



Evolutionäre Algorithmen

Übungsblatt 11, SS2011

Abgabe: 12.07.2011

Aufgabe 27 Schnittrestminimierung - (20 Punkte)

Eine Firma stanz kreisförmige Deckel aus rechteckigen Rohblechen und veredelt sie. Um besonders günstig zu produzieren, sollen möglichst wenig Schnittreste übrig bleiben. Sie wurden als Praktikant beauftragt, die Stanzmuster evolutionär zu optimieren. Die Aufgabe ist wie folgt beschrieben:

Gegeben sei eine rechteckige Stanzfläche der Maße $b \cdot h$ und eine feste Anzahl k an Kreisen mit Radien $r \in [r_{\min}, r_{\max}]$. Gesucht ist eine Liste von k Zentren (x_i, y_i) und Radien (r_i) , die automatisiert ausgeschnitten werden. Ein Individuum $I \in \mathbb{R}^{3k}$ sei also der Form $I = (x_1, y_1, r_1, x_2, y_2, r_2, \dots, x_k, y_k, r_k)$. Die zu maximierende Fitness der Liste entspricht der Gesamtfläche der resultierenden gültigen Schnittkreise, die also mit $r_i \in [r_{\min}, r_{\max}]$ vollständig innerhalb des Rechtecks liegen und sich nicht mit Scheiben niedrigeren Indexes überschneiden. Es ist klar, dass die Fitness durch $A = b \cdot h$ beschränkt ist, wodurch das analoge Minimierungsproblem gestellt werden kann. Es soll nun das Schnittreste-Problem in EvA2 implementiert werden. Laden Sie sich dazu das neue Framework von der Übungswebseite¹ herunter. Darin finden Sie bereits die benötigte Paketstruktur und Klassengerüste. Hierbei definiert `DiscCuttingProblem` das Optimierungsproblem. `DiscProblemView` dient der Visualisierung während der Optimierung.

- (a) Implementieren Sie das Schnittreste-Problem in der Klasse `DiscCuttingProblem` als reellwertiges Optimierungsproblem mit folgenden Rahmendaten: $h = b = 200$, $r_{\min} = 10$, $r_{\max} = 30$, $k = 20$. Implementieren Sie die mit dem Schlagwort `TODO` gekennzeichneten Methoden. Führen Sie dazu, falls nötig, Feldvariablen ein. (7 Punkte)
- (b) Vergleichen Sie mindestens drei Ihnen bekannte Varianten einer (15,50)-ES mit verschiedenen Mutationsoperatoren sowie *Differential Evolution* und *Simulated Annealing* in Standard-einstellung. Welche der Optimierer ist am erfolgreichsten? Liefern Sie einen beschrifteten Screenshot der Fitness-Plots und geben Sie an, inwiefern ihr Ergebnis aussagekräftig oder statistisch belegt ist. (4 Punkte)
- (c) Versuchen Sie, zu erklären, warum das formulierte Problem für die Ihnen bekannten Standard-ES-Operatoren (selbstadaptive Mutationsverfahren, Rekombination) schwierig ist. (3 Punkte)
- (d) Sowohl bei der Initialisierung als auch im Lauf der Optimierung treten häufig ungültige Schnittparameter auf, z. B. weil Scheiben über den Rand ragen oder sich überlagern. Wie könnten Fitnessfunktion bzw. Initialisierung modifiziert werden, um weniger ungültige Scheiben in den Individuen zu erhalten? Könnte die Modifizierung unerwünschte Nebeneffekte bzgl. der gefundenen lokalen Optima nach sich ziehen (Erklärung)? (3 Punkte)

¹http://www.cogsys.cs.uni-tuebingen.de/lehre/ss11/ga_es_ueb.html

- (e) Beschreiben Sie einen problemspezifischen Mutationsoperator, der mit höherer Wahrscheinlichkeit als die Standard-ES-Operatoren aus ungültigen Schnitten gültige macht. (1 Punkt)
- (f) Es ist leicht zu sehen, dass Schnittmuster der beschriebenen Form häufig dann gut sind, wenn sich ausgeschnittene Deckel genau berühren. Beschreiben Sie eine Möglichkeit, dieses zusätzliche Wissen im Optimierungsverfahren zu berücksichtigen. (2 Punkte)

Allgemeiner Hinweis: Abgabe per E-Mail (andreas.jahn@uni-tuebingen.de) ist erwünscht, besonders für die Programmieraufgaben. Bitte die Klassen vorher testen, da Syntaxfehler zu Punktabzug führen. Die Klassen sollten ohne irgendwelche Extras mit dem JDK 1.6 laufen. Bitte grundsätzlich keine grafischen Oberflächen programmieren, denn dies kostet meist deutlich mehr Zeit als geplant. Wer es dennoch für unerlässlich hält, möge ausschließlich das AWT und Swing verwenden.