

Moderne Methoden zur experimentellen Analyse evolutionärer Verfahren

Thomas Bartz-Beielstein und Mike Preuß

Universität Dortmund

Lehrstuhl Algorithm Engineering

Arbeitsgruppe Computational Intelligence

44221 Dortmund

Tel.: (0231) 755-7705

Fax: (0231) 755-7740

E-Mail: {thomas.bartz-beielstein@udo.edu, mike.preuss@cs.uni-dortmund.de}

Zusammenfassung

Die Beschreibung einer Methodik zur Analyse der Industrietauglichkeit evolutionärer Algorithmen steht im Mittelpunkt dieses Beitrags. Dazu werden moderne statistische Verfahren wie “Design and Analysis of Computer Experiments” mit klassischen Verfahren der experimentellen Versuchsplanung (Design of Experiments) kombiniert. Diese Methoden eignen sich sehr gut, um den experimentellen Vergleich verschiedener Algorithmen zu objektivieren und praxisnah durchzuführen. Sie wurden bereits für unterschiedliche Simulations- und Optimierungsszenarien eingesetzt, z.B. für die Optimierung von Transportproblemen in Gebäuden (Fahrstuhlsteuerungsprobleme). Aus diesen Szenarien lassen sich allgemeine und zugleich praxisrelevante Tests definieren, die die Grundlage für den Vergleich von CI-Methoden (Fuzzy-Systeme, neuronale Netze und evolutionäre Algorithmen) mit klassischen Verfahren der Optimierung bilden können.

Des Weiteren stellen wir eine frei verfügbare Sammlung von Programmen zur sequenziellen Analyse von Optimieralgorithmen zur Verfügung (SPOT: Sequential Parameter Optimization Toolbox), die für die oben genannten Zwecke entwickelt wird. Sie beinhaltet klar definierte Schnittstellen, so dass gängige Optimierungstools mit geringem Aufwand eingebunden und analysiert werden können. Die Analysemöglichkeiten gehen über klassische Vergleiche (Bericht von Mittelwerten, Standardabweichungen, etc.) weit hinaus und ermöglichen ein tiefer gehendes Verständnis der Funktionsweise der Algorithmen.

1 Einleitung

Naturalanaloge Optimierverfahren wie genetische Algorithmen (GA), Evolutionsstrategien (ES), Particle Swarm Optimization (PSO) und Differential Evolution (DE) haben in den letzten Jahren an Popularität gewonnen. Der Einfachheit halber fassen wir diese Verfahren als *Heuristiken* zusammen, wohingegen die mathematischen Algorithmen als *exakte Verfahren* referenziert werden. Heuristiken haben die akademischen Spielwiesen verlassen und werden mehr und mehr zur Lösung praktischer Problemstellungen eingesetzt. Es existiert ein ganzer Zoo unterschiedlicher Heuristiken und beinahe täglich werden neue Verfahren veröffentlicht. Jede dieser Heuristiken erfordert die Einstellung so genannter exogener Parameter wie der Populationsgröße oder der Rekombinationswahrscheinlichkeit bei ES/GA, die den Erfolg der Verfahren wesentlich beeinflussen. Zusätzlich stehen diese Verfahren in Konkurrenz zu den klassischen, exakten Verfahren der Optimierung.

Anwender, die auch nur halbwegs objektiv einen Vergleich durchführen möchten, sind durch die Komplexität dieses Unterfangens überfordert. So werden erfolgreiche Anwendungen als Einzelfälle in der Literatur beschrieben, diese sind zumeist stark problemspezifisch und nicht ohne Weiteres auf andere Probleminstanzen oder gar Problemklassen übertragbar.

An dieser Stelle kommt die sequentielle Parameteroptimierung (SPO) zum Einsatz. Mittels statistischer Verfahren werden gute Parametereinstellungen für die Verfahren bestimmt, so dass ein objektiver Vergleich ermöglicht wird. Bartz-Beielstein et al. haben in einer Reihe von Veröffentlichungen den entscheidenden Einfluß exogener Parameter auf die Güte (hier verstanden als Performance) der Heuristiken wie auch der exakten Verfahren demonstriert [1]. In vielen Fällen konnte die Laufzeit der Heuristiken mit SPO um den Faktor 100 verkürzt werden, ohne dass Qualitätseinbußen in Kauf genommen werden mussten.

Wir empfehlen daher eine Anpassung der exogenen Parameter für zeit- und kostenintensive praktische Optimierprobleme. Die hierzu erforderlichen statistischen Grundlagen werden in Abschnitt 2 eingeführt. Das eigentliche Verfahren wird in Abschnitt 3 beschrieben. Der Artikel schließt mit einer Zusammenfassung und einem Ausblick.

2 Klassische und moderne Verfahren

2.1 Klassisches Design of Experiments

Design of Experiments besitzt eine lange Tradition in der Statistik. Grundlegende Konzepte wurden bereits von 1935 Fischer [2] entwickelt. Sie fanden in den darauf folgenden Jahrzehnten Anwendung in der industriellen Optimierung[3] und im Bereich der computergestützten Simulation [4].

Eingabegrößen, die während des Experiments kontrolliert variiert werden, werden als Faktoren oder Designvariablen bezeichnet. Im folgenden bezeichnet x den Vektor dieser Eingabegrößen. Ihre unterschiedlichen Einstellungen werden als Stufen (Level) bezeichnet, ein Sample ist eine spezielle Einstellung. Ein experimentelles Design ist eine Auswahl bestimmter Level für die einzelnen Faktoren, die im Rahmen der Experimente ausgewertet (simuliert) werden. Der erhaltene Ausgabewert, y , wird als Response (abhängige Variable) bezeichnet. Hängt dieser Wert vom Zufall ab, z.B. bei stochastischen Simulationen oder Heuristiken, wird er als Zufallsvariable Y interpretiert. Der (Haupt-) Effekt eines Faktors A kann intuitiv als die durch Variation der Einstellungen von A hervorgerufene Änderung der Response Y verstanden werden, wobei über alle Einstellungen der verbleibenden Faktoren gemittelt wird.

Basierend auf diesen Annahmen lassen sich Varianz- (ANOVA) und Regressionanalysen durchführen, um die Effekte der einzelnen Faktoren zu schätzen. Ein einfaches Regressionsmodell hat in diesem Zusammenhang die Form:

$$y = X\beta + \epsilon, \quad (1)$$

wobei ϵ den Fehler, β die Regressionskoeffizienten und X die Regressionsmatrix darstellen. Zur weiteren Erläuterung verweisen wir auf Standardwerke der experimentellen Versuchsplanung [5]. Anwendungen in der Simulation werden anschaulich in [4, 6] dargestellt, [7] beschreibt beispielhaft die Analyse für evolutionäre Algorithmen.

2.2 Stochastische Prozessmodelle

In den letzten Jahren haben stochastische Prozessmodelle zur Analyse von Computerexperimenten große Popularität gewonnen. Diese auf Sacks et al. [8] beruhenden Verfahren können als eine Erweiterung des klassischen Regressionsmodells (1) interpretiert werden. Sie werden im folgenden als Design and Analysis of Computer Experiments (DACE) Verfahren bezeichnet. Im Gegensatz zu den klassischen Regressionsmodellen gehen sie nicht von einem konstanten Fehler ϵ im gesamten Suchraum aus und benutzen andere, sogenannte raumfüllende (space filling), Designs. Beliebte auf Grund ihrer Einfachheit sind Latin Hypercube Designs (LHD). Sie platzieren Designpunkte unter geringen Einschränkungen zufällig im Inneren des Suchraums, wohingegen klassische Verfahren die Designpunkte an den Rändern (niedrige und hohe Level) wählen. Die Erzeugung eines LHDs wird als Latin Hypercube Sampling (LHS) bezeichnet.

Nach unserer Erfahrung eignen sich stochastische Prozessmodelle besser zur Analyse von Heuristiken, wobei ein großer Vorteil vermutlich durch die Designauswahl gegeben ist. Weiterführende Untersuchungen stehen hierzu allerdings noch aus. Santner et al. [9] haben wichtige Ergebnisse zusammengetragen.

3 Sequentielle Parameteroptimierung

Nachdem im vorigen Abschnitt die grundlegenden Begriffe eingeführt wurden, wird nun gezeigt, wie diese Methoden zur Verbesserung von Heuristiken (Tuning) eingesetzt werden. Die sequentielle Parameteroptimierung (SPO) stellt einen Versuch dar, statistische Verfahren möglichst effizient für das Tuning zu benutzen. SPO wird ausführlich in [1] beschrieben. Es existieren Anwendungen in folgenden Bereichen:

1. Maschinenbau: Temperierbohrungen [10]
2. Luft- und Raumfahrt: Tragflächenoptimierung [11]
3. Simulation und Optimierung: Fahrstuhlsteuerung [12, 13, 14]
4. Verfahrenstechnik: Entwurf von Destillationsanlagen [15]
5. Wirtschaftswissenschaft: Modellierung eines Bodenmarktes [16]
6. Touroptimierungs- und Tourzuordnungsprobleme [17]

Anwendungsfelder im Bereich der Grundlagenforschung sind:

1. Algorithm Engineering: Graph drawing [18]
2. Statistik: Selektionsverfahren optimal computational budget allocation (OCBA) für Partikel-Schwarm-Verfahren [19, 20]
3. Evolutionsstrategien: Threshold Selektion und Schrittweitensteuerung [21]
4. Computational Intelligence: Algorithmische Chemie [15]

5. Partikel-Schwarm-Optimierung: Analyse und Anwendung [22]
6. Numerische Mathematik: Vergleich und Analyse klassischer und moderner Optimierungsalgorithmen [23]

SPO verwendet eine sequentiellen Vorgehensweise bei der Bestimmung geeigneter Designpunkte. Ausgangspunkt ist ein Initialdesign, das zumeist durch LHS generiert wird. Diese Designpunkte bestimmen eindeutig die Parametrisierung der Heuristik. Die Heuristik wird mit diesen Einstellungen ausgeführt, so dass ein Regressionsmodell gebildet werden kann. Dieses Modell wird zur Vorhersage guter Designpunkte herangezogen. Weitere Einstellungen, mit denen die Heuristik ausgeführt wird, sollen einerseits (i) die Güte der Heuristik und (ii) andererseits auch das Modell verbessern. Um diesen Tradeoff zu modellieren, bedient sich SPO des Ansatzes der erwarteten Verbesserung (expected improvement) nach Schonlau [24]. Es werden also sequentiell vielversprechende Designpunkte erzeugt, die das Modell und die Heuristik verbessern.

Da die meisten Heuristiken stochastische Suchverfahren sind, wird durch ein geschicktes Erhöhen der Wiederholungen im Laufe des Tunings die Fehlerwahrscheinlichkeit (Wahrscheinlichkeit, dass ein schlechter Designpunkt nur zufällig ein gutes Ergebnis liefert) verringert.

Für den Anwender verläuft dieser Tuningprozess im Verborgenen, allerdings können die dabei berechneten Statistiken jederzeit abgerufen werden und stehen somit für weiterführende Analysen zur Verfügung. Wir beschreiben nun SPO aus der Sicht des Anwenders, d.h. die Schritte, die ein Anwender für ein Tuning durchführen muss.

Schnittstellen

Die zu SPO gehörige Toolbox SPOT (Sequential Parameter Optimization Toolbox) ist vollständig in MATLAB implementiert und erfordert neben dem MATLAB Grundsystem die Statistik Toolbox. SPOT arbeitet dateibasiert, die Schnittstellen sind einfache Textdateien, die im folgenden beschrieben werden.

SPOT benötigt zwei Dateien:

1. `roi` und
2. `conf` Dateien.

`roi` Dateien spezifizieren den Suchraum (region of interest, ROI) der Einstellungen für die Heuristik. Bild 1 zeigt eine `roi` Datei. Die `conf` Datei beschreibt SPO interne Einstellungen wie z.B. die Wahl des stochastischen Prozessmodells oder die Anzahl initialer Designpunkte für das LHD. Wir verweisen hierzu auf die SPOT Dokumentation [25]. Nachdem diese Dateien erstellt worden sind, kann SPOT aufgerufen werden. SPOT erzeugt ein initiales Design, das in eine sog. `des` (Design) Datei geschrieben wird. Die Heuristik muss in der Lage sein, die Werte der `des` Datei auszulesen und mit den darin spezifizierten Designs zu starten. Für jeden Designpunkt erzielt die Heuristik einen Funktionswert, der zusammen mit diesem Designpunkt zeilenweise in eine `res` (result) Datei geschrieben wird. Die Werte aus der `res` Datei werden von SPOT zur Verbesserung des Regressionsmodells und zur Vorhersage neuer Designpunkte benutzt.

```

name low high isint pretty
NPARENTS 1 10 TRUE 'NPARENTS'
NU 1 5 FALSE 'NU'
TAU1 0.5 3 FALSE 'TAU1'

```

Bild 1: Region of interest (ROI) Datei.

Bild 2 illustriert das Zusammenspiel der einzelnen SPO Komponenten.

Neben der Darstellung des Regressionsmodells und des zugehörigen mittleren erwarteten Fehlers zur Abschätzung der Modellgüte stehen nach Beendigung der SPO eine Vielzahl von Auswertungsmöglichkeiten und Ergebnisdarstellungen (grafisch wie auch textuell) zur Verfügung. Wir empfehlen Methoden der explorativen Datenanalyse (EDA). Momentan existieren bereits mehr als ein Dutzend unterschiedlicher EDA Tools, von denen einige im folgenden dargestellt werden (Bild 3 und 4).

4 Zusammenfassung und Ausblick

Die sequentielle Parameteroptimierung ist ein erster Versuch, ein anwendungsfreundliches Tool zum Tunen zur zur experimentellen Analyse heuristischer Suchverfahren bereitzustellen. Der Anwender muss ein Minimum an Informationen über den zu untersuchenden Algorithmus zusammenstellen und erhält fast automatisch eine verbesserte Einstellung.

Da die theoretische Analyse in der Regel nicht in der Lage ist, sinnvolle Einstellungen für konkrete Optimierprobleme bereitzustellen, schließt SPO diese Lücke durch eine geschickte Kombination moderner statistischer Verfahren. Der Anwender erhält zusätzlich die Möglichkeit, verschiedene Heuristiken (oder auch klassische Optimierverfahren) auf eine möglichst objektive Art und Weise zu vergleichen. Durch die dateibasierten Schnittstellen ist SPO ohne großen Aufwand und unabhängig von Betriebssystemen oder Programmiersprachen einsatzfähig. Interessant ist für uns weiteres Feedback von industriellen Anwendern, um sicherzugehen, dass möglichst viele Anwendungsszenarien durch SPO abgedeckt werden.

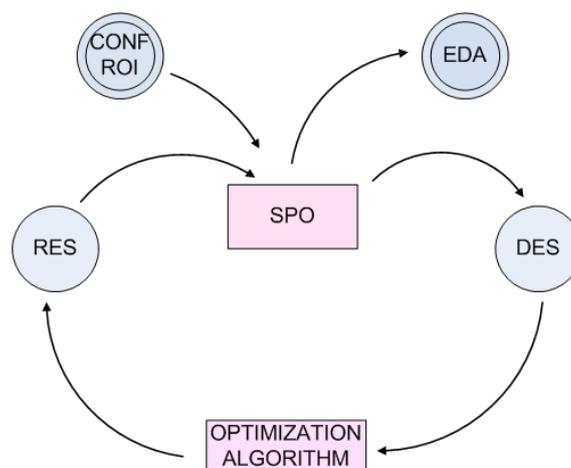


Bild 2: SPOT Datenfluss

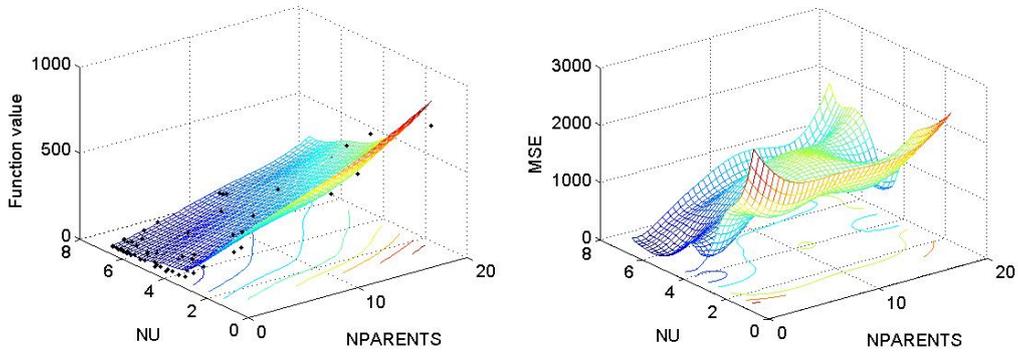


Bild 3: Regressionsmodell (li.) und mittlerer quadratischer Fehler (re.)

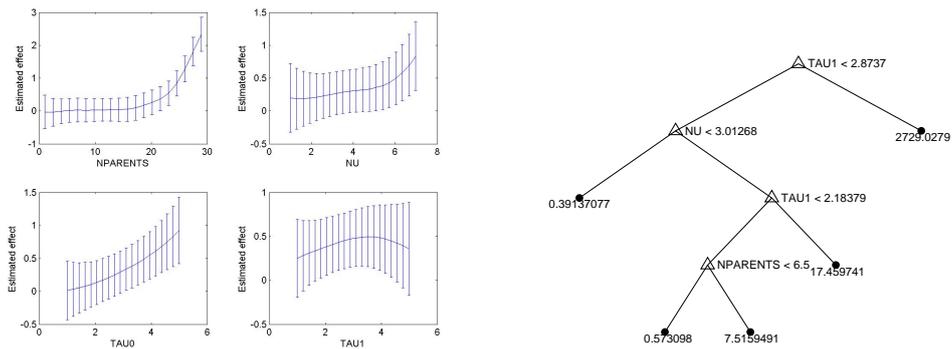


Bild 4: Haupteffekte (li.) und Darstellung der Effekte durch einen Regressionsbaum (re.)

Literatur

- [1] Bartz-Beielstein, T.: *Experimental Research in Evolutionary Computation—The New Experimentalism*. Berlin, Heidelberg, New York: Springer. 2006.
- [2] Fisher, R. A.: *The Design of Experiments*. Edinburgh: Oliver and Boyd. 1935.
- [3] Box, G. E. P.; Hunter, W. G.; Hunter, J. S.: *Statistics for Experimenters*. New York NY: Wiley. 1978.
- [4] Kleijnen, J. P. C.: *Statistical Tools for Simulation Practitioners*. New York NY: Marcel Dekker. 1987.
- [5] Montgomery, D. C.: *Design and Analysis of Experiments*. New York NY: Wiley, 5th Aufl. 2001.
- [6] Kleijnen, J. P. C.; Van Groenendaal, W.: *Simulation—A Statistical Perspective*. Chichester, U.K.: Wiley. 1992.
- [7] Bartz-Beielstein, T.: Experimental Analysis of Evolution Strategies—Overview and Comprehensive Introduction. Interner Bericht des Sonderforschungsbereichs 531 Computational Intelligence CI–157/03, Universität Dortmund, Germany. 2003.
- [8] Sacks, J.; Welch, W. J.; Mitchell, T. J.; Wynn, H. P.: Design and analysis of computer experiments. *Statistical Science* 4 (1989) 4, S. 409–435.
- [9] Santner, T. J.; Williams, B. J.; Notz, W. I.: *The Design and Analysis of Computer Experiments*. Berlin, Heidelberg, New York: Springer. 2003.
- [10] Mehnen, J.; Michelitsch, T.; Bartz-Beielstein, T.; Henkenjohann, N.: Systematic analyses of multi-objective evolutionary algorithms applied to real-world problems using statistical design of experiments. In: *Proceedings Fourth International Seminar Intelligent Computation in Manufacturing Engineering (CIRP ICME'04)* (Teti, R., Hg.), Bd. 4, S. 171–178. Naples, Italy. ISBN 88-87030-79-0. 2004.
- [11] Bartz-Beielstein, T.; Naujoks, B.: Tuning Multicriteria Evolutionary Algorithms for Airfoil Design Optimization. Interner Bericht des Sonderforschungsbereichs 531 Computational Intelligence CI–159/04, Universität Dortmund, Germany. 2004.
- [12] Beielstein, T.; Markon, S.: Threshold Selection, Hypothesis Tests, and DOE Methods. Interner Bericht des Sonderforschungsbereichs 531 Computational Intelligence CI–121/01, Universität Dortmund, Germany. 2001.
- [13] Beielstein, T.; Ewald, C.-P.; Markon, S.: Optimal Elevator Group Control by Evolution Strategies. In: *Proceedings Genetic and Evolutionary Computation Conf. (GECCO 2003), Chicago IL, Part II* (Cantú-Paz, E.; et al., Hg.), Bd. 2724 von *Lecture Notes in Computer Science*, S. 1963–1974. Berlin, Heidelberg, New York: Springer. ISBN 3-540-40603-4. 2003.
- [14] Markon, S.; Kita, H.; Kise, H.; Bartz-Beielstein, T. (Hg.): *Modern Supervisory and Optimal Control with Applications in the Control of Passenger Traffic Systems in Buildings*. Berlin, Heidelberg, New York: Springer. 2006.

- [15] Bartz-Beielstein, T.; Lasarczyk, C.; Preuß, M.: Sequential Parameter Optimization. In: *Proceedings 2005 Congress on Evolutionary Computation (CEC'05)*, Edinburgh, Scotland (McKay, B.; et al., Hg.), Bd. 1, S. 773–780. Piscataway NJ: IEEE Press. ISBN 0-7803-9363-5. 2005.
- [16] de Vegt, M.: Einfluss verschiedener Parametrisierungen auf die Dynamik des Partikel-Schwarm-Verfahrens: Eine empirische Analyse. Interner Bericht der Systems Analysis Research Group SYS-3/05, Universität Dortmund, Fachbereich Informatik, Germany. 2005.
- [17] Bartz-Beielstein, T.; Chmielewski, A.; Janas, M.; Naujoks, B.; Scheffermann, R.: Optimizing Door Assignment in LTL-Terminals by Evolutionary Multiobjective Algorithms. In: *Proceedings 2006 Congress on Evolutionary Computation (CEC'06) Within Fourth IEEE World Congress on Computational Intelligence (WCCI'06)*, Vancouver, Canada (Fogel, D. B.; et al., Hg.). Piscataway NJ: IEEE. 2006.
- [18] Tasic, M.: *Evolutionäre Kreuzungsminimierung*. Diploma thesis, University of Dortmund, Germany. 2006.
- [19] Bartz-Beielstein, T.; Blum, D.; Branke, J.: Particle Swarm Optimization and Sequential Sampling in Noisy Environments. In: *Proceedings 6th Metaheuristics International Conference (MIC2005)* (Hartl, R.; Doerner, K., Hg.), S. 89–94. Vienna, Austria. 2005.
- [20] Bartz-Beielstein, T.; Blum, D.; Branke, J.: Particle Swarm Optimization and Sequential Sampling in Noisy Environments. In: *Metaheuristics—Progress in Complex Systems Optimization* (Hartl, R.; Doerner, K., Hg.), Springer Operations Research/Computer Science Interfaces Book Series. Berlin, Heidelberg, New York: Springer. 2006.
- [21] Bartz-Beielstein, T.: Evolution Strategies and Threshold Selection. In: *Proceedings Second International Workshop Hybrid Metaheuristics (HM'05)* (Blesa Aguilera, M. J.; Blum, C.; Roli, A.; Sampels, M., Hg.), Bd. 3636 von *Lecture Notes in Computer Science*, S. 104–115. Berlin, Heidelberg, New York: Springer. ISBN 3-540-28535-0. 2005.
- [22] Bartz-Beielstein, T.; Parsopoulos, K. E.; Vrahatis, M. N.: Analysis of Particle Swarm Optimization Using Computational Statistics. In: *Proceedings International Conference Numerical Analysis and Applied Mathematics (ICNAAM)* (Simos, T.-E.; Tsitouras, C., Hg.), S. 34–37. Weinheim, Germany: Wiley-VCH. ISBN 3-527-40563-1. 2004.
- [23] Bartz-Beielstein, T.; Parsopoulos, K. E.; Vrahatis, M. N.: Design and analysis of optimization algorithms using computational statistics. *Applied Numerical Analysis & Computational Mathematics (ANACM)* 1 (2004) 2, S. 413–433.
- [24] Schonlau, M.: *Computer Experiments and Global Optimization*. Dissertation, University of Waterloo, Ontario, Canada. 1997.
- [25] Bartz-Beielstein, T.; Lasarczyk, C.; Preuß, M.: Sequential Parameter Optimization Toolbox. <http://ls11-www.cs.uni-dortmund.de/people/tom/ExperimentalResearchPrograms.html>. 2006.