) in der Population und es ist wahrscheinlicher, einen besseren Suchpunkt zu generieren. Der $(\mu+1)$ -EA hingegen erzeugt schnell viele zweitbeste Suchpunkte (nicht Kopien). Im Anschluss werden schnell einige beste Suchpunkte (nicht Kopien) in der Population erzeugt. Und es ist wahrscheinlicher, einen besseren Suchpunkt zu generieren.

Theorem 3.A.1 Sei $\mu \leq \text{poly}(n)$. Der $(\mu+1)$ -EA erzeugt das Optimum von ONEMAX in einer erwarteten Laufzeit von $\Theta(\sqrt{\mu} n \ln n + \mu n)$.

Gegenüber dem $\mu \times (1+1)$ -EA ist der $(\mu+1)$ -EA um einen Faktor $\mathcal{O}(\ln n)$ (und $\Omega(1)$) schneller. Mit $\mu = \Omega(\ln n)$ wird ein Faktor $\Theta(\ln n)$ angenommen. Gegenüber dem $(\mu+1)$ -EA* ist der $(\mu+1)$ -EA jedoch um einen Faktor $\mathcal{O}(\ln^{1/2} n)$ (und $\Omega(1)$) langsamer. Mit $\mu = \Theta(\ln n)$ wird ein Faktor $\Theta(\ln^{1/2} n)$ angenommen, wohingegen für $\mu = \Omega(\ln^2 n)$ die erwartete Laufzeit des $(\mu+1)$ -EA und des $(\mu+1)$ -EA* (asymptotisch) gleich sind.

Beweis (Theorem 3.A.1). Die untere Schranke folgt direkt nach Theorem 3.A.4, da ONEMAX ein eindeutiges Optimum hat. Im Folgenden sei n groß genug, was für den Beweis der Aussage angenommen werden darf, und wir betrachten die obere Schranke.

Für die Population \mathcal{P} sei $M_i := \{x_{(1),i}, \dots, x_{(|M_i|),i}\} = \{x \mid x \in \mathcal{P}, |x| = n - i\} \subseteq \mathcal{P}$, $0 \leq i \leq n$, und sei $m := \min\{i \mid M_i \neq \emptyset\}$. Die Population \mathcal{P} enthält mindestens ein Element mit m Nullen, aber keines mit weniger Nullen. Die Anzahl durchgeführter Funktionsauswertungen des $(\mu+1)$ -EA bis zum einschließlich $t(n)$ -ten Schritt mit $t(n) \geq 1$ beträgt $\mu + (t(n) - 1)$. Wir betrachten die erwartete Laufzeit, bis ein Suchpunkt mit weniger als m Nullen erzeugt wird, und unterscheiden die Fälle $m \geq n/4$ (Fall 1), $n/4 > m \geq 4$ (Fall 2) und $m < 4$ (Fall 3).

1. Die Wahrscheinlichkeit, ein Element mit höchstens $m - 1$ Nullen zu erzeugen, beträgt mindestens $(|M_m|/\mu) \cdot m \cdot (1/n)(1 - 1/n)^{n-1} \geq 1/(4e\mu)$, wobei die Ungleichung $|M_m| \geq 1$ und $m \geq n/4$ ausnutzt. In einer erwarteten Laufzeit von $\mathcal{O}(\mu)$ wird ein Element mit höchstens $m - 1$ Nullen erzeugt und in die Population genommen.
2. Solange $|M_m| + |M_{m+1}| < \min\{\mu, n^{2/3}/m^{2/3}\}$ ist, gilt das Folgende. Die Wahrscheinlichkeit, ein nicht in der Population enthaltenes Element x mit $n - |x| = m + 1$ zu erzeugen, beträgt mindestens $(|M_m|/\mu) \cdot (n - m - |M_{m+1}|) \cdot (1/n)(1 - 1/n)^{n-1} \geq 1/(2e\mu)$, wobei die Ungleichung $(n - m - |M_{m+1}|) \geq (n - m - n^{2/3}/m^{2/3}) \geq n/2$ ausnutzt. In einer

erwarteten Laufzeit von $\mathcal{O}(\mu) \cdot n^{2/3}/m^{2/3} = \mathcal{O}(\mu n^{2/3}/m^{2/3})$ wird entweder ein Individuum mit höchstens $m - 1$ Nullen erzeugt und in die Population genommen oder es gilt $|M_m| + |M_{m+1}| \geq \min\{\mu, n^{2/3}/m^{2/3}\}$.

Danach und solange $|M_m| < \min\{\mu, n^{2/3}/m^{2/3}\}^{1/2}$ sowie $|M_0| = \dots = |M_{m-1}| = 0$ ist, gilt das Folgende. Es ist stets $|M_{m+1}| \geq \min\{\mu, n^{2/3}/m^{2/3}\} - \min\{\mu, n^{2/3}/m^{2/3}\}^{1/2}$ und für jedes $x_{(i),m+1}$ existieren $m + 1$ spezielle 1-Bit-Mutationen, die ein x mit $n - |x| = m$ erzeugen. Die gesamte Anzahl berücksichtigter Mutationen beträgt $(m + 1)|M_{m+1}|$, von denen jede mit Wahrscheinlichkeit $(1/\mu) \cdot (1/n)(1 - 1/n)^{n-1} \geq 1/(e\mu n)$ eintritt. Sei $m_{(i)}$ die Anzahl solcher 1-Bit-Mutationen, die $x_{(i),m}$ generieren. Um $\sum_{i=1}^{|M_m|} m_{(i)} \leq (m + 1)|M_{m+1}|/2$ zu zeigen, nehmen wir $\sum_{i=1}^{|M_m|} m_{(i)} > (m + 1)|M_{m+1}|/2$ an und führen dies zum Widerspruch dazu, dass nur $(m + 1)|M_{m+1}|$ Mutationen betrachtet wurden. Wir bemerken, dass zwei unterschiedliche Individuen höchstens zwei gemeinsame Suchpunkte besitzen, die sie durch 1-Bit-Mutationen erzeugen, und diese beiden gemeinsamen Suchpunkte bestehen aus verschieden vielen Nullen. Also ist $x_{(i),m}$ das einzige Element mit m Nullen, welches die $m_{(i)}$ vielen aus $m + 1$ Nullen bestehenden Individuen der Population durch 1-Bit-Mutationen erzeugen. Des Weiteren ist $x_{(i),m}$ damit für mindestens $m_{(i)} - (|M_m| - 1)$ dieser $m_{(i)}$ Suchpunkte das einzige Element der Population, das durch eine betrachtete Mutation generiert wird. Die jeweils m weiteren betrachteten Mutationen, also zusammen $m \cdot (m_{(i)} - |M_m| + 1)$ viele, erzeugen keinen zur Population gehörenden Suchpunkt. Diese Mutationen werden nur in Verbindung mit $x_{(i),m}$ betrachtet. Insgesamt führt dies zu mindestens $\sum_{i=1}^{|M_m|} m \cdot (m_{(i)} - |M_m| + 1) = m \sum_{i=1}^{|M_m|} m_{(i)} - m \cdot (|M_m|^2 - |M_m|) > m \cdot (m + 1)|M_{m+1}|/2 - m \cdot |M_m|$ betrachteten Mutationen, die kein zur Population gehörendes Element erzeugen. Dabei nutzt die letzte Ungleichung die Voraussetzung und $|M_m|^2 < |M_{m+1}| - |M_m|$ aus. Für $m \geq 4$ jedoch gilt $m \cdot (m + 1)|M_{m+1}|/2 - m \cdot |M_m| > (m + 1)|M_{m+1}|$ und dies führt zum besagten Widerspruch. Mindestens mit Wahrscheinlichkeit $((m + 1)|M_{m+1}| - (m + 1)|M_{m+1}|/2) \cdot 1/(e\mu n) \geq m|M_{m+1}|/(2e\mu n)$ wird ein Individuum, das aus höchstens m Nullen besteht, generiert und in die Population aufgenommen. Dies führt zu einer erwarteten Laufzeit von höchstens $(2e\mu n/(m|M_{m+1}|)) \cdot \min\{\mu, n^{2/3}/m^{2/3}\}^{1/2} = \mathcal{O}(\mu n/(m|M_{m+1}|^{1/2})) = \mathcal{O}(\sqrt{\mu n}/m + \mu n^{2/3}/m^{2/3})$, bis entweder ein Individuum bestehend aus höchstens $m - 1$ Nullen erzeugt und in die Population aufgenommen wird oder $|M_m| \geq \min\{\mu, n^{2/3}/m^{2/3}\}^{1/2} = \mathcal{O}(|M_{m+1}|^{1/2})$ gilt.

Danach und solange $|M_0| = \dots = |M_{m-1}| = 0$ ist, beträgt die Wahrscheinlichkeit mindestens $(\min\{\mu, n^{2/3}/m^{2/3}\}^{1/2}/\mu)m(1/n)(1 - 1/n)^{n-1} \geq \min\{\mu, n^{2/3}/m^{2/3}\}^{1/2} \cdot m/(e\mu n)$, bis ein Individuum bestehend aus höchstens $m - 1$ Nullen erzeugt und in die Population aufgenommen wird. Es folgt eine erwartete Laufzeit von $\mathcal{O}(\sqrt{\mu n}/m + \mu n^{2/3}/m^{2/3})$.

Eine erwartete Laufzeit von insgesamt $\mathcal{O}(\mu n^{2/3}/m^{2/3}) + \mathcal{O}(\sqrt{\mu n}/m + \mu n^{2/3}/m^{2/3}) + \mathcal{O}(\sqrt{\mu n}/m + \mu n^{2/3}/m^{2/3}) = \mathcal{O}(\sqrt{\mu n}/m + \mu n^{2/3}/m^{2/3})$ resultiert, um ein Element bestehend aus höchstens $m - 1$ Nullen zu erzeugen und in die Population aufzunehmen.

3. Die Wahrscheinlichkeit, ein Element mit höchstens $m - 1$ Nullen zu erzeugen, ist mindestens $(|M_m|/\mu) \cdot m \cdot (1/n)(1 - 1/n)^{n-1} \geq 1/(e\mu n)$. In einer erwarteten Laufzeit von $\mathcal{O}(\mu n)$ wird ein solches Element erzeugt und in die Population aufgenommen.

Der $(\mu+1)$ -EA optimiert ONEMAX in einer erwarteten Laufzeit von

$$\sum_{m=1}^3 \mathcal{O}(\mu n) + \sum_{m=4}^{\lceil n/4 \rceil - 1} \mathcal{O}(\sqrt{\mu} n/m + \mu n^{2/3}/m^{2/3}) + \sum_{m=\lceil n/4 \rceil}^n \mathcal{O}(\mu) = \mathcal{O}(\sqrt{\mu} n \ln n + \mu n),$$

da $\sum_{m=1}^n 1/m \leq 1 + \int_1^n x^{-1} dx = \ln n + 1$ und $\sum_{m=1}^n 1/m^{2/3} \leq 1 + \int_1^n x^{-2/3} dx = 3n^{1/3} - 2$ gilt. \square

Betrachten wir die Optimierung einer beliebigen Funktion f mit eindeutigem Optimum. Analog zu den Analysen von Droste et al. (2002) folgt, dass auch der $(\mu+1)$ -EA mit $\mu = 1$ typischerweise eine Laufzeit von $\Omega(n \ln n)$ dazu benötigt. Genauer resultiert eine erwartete Laufzeit von $\Omega(\mu n \ln n)$ des $\mu \times (1+1)$ -EA, falls $\mu \leq \text{poly}(n)$ ist.

Eine Reihe populationsbasierter randomisierter Suchheuristiken kann als Spezifizierung des folgenden μ -EA aufgefasst werden.

μ -EA

1. Wähle $x_1, \dots, x_\mu \in \{0, 1\}^n$ beliebig. Setze $t := \mu$.
2. Setze $t := t + 1$, wähle $y_{t-1} = x_i$ mit $1 \leq i \leq t - 1$ beliebig, aber höchstens mit Wahrscheinlichkeit $1/\mu$ und setze $x_t := \text{mutate}_{1/n}(y_{t-1})$.
3. Gehe zu 2.

Das folgende Resultat ergibt sich analog zu den Betrachtungen von Witt (2006).

Lemma 3.A.2 Sei $f : \{0, 1\}^n \rightarrow \mathbb{R}$ eine beliebige Funktion mit eindeutigem Optimum $x^* \in \{0, 1\}^n$ und sei $\mu \leq \text{poly}(n)$. Jeder μ -EA mit $H[x_1, x^*], \dots, H[x_\mu, x^*] = \Omega(n)$ benötigt eine erwartete Laufzeit von $\Omega(n \ln n + \mu n)$ zur Optimierung von f .

Damit hat Witt (2006) eine erwartete Laufzeit von $\Omega(n \ln n + \mu n)$ nicht nur für den $(\mu+1)$ -EA* mit $\mu \leq \text{poly}(n)$ gezeigt, sondern für beliebige μ -EA mit $\mu \leq \text{poly}(n)$ – wozu auch der $(\mu+1)$ -EA gehört – solange die ersten μ Individuen mit konstanter, positiver Wahrscheinlichkeit einen linearen Hammingabstand zum Optimum haben. Wir weisen für den $(\mu+1)$ -EA sogar eine erwartete Laufzeit von $\Omega(\sqrt{\mu} n \ln n + \mu n)$ zur Optimierung von f nach. Zunächst betrachten wir die Population des $(\mu+1)$ -EA nach dem ersten Schritt detaillierter.

Lemma 3.A.3 Sei $x^* \in \{0, 1\}^n$ und n groß genug. Für den $(\mu+1)$ -EA gilt mindestens mit Wahrscheinlichkeit $1 - \mu 2^{-n/30}$

$$\lfloor n/3 \rfloor < H[x_{1,1}, x^*], \dots, H[x_{1,\mu}, x^*] < n - \lfloor n/3 \rfloor.$$

Beweis. Die Wahrscheinlichkeit dieses Ereignisses beträgt

$$\binom{\sum_{i=\lfloor n/3 \rfloor + 1}^{n-\lfloor n/3 \rfloor - 1} \binom{n}{i}}{\mu} / \binom{2^n}{\mu}.$$

Im Beweis des Theorems 2.E.1.a haben wir bereits festgestellt, dass für n groß genug gilt $\sum_{i=0}^{\lfloor n/3 \rfloor} \binom{n}{i} = \sum_{i=n-\lfloor n/3 \rfloor}^n \binom{n}{i} \leq 2^{28n/29}$. Insgesamt ist der obige Ausdruck also durch

$$\binom{2^n - 2 \cdot 2^{28n/29}}{\mu} / \binom{2^n}{\mu} \geq ((2^n - 2^{29n/30})^\mu / \mu!) \cdot (\mu! / (2^n)^\mu) = (1 - 2^{-n/30})^\mu \geq 1 - \mu 2^{-n/30}$$

nach unten beschränkt. \square

Theorem 3.A.4 Sei $f : \{0, 1\}^n \rightarrow \mathbb{Z}$ eine beliebige Funktion mit eindeutigem Optimum und sei $\mu \leq \text{poly}(n)$. Der $(\mu+1)$ -EA benötigt eine erwartete Laufzeit von $\Omega(\sqrt{\mu n} \ln n + \mu n)$ zur Optimierung von f .

Beweis. Sei $x^* \in \{0, 1\}^n$ das eindeutige Optimum von f . Im Folgenden sei n groß genug, was für den Beweis der Aussage angenommen werden darf. Mit Wahrscheinlichkeit $1 - 2^{-\Omega(n)}$ gilt $H[x_{1,1}, x^*], \dots, H[x_{1,\mu}, x^*] > \lfloor n/3 \rfloor$ nach Lemma 3.A.3, da $\mu \leq \text{poly}(n)$ ist. Wir nehmen dies an und betrachten $\mu > \ln^2 n$ (Fall 1) und $\mu \leq \ln^2 n$ (Fall 2).

1. Es genügt, eine erwartete Laufzeit von $\Omega(\mu n)$ nachzuweisen. Diese folgt direkt nach Lemma 3.A.2, da $(1 - 2^{-\Omega(n)}) \cdot \Omega(n \ln n + \mu n) = \Omega(\mu n)$ ist.
2. Es genügt, eine erwartete Laufzeit von $\Omega(\sqrt{\mu n} \ln n)$ nachzuweisen. Wir nehmen an, dass $H[x, x^*] \geq m$ ist für alle $x \in \mathcal{P}$. Nach dem ersten Schritt gilt dies für $m = 2 \lfloor n^{1/4} \rfloor \leq \lfloor n/3 \rfloor$. Dann sei \mathcal{E}_m das Ereignis, dass beim erstmaligen Erzeugen eines Suchpunkts x' mit $H[x', x^*] \leq m - 2$ sogar $H[x', x^*] < m - 2$ gilt. Die Wahrscheinlichkeit des Eintretens von \mathcal{E}_m beträgt höchstens em/n wie später gezeigt wird. Mindestens mit Wahrscheinlichkeit $1 - \sum_{i=1}^{\lfloor n^{1/4} \rfloor} e(2i)/n = 1 - \mathcal{O}(n^{-1/2})$ tritt \mathcal{E}_m für kein $m \in \{2i \mid 1 \leq i \leq \lfloor n^{1/4} \rfloor\}$ ein. Wir nehmen dies an. Damit tritt für jedes solches m die Situation ein, dass $H[x, x^*] \geq m$ für alle $x \in \mathcal{P}$ und ein x' mit $H[x', x^*] = m - 2$ erzeugt werden muss. Wie später gezeigt wird beträgt die erwartete Laufzeit $\Omega(\sqrt{\mu n}/m)$, bis dies zum ersten Mal eintritt. Insgesamt folgt eine erwartete Laufzeit $(1 - (2^{-\Omega(n)} + \mathcal{O}(n^{-1/2}))) \cdot \sum_{i=1}^{\lfloor n^{1/4} \rfloor} \Omega(\sqrt{\mu n}/(2i)) = \Omega(\sqrt{\mu n} \ln n)$, da $\sum_{i=1}^{\lfloor n^{1/4} \rfloor} 1/i \geq (\ln n)/4 - 1$ ist.

Wir beschränken die Wahrscheinlichkeit des Eintretens von \mathcal{E}_m durch em/n nach oben. Ein x' mit $H[x', x^*] \leq m - 2$ wird erstmals von einem x mit $H[x, x^*] = m - 1 + i$, $i \geq 0$, generiert. Höchstens mit Wahrscheinlichkeit $\binom{m-1+i}{i+2} \cdot 1/n^{i+2}$ gilt $H[x', x^*] < m - 2$ und mindestens mit Wahrscheinlichkeit $\binom{m-1+i}{i+1} \cdot 1/(en^{i+1})$ gilt $H[x', x^*] = m - 2$. Das Ereignis \mathcal{E}_m tritt höchstens mit Wahrscheinlichkeit $\binom{m-1+i}{i+2} \cdot (en^{i+1}) / (\binom{m-1+i}{i+1} \cdot n^{i+2}) = e(m-1)/((i+2)n) \leq em/n$ ein.

Wir beschränken die erwartete Laufzeit durch $\Omega(\sqrt{\mu n}/m)$, bis erstmals ein x' mit $H[x', x^*] = m - 2$ erzeugt wird, falls $H[x, x^*] \geq m$ ist für alle $x \in \mathcal{P}$. Dazu betrachten wir $\lfloor \sqrt{\mu n}/(3m) \rfloor$ Funktionsauswertungen und zeigen, dass diese mit positiver, konstanter Wahrscheinlichkeit nicht ausreichen, damit das gewünschte Ereignis eintritt. Für jeden Suchpunkt x mit $H[x, x^*] = m + i$, $i \geq 0$, beträgt die Wahrscheinlichkeit, ein x' zu erzeugen, höchstens $\binom{m+i}{i+1} \cdot 1/n^{i+1} \leq m/n$ für $H[x', x^*] = m - 1$ (Fall i) und

$\binom{m+i}{i+2} \cdot 1/n^{i+2} \leq m^2/n^2$ für $H[x', x^*] = m - 2$ (Fall ii). In $\lfloor \sqrt{\mu n}/(3m) \rfloor$ Schritten wird eine erwartete Anzahl solcher Suchpunkte von höchstens $(\sqrt{\mu n}/(3m)) \cdot m/n \leq \sqrt{\mu}/3$ für Fall i und $(\sqrt{\mu n}/(3m)) \cdot m^2/n^2 \leq \sqrt{\mu m}/(3n) \leq 1/n^{1/2}$ für Fall ii erzeugt. Zwei Anwendungen der Markoffungleichung zeigen, dass höchstens mit Wahrscheinlichkeit $1/3 + 1/n^{1/2}$ entweder mehr als $\sqrt{\mu}$ Elemente nach Fall i oder mindestens ein Element nach Fall ii generiert werden. Wir nehmen dies an. Höchstens mit Wahrscheinlichkeit $\sqrt{\mu} \cdot (1/\mu) \cdot m/n \leq m/(\sqrt{\mu n})$ erzeugt dann ein Suchpunkt x mit $H[x, x^*] = m - 1$ ein x' mit $H[x', x^*] = m - 2$ (Fall iii). In $\lfloor \sqrt{\mu n}/(3m) \rfloor$ Schritten wird eine erwartete Anzahl solcher Suchpunkte von höchstens $(\sqrt{\mu n}/(3m)) \cdot m/(\sqrt{\mu n}) = 1/3$ für Fall iii erzeugt. Eine Anwendung der Markoffungleichung zeigt, dass höchstens mit Wahrscheinlichkeit $1/3$ mindestens ein Element nach Fall iii generiert wird. Insgesamt resultiert eine erwarteten Laufzeit von mindestens $(1 - (1/3 + 1/n^{1/2} + 1/3)) \cdot (\lfloor \sqrt{\mu n}/(3m) \rfloor + 1) = \Omega(\sqrt{\mu n}/m)$.

Das gewünschte Resultat ist somit gezeigt. \square

Wir betrachten die weitere typische Funktion $\text{SHORTPATHINCREASE} : \{0, 1\}^n \rightarrow \mathbb{Z}$, die mit 1^n ein eindeutiges Optimum besitzt (vergleiche Jansen und Wegener, 2001a). Sei

$$\text{SHORTPATHINCREASE}(x) := \begin{cases} 3n & , \text{ falls } x = 1^n, \\ n + 2i & , \text{ falls } x = 1^i 0^{n-i} \text{ für } 0 \leq i < n, \\ n - |x| & , \text{ sonst.} \end{cases}$$

Die Suchpunkte $0^n, 10^{n-1}, \dots, 1^{n-1}0, 1^n$ stellen einen *Pfad* (engl. *path*) mit ansteigenden (engl. *increase*) Funktionswerten dar. (Da er höchstens $\text{poly}(n)$ Elemente umfasst, handelt es sich sogar um einen kurzen (engl. *short*) Pfad.)

Lemma 3.A.5 Sei $\mu \leq \text{poly}(n)$. Der $(\mu+1)$ -EA erzeugt das Optimum von SHORTPATHINCREASE in einer erwarteten Laufzeit von $\Theta(\mu n^2)$.

Beweis. Betrachten wir die obere Schranke. Nach Theorem 3.A.1 erzeugt der $(\mu+1)$ -EA in einer erwarteten Laufzeit von $\mathcal{O}(\sqrt{\mu n} \ln n + \mu n)$ einen Suchpunkt $1^i 0^{n-i}$, $0 \leq i \leq n$. Danach und solange das Optimum nicht erzeugt wurde, gilt das Folgende. Eine spezielle 1-Bit-Mutation des Suchpunkts in der Population mit größtem Funktionswert erzeugt ein Individuum mit noch größerem Funktionswert, das in die Population aufgenommen wird. Dies geschieht mindestens mit Wahrscheinlichkeit $1/(e\mu n)$. Nach höchstens n solcher Mutationen wird das Optimum generiert, also höchstens in einer erwarteten Laufzeit von $n \cdot e\mu n = \mathcal{O}(\mu n^2)$. Insgesamt resultiert eine erwartete Laufzeit von $\mathcal{O}(\sqrt{\mu n} \ln n + \mu n) + \mathcal{O}(\mu n^2) = \mathcal{O}(\mu n^2)$.

Betrachten wir die untere Schranke. Nach Lemma 3.A.3 gilt $|x_{1,1}|, \dots, |x_{1,\mu}| < n - \lfloor n/3 \rfloor$ mit Wahrscheinlichkeit $1 - 2^{-\Omega(n)}$, da $\mu \leq \text{poly}(n)$ ist. Wir nehmen dies an. Im Anschluss werden dann nur Individuen x mit $|x| < n - \lfloor n/3 \rfloor$ oder $x = 1^i 0^{n-i}$, $0 \leq i \leq n$, in die Population aufgenommen. Enthält die Population lediglich Suchpunkte x mit $\text{SHORTPATHINCREASE}(x) \leq n + 2i$ für ein $i \geq n - \lfloor n/4 \rfloor$, was nach dem ersten Schritt gilt, erhalten wir das Folgende. Höchstens mit Wahrscheinlichkeit $\sum_{j=0}^i (1/\mu) \cdot 1/n^{j+1} + 1 \cdot 2^{-\Omega(n \ln n)} \leq 2/(\mu n)$ wird ein

Element mit Funktionswert $n + 2(i + 1)$ (Fall 1) und höchstens mit Wahrscheinlichkeit $\sum_{j=0}^i (1/\mu) \cdot 1/n^{j+2} + 1 \cdot 2^{-\Omega(n \ln n)} \leq 2/(\mu n^2)$ ein Element mit Funktionswert mindestens $n + 2(i + 2)$ (Fall 2) erzeugt. Jedes Element $1^{i-j} 0^{n-(i-j)}$, $j \geq 0$, existiert höchstens einmal in der Population und für Fall 1 müssen genau $j + 1$ spezielle Bits flippen, während für Fall 2 mindestens $j + 2$ spezielle Bits flippen müssen. Jedes Element x mit $|x| < n - \lfloor n/3 \rfloor$ muss dazu $\Omega(n)$ (irgendwelche) Bits flippen, was höchstens Wahrscheinlichkeit $1/(\Omega(n)!) = 2^{-\Omega(n \ln n)}$ hat. Wir betrachten $\lfloor \mu n^2/16 \rfloor$ Schritte und zeigen, dass diese mit positiver, konstanter Wahrscheinlichkeit nicht ausreichen, damit das Optimum erzeugt wird. Eine Anwendung der Markoffungleichung zeigt, dass höchstens mit Wahrscheinlichkeit $(2/(\mu n^2)) \cdot \mu n^2/16 = 1/8$ Fall 2 mindestens einmal eintritt. Eine Anwendung der Chernoffungleichung zeigt, dass mit Wahrscheinlichkeit $e^{-\Omega(n)}$ Fall 1 mindestens $n/4$ -mal eintritt. Wir nehmen an, dass beide Ereignisse nicht eintreten. Es werden höchstens Suchpunkte $1^i 0^{n-i}$ mit $i \leq n - \lceil n/4 \rceil + (n/4 - 1) < n$ generiert. Insgesamt resultiert eine erwartete Laufzeit von mindestens $(1 - (2^{-\Omega(n)} + 1/8 + e^{-\Omega(n)})) \cdot (\lfloor \mu n^2/16 \rfloor + 1) = \Omega(\mu n^2)$. \square

An geeigneter Stelle vergleichen wir die erwartete Laufzeit des $(\mu+1)$ -EA zur Optimierung von SHORTPATHINCREASE mit denen des $\mu \times (1+1)$ -EA und des $(\mu+1)$ -EA*.

Das folgende Lemma verdeutlicht eine spezielle Eigenschaft des $(\mu+1)$ -EA, die wir im Anschluss – mit Hilfe von SHORTPATHINCREASE und Varianten dieser Funktion – verdeutlichen. Die Eigenschaft hilft in einigen Situationen, die Laufzeit des $(\mu+1)$ -EA zur Optimierung von einer Funktion auf eine andere zu übertragen.

Lemma 3.A.6 *Für eine beliebige Funktion $f : \{0, 1\}^n \rightarrow \mathbb{Z}$ sei A_1, A_2, A_3 eine f -basierte Partition. Sei $f^* : \{0, 1\}^n \rightarrow \mathbb{Z}$ eine beliebige Funktion mit*

- $f^*(x) = f(x)$ für alle $x \in A_1 \cup A_3$ und
- $\max\{f(z) \mid z \in A_1\} < f^*(x) < \max\{f(z) \mid z \in A_3\}$ für alle $x \in A_2$.

Sei $\mu \geq |A_2|$ und seien $x_{(1)}, \dots, x_{(\mu)} \in \{0, 1\}^n$. Für $\ell(n) \geq 0$ gilt die Gleichung:

$$\begin{aligned} & \Pr[T_{f, (\mu+1)\text{-EA}} > \ell(n) \mid x_{1,1} = x_{(1)}, \dots, x_{1,\mu} = x_{(\mu)}] \\ &= \Pr[T_{f^*, (\mu+1)\text{-EA}} > \ell(n) \mid x_{1,1} = x_{(1)}, \dots, x_{1,\mu} = x_{(\mu)}] \end{aligned}$$

Da der $(\mu+1)$ -EA die ersten μ Individuen unabhängig von der zu optimierenden Funktion erzeugt, folgt dann direkt

$$\mathbb{E}[T_{f, (\mu+1)\text{-EA}}] = \mathbb{E}[T_{f^*, (\mu+1)\text{-EA}}].$$

Beweis (Lemma 3.A.6). Für f und f^* sind die gleichen Elemente optimal. Wir weisen sogar für jede Population \mathcal{P} mit $\mathcal{P} \cap A_3 = \emptyset$ – damit wurde zuvor kein Optimum generiert – nach, dass der $(\mu+1)$ -EA sie mit gleicher Wahrscheinlichkeit nach $\ell(n) \geq 0$ Funktionsauswertungen bei Optimierung von f und f^* erzeugt. Das gewünschte Resultat ist somit gezeigt.

Wir führen diesen Beweis per Induktion über $\ell(n)$.

Sei $0 \leq \ell(n) \leq \mu$. Die Population $\{x_{1,1}, \dots, x_{1,\ell(n)}\}$ wird unabhängig von f und f^* erzeugt. Dies zeigt die Behauptung.

Sei $\ell(n) > \mu$. Die Erzeugung von $y_{\ell(n)-\mu+1}$ ist abhängig vom Elter, nicht aber von f und f^* . Falls $y_{\ell(n)-\mu+1}$ ein Duplikat ist, wird die Population nicht verändert. Falls $y_{\ell(n)-\mu+1} \in A_3$ und damit optimal ist, enthält die nächste Population ein Optimum. In beiden Fällen folgt die Behauptung direkt. Sonst, da $\mu \geq |A_2|$ ist, muss die Population zusammen mit $y_{\ell(n)-\mu+1}$ mindestens ein Element aus A_1 enthalten. Aufgrund der Eigenschaften von f und f^* hat jedes Element aus A_1 einen kleineren Funktionswert als jeder Suchpunkt aus A_2 . Die Wahl von $z_{\ell(n)-\mu+1}$ hängt nur von Elementen aus A_1 ab und diese haben für f und f^* gleiche Funktionswerte. Also wird mit gleicher Wahrscheinlichkeit das gleiche Element aus der Population entfernt. Dies zeigt die Behauptung. \square

Wir betrachten $\text{SHORTPATHCONSTANT} : \{0, 1\}^n \rightarrow \mathbb{Z}$ und $\text{SHORTPATHDECREASE} : \{0, 1\}^n \rightarrow \mathbb{Z}$ – zwei weitere Funktionen, die mit 1^n ein eindeutiges Optimum besitzen. Seien

$$\text{SHORTPATHCONSTANT}(x) := \begin{cases} 3n & , \text{ falls } x = 1^n, \\ 3n - 2 & , \text{ falls } x = 1^i 0^{n-i} \text{ für } 0 \leq i < n, \\ n - |x| & , \text{ sonst,} \end{cases}$$

und

$$\text{SHORTPATHDECREASE}(x) := \begin{cases} 3n & , \text{ falls } x = 1^n, \\ 3n - 2 - 2i & , \text{ falls } x = 1^i 0^{n-i} \text{ für } 0 \leq i < n, \\ n - |x| & , \text{ sonst.} \end{cases}$$

Höchstens in Funktionswerten von Elementen des kurzen Pfads von SHORTPATHINCREASE unterscheiden sich die drei Funktionen. Die Suchpunkte $0^n, 10^{n-1}, \dots, 1^{n-1}0, 1^n$ stellen einen Pfad mit konstanten (engl. constant) Funktionswerten für SHORTPATHCONSTANT dar, während es ein Pfad mit abfallenden (engl. decrease) Funktionswerten für SHORTPATHDECREASE ist.

Theorem 3.A.7 *Sei $n \leq \mu \leq \text{poly}(n)$. Der $(\mu+1)$ -EA erzeugt in gleicher erwarteter Laufzeit das Optimum von SHORTPATHINCREASE , SHORTPATHCONSTANT und SHORTPATHDECREASE . Diese erwartete Laufzeit zur Optimierung beträgt $\Theta(\mu n^2)$.*

Beweis. Sei $A_3 := \{1^n\}$, $A_2 := \{1^i 0^{n-i} \mid 0 \leq i < n\}$ und $A_1 := \{0, 1\}^n \setminus (A_2 \cup A_3)$. Da $\mu \geq |A_2| = n$ ist, können wir Lemma 3.A.6 anwenden und zusammen mit den Resultaten des Lemmas 3.A.5 erhalten wir die gewünschte Aussage. \square

Analog zu den Analysen von Jansen und Wegener (2001a) folgt, dass auch der $(\mu+1)$ -EA mit $\mu = 1$ typischerweise eine Laufzeit von $\Theta(n^2)$ zur Optimierung von SHORTPATHINCREASE benötigt. Genauer resultiert eine erwartete Laufzeit von $\Theta(\mu n^2)$ des $\mu \times (1+1)$ -EA, falls $\mu \leq \text{poly}(n)$ ist. Um das Optimum von SHORTPATHINCREASE zu generieren, benötigt der $(\mu+1)$ -EA also eine (asymptotisch) gleiche erwartete Laufzeit wie der $\mu \times (1+1)$ -EA, jedoch eine um den Faktor $\Omega(n/\ln n)$ größere als der $(\mu+1)$ -EA*, was aus den Betrachtungen

von Witt (2006) folgt. Zur Erzeugung des Optimums von SHORTPATHCONSTANT benötigt der $(\mu+1)$ -EA mit $n \leq \mu \leq \text{poly}(n)$ eine um den Faktor $\Omega(n/\ln n)$ kleinere erwartete Laufzeit als der $\mu \times (1+1)$ -EA und der $(\mu+1)$ -EA*, was aus den Betrachtungen von Jansen und Wegener (2001a) sowie Witt (2006) folgt. Ebenso optimiert der $(\mu+1)$ -EA mit $n \leq \mu \leq \text{poly}(n)$ die Funktion SHORTPATHDECREASE effizient, wohingegen der $\mu \times (1+1)$ -EA und der $(\mu+1)$ -EA* nur mit einer exponentiell kleinen Wahrscheinlichkeit weniger als $n^{n/4}$ Funktionsauswertungen benötigen.

3.B Erweiterungen der Methode der f -basierten Partition

Unsere erste Erweiterung der Methode der f -basierten Partition für den $(1+1)$ -EA nach Lemma 1.B.1 auf den $(\mu+1)$ -EA erlaubt das Vernachlässigen von Bereichen des Suchbereichs auf Kosten der Populationsgröße. Wir nennen A_0, \dots, A_m eine f -basierte Partition in Variante (1) genau dann, wenn $A_0 <_f \dots <_f A_m$ ist und weiterhin A_m nur Optima von f enthält, also für alle $x \in A_m$ gilt $x \in \arg \max\{f(y) \mid y \in \{0, 1\}^n\}$. Die Bedingung, dass A_0, \dots, A_m eine Partition des gesamten $\{0, 1\}^n$ ist, entfällt somit. Sei $A_* := \{0, 1\}^n \setminus (A_0 \cup \dots \cup A_m)$.

Lemma 3.B.1 *Für eine beliebige Funktion $f : \{0, 1\}^n \rightarrow \mathbb{Z}$ sei A_0, \dots, A_m eine f -basierte Partition in Variante (1). Sei $\mu \geq |A_*| + 1$. Der $(\mu+1)$ -EA (mit beliebiger Verteilung von $x_{1,1}, \dots, x_{1,\mu}$) erzeugt ein Optimum von f in einer erwarteten Laufzeit von höchstens*

$$\mu \left(1 + \frac{1}{p(A_0)} + \dots + \frac{1}{p(A_{m-1})} \right).$$

Mit $A_* = \emptyset$ und $\mu = 1$ erhalten wir das gleiche Resultat wie Lemma 1.B.1 und Wegener (2002) für den $(1+1)$ -EA.

Beweis (Lemma 3.B.1). Es existieren immer mindestens $\mu - |A_*| \geq 1$ Suchpunkte in der Population, die nicht aus A_* stammen. Wir betrachten den gleichen Markoffprozess \mathcal{M} wie in Lemma 2.D.1, wobei immer ein Suchpunkt $x \in \mathcal{P}_{t(n)}$ mit $x \in A_i$, aber $\mathcal{P}_{t(n)} \cap (A_{i+1} \cup \dots \cup A_m) = \emptyset$, beobachtet wird. Es ist unmöglich, dass $\mathcal{P}_{t(n)+1} \cap (A_i \cup A_{i+1} \cup \dots \cup A_m) = \emptyset$ gilt – dazu muss ein solches x durch ein Individuum aus $A_0 \cup \dots \cup A_{i-1}$ ersetzt werden. Weiterhin ist $p(A_i)/\mu$ eine untere Schranke der Wahrscheinlichkeit, dass der $(\mu+1)$ -EA eine Population $\mathcal{P}_{t(n)+1}$ mit $\mathcal{P}_{t(n)+1} \cap (A_{i+1} \cup \dots \cup A_m) \neq \emptyset$ generiert. Dazu genügt es, ein solches x zur Mutation auszuwählen und ein Element aus $A_{i+1} \cup \dots \cup A_m$ zu erzeugen. Befindet sich \mathcal{M} nach $t(n)$ Schritten im Zustand i , umfasst die Population $\mathcal{P}_{t(n)}$ des $(\mu+1)$ -EA mit zumindest gleicher Wahrscheinlichkeit ein x mit $x \in A_i \cup \dots \cup A_m$. Mit der Betrachtung für den ersten Schritt und da in jedem weiteren Schritt eine Funktionsauswertung anfällt, beträgt die erwartete Laufzeit des $(\mu+1)$ -EA zur Optimierung von f höchstens $\mu + \mu/p(A_0) + \dots + \mu/p(A_{m-1})$. \square

Unsere zweite Erweiterung der f -basierten Partition für den $(1+1)$ -EA nach Lemma 1.B.1 auf den $(\mu+1)$ -EA relaxiert die Bedingung $A_0 <_f \dots <_f A_m$. Wir nennen A_0, \dots, A_m mit b_0, \dots, b_{m-1} , $b_i \geq i + 1$, eine f -basierte Partition in Variante (2) genau dann, wenn A_0, \dots, A_m

eine Partition des $\{0, 1\}^n$ ist, $A_i <_f A_j$ für alle $j \geq b_i$ und $0 \leq i < m$, und weiterhin A_m nur Optima von f enthält, also für alle $x \in A_m$ gilt $x \in \arg \max\{f(y) \mid y \in \{0, 1\}^n\}$.

Lemma 3.B.2 Für eine beliebige Funktion $f : \{0, 1\}^n \rightarrow \mathbb{Z}$ sei A_0, \dots, A_m mit b_0, \dots, b_{m-1} eine f -basierte Partition in Variante (2). Sei

$$v_i := \sum_{0 \leq j < i, b_j > i} |A_j| \text{ für } 0 \leq i < m.$$

Sei $\mu \geq \max\{v_i \mid 0 \leq i < m\} + 1$. Der $(\mu+1)$ -EA (mit beliebiger Verteilung von $x_{1,1}, \dots, x_{1,\mu}$) erzeugt ein Optimum von f in einer erwarteten Laufzeit von höchstens

$$\mu \left(1 + \frac{1}{p(A_0)} + \dots + \frac{1}{p(A_{m-1})} \right).$$

Mit $b_i = i + 1$ für $0 \leq i < m$ und $\mu = 1$ erhalten wir das gleiche Resultat wie Lemma 1.B.1 und Wegener (2002) für den $(1+1)$ -EA.

Beweis (Lemma 3.B.2). Analog zum Beweis des Lemmas 3.B.1 genügt es, das Folgende zu zeigen. Es ist unmöglich, dass $\mathcal{P}_{t(n)+1} \cap (A_i \cup \dots \cup A_m) = \emptyset$ ist, falls $\mathcal{P}_{t(n)} \cap (A_i \cup \dots \cup A_m) \neq \emptyset$ gilt. Dazu muss $y_{t(n)+1} \in A_0 \cup \dots \cup A_{i-1}$ und $z_{t(n)+1} \in A_i$ gelten. Die einzigen solchen $y_{t(n)+1}$ mit $f(y_{t(n)+1}) \geq f(z_{t(n)+1})$ – dies ist notwendigerweise erfüllt – sind höchstens Elemente aus A_j mit $0 \leq j \leq i-1$ und $b_j > i$, da für $b_j \leq i$ gilt $A_j <_f A_i$. Diese umfassen insgesamt höchstens v_i Suchpunkte. Jedoch ist $\mu \geq \max\{v_i \mid 0 \leq i < m\} + 1 \geq v_i + 1$ und daher kann $\mathcal{P}_{t(n)+1} \cap (A_i \cup \dots \cup A_m) = \emptyset$ nicht auftreten, falls $\mathcal{P}_{t(n)} \cap (A_i \cup \dots \cup A_m) \neq \emptyset$ ist. \square

Wir wenden diese beiden Methoden nun exemplarisch an. Dazu sei $m := \lfloor n/4 \rfloor$ und wir betrachten mit $\text{SHORTPATHONEPEAK} : \{0, 1\}^n \rightarrow \mathbb{Z}$ eine Funktion, die ein eindeutiges Optimum besitzt. Sei

$$\text{SHORTPATHONEPEAK}(x) := \begin{cases} 3n & , \text{ falls } x = 1^n, \\ 3n - 1 & , \text{ falls } x = 1^{n-m}0^m, \\ n + 2i & , \text{ falls } x = 1^i0^{n-i} \text{ für } 0 \leq i < n \text{ und } i \neq n - m, \\ n - |x| & , \text{ sonst.} \end{cases}$$

Lediglich der Funktionswert eines Elements stimmt nicht überein mit dem der Funktion SHORTPATHINCREASE . Der zweitbeste Suchpunkt ist $1^{n-m}0^m$ und hat Hammingabstand m zum globalen Optimum 1^n . Somit stellt er ein lokales Optimum dar, welches auch als *Spitze* (engl. *peak*) bezeichnet wird. Bereits eine Populationsgröße von zwei vermeidet die Stagnation im lokalen Optimum, die bei einer kleineren Population auftritt, während bei größeren Populationen eine schleppende Bewegung zum globalen Optimum beobachtet wird. Präzisieren wir das Ergebnis für den $(\mu+1)$ -EA mit $\mu \geq 2$.

Lemma 3.B.3 Sei $\mu \geq 2$. Der $(\mu+1)$ -EA erzeugt das Optimum von SHORTPATHONEPEAK in einer erwarteten Laufzeit von $\Theta(\mu n^2)$.

Beweis. Betrachten wir die untere Schranke. Diese ergibt sich analog zum Beweis des Lemmas 3.A.5.

Betrachten wir die obere Schranke ...

... mittels Lemma 3.B.1. Sei

$$A_i := \begin{cases} \{x \mid \text{SHORTPATHONEPEAK}(x) = i + 1\} & , \text{ falls } 0 \leq i \leq n - 2, \\ \{1^{i-n+1}0^{2n-1-i}\} & , \text{ falls } n - 1 \leq i \leq 2n - m - 2, \\ \{1^{i-n+2}0^{2n-2-i}\} & , \text{ falls } 2n - m - 1 \leq i \leq 2n - 2. \end{cases}$$

Es gilt $A_* = \{1^{n-m}0^m\}$. Für das Element aus A_{2n-m-2} genügt eine spezielle 2-Bit-Mutation, um das Element aus A_{2n-m-1} zu erzeugen. Jedes Individuum aus A_i , $0 \leq i < 2n - 2$ und $i \neq 2n - m - 2$, generiert mittels spezieller 1-Bit-Mutation ein Element aus A_{i+1} . Damit ist

$$p(A_i) \geq \begin{cases} 1/(en) & , \text{ falls } 0 \leq i < 2n - 2 \text{ und } i \neq 2n - m - 2, \\ 1/(en^2) & , \text{ falls } i = 2n - m - 2. \end{cases}$$

Da $\mu \geq |A_*| + 1 = 2$ ist, liefert eine Anwendung des Lemmas 3.B.1 für den $(\mu+1)$ -EA mit $\mu \geq 2$ eine erwartete Laufzeit von höchstens $\mu(1 + (2n - 3)en + en^2) = \mathcal{O}(\mu n^2)$ zur Optimierung von SHORTPATHONEPEAK.

... mittels Lemma 3.B.2. Sei

$$A_i := \begin{cases} \{x \mid \text{SHORTPATHONEPEAK}(x) = i + 1\} & , \text{ falls } 0 \leq i \leq n - 2, \\ \{1^{i-n+1}0^{2n-1-i}\} & , \text{ falls } n - 1 \leq i \leq 2n - 1, \end{cases} \text{ und}$$

$$b_i := \begin{cases} i + 1 & , \text{ falls } 0 \leq i \leq 2n - 1 \text{ und } i \neq 2n - m - 1, \\ 2n & , \text{ falls } i = 2n - m - 1. \end{cases}$$

Es ist

$$v_i = \begin{cases} |\emptyset| = 0 & , \text{ falls } 0 \leq i \leq 2n - m - 1, \\ |A_{2n-m-1}| = 1 & , \text{ falls } 2n - m \leq i \leq 2n - 1, \end{cases}$$

und $p(A_i) \geq 1/(en)$, $0 \leq i < 2n - 2$, da zumindest eine spezielle 1-Bit-Mutation eines Elements aus A_i ein Element aus A_{i+1} erzeugt. Da $\mu \geq \max\{v_i \mid 0 \leq i < 2n - 1\} + 1 = 2$ ist, liefert eine Anwendung des Lemmas 3.B.2 für den $(\mu+1)$ -EA mit $\mu \geq 2$ eine erwartete Laufzeit von höchstens $\mu(1 + (2n - 2)en) = \mathcal{O}(\mu n^2)$ zur Optimierung von SHORTPATHONEPEAK.

Das gewünschte Resultat ist somit doppelt gezeigt. \square

Präzisieren wir das Ergebnis für den $(1+1)$ -EA.

Lemma 3.B.4

- a) Mit Wahrscheinlichkeit $1 - \mathcal{O}(1/n)$ benötigt der $(1+1)$ -EA eine Laufzeit von mehr als $n^{n/24}$ zur Optimierung von SHORTPATHONEPEAK.
- b) Mit Wahrscheinlichkeit $\Omega(1/n)$ benötigt der $(1+1)$ -EA eine Laufzeit von höchstens $4en^2$ zur Optimierung von SHORTPATHONEPEAK.

Der $(1+1)$ -EA benötigt also eine erwartete Laufzeit von $2^{\Omega(n \ln n)}$, um SHORTPATHONEPEAK zu optimieren.

Beweis (Lemma 3.B.4). Nach Lemma 3.A.3 gilt $|x_{1,1}| \geq n - \lfloor n/3 \rfloor$ höchstens mit Wahrscheinlichkeit $2^{-n/30}$. Ein Element x mit $x < n - \lfloor n/3 \rfloor$ erzeugt ein Element $1^{n-m+j}0^{m-j}$, $j \geq 0$, in $n^{n/24}$ Schritten höchstens mit Wahrscheinlichkeit $n^{n/24} \cdot 1/n^{n/12} = n^{-n/24}$, da dazu mindestens $n/12$ spezielle Bits flippen müssen.

- a) Ein Individuum $1^{n-m-i}0^{m+i}$, $i \geq 1$, generiert ein Element $1^{n-m+j}0^{m-j}$, $j \geq 1$, vor dem Element $1^{n-m}0^m$ höchstens mit Wahrscheinlichkeit $(1/n^{i+1})/(1/(en^i) + 1/n^{i+1}) \leq e/n$. Der Suchpunkt $1^{n-m}0^m$ generiert das Optimum 1^n – das einzige Element mit mindestens gleichem Funktionswert – in $n^{n/24}$ Schritten höchstens mit Wahrscheinlichkeit $n^{n/24} \cdot 1/n^m \leq n^{-n/5}$. Mindestens mit Wahrscheinlichkeit $1 - (2^{-n/30} + n^{-n/24} + e/n + n^{-n/5}) = 1 - \mathcal{O}(1/n)$ benötigt der $(1+1)$ -EA eine Laufzeit von mehr als $n^{n/24}$ zur Optimierung von SHORTPATHONEPEAK.
- b) Mindestens eine spezielle 1-Bit-Mutation, welche mindestens Wahrscheinlichkeit $1/(en)$ hat, erzeugt ein Individuum mit noch größerem Funktionswert. Dies geschieht, bis entweder das lokale Optimum $1^{n-m}0^m$ oder das globale Optimum 1^n generiert wird. Höchstens $2n$ solcher Mutationen sind dazu notwendig. Eine Anwendung der Chernoffungleichung zeigt, dass mit Wahrscheinlichkeit $1 - e^{-\Omega(n)}$ dazu $4en^2$ Funktionsauswertungen genügen. Ein Individuum $1^{n-m-i}0^{m+i}$, $i \geq 1$, generiert ein Element $1^{n-m+j}0^{m-j}$, $j \geq 1$, vor dem Element $1^{n-m}0^m$ mindestens mit Wahrscheinlichkeit $(1/(en^{i+1}))/(1/n^i + 1/(en^{i+1})) \geq 1/(3n)$. Weiterhin generiert ein Individuum $1^{n-m+i}0^{m-i}$, $i \geq 1$, das Element $1^{n-m}0^m$ vor einem Element $1^{n-m+i+j}0^{m-i-j}$, $j \geq 1$, höchstens mit Wahrscheinlichkeit $(1/n^i)/(1/(en) + 1/n^i) \leq \min\{e/(1+e), 3/n^{i-1}\}$. Die Wahrscheinlichkeit, dass der $(1+1)$ -EA in $4en^2$ Schritten das Optimum erzeugt, beträgt damit mindestens $(1/(3n)) \cdot (1 - e/(1+e))^2 \cdot (1 - \sum_{i=3}^{m-1} 3/n^{i-1}) - (2^{-n/30} - n^{-n/24} - e^{-\Omega(n)}) = \Omega(1/n)$.

Die gewünschten Resultate sind somit gezeigt. \square

Aus den Betrachtungen von Witt (2006) folgt, dass auch der $(\mu+1)$ -EA* sogar mit $\mu \leq \text{poly}(n)$ lediglich mit Wahrscheinlichkeit $\mathcal{O}(1/n)$ höchstens eine Laufzeit von $n^{n/24}$ zur Optimierung von SHORTPATHONEPEAK benötigt. Seine erwartete Laufzeit beträgt also $2^{\Omega(n \ln n)}$. Aus den obigen Betrachtungen folgt auch: Der $\mu \times (1+1)$ -EA benötigt eine erwartete Laufzeit von $2^{\Omega(n \ln n)}$ zur Optimierung von SHORTPATHONEPEAK, solange $\mu \leq c_1 n^2 \ln n$ ist für eine genügend kleine Konstante $c_1 > 0$. Mit $c_2 n^2 \ln n \leq \mu \leq \text{poly}(n)$ für eine genügend große Konstante c_2 genügt dem $\mu \times (1+1)$ -EA dazu jedoch eine erwartete Laufzeit von $\Theta(\mu n^2)$.

3.C Hierarchieresultat

Wir verdeutlichen nun exemplarisch die (mögliche) Sensibilität populationsbasierter evolutionärer Algorithmen gegenüber Veränderungen der Elternpopulationsgröße. Für jedes $d \leq \text{poly}(n)$ werden wir eine Funktion f_d angeben, so dass damit folgendes typische starke Hierarchieresultat für den $(\mu+1)$ -EA gezeigt wird. Wir demonstrieren, wie mit einer Veränderung um eins bereits die kleinstmögliche Modifikation der Populationsgröße zu einem wesentlich unterschiedlichen Verhalten des $(\mu+1)$ -EA führen kann.

Theorem 3.C.1 Sei $1 \leq d \leq \text{poly}(n)$.

- Sei $d + 1 \leq \mu \leq \text{poly}(n)$. Der $(\mu+1)$ -EA erzeugt das Optimum von f_d in einer erwarteten Laufzeit von $\mathcal{O}(\text{poly}(n))$.
- Sei $\mu \leq d$. Mit Wahrscheinlichkeit $1 - 2^{-\Omega(n)}$ benötigt der $(\mu+1)$ -EA eine Laufzeit von mehr als $2^{\varepsilon n}$ zur Optimierung von f_d für eine Konstante $\varepsilon > 0$.

Wir bezeichnen den Wert d auch als Schwellwert der Populationsgröße (engl. threshold value of the population size).

Die Ergebnisse aus Lemma 3.B.3 und aus Lemma 3.B.4 liefern nicht das gewünschte Resultat für $d = 1$, aber schon wesentliche Erkenntnisse für Funktionen mit solchen Eigenschaften. Für den Beweis des Theorem 3.C.1 unterscheiden wir $1 \leq d \leq n$ (Fall 1) und $n+1 \leq d \leq \lfloor n^c \rfloor$ für eine beliebige Konstante $c \geq 1$ (Fall 2). Lemma 3.C.4 zeigt den Fall 1 und Lemma 3.C.3 den Fall 2.

Betrachten wir aus technischen Gründen zunächst den Fall 2. Dazu benötigen wir die *binären Gray-Codes* (engl. *Gray binary codes*) (vergleiche Knuth, 2005). Der ℓ -stellige binäre Gray-Code G_ℓ , $\ell \in \mathbb{N}$, bildet eine natürliche Zahl x mit $0 \leq x \leq 2^\ell - 1$ bijektiv in $\{0, 1\}^\ell$ ab. Aber im Gegensatz zu den *lexikographischen binären Codes* (engl. *lexicographic binary codes*) gilt $H[G_\ell(x), G_\ell(x+1)] = 1$ für alle x mit $0 \leq x < 2^\ell - 1$. Und wie bei den lexikographischen binären Codes gilt $G_\ell(0) = 0^\ell$. Sei nun $m := \lfloor n/3 \rfloor$, $k := \lceil 2 \log_2 n + 2 \log_2 d \rceil$ und $a_{i_{\lfloor m/k+1 \rfloor}}$, $0 \leq i \leq 2^k - 1$, derjenige Suchpunkt $1^{n-m-1} 0^{m-k\lfloor m/k \rfloor+1} g_{k-1}^{\lfloor m/k \rfloor} \dots g_0^{\lfloor m/k \rfloor}$ mit $G_k^{-1}(g_{k-1} \dots g_0) = i$. Es bezeichne k_i , $0 \leq k_i \leq k-1$ und $i < 2^k - 1$, die eindeutige Position, in der sich $G_k(i)$ und $G_k(i+1)$ unterscheiden. Dann entspricht $a_{i_{\lfloor m/k+1 \rfloor+j}}$, $0 \leq i < 2^k - 1$ und $1 \leq j \leq \lfloor m/k \rfloor - 1$, dem Suchpunkt $a_{i_{\lfloor m/k+1 \rfloor}}$, wobei die Bits an den Positionen $n - k_i \lfloor m/k \rfloor, \dots, n - k_i \lfloor m/k \rfloor - j$ geflippt wurden. Diese Suchpunkte $a_0, \dots, a_{(2^k-1)\lfloor m/k+1 \rfloor}$ beschreiben einen Pfad der Länge $(2^k - 1)\lfloor m/k + 1 \rfloor + 1 \geq n^2 d^2$. Aufgrund der zuvor genannten Eigenschaften des binären Gray-Codes erhalten wir sofort die folgenden zwei Eigenschaften des Pfads $a_0, \dots, a_{n^2 d^2}$.

Lemma 3.C.2 Sei $0 \leq i \leq i+j \leq n^2 d^2$.

- $H[a_i, a_{i+j}] = j$, falls $j \leq \lfloor m/k \rfloor$ ist.
- $H[a_i, a_{i+j}] \geq \lfloor m/k \rfloor$, falls $j \geq \lfloor m/k \rfloor$ ist.

Eine Abkürzung bei einem Pfad liegt vor, wenn zwei seiner Elemente trotz eines großen Abstandes auf dem Pfad nur einen geringen Hammingabstand haben. Das Lemma 3.C.2 demonstriert, dass der zuvor definierte (kurze) Pfad keine solche Abkürzung enthält – im Gegensatz zum (langen) Pfad von Horn et al. (1994), wie Rudolph (1997) gezeigt hat.

Wir betrachten $\text{SHORTPATHONEPEAK}_d : \{0, 1\}^n \rightarrow \mathbb{Z}$, $d \leq \lfloor n^c \rfloor$ für konstante $c \geq 1$. Sei

$$\text{SHORTPATHONEPEAK}_d(x) := \begin{cases} 2n + 2i & , \text{ falls } x = a_i \text{ für } d \leq i \leq n^2 d^2, \\ 2n + 2n^2 d^2 - 1 & , \text{ falls } x = a_i \text{ für } 0 \leq i < d, \\ n + i & , \text{ falls } x = 1^i 0^{n-i} \text{ für } 0 \leq i < n - m - 1, \\ n - |x| & , \text{ sonst.} \end{cases}$$

Diese Funktionen sind Varianten von SHORTPATHONEPEAK , bei denen nun a_0, \dots, a_{d-1} eine Spitze der Größe d formen.

Lemma 3.C.3 Sei $n + 1 \leq d \leq \lfloor n^c \rfloor$ für eine beliebige Konstante $c \geq 1$.

- Sei $d + 1 \leq \mu \leq \text{poly}(n)$. Der $(\mu+1)$ -EA erzeugt das Optimum von $\text{SHORTPATHONEPEAK}_d$ in einer erwarteten Laufzeit von $\mathcal{O}(\mu n^3 d^2)$.
- Sei $\mu \leq d$. Mit Wahrscheinlichkeit $1 - 2^{-\Omega(n)}$ benötigt der $(\mu+1)$ -EA eine Laufzeit von mehr als $2^{\varepsilon n}$ zur Optimierung von $\text{SHORTPATHONEPEAK}_d$ für eine Konstante $\varepsilon > 0$.

Beweis.

- Wir beweisen dies analog zum Lemma 3.B.3 mittels Lemma 3.B.2. Sei

$$A_i := \begin{cases} \{x \mid \text{SHORTPATHONEPEAK}_d(x) = i\} & , \text{ falls } 0 \leq i < 2n - m - 1, \\ \{a_{i-2n+m+1}\} & , \text{ falls } 2n - m - 1 \leq i \text{ und } \\ & i \leq 2n - m - 1 + n^2 d^2, \end{cases}$$

$$b_i := \begin{cases} 2n - m - 1 + n^2 d^2 & , \text{ falls } 2n - m - 1 \leq i \leq 2n - m + d - 2, \\ i + 1 & , \text{ sonst.} \end{cases}$$

Es ist $v_i \leq d$ für alle i mit $0 \leq i < 2n - m - 1 + n^2 d^2$ und des Weiteren genügt eine spezielle 1-Bit-Mutation, damit ein Individuum aus A_i einen Suchpunkt aus A_{i+1} erzeugt. Diese hat mindestens Wahrscheinlichkeit $1/(en)$. Der $(\mu+1)$ -EA mit $\mu \geq d + 1 \geq \max\{v_i \mid 0 \leq i < 2n - m - 1 + n^2 d^2\} + 1$ generiert das Optimum von $\text{SHORTPATHONEPEAK}_d$ in einer erwarteten Laufzeit von $\mathcal{O}(\mu \cdot n^2 d^2 \cdot n) = \mathcal{O}(\mu n^3 d^2)$.

- Wir beweisen dies analog zum Lemma 3.B.4. Da $\mu \leq n^c$ ist, beträgt die Wahrscheinlichkeit $1 - 2^{-\Omega(n)}$, dass $|x_{1,1}|, \dots, |x_{1,\mu}| < n - \lfloor n/3 \rfloor$ gilt. Danach sind $a_0, \dots, a_{n^2 d^2}$ die einzigen Elemente, die aus mindestens $n - m$ Einsen bestehen und in die Population aufgenommen werden. Da jeder Suchpunkt $a_n, \dots, a_{n^2 d^2}$ aus mindestens $n - m + \lfloor m/k \rfloor$ Einsen besteht, beträgt die Wahrscheinlichkeit höchstens $(n^2 d^2 - n + 1) \cdot (1/n^{\lfloor m/k \rfloor + 1}) \cdot (2^{\varepsilon n} - 1) = 2^{-\Omega(n)}$, dass ein Individuum x mit $|x| < n - m$ in den verbleibenden höchstens $2^{\varepsilon n} - 1$ betrachteten Schritten ein Element $a_n, \dots, a_{n^2 d^2}$ generiert. Betrachten wir nun die Situation, dass erstmals ein Individuum a_0, \dots, a_{n-1} in der Population auftritt – dieses gehört zur Spitze. Zu zeigen bleibt, dass die Population typischerweise zuerst nur aus Suchpunkten der Spitze besteht, bevor das Ende des Pfades mit dem globalen Optimum erreicht wird. Auf der einen Seite zeigen wir, dass in dieser Situation mit Wahrscheinlichkeit $1 - e^{-\Omega(n)}$ in $\lceil 2\mu n d \rceil$ Schritten die Population aus $\mu \leq d$ Elementen der Spitze besteht

(Ereignis \mathcal{E}_1). Auf der anderen Seite wir zeigen, dass in dieser Situation mit Wahrscheinlichkeit $2^{-\Omega(n)}$ in $\lceil 2e\mu nd \rceil$ Schritten das Ende des Pfades erreicht und damit das Optimum erzeugt wird (Ereignis \mathcal{E}_2). Jede Population, die nur aus Elementen der Spitze besteht, erzeugt ein Element mit mindestens so großem Funktionswert wie jedes Element der Spitze mit Wahrscheinlichkeit $2^{-\Omega(n)}$ in $2^{\varepsilon n} - 1$ Schritten nach Lemma 3.C.2.b. Der $(\mu+1)$ -EA benötigt mit Wahrscheinlichkeit $1 - (2^{-\Omega(n)} + 2^{-\Omega(n)} + e^{-\Omega(n)} + 2^{-\Omega(n)}) = 1 - 2^{-\Omega(n)}$ mehr als $2^{\varepsilon n}$ Funktionsauswertungen zur Optimierung von $\text{SHORTPATHONEPEAK}_d$.

Wir betrachten das Ereignis \mathcal{E}_1 . Da $\mu \leq d$ ist, existiert – wenigstens solange nicht die ganze Population aus Elementen der Spitze besteht – mindestens ein Element a_i , $0 \leq i < d$, mit $a_i \notin \mathcal{P}$ und weiterhin $a_{i-1} \in \mathcal{P}$ oder $a_{i+1} \in \mathcal{P}$. Eine spezielle 1-Bit-Mutation von a_{i-1} und a_{i+1} erzeugt a_i und nimmt es in die Population auf. Dies hat somit mindestens Wahrscheinlichkeit $1/(e\mu n)$. Eine Anwendung der Chernoffungleichung zeigt weiter, dass mit Wahrscheinlichkeit $e^{-\Omega(n)}$ weniger als $d - 1 \geq \mu - 1$ solche speziellen 1-Bit-Mutationen in $\lceil 2e\mu nd \rceil$ Schritten auftreten.

Wir betrachten das Ereignis \mathcal{E}_2 . Nach Lemma 3.C.2.b beträgt die Wahrscheinlichkeit höchstens $n^2 d^2 \cdot (1/n^{\lfloor m/k \rfloor}) \cdot \lceil 2e\mu nd \rceil = 2^{-\Omega(n)}$ in den $\lceil 2e\mu nd \rceil$ betrachteten Schritten, ein Element $a_{\max\{i|a_i \in \mathcal{P}\} + \lfloor m/k \rfloor}, \dots, a_{n^2 d^2}$ zu erzeugen. Es werden damit nur nicht optimale Elemente a_i mit $i \leq (n-1) + \lceil 2e\mu nd \rceil \lfloor m/k - 1 \rfloor < n^2 d^2$ generiert.

Das gewünschte Resultat ist somit gezeigt. \square

Betrachten wir nun den Fall 1. Das Lemma 3.B.4 zeigt, dass der $(1+1)$ -EA zur Optimierung von SHORTPATHONEPEAK mit Wahrscheinlichkeit $\Theta(1/n)$ eine polynomielle Laufzeit benötigt, da dann die Spitze übersprungen und schnell das Optimum erzeugt wird. Wir modifizieren SHORTPATHONEPEAK derart, dass der $(1+1)$ -EA typischerweise mehrmals in die Situation gelangt, eine Spitze überspringen zu müssen – was typischerweise nicht gelingt – während der $(\mu+1)$ -EA mit $\mu \geq 2$ effizient bleibt. Diese Technik der Wahrscheinlichkeitsamplifizierung in der Funktion (engl. probability amplification in the function) hat Sudholt (2006) aufgegriffen, um das Verhalten memetischer Algorithmen zu analysieren.

Sei nun $m := \lfloor n/3 \rfloor$ und $k := \lceil 5 \log_2 n \rceil$. Wir betrachten einen Pfad, dessen Länge $n \cdot n^3 d + 1$ ist, und bemerken, dass für n groß genug gilt $\lfloor m/k \rfloor \geq \lceil n/(16 \log_2 n) \rceil$. Die Suchpunkte $P_j := \{a_{j \cdot n^3 d}, \dots, a_{(j+1) \cdot n^3 d - 1}\}$, $0 \leq j \leq n$, stellen die j -te Spitze (engl. peak) dar und $\{a_{j \cdot n^3 d + d}, \dots, a_{(j+1) \cdot n^3 d - 1}\}$ die Brücke (engl. bridge) zwischen der j -ten und $(j+1)$ -ten Spitze. Wir betrachten $\text{SHORTPATHMANYPEAKS}_d : \{0, 1\}^n \rightarrow \mathbb{Z}$, $d \leq n$. Sei

$$\text{SHORTPATHMANYPEAKS}_d(x) := \begin{cases} 2n + 2i & , \text{ falls } x = a_i \notin P_j, \\ & 0 \leq i \leq n \cdot n^3 d, \\ 2n + 2(j+1) \cdot n^3 d - 1 & , \text{ falls } x = a_i \in P_j, \\ & 0 \leq i \leq n \cdot n^3 d, \\ n + i & , \text{ falls } x = 1^i 0^{n-i}, \\ & 0 \leq i < n - m - 1, \\ n - |x| & , \text{ sonst.} \end{cases}$$

Lemma 3.C.4 Sei $1 \leq d \leq n$.

- a) Sei $d+1 \leq \mu \leq \text{poly}(n)$. Der $(\mu+1)$ -EA erzeugt das Optimum von $\text{SHORTPATHMANYPEAKS}_d$ in einer erwarteten Laufzeit von $\mathcal{O}(\mu n^5 d)$.
- b) Sei $\mu \leq d$. Mit Wahrscheinlichkeit $1 - 2^{-\Omega(n)}$ benötigt der $(\mu+1)$ -EA eine Laufzeit von mehr als $2^{\varepsilon n}$ zur Optimierung von $\text{SHORTPATHMANYPEAKS}_d$ für eine Konstante $\varepsilon > 0$.

Beweis.

- a) Wir beweisen dies analog zum Lemma 3.C.3. Bei gleicher Einteilung der Elemente in A_i wie im Beweis des Lemmas 3.C.3 ist

$$b_i := \begin{cases} 2n - m - 1 + (j + 1) \cdot n^3 d & , \text{ falls } 2n - m - 1 + j \cdot n^3 d \leq i \text{ und} \\ & i \leq 2n - m - 1 + j \cdot n^3 d + d - 1 \text{ für } 0 \leq j < n, \\ i + 1 & , \text{ sonst,} \end{cases}$$

und da $v_i \leq d$ ist für alle i mit $0 \leq i < 2n - m - 1 + n \cdot n^3 d$, folgt das gewünschte Resultat.

- b) Wir beweisen auch dies analog zum Lemma 3.C.3. Mit Wahrscheinlichkeit $1 - 2^{-\Omega(n)}$ werden bis auf die Elemente $a_0, \dots, a_{n \cdot n^3 d}$ keine Suchpunkte x mit $|x| \geq n - m$ in die Population aufgenommen. Wenn während der $2^{\varepsilon n}$ betrachteten Schritte erstmals ein Element $a_0, \dots, a_{n \cdot n^3 d}$ auftritt, ist dies mit Wahrscheinlichkeit $1 - 2^{-\Omega(n)}$ aus a_0, \dots, a_{n-1} . Sei $i^* := \max\{i \mid a_i \in \mathcal{P}\}$. Nach Lemma 3.C.2.b beträgt die Wahrscheinlichkeit $2^{-\Omega(n)}$, in den höchstens $2^{\varepsilon n} - 1$ verbleibenden Schritten eine Mutation durchzuführen, die ein $a_{i^* + \lceil n/(16 \log_2 n) \rceil}, \dots, a_{n \cdot n^3 d}$ generiert. Wie im Beweis des Lemmas 3.B.4.a und nach Lemma 3.C.2.a beträgt die Wahrscheinlichkeit $1 - \mathcal{O}(1/n)$, das Element $a_{j \cdot n^3 d} \in P_j$ mit $1 \leq j \cdot n^3 d - i^* < \lceil n/(16 \log_2 n) \rceil$ vor einem Suchpunkt $a_{j \cdot n^3 d + 1}, \dots, a_{i^* + \lceil n/(16 \log_2 n) \rceil - 1}$ zu generieren. Tritt dies ein, wird sicher ein Individuum der (nachfolgenden) Spitze vor einem Element hinter der (nachfolgenden) Spitze erzeugt und damit die (nachfolgende) Spitze nicht übersprungen. Da $j \cdot n^3 d + \lceil n/(16 \log_2 n) \rceil - 1 < (j + 1) \cdot n^3 d$ ist – also wird höchstens eine Spitze in einem Schritt übersprungen – und $n \leq n^3 d$ – also wird zuerst die erste Spitze betrachtet – muss der $(\mu+1)$ -EA mindestens $(n - 1)$ -mal eine Spitze überspringen, um das Optimum zu generieren, bevor eine Population mit $a_{i^*} \in P_j$ für ein $0 \leq j \leq n - 1$ entsteht. Höchstens mit Wahrscheinlichkeit $\mathcal{O}(1/n)^{n-1} = 2^{-\Omega(n)}$ tritt dies ein. Analog zum Beweis des Lemmas 3.C.3 besteht dann die Population nach $\lceil 2e\mu n^2 \rceil$ Schritten mit Wahrscheinlichkeit $1 - e^{-\Omega(n)}$ aus $\mu \leq d \leq n$ Elementen der Spitze P_j . Weiterhin beträgt die Wahrscheinlichkeit $2^{-\Omega(n)}$, in den $\lceil 2e\mu n^2 \rceil$ betrachteten Schritten ein Element $a_{(j+1) \cdot n^3 d}, \dots, a_{n \cdot n^3 d}$ zu generieren, da $d + \lceil 2e\mu n^2 \rceil \cdot (\lceil n/(16 \log_2 n) \rceil - 1) < n^3 d$ ist. Wenn die ganze Population aus Suchpunkten der Spitze P_j , $j < n$, besteht, wird in den betrachteten Schritten kein Individuum erzeugt, das einen mindestens so großen Funktionswert hat wie jedes Element der Spitze P_j .

Das gewünschte Resultat ist somit gezeigt. \square

Insbesondere wird die Funktion $\text{SHORTPATHMANYPEAKS}_1$ vom $(\mu+1)$ -EA mit $2 \leq \mu \leq \text{poly}(n)$ effizient optimiert, während sowohl der $(\mu+1)$ -EA^{*} als auch der $\mu \times (1+1)$ -EA sogar mit beliebigem $\mu \leq \text{poly}(n)$ total ineffizient sind. Dies zeigen die obigen Betrachtungen und solche von Witt (2006).

Dieser $(\mu+1)$ -EA wird in den Kapiteln 4 und 5 weiter analysiert.

4 TRANSFORMATIONEN DER ZIELFUNKTIONEN UND RANDOMISIERTE SUCHHEURISTIKEN

Transformationen der Zielfunktionen verändern die Laufzeiten randomisierter Suchheuristiken bei der Optimierung. Wenn die Suchheuristik A neben der Zielfunktion $f : \{0, 1\}^n \rightarrow \mathbb{Z}$ ebenso die transformierte Zielfunktion $g \circ f : \{0, 1\}^n \rightarrow \mathbb{Z}$ effizient optimiert, nennen wir die Transformation $g : \mathbb{Z} \rightarrow \mathbb{Z}$ *nicht verschlechternd für A bei Optimierung von f* . Diese Separation der Transformationen g hängt sowohl von der spezifischen Suchheuristik A als auch von der spezifischen Funktion f ab. Ist g nicht verschlechternd für A bei Optimierung von f für jede Funktion $f : \{0, 1\}^n \rightarrow \mathbb{Z}$, nennen wir die Transformation $g : \mathbb{Z} \rightarrow \mathbb{Z}$ *nicht verschlechternd für A* . Diese Separation der Transformationen g hängt lediglich von der spezifischen Suchheuristik A ab. Sei \mathcal{T}_A die Menge der für A nicht verschlechternden Transformationen. Die Bestimmung und der Vergleich solcher Mengen von Transformationen \mathcal{T}_A zeigen die Robustheit von Algorithmen A gegenüber Veränderungen der Zielfunktion. Im Folgenden betrachten wir einfache, aber auch ganze Klassen randomisierter Suchheuristiken.

4.A Alle randomisierten Suchheuristiken

Wir beginnen unsere Betrachtungen mit der Klasse aller randomisierten Suchheuristiken. Dazu sei \mathcal{T}_{af} die Menge aller Funktionen (engl. all functions) $g : \mathbb{Z} \rightarrow \mathbb{Z}$ und \mathcal{T}_{ti} die Menge aller partiellen Identitäten (engl. truncated identities).

Definition (partielle Identität) *Eine Funktion $g : \mathbb{Z} \rightarrow \mathbb{Z}$ heißt partielle Identität, falls ein $z_g \in \mathbb{Z} \cup \{-\infty, \infty\}$ existiert, so dass $g(m) = m$ für alle $m < z_g$ ist und $z_g \leq g(m) = g(m+1)$ für alle $m \geq z_g$ gilt.*

Eine solche Funktion ist bis zu einem Wert die Identität und ist dann konstant – eingeschlossen sind die Identität an sich und alle konstanten Funktionen. Wir zeigen für jede Suchheuristik A , dass gilt

$$\mathcal{T}_{ti} \subseteq \mathcal{T}_A \subseteq \mathcal{T}_{af}.$$

Um die mögliche Unechtheit der Inklusionen zu demonstrieren, werden wir daraufhin auf der einen Seite für eine (konstruierte) Suchheuristik A_{ti} beweisen, dass $\mathcal{T}_{A_{ti}} = \mathcal{T}_{ti}$ gilt und auf der anderen Seite für eine (konstruierte) Suchheuristik A_a zeigen, dass $\mathcal{T}_{A_a} = \mathcal{T}_{af}$ gilt.

Um das gewünschte Resultat für alle Suchheuristiken zu erhalten, genügt es, die erste Inklusion zu beweisen. Die zweite Inklusion folgt, da \mathcal{T}_{af} alle möglichen Transformationen enthält.

Theorem 4.A.1 Sei $f : \{0, 1\}^n \rightarrow \mathbb{Z}$ eine beliebige Funktion und A eine Suchheuristik. Falls $g \in \mathcal{T}_{\text{ii}}$ ist, gilt für alle $\ell(n) \geq 0$

$$\Pr[T_{g \circ f, A} > \ell(n)] \leq \Pr[T_{f, A} > \ell(n)].$$

Damit ist auch $E[T_{g \circ f, A}] \leq E[T_{f, A}]$ und die Suchheuristik A optimiert die Funktion $g \circ f$ insbesondere dann effizient, wenn sie f effizient optimiert.

Beweis (Theorem 4.A.1). Wir betrachten eine beliebige Folge von Elementen $x_1, \dots, x_{\ell(n)}$, die kein Optimum von $g \circ f$ enthält, also $x_t \in \{0, 1\}^n \setminus \arg \max\{g \circ f(x) \mid x \in \{0, 1\}^n\}$ gilt für $1 \leq t \leq \ell(n)$. Da $g \in \mathcal{T}_{\text{ii}}$ ist, gilt $\arg \max\{g \circ f(x) \mid x \in \{0, 1\}^n\} \supseteq \arg \max\{f(x) \mid x \in \{0, 1\}^n\}$ und $x_1, \dots, x_{\ell(n)}$ enthält auch keine Optima von f . Dann ist $g \circ f(x_t) = f(x_t)$ für alle t mit $1 \leq t \leq \ell(n)$ und die Behauptung folgt direkt, da A eine Suchheuristik ist und somit die Wahl von x_t nur von $(x_1, f(x_1)), \dots, (x_{t-1}, f(x_{t-1}))$, $1 \leq t \leq \ell(n)$, abhängt. \square

Die Umkehrung der Aussage ist typischerweise nicht korrekt. Betrachten wir eine beliebige Funktion f , die der Algorithmus A nicht effizient optimiert. Beispielsweise optimiert jede Heuristik eine Funktion NEEDLE_a für (mindestens) ein $a \in \{0, 1\}^n$ total ineffizient. Aber die Funktion $g \circ f$, wobei g eine konstante Funktion ist, optimiert jeder Algorithmus mit der ersten Funktionsauswertung – da nur optimale Suchpunkte existieren.

Betrachten wir statt der partiellen Identitäten nur die Identität, gilt $\Pr[T_{g \circ f, A} > \ell(n)] = \Pr[T_{f, A} > \ell(n)]$.

Wir beschreiben eine Suchheuristik A_{ii} mit $\mathcal{T}_{A_{\text{ii}}} = \mathcal{T}_{\text{ii}}$. Dazu sei G_ℓ , $\ell \in \mathbb{N}$, der ℓ -stellige binäre Gray-Code, der eine natürliche Zahl x mit $0 \leq x \leq 2^\ell - 1$ bijektiv in $\{0, 1\}^\ell$ abbildet (vergleiche Knuth, 2005). Und weiter sei $\alpha : \mathbb{Z} \rightarrow \mathbb{N}$ die bijektive Abbildung

$$\alpha(m) := \begin{cases} 2|m| & , \text{ falls } m \geq 0, \\ 2|m| - 1 & , \text{ falls } m < 0. \end{cases}$$

A_{ii}

1. Setze $t := 1$ und wähle 0^n .
2. Setze $t := t + 1$. Falls $t \leq 2^n$, dann wähle x_t mit $G_n^{-1}(x_t) = (\alpha(f(0^n)) \bmod (2^n - 1)) + 1$ und sonst wähle $x_t \in \{0, 1\}^n$ zufällig gleichverteilt.
3. Gehe zu 2.

Die randomisierte Suchheuristik A_{ii} optimiert jede Funktion in einer endlichen erwarteten Laufzeit.

Theorem 4.A.2 Sei n groß genug. Falls $g \notin \mathcal{T}_{\text{ii}}$ ist, dann existiert eine Funktion $f : \{0, 1\}^n \rightarrow \mathbb{Z}$ mit den folgenden Eigenschaften:

- a) A_{ii} erzeugt das Optimum von f in einer Laufzeit von höchstens 2.
- b) A_{ii} benötigt eine Laufzeit von mindestens $2^n + 1$ zur Optimierung von $g \circ f$.

Beweis. Es gibt mindestens zwei ganze Zahlen m_1 und m_2 mit $g(m_1) \neq m_1 < m_2$ und $g(m_1) \neq g(m_2)$. Für n groß genug ist $\alpha(m_1), \alpha(g(m_1)), \alpha(m_2), \alpha(g(m_2)) < 2^n - 1$. Wir unterscheiden $g(m_1) > g(m_2)$ (Fall 1) und $g(m_1) < g(m_2)$ (Fall 2).

1. Wir betrachten die Funktion $f_1 : \{0, 1\}^n \rightarrow \mathbb{Z}$ mit

$$f_1(x) := \begin{cases} m_2 & , \text{ falls } x = 0^n \text{ oder } G_n^{-1}(x) = \alpha(g(m_2)) + 1, \\ m_1 & , \text{ sonst.} \end{cases}$$

A_{ti} erzeugt mit 0^n im ersten Schritt ein Optimum von f_1 .

A_{ti} erzeugt in den ersten 2^n Schritten bei der Optimierung von $g \circ f_1$ die Individuen 0^n und x mit $G_n^{-1}(x) = \alpha(g(m_2)) + 1$. Dies sind beides keine Optima.

2. Wir betrachten die Funktion $f_2 : \{0, 1\}^n \rightarrow \mathbb{Z}$ mit

$$f_2(x) := \begin{cases} m_1 & , \text{ falls } x = 0^n \text{ oder } G_n^{-1}(x) = \alpha(g(m_1)) + 1, \\ m_2 & , \text{ sonst.} \end{cases}$$

A_{ti} erzeugt in den ersten zwei Schritten die Individuen 0^n und $x \neq 0^n$ mit $G_n^{-1}(x) = \alpha(m_1) + 1 \neq \alpha(g(m_1)) + 1$. Daher ist x ein Optimum für f_2 .

A_{ti} erzeugt in den ersten 2^n Schritten bei der Optimierung von $g \circ f_1$ die Individuen 0^n und x mit $G_n^{-1}(x) = \alpha(g(m_1)) + 1$. Dies sind beides keine Optima.

Das gewünschte Resultat ist somit gezeigt. □

Wir beschreiben eine Suchheuristik A_{af} mit $\mathcal{T}_{A_{af}} = \mathcal{T}_{af}$.

A_{af}

1. Setze $t := 1$ und wähle $x_1 \in \{0, 1\}^n$ zufällig gleichverteilt.
2. Setze $t := t + 1$. Falls $t \leq 2^{2^n}$, dann wähle $x_t = x_1$ und sonst wähle $x_t \in \{0, 1\}^n$ zufällig gleichverteilt.
3. Gehe zu 2.

Die randomisierte Suchheuristik A_{af} optimiert jede Funktion in einer endlichen erwarteten Laufzeit.

Theorem 4.A.3 Sei $f : \{0, 1\}^n \rightarrow \mathbb{Z}$ eine beliebige Funktion und $g \in \mathcal{T}_{af}$. A_{af} erzeugt das Optimum von f in einer erwarteten Laufzeit von $\ell(n) \leq 2^n$ genau dann, wenn A_{af} das Optimum von $g \circ f$ in einer erwarteten Laufzeit von $\ell(n)$ generiert.

Beweis. Es genügt zu zeigen, dass A_{af} nur konstante Funktionen f in einer erwarteten Laufzeit von $\ell(n) \leq 2^n$ optimiert. Dann gilt sogar $\ell(n) = 1$. Das gewünschte Resultat folgt, da jede Transformation einer konstanten Funktion wiederum in einer konstanten Funktion resultiert. Angenommen, f ist keine konstante Funktion. Dann existiert ein $x \in \{0, 1\}^n$, das kein Optimum darstellt. Mit Wahrscheinlichkeit 2^{-n} wählt A_{af} im ersten Schritt den Suchpunkt $x_1 = x$ und in den ersten 2^{2^n} Schritten wird kein Optimum erzeugt. Damit folgt eine erwartete Laufzeit des A_{af} zur Optimierung von f von mindestens $2^{-n} \cdot (2^{2^n} + 1) > 2^n$. □

4.B Black-Box-Algorithmen und -Komplexität

Wir haben in Abschnitt 1.A bereits Betrachtungen zu Black-Box-Algorithmen und -Komplexität vorgenommen. Da die Black-Box-Komplexität anstatt für eine einzelne Funktion $f : \{0, 1\}^n \rightarrow \mathbb{Z}$ Aussagen für Klassen \mathcal{F} solcher Funktionen macht, betrachten wir mit $g \circ \mathcal{F} := \{g \circ f \mid f \in \mathcal{F}\}$ auch die Klasse der transformierten Funktionen. Bei der Betrachtung von Black-Box-Algorithmen und -Komplexität ist zu unterscheiden, ob der Algorithmus „Zugriff“ oder „keinen Zugriff“ auf die Transformation g hat. Bei „Zugriff“ kann der Black-Box-Algorithmus den Wert $g(m)$ für alle m berechnen. Bei „keinem Zugriff“ ist $g(m)$ für alle m nicht zu bestimmen. Wir analysieren, welche Transformationen nicht verschlechternd für Black-Box-Algorithmen mit Zugriff auf g sind. Warum dieser Zugriff essentiell sein kann, demonstrieren wir im Anschluss daran.

Sei \mathcal{T}_{if} die Menge der monoton steigenden Funktionen (engl. increasing functions).

Definition (monoton steigende Funktion) *Eine Funktion $g : \mathbb{Z} \rightarrow \mathbb{Z}$ heißt monoton steigende Funktion, falls $g(m) \leq g(m+1)$ für alle m ist.*

Eine echte Einschränkung dieser Menge von Funktionen stellt die folgende Menge dar. Sei $\mathcal{T}_{\text{tsif}}$ die Menge der partiell streng monoton steigenden Funktionen (engl. truncated strictly increasing functions).

Definition (partiell streng monoton steigende Funktion) *Eine Funktion $g : \mathbb{Z} \rightarrow \mathbb{Z}$ heißt partiell streng monoton steigende Funktion, falls ein $z_g \in \mathbb{Z} \cup \{-\infty, \infty\}$ existiert, so dass $g(m) < g(m+1)$ für alle $m < z_g$ ist und $g(m) = g(m+1)$ für alle $m \geq z_g$ gilt.*

Eine solche Funktion ist bis zu einem Wert streng monoton steigend und ist dann konstant – eingeschlossen sind alle streng monoton steigenden und alle konstanten Funktionen.

Für alle Black-Box-Algorithmen werden wir nachweisen, dass genau die partiell streng monoton steigenden Funktionen $\mathcal{T}_{\text{tsif}}$ die nicht verschlechternden Transformationen sind.

Theorem 4.B.1 *Sei \mathcal{F} eine beliebige Klasse von Funktionen $f : \{0, 1\}^n \rightarrow \mathbb{Z}$ und A ein Black-Box-Algorithmus. Falls $g \in \mathcal{T}_{\text{tsif}}$ ist, dann existiert ein Black-Box-Algorithmus A_g mit Zugriff auf g , so dass für alle $\ell(n) \geq 0$ gilt*

$$\Pr[T_{g \circ f, A_g} > \ell(n)] \leq \Pr[T_{f, A} > \ell(n)].$$

Beweis. Wir betrachten den folgenden Black-Box-Algorithmus A_g für $g \circ f$. A_g simuliert A .

A_g

1. Setze $t := 0$.
2. Setze $t := t + 1$. Gilt $g \circ f(x_i) < g(z_g)$ für alle i mit $1 \leq i \leq t - 1$, berechne die (dann eindeutigen) Werte $f(x_1), \dots, f(x_{t-1})$. Wie Black-Box-Algorithmus A : Abhängig von $(x_1, f(x_1)), \dots, (x_{t-1}, f(x_{t-1}))$ bestimme eine Wahrscheinlichkeitsverteilung auf $\{0, 1\}^n$, wähle x_t bezüglich dieser Verteilung und bestimme $g \circ f(x_t)$.
3. Gehe zu 2.

Wir betrachten eine beliebige Folge von Elementen $x_1, \dots, x_{\ell(n)}$, die kein Optimum von $g \circ f$ enthält, also $x_t \in \{0, 1\}^n \setminus \arg \max\{g \circ f(x) \mid x \in \{0, 1\}^n\}$ gilt für $1 \leq t \leq \ell(n)$. Da $g \in \mathcal{T}_{\text{tsif}}$ ist, gilt $\max\{g \circ f(x) \mid x \in \{0, 1\}^n\} \leq g(z_g)$ und bei der Optimierung von $g \circ f$ durch A_g gilt $g \circ f(x_t) < g(z_g)$ für alle t mit $1 \leq t \leq \ell(n)$. Somit wurde x_t , $1 \leq t \leq \ell(n)$, durch A_g bei der Optimierung von $g \circ f$ mit der gleichen Wahrscheinlichkeitsverteilung gewählt wie x_t , $1 \leq t \leq \ell(n)$, durch A bei der Optimierung von f . Und es folgt die Behauptung. \square

Dieser Zugriff auf g durch den Black-Box-Algorithmus kann relaxiert werden. Sei $g \in \mathcal{G} \subseteq \mathcal{T}_{\text{tsif}}$. Für den Black-Box-Algorithmus $A_{\mathcal{G}}$, der jeweils einen Schritt des oben beschriebenen Algorithmus A_g für alle $g \in \mathcal{G}$ durchführt, gilt $\Pr[T_{g \circ f, A_{\mathcal{G}}} > |\mathcal{G}| \cdot \ell(n)] \leq \Pr[T_{f, A} > \ell(n)]$. Der Algorithmus $A_{\mathcal{G}}$ hat „beschränkten Zugriff“ auf g , da nur $g \in \mathcal{G}$ bekannt ist. Insbesondere ist $A_{\mathcal{G}}$ effizient, wenn A effizient und $|\mathcal{G}| \leq \text{poly}(n)$ ist.

Um nachzuweisen, dass genau die partiell streng monoton steigenden Funktionen $\mathcal{T}_{\text{tsif}}$ die nicht verschlechternden Transformationen für alle Black-Box-Algorithmen sind, genügt es, das Folgende zu zeigen.

Theorem 4.B.2 Falls $g \notin \mathcal{T}_{\text{tsif}}$ ist, dann existiert eine Klasse von Funktionen \mathcal{F} mit den folgenden Eigenschaften:

- Es existiert ein (deterministischer) Black-Box-Algorithmus, der das Optimum von f für alle $f \in \mathcal{F}$ in einer Laufzeit von höchstens $n + 1$ erzeugt.
- Die Black-Box-Komplexität von $g \circ \mathcal{F}$ beträgt $(2^n + 1)/2$.

Beweis. Wir unterscheiden, ob ein m_1 mit $g(m_1) > g(m_1 + 1)$ existiert (Fall 1) oder ob m_1 und m_2 mit $m_1 < m_2$ und $g(m_1) = g(m_1 + 1) < g(m_2)$ existieren (Fall 2).

- Wir betrachten die Klasse \mathcal{F}_1 aller Funktionen $f_{1,a} : \{0, 1\}^n \rightarrow \mathbb{Z}$, $a \in \{0, 1\}^n$, mit

$$f_{1,a}(x) := \begin{cases} m_1 & , \text{ falls } x = a, \\ m_1 + 1 & , \text{ sonst.} \end{cases}$$

Wir beschreiben einen Black-Box-Algorithmus $A_{\mathcal{F}_1}$ für \mathcal{F}_1 .

$A_{\mathcal{F}_1}$

- Bestimme $f_{1,a}(0^n)$ und $f_{1,a}(1^n)$.

Wenigstens eines der Elemente 0^n und 1^n ist optimal für jedes $f_{1,a} \in \mathcal{F}_1$.

Die Klasse von Funktionen $g \circ \mathcal{F}_1$ entspricht NEEDLE mit $\mathcal{A} = \{0, 1\}^n$ und nach Theorem 1.A.1 beträgt damit die Black-Box-Komplexität $(2^n + 1)/2$ für $g \circ \mathcal{F}_1$.

- Wir betrachten die Klasse \mathcal{F}_2 aller Funktionen $f_{2,a} : \{0, 1\}^n \rightarrow \mathbb{Z}$, $a = a_1 \cdots a_n \in \{0, 1\}^n$, mit

$$f_{2,a}(x) := \begin{cases} m_2 & , \text{ falls } x = a, \\ m_1 + 1 & , \text{ falls } x = p_i, 0 \leq i \leq n, \text{ und } x \neq a, \\ m_1 & , \text{ sonst,} \end{cases}$$

wobei $p_i := a_1 \cdots a_i 0^{n-i}$, $0 \leq i \leq n$, ist. Die Suchpunkte p_0, \dots, p_n stellen einen Pfad (engl. *path*) zum Optimum dar.

Wir beschreiben einen Black-Box-Algorithmus $A_{\mathcal{F}_2}$ für \mathcal{F}_2 .

$A_{\mathcal{F}_2}$

1. Setze $t := 1$, $y := 0^n$ und $x_1 := y$ und bestimme $f_{2,a}(x_1)$.
2. Setze $t := t + 1$. Setze $x_t := y_1 \cdots y_{t-2} 10^{n-t+1}$ und bestimme $f_{2,a}(x_t)$. Falls $f_{2,a}(x_t) \geq f_{2,a}(x_1)$, setze $y := x_t$.

Wir zeigen nun induktiv, dass nach dem $t(n)$ -ten Schritt $y = p_{t(n)-1}$ gilt und $f_{2,a}(y)$ bestimmt wurde. Nach dem $(n+1)$ -ten Schritt gilt dann $y = p_n = a$ und somit wurde das Optimum erzeugt.

Für $t(n) = 1$ ist $y = 0^n = p_0$. Dies zeigt die Behauptung.

Für $t(n) > 1$ bemerken wir, dass sich $p_{t(n)-2}$ und $p_{t(n)-1}$ höchstens im $(t(n)-1)$ -ten Bit $a_{t(n)-1}$ unterscheiden. Nach Voraussetzung gilt vor dem $t(n)$ -ten Schritt $y = p_{t(n)-2}$.

- Ist $a_{t(n)-1} = 0$, gilt $f_{2,a}(x_{t(n)}) = m_1 < m_1 + 1 \leq f_{2,a}(p_0) = f_{2,a}(0^n)$. Daher wird y nicht verändert und es gilt $y = p_{t(n)-2} = p_{t(n)-1}$.
- Ist $a_{t(n)-1} = 1$, gilt entweder $f_{2,a}(x_{t(n)}) = m_1 + 1 = f_{2,a}(p_0) = f_{2,a}(0^n)$ oder $f_{2,a}(x_{t(n)}) = m_2 \geq f_{2,a}(p_0) = f_{2,a}(0^n)$. Daher wird y verändert – das $(t(n)-1)$ -te Bit wird geflippt – und es ist $y = p_{t(n)-1}$.

Dies zeigt die Behauptung.

Die Klasse von Funktionen $g \circ \mathcal{F}_2$ entspricht NEEDLE mit $\mathcal{A} = \{0, 1\}^n$ und nach Theorem 1.A.1 beträgt damit die Black-Box-Komplexität $(2^n + 1)/2$ für $g \circ \mathcal{F}_2$.

Das gewünschte Resultat ist somit gezeigt. □

Wir weisen nun nach, dass der Zugriff des Black-Box-Algorithmus auf die Transformation essentiell sein kann – zumindest bis zu einem gewissen Grad. Für eine Klasse von Funktionen \mathcal{F} , für die ein effizienter Black-Box-Algorithmus existiert und damit auch ein effizienter Black-Box-Algorithmus für $g \circ \mathcal{F}$ mit $g \in \mathcal{T}_{\text{tsif}}$, werden wir zeigen, dass kein effizienter Black-Box-Algorithmus für $\mathcal{T}_{\text{tsif}} \circ \mathcal{F} := \bigcup_{g \in \mathcal{T}_{\text{tsif}}} g \circ \mathcal{F}$ existiert.

Wir betrachten die Klasse NEEDLEPOINTER aller Funktionen $\text{NEEDLEPOINTER}_a : \{0, 1\}^n \rightarrow \mathbb{Z}$, $a \in \{0, 1\}^n$, mit

$$\text{NEEDLEPOINTER}_a(x) := \begin{cases} 2^n & , \text{ falls } x = a, \\ G_n^{-1}(a) & , \text{ sonst.} \end{cases}$$

Mit jedem Funktionswert eines nicht optimalen Elements werden genügend Informationen zum Ort der Nadel – des Optimums a – aufgedeckt, was seine Bestimmung erleichtert. Damit zeigen (engl. point) die Funktionswerte direkt auf die *Nadel* (engl. *needle*). Durch die Betrachtung geeigneter partiell streng monoton steigender Transformationen werden jegliche Informationen zum Ort der Nadel – des Optimums a – überdeckt, was seine Bestimmung erschwert.

Theorem 4.B.3

- a) Es existiert ein (deterministischer) Black-Box-Algorithmus, der das Optimum von f für alle $f \in \text{NEEDLEPOINTER}$ in einer Laufzeit von höchstens 2 erzeugt.
- b) Die Black-Box-Komplexität von $\mathcal{T}_{\text{tsif}} \circ \text{NEEDLEPOINTER}$ beträgt $(2^n + 1)/2$.

Beweis.

- a) Wir beschreiben einen Black-Box-Algorithmus A_N für NEEDLEPOINTER .

A_N

1. Bestimme $f(0^n)$ und falls $0 \leq f(0^n) \leq 2^n - 1$, dann bestimme $f(G_n(f(0^n)))$.

Falls die zu optimierende Funktion NEEDLEPOINTER_a mit $a = 0^n$ ist, erzeugt bereits die erste Funktionsauswertung das Optimum. Falls $a \neq 0^n$ ist, generiert die zweite Funktionsauswertung das Optimum.

- b) Wir betrachten die Klasse \mathcal{F} aller Funktionen $f_a : \{0, 1\}^n \rightarrow \mathbb{Z}$, $a \in \{0, 1\}^n$, mit

$$f_a(x) := \begin{cases} 2^n & , \text{ falls } x = a, \\ 0 & , \text{ sonst.} \end{cases}$$

Da $f_a = g_{-G_n^{-1}(a)} \circ \text{NEEDLEPOINTER}_a$ ist, wobei $g_i : \mathbb{Z} \rightarrow \mathbb{Z}$ mit $-(2^n - 1) \leq i \leq 0$ die (partiell) streng monoton steigende Funktion

$$g_i(m) := \begin{cases} m - 2^n & , \text{ falls } m < 0, \\ m + i & , \text{ falls } 0 \leq m \leq 2^n - 1, \\ m & , \text{ falls } m \geq 2^n, \end{cases}$$

ist, gilt $\mathcal{F} \subseteq \mathcal{T}_{\text{tsif}} \circ \text{NEEDLEPOINTER}$. Die Klasse von Funktionen \mathcal{F} entspricht NEEDLE mit $\mathcal{A} = \{0, 1\}^n$ und nach Theorem 1.A.1 beträgt damit die Black-Box-Komplexität $(2^n + 1)/2$ für $\mathcal{F} \subseteq \mathcal{T}_{\text{tsif}} \circ \text{NEEDLEPOINTER}$.

Das gewünschte Resultat ist somit gezeigt. □

4.C Rangbasierte randomisierte Suchheuristiken

Einige randomisierte Suchheuristiken besitzen die Eigenschaft, dass sie ausschließlich Komponenten mit folgendem Merkmal verwenden. Diese Komponenten greifen nicht auf den konkreten Funktionswert von Individuen der Population zurück, sondern (höchstens) auf sein Verhältnis zu anderen Funktionswerten – also den Rang des Funktionswerts in der sortierten Folge der unterschiedlichen Funktionswerte, die in der Population auftreten. Zu diesen Komponenten gehören beispielsweise die uniforme, aber auch die Turnier-Selektion (vergleiche Bäck et al., 1997).

Wir formalisieren den Begriff *rangbasiert* (engl. *rank-based*) im Folgenden.

Für eine Folge von t Suchpunkten mit Funktionswerten $((x_1, m_1), \dots, (x_t, m_t)) =: H$ sei $\text{rank}_H(m_i) := |\{m_j \mid m_j \leq m_i, 1 \leq j \leq t\}|$ der Rang von m_i in der Folge H .

Definition (rangbasiert) Seien $H_0 = ((x_1, m_1^0), \dots, (x_t, m_t^0))$ und $H_1 = ((x_1, m_1^1), \dots, (x_t, m_t^1))$ beliebig mit $\text{rank}_{H_0}(m_i^0) = \text{rank}_{H_1}(m_i^1)$ für alle i mit $1 \leq i \leq t$. Sei A eine Suchheuristik und $x \in \{0, 1\}^n$. Falls

$$\Pr[x_{t+1} = x \mid A \text{ mit } H_0] = \Pr[x_{t+1} = x \mid A \text{ mit } H_1]$$

für alle $t \geq 0$ ist, dann heißt A rangbasiert.

Wir bemerken, dass die Historien H_0 und H_1 zu jedem Zeitpunkt die gleichen Suchpunkte beinhalten, nicht aber notwendigerweise die gleichen Funktionswerte.

Beispielsweise besitzen der (1,1)-EA und der (1+1)-EA die Eigenschaft, rangbasiert zu sein.

Theorem 4.C.1 Sei $f : \{0, 1\}^n \rightarrow \mathbb{Z}$ eine beliebige Funktion und A_{rank} eine rangbasierte Suchheuristik. Falls $g \in \mathcal{T}_{\text{tsif}}$ ist, gilt für alle $\ell(n) \geq 0$

$$\Pr[T_{g \circ f, A_{\text{rank}}} > \ell(n)] \leq \Pr[T_{f, A_{\text{rank}}} > \ell(n)].$$

Damit ist auch $E[T_{g \circ f, A}] \leq E[T_{f, A}]$ und die rangbasierte Suchheuristik A_{rank} optimiert die Funktion $g \circ f$ insbesondere dann effizient, wenn sie dies für f tut. Es gilt $\mathcal{T}_{A_{\text{rank}}} \supseteq \mathcal{T}_{\text{tsif}}$.

Beweis (Theorem 4.C.1). Wir betrachten eine beliebige Folge $x_1, \dots, x_{\ell(n)} \in \{0, 1\}^n$, die kein Optimum von $g \circ f$ enthält, also $x_t \in \{0, 1\}^n \setminus \arg \max\{g \circ f(x) \mid x \in \{0, 1\}^n\}$ gilt für $1 \leq t \leq \ell(n)$. Da $g \in \mathcal{T}_{\text{tsif}}$ ist, gilt $\arg \max\{g \circ f(x) \mid x \in \{0, 1\}^n\} \supseteq \arg \max\{f(x) \mid x \in \{0, 1\}^n\}$ und $x_1, \dots, x_{\ell(n)}$ enthält auch keine Optima von f . Weiter gilt, da $g \in \mathcal{T}_{\text{tsif}}$ ist,

$$\text{rank}_{(x_1, g \circ f(x_1)), \dots, (x_{t-1}, g \circ f(x_{t-1}))}(g \circ f(x_i)) = \text{rank}_{(x_1, f(x_1)), \dots, (x_{t-1}, f(x_{t-1}))}(f(x_i))$$

für alle t mit $1 \leq t \leq \ell(n)$ und $1 \leq i \leq t-1$. Da A_{rank} eine rangbasierte Suchheuristik ist, wird x_t , $1 \leq t \leq \ell(n)$, bei der Optimierung von $g \circ f$ mit der gleichen Wahrscheinlichkeitsverteilung gewählt wie x_t , $1 \leq t \leq \ell(n)$, bei der Optimierung von f . Es folgt die Behauptung. \square

Für den im Folgenden definierten rangbasierten evolutionären Algorithmus werden wir exemplarisch nachweisen, dass genau die partiell streng monoton steigenden Funktionen $\mathcal{T}_{\text{tsif}}$ die nicht verschlechternden Transformationen sind. Es gilt $\mathcal{T}_{(\mu+1)\text{-EA}} = \mathcal{T}_{\text{tsif}}$.

$(\mu+1)$ -EA

1. Setze $t := 1$ und wähle $x_{t,1}, \dots, x_{t,\mu} \in \{0, 1\}^n$ zufällig gleichverteilt, aber paarweise verschieden. Setze $\mathcal{P}_t := \{x_{t,1}, \dots, x_{t,\mu}\}$.
2. Setze $t := t + 1$, wähle $x_{t-1} \in \mathcal{P}_{t-1}$ zufällig gleichverteilt und setze $y_t := \text{mutate}_{1/n}(x_{t-1})$.
3. Wähle $z_t \in \arg \min\{f(x) \mid x \in \mathcal{P}_{t-1} \cup \{y_t\}\}$ zufällig gleichverteilt. Falls $y_t \notin \mathcal{P}_{t-1}$, dann setze $\mathcal{P}_t := (\mathcal{P}_{t-1} \cup \{y_t\}) \setminus \{z_t\}$ und sonst $\mathcal{P}_t := \mathcal{P}_{t-1}$.
4. Gehe zu 2.

Der $(\mu+1)$ -EA wurde in Kapitel 3 bereits ausführlich untersucht und wird in Kapitel 5 weiter analysiert.

Die obere Schranke folgt aus Theorem 4.C.1, da der $(\mu+1)$ -EA rangbasiert ist. Für die untere Schranke werden wir nachweisen, dass jede nicht partiell streng monoton steigende Transformation sogar zur totalen Ineffizienz des $(\mu+1)$ -EA führen kann.

Dazu analysieren wir das Verhalten des $(\mu+1)$ -EA auf Plateaus genauer. Wir betrachten mit $\text{SHORTPATHCONSTANT}_a^* : \{0, 1\}^n \rightarrow \mathbb{Z}$ eine Variante von SHORTPATHCONSTANT . Sei mit $a = a_1 \cdots a_n \in \{0, 1\}^n$

$$\text{SHORTPATHCONSTANT}_a^*(x) := \begin{cases} 3n & , \text{ falls } x = p_n, \\ 3n - 2 & , \text{ falls } x = p_i \neq p_n \text{ und für } 0 \leq i < n, \\ < 3n - 2 & , \text{ sonst,} \end{cases}$$

wobei $p_i := a_1 \cdots a_i 0^{n-i}$, $0 \leq i \leq n$, ist. Weiterhin betrachten wir mit $\text{NEEDLE}_{\varepsilon, a}^* : \{0, 1\}^n \rightarrow \mathbb{Z}$ eine Variante von NEEDLE . Sei mit $a \in \{1^{n-\lceil \varepsilon n \rceil} z \mid z \in \{0, 1\}^{\lceil \varepsilon n \rceil}\}$ und einer Konstanten ε , $0 < \varepsilon \leq 1$,

$$\text{NEEDLE}_{\varepsilon, a}^*(x) := \begin{cases} 3n & , \text{ falls } x = a, \\ 3n - 2 & , \text{ falls } x \in \{1^{n-\lceil \varepsilon n \rceil} z \mid z \in \{0, 1\}^{\lceil \varepsilon n \rceil}\} \setminus \{a\}, \\ < 3n - 2 & , \text{ sonst.} \end{cases}$$

Das folgende Resultat ergibt sich analog zu den Betrachtungen von Witt (2006).

Lemma 4.C.2 *Sei $\mu \leq \text{poly}(n)$. Die Population des $(\mu+1)$ -EA enthalte mindestens ein Individuum mit einem Funktionswert von mindestens $3n - 2$.*

- Der $(\mu+1)$ -EA erzeugt das Optimum von $\text{SHORTPATHCONSTANT}_a^*$ in einer erwarteten Laufzeit von $\mathcal{O}(\mu n^3)$.*
- Der $(\mu+1)$ -EA erzeugt das Optimum von $\text{NEEDLE}_{\varepsilon, a}^*$ in einer erwarteten Laufzeit von $\mathcal{O}(\mu 2^{\varepsilon n})$.*

Für die Aussage a) wird dabei das Resultat über die Laufzeit des $(1+1)$ -EA auf kleinen Plateaus nach Jansen und Wegener (2001a) verwendet und für die Aussage b) das Resultat über die Laufzeit des $(1+1)$ -EA auf großen Plateaus nach Garnier et al. (1999). In diesen Situationen genügt dem $(\mu+1)$ -EA eine um den Faktor $\mathcal{O}(\mu)$ größere erwartete Laufzeit zur Optimierung als dem $(1+1)$ -EA.

Theorem 4.C.3 *Sei $\mu \leq \text{poly}(n)$. Falls $g \notin \mathcal{T}_{\text{tsif}}$ ist, dann existiert eine Klasse von Funktionen \mathcal{F} mit den folgenden Eigenschaften:*

- Der $(\mu+1)$ -EA erzeugt ein Optimum von f für alle $f \in \mathcal{F}$ in einer erwarteten Laufzeit von $\mathcal{O}(\mu n^3)$.*
- Der $(\mu+1)$ -EA benötigt für einen Anteil von $1 - 2^{-\Omega(n)}$ aller Funktionen $f \in \mathcal{F}$ mit Wahrscheinlichkeit $1 - 2^{-\Omega(n)}$ eine Laufzeit von mehr als $2^{n/34}$ zur Optimierung von $g \circ f$.*
- Die Black-Box-Komplexität von $g \circ \mathcal{F}$ beträgt mindestens $(2^{\lceil n/32 \rceil} + 1)/2$.*

Beweis. Wir unterscheiden drei Fälle.

Es existiert ein m_1 mit $g(m_1) > g(m_1 + 1)$. (Fall 1)

Existiert ein solcher Wert nicht, dann genügt es, Transformationen mit $g(m) \leq g(m + 1)$ für alle m zu betrachten, wobei für mindestens ein m sogar $g(m) < g(m + 1)$ gilt. Wir unterscheiden hier zwei weitere Fälle.

Es existieren mindestens n Werte m_0, \dots, m_{n-1} mit $m_i < m_{i+1}$ und $g(m_i) = g(m_{i+1})$, $0 \leq i < n - 1$, sowie ein weiterer Wert $m_n > m_{n-1}$ mit $g(m_{n-1}) < g(m_n)$. Also werden m_0, \dots, m_{n-1} zum gleichen, nicht maximalen Funktionswert transformiert. (Fall 2)

Existieren n solche Werte nicht, dann existieren aber mindestens drei Werte m_n, m_{n+1}, m_{n+2} mit $m_n < m_{n+1} < m_{n+2}$ und $g(m_n) = g(m_{n+1}) < g(m_{n+2})$. Also werden m_n, m_{n+1} zum gleichen, nicht maximalen Funktionswert transformiert. Da zusätzlich nicht Fall 2 vorliegt, existieren $\infty > n - 1$ weitere Werte m_1, \dots, m_{n-1} mit $m_i < m_{i+1}$ und $g(m_i) < g(m_{i+1})$, $1 \leq i < n$. (Fall 3)

Im Folgenden sei n groß genug, was für den Beweis der Aussage angenommen werden darf.

1. Wir betrachten die Klasse \mathcal{F}_1 aller Funktionen $f_{1,a} : \{0, 1\}^n \rightarrow \mathbb{Z}$, $a \in \{0, 1\}^n$, mit

$$f_{1,a}(x) := \begin{cases} m_1 & , \text{ falls } x = a, \\ m_1 + 1 & , \text{ sonst.} \end{cases}$$

Für Aussage a) betrachten wir $f_{1,a} \in \mathcal{F}_1$. Wir zeigen, dass der $(\mu+1)$ -EA ein Optimum von $f_{1,a}$ in einer erwarteten Laufzeit von $\mathcal{O}(1)$ erzeugt. Falls $\mu = 1$ und der erste Suchpunkt das einzige nicht optimale Element a ist, dann wird ein Optimum mit Wahrscheinlichkeit $\sum_{i=1}^n \binom{n}{i} \cdot (1/n)^i (1 - 1/n)^{n-i} = 1 - (1 - 1/n)^n \geq 1 - 1/e$ generiert. Dazu genügt eine beliebige mindestens ein Bit flippende Mutation. Sonst wird bei der ersten oder zweiten Funktionsauswertung ein Optimum bestimmt und das gewünschte Resultat folgt.

Für Aussage b) betrachten wir zunächst den $(\mu+1)$ -EA und die konstante Funktion $h : \{0, 1\}^n \rightarrow \mathbb{Z}$ mit

$$h(x) := g(m_1 + 1) \text{ für alle } x \in \{0, 1\}^n.$$

Sei $Z \subseteq \{0, 1\}^n$ die Menge der Suchpunkte, die der $(\mu+1)$ -EA in den ersten $\lfloor 2^{n/34} \rfloor$ Funktionsauswertungen mindestens mit Wahrscheinlichkeit $2^{-n/3}$ erzeugt. Es gilt $|Z| \leq 2^{n/2}$, da der $(\mu+1)$ -EA sonst widersprüchlich mindestens $(2^{n/2} + 1) \cdot 2^{-n/3} > \lfloor 2^{n/34} \rfloor$ Funktionsauswertungen durchgeführt hätte. Also erzeugt der $(\mu+1)$ -EA für jedes $a \notin Z$ mit Wahrscheinlichkeit $1 - 2^{-\Omega(n)}$ eine Folge von Suchpunkten $x_1 \neq a, \dots, x_{\lfloor 2^{n/34} \rfloor} \neq a$. Betrachten wir nun die Menge von Funktion $g \circ \mathcal{F}_1$. Der $(\mu+1)$ -EA erzeugt auch bei der Optimierung von $g \circ f_{1,a}$ eine Folge von Suchpunkten $x_1 \neq a, \dots, x_{\lfloor 2^{n/34} \rfloor} \neq a$ mit Wahrscheinlichkeit $1 - 2^{-\Omega(n)}$, da $h(x) = g \circ f_{1,a}(x)$ ist für $x \neq a$. Wir haben einen Anteil von $(2^n - |Z|)/2^n \geq 1 - 2^{n/2}/2^n = 1 - 2^{-\Omega(n)}$ aller Funktionen $f \in \mathcal{F}_1$ betrachtet. Die Aussage c) folgt direkt nach Theorem 1.A.1 mit $\mathcal{A} = \{0, 1\}^n$.

2. Wir betrachten die Klasse \mathcal{F}_2 aller Funktionen $f_{2,a} : \{0, 1\}^n \rightarrow \mathbb{Z}$, $a \in \{0, 1\}^n$, mit (H bezeichne den Hammingabstand)

$$f_{2,a}(x) := m_{n-H[x,a]} \text{ für alle } x \in \{0, 1\}^n.$$

Für Aussage a) betrachten wir $f_{2,a} \in \mathcal{F}_2$. Wir zeigen, dass der $(\mu+1)$ -EA ein Optimum von $f_{2,a}$ in einer erwarteten Laufzeit von $\mathcal{O}(\sqrt{\mu n} \ln n + \mu n)$ erzeugt. Nach der Transformation durch eine partiell streng monoton steigende Funktion, die für den $(\mu+1)$ -EA nicht verschlechternd ist, entspricht $f_{2,a}$ der Funktion ONEMAX und das gewünschte Resultat folgt nach Theorem 3.A.1.

Für Aussage b) zeigen die gleichen Betrachtungen wie für $g \circ \mathcal{F}_1$ das Ergebnis für $g \circ \mathcal{F}_2$. Die Aussage c) folgt direkt nach Theorem 1.A.1 mit $\mathcal{A} = \{0, 1\}^n$.

3. Wir betrachten die Klasse \mathcal{F}_3 aller Funktionen $f_{3,a} : \{0, 1\}^n \rightarrow \mathbb{Z}$, $a = a_1 \cdots a_n \in \{1^{n-\lfloor n/32 \rfloor} z \mid z \in \{0, 1\}^{\lfloor n/32 \rfloor}\}$, mit

$$f_{3,a}(x) := \begin{cases} m_{n+2} & , \text{ falls } x = p_n, \\ m_{n+1} & , \text{ falls } x = p_i, 0 \leq i < n, \text{ und } p_i \neq p_n, \\ m_n & , \text{ falls } x \in T, \\ m_{n-|x|} & , \text{ sonst,} \end{cases}$$

wobei die Elemente $p_i := a_1 \cdots a_i 0^{n-i}$, $0 \leq i \leq n$, einen *Pfad* (engl. *path*) und die Menge $T := \{1^{n-\lfloor n/32 \rfloor} z \mid z \in \{0, 1\}^{\lfloor n/32 \rfloor}\} \setminus \{p_0, \dots, p_n\}$ eine *Falle* (engl. *trap*) darstellt.

Für Aussage a) betrachten wir $f_{3,a} \in \mathcal{F}_3$. Wir zeigen, dass der $(\mu+1)$ -EA ein Optimum von $f_{3,a}$ in einer erwarteten Laufzeit von $\mathcal{O}(\mu n^3)$ erzeugt. Auf allen Elementen aus $\{0, 1\}^n \setminus (T \cup P)$ entspricht $f_{3,a}$ der Funktion $n - \text{ONEMAX}$ nach der Transformation durch eine partiell streng monoton steigende Funktion, die für den $(\mu+1)$ -EA nicht verschlechternd ist. In einer erwarteten Laufzeit von $\mathcal{O}(\sqrt{\mu n} \ln n + \mu n)$ wird somit ein Element aus $T \cup P$ erzeugt. Sei \mathcal{E} das Ereignis, ein Element aus T vor einem Element aus $P := \{p_i \mid 0 \leq i \leq n\}$ zu erzeugen. Nach Lemma 3.A.3 gilt $|x_{1,1}|, \dots, |x_{1,\mu}| < n - \lfloor n/3 \rfloor$ mindestens mit Wahrscheinlichkeit $1 - \mu 2^{-n/30}$. Danach werden nur Individuen x mit $|x| < n - \lfloor n/3 \rfloor$ oder $x \in T \cup P$ in die Population aufgenommen. Wir betrachten $\lceil 2e\mu n^2 \rceil$ weitere Schritte. Solange die Population kein Element aus $T \cup P$ beinhaltet, beträgt die Wahrscheinlichkeit höchstens $2e\mu n^2 / (\lfloor n/4 \rfloor!) = 2^{-\Omega(n \ln n)}$, in diesen Schritten ein Individuum aus T zu generieren, da mindestens $\lfloor n/4 \rfloor$ (irgendwelche) Bits flippen müssen. Danach und solange die Population kein Element aus $T \cup P$ beinhaltet, wird mindestens mit Wahrscheinlichkeit $1/(e\mu n)$ ein Individuum x' mit $|x'| < \min\{|x| \mid x \in \mathcal{P}\}$ erzeugt und in die Population aufgenommen. Eine Anwendung der Chernoffungleichung zeigt weiter, dass mindestens mit Wahrscheinlichkeit $1 - e^{-2n \cdot (1/2)^2/2} = 1 - e^{-n/4}$ mindestens n solche Mutationen in den betrachteten Schritten auftreten. Dann ist sicher ein Element aus P erzeugt worden. Es gilt also $\Pr[\mathcal{E}] \leq \mu 2^{-n/30} + 2^{-\Omega(n \ln n)} + e^{-n/4} \leq 2^{-n/31}$. Tritt \mathcal{E} dennoch ein, wird nach Lemma 4.C.2.b in einer erwarteten Laufzeit von $\mathcal{O}(\mu 2^{n/32})$ ein Suchpunkt aus P erzeugt, da $f_{3,a}$ auf allen Elementen aus $(\{0, 1\}^n \setminus P) \cup \{a\}$ der Funktion $\text{NEEDLE}_{1/32,a}^*$ entspricht – nach einer geeigneten Transformation. Ist ein Individuum der Population aus P , wird nach Lemma 4.C.2.a in einer erwarteten Laufzeit von $\mathcal{O}(\mu n^3)$ ein Optimum generiert, da $f_{3,a}$ dann auf allen Elementen aus $\{0, 1\}^n$ der Funktion $\text{SHORTPATHCONSTANT}_a^*$ entspricht – nach einer geeigneten Transformation. Insgesamt führt dies zu einer erwarteten Laufzeit von $\mathcal{O}(\sqrt{\mu n} \ln n + \mu n) + 2^{-n/31} \cdot \mathcal{O}(\mu 2^{n/32}) + \mathcal{O}(\mu n^3) = \mathcal{O}(\mu n^3)$ für den

$(\mu+1)$ -EA zur Optimierung von $f_{3,a}$.

Für Aussage b) argumentieren wir wie im Fall 1. Jedoch betrachten wir

$$h(x) := \begin{cases} g(m_{n+2}) & , \text{ falls } x = p_n, \\ g(m_{n+1}) & , \text{ falls } x \in T \cup \{p_i \mid 0 \leq i < n \text{ und } p_i \neq p_n\}, \\ g(m_{n-|x|}) & , \text{ sonst,} \end{cases}$$

und sei $Z \subseteq \{1^{n-\lceil n/32 \rceil} z \mid z \in \{0, 1\}^{\lceil n/32 \rceil}\}$ die Menge der Suchpunkte, die der $(\mu+1)$ -EA in den ersten $\lfloor 2^{n/34} \rfloor$ Funktionsauswertungen mindestens mit Wahrscheinlichkeit $2^{-n/c}$ für eine passende Konstante $c > 0$ erzeugt. Dann gilt $|Z| \leq 2^{n/33}$, wir haben einen Anteil von $(2^{\lceil n/32 \rceil} - |Z|)/2^{\lceil n/32 \rceil} = 1 - 2^{-\Omega(n)}$ aller Funktionen $f \in \mathcal{F}_3$ betrachtet und erhalten so das gewünschte Resultat. sdfgsdfg

Die Aussage c) folgt direkt nach Theorem 1.A.1 mit $\mathcal{A} = \{0, 1\}^{\lceil n/32 \rceil}$. Der Funktionswert jedes Suchpunkts $x \notin \{1^{n-\lceil n/32 \rceil} z \mid z \in \{0, 1\}^{\lceil n/32 \rceil}\}$ ist für jede Funktion $g \circ f$, $f \in \mathcal{F}$, gleich. Es genügt also, Black-Box-Algorithmen, die nur Funktionswerte von Elementen $x \in \{1^{n-\lceil n/32 \rceil} z \mid z \in \{0, 1\}^{\lceil n/32 \rceil}\}$ bestimmen, zu betrachten. Damit ist diese Klasse von Funktionen äquivalent zu NEEDLE mit $\mathcal{A} = \{0, 1\}^{\lceil n/32 \rceil}$.

Das gewünschte Resultat ist somit gezeigt. \square

4.D Nicht funktionswertbasierte randomisierte Suchheuristiken

Einige randomisierte Suchheuristiken besitzen die Eigenschaft, dass sie ausschließlich Komponenten mit folgendem Merkmal verwenden. Diese Komponenten greifen nicht auf den Funktionswert von Individuen der Population zurück.

Wir formalisieren den Begriff *nicht funktionswertbasiert* (engl. *not function-value-based*) im Folgenden.

Definition (nicht funktionswertbasiert) Seien $H_0 = ((x_1, m_1^0), \dots, (x_t, m_t^0))$ und $H_1 = ((x_1, m_1^1), \dots, (x_t, m_t^1))$ beliebig. Sei A eine Suchheuristik und $x \in \{0, 1\}^n$. Falls

$$\Pr[x_{t+1} = x \mid A \text{ mit } H_0] = \Pr[x_{t+1} = x \mid A \text{ mit } H_1]$$

für alle $t \geq 0$ ist, dann heißt A nicht funktionswertbasiert.

Wir bemerken, dass die Historien H_0 und H_1 zu jedem Zeitpunkt die gleichen Suchpunkte beinhalten, nicht aber notwendigerweise die gleichen Funktionswerte.

Einige randomisierte Suchheuristiken wie der (1,1)-EA, nicht jedoch der (1+1)-EA, besitzen die Eigenschaft, nicht funktionswertbasiert zu sein.

Jede nicht funktionswertbasierte Suchheuristik $A_{\text{not } f\text{-value}}$ ist insbesondere rangbasiert und damit gilt $\mathcal{T}_{A_{\text{not } f\text{-value}}} \supseteq \mathcal{T}_{\text{tsif}}$. Wir beweisen sogar $\mathcal{T}_{A_{\text{not } f\text{-value}}} \supseteq \mathcal{T}_{\text{if}}$.

Theorem 4.D.1 Sei $f : \{0, 1\}^n \rightarrow \mathbb{Z}$ eine beliebige Funktion und $A_{\text{not } f\text{-value}}$ eine nicht funktionswertbasierte Suchheuristik. Falls $g \in \mathcal{T}_{\text{if}}$ ist, gilt für alle $\ell(n) \geq 0$

$$\Pr[T_{g \circ f, A_{\text{not } f\text{-value}}} > \ell(n)] \leq \Pr[T_{f, A_{\text{not } f\text{-value}}} > \ell(n)].$$

Beweis. Da $g \in \mathcal{T}_{\text{if}}$ ist, gilt $\arg \max\{g \circ f(x) \mid x \in \{0, 1\}^n\} \supseteq \arg \max\{f(x) \mid x \in \{0, 1\}^n\}$. Da $A_{\text{not } f\text{-value}}$ eine nicht funktionswertbasierte Suchheuristik ist, wird jede Folge $x_1, \dots, x_{\ell(n)} \in \{0, 1\}^n$ bei der Optimierung von $g \circ f$ mit gleicher Wahrscheinlichkeit gewählt wie bei der Optimierung von f . Es folgt die Behauptung. \square

Für die im Folgenden definierte nicht funktionswertbasierte randomisierte Suche A_{random} werden wir exemplarisch nachweisen, dass genau die monoton steigenden Funktionen \mathcal{T}_{if} die nicht verschlechternden Transformationen sind. Es gilt $\mathcal{T}_{A_{\text{random}}} = \mathcal{T}_{\text{if}}$.

A_{random}

1. Setze $t := 0$.
2. Setze $t := t + 1$, wähle $x_t \in \{0, 1\}^n$ zufällig gleichverteilt und bestimme $f(x_t)$.

Die obere Schranke folgt aus Theorem 4.D.1, da der A_{random} nicht funktionswertbasiert ist. Für die untere Schranke werden wir nachweisen, dass jede nicht monoton steigende Transformation sogar zur totalen Ineffizienz des A_{random} führen kann.

Theorem 4.D.2 Falls $g \notin \mathcal{T}_{\text{if}}$ ist, dann existiert eine Klasse von Funktionen \mathcal{F} mit den folgenden Eigenschaften:

- a) A_{random} erzeugt ein Optimum von f für alle $f \in \mathcal{F}$ in einer erwarteten Laufzeit von genau $1/(1 - 2^{-n})$.
- b) A_{random} benötigt mit Wahrscheinlichkeit $1 - 2^{-n/2}$ eine Laufzeit von mehr als $2^{n/2}$ zur Optimierung von $g \circ f$ für alle $f \in \mathcal{F}$.
- c) Die Black-Box-Komplexität von $g \circ \mathcal{F}$ beträgt $(2^n + 1)/2$.

Beweis. Es existiert ein m_1 mit $g(m_1) > g(m_1 + 1)$. Wir betrachten die Klasse \mathcal{F} aller Funktionen $f_a : \{0, 1\}^n \rightarrow \mathbb{Z}$, $a \in \{0, 1\}^n$, mit

$$f_a(x) := \begin{cases} m_1 & , \text{ falls } x = a, \\ m_1 + 1 & , \text{ sonst.} \end{cases}$$

Für Aussage a) betrachten wir $f_a \in \mathcal{F}$. A_{random} erzeugt ein Optimum von f_a in jedem Schritt unabhängig mit Wahrscheinlichkeit $1 - 2^{-n}$.

Für Aussage b) betrachten wir $g \circ f_a \in g \circ \mathcal{F}$. A_{random} erzeugt ein Optimum von $g \circ f_a$ in jedem Schritt unabhängig mit Wahrscheinlichkeit 2^{-n} und somit in $\lfloor 2^{n/2} \rfloor$ Funktionsauswertungen mit Wahrscheinlichkeit $\lfloor 2^{n/2} \rfloor \cdot 2^{-n} \leq 2^{-n/2}$.

Die Aussage c) folgt direkt nach Theorem 1.A.1. \square

4.E Nicht rangbasierte randomisierte Suchheuristiken

Einige randomisierte Suchheuristiken besitzen die Eigenschaft, dass sie auch Komponenten mit folgendem Merkmal verwenden. Diese Komponenten greifen auf den konkreten Funktionswert von Individuen der Population zurück und nicht etwa nur auf ihren Rang. Zu diesen Komponenten gehört beispielsweise die Roulette-Selektion (vergleiche Bäck et al., 1997).

Wir untersuchen nun Suchheuristiken, die nicht rangbasiert sind und damit sicher funktionswertbasiert.

Exemplarisch betrachten wir den im Folgenden definierten nicht rangbasierten evolutionären Algorithmus $(\mu+1)$ -EA[⊗], der die Roulette-Selektion verwendet. Sonst entspricht er dem $(\mu+1)$ -EA, der die uniforme Selektion verwendet. Der $(\mu+1)$ -EA[⊗] ist lediglich anwendbar zur Optimierung von Funktionen $f : \{0, 1\}^n \rightarrow \mathbb{Z}^+$, lässt sich jedoch kanonisch erweitern, so dass er geeignet ist, beliebige Funktionen $f : \{0, 1\}^n \rightarrow \mathbb{Z}$ zu optimieren.

$(\mu+1)$ -EA[⊗]

1. Setze $t := 1$ und wähle $x_{t,1}, \dots, x_{t,\mu} \in \{0, 1\}^n$ zufällig gleichverteilt, aber paarweise verschieden. Setze $\mathcal{P}_t := \{x_{t,1}, \dots, x_{t,\mu}\}$.
2. Setze $t := t + 1$, wähle $x_{t-1} \in \mathcal{P}_{t-1}$ mit Wahrscheinlichkeit $f(x_{t-1}) / \sum_{x \in \mathcal{P}_{t-1}} f(x)$ und setze $y_t := \text{mutate}_{1/n}(x_{t-1})$.
3. Wähle $z_t \in \arg \min\{f(x) \mid x \in \mathcal{P}_{t-1} \cup \{y_t\}\}$ zufällig gleichverteilt. Falls $y_t \notin \mathcal{P}_{t-1}$, dann setze $\mathcal{P}_t := (\mathcal{P}_{t-1} \cup \{y_t\}) \setminus \{z_t\}$ und sonst $\mathcal{P}_t := \mathcal{P}_{t-1}$.
4. Gehe zu 2.

Mit $\mu = 1$ sind der $(\mu+1)$ -EA und der $(\mu+1)$ -EA[⊗] die gleichen Optimierer.

Auch dann, wenn jedes Element in der Population den gleichen Funktionswert hat, sind der $(\mu+1)$ -EA und der $(\mu+1)$ -EA[⊗] gleich. Da jedes Element in der Population mit größtem Funktionswert mindestens mit Wahrscheinlichkeit $1/\mu$ vom $(\mu+1)$ -EA[⊗] zur Mutation ausgewählt wird, folgt direkt das Lemma 4.C.2 und dann auch das Theorem 4.C.3 ebenfalls für den $(\mu+1)$ -EA[⊗] gelten. Damit ist $\mathcal{T}_{(\mu+1)\text{-EA}^\otimes} \subseteq \mathcal{T}_{\text{tsif}}$ gezeigt. Wir demonstrieren sogar $\mathcal{T}_{(\mu+1)\text{-EA}^\otimes} \subset \mathcal{T}_{\text{tsif}}$.

Dazu betrachten wir die Funktion $\text{SHORTPATHMANYPEAKS}_1 + n^5 =: \text{SHORTPATHMANYPEAKS}^*$, wobei die Funktion $\text{SHORTPATHMANYPEAKS}_1$ in Abschnitt 3.C definiert ist. Die Funktion $\text{SHORTPATHMANYPEAKS}^*$ entspricht der Funktion $\text{SHORTPATHMANYPEAKS}_1$ nach der Transformation durch eine (partiell) streng monotone steigende Funktion. Der $(\mu+1)$ -EA mit $2 \leq \mu \leq \text{poly}(n)$ optimiert diese Funktion also effizient. Da für jedes Element $x \in \{0, 1\}^n$ $n^5 \leq \text{SHORTPATHMANYPEAKS}^*(x) \leq n^5 + 3n^4$ gilt, wird jedes Individuum in jeder Population mit Wahrscheinlichkeit $\Theta(1/\mu)$ vom $(\mu+1)$ -EA[⊗] mit $\mu \leq \text{poly}(n)$ zur Mutation ausgewählt. Nach Lemma 3.C.4.a folgt direkt Aussage a) des folgenden Theorems.

Weiter betrachten wir die folgende Funktion $g : \mathbb{Z} \rightarrow \mathbb{Z}$ mit

$$g(m) := \begin{cases} m & , \text{ falls } m < 0, \\ 2^{m^2} & , \text{ sonst.} \end{cases}$$

Die Transformation g ist (partiell) streng monoton steigend.

Betrachten wir nun $g \circ \text{SHORTPATHMANYPEAKS}^*$, die der $(\mu+1)$ -EA mit $2 \leq \mu \leq \text{poly}(n)$ effizient, der $(1+1)$ -EA nach Lemma 3.C.4.b jedoch nicht effizient optimiert. Analog zum Beweis dieses Lemmas gilt für den $(\mu+1)$ -EA[⊗] mit $\mu \leq \text{poly}(n)$ und bei der Optimierung

von $g \circ \text{SHORTPATHMANYPEAKS}^*$ das Folgende mit Wahrscheinlichkeit $1 - 2^{-\Omega(n)}$. In den ersten $2^{\varepsilon n}$ Schritten für eine genügend kleine Konstante $\varepsilon > 0$ wird kein Element des Pfades erzeugt – dann bleibt nichts zu zeigen – oder eines der ersten n Elemente des Pfades wird vor einem anderen Element des Pfades generiert. Wir nehmen dies an. Sei \mathcal{P} eine Population, die mindestens ein Element des Pfades enthält. Jedes Element des Pfades hat einen unterschiedlichen Funktionswert. Sei x das Element der Population \mathcal{P} mit größtem Funktionswert 2^{m^2} . Dann wählt der $(\mu+1)$ -EA[⊗] bei Optimierung von $g \circ \text{SHORTPATHMANYPEAKS}^*$ im nachfolgenden Schritt das Individuum x mindestens mit Wahrscheinlichkeit

$$1 - \frac{(\mu - 1) \cdot 2^{(m-1)^2}}{2^{m^2}} = 1 - (\mu - 1) \cdot 2^{-2m+1} \geq 1 - (\mu - 1) \cdot 2^{-2n^5+1} \geq 1 - 2^{-n^5}$$

zur Mutation aus. In jedem von $2^{\varepsilon n}$ Schritten passiert jenes mindestens mit Wahrscheinlichkeit $1 - 2^{\varepsilon n} \cdot 2^{-n^5} = 1 - 2^{-\Omega(n)}$. Wir nehmen dies an. Die Wahrscheinlichkeit, $x' \in \{0, 1\}^n$ zu generieren, ist dann für den $(\mu+1)$ -EA mit Population \mathcal{P} und den $(1+1)$ -EA mit Element x gleich. Lediglich falls $g \circ \text{SHORTPATHMANYPEAKS}^*(x') > g \circ \text{SHORTPATHMANYPEAKS}^*(x)$ ist, wird x' das Element der Population des $(\mu+1)$ -EA[⊗] mit größtem Funktionswert und des $(1+1)$ -EA. In dieser Situation erzeugen also der $(\mu+1)$ -EA[⊗] und der $(1+1)$ -EA das Optimum in höchstens $2^{\varepsilon n}$ Schritten mit gleicher Wahrscheinlichkeit. Nach Lemma 3.C.4.b folgt direkt Aussage b) des folgenden Theorems.

Theorem 4.E.1

- Sei $2 \leq \mu \leq \text{poly}(n)$. Der $(\mu+1)$ -EA[⊗] erzeugt das Optimum von $\text{SHORTPATHMANYPEAKS}^*$ in einer erwarteten Laufzeit von $\mathcal{O}(\mu n^5)$.
- Sei $1 \leq \mu \leq \text{poly}(n)$. Mit Wahrscheinlichkeit $1 - 2^{-\Omega(n)}$ benötigt der $(\mu+1)$ -EA[⊗] eine Laufzeit von $2^{\varepsilon n}$ zur Optimierung von $g \circ \text{SHORTPATHMANYPEAKS}^*$ für eine Konstante $\varepsilon > 0$.

In diesem Kapitel haben wir insgesamt insbesondere gezeigt:

$$\begin{array}{cccc}
 \mathcal{T}_A & & \mathcal{T}_{A_{\text{rank}}} & & \mathcal{T}_{A_{\text{not } f\text{-value}}} & & \mathcal{T}_A \\
 \cup \mid & & \cup \mid & & \cup \mid & & \cap \mid \\
 \mathcal{T}_{\text{ti}} & \subset & \mathcal{T}_{\text{tsif}} & \subset & \mathcal{T}_{\text{if}} & \subset & \mathcal{T}_{\text{af}} \\
 \parallel & & \parallel & & \parallel & & \parallel \\
 \mathcal{T}_{A_{\text{ti}}} & & \mathcal{T}_{(\mu+1)\text{-EA}} & & \mathcal{T}_{A_{\text{random}}} & & \mathcal{T}_{A_{\text{af}}} \\
 & & \cup & & & & \\
 & & \mathcal{T}_{(\mu+1)\text{-EA}^{\otimes}} & & & &
 \end{array}$$

5 CLIQUENPROBLEM UND RANDOMISIERTE SUCHHEURISTIKEN

Wir analysieren, auf welche Art randomisierte Suchheuristiken das Cliquenproblem lösen. Dieses algorithmische Problem besteht im Auffinden einer größten Clique in einem (einfachen) ungerichteten Graphen (vergleiche Wegener, 2005). Eine *Clique* (engl. *clique*) ist eine Teilmenge von Knoten des Graphen, in der je zwei (verschiedene) Knoten durch eine Kante verbunden sind. Eine *größte Clique* (engl. *maximum clique*) ist eine Clique maximaler Kardinalität. Eine große Clique in einem Graphen zu finden, ist sowohl von praktischem als auch von theoretischem Interesse (vergleiche Bomze et al., 1999; Johnson und Trick, 1996).

Das Cliquenproblem gehört zu den ersten Optimierungsproblemen, die als NP-hart klassifiziert wurden (Karp, 1972). Daher können wir nicht auf Algorithmen hoffen, die (1) in polynomieller Laufzeit (2) eine größte Clique (3) für jeden Graphen berechnen. Jede dieser drei Bedingungen können wir relaxieren.

- (1) Eine größte Clique und für jeden Graphen berechnen die besten bekannten exakten Algorithmen in einer Laufzeit von $\mathcal{O}(1,189^n)$ und exponentiellem Speicherplatz oder in einer Laufzeit von $\mathcal{O}(1,203^n)$ und polynomielltem Speicherplatz (Robson, 2001).
- (2) In polynomieller Laufzeit und für jeden Graphen berechnet der beste bekannte approximative Algorithmus eine Clique, deren Größe um einen Faktor $\mathcal{O}(n(\ln \ln n)^2 / \ln^3 n)$ von der einer größten Clique abweicht (Feige, 2004). Unter vernünftigen Komplexitätstheoretischen Annahmen ist eine effiziente Approximation um einen Faktor von $n/e^{\ln^{3/4+\varepsilon} n}$ für jede Konstante $\varepsilon > 0$ nicht möglich (Khot und Ponnuswami, 2006).
- (3) In polynomieller Laufzeit und für jeden planaren Graphen (diese können in die Ebene gezeichnet werden, ohne dass zwei Kanten sich scheiden) berechnet ein optimaler Algorithmus eine größte Clique (Eppstein, 1999). (Dieser Algorithmus benötigt sogar lediglich eine lineare Laufzeit $\mathcal{O}(n)$.)

Hier definiert eine „schwerste“ Eingabe die (erwartete) Laufzeit des Algorithmus. Wir betrachten weiterhin die (erwartete) Laufzeit eines Algorithmus für „typische“ Eingaben, was seine Performanz in Anwendungen verdeutlichen soll. Ein Eingabemodell versucht „typische“ Eingaben durch Verteilungen über (alle) Eingaben – durch zufällige Eingaben – zu beschreiben. Dies führt von Worst- zu Average-Case-Betrachtungen. Wir bemerken, dass dabei insbesondere wieder die obigen Relaxierungen vorgenommen werden können. Betrachten wir für das Cliquenproblem zufällige Graphen (engl. *random graphs*), diese enthalten jede Kante mit Wahrscheinlichkeit $1/2$, was der Gleichverteilung bezüglich aller Graphen entspricht (vergleiche Bollobás, 1985). Wird jede Kante mit Wahrscheinlichkeit wesentlich kleiner als $1/2$ als im Graphen enthalten klassifiziert, so resultieren spärliche zufällige Graphen

(engl. sparse random graphs). Der beste bekannte Algorithmus findet mit hoher Wahrscheinlichkeit $1 - o(1)$ (bezüglich des Zufalls der Eingabe) in einer Laufzeit von $n^{\Theta(\ln n)}$ eine größte Clique eines zufälligen Graphen und in polynomieller Laufzeit eine Clique, deren Kardinalität um einen Faktor nahe 2 von der einer größten Clique abweicht (Frieze und McDiarmid, 1997). Wird eine Clique der Größe $k(n) = \Omega(n^{1/2})$ in einen zufälligen Graphen eingepflanzt, also alle (fehlenden) Kanten zwischen $k(n)$ zufällig gleichverteilt gewählten Knoten in den Graphen aufgenommen, sind Algorithmen mit polynomieller Laufzeit bekannt, die mit hoher Wahrscheinlichkeit (bezüglich des Zufalls der Eingabe) eine größte Clique finden (Alon et al., 1998). Bereits für Werte $k(n)$ unwesentlich größer als $2 \log_2 n$ – dies entspricht mit hoher Wahrscheinlichkeit asymptotisch der Kardinalität einer größten Clique in einem zufälligen Graphen – ist die eingepflanzte auch die größte Clique.

Jerrum (1992) hat das Verhalten einer randomisierten Suchheuristik, genauer eines Metropolis-Algorithmus, für solche Eingaben analysiert. Er hat insbesondere bewiesen, dass (auch) der betrachtete Optimierer für zufällige Graphen, ohne und mit eingepflanzter Clique, mit hoher Wahrscheinlichkeit (bezüglich des Zufalls der Eingabe) nicht effizient eine Clique der Größe mindestens $(1 + \varepsilon) \log_2 n$ für jede Konstante $\varepsilon > 0$ erzeugt. Wir gehen an geeigneter Stelle näher auf den Metropolis-Algorithmus für das Cliquesproblem ein.

Ein Zusammenhang zwischen zufälligen und „typischen“ Eingaben ist höchst schwierig zu begründen, da zufällige Eingaben üblicherweise sehr gleichförmig sind, „typische“ Eingaben jedoch nicht notwendigerweise. Weitere Eingabemodelle versuchen daher „typische“ Eingaben durch eine Kombination von „schwersten“ und zufälligen Eingaben – durch semizufällige Eingaben – zu beschreiben. Dies führt von Worst- und Average-Case-Betrachtungen zu Semi-Average-Case-Betrachtungen. Dabei unterscheiden wir zwei Typen von Eingabemodellen (Blum und Spencer, 1995). Bei der ersten Variante ist es einem Gegner erlaubt, eine beliebige Eingabe zu präsentieren, diese wird jedoch in Grenzen zufällig verändert. Bei der zweiten Variante ist es einem Gegner erlaubt, eine zufällige Eingabe in Grenzen zu verändern. Algorithmen sind bekannt, die eine größte Clique mit hoher Wahrscheinlichkeit (bezüglich des Zufalls der Eingabe) auch dann noch in zufälligen Graphen mit eingepflanzter Clique der Größe $k(n) = \Omega(n^{1/2})$ finden, wenn es einem Gegner gestattet ist, beliebige, jedoch nicht zur eingepflanzten Clique gehörende Kanten zu entfernen (vergleiche Feige und Kilian, 2001). Es kann sogar zertifiziert werden, dass die gefundene Clique in solchen semizufälligen Graphen (engl. semi-random graphs) eine größte Clique ist (Feige und Krauthgamer, 2000). Hier bleibt die eingepflanzte Clique die größte Clique des Graphen und die Anzahl Cliques kann von einem Gegner nur reduziert werden. Einem Gegner wird lediglich erlaubt, das Problem „einfacher“ zu gestalten. Jedoch kann es einem Gegner auch erlaubt werden, das Problem „(moderat) schwerer“ zu gestalten. Kanonischerweise kann es gestattet werden, einige beliebige Kanten in einen zufälligen Graphen einzufügen. Hier kann ein Gegner sowohl die Anzahl Cliques des Graphen als auch die Kardinalität einer größten Clique sogar erhöhen. Dabei sind speziell spärliche semizufällige Graphen von Interesse, da viele in Anwendungen auftretende Graphen zwar spärlich sind, jedoch weit davon entfernt, spärlich zufällig zu sein. Dies motiviert die Betrachtung eines solchen mächtigen Gegners und zeigt die Robustheit von Algorithmen gegenüber Veränderungen des Graphen.

Sollen Suchheuristiken eine große Clique in einem Graphen $G = (V, E)$ berechnen, muss zunächst eine passende – und effizient auswertbare – Zielfunktion entworfen werden (vergleiche Jerrum, 1992). Der Suchbereich $\{0, 1\}^n$ mit $n = |V|$ stellt eine kanonische Wahl dar. Ein Element wird als charakteristischer Vektor der Knoten des Graphen $\{v_1, \dots, v_n\} := V$ in einer beliebigen, aber festen Reihenfolge interpretiert. Wählen wir die Größe der Clique als Funktionswert, falls die gewählten Knoten eine Clique repräsentieren, ist das Ziel die Maximierung. Wählen wir $-\infty$ (oder -1) als Funktionswert, falls die gewählten Knoten keine Clique repräsentieren, führt dies für viele typische randomisierte Suchheuristiken, die insbesondere die ersten Suchpunkte zufällig gleichverteilt wählen, zur totalen Ineffizienz – bereits beim Auffinden einer beliebigen Clique im leeren Graphen, welcher keine Kanten enthält. Indem die ersten Suchpunkte etwa lediglich aus Nullen bestehen und somit die 0-Clique repräsentieren, kann dies vermieden werden oder durch das Verwenden der kanonischen Zielfunktion $\text{CLIQUE}_G : \{0, 1\}^n \rightarrow \mathbb{Z}$ mit $x = x_1 \cdots x_n \in \{0, 1\}^n$ und

$$\text{CLIQUE}_G(x) := \begin{cases} +|x| & , \text{ falls } \{v_i \mid x_i = 1, 1 \leq i \leq n\} \text{ eine Clique ist,} \\ -|x| & , \text{ falls } \{v_i \mid x_i = 1, 1 \leq i \leq n\} \text{ keine Clique ist.} \end{cases}$$

Die Funktion CLIQUE_G kann für jeden Graphen G effizient ausgewertet werden. Und ist die Funktion CLIQUE_G optimiert oder der Funktionswert $k(n) \geq 0$ erreicht, so ist eine größte Clique von G oder eine Clique der Kardinalität $k(n)$ repräsentierendes Individuum erzeugt worden. Das eine *Clique repräsentierende Element* wird ebenfalls als *Clique* bezeichnet. Weiter sei $\mathcal{C}^{\geq k(n)}(G)$ die Menge aller Cliquen der Größe mindestens $k(n)$ des Graphen G und $\omega(G)$ bezeichne die Kardinalität der größten Cliquen.

Der Einsatz einer großen Population *und* eines globalen Suchoperators gelten als die wesentlichsten Merkmale eines evolutionären Algorithmus (vergleiche Schwefel, 1995). Wir werden im Folgenden die Effekte einer Kombination dieser beiden Komponenten (in der kombinatorischen Optimierung) untersuchen. Während für ein Element $x \in \{0, 1\}^n$ der globale Suchoperator $\text{mutate}_{1/n}(x)$ jedes Bit in x unabhängig mit Wahrscheinlichkeit $1/n$ flippt, flippt der lokale Suchoperator $\text{mutate}_1^*(x)$ genau ein zufällig gleichverteilt gewähltes Bit in x . Beide Operatoren flippen im Erwartungswert exakt ein Bit.

$(\mu+1)$ -EA/RLS

1. Setze $t := 1$ und wähle $x_{t,1}, \dots, x_{t,\mu} \in \{0, 1\}^n$ zufällig gleichverteilt, aber paarweise verschieden. Setze $\mathcal{P}_t := \{x_{t,1}, \dots, x_{t,\mu}\}$.
2. Setze $t := t + 1$ und wähle $x_{t-1} \in \mathcal{P}_{t-1}$ zufällig gleichverteilt.
 $(\mu+1)$ -EA: Setze $y_t := \text{mutate}_{1/n}(x_{t-1})$.
 $(\mu+1)$ -RLS: Setze $y_t := \text{mutate}_1^*(x_{t-1})$.
3. Wähle $z_t \in \arg \min\{f(x) \mid x \in \mathcal{P}_{t-1} \cup \{y_t\}\}$ zufällig gleichverteilt. Falls $y_t \notin \mathcal{P}_{t-1}$, dann setze $\mathcal{P}_t := (\mathcal{P}_{t-1} \cup \{y_t\}) \setminus \{z_t\}$ und sonst $\mathcal{P}_t := \mathcal{P}_{t-1}$.
4. Gehe zu 2.

Für $\mu = 2^n$ sind der $(\mu+1)$ -EA und die $(\mu+1)$ -RLS die gleichen Optimierer. Der $(\mu+1)$ -EA wurde bereits in den Kapiteln 3 und 4 ausführlich untersucht.

Diese randomisierten Suchheuristiken erlauben eine Analyse der Effekte der Wahl eines lokalen oder globalen Suchoperators sowie der Wahl der Elternpopulationsgröße und vermeiden zugleich Einflüsse weiterer Komponenten solcher Heuristiken. Die Resultate für den $(\mu+1)$ -EA werden daher insbesondere mit denen für die $(\mu+1)$ -RLS und den $(1+1)$ -EA verglichen.

5.A Approximierung für alle Graphen

Wir beginnen unsere Betrachtungen für Suchheuristiken auf allen Graphen und sind an approximativen Lösungen für das Cliquesproblem interessiert – Ergebnisse für exakte Lösungen folgen hieraus direkt. Sei G ein beliebiger Graph auf n Knoten. Da 0^n die 0-Clique darstellt und, solange keine Clique generiert wurde, die zu optimierende Funktion $-\text{ONEMAX}$ entspricht, erzeugt der $(\mu+1)$ -EA nach Theorem 3.A.1 in einer erwarteten Laufzeit von $\mathcal{O}(\sqrt{\mu n} \ln n + \mu n)$ eine Clique von G und nimmt diese in die Population \mathcal{P} auf. Sei $\ell := \min\{|x| \mid x \in \mathcal{P}\} \leq \omega(G)$. Danach und solange die Population weder die 0-Clique enthält noch nur aus Cliques besteht, wird mindestens mit Wahrscheinlichkeit $\ell/(e\mu n)$ ein Element mit weniger als ℓ Einsen erzeugt und in die Population aufgenommen. Ein solches Individuum erzeugen wenigstens die ℓ speziellen 1-Bit-Mutationen, die genau eine Eins eines Elements $x \in \mathcal{P}$ mit $|x| = \ell$ zu einer Null flippen. Und weiter hat jeder eine Clique repräsentierender Suchpunkt einen größeren Funktionswert als jeder Suchpunkt, der keine Clique darstellt. In einer erwarteten Laufzeit von $\sum_{\ell=1}^{\omega(G)} e\mu n/\ell = \mathcal{O}(\mu n(1 + \ln \omega(G)))$ beinhaltet die Population die 0-Clique oder sie enthält nur Cliques. Sei C eine größte Clique von G und $\ell := \max\{|x| \mid x \in \mathcal{P}, x \subseteq C\} \geq 0$. Danach und solange die Population weder eine größte Clique enthält noch nur aus Cliques besteht, wird mindestens mit Wahrscheinlichkeit $(\omega(G) - \ell)/(e\mu n)$ eine Teilmenge von C der Größe mindestens $\ell + 1$ erzeugt und in die Population aufgenommen. In einer erwarteten Laufzeit von $\sum_{\ell=0}^{\omega(G)-1} e\mu n/(\omega(G) - \ell) = \mathcal{O}(\mu n(1 + \ln \omega(G)))$ beinhaltet die Population eine größte Clique oder sie enthält nur Cliques.

Wir bemerken weiter, falls $\omega(G) = 1$ und $\mu \geq 2$ ist, dann umfasst die Population sicher eine größte Clique. Falls $\omega(G) = 1$ und $\mu = 1$ ist, ist dazu höchstens eine beliebige 1-Bit-Mutation notwendig, welche mindestens Wahrscheinlichkeit $1 - 1/e$ hat. Falls $\omega(G) \geq 2$ und $\mu \geq \mathcal{C}^{\geq 0}(G) - 2^{\omega(G)} + 1$ ist, umfasst die Population dann immer mindestens eine Clique x mit $x \subseteq C$, da C genau $2^{\omega(G)}$ Teilmengen enthält. Analog zu den vorigen Betrachtungen wird in einer erwarteten Laufzeit von $\mathcal{O}(\mu n(1 + \ln \omega(G)) + \sqrt{\mu n} \ln n)$ eine größte Clique erzeugt.

Da lediglich 1-Bit-Mutationen betrachtet wurden, von denen jede bei der $(\mu+1)$ -RLS ebenfalls mit Wahrscheinlichkeit $\Theta(1/(\mu n))$ auftritt, gelten die gemachten Beobachtungen nicht nur für den $(\mu+1)$ -EA, sondern auch für die $(\mu+1)$ -RLS.

Lemma 5.A.1 *Sei G ein beliebiger Graph auf n Knoten. Der $(\mu+1)$ -EA und die $(\mu+1)$ -RLS (auf CLIQUE_G) erzeugen in einer erwarteten Laufzeit von $\mathcal{O}(\mu n(1 + \ln \omega(G)) + \sqrt{\mu n} \ln n)$*

- a) *eine größte Clique von G oder eine Population nur bestehend aus Cliques und*
- b) *eine größte Clique von G , falls $\mu \geq |\mathcal{C}^{\geq 0}(G)| - 2^{\omega(G)} + 1$ ist.*

Wir betrachten den Graphen $G = (\{v_1, \dots, v_n\}, \{\{v_i, v_j\} \mid 1 \leq i < j \leq k(n)\})$, $2 \leq k(n) \leq n$, dessen größte Clique $\{v_1, \dots, v_{k(n)}\}$ ist. Nach Lemma 5.A.1 erzeugt die $(\mu+1)$ -RLS mit $\mu \geq n - k(n) + 1$ in einer erwarteten Laufzeit von $\mathcal{O}(\mu n \ln n)$ die größte Clique von G . Dazu benötigt die $(\mu+1)$ -RLS mit $\mu < n - k(n) + 1$ eine unendliche erwartete Laufzeit, da mit positiver Wahrscheinlichkeit die erste Population nur aus den Cliques $\{v_{k(n)+1}\}, \dots, \{v_{\mu+k(n)}\}$ besteht. Im Anschluss führen keine 1-Bit-Mutationen zu Veränderungen der Population. Wir analysieren die erwartete Laufzeit des $(\mu+1)$ -EA zur Erzeugung einer großen Clique, ohne notwendigerweise eine große Population einzusetzen. Eine *maximale Clique* ist eine Clique, die in keiner größeren Clique vollständig enthalten ist.

Theorem 5.A.2 *Sei G ein beliebiger Graph auf n Knoten. Der $(\mu+1)$ -EA (auf CLIQUE_G) erzeugt eine Clique von G der Größe mindestens $s(n) \leq \omega(G)$ in einer erwarteten Laufzeit von*

$$\mathcal{O}\left(n^{2s(n)-2} / \binom{\omega(G)}{s(n)-1} + \mu n \ln n\right).$$

Beweis. Der $(\mu+1)$ -EA erzeugt nach Lemma 5.A.1.a eine größte Clique von G oder eine Population nur bestehend aus Cliques in einer erwarteten Laufzeit von $\mathcal{O}(\mu n \ln n)$. Danach und solange die Population keine Clique mindestens der Größe $s(n)$ enthält, umfasst sie lediglich Cliques der Größe höchstens $s(n) - 1$. Im Graphen G existieren mindestens $\binom{\omega(G)}{s(n)-1}$ nicht maximale $(s(n) - 1)$ -Cliques, da dies bereits der Anzahl $(s(n) - 1)$ -Cliques entspricht, die Teilmengen einer größten Clique sind. Da $|x| \leq s(n) - 1$ für jedes $x \in \mathcal{P}$ ist und jede nicht maximale $(s(n) - 1)$ -Clique aus $s(n) - 1$ Einsen besteht, wird jede solche Clique durch eine spezielle Mutation von x generiert, die höchstens $|x| + (s(n) - 1) \leq 2s(n) - 2$ Bits flippt. Beinhaltet eine Population nur $(s(n) - 1)$ -Cliques, wird eine nicht maximale $(s(n) - 1)$ -Clique mit Wahrscheinlichkeit $\mu/(\mu + 1) \geq 1/2$ in sie aufgenommen und sonst sicher. In einer erwarteten Laufzeit von $\mathcal{O}(n^{2s(n)-2} / \binom{\omega(G)}{s(n)-1})$ existiert daher eine nicht maximale $(s(n) - 1)$ -Clique in der Population. In jedem Schritt wird dann mindestens mit Wahrscheinlichkeit $1/(e\mu n)$ eine Clique der Größe zumindest $s(n)$ erzeugt, da wenigstens eine spezielle 1-Bit-Mutation einer nicht maximalen $(s(n) - 1)$ -Clique dazu genügt. Zeigen wir, dass dann in jedem Schritt höchstens mit Wahrscheinlichkeit $2/(\mu n)$ eine nur aus maximalen $(s(n) - 1)$ -Cliques bestehende Population entsteht, folgt für die erwartete Laufzeit, bis eine Clique der Größe mindestens $s(n)$ generiert wird, $E \leq 1 + \frac{1}{e\mu n} \cdot 0 + \frac{2}{\mu n} \cdot \mathcal{O}(n^{2s(n)-2} / \binom{\omega(G)}{s(n)-1}) + (1 - \frac{2}{\mu n} - \frac{1}{e\mu n}) \cdot E$ und äquivalent $E = \mathcal{O}(\mu n + n^{2s(n)-2} / \binom{\omega(G)}{s(n)-1})$. Die genannte erwartete Laufzeit folgt direkt.

Beinhaltet die Population mehr als eine nicht maximale $(s(n) - 1)$ -Clique oder Cliques kleiner $s(n) - 1$, entsteht sicher keine nur aus maximalen $(s(n) - 1)$ -Cliques bestehende Population.

Beinhaltet die Population nur $(s(n) - 1)$ -Cliques, von denen genau eine nicht maximal ist, entsteht eine nur aus maximalen $(s(n) - 1)$ -Cliques bestehende Population lediglich, wenn eine maximale $(s(n) - 1)$ -Clique erzeugt wird und dann die nicht maximale Clique entfernt wird, was Wahrscheinlichkeit $1/(\mu + 1) \leq 1/\mu$ besitzt. Beschränken wir nun die Wahrscheinlichkeit durch $2/n$ nach oben, eine maximale $(s(n) - 1)$ -Clique zu generieren. Sei V_1 eine beliebige Clique – hier der Größe $s(n) - 1$. Für jede Menge $S_0 \subseteq V \setminus V_1$ existiert höchstens eine Menge $S_1 \subseteq V_1$ mit $|S_0| = |S_1|$, so dass $(V_1 \setminus S_1) \cup S_0$ eine maximale Clique der Größe $|V_1|$ bildet – hier der Größe $s(n) - 1$. Nehmen wir an, zwei unterschiedliche Mengen S_1

und S_2 mit $|S_1| = |S_2|$ existieren für die dies gilt. Dann sei $a \in S_2$, aber $a \notin S_1$, und damit bildet $(V_1 \setminus S_1) \cup S_0 \cup \{a\}$ eine Clique der Größe $|V_1| + 1$ – hier der Größe $s(n)$ – was ein Widerspruch ist, da $(V_1 \setminus S_1) \cup S_0$ als maximal angenommen wurde. Für jedes Individuum der Population beträgt die Wahrscheinlichkeit höchstens $\sum_{k=1}^{s(n)-1} \binom{n}{k} \cdot (1/n)^{2k} (1-1/n)^{n-2k} \leq \sum_{k \geq 1} n^k \cdot 1/n^{2k} = \sum_{k \geq 1} 1/n^k \leq 2/n$, eine maximale $(s(n) - 1)$ -Clique zu generieren. \square

Wir kombinieren die Ansätze des Lemmas 5.A.1.b und des Theorems 5.A.2. Genauer vereinigen wir die Stärken einer Population mit denen eines globalen Suchoperators. Dazu zeigen wir, warum es von Vorteil sein kann, wenn eine Population kleine Cliquen enthält. Sei $|\mathcal{C}^{\geq s(n)}(G)| \geq 1$ für ein beliebig gewähltes $s(n)$. Nach Lemma 5.A.1.a erzeugt der $(\mu+1)$ -EA in einer erwarteten Laufzeit von $\mathcal{O}(\mu n \ln n)$ eine größte Clique oder eine Population nur bestehend aus Cliquen. Danach und falls $\mu \geq 2|\mathcal{C}^{\geq s(n)}(G)|$ ist, existieren mindestens $\mu/2$ Cliquen in der Population, die höchstens Größe $s(n) - 1$ besitzen. Ein solches Individuum wird mindestens mit Wahrscheinlichkeit $(\mu/2)/\mu \geq 1/2$ zur Mutation ausgewählt. Höchstens $(s(n) - 1) + s(n) = 2s(n) - 1$ Bits eines solchen Elements müssen flippen, um eine Clique der Größe mindestens $s(n)$ zu erzeugen, welche die Teilmenge einer größten Clique ist. In einer erwarteten Laufzeit von $\mathcal{O}(n^{2s(n)-1})$ wird diese Clique generiert und nicht mehr aus der Population entfernt, da $\mu \geq |\mathcal{C}^{\geq s(n)}(G)|$ ist. Analog zum Beweis des Lemmas 5.A.1.a wird dann in einer erwarteten Laufzeit von $\sum_{\ell=s(n)}^{\omega(G)-1} e\mu n / (\omega(G) - \ell) = \mathcal{O}(\mu n (1 + \ln \omega(G)))$ eine größte Clique erzeugt.

Lemma 5.A.3 Sei G ein beliebiger Graph auf n Knoten und $\mu \geq \max\{2|\mathcal{C}^{\geq s(n)}(G)|, 1\}$ für beliebiges $s(n)$. Der $(\mu+1)$ -EA (auf CLIQUE_G) erzeugt eine größte Clique von G in einer erwarteten Laufzeit von $\mathcal{O}(n^{2s(n)-1} + \mu n \ln n)$.

Auf der einen Seite benötigt die $(\mu+1)$ -RLS nach Lemma 5.A.1.b eine erwartete Laufzeit von $\mathcal{O}(\mu n \ln n)$, um eine Clique der Größe mindestens $s(n)$ zu generieren, falls $\mu \geq 1$ und $s(n) \leq 1$ ist oder falls $\mu \geq \sum_{i=0}^{s(n)-1} \binom{n}{i} - 1$ und $s(n) \geq 2$ ist, wobei des Weiteren gilt $\sum_{i=1}^{s(n)-1} \binom{n}{i} \leq n^{s(n)-1} / (s(n) - 2)! \leq \left(\frac{3n}{s(n)-1}\right)^{s(n)-1}$. Auf der anderen Seite werden wir Graphen präsentieren, für welche die $(\mu+1)$ -RLS mit $\mu < \left(\frac{n/2}{s(n)-1}\right)^{s(n)-1}$ eine unendliche erwartete Laufzeit zur Erzeugung einer Clique der Größe mindestens $s(n) \geq 2$ benötigt. Der $(1+1)$ -EA benötigt hierzu eine erwartete Laufzeit von $\Omega(n^{2s(n)-2} / \binom{\omega(G)}{s(n)-1} + n \ln n)$ für $s(n) \geq 0$. Dazu beweisen wir jedoch zunächst ein technisches Lemma, dass uns gestattet, für viele Graphen Aussagen über das erste eine Clique repräsentierende Individuum des $(1+1)$ -EA zu treffen.

Lemma 5.A.4 Für $x = x_1 \cdots x_n \in \{0, 1\}^n$ und eine Konstante $0 < \varepsilon \leq 1/2$ sei $\text{left}(x) := |x_1 \cdots x_{\lfloor \varepsilon n \rfloor}|$ und $\text{right}(x) := |x_{\lfloor \varepsilon n \rfloor + 1} \cdots x_n|$. Sei $f : \{0, 1\}^n \rightarrow \mathbb{Z}$ eine beliebige Funktion mit

- $f(x) = -|x|$ für alle x mit $\text{left}(x) \geq 1$ und $\text{right}(x) \geq 1$ und
- $f(x) = +|x|$ für alle x mit $\text{left}(x) + \text{right}(x) \leq 1$.

Für den $(1+1)$ -EA bei Optimierung von f sei x^* das erste Individuum mit $f(x^*) > 0$. Dann gilt $\Pr[x^* = 0^{i-1} 10^{n-i}] = \Omega(1/n)$ für alle i mit $1 \leq i \leq n$.

Beweis. Zunächst betrachten wir den (1+1)-EA bei Optimierung von $-\text{ONEMAX}$. Droste et al. (2002) haben festgestellt, dass dann zu jedem Zeitpunkt jedes Element mit gleicher Anzahl Einsen mit gleicher Wahrscheinlichkeit sowohl erzeugt wird als auch die Population darstellt. Wenn die Population erstmals ein Individuum x mit $|x| = k \geq 3$ ist, gilt daher $\text{left}(x) \leq 1$ oder $\text{right}(x) \leq 1$ mit Wahrscheinlichkeit

$$\frac{\binom{\lfloor \varepsilon n \rfloor}{0} \binom{n - \lfloor \varepsilon n \rfloor}{k} + \binom{\lfloor \varepsilon n \rfloor}{1} \binom{n - \lfloor \varepsilon n \rfloor}{k-1} + \binom{\lfloor \varepsilon n \rfloor}{k-1} \binom{n - \lfloor \varepsilon n \rfloor}{1} + \binom{\lfloor \varepsilon n \rfloor}{k} \binom{n - \lfloor \varepsilon n \rfloor}{0}}{\binom{n}{k}} \\ \leq 4 \cdot \frac{\binom{n - \lfloor \varepsilon n \rfloor}{k}}{\binom{n}{k}} \leq 4 \cdot \left(\frac{n - \lfloor \varepsilon n \rfloor}{n} \right)^k = 2^{-\Omega(k)}.$$

Gilt für das Individuum x also $\text{left}(x) \geq 2$ und $\text{right}(x) \geq 2$, müssen mindestens zwei spezielle Bits in $x_1 \cdots x_{\lfloor \varepsilon n \rfloor}$ oder $x_{\lfloor \varepsilon n \rfloor + 1} \cdots x_n$ flippen, um ein x' mit $\text{left}(x') = 0$ oder $\text{right}(x') = 0$ zu generieren. Die Wahrscheinlichkeit, dass dies passiert, bevor ein Element mit weniger als k Einsen erzeugt wird, beträgt höchstens

$$2 \cdot \frac{1/n^2}{1/n^2 + k/(en)} + 4 \cdot \frac{1/n}{1/n + k/(en)} \cdot \frac{1/n}{1/n + k/(en)} = 2 \cdot \frac{e}{e + kn} + 4 \cdot \left(\frac{e}{e + k} \right)^2 = \mathcal{O}(1/k^2).$$

Dies gilt, da mit $k \leq n$ auch $\mathcal{O}(1/(kn)) = \mathcal{O}(1/k^2)$ ist und da vor einer von k speziellen 1-Bit-Mutationen, die eine Eins in x flippt,

- eine von zwei Mutationen durchgeführt werden muss, die mindestens zwei spezielle Bits in x flippt oder
- eine von vier Mutationen durchgeführt werden muss, die ein spezielles Bit in x flippt und dann eine Mutation, die mindestens ein weiteres spezielles Bit flippt.

Sei c eine genügend große Konstante abhängig von ε . Betrachten wir den ersten Zeitpunkt t' mit $|x_{t'}| < c$. Jedes solche Element mit der gleichen Anzahl an Einsen tritt mit gleicher Wahrscheinlichkeit auf. Zuvor wurden mindestens mit Wahrscheinlichkeit

$$1 - \sum_{k=c}^n (2^{-\Omega(k)} + \mathcal{O}(1/k^2)) \geq 1/2$$

nur Elemente x mit $\text{left}(x) \geq 1$ und $\text{right}(x) \geq 1$ generiert, wobei die obige Ungleichung sowohl $\sum_{k=1}^{\infty} 2^{-k} = 1$ als auch $\sum_{k=1}^{\infty} 1/k^2 = \pi^2/6$ ausnutzt. Bei diesen Elementen unterscheidet sich der Funktionswert von $-\text{ONEMAX}$ und f nicht. Daher erzeugt der (1+1)-EA auch bei Optimierung von f ein solches Individuum $x_{t'}$ mindestens mit Wahrscheinlichkeit $1/2$.

Gilt $|x_{t'}| = 0$, wird jedes $0^{i-1}10^{n-i}$, $1 \leq i \leq n$, bei nachfolgender Mutation mit Wahrscheinlichkeit $\Omega(1/n)$ generiert.

Gilt $|x_{t'}| = 1$, tritt jedes $0^{i-1}10^{n-i}$, $1 \leq i \leq n$, mit Wahrscheinlichkeit $1/\binom{n}{1} = \Omega(1/n)$ auf.

Gilt $2 \leq |x_{t'}| = k < c$, ist das i -te Bit von $x_{t'}$ Eins, $\text{left}(x_{t'}) \geq 1$ und $\text{right}(x_{t'}) \geq 1$ mit Wahrscheinlichkeit $((\binom{n-1}{k-1} - \binom{\lfloor \varepsilon n \rfloor - 1}{k-1})) / \binom{n}{k} = \Omega(1/n)$, falls $i \leq \lfloor \varepsilon n \rfloor$ (Fall 1) ist, und mit Wahrscheinlichkeit $((\binom{n-1}{k-1} - \binom{n - \lfloor \varepsilon n \rfloor - 1}{k-1})) / \binom{n}{k} = \Omega(1/n)$, falls $i > \lfloor \varepsilon n \rfloor$ (Fall 2) ist. Dann existiert ein i' , so dass das i' -te Bit von $x_{t'}$ Eins ist mit $i' > \lfloor \varepsilon n \rfloor$ im Fall 1 und mit $i' \leq \lfloor \varepsilon n \rfloor$ im Fall 2.

Sei x ein Suchpunkt, der aus $x_{i'}$ durch das Flippen von j Einsen entsteht, jedoch nicht des i -ten und i' -ten Bits. Damit gilt $\text{left}(x) \geq 1$ und $\text{right}(x) \geq 1$. Falls $j < k - 2$ ist, wird mindestens mit Wahrscheinlichkeit $(k - j - 2)/(en)$ ein solches Element mit $j - 1$ Einsen erzeugt. Falls $j = k - 2$ ist, wird mindestens mit Wahrscheinlichkeit $1/(en)$ das Element $0^{i-1}10^{n-i}$ generiert. Für $j \leq k - 2$ wird höchstens mit Wahrscheinlichkeit $(k - j)/n$ eine andere Mutation durchgeführt, die mindestens eines der $k - j$ Bits in x flippt, die Eins sind. Dies ist notwendig, um einen Suchpunkt x' mit $|x'| < |x| - 1$ oder $f(x') \neq -|x'|$ zu erzeugen. Mindestens mit Wahrscheinlichkeit

$$\Omega(1/n) \cdot \left(\prod_{j=0}^{k-3} \frac{(k-j-2)/(en)}{(k-j-2)/(en) + (k-j)/n} \right) \cdot \frac{1/(en)}{1/(en) + 2/n} \geq \Omega(1/n) \cdot \Omega(1)^{c-2} \cdot \Omega(1) = \Omega(1/n)$$

wird somit $0^{i-1}10^{n-i}$ als erstes Element mit positivem Funktionswert erzeugt. \square

Wir können nun die Schärfe des Lemmas 5.A.1.b und des Theorems 5.A.2 zeigen.

Theorem 5.A.5 *Es existieren Graphen G mit $\omega(G) \leq n/2$, so dass Folgendes gilt.*

- Die $(\mu+1)$ -RLS mit $\mu < (\ell+1)^{s(n)-1}$, $\ell := \lfloor n/(2s(n)-2) \rfloor \geq 1$, (auf CLIQUE_G) benötigt eine unendliche erwartete Laufzeit zur Erzeugung einer Clique von G der Größe mindestens $s(n)$ mit $2 \leq s(n) \leq \omega(G)$.
- Der $(1+1)$ -EA (auf CLIQUE_G) benötigt eine erwartete Laufzeit von

$$\Omega\left(n^{2s(n)-2} / \binom{\omega(G)}{s(n)-1} + n \ln n\right)$$

zur Erzeugung einer Clique von G der Größe mindestens $s(n)$ mit $0 \leq s(n) \leq \omega(G)$.

Beweis. Für eine größte Clique der Kardinalität $k(n)$ sei sowohl $V_i := \{v_{(i-1)\ell+1}, \dots, v_{i\ell}\}$, $1 \leq i < s(n)$, $E_0 := \{\{v_i, v_j\} \mid v_i \in V_i, v_j \in V_j, 1 \leq i < j < s(n)\}$ als auch $E_1 := \{\{v_i, v_j\} \mid n-k(n) < i < j \leq n\}$ für $k(n) \geq s(n)$. Wir betrachten den Graphen $W_{n,k(n),s(n)-1}^* := (V, E_0 \cup E_1)$, wobei $V := \bigcup_{i=1}^{s(n)-1} V_i$. Es existieren $\sum_{i=0}^{s(n)-1} \binom{s(n)-1}{i} \cdot \ell^i = (\ell+1)^{s(n)-1}$ Cliques der Größen $0, \dots, s(n) - 1$ auf dem durch $V_1 \cup \dots \cup V_{s(n)-1}$ induzierten Graphen, jedoch keine größeren Cliques. Die größte Clique ist $\{v_{n-k(n)+1}, \dots, v_n\}$ und damit gilt $\omega(W_{n,k(n),s(n)-1}^*) = k(n)$.

- Mit positiver Wahrscheinlichkeit besteht die erste Population nur aus den Cliques der Größe mindestens eins auf dem durch $V_1 \cup \dots \cup V_{s(n)-1}$ induzierten Graphen. Danach hat jedes Element einen Hammingabstand von mindestens zwei zu jeder Clique, die Teilmenge der größten Clique ist – mit Ausnahme der 0-Clique, die jedoch nicht in die Population aufgenommen wird. Es resultiert eine unendliche erwartete Laufzeit.
- Nach Lemma 5.A.4 mit $\varepsilon = 1/2$ wird jede 1-Clique als die Population des $(1+1)$ -EA mit Wahrscheinlichkeit $\Omega(1/n)$ erzeugt, da insbesondere keine Kanten zwischen Knoten aus $V_1 \cup \dots \cup V_{s(n)-1}$ und $\{v_{n-k(n)+1}, \dots, v_n\}$ verlaufen. Droste et al. (2002) haben speziell gezeigt, dass der $(1+1)$ -EA eine erwartete Laufzeit von $\Omega(n \ln n)$ benötigt, um ein irgendein Individuum $0^{i-1}10^{n-i}$, $1 \leq i \leq n$, zu erzeugen. Da ein solches Element mit Wahrscheinlichkeit $\Omega(1/n) \cdot n = \Omega(1)$ generiert wird, resultiert eine erwartete Laufzeit von

$\Omega(1) \cdot \Omega(n \ln n) = \Omega(n \ln n)$, erstmals eine Clique zu erzeugen. Die gewünschte Aussage für $s(n) \leq 1$ folgt direkt. Sei $s(n) \geq 2$. Mit Wahrscheinlichkeit $\ell \cdot (s(n) - 1) \cdot \Omega(1/n) = \Omega(1)$ wird eine 1-Clique $C \subseteq \{v_1, \dots, v_{\ell \cdot (s(n) - 1)}\}$ als Population des (1+1)-EA erzeugt. Wenn das Individuum des (1+1)-EA eine nicht leere Clique $C \subseteq \{v_1, \dots, v_{\ell \cdot (s(n) - 1)}\}$ repräsentiert, wird höchstens mit Wahrscheinlichkeit $1/n^{|C|}$ eine Clique C' mit $C' \subseteq \{v_{\ell \cdot (s(n) - 1) + 1}, \dots, v_n\}$ erzeugt, da $C \cap C' = \emptyset$ ist. Dagegen wird mindestens mit Wahrscheinlichkeit $1/(en)$ eine größere Clique auf dem durch $\{v_1, \dots, v_{\ell \cdot (s(n) - 1)}\}$ induzierten Graphen generiert, falls $|C| < s(n) - 1$ ist. Da $\sum_{i=1}^{s(n)-2} \frac{1/n^i}{1/n^i + 1/(en)} = \sum_{i=1}^{s(n)-2} \frac{e}{e + n^{i-1}} \leq 3/4$ ist für n groß genug, beträgt die Wahrscheinlichkeit $\Omega(1) \cdot 1/4 = \Omega(1)$, eine $(s(n) - 1)$ -Clique $C \subseteq \{v_1, \dots, v_{n - \ell \cdot (s(n) - 1)}\}$ vor einer nicht leeren Clique $C' \subseteq \{v_{n - \ell \cdot (s(n) - 1) + 1}, \dots, v_n\}$ zu generieren. Im Anschluss beträgt die Wahrscheinlichkeit, eine nicht maximale $(s(n) - 1)$ -Clique oder eine Clique der Größe mindestens $s(n)$ zu generieren, da $\binom{k(n)}{s(n) - 1 + i} \leq \binom{k(n)}{s(n) - 1} n^i$ und $\sum_{i \geq 0} 1/n^i \leq 2$ ist, höchstens

$$\sum_{i=s(n)-1}^{k(n)} \binom{k(n)}{i} \cdot 1/n^{2i} \leq \sum_{i=0}^{\infty} \binom{k(n)}{s(n) - 1} n^i \cdot 1/n^{2(s(n) - 1 + i)} \leq \binom{k(n)}{s(n) - 1} \cdot 2/n^{2s(n) - 2}.$$

Das gewünschte Resultat ist somit gezeigt. \square

Eine größte Clique und das für jeden Graphen berechnen die $(2^n + 1)$ -RLS und der $(2^n + 1)$ -EA in einer Laufzeit von höchstens 2^n . Für einige Graphen (und n groß genug) dagegen benötigt dazu die $(\mu + 1)$ -RLS noch für $\mu \leq 2^{n/2 - 3}$ eine unendliche erwartete Laufzeit und der (1+1)-EA eine erwartete Laufzeit von mindestens n^{n-5} . Eine Clique, deren Größe um einen Faktor von höchstens n/c für eine beliebig große Konstante c von der einer größten Clique abweicht, berechnet der $(\mu + 1)$ -EA mit $\mu \leq \text{poly}(n)$ in erwarteter polynomieller Laufzeit (abhängig von c) und für jeden Graphen. Für einige Graphen dagegen ist eine Approximation um einen Faktor von $o(n)$ für die $(\mu + 1)$ -RLS und den (1+1)-EA nicht effizient möglich.

5.B Optimierung für spärliche Graphen

Bevor wir uns den spärlichen semizufälligen Graphen widmen, betrachten wir die spärlichen Graphen und sind an optimalen Lösungen für das Cliquesproblem interessiert. Wir beobachten folgenden Zusammenhang zwischen der Anzahl Kanten im Graphen und der möglichen Anzahl Cliques.

Lemma 5.B.1 Sei G ein beliebiger Graph auf n Knoten mit $|E| \leq \binom{k(n)}{2}$. Es gilt $|\mathcal{C}^{\geq 0}(G)| \leq 2^{k(n)} - k(n) + n$ für $k(n) \geq 0$ und $|\mathcal{C}^{\geq k(n)}(G)| \leq 1$ für $k(n) \geq 2$.

Beweis. Die zweite Aussage folgt direkt, da $\binom{k(n)}{2}$ Kanten benötigt werden, um eine $k(n)$ -Clique zu bilden, falls $k(n) \geq 2$ ist.

Für die erste Aussage bemerken wir, dass G sicher genau eine 0-Clique und n viele 1-Cliques umfasst. Es genügt also $|\mathcal{C}^{\geq 2}(G)| \leq 2^{k(n)} - k(n) - 1$ per Induktion über $k(n)$ zu zeigen.

Sei $0 \leq k(n) \leq 1$. Der Graph G umfasst $2^0 - 0 - 1 = 2^1 - 1 - 1 = 0$ Cliques der Größe mindestens zwei.

Sei $k(n) \geq 2$. Wir unterscheiden, ob der Graph G eine $k(n)$ -Clique (Fall 1) oder keine $k(n)$ -Clique (Fall 2) umfasst.

1. Der Graph G beinhaltet $\sum_{i=2}^{k(n)} \binom{k(n)}{i} = 2^{k(n)} - k(n) - 1$ Cliques der Größe mindestens zwei.
2. Sei $G_0 := G$. Weiter entstehe G_i aus G_{i-1} indem alle Kanten entfernt werden, die zu einem ausgewählten Knoten $v_{i'}$ aus G_{i-1} mit kleinstem, aber positivem Grad $\deg(v_{i'}) > 0$ inzident sind. Sei $\ell \geq 0$ der kleinste Wert, für den G_ℓ höchstens $\binom{k(n)-1}{2} = \binom{k(n)}{2} - (k(n) - 1)$ Kanten enthält. Da nicht Fall 1 vorliegt, gilt $\deg(v_{i'}) \leq k(n) - 2$ und weiter $\sum_{i=1}^{\ell-1} \deg(v_{i'}) \leq k(n) - 2$ für $i \geq 1$. Das Entfernen der Kanten von $v_{i'}$ eliminiert höchstens $\binom{\deg(v_{i'})}{j}$ mit $1 \leq j \leq \deg(v_{i'})$ viele $(j+1)$ -Cliques. Bis G_ℓ entstanden ist, wurden somit höchstens $\sum_{i=1}^{\ell} \sum_{j=1}^{\deg(v_{i'})} \binom{\deg(v_{i'})}{j} = \sum_{i=1}^{\ell} (2^{\deg(v_{i'})} - 1) \leq 2^{\sum_{i=1}^{\ell-1} \deg(v_{i'})} + 2^{\deg(v_{i'})} - 1 \leq 2^{k(n)-2} + (2^{k(n)-2} - 1) = 2^{k(n)-1} - 1$ Cliques der Größe mindestens zwei entfernt. Der Graph G_ℓ umfasst nach Voraussetzung höchstens $2^{k(n)-1} - (k(n) - 1) - 1$ Cliques der Größe mindestens zwei. Insgesamt beinhaltet G daher höchstens $(2^{k(n)-1} - 1) + (2^{k(n)-1} - (k(n) - 1) - 1) = 2^{k(n)} - k(n) - 1$ Cliques der Größe mindestens zwei.

Das gewünschte Resultat ist somit gezeigt. \square

Für $k(n) = n$ ergeben sich beliebige Graphen. Wir können die Betrachtungen des vorherigen Abschnitts übertragen und erhalten das folgende Ergebnis mit Lemma 5.B.1 sowie Lemma 5.A.1.b und Theorem 5.A.2.

Theorem 5.B.2 Sei G ein beliebiger Graph auf n Knoten mit $|E| \leq \binom{k(n)}{2}$.

- a) Der $(\mu+1)$ -EA und die $(\mu+1)$ -RLS mit $\mu \geq 2^{k(n)} - k(n) + n$ (auf CLIQUE_G) erzeugen eine größte Clique von G in einer erwarteten Laufzeit von $\mathcal{O}(\mu n \ln k(n) + \sqrt{\mu} n \ln n)$.
- b) Der $(1+1)$ -EA (auf CLIQUE_G) erzeugt eine größte Clique von G in einer erwarteten Laufzeit von $\mathcal{O}(n^{2^{k(n)-2}}/k(n))$.

Analog zum Theorem 5.A.5 zeigen wir nun die Schärfe des Theorems 5.B.2.

Theorem 5.B.3 Es existieren Graphen G auf n Knoten mit $|E| \leq \binom{k(n)}{2}$ und $3 \leq k(n) \leq n/\sqrt{2}$, so dass Folgendes gilt.

- a) Die $(\mu+1)$ -RLS mit $\mu < 2^{\lfloor k(n)/\sqrt{2} \rfloor - 1} - 2\lfloor k(n)/\sqrt{2} \rfloor + n$ (auf CLIQUE_G) benötigt eine unendliche erwartete Laufzeit zur Erzeugung einer größten Clique von G .
- b) Der $(1+1)$ -EA (auf CLIQUE_G) benötigt eine erwartete Laufzeit von $\Omega(n^{\sqrt{2}k(n)-2}/k(n))$ zur Erzeugung einer größten Clique von G .

Beweis. Wir betrachten den Graphen $W_{n,k(n)} := (V, \{\{v_i, v_j\} \mid 1 \leq i < j \leq \lfloor k(n)/\sqrt{2} \rfloor - 1 \text{ oder } n - \lfloor k(n)/\sqrt{2} \rfloor + 1 \leq i < j \leq n\})$. Dieser Graph beinhaltet $\binom{\lfloor k(n)/\sqrt{2} \rfloor - 1}{2} + \binom{\lfloor k(n)/\sqrt{2} \rfloor}{2} \leq \binom{k(n)}{2}$ Kanten.

- a) Betrachten wir die Cliques der Größe mindestens eins auf dem durch $\{v_1, \dots, v_{\lfloor k(n)/\sqrt{2} \rfloor - 1}, v_{\lfloor k(n)/\sqrt{2} \rfloor}, \dots, v_{n - \lfloor k(n)/\sqrt{2} \rfloor}\}$ induzierten Graphen. Mit positiver Wahrscheinlichkeit besteht die erste Population nur aus solchen Cliques und das Ergebnis ergibt sich analog zum Beweis des Theorems 5.A.5.a.
- b) Falls das Individuum des (1+1)-EA eine 1-Clique $\{v_{\lfloor k(n)/\sqrt{2} \rfloor}, \dots, v_{n - \lfloor k(n)/\sqrt{2} \rfloor}\}$ repräsentiert, wird eine 1-Clique $\{v_1, \dots, v_{\lfloor k(n)/\sqrt{2} \rfloor - 1}\}$ mindestens mit Wahrscheinlichkeit

$$\frac{(\lfloor k(n)/\sqrt{2} \rfloor - 1)/(en^2)}{(\lfloor k(n)/\sqrt{2} \rfloor - 1)/(en^2) + \lfloor k(n)/\sqrt{2} \rfloor/n^2} = \Omega(1)$$

vor einer nicht leeren Clique auf dem durch $\{v_{n - \lfloor k(n)/\sqrt{2} \rfloor + 1}, \dots, v_n\}$ induzierten Graphen erzeugt. Insgesamt wird analog zum Beweis des Theorems 5.A.5.b mit Wahrscheinlichkeit $(\lfloor k(n)/\sqrt{2} \rfloor - 1) \cdot \Omega(1/n) + (n - (2\lfloor k(n)/\sqrt{2} \rfloor - 1)) \cdot \Omega(1/n) \cdot \Omega(1) = \Omega(1)$ eine nicht leere Clique auf dem durch $\{v_1, \dots, v_{\lfloor k(n)/\sqrt{2} \rfloor - 1}\}$ induzierten Graphen vor einer nicht leeren Clique auf dem durch $\{v_{n - \lfloor k(n)/\sqrt{2} \rfloor + 1}, \dots, v_n\}$ induzierten Graphen generiert. Das Ergebnis ergibt sich analog zum Beweis des Theorems 5.A.5.b.

Das gewünschte Resultat ist somit gezeigt. \square

Wenn die Population groß genug ist, erzeugen also der $(\mu+1)$ -EA und die $(\mu+1)$ -RLS für jeden Graphen mit $\mathcal{O}(\ln^2 n)$ Kanten effizient eine größte Clique. Für einige Graphen mit bereits $\omega(1)$ Kanten generiert der (1+1)-EA dagegen lediglich ineffizient eine größte Clique. Schon dies zeigt exemplarisch die mögliche wesentliche Steigerung der Effizienz randomisierter Suchheuristiken durch den Einsatz von großen Populationen (in der kombinatorischen Optimierung).

5.C Optimierung für spärliche semizufällige Graphen

Wir widmen uns nun den am Anfang dieses Kapitels motivierten Semi-Average-Case-Betrachtungen und sind an optimalen Lösungen für das Cliquesproblem interessiert.

5.C.1 Kanten entfernen

Wir beginnen unsere Untersuchungen für Suchheuristiken auf (spärlichen) semizufälligen Graphen, bei denen einem Gegner nur erlaubt wird, das Problem „einfacher“ zu gestalten. Dazu präzisieren wir das Eingabemodell der (spärlichen) zufälligen Graphen mit eingepflanzter Clique $\mathcal{G}_{n,p(n),k(n)}$, was eine Verteilung über alle Graphen definiert. Hierfür sei $V = \{v_1, \dots, v_n\}$ die Menge der n Knoten des Graphen und $P \subseteq V$ zufällig gleichverteilt gewählt mit $|P| = k(n)$. Die Kanten $\{v_i, v_j\}$ mit $1 \leq i < j \leq n$ werden unabhängig voneinander mit Wahrscheinlichkeit

- 1, falls $v_i, v_j \in P$ ist, und (die Menge dieser Kanten sei mit E_P bezeichnet)
- $p(n)$, sonst (die Menge dieser Kanten sei mit E_{-P} bezeichnet)

in den Graphen eingefügt. Für $k(n) = n$ erhalten wir den vollständigen Graphen. Für $k(n) = 0$ erhalten wir das Eingabemodell der (spärlichen) zufälligen Graphen (ohne eingepflanzte Clique). Nach Frieze und McDiarmid (1997) beträgt die Kardinalität einer größten Clique eines solchen Graphen mit hoher Wahrscheinlichkeit asymptotisch

$$\frac{2 \ln((1 - p(n)) \cdot n)}{\ln \frac{1}{p(n)}} =: c_{n,p(n)}$$

solange $0 < p(n) = 1 - \omega(1/n)$ ist. Für $p(n) = n^{-\varepsilon(n)}$ mit $\varepsilon(n) > 0$ entspricht $c_{n,n^{-\varepsilon(n)}}$ annähernd $2/\varepsilon(n)$. Wir betrachten das folgende Eingabemodell der (spärlichen) semizufälligen Graphen $\mathcal{G}_{n,p(n),k(n)}^*$ (vergleiche Feige und Kilian, 2001). Sei G ein Graph bezüglich $\mathcal{G}_{n,p(n),k(n)}$ gewählt. Diesen Graphen nennen wir *unbehandelt*. Einem Gegner ist es dann erlaubt, beliebige Kanten aus E_{-p} von G zu entfernen. Diesen Graphen nennen wir *behandelt*. Ein Gegner kann somit jeden Graphen zwischen $(V, E_p \cup E_{-p})$ und (V, E_p) erzeugen.

Untersuchen wir randomisierte Suchheuristiken auf (semi)zufälligen Eingaben, ist zu unterscheiden, ob eine Wahrscheinlichkeit bezüglich des Zufalls der Suchheuristik oder bezüglich des Zufalls der (unbehandelten) Eingabe betrachtet wird.

Theorem 5.C.1 *Sei G ein Graph bezüglich $\mathcal{G}_{n,n^{-\varepsilon(n)},k(n)}^*$ gewählt. Der $(\mu+1)$ -EA (auf CLIQUE_G) erzeugt eine größte Clique von G in einer erwarteten (bezüglich des Zufalls der Suchheuristik) Laufzeit von*

- a) $\mathcal{O}(n^{9/\varepsilon(n)} + \mu n \ln n)$ mit Wahrscheinlichkeit $1 - n^{-\Omega(1/\varepsilon(n))}$ und
- b) $\mathcal{O}(n^{12/\varepsilon(n)+5} + \mu n \ln n)$ im Erwartungswert

(bezüglich des Zufalls der Eingabe).

Beweis. Betrachten wir die Zufallsvariablen X_i , $i \geq 1$, welche die Anzahl Paare disjunkter Mengen S, T mit $|S| = |T| = i$, $S \subseteq V \setminus P$, $T \subseteq V$ und $\{\{s, t\} \mid s \in S, t \in T\} \subseteq E$ bezüglich des unbehandelten Graphen beschreibt. Um die Wahrscheinlichkeit der Existenz eines solchen Paares zu beschränken, verwenden wir die Methode des ersten Moments. Genauer zeigt eine Anwendung der Markoffungleichung, dass $\Pr[X_i \geq 1] \leq E[X_i]/1 \leq \binom{n-k(n)}{i} \binom{n}{i} n^{-\varepsilon(n) \cdot i^2} \leq n^{2i-\varepsilon(n) \cdot i^2}$ da $\binom{n-k(n)}{i} \leq \binom{n}{i} \leq n^i$ ist. Nehmen wir $X_i = 0$ an, existiert auch im behandelten Graphen G kein solches Paar von Mengen der Größe i . Sei C eine größte Clique in G und wir betrachten eine beliebige Clique C' mit $|C'| < |C|$. Eine Clique der Größe mindestens $|C'| + 1$, die Teilmenge von C ist, wird mit Wahrscheinlichkeit $\Omega(n^{-2(|C' \setminus C|+1)})$ durch Mutation von C' erzeugt. Wir unterscheiden $|C'| < 2i$ (Fall 1) und $|C'| \geq 2i$ (Fall 2).

1. Es gilt $|C' \setminus C| \leq |C'|$ und damit $\Omega(n^{-2(|C' \setminus C|+1)}) = \Omega(n^{-(2(2i-1)+1)}) = \Omega(n^{-4i+1})$.
2. Es gilt $|C' \setminus C| = |C' \setminus (C \cup P)| + |(C' \cap P) \setminus C| \leq |C' \setminus P| + |P \setminus C|$. Weiter ist $|C' \setminus P| \leq i - 1$, da sonst widersprüchlich zu $X_i = 0$ die Mengen $S \subseteq C' \setminus P \subseteq V \setminus P$ und $T \subseteq C' \setminus S \subseteq V$ der Größe i existieren würden. Und da $|C| \geq |P|$ ist, gilt $|P \setminus C| \leq |C \setminus P| \leq i - 1$. Also ist $|C' \setminus P| + |P \setminus C| \leq 2i - 2$ und damit $\Omega(n^{-2(|C' \setminus C|+1)}) = \Omega(n^{-(2(2i-2)+1)}) = \Omega(n^{-4i+1})$.

Nach Lemma 5.A.1 erzeugt der $(\mu+1)$ -EA in einer erwarteten Laufzeit von $\mathcal{O}(\mu n \ln n)$ eine größte Clique oder eine Population nur bestehend aus Cliques. Nehmen wir an, die Population beinhaltet ein $C' \subseteq C$. Dann wird eine Teilmenge von C der Größe mindestens $|C'| + 1$ mit

Wahrscheinlichkeit $\Omega((|C| - |C'|)/(\mu n))$ erzeugt. Dagegen wird C' höchstens aus der Population entfernt, wenn die Population nur Cliques der Größe mindestens $|C'|$ umfasst. Dann wird wiederum eine Teilmenge von C der Größe mindestens $|C'| + 1$ mit Wahrscheinlichkeit $\Omega(n^{-4i+1})$ erzeugt. Also generiert jeder Schritt in irgendeiner dieser beiden Situationen mit $|C'| =: \ell$ mindestens mit Wahrscheinlichkeit $\min\{\Omega((|C| - \ell)/(\mu n)), \Omega(n^{-4i+1})\}$ eine Clique der Größe wenigstens $\ell + 1$, die Teilmenge von C ist. Zu Beginn unserer Betrachtungen gilt sicher $\ell \geq 0$ und es ist $|C| \leq n$. Insgesamt führt dies zu einer erwarteten Laufzeit von

$$\begin{aligned} & \mathcal{O}(\mu n \ln n) + \sum_{\ell=0}^{|C|-1} \max\left\{\mathcal{O}\left(\frac{\mu n}{|C| - \ell}\right), \mathcal{O}(n^{4i-1})\right\} \\ & \leq \mathcal{O}(\mu n \ln n) + \mathcal{O}\left(\sum_{\ell=0}^{n-1} \mu n / (n - \ell)\right) + \mathcal{O}\left(\sum_{\ell=0}^{n-1} n^{4i-1}\right) \leq \mathcal{O}(n^{4i} + \mu n \ln n), \end{aligned}$$

falls $X_i = 0$ ist.

- a) Wie bemerkt, ist $\Pr[X_{9/(4\varepsilon(n))} = 0] \geq 1 - n^{2 \cdot 9/(4\varepsilon(n)) - \varepsilon(n) \cdot 81/(16\varepsilon(n)^2)} = 1 - n^{-9/(16\varepsilon(n))}$ und mit den obigen Betrachtungen folgt direkt das Ergebnis.
- b) Mit $p_i := \Pr[X_{i-1} \geq 1 \text{ und } X_i = 0]$ erhalten wir eine erwartete Laufzeit von höchstens

$$\begin{aligned} & \sum_{i=1}^n p_i \cdot (\mathcal{O}(n^{4i} + \mu n \ln n)) = \sum_{i=1}^n p_i \cdot \mathcal{O}(n^{4i}) + \sum_{i=1}^n p_i \cdot \mathcal{O}(\mu n \ln n) \\ & = \sum_{i=1}^{3/\varepsilon(n)} 1 \cdot \mathcal{O}(n^{4(3/\varepsilon(n))}) + \sum_{i=3/\varepsilon(n)+1}^n n^{2(i-1) - \varepsilon(n) \cdot (i-1)^2} \cdot \mathcal{O}(n^{4i}) + 1 \cdot \mathcal{O}(\mu n \ln n) \\ & = \sum_{i=1}^{3/\varepsilon(n)} \mathcal{O}(n^{12/\varepsilon(n)}) + \sum_{i=0}^{n-3/\varepsilon(n)-1} \mathcal{O}(n^{9/\varepsilon(n) - \varepsilon(n) \cdot i^2 + 4}) + \mathcal{O}(\mu n \ln n) \\ & = \mathcal{O}(n^{12/\varepsilon(n)+1}) + \sum_{i=0}^n \mathcal{O}(n^{9/\varepsilon(n)+4}) + \mathcal{O}(\mu n \ln n) = \mathcal{O}(n^{12/\varepsilon(n)+5} + \mu n \ln n). \end{aligned}$$

Das gewünschte Resultat ist somit gezeigt. \square

Wir zeigen nun die Schärfe des Theorems 5.C.1.

Theorem 5.C.2 Sei G ein Graph bezüglich $\mathcal{G}_{n, n^{-\varepsilon(n)}, k(n)}^*$ gewählt sowie $1/\log_2 n \leq \varepsilon(n) \leq 1/4$ und $3/\varepsilon(n) \leq k(n) \leq n/2$. Mit hoher Wahrscheinlichkeit (bezüglich des Zufalls der Eingabe) benötigt

- a) die $(\mu+1)$ -RLS mit $\mu < n^{1/(5\varepsilon(n))}$ (auf CLIQUE_G) eine unendliche erwartete (bezüglich des Zufalls der Suchheuristik) Laufzeit zur Erzeugung einer größten Clique von G und
- b) der $(1+1)$ -EA (auf CLIQUE_G) eine erwartete (bezüglich des Zufalls der Suchheuristik) Laufzeit von $n^{\Theta(1/\varepsilon(n))}$ zur Erzeugung einer größten Clique von G .

Wir bemerken, dass die Werte $\varepsilon(n)$ zu Wahrscheinlichkeiten von $1/2$ bis $1/n^{1/4}$ führen, eine Kante in den Graphen aufzunehmen – also von typischerweise dichten zu recht spärlichen Graphen. Die Werte $k(n)$ stellen lediglich sicher, dass die eingepflanzte Clique typischerweise auch die größte Clique ist.

Beweis (Theorem 5.C.2). Sei $V_0 \subseteq V \setminus P$ mit $|V_0| = |V|/2$ zufällig gleichverteilt gewählt. Betrachten wir den unbehandelten Graphen. Da P mit $|P| = k(n)$ ebenfalls zufällig gleichverteilt aus V gewählt wurde, ist der durch V_0 induzierte Graph ein zufälliger Graph auf $n/2$ Knoten, welcher jede Kante mit Wahrscheinlichkeit $n^{-\varepsilon(n)}$ enthält. Mit hoher Wahrscheinlichkeit treten alle folgenden Ereignisse ein, was am Ende dieses Beweises gezeigt wird. Wir nehmen dies an.

- \mathcal{E}_1 : Die Kardinalität einer größten Clique auf dem durch V_0 induzierten Graphen beträgt höchstens $9/(4\varepsilon(n))$.
- \mathcal{E}_2 : Die Größe einer kleinsten maximalen Clique auf dem durch V_0 induzierten Graphen beträgt mindestens $3/(4\varepsilon(n))$.
- \mathcal{E}_3 : Die Anzahl Cliques der Größe $1/\varepsilon(n)$ auf dem durch V_0 induzierten Graphen beträgt mindestens $n^{1/(5\varepsilon(n))}$.

Wir betrachten den behandelten Graphen, der entsteht, indem ein Gegner alle Kanten, deren beide inzidenten Knoten nicht entweder nur in P oder nur in V_0 liegen, aus dem unbehandelten Graphen entfernt. Da Ereignis \mathcal{E}_1 eintritt, ist die eingepflanzte Clique P mit $9/(4\varepsilon(n)) < 3/\varepsilon(n)$ auch die größte Clique.

- a) Da auch Ereignis \mathcal{E}_3 eintritt, existieren mindestens $n^{1/(5\varepsilon(n))}$ Cliques der Größe $1/\varepsilon(n) \geq 1$ im durch V_0 induzierten Graphen. Mit positiver Wahrscheinlichkeit besteht die erste Population nur aus solchen Cliques und das Ergebnis ergibt sich analog zum Beweis des Theorems 5.A.5.a.
- b) Nach Lemma 5.A.4 mit $\varepsilon = 1/2$ wird jede 1-Clique als Population des (1+1)-EA mit Wahrscheinlichkeit $\Omega(1/n)$ erzeugt, da insbesondere keine Kanten zwischen Knoten aus V_0 und P verlaufen. Irgendeine 1-Clique $\{v\} \subseteq V_0$ wird mit Wahrscheinlichkeit $n/2 \cdot \Omega(1/n) = \Omega(1)$ generiert. Da auch Ereignis \mathcal{E}_2 eintritt, wird analog zum Beweis des Theorems 5.A.5.b mit Wahrscheinlichkeit $\Omega(1)$ eine Clique der Größe mindestens $0.9/\varepsilon(n)$ auf dem durch V_0 induzierten Graphen erzeugt, bevor eine nicht leere Clique auf dem durch $V \setminus V_0$ induzierten Graphen entsteht. Im Anschluss müssen mindestens $3/(4\varepsilon(n))$ spezielle Bits flippen, um eine Teilmenge der größten Clique $P \subseteq V \setminus V_0$ zu erzeugen. Insgesamt führt dies zu einer erwarteten Laufzeit von $\Omega(1) \cdot \Omega(n^{3/(4\varepsilon(n))})$.

Die obere Schranke folgt direkt nach Theorem 5.C.1.

Bollobás (1985) hat bereits nachgewiesen, dass die Ereignisse \mathcal{E}_1 und \mathcal{E}_2 mit hoher Wahrscheinlichkeit eintreten. Betrachten wir das Ereignis \mathcal{E}_3 . Sei X die Zufallsvariable, welche die Anzahl Cliques der Größe $1/\varepsilon(n)$ im durch V_0 induzierten Graphen beschreibt. Bollobás (1985) hat gezeigt, dass

$$\mathbb{E}[X] = \binom{n/2}{1/\varepsilon(n)} n^{-\varepsilon(n) \cdot (1/\varepsilon(n))} \geq n^{1/\varepsilon(n)} \cdot \left(\frac{\varepsilon(n)}{2}\right)^{1/\varepsilon(n)} \cdot n^{-1/(2\varepsilon(n))} \cdot n^{-1/2} \geq n^{1/(4\varepsilon(n))}$$

da $\binom{n/2}{1/\varepsilon(n)} \geq (n \cdot \varepsilon(n)/2)^{1/\varepsilon(n)}$, $(\varepsilon(n)/2)^{1/\varepsilon(n)} \geq (2 \log_2 n)^{-1/\varepsilon(n)} \geq n^{-1/(8\varepsilon(n))}$ und $n^{-1/2} \geq n^{-1/(8\varepsilon(n))}$ ist. Weiter hat Bollobás (1985) gezeigt, dass

$$\text{Var}[X] \leq \mathbb{E}[X]^2 \left(\frac{(n^{\varepsilon(n)} - 1)}{2\varepsilon(n)^4 n^2} + \frac{1}{\mathbb{E}[X]} + \frac{(n^{3\varepsilon(n)} - 1)}{6\varepsilon(n)^7 n^3} + \frac{n \cdot n^{-\varepsilon(n)(1/\varepsilon(n)-1)}}{\varepsilon(n)\mathbb{E}[X]} \right) \leq \mathbb{E}[X]^2 / n^{1/2}$$

da $(n^{\varepsilon(n)} - 1)/(2\varepsilon(n)^4 n^2) \leq n^{1/4}(\log_2 n)^4/(2n^2) \leq 1/(4n^{1/2})$, $1/E[X] \leq n^{-1/(4\varepsilon(n))} \leq 1/(4n^{1/2})$, $(n^{3\varepsilon(n)} - 1)/(6\varepsilon(n)^7 n^3) \leq n^{3/4}(\log_2 n)^7/(6n^3) \leq 1/(4n^{1/2})$ und $n \cdot n^{-\varepsilon(n)(1/\varepsilon(n)-1)}/(\varepsilon(n)E[X]) \leq n^{1/4}(\log_2 n)/n^{1/(4\varepsilon(n))} \leq 1/(4n^{1/2})$ ist.

Um die Wahrscheinlichkeit der Existenz von mindestens $n^{1/(5\varepsilon(n))}$ Cliques der Größe $1/\varepsilon(n)$ im durch V_0 induzierten Graphen zu beschränken, verwenden wir die Methode des zweiten Moments. Genauer zeigt eine Anwendung der Tschebyscheffungleichung, dass

$$\Pr[X \leq E[X]/2] \leq \Pr[|X - E[X]| \geq E[X]/2] \leq \text{Var}[X]/(E[X]/2)^2 \leq 4/n^{1/2}$$

ist. Da $E[X]/2 \geq n^{1/(4\varepsilon(n))}/2 \geq n^{1/(5\varepsilon(n))}$ gilt, ist das gewünschte Resultat bewiesen. \square

Wir untersuchen nun, wie sich das Verhalten von Suchheuristiken verändert, wenn die Einschränkung an einen Gegner aufgehoben wird, lediglich Kanten aus E_{-P} entfernen zu können. Dazu erweitern wir das Eingabemodell $\mathcal{G}_{n,p(n),k(n)}^*$ und betrachten das folgende Eingabemodell der (spärlichen) semizufälligen Graphen $\mathcal{G}_{n,p(n),k(n)}^{**}$. Sei G ein Graph bezüglich $\mathcal{G}_{n,p(n),k(n)}$ gewählt. Einem Gegner ist es dann erlaubt, beliebige Kanten aus $E_{-P} \cup E_P = E$ von G zu entfernen. Ein Gegner kann somit jeden Graphen zwischen $(V, E_P \cup E_{-P})$ und (V, \emptyset) erzeugen. Das Eingabemodell $\mathcal{G}_{n,p(n),k(n)}^{**}$ erlaubt mehr Veränderungen im Graphen als das Eingabemodell $\mathcal{G}_{n,p(n),k(n)}^*$.

Theorem 5.C.3 Sei G ein Graph bezüglich $\mathcal{G}_{n,n^{-\varepsilon(n)},k(n)}^{**}$ gewählt und $\varepsilon(n) \geq 2 \log_2 \log_2 n / \log_2 n$. Der $(\mu+1)$ -EA mit $\mu \geq 2^{4k(n)}$ (auf CLIQUE_G) erzeugt eine größte Clique von G in einer erwarteten (bezüglich des Zufalls der Suchheuristik) Laufzeit von $\mathcal{O}(n^{9/\varepsilon(n)-1} + \mu n \ln n)$ mit Wahrscheinlichkeit $1 - n^{-\Omega(1/\varepsilon(n))}$ (bezüglich des Zufalls der Eingabe).

Wir bemerken, dass die Werte $\varepsilon(n)$ zu Wahrscheinlichkeiten bis $1/\log_2^2 n$ führen, eine Kante in den Graphen aufzunehmen – also zu typischerweise recht dichten Graphen.

Beweis (Theorem 5.C.3). Im Beweis des Theorems 5.C.1 haben wir gezeigt, dass mit Wahrscheinlichkeit $1 - n^{-\Omega(1/\varepsilon(n))}$ das Ereignis \mathcal{E}_1 eintritt, dass keine zwei disjunkten Mengen S, T mit $|S| = |T| = 9/(4\varepsilon(n))$, $S \subseteq V \setminus P$, $T \subseteq V$ und $\{\{s, t\} \mid s \in S, t \in T\} \subseteq E$ bezüglich des unbehandelten Graphen existieren und damit auch nicht im behandelten Graphen. Falls $9/(4\varepsilon(n)) \leq k(n) \leq 9 \log_2 n / (4\varepsilon(n))$ ist, betrachten wir zusätzlich das Ereignis \mathcal{E}_2 , dass keine $2k(n)$ Knoten aus $V \setminus P$ existieren, die adjazent zu mindestens $9/(4\varepsilon(n))$ Knoten aus P sind bezüglich des unbehandelten Graphen und damit auch nicht im behandelten Graphen. Sei $\ell(n) := k(n) \cdot 4\varepsilon(n)/9$. Analog zum Beweis des Theorems 5.C.1 gilt

$$\begin{aligned} \Pr[\mathcal{E}_2] &\geq 1 - \binom{n - k(n)}{2k(n)} \cdot \left(\frac{k(n)}{4\varepsilon(n)}\right)^{2k(n)} \cdot n^{-\varepsilon(n) \cdot 2k(n) \cdot \frac{9}{4\varepsilon(n)}} \\ &\geq 1 - n^{2k(n)} \cdot n^{(19/16) \cdot 2k(n)} \cdot n^{-\frac{9k(n)}{2}} = 1 - n^{k(n)/8} = 1 - n^{-\Omega(1/\varepsilon(n))} \end{aligned}$$

da $\binom{k(n)}{9/(4\varepsilon(n))} \leq \left(\frac{e \cdot 9/(4\varepsilon(n)) \ell(n)}{9/(4\varepsilon(n))}\right)^{9/(4\varepsilon(n))} \leq (e \log_2 n)^{9/(4 \cdot 2 \log_2 \log_2 n / \log_2 n)} \leq n^{19/16}$ gilt mit $\ell(n) \leq \log_2 n$. Wir nehmen dies an und unterscheiden, ob $k(n) < 9/(4\varepsilon(n))$ (Fall 1), $9/(4\varepsilon(n)) \leq k(n) \leq 9 \log_2 n / (4\varepsilon(n))$ (Fall 2) oder $k(n) > 9 \log_2 n / (4\varepsilon(n))$ (Fall 3) gilt.

1. Da Ereignis \mathcal{E}_1 eintritt, existieren keine Cliques, die größer als $9/(2\varepsilon(n)) - 1$ sind. Nach Theorem 5.A.2 erzeugt der $(\mu+1)$ -EA dann eine größte Clique von G in einer erwarteten Laufzeit von $\mathcal{O}(n^{9/\varepsilon(n)-4} + \mu n \ln n)$.
2. Sei P' die Menge der Knoten aus $V \setminus P$, die adjazent zu mindestens $9/(4\varepsilon(n))$ Knoten aus P sind bezüglich des behandelten Graphen. Da Ereignis \mathcal{E}_2 eintritt, gilt $|P'| < 2k(n)$. Da auch Ereignis \mathcal{E}_1 eintritt, gilt für jede Clique C mit $C \cap (V \setminus (P \cup P')) \neq \emptyset$ sicher $|C| < 9/(2\varepsilon(n))$. Für jede Clique C mit $|C| \geq 9/(2\varepsilon(n))$ gilt somit $C \subseteq P \cup P'$. Da $|P \cup P'| = |P| + |P'| \leq k(n) + (2k(n) - 1) = 3k(n) - 1$ ist, gilt für den behandelten Graphen $|\mathcal{C}^{\geq 9/(2\varepsilon(n))}(G)| \leq \sum_{i=9/(2\varepsilon(n))}^{3k(n)-1} \binom{3k(n)-1}{i} \leq 2^{3k(n)-1}$. Nach Lemma 5.A.3 mit $s(n) = 9/(2\varepsilon(n))$ erzeugt der $(\mu+1)$ -EA dann eine größte Clique von G in einer erwarteten Laufzeit von $\mathcal{O}(n^{9/\varepsilon(n)-1} + \mu n \ln n)$, falls $\mu \geq 2^{3k(n)}$ ist.
Die Kardinalität einer größten Clique ist höchstens $3k(n) - 1$.
3. Da Ereignis \mathcal{E}_1 eintritt, existieren im durch $V \setminus P$ induzierten Graphen bezüglich des behandelten Graphen keine Cliques größer als $9/(2\varepsilon(n))$. Somit gibt es insgesamt höchstens $\sum_{i=1}^{9/(2\varepsilon(n))} \binom{n-k(n)}{i} \leq (e-1) \cdot n^{9/(2\varepsilon(n))} \leq 2^{2k(n)+1}$ maximale Cliques auf diesem Graphen. Weiter ist jede Clique des behandelten Graphen G sicher eine Teilmenge von $C \cup P$ für eine maximale Clique C . Die Anzahl Cliques auf $C \cup P$ beträgt höchstens $2^{|C \cup P|} = 2^{|C|+|P|} \leq 2^{2k(n)-1}$, da $|C| \leq 9/(2\varepsilon(n)) < k(n)$ ist, und es resultiert $|\mathcal{C}^{\geq 0}(G)| \leq 2^{2k(n)+1} \cdot 2^{2k(n)-1} \leq 2^{4k(n)}$. Nach Lemma 5.A.1.b erzeugt der $(\mu+1)$ -EA mit $\mu \geq 2^{4k(n)}$ dann eine größte Clique von G in einer erwarteten Laufzeit von $\mathcal{O}(\mu n \ln n)$.
Die Kardinalität einer größten Clique ist höchstens $2k(n) - 1$.

Das gewünschte Resultat ist somit gezeigt. \square

Wir zeigen nun die Schärfe des Theorems 5.C.3.

Theorem 5.C.4 Sei G ein Graph bezüglich $\mathcal{G}_{n, n-\varepsilon(n), k(n)}^{**}$ gewählt sowie $1/\log_2 n \leq \varepsilon(n) \leq 1/4$ und $3/\varepsilon(n) \leq k(n) \leq n/2$. Mit hoher Wahrscheinlichkeit (bezüglich des Zufalls der Eingabe) benötigt

- a) die $(\mu+1)$ -RLS mit $\mu < n^{1/(5\varepsilon(n))} + 2^{k(n)/5} - 1$ (auf CLIQUE_G) eine unendliche erwartete (bezüglich des Zufalls der Suchheuristik) Laufzeit zur Erzeugung einer größten Clique von G und
- b) der $(1+1)$ -EA (auf CLIQUE_G) eine erwartete (bezüglich des Zufalls der Suchheuristik) Laufzeit von $n^{\Theta(k(n))}$ zur Erzeugung einer größten Clique von G .

Wir bemerken, dass wir lediglich die gleichen Einschränkungen für die Werte $\varepsilon(n)$ und $k(n)$ vorgenommen haben wie im Theorem 5.C.2.

Beweis (Theorem 5.C.4).

- a) Analog zum Beweis des Theorems 5.C.2 sei $V_0 \subseteq V \setminus P$ mit $|V_0| = |V|/2$ zufällig gleichverteilt gewählt und sei $P_1 \subseteq P$ mit $|P_1| = 4k(n)/5$ und $P_0 := P \setminus P_1$. Wir betrachten den behandelten Graphen, der entsteht, indem ein Gegner alle Kanten, deren beide inzidenten Knoten nicht entweder nur in P_1 , nur in P_0 oder nur in V_0 liegen, aus dem unbehandelten Graphen entfernt. Mit hoher Wahrscheinlichkeit treten daher auch die

Ereignisse \mathcal{E}_1 und \mathcal{E}_3 des Beweises des Theorems 5.C.2 ein. Wir nehmen dies an. Da Ereignis \mathcal{E}_1 eintritt, ist die Clique P_1 mit $9/(4\varepsilon(n)) < 4k(n)/5$ auch die größte Clique. Da auch Ereignis \mathcal{E}_3 eintritt, existieren mindestens $n^{1/(5\varepsilon(n))}$ Cliques der Größe $1/\varepsilon(n)$ im durch V_0 induzierten Graphen und $2^{k(n)/5} - 1$ nicht leere Cliques im durch P_0 induzierten Graphen. Mit positiver Wahrscheinlichkeit besteht die erste Population nur aus solchen Cliques und das Ergebnis ergibt sich analog zum Beweis des Theorems 5.A.5.a.

- b) Sei $P_1 \subseteq P$ mit $|P_1| = \lfloor k(n)/2 \rfloor$ und $P_0 \subseteq P \setminus P_1$ mit $|P_0| = \lfloor k(n)/2 \rfloor - 1$. Wir betrachten den behandelten Graphen, der entsteht, indem ein Gegner alle Kanten, deren beide inzidenten Knoten nicht entweder nur in P_1 oder nur in P_0 liegen, aus dem unbehandelten Graphen entfernt. Dieser Graph ist isomorph, also mit einer Umbenennung der Knoten äquivalent zu $W_{n, k(n)/\sqrt{2}}$, welcher im Beweis des Theorems 5.B.3 definiert ist. Das Ergebnis folgt damit direkt.

Die obere Schranke folgt direkt nach Theorem 5.A.2, da bereits im Beweis des Theorems 5.C.3 gezeigt wurde, dass die Kardinalität einer größten Clique im behandelten Graphen mit hoher Wahrscheinlichkeit höchstens $3k(n) - 1$ beträgt.

Das gewünschte Resultat ist somit gezeigt. \square

5.C.2 Kanten einfügen

Wir erweitern unsere Untersuchungen für Suchheuristiken auf (spärlichen) semizufälligen Graphen, bei denen einem Gegner sogar erlaubt wird, das Problem „(moderat) schwerer“ zu gestalten. Wir betrachten das folgende Eingabemodell der (spärlichen) semizufälligen Graphen $\mathcal{G}_{n,p(n),m(n)}^{***}$. Sei G ein Graph bezüglich $\mathcal{G}_{n,p(n),0}$ gewählt. Der Graph G ist damit ein (spärlicher) zufälliger Graph (ohne eingepflanzte Clique). Einem Gegner ist es dann erlaubt, beliebige, jedoch höchstens $m(n)$ viele Kanten in G einzufügen. Für $p(n) = 0$ erhalten wird das Eingabemodell der spärlichen Graphen aus Abschnitt 5.B. Es sei bemerkt, dass die Resultate dieses Unterabschnitts auch dann noch gelten, wenn es einem Gegner zusätzlich erlaubt ist, beliebige Kanten aus G zu entfernen.

Theorem 5.C.5 Sei G ein Graph bezüglich $\mathcal{G}_{n, n^{-\varepsilon(n)}, \binom{k(n)}{2}}^{***}$ gewählt sowie $1/\log_2^{1/16} n \leq \varepsilon(n) \leq 1$ und $\log_2^{1/4} n \leq k(n) \leq 2^{(\log_2^{15/16} n)/32}$. Der $(\mu+1)$ -EA mit $\mu \geq 2^{11k(n)/10}$ (auf CLIQUE_G) erzeugt eine größte Clique von G in einer erwarteten (bezüglich des Zufalls der Suchheuristik) Laufzeit von $\mathcal{O}(n^{9/\varepsilon(n)-1} + \mu n \ln n)$ mit Wahrscheinlichkeit $1 - n^{-\Omega(1/\varepsilon(n))}$ (bezüglich des Zufalls der Eingabe).

Wir bemerken, dass die Werte $\varepsilon(n)$ von typischerweise recht dichten zu spärlichen Graphen führen. Die Werte $k(n)$ erlauben sub-lineare bis super-polynomielle Populationsgrößen.

Beweis (Theorem 5.C.5). Betrachten wir den unbehandelten Graphen. Mit Wahrscheinlichkeit $1 - n^{-\Omega(1/\varepsilon(n))}$ treten alle folgenden Ereignisse ein, was am Ende dieses Beweises gezeigt wird. Wir nehmen dies an.

\mathcal{E}_1 : Es existieren keine zwei disjunkten Mengen $S, T \subseteq V$ mit $|S| = |T| = 9/(4\varepsilon(n))$ und $\{\{s, t\} | s \in S, t \in T\} \subseteq E$.

\mathcal{E}_2 : Es existieren keine zwei disjunkten Mengen $S, T \subseteq V$ mit $|S| = |T| = 2\binom{k(n)}{2}$ und für jedes $t \in T$ gilt $|\{s, t\} | s \in S\}| \geq 9/(4\varepsilon(n))$.

\mathcal{E}_3 : Es existiert keine Menge $S \subseteq V$ mit $|S| = 4\binom{k(n)}{2}$ und in dem durch S induzierten Graphen liegen mindestens $\binom{k(n)}{2}^{19/16}$ Kanten.

\mathcal{E}_4 : Es existiert keine Menge $S \subseteq V$ mit $|S| = \binom{k(n)}{2}^{13/16}$ und in dem durch S induzierten Graphen liegen mindestens $\binom{k(n)}{2}/5$ Kanten.

Betrachten wir den aus dem unbehandelten Graphen entstandenen behandelten Graphen. Sei $V_0^\ddagger \subseteq V$ die Menge aller zu einer vom Gegner eingefügten Kante inzidenten Knoten. Dann ist $|V_0^\ddagger| \leq 2\binom{k(n)}{2}$. Weiter sei $V_0^\dagger \subseteq V \setminus V_0^\ddagger$ die Menge aller zu mindestens $9/(4\varepsilon(n))$ Knoten aus V_0^\ddagger adjazenten Knoten. Da Ereignis \mathcal{E}_2 eintritt, ist $|V_0^\dagger| \leq 2\binom{k(n)}{2}$. Des Weiteren sei $V_0 := V_0^\ddagger \cup V_0^\dagger$ mit $|V_0| \leq 4\binom{k(n)}{2}$. Wir vermuten und zeigen im Folgenden, dass auf dem durch V_0 induzierten Graphen „viel“ und auf dem Rest des Graphen „wenig“ passieren kann.

Betrachten wir eine Clique C mit $C \cap (V \setminus V_0) \neq \emptyset$. Da sowohl Ereignis \mathcal{E}_2 als auch Ereignis \mathcal{E}_1 eintritt, ist $|C| < 9/(2\varepsilon(n))$.

Betrachten wir eine Clique C mit $C \subseteq V_0$. Weil V_0 nur wenige Knoten aus V umfasst, liegen im durch V_0 induzierten Graphen auch nur wenige Kanten. Da Ereignis \mathcal{E}_3 eintritt, existieren genauer weniger als $\binom{k(n)}{2}^{19/16} + \binom{k(n)}{2}$ solche Kanten im behandelten Graphen. Somit gibt es ebenfalls lediglich wenige Knoten $V_0^- \subseteq V_0$, die zu vielen, genauer zu mindestens $5\binom{k(n)}{2}^{3/8}$ vielen Knoten aus V_0 adjazent sind. Es gilt $|V_0^-| \leq \binom{k(n)}{2}^{13/16}$, da sonst widersprüchlich wenigstens

$$\binom{k(n)}{2}^{13/16} \cdot 5 \binom{k(n)}{2}^{3/8} / 2 > \binom{k(n)}{2}^{19/16} + \binom{k(n)}{2}$$

Kanten im durch V_0 induzierten Graphen liegen würden.

Betrachten wir eine Clique C mit $C \subseteq V_0$ und $C \cap (V_0 \setminus V_0^-) \neq \emptyset$. Da zumindest ein Knoten von C höchstens Grad $5\binom{k(n)}{2}^{3/8} - 1$ im durch V_0 induzierten Graphen besitzt, ist $|C| \leq 5\binom{k(n)}{2}^{3/8}$. Somit gibt es höchstens

$$\sum_{i=0}^{5\binom{k(n)}{2}^{3/8}} \binom{4\binom{k(n)}{2}}{i} \leq e \binom{4\binom{k(n)}{2}}{5\binom{k(n)}{2}^{3/8}} \leq e \left(4\binom{k(n)}{2}\right)^{5\binom{k(n)}{2}^{3/8}} \leq 2^{k(n)7/8}$$

solche Cliques.

Betrachten wir eine Clique C mit $C \subseteq V_0^-$. Weil V_0^- noch weniger Knoten aus V umfasst als V_0 , liegen auch im durch V_0^- induzierten Graphen noch weniger Kanten. Da Ereignis \mathcal{E}_4 eintritt, existieren genauer weniger als $\binom{k(n)}{2}/5 + \binom{k(n)}{2} = 6\binom{k(n)}{2}/5$ solche Kanten im behandelten Graphen. Nach Lemma 5.B.1 gibt es höchstens

$$2\sqrt{6/5 \cdot k(n)} + \binom{k(n)}{2}^{13/16}$$

Cliquen im durch V_0^- induzierten Graphen.

Insgesamt gilt also $|\mathcal{C}^{\geq 9/(2\varepsilon(n))}(G)| \leq 2^{k(n)^{7/8}} + 2\sqrt{6/5 \cdot k(n)} + \binom{k(n)}{2}^{13/16} \leq 2^{11k(n)/10-1}$ und das gewünschte Resultat folgt direkt nach Lemma 5.A.3.

\mathcal{E}_1 : Analog zum Beweis des Theorems 5.C.3 beträgt die Wahrscheinlichkeit, dass \mathcal{E}_1 nicht eintritt, höchstens

$$\binom{n}{9/(4\varepsilon(n))} \binom{n}{9/(4\varepsilon(n))} n^{-\varepsilon(n) \cdot \left(\frac{9}{4\varepsilon(n)}\right)^2} \leq n^{2 \cdot \frac{9}{4\varepsilon(n)} - \varepsilon(n) \cdot \frac{81}{16\varepsilon(n)^2}} = n^{-\frac{9}{16\varepsilon(n)}} = n^{-\Omega(1/\varepsilon(n))}.$$

\mathcal{E}_2 : Falls zwei solche Mengen S, T existieren, dann gibt es für jedes $t \in T$ eine Menge $S_t \subseteq S$ mit $|S_t| = 9/(4\varepsilon(n))$ und t ist adjazent zu jedem Knoten aus S_t . Abermals analog zum Beweis des Theorems 5.C.3 beträgt die Wahrscheinlichkeit, dass \mathcal{E}_2 nicht eintritt, höchstens

$$\begin{aligned} & \binom{n}{2 \binom{k(n)}{2}} \binom{n - 2 \binom{k(n)}{2}}{2 \binom{k(n)}{2}} \binom{2 \binom{k(n)}{2}}{9/(4\varepsilon(n))}^{2 \binom{k(n)}{2}} n^{-\varepsilon(n) \cdot 2 \binom{k(n)}{2} \cdot 9/(4\varepsilon(n))} \\ & \leq n^{2 \cdot 2 \binom{k(n)}{2}} n^{(1/6) \cdot 2 \binom{k(n)}{2}} n^{-9 \binom{k(n)}{2} / 2} = n^{-\binom{k(n)}{2} / 6} = n^{-\Omega(1/\varepsilon(n))}, \end{aligned}$$

da $\binom{2 \binom{k(n)}{2}}{9/(4\varepsilon(n))} \leq \left(\frac{e \cdot 2 \binom{k(n)}{2}}{9/(4\varepsilon(n))}\right)^{9/(4\varepsilon(n))} \leq 2^{\log_2(4 \binom{k(n)}{2} \cdot 9/(4\varepsilon(n)))} \leq 2^{(1 + (\log_2^{15/16} n)/16) \cdot 9(\log_2^{1/16} n)/4} \leq n^{1/6}$ ist mit $\frac{e-2}{9/(4\varepsilon(n))} \leq 4$.

\mathcal{E}_3 : Betrachten wir eine beliebige, aber feste Auswahl von $4 \binom{k(n)}{2}$ Knoten. Die Wahrscheinlichkeit, dass im dadurch induzierten Graphen bezüglich des unbehandelten Graphen mindestens $\binom{k(n)}{2}^{19/16}$ Kanten existieren, beträgt höchstens

$$\begin{aligned} & \binom{4 \binom{k(n)}{2}}{\binom{k(n)}{2}^{19/16}} \cdot n^{-\varepsilon(n) \binom{k(n)}{2}^{19/16}} \leq \left(16 \binom{k(n)}{2} / 2\right)^{\binom{k(n)}{2}^{16/19}} \cdot n^{-\varepsilon(n) \binom{k(n)}{2}^{19/16}} \\ & \leq n^{\varepsilon(n) \binom{k(n)}{2}^{19/16} / 2} \cdot n^{-\varepsilon(n) \binom{k(n)}{2}^{19/16}} = n^{-\varepsilon(n) \binom{k(n)}{2}^{19/16} / 2}, \end{aligned}$$

da $16 \binom{k(n)}{2} / 2 \leq 2^{2+2 \cdot 2 \log_2 k(n)} \leq 2^{2+(\log_2^{15/16} n)/8} \leq 2^{(\log_2^{15/16} n)/2} = n^{1/(2 \log_2^{1/16} n)} \leq n^{\varepsilon(n)/2}$ ist. Abermals analog zum Beweis des Theorems 5.C.3 beträgt die Wahrscheinlichkeit, dass \mathcal{E}_3 nicht eintritt, höchstens

$$\binom{n}{4 \binom{k(n)}{2}} n^{-\varepsilon(n) \binom{k(n)}{2}^{19/16} / 2} \leq n^{\binom{k(n)}{2} (4 - \varepsilon(n) \binom{k(n)}{2}^{3/16} / 2)} \leq n^{-\binom{k(n)}{2}} = n^{-\Omega(1/\varepsilon(n))},$$

da mit $\binom{k(n)}{2} \geq k(n)^2/4 \geq (\log_2^{1/2} n)/4$ gilt

$$\begin{aligned} 4 - \varepsilon(n) \binom{k(n)}{2}^{3/16} / 2 &\leq 4 - (1/\log_2^{1/16} n) \cdot ((\log_2^{1/2} n)/4)^{3/16} / 2 \\ &= 4 - (\log_2^{1/32} n) / (2 \cdot 4^{3/16}) \leq -1. \end{aligned}$$

\mathcal{E}_4 : Betrachten wir eine beliebige, aber feste Auswahl von $\binom{k(n)}{2}^{13/16}$ Knoten. Die Wahrscheinlichkeit, dass im dadurch induzierten Graphen bezüglich des unbehandelten Graphen mindestens $\binom{k(n)}{2}/5$ Kanten existieren, beträgt höchstens

$$\begin{aligned} \binom{\binom{k(n)}{2}^{13/16}}{\binom{k(n)}{2}/5} \cdot n^{-\varepsilon(n)\binom{k(n)}{2}/5} &\leq \left(\binom{k(n)}{2}^{2 \cdot 13/16} / 2 \right)^{\binom{k(n)}{2}/5} \cdot n^{-\varepsilon(n)\binom{k(n)}{2}/5} \\ &\leq n^{\varepsilon(n)\binom{k(n)}{2}/10} \cdot n^{-\varepsilon(n)\binom{k(n)}{2}/5} \leq n^{-\varepsilon(n)\binom{k(n)}{2}/10}, \end{aligned}$$

da $\binom{k(n)}{2}^{2 \cdot 13/16} / 2 \leq 2^{(2 \cdot 13/16) \cdot 2 \log_2 k} \leq 2^{13(\log_2^{15/16} n)/128} \leq 2^{(\log_2^{15/16} n)/2} = n^{1/(2 \log_2^{1/16} n)} \leq n^{\varepsilon(n)/2}$ ist. Abermals analog zum Beweis des Theorems 5.C.3 beträgt die Wahrscheinlichkeit, dass \mathcal{E}_4 nicht eintritt, höchstens

$$\binom{n}{\binom{k(n)}{2}^{13/16}} n^{-\varepsilon(n)\binom{k(n)}{2}/10} \leq n^{\binom{k(n)}{2}^{13/16} (1 - \varepsilon(n)\binom{k(n)}{2}^{3/16})/10} \leq n^{-\binom{k(n)}{2}^{13/16}} = n^{-\Omega(1/\varepsilon(n))},$$

da mit $\binom{k(n)}{2} \geq k(n)^2/4 \geq (\log_2^{1/2} n)/4$ gilt

$$\begin{aligned} 1 - \varepsilon(n) \binom{k(n)}{2}^{3/16} / 10 &\leq 1 - (1/\log_2^{1/16} n) \cdot ((\log_2^{1/2} n)/4)^{3/16} / 10 \\ &\leq 1 - (\log_2^{1/32} n) / (10 \cdot 4^{3/16}) \leq -1. \end{aligned}$$

Das gewünschte Resultat ist somit gezeigt. \square

Wir zeigen nun die Schärfe des Theorems 5.C.5.

Theorem 5.C.6 Sei G ein Graph bezüglich $\mathcal{G}_{n, n^{-\varepsilon(n)}, \binom{k(n)}{2}}^{***}$ gewählt sowie $\varepsilon(n) \geq 27/! \log_2^{1/4} n$ und $\log_2^{1/4} n \leq k(n) \leq n^{1/4}$. Mit Wahrscheinlichkeit $1 - n^{-\Omega(k(n))}$ (bezüglich des Zufalls der Eingabe) benötigt

- die $(\mu+1)$ -RLS mit $\mu \leq 2^{k(n)/4}$ (auf CLIQUE_G) eine unendliche erwartete (bezüglich des Zufalls der Suchheuristik) Laufzeit zur Erzeugung einer größten Clique von G und
- der $(1+1)$ -EA (auf CLIQUE_G) eine erwartete (bezüglich des Zufalls der Suchheuristik) Laufzeit von $n^{\Theta(k(n))}$ zur Erzeugung einer größten Clique von G .

Wir bemerken, dass insbesondere die gleichen Werte $\varepsilon(n)$ und $k(n)$ betrachtet werden wie im Theorem 5.C.5.

Beweis (Theorem 5.C.6). Betrachten wir den unbehandelten Graphen. Mit Wahrscheinlichkeit $1 - n^{-\Omega(k(n))}$ treten alle folgenden Ereignisse ein, was am Ende dieses Beweises gezeigt wird. Wir nehmen dies an.

\mathcal{E}_1 : Es existieren keine zwei disjunkte Mengen $S, T \subseteq V$ mit $|S| = |T| = k(n)/13$ sowie $\{\{s, t\} \mid s \in S, t \in T\} \subseteq E$.

\mathcal{E}_2 : Es existiert keine Menge $S \subseteq V$ mit $|S| = 2k(n)$ sowie $|\{\{s_1, s_2\} \mid s_1 \neq s_2 \text{ und } s_1, s_2 \in S\} \setminus E| \leq \binom{k(n)}{2}$.

Sei $V_0 := \{v_1, \dots, v_{8k(n)/13}\}$ und $V_1 := \{v_{n-9k(n)/13+1}, \dots, v_n\}$. Wir betrachten den behandelten Graphen, der entsteht, indem ein Gegner alle (fehlenden) Kanten aus der Menge $\{\{v_i, v_j\} \mid 1 \leq i < j \leq 8k(n)/13 \text{ oder } n - 9k(n)/13 + 1 \leq i < j \leq n\}$ in den unbehandelten Graphen eingefügt. Dies sind höchstens $\binom{8k(n)/13}{2} + \binom{9k(n)/13}{2} \leq \binom{k(n)}{2}$ viele Kanten.

Betrachten wir beliebige Cliques C mit $|C| \geq 2k(n)/13$. Da Ereignis \mathcal{E}_1 eintritt, ist entweder $|C \cap (V \setminus V_0)| < k(n)/13$ oder $|C \cap (V \setminus V_1)| < k(n)/13$. Für jede größte Clique gilt sogar $|C| \geq |V_1| > |V_0| + (k(n)/13 - 1)$ und damit $|C \cap V_0| < k(n)/13$.

- a) Mit positiver Wahrscheinlichkeit besteht die erste Population nur aus Cliques der Größe $2k(n)/13$ aus dem durch V_0 induzierten Graphen. Wenigstens $\binom{8k(n)/13}{2k(n)/13} \geq 2^{k(n)/4}$ solche Cliques gibt es. Danach existieren ebenfalls nur Cliques C mit $|C \cap V_0| \geq k(n)/13 + 1$ in der Population auf, da sonst widersprüchlich eine Clique C mit $|C \cap V_0| = k(n)/13$ und $|C \cap (V \setminus V_0)| \geq k(n)/13$ in der Population aufgetreten wäre. Das Ergebnis ergibt sich analog zum Beweis des Theorems 5.A.5.a.
- b) Betrachten wir den ersten Zeitpunkt, zu dem ein Individuum x mit $|x_1 \cdots x_{n-9k(n)/13}| < k(n)/13$ (Fall 1) oder $|x_{9k(n)/13+1} \cdots x_n| < k(n)/13$ (Fall 2) als Population auftritt. Wir bemerken, dass keines der Individuen, die zuvor betrachtet wurden, eine Clique darstellt. Bereits im Beweis des Lemmas 5.A.4 haben wir festgestellt, dass dann zu jedem Zeitpunkt jedes Element mit gleicher Anzahl Einsen mit gleicher Wahrscheinlichkeit erzeugt wird. Also tritt Fall 2 mindestens mit Wahrscheinlichkeit $1/2$ ein. Wir nehmen dies an. Im nächsten Schritt erzeugt eine spezielle höchstens $10k(n)/13 - 1$ Bits flippende Mutation die Clique V_0 . Diese Mutation flippt höchstens $9k(n)/13$ Bits in $x_1 \cdots x_{9k(n)/13}$ sowie höchstens $k(n)/13 - 1$ weitere Bits und hat damit mindestens Wahrscheinlichkeit $1/(en^{10k(n)/13+1})$. Wir nehmen dies an. In jedem Schritt wird dann höchstens mit Wahrscheinlichkeit

$$\binom{8k(n)/13}{6k(n)/13+2} \binom{9k(n)/13}{6k(n)/13+2} n^{-2(6k(n)/13+2)} \leq 2^{k(n)} n^{-12k(n)/13} \leq n^{-11k(n)/13}$$

eine Clique C mit $|C \cap V_0| < 7k(n)/13 + 1$ – und daher nach obigen Betrachtungen sogar mit $|C \cap V_0| < k(n)/13$ – und $|C \cap V_1| \geq 7k(n)/13 + 1$ erzeugt. Hierzu müssen mindestens $6k(n)/13 + 2$ (irgendwelche) Bits in $x_1 \cdots x_{8k(n)/13}$ und mindestens $6k(n)/13 + 2$ (irgendwelche) Bits in $x_{9k(n)/13+1} \cdots x_n$ kippen. Insgesamt führt dies zu einer erwarteten Laufzeit von mindestens $(1/2) \cdot (1/(en^{10k(n)/13+1})) \cdot n^{11k(n)/13} = n^{\Omega(k(n))}$.

Die obere Schranke folgt direkt nach Theorem 5.A.2. Da Ereignis \mathcal{E}_2 eintritt, kann ein Gegner höchstens Cliques der Größe $2k(n) - 1$ erzeugen.

\mathcal{E}_1 : Analog zum Beweis des Theorems 5.C.3 beträgt die Wahrscheinlichkeit, dass \mathcal{E}_1 nicht eintritt, höchstens

$$\binom{n}{k(n)/13} \binom{n}{k(n)/13} n^{-\varepsilon(n) \cdot \left(\frac{k(n)}{13}\right)^2} \leq n^{2 \cdot \frac{k(n)}{13} - \frac{27}{\log_2^{1/4} n} \cdot \frac{(\log_2^{1/4} n) \cdot k(n)}{169}} = n^{-k(n)/169}.$$

\mathcal{E}_2 : Betrachten wir eine beliebige, aber feste Auswahl von $2k(n)$ Knoten $S \subseteq V$. Sei X_i , $1 \leq i \leq \binom{2k(n)}{2}$, die Indikatorvariable, welche die Existenz der i -ten Kante im dadurch induzierten Graphen bezüglich des unbehandelten Graphen beschreibt. Sei $X := X_1 + \dots + X_{\binom{2k(n)}{2}}$, dann gilt $E[X] = \binom{2k(n)}{2} n^{-\varepsilon(n)}$ und eine Anwendung der Chernoffgleichung mit $\delta = \frac{\binom{2k(n)}{2} - \binom{k(n)}{2}}{E[X]} > 1$ zeigt, dass

$$\begin{aligned} \Pr[X \geq \binom{2k(n)}{2} - \binom{k(n)}{2}] &= \Pr[X \geq (1 + (\delta - 1)) \cdot E[X]] \leq e^{-E[X]} (e/\delta)^{\delta E[X]} \\ &\leq (2n^{-\varepsilon(n)})^{\binom{2k(n)}{2} - \binom{k(n)}{2}} \leq n^{(-\varepsilon(n)/3)k(n)^2} \leq n^{-9k(n)} \end{aligned}$$

gilt, da $e^{-E[X]} \leq 1$, $\frac{e^{\binom{2k(n)}{2}}}{\binom{2k(n)}{2} - \binom{k(n)}{2}} \leq 2$ und $2n^{-\varepsilon(n)} \leq n^{(-\varepsilon(n)/3)}$ ist. Abermals analog zum Beweis des Theorems 5.C.3 beträgt die Wahrscheinlichkeit, dass \mathcal{E}_2 nicht eintritt, höchstens

$$\binom{n}{2k(n)} n^{-9k(n)} \leq n^{2k(n)} n^{-9k(n)} = n^{-\Omega(k(n))}.$$

Das gewünschte Resultat ist somit gezeigt. \square

Betrachten wir einen Graphen bezüglich $\mathcal{G}_{n, n^{-\varepsilon(n)}, m(n)}^{***}$ gewählt. Wenn die Population groß genug ist, erzeugt also der $(\mu+1)$ -EA mit hoher Wahrscheinlichkeit (bezüglich des Zufalls der Eingabe) effizient eine größte Clique, falls $\varepsilon(n) = \Omega(1)$ und $m(n) = \mathcal{O}(\ln^2 n)$ ist. Dagegen generiert der $(1+1)$ -EA mit hoher Wahrscheinlichkeit (bezüglich des Zufalls der Eingabe) lediglich ineffizient eine größte Clique, insbesondere falls $m(n) = \omega(1)$ ist. Dies zeigt in einer sehr allgemeinen Form die mögliche wesentliche Steigerung der Effizienz randomisierter Suchheuristiken durch den Einsatz von großen Populationen (in der kombinatorischen Optimierung).

5.D Optimierung für planare und zufällige planare Graphen

Bereits am Anfang dieses Kapitels haben wir die Betrachtung von planaren Graphen motiviert. Diese Graphen können in die Ebene gezeichnet werden, ohne dass zwei Kanten sich schneiden, und stellen somit eine Teilmenge aller Graphen dar. Wir sind dabei an optimalen Lösungen für das Cliquenproblem interessiert und betrachten neben allen planaren Graphen auch das Eingabemodell der zufälligen planaren Graphen $\mathcal{G}_n^{\text{planar}}$, was einen planaren Graphen auf n Knoten zufällig gleichverteilt wählt. Wir haben festgestellt, dass ein optimaler

Algorithmus eine größte Clique eines beliebigen planaren Graphen (deterministisch) in einer Laufzeit von $\mathcal{O}(n)$ berechnet (Eppstein, 1999). Für zufällige planare Graphen dagegen bestimmt ein optimaler Algorithmus (deterministisch) in einer erwarteten (bezüglich des Zufalls der Eingabe) Laufzeit von $\mathcal{O}(1)$ eine größte Clique (McDiarmid et al., 2005). Wir halten einige Eigenschaften beliebiger und zufälliger planarer Graphen fest. Eine Menge $V' \subseteq V$ nennen wir *lose eingebunden* in den Graphen $G = (V, E)$, falls $|\{\{v, w\} \mid v \in V', w \in V \setminus V'\}| \leq 1$ ist.

Lemma 5.D.1

a) Sei n groß genug und G ein beliebiger planarer Graph auf n Knoten. Es gilt:

$k =$	0	1	2	3	4	5 ...
Anzahl k -Cliquen \leq	1	n	$3n - 6$	$6n - 12$	$6n - 12$	0

b) Sei G ein Graph bezüglich $\mathcal{G}_n^{\text{planar}}$ gewählt. Mit Wahrscheinlichkeit $1 - 2^{-\Omega(n)}$ gilt:

$k =$	3	4
Anzahl lose eingebundener k -Cliquen \geq	$\Omega(n)$	$\Omega(n)$

Beweis.

- a) Der Graph G umfasst genau eine 0-Clique und alle n viele 1-Cliquen. Die Anzahl 2-Clique beträgt höchstens $3n - 6$ nach Eulers Polyederformel (Legendre, 1794). Ebenso existiert in jedem nicht leeren planaren Graphen ein Knoten $v \in V$ mit Grad $1 \leq \deg(v) \leq 5$. Das Entfernen der zu v inzidenten Knoten aus G eliminiert höchstens $\binom{\deg(v)}{2} \leq 2 \deg(v)$ viele 3-Cliquen und höchstens $\binom{\deg(v)}{3} \leq 2 \deg(v)$ viele 4-Cliquen, wobei die Ungleichungen für $\deg(v) \leq 5$ gelten. Die Summe der Grade aller Knoten sinkt ebenfalls um $2 \deg(v)$. Insgesamt werden so höchstens $\sum_{v \in V} \deg(v) \leq 2 \cdot (3n - 6) = 6n - 12$ viele 3-Cliquen und 4-Cliquen eliminiert. Es existieren keine Cliquen der Größe mindestens fünf in einem planaren Graphen nach Kuratowskis Theorem (Kuratowski, 1930).
- b) Das Ergebnis folgt direkt nach McDiarmid et al. (2005), da die 3-Clique und die 4-Clique zusammenhängende planare Graphen sind.

Das gewünschte Resultat ist somit gezeigt. \square

Für jeden Graphen G sei $V_i \subseteq V$, $i \geq 3$, die Menge der Knoten aus V , die in – notwendigerweise genau einer – lose eingebundenen i -Clique liegen. Des Weiteren sei $V'_i \subseteq V_i$, $i \geq 3$, die Menge der Knoten aus V_i , die genau Grad $i - 1$ (und nicht i) haben. Nach Lemma 5.D.1.b gilt für planare Graphen $|V_3| = \Omega(n)$ sowie $|V_4| = \Omega(n)$ und damit $|V'_3| \geq |V_3| - |V_3|/3 = \Omega(n)$ sowie $|V'_4| \geq |V_4| - |V_4|/4 = \Omega(n)$ mit Wahrscheinlichkeit $1 - 2^{-\Omega(n)}$. Nach Lemma 5.D.1.a beinhaltet jeder planare Graph höchstens $16n - 29$ Cliquen.

Nach Lemma 5.A.1.b berechnet die $(\mu+1)$ -RLS mit $\mu \geq 16n - 29$ (auf CLIQUE_G) daher in einer erwarteten Laufzeit von $\mathcal{O}(\mu n)$ eine größte Clique für jeden planaren Graphen. Bereits mit $\mu \leq n - 2$ benötigt die $(\mu+1)$ -RLS (auf CLIQUE_G) dazu für einige planare Graphen eine unendliche erwartete Laufzeit nach den Betrachtungen zum Lemma 5.A.1. Analog zum

Beweis des Theorems 3.A.4 benötigt nicht nur der $(\mu+1)$ -EA, sondern auch die $(\mu+1)$ -RLS mit $\mu \leq \text{poly}(n)$ (auf CLIQUE_G) schon eine erwartete Laufzeit von $\Omega(\mu n)$, um irgendeine Clique für irgendeinen planaren Graphen zu erzeugen, da zumindest jedes Element bestehend aus mehr als vier Einsen sicher keine Clique repräsentiert. Notwendigerweise können somit (asymptotisch) keine Unterschiede für die erwartete Laufzeit der $(\mu+1)$ -RLS mit $16n - 29 \leq \mu \leq \text{poly}(n)$ (auf CLIQUE_G) zur Erzeugung einer größten Clique von G für einen „schwersten“, zufälligen oder „einfachsten“ planaren Graphen G beobachtet werden.

Betrachten wir die bereits am Anfang dieses Kapitels motivierte kanonische Zielfunktion $\text{CLIQUE}_G^* : \{0, 1\}^n \rightarrow \mathbb{Z}$ mit $x = x_1 \cdots x_n \in \{0, 1\}^n$ und

$$\text{CLIQUE}_G^*(x) := \begin{cases} +|x| & , \text{ falls } \{v_i \mid x_i = 1, 1 \leq i \leq n\} \text{ eine Clique ist,} \\ -\infty & , \text{ falls } \{v_i \mid x_i = 1, 1 \leq i \leq n\} \text{ keine Clique ist.} \end{cases}$$

Ebenfalls am Anfang dieses Kapitels haben wir festgestellt, dass die Suchheuristiken für die Funktion CLIQUE_G^* anzupassen sind, indem die ersten Suchpunkte etwa lediglich aus Nullen bestehen und somit die 0-Clique repräsentieren. Dies ist die typische (nicht zufällige) Belegung der ersten Suchpunkte einer randomisierten Suchheuristik für das Cliquesproblem (Jerrum, 1992). Es sei bemerkt, dass sich die Ergebnisse der Abschnitte 5.A, 5.B und 5.C diesbezüglich kanonisch übertragen lassen. Wir werden weiter den Einsatz einer großen Population *oder* eines globalen Suchoperators untersuchen.

Als Erstes analysieren wir eine populationsbasierte randomisierte Suchheuristik – ähnlich zur $(\mu+1)$ -RLS. Die Population wird hier durch eine Multimenge von Suchpunkten beschrieben.

$(\mu+1)$ -RLS⁰ⁿ

1. Setze $t := 1$ und $x_{t,1} := 0^n, \dots, x_{t,\mu} := 0^n$. Setze $\mathcal{P}_t := \{x_{t,1}, \dots, x_{t,\mu}\}$.
2. Setze $t := t + 1$ und wähle $x_{t-1} \in \mathcal{P}_{t-1}$ zufällig gleichverteilt. Setze $y_t := \text{mutate}_1^*(x_{t-1})$.
3. Wähle $z_t \in \arg \min\{f(x) \mid x \in \mathcal{P}_{t-1} \cup \{y_t\}\}$ zufällig gleichverteilt. Falls $y_t \notin \mathcal{P}_{t-1}$, dann setze $\mathcal{P}_t := (\mathcal{P}_{t-1} \cup \{y_t\}) \setminus \{z_t\}$ und sonst $\mathcal{P}_t := \mathcal{P}_{t-1}$.
4. Gehe zu 2.

Wir können annehmen, dass der erste Schritt der $(\mu+1)$ -RLS⁰ⁿ im Gegensatz zur $(\mu+1)$ -RLS mit μ Funktionsauswertungen nur eine Funktionsauswertung durchführt.

Wir betrachten zunächst beliebige planare Graphen.

Theorem 5.D.2 Sei $\mu \geq 16n - 29$.

- a) Sei G ein beliebiger planarer Graph auf n Knoten. Die $(\mu+1)$ -RLS⁰ⁿ (auf CLIQUE_G^*) erzeugt eine größte Clique von G in einer erwarteten Laufzeit von $\mathcal{O}(\mu n)$.
- b) Es existieren planare Graphen G auf n Knoten, so dass die $(\mu+1)$ -RLS⁰ⁿ (auf CLIQUE_G^*) eine erwartete Laufzeit von $\Omega(\mu n)$ zur Erzeugung einer größten Clique von G benötigt.

Optimale Parameter für die $(\mu+1)$ -RLS⁰ⁿ sind $16n - 29 \leq \mu = \mathcal{O}(n)$. Hier resultiert eine erwartete Laufzeit von $\Theta(n^2)$ und die $(\mu+1)$ -RLS⁰ⁿ ist effizient genau dann, wenn $16n - 29 \leq \mu \leq \text{poly}(n)$ ist.

Beweis (Theorem 5.D.2).

- a) Wird eine nicht in der Population enthaltene Clique erzeugt, wird diese in die Population aufgenommen und eine Kopie von 0^n dafür gelöscht, da zum einen nach dem ersten Schritt keine Duplikate in die Population aufgenommen werden und zum anderen die Populationsgröße mindestens der Anzahl Cliques in G entspricht nach Lemma 5.D.1.a. Sei C eine größte Clique von G . Wenn die Population nur Teilmengen der Größe $\ell < |C| \leq 4$ von C umfasst, dann wird mindestens mit Wahrscheinlichkeit $1/(\mu n)$ eine solche Teilmenge der Größe mindestens $\ell + 1$ erzeugt. Insgesamt führt dies zu einer erwarteten Laufzeit von höchstens $\sum_{\ell=0}^{|C|-1} \mu n = \mathcal{O}(\mu n)$.
- b) Wir betrachten folgenden planaren Graphen G . Sei $E := \{\{v_1, v_2\}\}$. Solange die größte Clique nicht erzeugt ist, hat dies eine Wahrscheinlichkeit von höchstens $2/(\mu n)$, da genau eine spezielle Mutation von $\{v_1\}$ oder $\{v_2\}$ zu $\{v_1, v_2\}$ führt. Die Population umfasst jede solche 1-Clique höchstens einmal. Insgesamt führt dies zu einer erwarteten Laufzeit von $\Omega(\mu n)$.

Das gewünschte Resultat ist somit gezeigt. \square

Es sei bemerkt, dass hier die Population in einer erwarteten Laufzeit von höchstens

$$\sum_{\ell=0}^{n-1} \frac{\mu n}{n-\ell} + \sum_{\ell=0}^{3n-5} \frac{\mu n}{3n-6-\ell} + \sum_{\ell=0}^{6n-11} \frac{\mu n}{6n-12-\ell} + \sum_{\ell=0}^{6n-11} \frac{\mu n}{6n-12-\ell} = \mathcal{O}(\mu n \ln n)$$

jede Clique beinhaltet.

Wir betrachten nun zufällige planare Graphen.

Theorem 5.D.3 Sei $\mu \geq 16n - 29$ und G ein Graph bezüglich $\mathcal{G}_n^{\text{planar}}$ gewählt. Mit Wahrscheinlichkeit $1 - 2^{-\Omega(n)}$ und im Erwartungswert (bezüglich des Zufalls der Eingabe) erzeugt die $(\mu+1)$ -RLS $^{0^n}$ (auf CLIQUE_G^*) eine größte Clique von G in einer erwarteten (bezüglich des Zufalls der Suchheuristik) Laufzeit von $\Theta(\mu n^{2/3})$.

Optimale Parameter und solche, für welche die $(\mu+1)$ -RLS $^{0^n}$ effizient ist, sind die gleichen wie für Theorem 5.D.2.

Beweis (Theorem 5.D.3). Nach Lemma 5.D.1.b enthält der Graph G mindestens $\Omega(n)$ lose eingebundene 3-Cliques und 4-Cliques mit Wahrscheinlichkeit $1 - 2^{-\Omega(n)}$. Wir nehmen dies an. Die Ergebnisse für die Erwartungswerte folgen direkt.

Betrachten wir zunächst die obere Schranke. Solange die Population keine 4-Clique beinhaltet, enthält sie mindestens $\mu - (10n - 17)$ viele 0-Cliques nach Lemma 5.D.1.a. Mindestens mit Wahrscheinlichkeit $\frac{\mu - 10n + 17}{\mu} \geq 1/4$ wird dann eine 0-Clique zur Mutation ausgewählt. Falls die Population $0 \leq \ell < n$ viele 1-Cliques umfasst, dann wird mindestens mit Wahrscheinlichkeit $(1/4) \cdot (n - \ell)/n$ eine nicht in der Population enthaltene 1-Clique erzeugt. In einer erwarteten Laufzeit von $\sum_{\ell=0}^{n-1} 4n/(n - \ell) = \mathcal{O}(n \ln n)$ beinhaltet die Population alle 1-Cliques. Danach und solange die Population weniger als $n^{2/3}$ viele 2-Cliques im durch V_4 induzierten Graphen enthält, wird eine solche, nicht in der Population enthaltene 2-Clique

mindestens mit Wahrscheinlichkeit $(\Omega(n) - n^{2/3})/(\mu n) = \Omega(1/\mu)$ erzeugt. In einer erwarteten Laufzeit von $n^{2/3} \cdot \mathcal{O}(\mu) = \mathcal{O}(\mu n^{2/3})$ beinhaltet die Population mindestens $n^{2/3}$ dieser 2-Cliques. Analog enthält die Population in einer erwarteten Laufzeit von $\mathcal{O}(\mu n^{2/3})$ mindestens $n^{1/3}$ viele 3-Cliques im durch V_4 induzierten Graphen. Eine größte Clique wird dann in einer erwarteten Laufzeit von höchstens $(\mu/n^{1/3}) \cdot n = \mathcal{O}(\mu n^{2/3})$ generiert. Insgesamt resultiert eine erwarteten Laufzeit von $\mathcal{O}(n \ln n) + \mathcal{O}(\mu n^{2/3}) + \mathcal{O}(\mu n^{2/3}) + \mathcal{O}(\mu n^{2/3}) = \mathcal{O}(\mu n^{2/3})$.

Betrachten wir nun die untere Schranke. Im Weiteren beschränken wir die Wahrscheinlichkeit durch $1/4$ nach oben, in einer Laufzeit von $\mu n^{2/3}/6$ eine größte Clique zu erzeugen. Insgesamt resultiert somit eine erwartete Laufzeit von mindestens $(3/4) \cdot \mu n^{2/3}/6 = \Omega(\mu n^{2/3})$. Bis auf die 0-Cliques (des ersten Schritts) entstehen alle Cliques der Population durch Mutation einer um genau eins kleineren oder größeren Clique. Dies kann durch eine (zufällige) Folge von unterschiedlichen Cliques beschrieben werden. Seien $x_{(\ell)}$ und $y_{(\ell)}$, $1 \leq \ell \leq 4$, ℓ -Cliques. Jede erste auftretende 4-Clique entsteht *direkt* durch eine Folge von Cliques $0^n, \dots, x_{(1)}, x_{(2)}, x_{(3)}, x_{(4)}$ (Fall 1) oder *indirekt* durch eine Folgen von Cliques $0^n, \dots, y_{(2)}, y_{(3)}, x_{(2)}, x_{(3)}, x_{(4)}$ (Fall 2). Es bleibt die Wahrscheinlichkeit des Auftretens einer solchen Folge in einer Laufzeit von $\mu n^{2/3}/6$ durch $1/4$ nach oben zu beschränken.

1. Sei $x_{(1)}$ ein Element der Population. Die Anzahl betrachteter Enden $x_{(1)}, x_{(2)}, x_{(3)}, x_{(4)}$ von Folgen beträgt höchstens $4 \cdot 3 \cdot 2 \cdot 1 \cdot (6n - 12) \leq 144n$ nach Lemma 5.D.1.a. Ein spezielles solches Ende einer Folge tritt höchstens mit Wahrscheinlichkeit $\binom{\mu n^{2/3}}{3} \cdot 1/(\mu n)^3$ in den höchstens $\mu n^{2/3}/6$ verbleibenden Schritten auf, da drei Mal eine spezielle 1-Bit-Mutation eines speziellen Elements auftreten muss. Damit beträgt die Wahrscheinlichkeit, eine 4-Clique zu erzeugen, höchstens

$$144n \cdot \binom{\frac{\mu n^{2/3}}{6}}{3} \cdot \left(\frac{1}{\mu n}\right)^3 \leq \frac{(\mu^3 n^2) \cdot (144n)}{(3! \cdot 6^3) \cdot (\mu^3 n^3)} = \frac{1}{9}.$$

2. Sei $y_{(2)}$ ein Element der Population und beschreibe $\ell_{y_{(3)}}$ die Anzahl 4-Cliques, mit denen die 3-Clique $y_{(3)}$ mindestens eine gemeinsame Kante hat. Sei $V' \subseteq V$ die Menge der zu mindestens einem Knoten aus $y_{(3)}$ adjazenten Knoten. Dann gilt $\sum_{v \in y_{(3)}} \deg(v) \geq \ell_{y_{(3)}}/7$. Sonst würden im durch V' induzierten planaren Graphen, der alle betrachteten 4-Cliques enthält, widersprüchlich höchstens $6 \cdot (3 + \ell_{y_{(3)}}/7 - 1) - 12 = 6\ell_{y_{(3)}}/7 < \ell_{y_{(3)}}$ viele 4-Cliques existieren nach Lemma 5.D.1.a. Wir betrachten alle 3-Cliques $y_{(3)}$ mit $\ell_1 \leq \ell_{y_{(3)}} < \ell_2 \leq 6n$. Es gibt höchstens $\min\{(6n - 12)/(\ell_1/7), 6n - 12\} \leq \min\{42n/\ell_1, 6n\}$ solche 3-Cliques, da sowohl die gesamte Anzahl 3-Cliques höchstens $6n - 12$ als auch die Summe der Grade der Knoten höchstens $2 \cdot (3n - 6)$ beträgt. Die Anzahl betrachteter Enden $y_{(2)}, y_{(3)}, x_{(2)}, x_{(3)}, x_{(4)}$ von Folgen mit irgendeinem $y_{(3)}$, wobei $\ell_1 \leq \ell_{y_{(3)}} < \ell_2$ ist, beträgt höchstens $\min\{42n/\ell_1, 6n\} \cdot 3 \cdot 2 \cdot 2 \cdot 1 \cdot \ell_2 = \min\{42n/\ell_1, 6n\} \cdot 12 \cdot \ell_2$. Wir unterscheiden $\ell_1 = n^{2/3}$, $\ell_2 = 6n$ (Fall 2.i), $\ell_1 = n^{1/3}$, $\ell_2 = n^{2/3}$ (Fall 2.ii) und $\ell_1 = 1$, $\ell_2 = n^{1/3}$ (Fall 2.iii). Analog zum obigen Fall beträgt die Wahrscheinlichkeit, eine 4-Clique zu erzeugen, höchstens:

i.

$$\frac{42n}{n^{2/3}} \cdot 12 \cdot 6n \cdot \left(\frac{\mu n^{2/3}}{6}\right) \cdot \left(\frac{1}{\mu n}\right)^4 \leq \frac{(\mu^4 n^{8/3}) \cdot (3024 n^{4/3})}{(4! \cdot 6^4) \cdot (\mu^4 n^4)} = \frac{7}{72}$$

ii.

$$\frac{42n}{n^{1/3}} \cdot 12 \cdot n^{2/3} \cdot \left(\frac{\mu n^{2/3}}{6}\right) \cdot \left(\frac{1}{\mu n}\right)^4 \leq \frac{(\mu^4 n^{8/3}) \cdot (504 n^{4/3})}{(4! \cdot 6^4) \cdot (\mu^4 n^4)} = \frac{7}{432}$$

iii.

$$6n \cdot 12 \cdot n^{1/3} \cdot \left(\frac{\mu n^{2/3}}{6}\right) \cdot \left(\frac{1}{\mu n}\right)^4 \leq \frac{(\mu^4 n^{8/3}) \cdot (72 n^{4/3})}{(4! \cdot 6^4) \cdot (\mu^4 n^4)} = \frac{1}{432}$$

Höchstens mit Wahrscheinlichkeit $\frac{1}{9} + \frac{7}{72} + \frac{7}{432} + \frac{1}{432} \leq \frac{1}{4}$ wird in einer Laufzeit von $\mu n^{2/3}/6$ eine größte Clique generiert. \square

Die Resultate des Theorems 5.D.2.a und des Theorems 5.D.3.a gelten auch für den (sich kanonischerweise ergebenen) $(\mu+1)$ -EA $^{0^n}$. Aber diesem $(\mu+1)$ -EA $^{0^n}$ mit $\mu \geq 16n - 29$ genügt andererseits eine erwartete Laufzeit von $\mathcal{O}(n^4)$ für beliebige planare Graphen und $\mathcal{O}(n^3)$ für zufällige planare Graphen. Die erwartete Laufzeit ist unabhängig von der Populationsgröße beschränkt. Solange keine größte Clique berechnet und $\mu \geq 20n$ ist, erzeugt jede Mutation eine solche Clique mindestens mit Wahrscheinlichkeit $\frac{\mu-10n+17}{\mu} \cdot 1/(en^4) \geq 1/(2en^4)$ für beliebige planare Graphen und $\frac{\mu-10n+17}{\mu} \cdot \Omega(n)/(en^4) = \Omega(1/(2en^3))$ für zufällige planare Graphen nach Lemma 5.D.1.

Als zweites analysieren wir eine mutationsbasierte randomisierte Suchheuristik – ähnlich zum $(1+1)$ -EA.

$(1+1)$ -EA $^{0^n}$

1. Setze $t := 1$ und $x_t := 0^n$.
2. Setze $t := t + 1$ und $y_t := \text{mutate}_{1/n}(x_{t-1})$.
3. Falls $f(y_t) \geq f(x_{t-1})$, dann setze $x_t := y_t$ und sonst $x_t := x_{t-1}$.
4. Gehe zu 2.

Wir betrachten zunächst beliebige planare Graphen.

Theorem 5.D.4

- a) Sei G ein beliebiger planarer Graph auf n Knoten. Der $(1+1)$ -EA $^{0^n}$ (auf CLIQUE_G^*) erzeugt eine größte Clique von G in einer erwarteten Laufzeit von $\mathcal{O}(n^6)$.
- b) Es existieren planare Graphen G auf n Knoten, so dass der $(1+1)$ -EA $^{0^n}$ (auf CLIQUE_G^*) eine erwartete Laufzeit von $\Omega(n^6)$ zur Erzeugung einer größten Clique von G benötigt.

Nach Theorem 5.A.2 und Theorem 5.B.3 folgt das gleiche Resultat für den $(1+1)$ -EA.

Beweis (Theorem 5.D.4).

- a) Dies folgt analog zum Beweis des Theorems 5.A.2. Sei C eine größte Clique von G . In einer erwarteten Laufzeit von $\mathcal{O}(n^6)$ wird eine Teilmenge von C der Größe mindestens $|C| - 1$ erzeugt. Danach wird mindestens mit Wahrscheinlichkeit $1/(en)$ eine größte Clique generiert und höchstens mit Wahrscheinlichkeit $(6n - 13) \cdot 1/n^2 \leq 6/n$ eine der höchstens $6n - 13$ anderen $(|C| - 1)$ -Cliques nach Lemma 5.D.1.a. Daher folgt in einer erwarteten Laufzeit von $E \leq 1 + \frac{1}{en} \cdot 0 + \frac{6}{n} \cdot \mathcal{O}(n^6) + (1 - \frac{6}{n} - \frac{1}{en}) \cdot E$ und äquivalent $E = \mathcal{O}(n^6)$ das Erzeugen einer größten Clique. Die genannte erwartete Laufzeit folgt direkt.
- b) Dies folgt analog zum Beweis des Theorems 5.B.3. Wir betrachten den planaren Graphen $W_n := (V, \{\{v_i, v_j\} \mid 1 \leq i < j \leq 3 \text{ oder } n - 3 \leq i < j \leq n\})$. Mit Wahrscheinlichkeit $\Omega(1)$ erzeugt die erste Mutation eine 1-Clique $\{v_4\}, \dots, \{v_{n-4}\}$ und danach mit Wahrscheinlichkeit $\Omega(1)$ eine 1-Clique $\{v_1\}, \{v_2\}, \{v_3\}$ vor einer nicht leeren Clique $C \subseteq \{v_{n-3}, \dots, v_n\}$. Es resultiert, dass jeweils mit Wahrscheinlichkeit $\Omega(1)$ eine 2-Clique $\{v_1, v_2\}, \{v_1, v_3\}, \{v_2, v_3\}$ und die 3-Clique $\{v_1, v_2, v_3\}$, welche wiederum höchstens mit Wahrscheinlichkeit $\binom{4}{3}/n^6 + \binom{4}{4}/n^7 = \mathcal{O}(1/n^6)$ ersetzt wird, vor einer Clique $C \subseteq \{v_{n-3}, \dots, v_n\}$ erzeugt wird. Insgesamt resultiert eine erwartete Laufzeit von $\Omega(1) \cdot \Omega(1) \cdot \Omega(1) \cdot \Omega(1) \cdot \Omega(n^6) = \Omega(n^6)$.

Das gewünschte Resultat ist somit gezeigt. \square

Wir betrachten nun zufällige planare Graphen.

Theorem 5.D.5 *Sei G ein Graph bezüglich $\mathcal{G}_n^{\text{planar}}$ gewählt. Mit Wahrscheinlichkeit $1 - 2^{-\Omega(n)}$ und im Erwartungswert (bezüglich des Zufalls der Eingabe) erzeugt der $(1+1)$ -EA $^{0^n}$ (auf CLIQUE_G^*) eine größte Clique von G in einer erwarteten (bezüglich des Zufalls der Suchheuristik) Laufzeit von $\Theta(n^5)$.*

Beweis. Nach Lemma 5.D.1.b enthält der Graph G mindestens $\Omega(n)$ lose eingebundene 3-Cliques und 4-Cliques mit Wahrscheinlichkeit $1 - 2^{-\Omega(n)}$. Wir nehmen dies an. Die Ergebnisse für die Erwartungswerte folgen direkt.

Betrachten wir zunächst die obere Schranke. Analog zum Beweis des Theorems 5.D.4 wird mindestens mit Wahrscheinlichkeit $\binom{4}{3} \cdot \Omega(n) \cdot 1/(en^6)$, also in einer erwarteten Laufzeit von $\mathcal{O}(n^5)$, eine Teilmenge einer (lose eingebundenen) 4-Clique der Größe mindestens drei erzeugt. Danach wird mindestens mit Wahrscheinlichkeit $1/(en)$ eine größte Clique generiert und höchstens mit Wahrscheinlichkeit $6/n$ eine andere 3-Clique. Dann resultiert eine erwartete Laufzeit von $\mathcal{O}(n^5)$ für das Erzeugen einer größten Clique. Die genannte erwartete Laufzeit folgt direkt.

Betrachten wir nun die untere Schranke. Mit Wahrscheinlichkeit $\Omega(1)$ erzeugt die erste Mutation eine 1-Clique $\{v\} \subseteq V_3$. Es resultiert, dass jeweils mit Wahrscheinlichkeit $\Omega(1)$ eine 2-Clique und eine 3-Clique, die Teilmengen einer lose eingebundenen 3-Clique sind, vor einer anderen Clique erzeugt werden. Eine lose eingebundene 3-Clique wird höchstens mit Wahrscheinlichkeit $(6n - 13) \cdot 1/n^6 + (6n - 12) \cdot 1/n^7 = \mathcal{O}(1/n^5)$ ersetzt. Insgesamt resultiert eine erwartete Laufzeit von $\Omega(1) \cdot \Omega(1) \cdot \Omega(1) \cdot \Omega(n^5) = \Omega(n^5)$. \square

Betrachten wir den (sich kanonischerweise ergebenden) $(1+\lambda)$ -EA $^{0^n}$. Jeder Nachkomme des zweiten Schritts erzeugt mindestens mit Wahrscheinlichkeit $1/(en^4)$ eine größte Clique für jeden planaren Graphen, während dies mindestens mit Wahrscheinlichkeit $1/(en^7)$ für jeden Nachkommen weiterer Schritte gilt. Es resultiert eine erwartete Laufzeit von höchstens

$$1 + en^4 + \left(1 - \frac{1}{en^4}\right)^\lambda \cdot (\lambda + en^7).$$

Mit $\lambda \geq 3en^4 \ln n$ genügt also eine erwartete Laufzeit von $\mathcal{O}(n^4)$ zur Erzeugung einer größten Clique eines beliebigen planaren Graphen. Die erwartete Laufzeit ist unabhängig von der Nachkommenpopulationsgröße beschränkt. Für zufällige planare Graphen resultiert analog eine erwartete Laufzeit von $\mathcal{O}(n^3) + (1 - \Omega(1/n^3))^\lambda \cdot (\lambda + en^7)$ nach Lemma 5.D.1.b. Mit $\lambda \geq cn^3 \ln n$ für eine genügend große Konstante c genügt also eine erwartete Laufzeit von $\mathcal{O}(n^3)$ zu Erzeugung einer größten Clique eines zufälligen planaren Graphen.

5.D.1 Weitere randomisierte Suchheuristiken

Neben der Verwendung einer großen Population oder eines globalen Suchoperators sind die Verwendung einer Restart-Strategie oder einer Metropolis-Selektion wohl die bekanntesten Komponenten zur Vermeidung der schnellen Stagnation in lokalen Optima.

Als Erstes analysieren wir eine Variante des $(1+1)$ -RLS $^{0^n}$, welche eine Restart-Strategie verwendet (Jansen, 2002) – eine Restart randomisierte lokale Suche.

$(1+1)$ -RRLS $_r^{0^n}$

1. Setze $t := 1$.
2. Setze $\ell := 1$ und $x_t := 0^n$.
3. Setze $t := t + 1$, $\ell := \ell + 1$ und $y_t := \text{mutate}_1^*(x_{t-1})$.
4. Falls $f(y_t) \geq f(x_{t-1})$, dann setze $x_t := y_t$ und sonst $x_t := x_{t-1}$.
5. Falls $\ell = r$, gehe zu 2. und sonst gehe zu 3.

Wie bemerken, dass die $(1+1)$ -RRLS $_r^{0^n}$ mit $r = \infty$ hier der $(1+1)$ -RLS $^{0^n}$ entspricht. Die r Schritte zwischen Restarts bezeichnen wir als einen Lauf.

Diese randomisierte Suchheuristik erlaubt eine Analyse der Effekte der Restart-Strategie und vermeidet zugleich Einflüsse weiterer Komponenten solcher Heuristiken.

Weiterhin stellen wir fest, dass die $(1+1)$ -RRLS $_r^{0^n}$ höchstens Cliques der Größe $r-1$ erzeugen kann. Daher genügt hier die Untersuchung von Werten $r \geq 5$.

Wir betrachten zunächst beliebige planare Graphen.

Theorem 5.D.6 Sei $r \geq 5$.

- Sei G ein beliebiger planarer Graph auf n Knoten. Die $(1+1)$ -RRLS $_r^{0^n}$ (auf CLIQUE_G^*) erzeugt eine größte Clique von G in einer erwarteten Laufzeit von $\mathcal{O}(n^4 + rn^3)$.
- Es existieren planare Graphen G auf n Knoten, so dass die $(1+1)$ -RRLS $_r^{0^n}$ (auf CLIQUE_G^*) eine erwartete Laufzeit von $\Omega(n^4 + rn^3)$ zur Erzeugung einer größten Clique von G benötigt.

Optimale Parameter für die $(1+1)$ -RRLS $_r^{0^n}$ sind $5 \leq r = \mathcal{O}(n)$. Hier resultiert eine erwartete Laufzeit von $\Theta(n^4)$ und die $(1+1)$ -RRLS $_r^{0^n}$ ist effizient genau dann, wenn $5 \leq r \leq \text{poly}(n)$ ist.

Beweis (Theorem 5.D.6).

- Sei C eine größte Clique von G . Mindestens mit Wahrscheinlichkeit $\prod_{i=0}^{|C|-2} 1/n = \Omega(1/n^3)$ wird eine Teilmenge von C der Größe $|C| - 1 \leq 3$ im $|C|$ -ten Schritt eines Laufs generiert. Danach wird mindestens mit Wahrscheinlichkeit $1/n$ in jedem der verbleibenden $r - |C| \geq r - 4$ Schritte eines Laufs eine größte Clique generiert, also insgesamt mindestens mit Wahrscheinlichkeit $1 - (1 - 1/n)^{r-4}$. Falls $r < n$ ist, gilt $1 - (1 - 1/n)^{r-4} \geq (r - 4)/(2n) = \Omega(r/n)$ und ein Lauf erzeugt eine größte Clique mit Wahrscheinlichkeit $\Omega(1/n^3) \cdot \Omega(r/n) = \Omega(r/n^4)$. Falls $r \geq n$ ist, gilt $1 - (1 - 1/n)^{r-4} \geq 1 - e^{-(r-4)/n} = \Omega(1)$ und ein Lauf erzeugt eine größte Clique mit Wahrscheinlichkeit $\Omega(1/n^3) \cdot \Omega(1) = \Omega(1/n^3)$. In einer erwarteten Anzahl von $\mathcal{O}(n^4/r + n^3)$ Läufen wird eine größte Clique generiert. Insgesamt resultiert eine erwartete Laufzeit von höchstens $r \cdot \mathcal{O}(n^4/r + n^3) = \mathcal{O}(n^4 + rn^3)$.
- Wir betrachten folgenden planaren Graphen G . Sei $\ell := \lfloor (n - 4)/6 \rfloor$, $V_0 := \{v_1, \dots, v_4\}$ und $V_i := \{v_{5+(i-1)\ell}, \dots, v_{4+i\ell}\}$, $1 \leq i \leq 6$. Sei $e_1 := \{v_1, v_2\}$, $e_2 := \{v_1, v_3\}$, $e_3 := \{v_1, v_4\}$, $e_4 := \{v_2, v_3\}$, $e_5 := \{v_2, v_4\}$, $e_6 := \{v_3, v_4\}$ und $E_i := \{\{v, u\}, \{v, w\} \mid v \in V_i, e_i = \{u, w\}\}$, $1 \leq i \leq 6$. Damit sei $E := \bigcup_{i=1}^6 E_i \cup \{e_i\}$. Die größte Clique von G ist V_0 .
Im zweiten Schritt jedes Laufs wird mit Wahrscheinlichkeit $4/n$ eine Teilmenge von V_0 der Größe eins erzeugt. Danach wird mit Wahrscheinlichkeit $(3/n)/(3/n + 3\ell/n) = \mathcal{O}(1/n)$ und dann $(2/n)/(2/n + \ell/n) = \mathcal{O}(1/n)$ eine Teilmenge von V_0 der Größe zwei und dann drei vor einer anderen 2-Clique und dann 3-Clique erzeugt. Bestenfalls nach vier Schritten ist eine solche 3-Clique generiert worden und, um V_0 in diesem Lauf zu erzeugen, muss in den verbleibenden $r - 4$ Schritten eine spezielle 1-Bit-Mutation durchgeführt werden. Dies passiert höchstens mit Wahrscheinlichkeit $\min\{(r - 4)/n, 1\}$. In einem Lauf wird mit Wahrscheinlichkeit $\min\{\mathcal{O}((r - 4)/n^4), \mathcal{O}(1/n^3)\}$ die größte Clique generiert. Insgesamt resultiert eine erwartete Laufzeit von mindestens $r \cdot (\Omega(n^4/r + n^3) - 1) = \Omega(n^4 + rn^3)$, da alle bis auf den letzten Lauf keine größte Clique erzeugt haben.

Das gewünschte Resultat ist somit gezeigt. \square

Wir betrachten nun zufällige planare Graphen.

Theorem 5.D.7 Sei $r \geq 5$ und G ein Graph bezüglich $\mathcal{G}_n^{\text{planar}}$ gewählt. Mit Wahrscheinlichkeit $1 - 2^{-\Omega(n)}$ und im Erwartungswert (bezüglich des Zufalls der Eingabe) erzeugt die $(1+1)$ -RRLS $_r^{0^n}$ (auf CLIQUE_G^*) eine größte Clique von G in einer erwarteten (bezüglich des Zufalls der Suchheuristik) Laufzeit von $\Theta(n^3/r^2 + r)$.

Optimale Parameter für die $(1+1)$ -RRLS $_r^{0^n}$ sind $r = \Theta(n)$. Hier resultiert eine erwartete Laufzeit von $\Theta(n)$ und die $(1+1)$ -RRLS $_r^{0^n}$ ist effizient genau dann, wenn $5 \leq r \leq \text{poly}(n)$ ist. Betrachten wir den für Theorem 5.D.6 optimalen Parameter $r = 5$, so resultiert hier eine erwartete Laufzeit von $\Theta(n^3)$, also bei weitem keine optimale erwartete Laufzeit.

Beweis (Theorem 5.D.7). Nach Lemma 5.D.1.b enthält der Graph G mindestens $\Omega(n)$ lose eingebundene 3-Cliquen und 4-Cliquen mit Wahrscheinlichkeit $1 - 2^{-\Omega(n)}$. Wir nehmen dies an. Die Ergebnisse für die Erwartungswerte folgen direkt.

- a) Im zweiten Schritt eines Laufs wird mit Wahrscheinlichkeit $\Omega(n)/n = \Omega(1)$ eine Clique $\{v\} \subseteq V_4'$ generiert. Für die lose eingebundene 4-Clique C gelte $C \supseteq \{v\}$. Wir betrachten die verbleibenden $r - 2$ Schritte eines Laufs in drei Phasen jeweils der Länge $(r - 2)/3 \geq 1$. Sei $C' \subseteq C$ mit $1 \leq |C'| \leq |C| - 1$ das betrachtete Individuum. Analog zum Beweis des Theorems 5.D.6.a beträgt die Wahrscheinlichkeit, eine Teilmenge von $|C|$ der Größe $|C'| + 1$ zu erzeugen, $\Omega(r/n)$, falls $r < n$ ist, und $\Omega(1)$, falls $r \geq n$ ist. Es resultiert eine Wahrscheinlichkeit von $\Omega(1) \cdot \Omega(r/n) \cdot \Omega(r/n) \cdot \Omega(r/n) = \Omega(r^3/n^3)$ und $\Omega(1) \cdot \Omega(1) \cdot \Omega(1) \cdot \Omega(1) = \Omega(1)$, in einem Lauf eine größte Clique zu generieren. In einer erwarteten Anzahl von $\mathcal{O}(n^3/r^3 + 1)$ Läufen wird eine solche Clique generiert. Insgesamt resultiert eine erwartete Laufzeit von höchstens $r \cdot \mathcal{O}(n^3/r^3 + 1) = \mathcal{O}(n^3/r^2 + r)$.
- b) Wir unterscheiden $r < n$ (Fall 1) und $r \geq n$ (Fall 2).

1. Sei $x_{(\ell)}$, $1 \leq \ell \leq 4$, eine ℓ -Clique. Analog zum Beweis des Theorems 5.D.3 entsteht hier jede 4-Clique (direkt) durch eine Folge von Cliquen $0^n, x_{(1)}, x_{(2)}, x_{(3)}, x_{(4)}$. Mit Wahrscheinlichkeit $1/n$ entsteht $x_{(1)}$ im zweiten Schritt eines Laufs. Die Anzahl betrachteter Folgen ist höchstens $4 \cdot 3 \cdot 2 \cdot 1 \cdot (6n - 12) \leq 144n$ nach Lemma 5.D.1.a. Eine spezielle solche Folge tritt höchstens mit Wahrscheinlichkeit $\binom{r-2}{3} \cdot 1/n^3$ in den $r - 2$ verbleibenden Schritten eines Laufs auf. Damit beträgt die Wahrscheinlichkeit, eine 4-Clique in einem Lauf zu erzeugen, höchstens

$$144n \cdot \frac{1}{n} \cdot \binom{r-2}{3} \cdot \left(\frac{1}{n}\right)^3 \leq 144 \cdot \frac{r^3}{3!} \cdot \frac{1}{n^3} = \frac{r^3}{24n^3}.$$

Analog zum Beweis des Theorems 5.D.6.a resultiert eine erwartete Laufzeit von mindestens $r \cdot (24n^3/r^3 - 1) = \Omega(n^3/r^2)$, da $r < n$ ist.

2. Mit Wahrscheinlichkeit $\Omega(n)/n = \Omega(1)$ entsteht im zweiten Schritt des ersten Laufs eine 1-Clique $\{v\} \subseteq V_3$ und ein zweiter Lauf ist notwendig. Es resultiert eine erwartete Laufzeit von mindestens $r \cdot \Omega(1) = \Omega(r)$.

Das Ergebnis folgt, da $\Omega(n^3/r^2 + r) = \Omega(n^3/r^2)$ für $r < n$ und $\Omega(n^3/r^2 + r) = \Omega(r)$ für $r \geq n$ ist.

Das gewünschte Resultat ist somit gezeigt. □

Als zweites analysieren wir eine Variante des $(1+1)$ -RLS $^{0^n}$, welche eine Metropolis-Selektion verwendet (Jerrum, 1992) – einen Metropolis-Algorithmus.

MA_T⁰ⁿ

1. Setze $t := 1$ und $x_t := 0^n$.
2. Setze $t := t + 1$ und $y_t := \text{mutate}_1^*(x_{t-1})$.
3. Mit Wahrscheinlichkeit $\min\{e^{T \cdot (f(y_t) - f(x_{t-1}))}, 1\}$, setze $x_t := y_t$ und sonst $x_t := x_{t-1}$.
4. Gehe zu 2.

Wir bemerken, dass der MA_T⁰ⁿ mit $T = \infty$ hier der (1+1)-RLS⁰ⁿ entspricht.

Diese randomisierte Suchheuristik erlaubt eine Analyse der Effekte der Metropolis-Selektion und vermeidet zugleich Einflüsse weiterer Komponenten solcher Heuristiken.

Weiterhin stellen wir fest, dass der MA_T⁰ⁿ mit $T < 0$ Verbesserungen nicht mehr mit Wahrscheinlichkeit 1, dafür hingegen Verschlechterungen sicher akzeptiert. Daher genügt hier eine Untersuchung von Werten $T \geq 0$.

Wir führen Ergebnisse über Markoffprozesse und Irrläufe ein, bevor wir die gewünschten Resultate beweisen. Markoffprozesse haben wir bereits in Abschnitt 2.A ausführlich betrachtet. Ein *Irrlauf* (engl. *random walk*) \mathcal{R} ist durch einen ungerichteten und kantengewichteten Graphen $W = (V, E)$ mit den Gewichten $r_{\{u,w\}} \geq 1$ für $\{u,w\} \in E$ und einem Knoten $v \in V$ beschrieben. Befindet sich der Irrlauf im Knoten $v \in V$, so findet mit Wahrscheinlichkeit $r_{\{v,w\}} / \sum_{\{v,u\} \in E} r_{\{v,u\}}$ ein Wechsel in Knoten $w \in V$ mit $\{v,w\} \in E$ statt. (Eine ausführliche Betrachtung von Irrläufen findet sich in Mitzenmacher und Upfal (2005) sowie in Motwani und Raghavan (1995).) Aussage a) des folgenden Lemmas ergibt sich nach Motwani und Raghavan (1995) und Aussage b) analog zu den Betrachtungen von Chandra et al. (1989).

Lemma 5.D.8

- a) Der Markoffprozess mit m Zuständen sei irreduzibel und aperiodisch. Dann existiert eine eindeutige stationäre Verteilung π auf $\{0, \dots, m-1\}$ mit $\pi_0, \dots, \pi_{m-1} > 0$, und die erwartete Anzahl Schritte, sich in Zustand i befindend in Zustand i zurückzukehren, beträgt $1/\pi_i$ für $0 \leq i \leq m-1$.
- b) Der Irrlauf befinde sich im Knoten v . Dann beträgt die erwartete Anzahl Schritte, Knoten w zu erreichen und dann in Knoten v zurückzukehren, höchstens $2|E| \cdot R_{v,w}$, wobei $R_{v,w} = r_{\{v,u_1\}} + \dots + r_{\{u_{k-1},w\}}$ für einen beliebigen Weg $\{v, u_1\}, \dots, \{u_{k-1}, w\} \in E$.

Wir betrachten zunächst beliebige planare Graphen.

Theorem 5.D.9 Sei $T \geq 0$.

- a) Sei G ein beliebiger planarer Graph auf n Knoten. Der MA_T⁰ⁿ (auf CLIQUE_G^{*}) erzeugt eine größte Clique von G in einer erwarteten Laufzeit von $\mathcal{O}(n^2 e^{4T})$.
- b) Es existieren planare Graphen G auf n Knoten, so dass der MA_T⁰ⁿ eine erwartete Laufzeit von $\Omega(n^2 e^T)$ (auf CLIQUE_G^{*}) zur Erzeugung einer größten Clique von G benötigt.

Optimale Parameter für den MA_T⁰ⁿ sind $0 \leq T = \mathcal{O}(1)$. Hier resultiert eine erwartete Laufzeit von $\Theta(n^2)$ und der MA_T⁰ⁿ ist effizient genau dann, wenn $0 \leq T = \mathcal{O}(\ln n)$ ist.

Beweis (Theorem 5.D.9).

- a) Für jede Clique C beträgt die Wahrscheinlichkeit, in eine Clique C' zu wechseln,
- $1/n$, falls $C' \supseteq C$ mit $|C'| = |C| + 1$ ist,
 - e^{-T}/n , falls $C' \subseteq C$ mit $|C'| = |C| - 1$ ist und
 - 0 für alle anderen Cliques $C' \neq C$.

In einer erwarteten Laufzeit von höchstens ne^T wird jede Clique also verlassen. Betrachten wir den Algorithmus, der in jedem Schritt die Clique wechselt. Äquivalent dazu können wir einen Irrlauf auf den folgenden zusammenhängenden Graphen W betrachten (vergleiche Jerrum, 1992). Die Knoten entsprechen den Cliques und eine Kante $\{C, C'\}$ zwischen einer ℓ -Clique C und einer $(\ell + 1)$ -Clique C' existiert, falls $C \subseteq C'$ ist, und hat das Gewicht $r_{\{C, C'\}} = e^{T-\ell} \geq 1$. Die Wahrscheinlichkeit, im betrachteten Algorithmus von C nach C' zu wechseln, beträgt $(1/n)/(\sum_{C^+ \supseteq C, |C^+|=|C|+1} 1/n + \sum_{C^- \subseteq C, |C^-|=|C|-1} e^{-T}/n) = r_{\{C, C'\}} / \sum_{\{C, C''\} \in E} r_{\{C, C''\}}$ und $(e^{-T}/n)/(\sum_{C^+ \supseteq C', |C^+|=|C'+1} 1/n + \sum_{C^- \subseteq C', |C^-|=|C'|-1} e^{-T}/n) = r_{\{C, C'\}} / \sum_{\{C', C''\} \in E} r_{\{C', C''\}}$ für einen Wechsel von C' nach C . Wir wollen Lemma 5.D.8.b anwenden. Der Graph W beinhaltet höchstens $1 \cdot n + 2 \cdot (3n - 6) + 3 \cdot (6n - 12) + 4 \cdot (6n - 12) = \mathcal{O}(n)$ Kanten und die Summe der Gewichte auf einem (kürzesten) Weg zwischen der 0-Clique und einer beliebigen größten Clique beträgt höchstens $e^{T \cdot 0} + e^{T \cdot 1} + e^{T \cdot 2} + e^{T \cdot 3} = \mathcal{O}(e^{3T})$. Eine Anwendung des Lemmas 5.D.8.b liefert eine erwartete Laufzeit von $\mathcal{O}(n) \cdot \mathcal{O}(e^{3T}) = \mathcal{O}(ne^{3T})$, bis ein Irrlauf in der 0-Clique befindend eine 4-Clique erreicht (und sogar wieder zurückkehrt). Insgesamt resultiert für den MA_T^{0n} eine erwartete Laufzeit von höchstens $ne^T \cdot \mathcal{O}(ne^{3T}) = \mathcal{O}(n^2 e^{4T})$.

- b) Wir betrachten folgenden planaren Graphen G . Sei $E := \{\{v_1, v_2\}\}$. Um $\{v_1, v_2\}$ zu erzeugen, muss zuvor eine der Cliques $\{v_1\}$ oder $\{v_2\}$ generiert werden. Die 0-Clique erzeugt eine solche 1-Clique mit Wahrscheinlichkeit $2/n$. Eine Clique $\{v_3\}, \dots, \{v_n\}$ generiert die 0-Clique mit Wahrscheinlichkeit e^{-T}/n . Insgesamt resultiert eine erwartete Laufzeit von mindestens $ne^T \cdot (n/2 - 1) = \Omega(n^2 e^T)$.

Das gewünschte Resultat ist somit gezeigt. \square

Wir betrachten nun zufällige planare Graphen.

Theorem 5.D.10 Sei $T \geq 0$ und G ein Graph bezüglich $\mathcal{G}_n^{\text{planar}}$ gewählt. Mit Wahrscheinlichkeit $1 - 2^{-\Omega(n)}$ und im Erwartungswert (bezüglich des Zufalls der Eingabe) erzeugt der MA_T^{0n} (auf CLIQUE_G^*) eine größte Clique von G in einer erwarteten (bezüglich des Zufalls der Suchheuristik) Laufzeit von $ne^{\Theta(T+1)}$.

Optimale Parameter und solche, für welche der MA_T^{0n} effizient ist, sind die gleichen wie für Theorem 5.D.9.

Beweis (Theorem 5.D.10). Nach Lemma 5.D.1.b enthält der Graph G mindestens $\Omega(n)$ lose eingebundene 3-Cliques und 4-Cliques mit Wahrscheinlichkeit $1 - 2^{-\Omega(n)}$. Wir nehmen dies an. Die Ergebnisse für die Erwartungswerte folgen direkt.

- a) Betrachten wir den zum MA_T^{0n} auf CLIQUE_G^* äquivalenten irreduzibelen und, da G für $n \geq 5$ nach Lemma 5.D.1.a nicht der vollständige Graph sein kann, auch aperiodischen

Markoffprozess genauer (vergleiche Jerrum, 1992). Sei S der die 0-Clique repräsentierende Zustand. Eine Anwendung des Lemmas 5.D.8.a liefert nach Jerrum (1992) eine erwartete Laufzeit von $1/\pi_S = \sum_{C''} e^{T \cdot |C''|} \leq (16n - 29) \cdot e^{T \cdot 4} = \mathcal{O}(ne^{4T})$, um sich in Zustand S befindend in Zustand S zurückzukehren. Sei \mathcal{E} das Ereignis, sich in Zustand S befindend, eine 4-Clique zu erzeugen, bevor eine Rückkehr in Zustand S erfolgt.

- Wir beschränken die Wahrscheinlichkeit des Eintretens von \mathcal{E} zunächst nach unten. Im zweiten Schritt wird mit Wahrscheinlichkeit $\Omega(n)/n = \Omega(1)$ eine Clique $\{v\} \subseteq V'_4$ generiert. Für die lose eingebundene 4-Clique C gelte $C \supseteq \{v\}$. Mindestens mit Wahrscheinlichkeiten $(3/n)/(3/n + e^{-T}/n)$, $(2/n)/(2/n + 2e^{-T}/n)$ und $(1/n)/(1/n + 3e^{-T}/n)$ wird jeweils eine Teilmenge von C der Größen zwei, drei und vier vor einer 0-, 1- und 2-Clique erzeugt. Es gilt also

$$\Pr[\mathcal{E}] \geq \Omega(1) \cdot \frac{3}{3 + e^{-T}} \cdot \frac{2}{2 + 2e^{-T}} \cdot \frac{1}{1 + 3e^{-T}} = \Omega(1).$$

- Wir beschränken die Wahrscheinlichkeit des Eintretens von \mathcal{E} nun nach oben. Die Summe der Grade aller Knoten in G ist höchstens $2 \cdot (3n - 6)$. Daher existieren mindestens $n/2$ Knoten mit einem Grad von höchstens zwölf. Im zweiten Schritt wird mindestens mit Wahrscheinlichkeit $1/2$ eine solche 1-Clique generiert. Mindestens mit Wahrscheinlichkeit $(e^{-T}/n)/(e^{-T}/n + 12/n)$ wird die 0-Clique vor einer 2-Clique erzeugt. Es gilt also

$$\Pr[\mathcal{E}] \leq 1 - \frac{1}{2} \cdot \frac{1}{1 + 12e^{-T}} = 1 - \Omega(e^{-T}).$$

Unter der Bedingung, dass \mathcal{E} eintritt, wird eine 4-Clique in einer erwarteten Laufzeit von $\mathcal{O}(ne^{4T})/\Omega(1) = \mathcal{O}(ne^{4T})$ erzeugt und unter der Bedingung, dass \mathcal{E} nicht eintritt, wird die 0-Clique in einer erwarteten Laufzeit von $\mathcal{O}(ne^{4T})/\Omega(e^{-T}) = \mathcal{O}(ne^{5T})$ wieder generiert.

Insgesamt resultiert eine erwartete Laufzeit von $\mathcal{O}(1) \cdot \mathcal{O}(ne^{5T}) + \mathcal{O}(ne^{4T}) = ne^{\mathcal{O}(1+T)}$.

- b) Mit Wahrscheinlichkeit $\Omega(n)/n = \Omega(1)$ entsteht im zweiten Schritt eine 1-Clique $\{v\} \subseteq V'_3$ und dann muss notwendigerweise eine von höchstens drei möglichen Verschlechterungen akzeptiert werden, um eine Teilmenge einer größten Clique zu generieren. Insgesamt resultiert eine erwartete Laufzeit von $\Omega(1) \cdot ne^T/3 = ne^{\Omega(1+T)}$.

Das gewünschte Resultat ist damit gezeigt. \square

Im Gegensatz etwa zur $(16n+1)$ -RLS 0n und zur $(1+1)$ -RRLS 0n ist etwa der MA $_1^{0n}$ sowohl auf beliebigen als auch auf zufälligen planaren Graphen eine bisher effizienteste Suchheuristik.

5.D.2 Weitere semizufällige Graphen

Wir erweitern unsere Worst- und Average-Case-Betrachtungen für planare Graphen zu Semi-Average-Case-Betrachtungen, wie sie am Anfang dieses Kapitel motiviert wurden, und sind an optimalen Lösungen für das Cliquenproblem interessiert.

Wir betrachten Suchheuristiken auf den folgenden drei Eingabemodellen für planare semizufällige Graphen. Sei G ein Graph bezüglich $\mathcal{G}_n^{\text{planar}}$ gewählt. In jedem der drei Eingabemodelle darf ein Gegner lediglich planare Graphen erzeugen.

- $\mathcal{G}_n^{\text{planar,+}}$ Einem Gegner ist es dann erlaubt, beliebige Kanten dem Graphen G hinzuzufügen.
- $\mathcal{G}_n^{\text{planar,-}}$ Einem Gegner ist es dann erlaubt, beliebige Kanten dem Graphen G zu entnehmen.
- $\mathcal{G}_{n,\varepsilon}^{\text{planar,*}}$ Einem Gegner ist es dann erlaubt, beliebige, jedoch höchstens εn viele Kanten dem Graphen G hinzuzufügen oder zu entnehmen.

Betrachten wir den unbehandelten Graphen G . Nach Lemma 5.D.1.b enthält der Graph G mindestens $\Omega(n)$ lose eingebundene 3-Cliquen und 4-Cliquen mit Wahrscheinlichkeit $1 - 2^{-\Omega(n)}$. Sei $H_{\ln^{1/2}n}$ der im Beweis des Theorems 5.D.6.b definierte Graph auf $6 \ln^{1/2}n + 4$ Knoten. Mit Wahrscheinlichkeit $1 - n^{-\omega(1)}$ existiert $V' \subseteq V$, so dass V' lose eingebunden in G ist, und der durch V' induzierte Graph ist isomorph, also mit einer Umbenennung der Knoten äquivalent, zu $H_{\ln^{1/2}n}$ nach McDiarmid et al. (2005). Wir nehmen dies an.

- $\mathcal{G}_n^{\text{planar,+}}$ Die Anzahl 4-Cliquen im behandelten Graphen beträgt weiter $\Omega(n)$ und es existieren $\Omega(1)$ Knoten aus V_4 mit Grad $\mathcal{O}(1)$, da höchstens $3n - 6$ Kanten in den unbehandelten Graphen eingefügt wurden.
Eine erwartete Laufzeit von $\Theta(n^5)$ für den (1+1)-EA⁰ⁿ folgt nach dem Beweis des Theorems 5.D.5. Eine erwartete Laufzeit von $\Omega(n \ln n)$ für die (16n+1)-RLS⁰ⁿ folgt nach dem Beweis des Theorems 5.D.3. Eine erwartete Laufzeit von $\Theta(n)$ für die (1+1)-RRLS_n⁰ⁿ und den MA₁⁰ⁿ folgt analog zu den Beweisen des Theorems 5.D.7 und des Theorems 5.D.10.
- $\mathcal{G}_n^{\text{planar,-}}$ Einem Gegner ist es möglich, einen Graphen isomorph zu denen der Beweise des Theorems 5.D.4.b, des Theorems 5.D.2.b und des Theorems 5.D.9.b zu erzeugen, und ebenso ist es ihm möglich, alle Kanten aus dem unbehandelten Graphen zu entfernen, bei denen nicht beide inzidenten Knoten in V' liegen.
Eine erwartete Laufzeit von $\Theta(n^6)$ für den (1+1)-EA⁰ⁿ folgt nach dem Beweis des Theorems 5.D.5. Eine erwartete Laufzeit von $\Theta(n^2)$ für die (16n+1)-RLS⁰ⁿ und den MA₁⁰ⁿ folgt nach den Beweisen des Theorems 5.D.2 und des Theorems 5.D.9. Eine erwartete Laufzeit von $\Omega(n^2 \ln n)$ folgt analog zum Beweis des Theorems 5.D.6.b.
- $\mathcal{G}_{n,\varepsilon}^{\text{planar,*}}$ Für $\varepsilon \geq 6$ ist es einem Gegner möglich, beliebige planare Graphen zu erzeugen. Für eine genügend kleine Konstante $\varepsilon > 0$ enthält der unbehandelte Graph mindestens $3\varepsilon n$ lose eingebundene 3-Cliquen und 4-Cliquen. Der behandelte Graph enthält dann mindestens εn lose eingebundene 3-Cliquen und 4-Cliquen.
Eine erwartete Laufzeit von $\Theta(n^5)$ für den (1+1)-EA⁰ⁿ folgt nach dem Beweis des Theorems 5.D.5. Eine erwartete Laufzeit von $\Omega(n \ln n)$ für die (16n+1)-RLS⁰ⁿ folgt nach dem Beweis des Theorems 5.D.3. Eine erwartete Laufzeit von $\Theta(n)$ für die (1+1)-RRLS_n⁰ⁿ und den MA₁⁰ⁿ folgt analog zu den Beweisen des Theorems 5.D.7 und des Theorems 5.D.10.

Theorem 5.D.11 Sei G ein Graph bezüglich des in der Tabelle genannten Eingabemodells für planare semizufällige Graphen gewählt. Mit Wahrscheinlichkeit $1 - n^{-\omega(1)}$ und im Erwartungswert (bezüglich des Zufalls der Eingabe) erzeugt die in der Tabelle genannte Suchheuristik (auf CLIQUE_G^*) eine größte Clique von G in einer erwarteten (bezüglich des Zufalls der Suchheuristik) Laufzeit von dem in der Tabelle genannten Wert. Sei $\varepsilon > 0$ klein genug.

	$(1+1)\text{-EA}^{0^n}$	$(16n+1)\text{-RLS}^{0^n}$	$(1+1)\text{-RRLS}_n^{0^n}$	$\text{MA}_1^{0^n}$
$\mathcal{G}_n^{\text{planar},+}$	$\Theta(n^5)$	$\Omega(n \ln n)$	$\Theta(n)$	$\Theta(n)$
$\mathcal{G}_n^{\text{planar},-}$	$\Theta(n^6)$	$\Theta(n^2)$	$\Omega(n^2 \ln n)$	$\Theta(n^2)$
$\mathcal{G}_{n,\varepsilon}^{\text{planar},*}$	$\Theta(n^5)$	$\Omega(n \ln n)$	$\Theta(n)$	$\Theta(n)$

Abermals ist etwa der $\text{MA}_1^{0^n}$ im Gegensatz etwa zur $(16n+1)\text{-RLS}^{0^n}$ und zur $(1+1)\text{-RRLS}_n^{0^n}$ auf allen planaren semizufälligen Graphen eine bisher effizienteste Suchheuristik.

5.E Black-Box-Algorithmen und -Komplexität

Wir haben bewiesen, dass der $\text{MA}_1^{0^n}$ nicht nur für alle betrachteten planaren semizufälligen Graphen, sondern mit einer erwarteten Laufzeit von $\Theta(n^2)$ für beliebige und $\Theta(n)$ für zufällige planare Graphen (asymptotisch) eine effizienteste Suchheuristik für das Berechnen einer größten Clique ist. Wir werden nun zeigen, dass es (asymptotisch) keine effizienteren Suchheuristiken (auf CLIQUE_G^*) geben kann, um eine größte Clique in einem beliebigen oder zufälligen Graphen zu berechnen. Wir haben in Abschnitt 1.A bereits Betrachtungen zu Black-Box-Algorithmen und -Komplexität vorgenommen. Im Weiteren werden wir die Black-Box-Komplexität und die durchschnittliche Black-Box-Komplexität für das Cliquesproblem (mit CLIQUE_G und CLIQUE_G^*) genauer analysiert.

Für die Black-Box-Komplexität ist es unerheblich, ob die Funktion CLIQUE_G oder CLIQUE_G^* betrachtet wird, da aus dem Eintrag der Historie $(x, \text{CLIQUE}_G(x))$ effizient $(x, \text{CLIQUE}_G^*(x))$ berechnet werden kann, und umgekehrt. Daher genügt die Untersuchung von CLIQUE_G^* .

Sei $\mathcal{A}_n^{\text{all}}$ die Menge aller Graphen auf n Knoten, $\mathcal{A}_n^{\text{planar}}$ die Menge aller planaren Graphen auf n Knoten und $\mathcal{A}_n^{\text{one edge}}$ die Menge der $\binom{n}{2}$ vielen planaren Graphen auf n Knoten, die genau eine Kante beinhalten. Es gilt $\mathcal{A}_n^{\text{all}} \supseteq \mathcal{A}_n^{\text{planar}} \supseteq \mathcal{A}_n^{\text{one edge}}$. Dann sei weiter $\text{CLIQUE}_{\mathcal{A}}^*$ die Klasse der Funktionen CLIQUE_G^* mit $G \in \mathcal{A}$.

Betrachten wir den folgenden deterministischen Black-Box-Algorithmus A und seine Variante A^* zur Berechnung einer größten Clique.

A/A^*

1. Setze $A_1 := \emptyset, \dots, A_n := \emptyset$ und $i := 1$.
2. Für $j := i + 1, \dots, n$: Bestimme Funktionswert von $\{v_i, v_j\}$. Ist dieser 2, setze $A_i := A_i \cup \{v_j\}, A_j := A_j \cup \{v_i\}$.
- ★. *nur A^** : Falls $|A_i| = 3$, bestimme Funktionswert von $A_i \cup \{v_i\}$.
3. Setze $i := i + 1$. Falls $i < n$, gehe zu 2. und sonst gehe zu 4.
4. Berechne eine größte Clique C (für beliebige planare Graphen effizient mit Algorithmus von Eppstein (1999); für beliebige Graphen mit Algorithmus von Robson (2001)) und bestimme Funktionswert von C .

Wir bemerken, dass bei der Bearbeitung von Anweisung 4 die Menge A_i , $1 \leq i \leq n$, genau die zu v_i adjazenten Knoten enthält, was die Berechnung einer größten Clique ohne Funktionsauswertungen ermöglicht.

Wir betrachten zunächst beliebige (planare) Graphen.

Theorem 5.E.1

- a) Sei G ein beliebiger (planarer) Graph auf n Knoten. Der Black-Box-Algorithmus A erzeugt eine größte Clique von G in einer Laufzeit von höchstens $\binom{n}{2} + 1 = \mathcal{O}(n^2)$.
- b) Die Black-Box-Komplexität von $\text{CLIQUE}_{\mathcal{A}_n^{\text{one edge}}}^*$ beträgt $(\binom{n}{2} + 1)/2 = \Theta(n^2)$.

Es folgen eine Black-Box-Komplexität von höchstens $\binom{n}{2} + 1 = \mathcal{O}(n^2)$ und von mindestens $(\binom{n}{2} + 1)/2 = \Omega(n^2)$ also von $\Theta(n^2)$ direkt für $\text{CLIQUE}_{\mathcal{A}_n^{\text{planar}}}^*$ und $\text{CLIQUE}_{\mathcal{A}_n^{\text{all}}}^*$.

Beweis (Theorem 5.E.1).

- a) Der Black-Box-Algorithmus A berechnet spätestens in Anweisung 4 eine größte Clique und führt insgesamt $\binom{n}{2}$ Funktionsauswertungen in Anweisung 2 und eine Funktionsauswertung in Anweisung 4 durch.
- b) Die obere Schranke folgt direkt nach Aussage a). Betrachten wir die untere Schranke. Der Funktionswert jedes Elements x beträgt $-\infty$, falls $|x| \geq 3$ ist, und $|x|$, falls $|x| \leq 1$ ist. Es genügt also Black-Box-Algorithmen zu betrachten, die nur Funktionswerte von Elementen x mit $|x| = 2$ bestimmen. Dann ist $\text{CLIQUE}_{\mathcal{A}_n^{\text{one edge}}}^*$ jedoch äquivalent zu NEEDLE mit dem Suchbereich $\mathcal{A} = \{x \mid x \in \{0, 1\}^n, |x| = 2\}$ und dem Wertebereich $\mathcal{B} = \{-\infty, 2\}$. Das Ergebnis folgt nach Theorem 1.A.1.

Das gewünschte Resultat ist somit gezeigt. □

Wir betrachten nun zufällige planare Graphen.

Theorem 5.E.2

- a) Sei G ein Graph bezüglich $\mathcal{G}_n^{\text{planar}}$ gewählt. Der Black-Box-Algorithmus A^* erzeugt eine größte Clique von G in einer erwarteten (bezüglich des Zufalls der Eingabe) Laufzeit von $\mathcal{O}(n)$.
- b) Die durchschnittliche Black-Box-Komplexität von $\text{CLIQUE}_{\mathcal{A}_n^{\text{planar}}}^*$ beträgt $\Theta(n)$.

Beweis. Nach Lemma 5.D.1.b enthält der Graph G mindestens εn lose eingebundene 3-Cliquen und 4-Cliquen mit Wahrscheinlichkeit $1 - 2^{-\Omega(n)}$ für eine genügend kleine Konstante $0 < \varepsilon \leq 1/2$. Wir nehmen dies zunächst an.

- a) Betrachten wir die in Abschnitt 5.D definierte Knotenmenge V'_4 . Für jede Permutation π auf $\{1, \dots, n\}$ gilt $\{v_{i_1}, \dots, v_{i_{3\varepsilon n}}\} \subseteq V'_4$ und $\{v_{\pi_{i_1}}, \dots, v_{\pi_{i_{3\varepsilon n}}}\} \subseteq V'_4$ für den gleichen Anteil aller planaren Graphen. Falls $v_i \in V'_4$ ist und er damit Grad drei besitzt und in einer 4-Clique liegt, wird spätestens im i -ten Schritt von A^* eine 4-Clique generiert. Nach dem i -ten Schritt mit $i < \varepsilon n$ ist höchstens für einen Anteil von

$$\binom{n-i}{3\varepsilon n} / \binom{n}{3\varepsilon n} = \frac{(n-3\varepsilon n) \cdots (n-3\varepsilon n - i + 1)}{n \cdots (n-i+1)} \leq \left(\frac{n-3\varepsilon n}{n-\varepsilon n}\right)^i \leq (1-\varepsilon)^i$$

aller planaren Graphen keine größte Clique in Anweisung \star erzeugt worden. Nach dem $(\varepsilon n - 1)$ -ten Schritt ist also höchstens für einen Anteil von $(1-\varepsilon)^{\varepsilon n - 1} \leq e^{-\varepsilon^2 n - 1}$ aller planaren Graphen keine größte Clique erzeugt worden. Dann jedoch genügen höchstens $\binom{n}{2} + (n-1) + 1 \leq n^2$ Funktionsauswertungen zur Erzeugung einer größten Clique. Insgesamt resultiert mit der Konvergenzeigenschaft unendlicher geometrischer Reihen und da $\varepsilon \leq 1/2$ ist, eine erwartete Laufzeit von höchstens $\sum_{i=0}^{\varepsilon n - 1} ((1-\varepsilon)^i \cdot (n-i)) + (2^{-\Omega(n)} + e^{-\varepsilon^2 n - 1}) \cdot n^2 \leq n/\varepsilon + 1 = \mathcal{O}(n)$.

- b) Die obere Schranke folgt direkt nach Aussage a). Betrachten wir die untere Schranke. Wir werden im Folgenden zeigen, dass jeder deterministische Black-Box-Algorithmus für mindestens die Hälfte aller planaren Graphen in einer Laufzeit von $n/13$ keine größte Clique erzeugt. Da jeder randomisierte Black-Box-Algorithmus als Wahrscheinlichkeitsverteilung über deterministische Black-Box-Algorithmen angesehen und diese Wahl vor der ersten Funktionsauswertung getroffen werden kann, folgt eine durchschnittliche erwartete Laufzeit von mindestens $(1/2) \cdot n/13 = \Omega(n)$ und damit das Ergebnis.

Sei $x_1, \dots, x_{n/13}$ die beliebige Folge von ersten Elementen, die der deterministische Black-Box-Algorithmus erzeugt, falls Funktionswerte $-\infty$ für $|x_i| \geq 2$ und $|x_i|$ für $|x_i| \leq 1$ mit $1 \leq i \leq n/13$ auftreten. Höchstens für den leeren Graphen, der einen von mindestens $2^{\Omega(n)}$ planarer Graphen darstellt, ist dann eine größte Clique erzeugt worden. Wir werden im Folgenden zeigen, dass für $6/13$ aller planaren Graphen diese Historie auftritt. Offensichtlich gilt dies für $|x_i| \leq 1$ und $|x_i| \geq 5$ und alle planarer Graphen. Betrachten wir nun ein Element x_i mit $2 \leq |x_i| \leq 4$. Für jede Permutation π auf $\{1, \dots, n\}$ und für alle $\ell \geq 0$ entspricht $\{v_{i_1}, \dots, v_{i_\ell}\}$ und $\{v_{\pi_{i_1}}, \dots, v_{\pi_{i_\ell}}\}$ einer ℓ -Clique für den gleichen Anteil aller planaren Graphen. Nach Lemma 5.D.1.a beträgt der Anteil aller planaren Graphen mit der Eigenschaft,

- $\{v_1, v_2\}$ ist eine Clique, höchstens $(3n-6)/\binom{n}{2} \leq 6/n$,
- $\{v_1, v_2, v_3\}$ ist eine Clique, höchstens $(6n-12)/\binom{n}{3} \leq 37/n^2 \leq 6/n$ und
- $\{v_1, v_2, v_3, v_4\}$ ist eine Clique, höchstens $(6n-12)/\binom{n}{4} \leq 145/n^3 \leq 6/n$.

Der Anteil aller planaren Graphen, bei dem entweder die größte Clique kleiner als zwei ist oder eine Clique der Größe mindestens zwei in den ersten $n/13$ Funktionsauswertungen erzeugt wurde, beträgt höchstens $2^{-\Omega(n)} + (n/13) \cdot 6/n \leq 1/2$.

Das gewünschte Resultat ist somit gezeigt. \square

6 AUSBLICK

Wir haben eine Reihe von Aspekten randomisierter Suchheuristiken detailliert analysiert. Für die Nachkommenpopulation evolutionärer Algorithmen konnten wir die wechselseitigen Effekte der Wahl der Populationsgröße und des Selektionsoperators illustrieren. Für die Elternpopulation evolutionärer Algorithmen wurde die Sensibilität der Wahl der Populationsgröße mittels eines Hierarchieresultats verdeutlicht, ebenso wie die Effekte von Methoden zur Erhaltung der Diversität. In diesem Rahmen haben wir bestehende Beweistechniken erweitert. Die Betrachtungen von Transformationen der Zielfunktionen führten zur Charakterisierung randomisierter Suchheuristiken und ihrer Komponenten. Die erschöpfenden Betrachtungen zum Cliquesproblem zeigten die wesentliche Steigerung der Effizienz einer einfachen randomisierten Suchheuristik durch den Einsatz einer (großen) Elternpopulation in einer – mit den semizufälligen Graphen – sehr allgemeinen Form. Für (zufällige) planare Graphen konnte mit Analysen der (durchschnittlichen) Black-Box-Komplexität die (asymptotische) Optimalität einer speziellen einfachen randomisierten Suchheuristik zur Berechnung einer größten Clique bewiesen werden.

Für die randomisierten Suchheuristiken und die Black-Box-Komplexität wird eine Erweiterung bestehender und Entwicklung neuer Beweistechniken von Interesse sein.

Bei den randomisierten Suchheuristiken wird des Weiteren ein Augenmerk auf Ansätzen zur Illustrierung ihres Erfolgs in Anwendungen liegen – etwa durch die Betrachtung von semizufälligen Eingaben.

Für die Black-Box-Komplexität bedarf es weiterer Analysen bei Beschränkung der Ressourcen der betrachteten Black-Box-Algorithmen – etwa des Speicherplatzes, wie es bei Anwendungen randomisierter Suchheuristiken der Fall ist.

7 NOTATIONEN, GLEICHUNGEN UND UNGLEICHUNGEN

7.A Notationen

Wir bezeichnen mit \mathbb{N} die Menge der natürlichen Zahlen $\{0, 1, 2, \dots\}$ und mit \mathbb{R} die Menge der reellen Zahlen. Mit \mathbb{Z} bezeichnen wir die Menge ganzer Zahlen $\{\dots, -2, -1, 0, 1, 2, \dots\}$, wobei \mathbb{Z}^+ die Menge positiver ganzer Zahlen bezeichnet. Wir bezeichnen mit $\ln(n)$ den Logarithmus von n zur Basis e , wobei e die Eulersche Zahl ist, und mit $\log_2(n)$ den Logarithmus von n zur Basis 2. Ein Polynom in n wird mit $\text{poly}(n)$ bezeichnet.

Wir bezeichnen mit $H[x, y]$ den Hammingabstand von x und y für $x, y \in \{0, 1\}^n$ und mit $|x|$ die Anzahl Einsen in x . Mit 0^n und 1^n bezeichnen wir die nur aus Nullen und Einsen bestehenden Elemente aus $\{0, 1\}^n$.

Wir bezeichnen die Wahrscheinlichkeit eines Ereignisses \mathcal{E} mit $\Pr[\mathcal{E}]$. Mit $E[X]$ und $\text{Var}[X]$ bezeichnen wir den Erwartungswert und die Varianz einer Zufallsvariablen X .

Definition (\mathcal{O} -Notation) Seien $f, g : \mathbb{N} \rightarrow \mathbb{R}_0^+$ zwei Funktionen.

- $f(n) = \mathcal{O}(g(n))$, falls $c, n_0 \in \mathbb{R}^+$ existieren, so dass $f(n) \leq c \cdot g(n)$ für alle $n \geq n_0$ gilt.
- $f(n) = \Omega(g(n))$, falls $g(n) = \mathcal{O}(f(n))$ gilt.
- $f(n) = \Theta(g(n))$, falls $f(n) = \mathcal{O}(g(n))$ und $g(n) = \mathcal{O}(f(n))$ gilt.
- $f(n) = o(g(n))$, falls $g(n) \neq 0$ und $\lim_{n \rightarrow \infty} f(n)/g(n) = 0$ gilt.
- $f(n) = \omega(g(n))$, falls $g(n) = o(f(n))$ gilt.

Wir verweisen auf Mitzenmacher und Upfal (2005), Motwani und Raghavan (1995) und Wegener (2005) für eine ausführlichere Darstellung der verwendeten Notationen.

7.B Gleichungen und Ungleichungen

Lemma 7.B.1 Für $t \in \mathbb{R}$ gilt

$$e^t \geq 1 + t.$$

Lemma 7.B.2 Für $1 \leq k \leq n$ gilt

$$\frac{k}{2n} \leq 1 - \left(1 - \frac{1}{n}\right)^k \leq \frac{k}{n}.$$

Lemma 7.B.3 Für $0 \leq k \leq n$ gilt

$$\left(\frac{n}{k}\right)^k \leq \binom{n}{k} \leq \frac{n^k}{k!} \leq \left(\frac{en}{k}\right)^k.$$

Lemma 7.B.4 [unendliche geometrische Reihe] Für $|x| < 1$ gilt

$$\sum_{n=0}^{\infty} x^n = \frac{1}{1-x}.$$

Lemma 7.B.5 [Harmonische Zahl] Für $n \geq 1$ gilt

$$\ln n \leq \sum_{i=1}^n \frac{1}{i} \leq \ln n + 1.$$

Lemma 7.B.6 [Stirlingungleichung] Für n groß genug gilt

$$2\sqrt{n} \cdot \left(\frac{n}{e}\right)^n \leq n! \leq 3\sqrt{n} \cdot \left(\frac{n}{e}\right)^n.$$

Lemma 7.B.7 [Markoffungleichung] Sei X eine Zufallsvariable, die nur nicht negative Werte annimmt. Dann gilt für alle $t \in \mathbb{R}^+$

$$\Pr[X \geq t] \leq \frac{E[X]}{t}.$$

Lemma 7.B.8 [Tschebyscheffungleichung] Sei X eine Zufallsvariable. Dann gilt für alle $t \in \mathbb{R}^+$

$$\Pr[|X - E[X]| \geq t] \leq \frac{\text{Var}[X]}{t^2}.$$

Lemma 7.B.9 [Chernoffungleichung] Seien X_1, \dots, X_n unabhängige Bernoulliversuche sowie $0 < \Pr[X_i = 1] < 1$ mit $1 \leq i \leq n$ und $X := X_1 + \dots + X_n$. Dann gilt für alle $\delta \in \mathbb{R}^+$

$$\Pr[X \geq (1 + \delta)E[X]] \leq \left[\frac{e^\delta}{(1 + \delta)^{1+\delta}} \right]^{E[X]}.$$

Wir verweisen auf Mitzenmacher und Upfal (2005), Motwani und Raghavan (1995) und Wegener (2005) für eine ausführlichere Darstellung der verwendeten Gleichungen und Ungleichungen.

LITERATURVERZEICHNIS

- Aarts, E. und Lenstra, J.K., Hg. (2003): *Local Search in Combinatorial Optimization*. Princeton University Press.
- Alon, N., Krivelevich, M. und Sudakov, B. (1998): Finding a large hidden clique in a random graph. *Random Structures and Algorithms*, 13(3–4):457–466.
- Bäck, T., Fogel, D.B. und Michalewicz, Z., Hg. (1997): *Handbook of Evolutionary Computation*. Institute of Physics Publishing and Oxford University Press.
- Beyer, H.G., Schwefel, H.P. und Wegener, I. (2002): How to analyse evolutionary algorithms. *Theoretical Computer Science*, 287(1):101–130.
- Blum, A. und Spencer, J. (1995): Coloring random and semi-random k -colorable graphs. *Journal of Algorithms*, 19(2):204–234.
- Bollobás, B. (1985): *Random Graphs*. Academic Press.
- Bomze, I.M., Budinich, M., Pardalos, P.M. und Pelillo, M. (1999): The maximum clique problem. In: D.Z. Du et al., Hg., *Handbook of Combinatorial Optimization*, S. 1–74. Kluwer.
- Chandra, A.K., Raghavan, P., Ruzzo, W.L., Smolensky, R. und Tiwari, P. (1989): The electrical resistance of a graph captures its commute and cover times. In: D.S. Johnson, Hg., *Proceedings of the 21st Annual Symposium on Theory of Computing – STOC 1989*, S. 574–586. ACM Press.
- Droste, S. (2006): A rigorous analysis of the compact genetic algorithm for linear functions. *Natural Computing*, 5(3):257–283.
- Droste, S., Jansen, T., Rudolph, G., Schwefel, H.P., Tinnefeld, K. und Wegener, I. (2003): Theory of evolutionary algorithms and genetic programming. In: H.P. Schwefel et al., Hg., *Advances in Computational Intelligence*, S. 107–144. Springer.
- Droste, S., Jansen, T. und Wegener, I. (2002): On the analysis of the (1+1) evolutionary algorithm. *Theoretical Computer Science*, 276(1–2):51–81.
- Droste, S., Jansen, T. und Wegener, I. (2006): Upper and lower bounds for randomized search heuristics in black-box optimization. *Theory of Computing Systems*, 39(4):525–544.

- Eppstein, D. (1999): Subgraph isomorphism in planar graphs and related problems. *Journal of Graph Algorithms and Applications*, 3(3):1–27.
- Feige, U. (2004): Approximating maximum clique by removing subgraphs. *SIAM Journal on Discrete Mathematics*, 18(2):219–225.
- Feige, U. und Kilian, J. (2001): Heuristics for semi-random graph problems. *Journal of Computer and System Science*, 63(4):639–671.
- Feige, U. und Krauthgamer, R. (2000): Finding and certifying a large hidden clique in a semi-random graph. *Random Structures and Algorithms*, 16(2):195–208.
- Frieze, A.M. und McDiarmid, C. (1997): Algorithmic theory of random graphs. *Random Structures and Algorithms*, 10(1–2):5–42.
- Garnier, J., Kallel, L. und Schoenauer, M. (1999): Rigorous hitting times for binary mutations. *Evolutionary Computation*, 7(2):173–203.
- Giel, O. und Wegener, I. (2004): Maximum cardinality matchings on trees by randomized local search. In: *Electronic Colloquium on Computational Complexity*, Bd. 076.
- He, J. und Yao, X. (2001): Drift analysis and average time complexity of evolutionary algorithms. *Artificial Intelligence*, 127(1):57–85.
- Holland, J.H. (1975): *Adaptation in Natural and Artificial System*. University of Michigan Press.
- Horn, J., Goldberg, D.E. und Deb, K. (1994): Long path problems. In: Y. Davidor et al., Hg., *Proceedings of the International Conference on Parallel Problem Solving from Nature III – PPSN 1994*, Bd. 866 von *Lecture Notes in Computer Science*, S. 149–158. Springer.
- Jansen, T. (2002): On the analysis of dynamic restart strategies for evolutionary algorithms. In: J.J. Merelo Guervós et al., Hg., *Proceedings of the International Conference on Parallel Problem Solving From Nature VII – PPSN 2002*, Bd. 2439 von *Lecture Notes in Computer Science*, S. 33–43. Springer.
- Jansen, T., De Jong, K. und Wegener, I. (2005): On the choice of the offspring population size in evolutionary algorithms. *Evolutionary Computation*, 13(4):413–440.
- Jansen, T. und Wegener, I. (2000): On the choice of the mutation probability for the (1+1) EA. In: M. Schoenauer et al., Hg., *Proceedings of the International Conference on Parallel Problem Solving From Nature VI – PPSN 2000*, Bd. 1917 von *Lecture Notes in Computer Science*, S. 89–98. Springer.
- Jansen, T. und Wegener, I. (2001a): Evolutionary algorithms – how to cope with plateaus of constant fitness and when to reject strings with the same fitness. *IEEE Transactions on Evolutionary Computation*, 5(6):589–599.

- Jansen, T. und Wegener, I. (2001b): On the utility of populations. In: L. Spector et al., Hg., *Proceedings of the Genetic and Evolutionary Computation Conference – GECCO 2001*, S. 375–382. Morgan Kaufmann.
- Jansen, T. und Wegener, I. (2005): Real royal road functions – where crossover provably is essential. *Discrete Applied Mathematics*, 149(1–3):111–125.
- Jerrum, M. (1992): Large cliques elude the Metropolis process. *Random Structures and Algorithms*, 3(4):347–360.
- Johnson, D.S. und Trick, M.A. (1996): *Cliques, Coloring, and Satisfiability*. American Mathematical Society.
- Karp, R.M. (1972): Reducibility among combinatorial problems. In: R. Miller et al., Hg., *Complexity of Computer Computations*, S. 85–103. Plenum Press.
- Khot, S. und Ponnuswami, A.K. (2006): Better inapproximability results for maxclique, chromatic number and min-3lin-deletion. In: M. Bugliesi et al., Hg., *Proceedings of the 33th International Colloquium on Automata, Languages and Programming – ICALP 2006*, Bd. 4051 von *Lecture Notes in Computer Science*, S. 226–237. Springer.
- Knuth, D.E. (2005): *Generating All Tuples and Permutations*, Bd. 4, Fasciculus 2 von *The Art of Computer Programming*. Addison Wesley.
- Kuratowski, K. (1930): Sur le problème des courbes gauches en topologie. *Fundamenta mathematicae*, 15:271–283.
- Legendre, A.M. (1794): *Eléments de géométrie*. Didot.
- McDiarmid, C., Steger, A. und Welsh, D.J.A. (2005): Random planar graphs. *Journal of Combinatorial Theory, Series B*, 93(2):187–205.
- Mühlenbein, H. (1992): How genetic algorithms really work: Mutation and hillclimbing. In: R. Männer und B. Manderick, Hg., *Proceedings of the International Conference on Parallel Problem Solving from Nature II – PPSN 1992*, *Lecture Notes in Computer Science*, S. 15–26. Elsevier.
- Mitzenmacher, M. und Upfal, E. (2005): *Probability and Computing*. Cambridge University Press.
- Motwani, R. und Raghavan, P. (1995): *Randomized Algorithms*. Cambridge University Press.
- Neumann, F. und Wegener, I. (2004): Randomized local search, evolutionary algorithms, and the minimum spanning tree problem. In: K. Deb et al., Hg., *Proceedings of the Genetic and Evolutionary Computation Conference – GECCO 2004*, Bd. 3102 von *Lecture Notes in Computer Science*, S. 713–724. Springer.

- Neumann, F. und Witt, C. (2006): Runtime analysis of a simple ant colony optimization algorithm (extended abstract). In: *Proceedings of the 17th International Symposium on Algorithms and Computation – ISAAC 2006*, Bd. 4288 von *Lecture Notes in Computer Science*, S. 618–627. Springer.
- Robson, J.M. (2001): Finding a maximum independent set in time $\mathcal{O}(2^{n/4})$. In: *Technical Report 1251-01, Université Bordeaux, LaBRI*.
- Rudolph, G. (1997): How mutation and selection solve long-path problems in polynomial expected time. *Evolutionary Computation*, 4(2):195–205.
- Schwefel, H.P. (1995): *Evolution and Optimum Seeking*. Wiley.
- Storch, T. und Wegener, I. (2004): Real royal road functions for constant population size. *Theoretical Computer Science*, 320(1):123–134.
- Sudholt, D. (2006): On the analysis of the (1+1) memetic algorithm. In: M. Keijzer et al., Hg., *Proceedings of the Genetic and Evolutionary Computation Conference – GECCO 2006*, Bd. 1, S. 493–500. ACM Press.
- Wegener, I. (2001): Theoretical aspects of evolutionary algorithms. In: F. Orejas et al., Hg., *Proceedings of the 28th International Colloquium on Automata, Languages and Programming – ICALP 2001*, Bd. 2076 von *Lecture Notes in Computer Science*, S. 64–78. Springer.
- Wegener, I. (2002): Methods for the analysis of evolutionary algorithms on pseudo-boolean functions. In: R. Sarker et al., Hg., *Evolutionary Optimization*, S. 349–369. Kluwer.
- Wegener, I. (2005): *Complexity Theory*. Springer.
- Wegener, I. und Witt, C. (2005): On the optimization of monotone polynomials by simple randomized search heuristics. *Combinatorics, Probability and Computing*, 14(1):225–247.
- Witt, C. (2003): Population size vs. runtime of a simple EA. In: R. Sarker et al., Hg., *Proceedings of the Congress on Evolutionary Computation – CEC 2003*, Bd. 3, S. 1996–2003. IEEE Press.
- Witt, C. (2005): Worst-case and average-case approximations by simple randomized search heuristics. In: V. Diekert und B. Durand, Hg., *Proceedings of the 22nd Symposium on Theoretical Aspects of Computer Science – STACS 2005*, Bd. 3403 von *Lecture Notes in Computer Science*, S. 44–46. Springer.
- Witt, C. (2006): Runtime analysis of the $(\mu+1)$ EA on simple pseudo-boolean functions. *Evolutionary Computation*, 14(1):65–86.
- Wolpert, D.H. und Macready, W.G. (1997): No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82.