

Stochastic optimization of a cold atom experiment using a genetic algorithm

W. Rohringer, R. Bücker, S. Manz, T. Betz, Ch. Koller, M. Göbel, A. Perrin,
J. Schmiedmayer, and T. Schumm^{a)}

Atominstitut der Österreichischen Universitäten, TU-Wien, Stadionallee 2, 1020 Vienna, Austria

(Received 30 October 2008; accepted 6 December 2008; published online 29 December 2008)

We employ an evolutionary algorithm to automatically optimize different stages of a cold atom experiment without human intervention. This approach closes the loop between computer based experimental control systems and automatic real time analysis and can be applied to a wide range of experimental situations. The genetic algorithm quickly and reliably converges to the most performing parameter set independent of the starting population. Especially in many-dimensional or connected parameter spaces, the automatic optimization outperforms a manual search. © 2008 American Institute of Physics. [DOI: [10.1063/1.3058756](https://doi.org/10.1063/1.3058756)]

In experimental research, scientists are often confronted with the task of finding ideal parameters to perform a measurement. Although in modern setups, experimental parameters are handled by computer-based control systems, and also the evaluation of the experimental output is often performed by real-time computer systems; optimization is still essentially carried out by hand.

In this article we present the application of a genetic algorithm to automatically optimize control parameters in an atomic physics experiment. Such algorithms have been employed to tackle optimization problems in different fields, ranging from mathematics and algorithmics¹ over economics² and aerospace engineering³ to game theory.⁴ Our algorithm generates a set of random starting parameters, automatically performs the experiment, and recovers information about the performance of the parameter set from real-time analysis of the experimental result. This information is used to generate a next generation of parameters with better performance until an optimum is reached (see Fig. 1). This approach can be implemented in any experimental setup that requires optimization, provided that it is computer controlled and features real-time analysis.

The experimental setup used in these investigations is a single-chamber atom chip setup for the creation and study of Bose–Einstein condensates.⁵ Precooled ⁸⁷Rb atoms are loaded into a magnetic trap created by current carrying macroscopic copper structures. Evaporative cooling is applied, creating a sample of 10^6 atoms at microkelvin temperatures. These atoms are loaded onto an atom chip where microscopic traps are generated by lithographically patterned wires^{6,7} and cooled to quantum degeneracy in a second phase of evaporative cooling. The wire structures on the atom chip allow a large variety of experiments such as matter wave interferometry⁸ or the study of low-dimensional quantum gases.⁹ After the experiment is performed, the atomic density distribution is recorded using standard absorption or fluorescence imaging¹⁰ either in the atom chip trap or in free time-of-flight expansion. As the atomic sample is destroyed in the imaging process, the above procedure has to be repeated many times to accumulate statistics. The duration of an experiment is 35 s, the experimental cycle is repeated continu-

ously, and data taking can cover several days. It shall be pointed out that individual operations within the experimental sequence take place on a timescale of the order of tens of microseconds, necessitating a relative temporal resolution on the 10^{-6} level. A commercial real time control system (ADwin Pro I) in connection with a custom interface software coordinates the 60 involved individual devices. They are governed by a total parameter space of 2300 values during a typical experimental sequence, a subspace of which is made accessible for each individual optimization task.

Each experimental cycle produces an image of the atom cloud, which yields a projected two-dimensional measurement of the particle density distribution. Important physical quantities such as the total number of atoms N , the peak density n_0 , the position of the cloud, or the phase space density can already be derived from a single picture. In terms of optimization theory, the above quantities can be interpreted as objective functions $f(j_i)$, where j_i denotes a single parameter set out of the overall parameter space. Optimizing the experimental setup corresponds to finding a parameter set j_i that maximizes (minimizes) a given objective function. Probing this function hence means performing the experimental sequence with a given parameter set and determining the desired measurement value. Since in our experiments usually only few or even a single output quantity is considered, the problem is a prime example of a single-objective optimization problem. Such tasks can be tackled effectively by stochastic algorithms.¹¹

The simplest approach would be to evaluate the objective function on the entire parameter space. However, the number of necessary experimental runs grows rapidly with the dimensions of the parameter space. With ten evaluations per parameter, a scan over one, two, three, and four param-

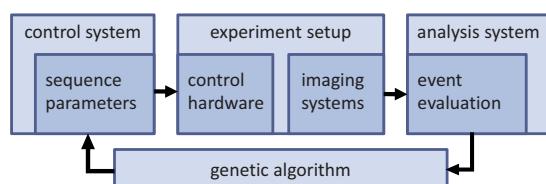


FIG. 1. (Color online) The feedback loop between the control system, experiment, and real-time analysis is closed by the genetic algorithm.

^{a)}Electronic mail: schumm@atomchip.org.

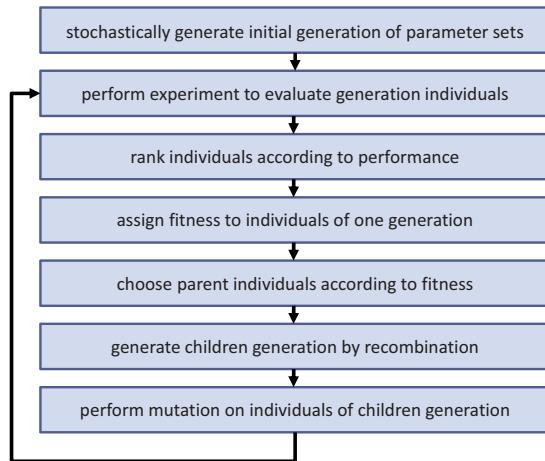


FIG. 2. (Color online) Block diagram of the optimization process.

eters (i.e., 10, 100, 1000, and 10 000 parameter sets) takes about 6 min, 1 h, 10 h, or 4 days, respectively.

An alternative approach are deterministic algorithms such as hill climbing,¹² where knowledge on the gradient of the objective function is used to generate improved parameter sets. They perform well for problems that behave approximately linear in the objective function and guarantee convergence for unimodal problems represented by objective functions featuring only one extremum. For multimodal problems, however, they are prone to converging against local extrema.

Stochastic algorithms trade guaranteed convergence for shorter runtime in applications with a high number of (interacting) parameters.¹² They introduce a random element into the selection of points in parameter space. There is no formal way to prove whether a found solution is really the global optimum. Within stochastic algorithms, genetic algorithms belong to the class of parallel optimization algorithms, which do not rest on the evaluation of individual states in parameter space but rather evaluate the performance of an ensemble of states (generation). Compared to local algorithms, this approach is less dependent on the specific initial conditions for each optimization run. Out of the different parallel methods, genetic algorithms are well explored, with numerous known implementational strategies.^{13,14}

Figure 2 illustrates the working concept of our genetic algorithm. In a first step, a starting population is distributed randomly in a parameter subspace spanned by two to five variables associated with a certain stage of the experiment. Sometimes population individuals near a known optimum or on the edges of parameter space are added to speed up convergence or test the performance of the algorithm. The size of the initial population has to be chosen carefully: large populations are time consuming as they increase the number of evaluations per generations as well as the number of generations. Too small starting populations have shown to lead to premature convergence as the diversity of the individuals decreases quickly. An intermediate initial population size of 10–15 individuals has shown to perform well in various optimization tasks.

To evaluate the performance of a population, the experimental sequence is executed with the parameter settings represented by the individuals. The individuals are then sorted according to their performance with respect to the selected

objective function, e.g., particle number or temperature.

In analogy to evolution terminology, the fitness of a population individual $F(j_i)$ is a measure for the probability for this state to be selected for reproduction. We chose linear ranking based fitness assignment,¹⁵ which considers the relative performance of individuals but is independent of the achieved absolute value of the objective function. It distributes the individual fitnesses between 0 and 1: $F(j_i)=2/n(1-i-1/n-1)$, where n is the size of the population.

The selection process determines the parent individuals that will “breed” the next generation. From several possible selection schemes, we chose stochastic universal sampling¹³ where parent individuals are chosen according to their fitness but which still introduces a stochastic process so that less performing individuals can contribute to the next generation and population diversity is kept up. The total number of parent states is adjusted to $\frac{2}{3}$ of the total population with two children per pair of parent states. Since the population size is chosen to be fixed throughout the optimization process, the free slots are filled up by the best states from the previous generation. This makes our algorithm “elitist,” assuring that the states showing best performance are not lost during the optimization process, which speeds up convergence.

The next generation individuals are created by intermediary recombination.^{16,17} The parameters of the two parent states are taken as limiting edges of a hypercube in parameter space; the two child states are distributed randomly within this volume. To avoid too rapid decrease in the children’s parameter space, its volume is stretched by a factor of 1.5.^{16,17}

Mutation is introduced to further impede convergence against local extrema. Its influence on the performance of genetic algorithms has been investigated in great detail, and several well performing schemes are known.^{17,18} For each state within one generation, there is a certain probability for random alteration (in our case it is 1%) with the additional constraint that only one of the state’s variables is affected. The mutated values are obtained by a mechanism (shown in Ref. 19) that constructs $j_i^{\text{mut}}=j_i+s_i r D_i 2^{-uk}$, where j_i^{mut} and j_i denote the mutated and the source state, respectively, s_i randomly chooses the sign of the mutation step, and r defines the mutation range as a fraction of the accessible parameter space D_i . The last factor designates the actual distribution, characterized by the random number u uniformly distributed in $[0, 1]$ and the mutation precision k (Mühlenbein’s mutation). With $r=0.1$ and $k=10$ we adapted established values.¹⁶ After the mutation step the next child population is ready to be tested on the experiment.

For simplicity, the optimization cycle is repeated for a preset number of ten times or is stopped manually when the diversity in the population drops to the point that all states are equal. More evolved convergence criteria such as the distance of individuals in parameter space or convergence of the objective function can be implemented easily.

The described genetic algorithm has been used to optimize various stages of the experimental sequence,⁵ such as the optical pumping into magnetically trappable states or the evaporative cooling ramps. Figure 3 shows an example of an optimization of critical experimental parameters during the phase of optical molasses, which effectuates an additional cooling of the atoms coming from the magneto-optical trap before they are loaded into a magnetic wire trap. The three-

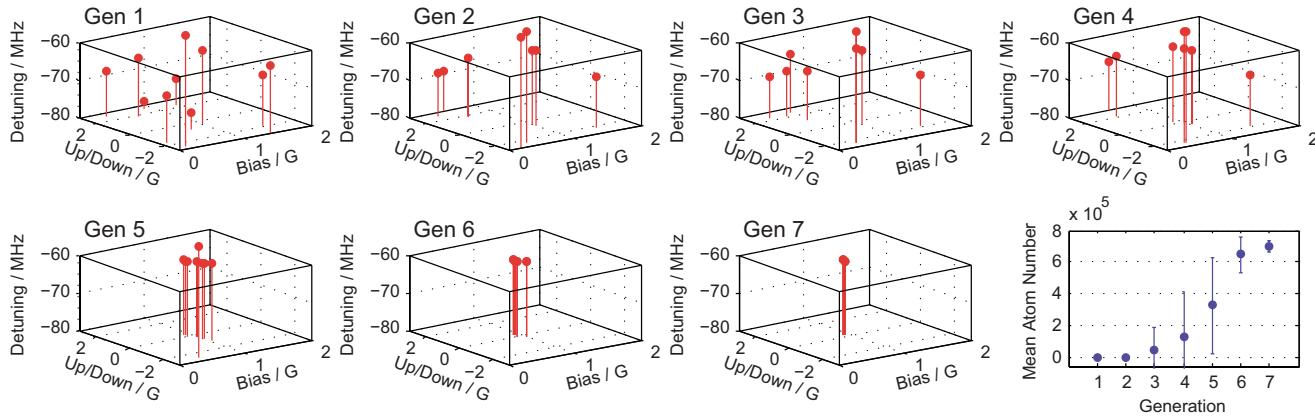


FIG. 3. (Color online) Positions of the individuals in the three-dimensional parameter space through seven generations. Convergence is clearly observed. On the bottom right, the achieved mean atom number for each generation is depicted. The error bars correspond to the standard deviation of the individual's results within each generation.

dimensional parameter space is spanned by two magnetic fields (up/down and bias) and the detuning of the cooling laser frequency with respect to the atomic transition. The two magnetic fields have to be optimized to compensate the Earth's magnetic field and possible unknown stray fields. The laser frequency has to be optimized to achieve an efficient cooling (high detuning) while keeping the atoms confined (low detuning). As shown in Fig. 3, the genetic algorithm converges to optimum parameters within seven generations with a starting population of ten individuals. The overall runtime was 30 min; as compared to 386 h it would need to sample the entire parameter volume with the same resolution. Note that the optimum was found although the first two generations contained not a single well performing individual. The algorithm rapidly and reliably finds optimum values for various optimization tasks and reproduces or outperforms manually optimized parameters sets. For too small starting populations, it has shown to be prone to premature convergence leading to suboptimal final parameters. Numerous characteristics of the algorithm, such as the size of the starting population, the amount of reproducing individuals per generation, or the mutation rate and range, can be adapted, weighing optimality against runtime.

In a future work we will extend the approach to objective functions that cannot be evaluated in a single experimental run (image) but probe temporal dynamics of the system or the scattering of measured parameters around the mean value, requiring several measurements to evaluate a single parameter set. This will allow applications of the genetic algorithms beyond mere optimization tasks such as the reduction in excitations or the enhancement of squeezing or entanglement in dynamic matter wave interferometry.²⁰ Ultimately, we aim to develop “experimental optimal control,” in analogy to optimal control theory, where the result of optimization directly provides scientific insight, not accessible by other means.

To conclude, we implemented a robust and efficient stochastic genetic algorithm to automatically perform time-consuming optimization tasks in a complex cold atom experiment. In the given example, the algorithm converged to optimal settings within 30 min, as compared to several hours

typically required for manual search. Our approach closes the loop between computer-based experimental control systems and real-time data analysis and can be implemented in many situations in experimental research.

This work was supported by the EU Integrated Project FET/QUIP “SCALA” and the Austrian Science Fund FWF Project Nos. P20372 and P21080. S.M. acknowledges support from the Vienna graduate school COQUS.

- ¹C. M. Anderson-Cook, R. L. Haupt, and S. E. Haupt, *J. Am. Stat. Assoc.* **100**, 1099 (2005).
- ²E. K. Burke and J. P. Newall, *IEEE Trans. Evol. Comput.* **3**, 63 (1999).
- ³S. Obayashi, D. Sasaki, Y. Takeuchi, and N. Hirose, *IEEE Trans. Evol. Comput.* **4**, 182 (2000).
- ⁴J. W. Weibull, *Evolutionary Game Theory* (MIT Press, Cambridge, 1995).
- ⁵S. Wildermuth, P. Krüger, C. Becker, M. Brajdic, S. Haupt, A. Kasper, R. Folman, and J. Schmiedmayer, *Phys. Rev. A* **69**, 030901 (2004).
- ⁶S. Groth, P. Krüger, S. Wildermuth, R. Folman, T. Fernholz, J. Schmiedmayer, D. Mahalu, and I. Bar-Joseph, *Appl. Phys. Lett.* **85**, 2980 (2004).
- ⁷M. Trinker, S. Groth, S. Haslinger, S. Manz, T. Betz, S. Schneider, I. Bar-Joseph, T. Schumm, and J. Schmiedmayer, *Appl. Phys. Lett.* **92**, 254102 (2008).
- ⁸T. Schumm, S. Hofferberth, L. M. Andersson, S. Wildermuth, S. Groth, I. Bar-Joseph, J. Schmiedmayer, and P. Krüger, *Nat. Phys.* **1**, 57 (2005).
- ⁹S. Hofferberth, I. Lesanovsky, B. Fischer, T. Schumm, and J. Schmiedmayer, *Nature (London)* **449**, 324 (2007).
- ¹⁰W. Ketterle, D. S. Durfee, and D. Stamper-Kurn, Proceedings of the International School of Physics “Enrico Fermi”, 1999 (unpublished), p. 67.
- ¹¹A. Assion, T. Baumert, M. Bergt, T. Brixner, B. Kiefer, V. Seyfried, M. Strehle, and G. Gerber, *Science* **282**, 919 (1998).
- ¹²T. Weise, *Global Optimization Algorithms* (GNU, Boston, 2008).
- ¹³N. A. Barricelli, *Acta Biotheor.* **16**, 69 (1962).
- ¹⁴N. A. Barricelli, *Acta Biotheor.* **16**, 99 (1963).
- ¹⁵J. E. Baker, Proceedings of the Second International Conference on Genetic Algorithms, Mahwah, USA, 1987 (unpublished), p. 14.
- ¹⁶H. Pohlheim, *Evolutionäre Algorithmen: Verfahren, Operatoren und Hinweise für die Praxis* (Springer, Berlin, 1999).
- ¹⁷F. Herrera, M. Lozano, and J. L. Verdegay, *Artif. Intell. Rev.* **12**, 265 (1998).
- ¹⁸R. L. Haupt, IEEE Antennas and Propagation Society International Symposium, 2000 (unpublished), Vol. 2, p. 1034.
- ¹⁹H. Mühlenein and D. Schlierkamp-Voosen, *Analysis of Selection, Mutation and Recombination in Generic Algorithms* (Springer, London, UK, 1995).
- ²⁰J. Esteve, C. Gross, A. Weller, S. Giovanazzi, and M. Oberthaler, *Nature (London)* **445**, 1216 (2008).