

The Genetic Hybrid Algorithm (GHA)
**A General Platform for Distributed Numerical
Computations and Algorithmic Design**

Background and Examples

Ralf Östermark
ralf.ostermark@abo.fi
<http://web.abo.fi/fak/tkf/at/ose/ralfostermark.html>
School of Business and Economics
at Åbo Akademi University
FIN-20500 ÅBO, Finland

1. Background

The Genetic Hybrid Algorithm (GHA) is a flexible platform for high-performance numerical computation, developed for single and parallel processor computers since 1999. The key idea of the platform is to provide powerful equipment for designing new algorithms in numerical computation. GHA has been tested on numerous difficult problems of finance and engineering, vector valued time series modeling, mathematical programming - especially mixed integer nonlinear programming - and simulation. In this user's guide we will show by example, how different problems are solved by the platform and indicate how it could be used in developing own algorithms for specific numerical problems. GHA and its support libraries are placed as linkable libraries on the Linux main frame computer of Åbo Akademi University and the massively parallel Cray XT supercomputer at the Centre of Scientific Computing (CSC) in Helsinki. For testing the algorithm on these computers, you need a user id as provided by the respective computer centre. Any commercial applications require acquisition of the platform.

The core algorithm is built in object-oriented strict ANSI C. Therefore, the platform can be further developed in future research, without hampering the functionality or solvability of previous applications. The algorithm runs on both single and parallel processor computers. On parallel machines, communication between processors is handled through MPI functions. Extensive heap memory checking by Valgrind (Julian Seward) show that no memory leaks are possible. Some memchecks are presented in the MINLP-discussion subsequently.

A selection of the leading nonlinear and linear programming algorithms known today is connected as support libraries to the platform and thoroughly tested on single and parallel computers. The source code for these algorithms has been obtained from the corresponding research groups during 2005-2009. These algorithms have been developed in the universities of Stanford, California and Maryland in USA and the University of Bayreuth in Germany. The leaders of the research groups have been key authorities in non-linear programming over the last 25 years. A high performance nonlinear algorithm has been developed by the author for comparisons to the established codes. The algorithms are used as node solvers in difficult mixed-integer nonlinear programming problems on single or parallel processors, where the communication between local solution trees is monitored by GHA.

Large scale mixed integer nonlinear programming (MINLP), General Disjunctive programming (GDP) and Quadratic Assignment (QAP) problems arise frequently in economics and engineering. For example, when assessing the risk surface of the firm within a multi-period setting, where the corporate decisions are connected to the financial statements through internal accounting logic, large scale GDP- or MINLP-problems will be encountered.

Active-set Sequential Programming (SQP) methods and interior point methods are currently considered the most powerful algorithms for large-scale nonlinear programming. In non-convex or irregular problems, the algorithms cannot guarantee the global solution. However, the established algorithms usually yield at least a feasible MINLP-solution when used in a branch-and-bound search process. Certain non-smooth problems can be reformulated as smooth optimization problems, but in general a methodology for non-differentiable functions is required for non-smooth optimization. New methods for difficult optimization problems

are readily connected to GHA, for example in order to utilize its parallel capabilities.

Integrated geno-mathematical systems, where artificial intelligence is connected to mathematical programming methodology on parallel supercomputers, provide a powerful basis for simplifying difficult irregular optimization problems and solving them concurrently. In several cases of relevance in practice, the local branch-and-bound trees of the parallel processors are considerably smaller and the solution superior to the one obtained from the large search conducted by a single processor.

Several vector-valued time series algorithms have been developed by the author and connected to the platform as separate linkable libraries. A vector-valued state space algorithm derived by professor Masanao Aoki in University of California has been implemented based on the cooperation and joint reporting during 1995-1997.

Research linked to GHA focuses on high performance computing in finance and engineering, with the target to enhance the development of single and parallel geno-mathematical solutions to difficult numerical problems.

GHA has been used for difficult mixed-integer nonlinear programming problems in both sequential and parallel tests. The results have been encouraging in comparison to competing approaches. One of the key features of GHA is the ability to combine rigorous mathematical algorithms with artificial search engines, an advantage frequently needed in, e.g., MINLP-problems.

The heap memory usage of GHA and its central support libraries have been checked using the powerful Valgrind debugger (cf. <http://valgrind.org>).

The scalability of the platform has been previously demonstrated on the massively parallel supercomputers Cray T3E and IBMSC at the Centre of Scientific Computing (CSC) Helsinki in vector-valued time series estimation problems and MINLP-problems.

During 2009-2011, we have demonstrated scalability of GHA on Cray XT at CSC with up to 4048 cores and at the Jugene supercomputer within a PRACE project with up to 65536 cores. Jugene is currently the fastest massively parallel supercomputer in Europe. The computational platform does not restrict the number of processors to be used. The possible limitations arise from the computational problem at hand.

During 2013, scalability has been demonstrated on the CrayXC30 at CSC with the maximum number of processors made available for the test.

We have shown that the complexity of binary mixed-integer-nonlinear problems can be significantly reduced on parallel processors using asynchronous mesh interrupts and binary coding of local box constraints. The local branch-and-bound trees are solved using efficient non-linear optimization algorithms monitored by GHA. Lately, we have extended the approach to general discrete valued MINLP-problems using shifted Gray-coding of the local box constraints. This approach allows a complete mapping of the Cartesian search space in

MINLP-problems and a corresponding simplification of the computational task for the local processors.

The scalability of the multi-period firm model Firm_GMP (Östermark [2015]) has been demonstrated on Cray XC40 with the maximum allowed number of cores. During September 2019, GHA and support libraries have been ported to the new Atos supercomputer at CSC. Scalability of GHA → Firm_GMP was demonstrated with the maximum number of processors when deriving the risk surface of example firms.

An accelerator function placed in critical stages of the main loop of GHA enables the connection of external algorithms – available packages or tailor made algorithms designed by the user - to the platform. For example, high-functionality MATLAB-code can be integrated into GHA on platforms having the mcc-compiler and the necessary object libraries. The accelerator forms a window that allows the researcher/problem solver to tackle the following question: **How can I solve the computational problem at hand using the best available algorithms in the world?**

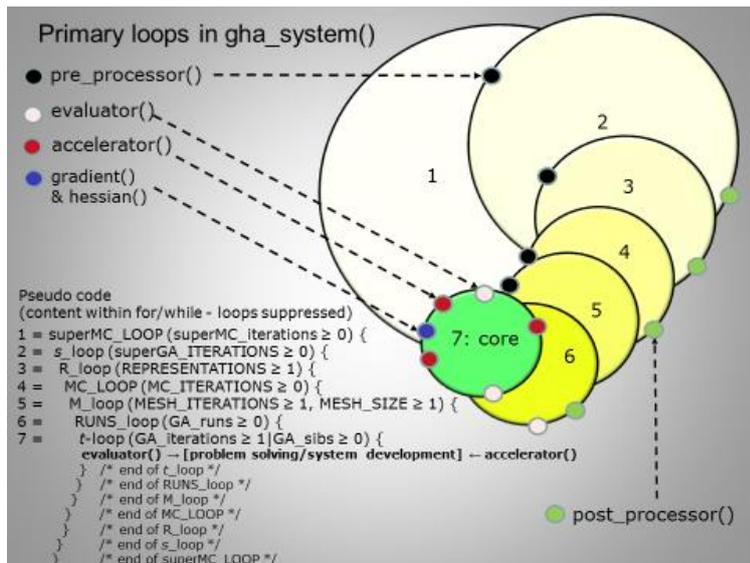
GHA is founded on two main principles:

- (i) **allowing meaningful connections to available high-performance algorithms,**
- (ii) **maximizing the intelligence of the processors with respect to computational resources.**

These principles support the construction of scalable algorithms for numerical problems in computational finance and engineering.

We welcome new ideas that will stimulate the continuing efforts to simplify numerical problem solving and extend the solution potential of established and new algorithms through parallel processing.

Note: the links embedded in the below documents may not open properly with Firefox.



2. [Introduction to GHA with code examples](#)

3. [Using GHA and its support library MINLP_bb in MINLP-problems](#)

References where GHA has been tested in single and parallel processing (cf. <http://www.academia.edu>)

Östermark, R (1999): Solving irregular econometric and mathematical optimization problems with a genetic hybrid algorithm. *Computational Economics* **13**:2, 103-115.

Östermark, R (1999): Solving a nonlinear nonconvex trim loss problem with a genetic hybrid algorithm, *Computers & Operations Research* **26**, 623-635.

Östermark, R (1999): A Neuro-genetic algorithm for heteroskedastic time series processes, *Soft Computing* **3**:4, 206-220.

Östermark, R (2000): A hybrid genetic fuzzy neural network algorithm designed for classification problems involving several groups, *Fuzzy Sets and Systems* **114**:2, 311-324

Östermark, R (2001): New tests with a parallel genetic hybrid algorithm, *Kybernetes. The International Journal of Systems and Cybernetics* **30**:2, 193-203.

Östermark, R (2001): Genetic modelling of multivariate EGARCHX-processes. Evidence on the international asset return signal response mechanism. *Computational Statistics & Data Analysis*, **38**:1, November, 71-94.

Östermark, R (2002): Automatic detection of parsimony in heteroskedastic time series processes. Empirical tests on global asset returns with parallel geno-mathematical programming, *Soft Computing* **6**:1, 45-63.

Östermark, R (2002): Designing a superstructure for parametric search for optimal search spaces in non-trivial optimization problems, *Kybernetes. The International Journal of Systems and Cybernetics* **31**:2, 2002, 255-281. (EMERALD Highly Commended Award winning article).

- Östermark, R (2002): A flexible Genetic Hybrid Algorithm for nonlinear mixed-integer programming problems, *Evolutionary Optimization* **1**:1, 41-52.
- Östermark, (2003): A multipurpose parallel Genetic Hybrid Algorithm for nonlinear nonconvex programming problems. *European Journal of Operational Research*, **152**, 195-214.
- Östermark, R (2005): Dynamic Portfolio Management under Competing Representations. *Kybernetes. The International Journal of Systems and Cybernetics*. **34**:9/10, 1517-1550.
- Östermark, R (2007): A flexible platform for mixed-integer non-linear programming problems. *Kybernetes. The International Journal of Systems and Cybernetics*. **36**:5/6, 652-670. (EMERALD Highly Commended Award winning article).
- Östermark, R (2008): Scalability of the Genetic Hybrid Algorithm on a parallel supercomputer. *Kybernetes. The International Journal of Systems and Cybernetics*, **37**:9/10, 1492-1507.
- Östermark, R (2009): Geno-mathematical identification of the multi-layer perceptron. *Neural Computing and Applications* **18**:4, 331-344. (<http://www.springerlink.com/openurl.asp?genre=article&id=doi:10.1007/s00521-008-0184-4>).
- Östermark R (2009): Concurrent processing of mixed-integer nonlinear programming problems. *Kybernetes. The International Journal of Systems and Cybernetics*, **38**:6, 966-989. DOI 10.1108/03684920910973180.
- Östermark R (2009): A fuzzy vector valued KNN-algorithm for automatic outlier detection. *Applied Soft Computing* **9**, 1263-1272. (<http://dx.doi.org/10.1016/j.asoc.2009.03.009>)
- Östermark, R (2010): Genetic hybrid tuning of VARMAX and State Space Algorithms. *Soft Computing*, **14**:1, 91-100. (<http://www.springer.com/home?SGWID=0-0-1003-0-0&aqId=550805&checkval=99567>).
- Östermark R (2010): Concurrent processing of heteroskedastic vector-valued mixture density models. *Journal of Applied Statistics* **37**: 9-10, 1637-1659.
- Östermark R (2011): Hedging with options and cardinality constraints in multi-period portfolio management systems. *Kybernetes. The International Journal of Systems and Cybernetics*, **40**:5/6, ISSN: 0368-492X, 703-718.
- Östermark, R (2012): Incorporating asset growth potential and bear market safety switches in international portfolio decisions. *Applied Soft Computing*, **12**, 2538-2549. (<http://dx.doi.org/10.1016/j.asoc.2012.03.052>).
- Östermark, R (2014): A parallel fuzzy GMM-algorithm for approximate VGARCH-modeling with a multi-modal discontinuous merit function. *Fuzzy Sets and Systems*. **253**, 28-43. (DOI 10.1016/j.fss.2014.01.013).
- Östermark, R (2014): Solving difficult mixed integer and disjunctive non-linear problems on single and parallel processors. *Applied Soft Computing* **24**, 385-405. (DOI 10.1016/j.asoc.2014.07.018).

Östermark, R (2015): A parallel algorithm for optimizing the capital structure contingent on maximum value at risk. *Kybernetes. The International Journal of Systems and Cybernetics* **44** No 3, 384-405. (<http://www.emeraldinsight.com/doi/abs/10.1108/K-08-2014-0171>).

Östermark R (2017): Massively parallel processing of recursive multiperiod portfolio models. *European Journal of Operational Research* **259**, 344-366. (<http://dx.doi.org/10.1016/j.ejor.2016.10.009>).

Östermark R (2018): Executive decision support for modern business management. @CSC news, 20.6.2018. <https://www.csc.fi/web/atcsc/-executive-decision-support-for-modern-business-management>.

Lahti A, Östermark R, Kokko K (2019): Optimizing atomic structures through genomathematical programming. *Communications in Computational Physics* **25**:3, pp 911-927.