

**IMPLEMENTASI ALGORITMA *SPIRAL DYNAMICS* DENGAN  
*PARALLEL COMPUTING* UNTUK PERMASALAHAN OPTIMASI  
KONTINU**

**SKRIPSI**

Diajukan untuk Memenuhi Sebagian dari  
Syarat Memperoleh Gelar Sarjana Komputer  
Program Studi Ilmu Komputer



Oleh  
Anshar Alifya Ihsan  
1301451

**PROGRAM STUDI ILMU KOMPUTER  
DEPARTEMEN PENDIDIKAN ILMU KOMPUTER  
FAKULTAS PENDIDIKAN MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS PENDIDIKAN INDONESIA  
BANDUNG  
2020**

**IMPLEMENTASI ALGORITMA *SPIRAL DYNAMICS* DENGAN  
*PARALLEL COMPUTING* UNTUK PERMASALAHAN OPTIMASI  
KONTINU**

Oleh  
Anshar Alifya Ihsan  
NIM 1301451

Sebuah Skripsi yang Diajukan untuk Memenuhi Salah Satu Syarat Memperoleh  
Gelar Sarjana Komputer di Fakultas Pendidikan Matematika dan Ilmu  
Pengetahuan Alam

© Anshar Alifya Ihsan 2020  
Universitas Pendidikan Indonesia  
Agustus 2020

Hak Cipta Dilindungi Undang-Undang  
Skripsi ini tidak boleh diperbanyak seluruhnya atau sebagian, dengan dicetak  
ulang, difoto kopi, atau cara lainnya tanpa izin dari penulis

**ANSHAR ALIFYA IHSAN**

1301451

**IMPLEMENTASI ALGORITMA *SPIRAL DYNAMICS* DENGAN  
*PARALLEL COMPUTING* UNTUK PERMASALAHAN OPTIMASI  
KONTINU**

DISETUJUI DAN DISAHKAN OLEH PEMBIMBING:

Pembimbing I,

**Lala Septem Riza, M.T., Ph.D.**

NIP. 197809262008121001

Pembimbing II,

**Drs. H. Eka Fitrajaya Rahman, M.T.**

NIP. 196402141990031003

Mengetahui,

Ketua Departemen Pendidikan Ilmu Komputer

**Lala Septem Riza, M.T., Ph.D.**

NIP. 197809262008121001

## **PERNYATAAN**

Dengan ini penulis menyatakan bahwa skripsi dengan judul “Algoritma Optimasi Spiral Dynamics Menggunakan Bahasa Pemrograman R Untuk Dna Barcoding” ini beserta seluruh isinya adalah benar-benar karya penulis sendiri. Penulis tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika ilmu yang berlaku dalam masyarakat keilmuan. Atas pernyataan ini, penulis siap menanggung risiko/sanksi apabila di kemudian hari ditemukan adanya pelanggaran etika keilmuan atau ada klaim dari pihak lain terhadap keaslian karya penulis ini.

Bandung, 27 Agustus 2020  
Yang Membuat Pernyataan,

Anshar Alifya Ihsan  
NIM 1301451

# PENGEMBANGAN ALGORITMA *SPIRAL DYNAMICS* UNTUK OPTIMASI DENGAN MENGGUNAKAN *PARALLEL COMPUTING*

Oleh

Anshar Alifya Ihsan — ansharalif@gmail.com

1301451

## ABSTRAK

Penelitian ini mengimplementasikan algoritma *spiral dynamic* digabungkan dengan *parallel computing* untuk melakukan optimasi. Jenis optimasi yang dilakukan adalah optimasi kontinu dimana nilai optimasinya bisa dipecah karena menggunakan bilangan riil. Data yang digunakan pada penelitian ini adalah *benchmark test function* yang merupakan fungsi objektif untuk mengukur kinerja dari algoritma optimasi. *Parallel computing* pada penelitian ini berfungsi untuk mempersingkat atau mengefisienkan waktu eksekusi program dengan cara membagi kerja dari setiap *core* prosesor perangkat komputer pada saat program dijalankan. Hasil dari penelitian akan menunjukkan vektor dari fungsi objektif yang menunjukkan solusi paling optimum beserta dengan waktu eksekusinya. Hasil akhir dikatakan optimum apabila nilai mencapai atau mendekati nilai global optimum dan tidak melewati range. Setiap fungsi objektif memiliki nilai global optimum dan range yang berbeda-beda.

Kata Kunci: *parallel computing, optimization, spiral dynamics, test function, bahasa pemrograman R*

***DEVELOPING SPIRAL DYNAMICS ALGORITHM FOR OPTIMIZATION  
USING PARALLEL COMPUTING***

*Arranged by*

*Anshar Alifya Ihsan — ansharalif@gmail.com*

*1301451*

***ABSTRACT***

This study implements spiral dynamic algorithm combined with parallel computing to perform optimization. The type of optimization that is carried out is continuous optimization where the optimization value can be broken down because it uses real numbers. The data used in this study is a benchmark test function which is an objective function to measure the performance of the optimization algorithm. Parallel computing in this study functions to shorten or streamline program execution time by dividing the work of each computer device processor core at the time the program is run. The results of the study will show the vector of the objective function which shows the most optimum solution along with its execution time. The final result is said to be optimum if the values reach or approaches the optimum global value and does not cross the range. Each objective function has an optimum global value and a different range.

*Keyword: parallel computing, optimization, spiral dynamics, test function, R programming*

## KATA PENGANTAR

Puji dan syukur penulis panjatkan kehadirat Allah SWT, atas kehendak serta rida-Nya-lah penulis dapat menyelesaikan skripsi yang berjudul “Implementasi Algoritma *Spiral Dynamic* dengan *Parallel Computing* untuk Permasalahan Optimasi” ini. Skripsi ini ditulis untuk memenuhi sebagian syarat dalam meraih gelar sarjana komputer di Program Studi Ilmu Komputer, Fakultas Pendidikan Matematika dan Ilmu Pengetahuan Alam, Universitas Pendidikan Indonesia.

Selesainya skripsi ini tak lepas dari bantuan banyak pihak. Untuk itu penulis mengucapkan banyak terima kasih untuk pihak-pihak terkait yang telah membantu dan mendukung penulis selama proses pembuatan skripsi.

Akhir kata penulis sampaikan permohonan maaf atas segala kesalahan dan kekurangan dalam skripsi ini. Penulis sangat mengharapkan kritik dan saran yang dapat membangun dari pembaca. Semoga skripsi ini dapat bermanfaat bagi yang membacanya dan juga kemajuan pendidikan, ilmu pengetahuan serta teknologi.

Bandung, 27 Agustus 2020

Penulis

## UCAPAN TERIMA KASIH

Alhamdulillahirabbilalamin, puji serta syukur kehadiran Allah SWT. Yang telah melimpahkan rahmat serta hidayah-Nya sehingga penulis diberikan kelancaran dan kemudahan dalam menyelesaikan penulisan skripsi ini. Dalam proses menyelesaikan penelitian dan penyusunan skripsi ini, penulis banyak mendapat bimbingan, dorongan, serta bantuan dari berbagai pihak. Oleh karena itu, pada kesempatan ini peneliti mengucapkan terimakasih serta penghargaan yang setinggi-tingginya, kepada:

1. Bapak Lala Septem Riza, M.T., Ph.D. selaku Kepala Departemen Pendidikan Ilmu Komputer FPMIPA Universitas Pendidikan Indonesia dan pembimbing I yang telah telah memberikan ilmunya kepada penulis hingga skripsi ini dapat diselesaikan .
2. Ibu Dr. Rani Megasari, M.T., selaku Ketua Program Studi Ilmu Komputer dan Ibu Enjun Junaeti, S.Si., M.Si. selaku Ketua Program Studi Pendidikan Ilmu Komputer FPMIPA Universitas Pendidikan Indonesia.
3. Bapak Eka Fitrajaya Rahman, M.T. selaku pembimbing II yang telah banyak membantu memberikan masukan dan saran dalam penyelesaian skripsi ini.
4. Bapak Herbert Siregar, M.T. selaku dosen pembimbing akademik yang telah memberikan arahan dan bimbingan selama penulis menjalani perkuliahan
5. Bapak dan Ibu Dosen Prodi Pendidikan Ilmu Komputer dan Ilmu Komputer yang telah memberikan ilmu dan pelajaran yang sangat bermanfaat dan berguna kepada penulis.
6. Kedua orang tua, Kusnandi dan Yeni Nuraeni serta adik-adik Egi Ghifari Maulana, Audya Fathana Zahira, Andien Putri Maharani, Kyran Rayhana Ramadhani, Farris Ahmad Arkhan, dan Farras Ahmad Azzam yang selalu memberikan doa dan dukungan, baik dukungan secara moral, dukungan materil maupun secara spiritual sehingga dapat mendorong penulis menyelesaikan skripsi ini.
7. Taufik Rizki Sulaksana, Reyhan Maulana, Yupa Umigi, Naufal Lingga Ibrahim, Muhammad Irfan Firmansyah, Panji Nugroho, yang telah memberikan dukungan, memberikan pengalaman, pelajaran, nasihat dan banyak ilmu kepada penulis.



8. Teman seperjuangan mahasiswa Ilmu Komputer FPMIPA UPI 2013. Khususnya untuk teman-teman C1-2013 yang telah berjuang bersama dari awal perkuliahan hingga saat ini.

Semoga semua amal baik yang telah diberikan kepada penulis dapat bermanfaat dan mendapatkan balasan yang berlipat ganda dari Allah SWT. Aamiin.

Bandung, 27 Agustus 2020

Anshar Alifya Ihsan

## DAFTAR ISI

<b>ABSTRAK</b> .....	5
<b>ABSTRACT</b> .....	6
<b>KATA PENGANTAR</b> .....	7
DAFTAR ISI .....	10
DAFTAR GAMBAR .....	13
DAFTAR TABEL .....	14
<b>BAB I PENDAHULUAN</b> .....	15
1.1. Latar Belakang .....	15
1.2. Rumusan Masalah .....	18
1.3. Batasan Masalah .....	18
1.4. Tujuan Penelitian .....	18
1.5. Manfaat Penelitian .....	18
1.6. Sistematika Penelitian .....	18
<b>BAB II KAJIAN PUSTAKA</b> .....	20
2.1. Pengertian Optimasi .....	20
2.1.1. Mathematical Programming .....	22
2.1.2. Combinatorial Optimization .....	22
2.1.3. Algoritma Optimasi .....	22
2.1.4. Optimasi Metaheuristik .....	23
2.1.5. Karakteristik Metaheuristik .....	24
2.1.6. <i>Test Function</i> untuk Optimasi .....	24
2.2. Algoritma <i>Spiral Dynamic</i> Untuk Optimasi .....	26
2.2.1. Contoh Perhitungan <i>Spiral Dynamic Optimization</i> Model 2 Dimensi .....	28
2.2.2. Contoh Perhitungan <i>Spiral Dynamic Optimization</i> Model n Dimensi .....	34
2.3. Bahasa Pemrograman R .....	46
2.3.1. Instalasi Bahasa R pada Sistem Operasi Windows .....	47
2.3.2. Rstudio .....	48
2.3.3. <i>Layout</i> Rstudio .....	49
2.3.4. Sintaksis Dasar dalam Bahasa Pemrograman R .....	53

2.3.5. Tipe Data dalam Bahasa R .....	54
2.3.6. Operator dalam Bahasa R.....	58
2.3.7. Perulangan dalam Bahasa R.....	64
2.3.8. Fungsi dalam Bahasa R.....	66
2.3.9. <i>Packages</i> dalam Bahasa R.....	69
2.4. Parallel Computing.....	75
2.4.1. <i>Bit-level Parallelism</i> .....	76
2.4.2. <i>Instruction-level Parallelism</i> .....	77
2.4.3. <i>Task-level Parallelism</i> .....	79
2.4.4. <i>Superword-level Parallelism</i> .....	80
2.5. <i>Parallel Computing</i> dalam Bahasa R .....	81
2.5.1. Penjelasan <i>Package</i> “foreach” .....	83
2.5.2. Penggunaan “foreach” dan “doParallel” dalam Bahasa R .....	84
BAB III METODE PENELITIAN .....	88
3.1. Desain Penelitian .....	88
3.2. Metode Penelitian.....	92
3.2.1. Metode Pengumpulan Data .....	92
3.2.2. Metode Pengembangan Perangkat Lunak .....	92
3.3. Alat Penelitian dan Bahan Penelitian .....	93
3.3.1. Alat Penelitian .....	93
3.3.2. Bahan Penelitian.....	93
BAB IV HASIL DAN PEMBAHASAN .....	94
4.1 Pengumpulan Data .....	94
4.2. Pengembangan Perangkat Lunak .....	95
4.2.1 Analisis.....	95
4.2.2. Perancangan Parameter Untuk Setiap Fungsi .....	96
4.2.3. Desain.....	100
4.2.3. Implementasi .....	103
4.3. Skenario Eksperimen.....	112
4.4. Hasil Eksperimen .....	113
4.4.1. Hasil Ekseperimen <i>Standalone</i> .....	113

4.4.2. Hasil Ekseperimen 2 <i>Core</i> .....	114
4.4.3. Hasil Ekseperimen 3 <i>Core</i> .....	115
4.4.4. Hasil Ekseperimen 4 <i>Core</i> .....	116
4.5. Pembahasan Hasil Eksperimen .....	117
4.5.1 Hasil Solusi Optimum <i>Test Function</i> .....	117
4.5.2. Perbandingan Hasil Solusi Optimum Spiral Dynamic dengan Algoritma Optimasi Lainnya.....	118
4.5.3. Perbandingan Waktu Eksekusi untuk Mencapai Solusi Optimum ....	119
4.5.4 Hasil Solusi Optimum Test Function Per Skenario .....	121
BAB V KESIMPULAN DAN SARAN .....	123
5.1. Kesimpulan.....	123
5.2. Saran .....	123
DAFTAR PUSTAKA .....	125
LAMPIRAN.....	129

## DAFTAR GAMBAR

Gambar 2.1 Travel Salesman Problem.....	21
Gambar 2.2 Fungsi nilai fitness SDA untuk (a) langkah 1 (b) langkah 2 (Nasir, Tokhi, & Ghani, 2014). .....	27
Gambar 2.3 Logaritma Spiral (Tamura & Yasuda, 2011).....	27
Gambar 2.4 Halaman Unduh Bahasa R .....	47
Gambar 2.5 Tampilan RGui pada bahasa pemrograman R.....	48
Gambar 2.6 Layout Rstudio .....	49
Gambar 2.7 Panel Source Editor .....	49
Gambar 2.8 Panel Environment(dan History, Connections).....	50
Gambar 2.9 Panel Console .....	50
Gambar 2.10 Panel Plots .....	51
Gambar 2.11 Flowchart Loop Statement .....	65
Gambar 2.12 Ilustrasi Proses Komputasi Paralel (Barney, 2007).....	75
Gambar 2.13 Prosesor tanpa pipeline (Culler, Singh, & Gupta, 1997).....	77
Gambar 2.14 Prosesor dengan 5 tahapan pipeline (Culler, Singh, & Gupta, 1997) .....	78
Gambar 2.15 Prosesor superscalar dengan 5 tahapan pipeline (Culler, Singh, & Gupta, 1997).....	78
Gambar 2.16 Halaman task view pada website CRAN (CRAN, 2020).....	82
Gambar 2.17 Pengelompokan package HPC (Riza, Ashari, & Megasari, 2018)..	82
Gambar 3.1 Desain Penelitian .....	88
Gambar 4.1 Desain Eksperimen Standalone .....	101
Gambar 4.2 Desain Eksperimen Paralel.....	102
Gambar 4.3 Waktu yang Diperlukan Untuk Mencapai Solusi Optimum Pada Iterasi 100 dan 50 Populasi Dalam Bentuk Grafik.....	120
Gambar 4.4 Waktu yang Diperlukan Untuk Mencapai Solusi Optimum Pada Iterasi 1000 dan 100 Populasi Dalam Bentuk Grafik.....	121

## DAFTAR TABEL

Tabel 2. 1 Contoh Fungsi Nonlinear .....	21
Tabel 2.2 Contoh Test Function (Yao, Liu, & Lin, 1999) .....	25
Tabel 2. 3 Tipe data vektor atom.....	54
Tabel 2. 4 Operator Aritmatika .....	59
Tabel 2. 5 Operator Relasional.....	60
Tabel 2. 6 Operator Logika .....	61
Tabel 2. 7 Operator Assignment .....	63
Tabel 2. 8 Operator Lainnya .....	63
Tabel 2.9 Perulangan.....	65
Tabel 2.10 Komponen Fungsi .....	67
Tabel 2. 11 Argumen pada “foreach”.....	86
Tabel 3.1 Test Function (Yao, Liu, & Lin, 1999) .....	90
Tabel 4.1 Test Function.....	94
Tabel 4.2 Deskripsi parameter fungsi SDO().....	96
Tabel 4.3 Deskripsi parameter fungsi SDOPar() .....	98
Tabel 4.4 Variasi Eksperimen .....	113
Tabel 4.5 Hasil Eksperimen Standalone .....	114
Tabel 4. 6 Hasil Eksperimen 2 core .....	115
Tabel 4. 7 Hasil Eksperimen 3 core .....	115
Tabel 4.8 Hasil Eksperimen 4 core .....	116
Tabel 4.9 Solusi Optimum Test Function .....	117
Tabel 4. 10 Perbandingan Solusi Optimum Spiral Dynamic dengan Algoritma Optimasi Lainnya.....	118
Tabel 4.11 Waktu yang Diperlukan Untuk Mencapai Solusi Optimum Pada Iterasi 100 dan 50 Populasi .....	119
Tabel 4.12 Waktu yang Diperlukan Untuk Mencapai Solusi Optimum Pada Iterasi 1000 dan 100 Populasi .....	120
Tabel 4. 13 Hasil Solusi Optimum Test Function Per Skenario .....	121

## DAFTAR PUSTAKA

- Adve, S. V., Adve, V. S., & Frank, M. I. (2008, November 1). *Site: Graphics and Data Visualization at the University of Illinois*. Retrieved from Graphics and Data Visualization at the University of Illinois: [graphics.cs.illinois.edu/](http://graphics.cs.illinois.edu/)
- Aiyoshi, E., & Yasuda, K. (2007). *Metaheuristics and Their Applications*. Ohmsha.
- Alzola, C., & Harrell, F. (2006). *An Introduction to S and The Hmisc and Design Libraries*.
- Applegate, D., Chvátal, V., Bixby, R. E., & Cook, W. J. (2006). *The Traveling Salesman Problem: A Computational Study*. New Jersey: Princeton University Press.
- Back, T. (1995). *Evolutionary algorithms in theory and practice : evolution strategies, evolutionary programming, genetic algorithms*. Oxford: Oxford University Press.
- Barney, B. (2007, September 13). *Introduction to Parallel Computing*. Retrieved from Lawrence Livermore National Laboratory: <https://computing.llnl.gov>
- Coban, M. Z., & Mersereau, R. M. (1998). A Fast Exhaustive Search Algorithm for Rate-Constrained Motion Estimation. *IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 7, NO. 5* (pp. 769-773). IEEE.
- Conn, A. R., Scheinberg, K., & Toint, P. L. (1997). Recent progress in unconstrained nonlinear optimization without derivatives. *Mathematical Programming* , 397-414.
- CRAN. (2020, August 4). *CRAN Task Views*. Retrieved from The R Project for Statistical Computing: <https://cran.r-project.org>
- Culler, D., Singh, J. P., & Gupta, A. (1997). Parallel Computer Architecture: A Hardware/Software Approach. In D. Culler, J. P. Singh, & A. Gupta, *Parallel Computer Architecture: A Hardware/Software Approach* (p. 15). San Fransisco: Morgan Kauffman Publisher Inc.
- Dorigo, M., & Stutzle, T. (2004). *Ant Colony Optimization*. The MIT Press.
- ElKady, S. K., & Abdelsalam, H. M. (2015). A simulation-based optimization approach for healthcare facility location allocation decision. *Science and Information Conference* (pp. 500-505). London: IEEE.
- Fletcher, R. (2000). *Practical Methods of Optimization, Second Edition*. Hoboken: John Wiley & Son, Inc.

- Glaskowsky, P. (2003). *Xelerated's Xtraordinary NPU — World's First 40Gb/s Packet Processor Has 200 CPUs*. Mountain View: Microprocessor Report.
- Goldberg, D. E. (1989). *Genetic Algorithim in Search, Optimization and Machine Learning*. Adison-Wesley.
- Gottlieb, A. A. (1989). *Highly parallel computing*. Redwood City: Benjamin/Cummings.
- Kennedy, J., & Ebenhart, R. C. (1995). Particle Swarm Optimization. *Proc. IEEE Int. Conference on Neural Networks*, 1942-1948.
- Kennedy, J., & Russel, E. (1995). Particle Swarm Optimization. (pp. 1942-1948). Washington: IEEE.
- Larsen, S., & Amarasinghe, S. (2000). Exploiting superword level parallelism with multimedia instruction sets. *SIGPLAN Notices (ACM Special Interest Group on Programming Languages)*, 145-156.
- Li, Z., & Zhang, L. (2011). Carbon Finance Development Status and Development Strategy of Our Country under Low-Carbon Economy. *2011 Fourth International Joint Conference on Computational Sciences and Optimization* (pp. 1150-1154). Yunnan: IEEE.
- Liu, W., Lu, Y., Chen, C., Du, Y., Yan, F., & Xu, S. (2012). UE-Based Optimization for Self-Service Community Healthcare Kiosk. *International Conference on e-Health Networking, Applications and Services (Healthcom)* (pp. 309-312). Beijing: IEEE.
- Lockhart, S. D., & Johnson, C. (1996). *Engineering design communication*. Reading: MA: Addison-Wesley.
- Matloff, N. (2011). *THE ART OF R PROGRAMMING*. San Francisco: William Pollock.
- Merril, C., Custer, R., Daugherty, J., Westrick, M., & Zeng, Y. (2007). Delivering core engineering concepts. . *American Society for Engineering Education Conference Paper*.
- Nasir, A. N., Tokhi, M. O., & Ghani, N. M. (2014). Novel Adaptive Bacteria Foraging Algorithms for Global Optimization. *Applied Computational Intelligence and Soft Computing*, 1-7.
- Patt, Y. (2004, April 29). The Microprocessor Ten Years From Now: What Are The Challenges, How Do We Meet Them? Pittsburgh, Pennsylvania, United State of America.



- Petterson, D. A., Hennessy, J. L., & Goldberg, D. (1996). *Computer architecture : a quantitative approach*. San Fransisco: Morgan Kaufmann Publishers.
- Ponsich, A., Jaimes, A. L., & Coello, C. A. (2013). A Survey on Multiobjective Evolutionary Algorithms for the Solution of the Portfolio Optimization Problem and Other Finance and Economics Applications. *IEEE Transactions on Evolutionary Computation* (pp. 321-344). IEEE.
- Pressman, R. (2001). *Software Engineering: A Practitioner's Approach, Fifth Ed.* New York: McGraw-Hill Book Company.
- Quinn, M. J. (2007). *Parallel programming in C with MPI and openMP (Tata McGraw-Hill ed.)*. New Delhi: Tata McGraw-Hill Pub.
- Reinders, J. (2007, September 10). *Understanding task and data parallelism*. Retrieved from ZDNet: <https://www.zdnet.com/article/understanding-task-and-data-parallelism-3039289129/>
- Riza, L., Ashari, M., & Megasari, R. (2018). The Implementation of Gradient Descent Based Methods Using Parallel Computing in R for Regression Tasks. *International Symposium on Advanced Intelligent Informatics (SAIN)*, 37-42.
- Rubinov, A. (2006). Methods for global optimization of nonsmooth functions with applications. *Applied and computational mathematics*.
- Seo, J.-H., Im, C.-H., Heo, C.-G., Kim, J.-K., Jung, H.-K., & Lee, C.-G. (2006). Multimodal Function Optimization Based on Particle Swarm Optimization. *IEEE TRANSACTIONS ON MAGNETICS, VOL. 42, NO. 4, APRIL 2006* (pp. 1095-1098). Busan: IEEE.
- Sorensen, K., & Glover, F. (2013). Metaheuristics. *In Encyclopedia of operations research and management science* , 960-970.
- Tamura, K., & Yasuda, K. (2011). Spiral Dynamics Inspired Optimization. 1116-1122.
- Versani. (2002). *simpleR - Using R for Introductory Statistics*.
- Weston, S. (2020, August 4). *Using the foreach package*. Retrieved from CRAN: <https://cran.r-project.org/web/packages/foreach/vignettes/foreach.html>
- Yao, X., Liu, Y., & Lin, G. (1999). Evolutionary Programming Made Faster. *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, VOL. 3, NO. 2*, 82-102.