Distributed Parallel Cooperative Coevolutionary Multiobjective Evolutionary Algorithm for Large-Scale Optimization

EE664 : Introduction to Parallel Computing

by

Abhinav Bollam
(150102075)

Saurabh Dhall
(150102060)

Deepak Meghwal
(150108009)

under the guidance of

Dr. Gaurav Trivedi - IIT Guwahati
Sukanta Dey, Meenali Janveja

to the

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI
GUWAHATI - 781039, ASSAM

Contents

Abstract

For Large-Scale Optimization Problems(LSOP) serial implementation of Cooperative Co-evolutionary Multi-objective Evolutionary Algorithm(CC-MOEA)s are used. But the computational time and memory required for serial implementation of this algorithm increases drastically with scale. This problem can be solved using parallelization. The implementation of this algorithm is tested on Deb-Theile-Laumanns-Zitzler(DTLZ) Problem sets.

Chapter 1

Introduction

Multi-Objective Problems(MOPs) based on large number of variables are usually solved using Evolutionary Algorithms. Evolutionary algorithms such as Non-dominated Sorting Algorithms(NSGA-II) and MOEA based Decomposition(MOEA/D) are generally used to solve MOPs. MOEA/D is implemented and performance statistics and results are reported. MOEA/D is further parallelized to run on distributed platforms optimizing computation time.

The Distributed Parallel Cooperative Coevolutionary Multi Objective Evolutionary Algorithm(DPCCMOEA) is based on:

1.1 Variable Property Analysis and Grouping

CCGDE3 and MOEA/DVA are two techniques mostly used for grouping of variables. In Cooperative Coevolutionary Generalized Differential Evolution 3(CCGDE3), fixed grouping is used. In MOEA based on Decision Variable Analyses(MOEA/DVA), there are two type of variable analysis which are described below.

A. Control Variable Analysis

In MOPs, decision variables control the convergence and spread aspect of obtained solutions.Control variable analysis classifies the variables mainly into three types.

1. Distance variables
2. Position variables
3. Mixed variables

The mixed variable and position variable are useful to find the conflict between objective function , while distance variable does direct impact on the convergence of evolutionary population.

B. Interdependent Analysis Between Two Decision Variable

The Interdependence analysis between two decision variables is utilized to break down the association connection between two decision variables.
This analysis states that if two variables interact with each other then the partial differential of an objective function with respect to one variable depends on other variable.

1.2 Differential Evolution

Differential Evolution is an evolutionary algorithm which can be applied to optimization of multi-objective problems. DE solves a problem by initializing a population P by randomly assigning each parameter from a prescribed range which can be stored in two D-dimensional vectors for upper (b_U) and lower (b_L) bounds.
The current population is composed of vectors x_i, g where $i = 0$ to $P - 1$ and g is the generation ranging from 0 to g_{max}. Each $x_{i,g}$ is a D-dimensional vector with parameters $x_{j,i,g}$ with $j = 0$ to $D - 1$.
A random number generator gives a uniformly distributed random number rand(0, 1) between 0 and 1. Then the parameters can be initialized as:

$$x_{j,i,0} = rand_j(0, 1).(b_{j,U} - b_{j,L}) + b_{j,L} \tag{1.1}$$

where $x_{j,i,0}$ denotes the j^{th} parameter of the i^{th} vector and generation $g = 0$.
DE generates new vectors by adding the scaled difference of two random distinct vectors to another random distinct vector. This process is termed as mutation.

$$v_{i,g} = x_{r0,g} + F.(x_{r1,g} - x_{r2,g}) \tag{1.2}$$

The scaling factor $F \in (0, 1+)$ controls the rate at which the population evolves.
These new vectors $v_{i,g}$ are termed trial vectors. To further complement mutation, DE integrates the trial vectors with the existing population using crossover.

$$trail_{i,j} = \begin{cases} v_{i,j} & if\ j \in Index \\ x_{i,j} & if\ j \notin Index \wedge r_1 \leq 0.5 \\ x_{b1,j} & if\ j \notin Index \wedge r_1 > 0.5 \wedge r_2 \leq 0.5 \\ x_{b2,j} & otherwise \end{cases} \tag{1.3}$$

where $r1$ and $r2$ are uniform random number between 0 and 1, and b1 and b2 are randomly selected solutions other than i. After this, polynomial mutation is performed on $trail_i$ followed by fitness evaluation.

1.3 Polynomial Mutation

Mutation operators are used for maintaining diversity in a genetic algorithm. Polynomial mutation is mostly used in evolutionary optimization algorithms as a variation operator.

Polynomial mutation operator $\eta_m \in [20, 100]$ is suitable in most of the problem.In this operator, a polynomial probability distribution is used to perturb a solution in a parents vicinity.

The probability distribution in both left and right of a variable esteem is balanced with the goal that no incentive outside the predetermined range [a, b] is made by the mutation operator. For a particular solution $p \in$ [a,b] and mutated solution $p^{'}$ for specific variable is created. For a random number u created in range of [0,1] as figure below.

$$p' = \begin{cases} p + \overline{\delta_L}(p - x_i^{(L)}), & for\ u \le 0.5, \\ p + \overline{\delta_R}(x_i^{(U)} - p), & for\ u > 0.5. \end{cases} \tag{1.4}$$

$$\begin{aligned} \overline{\delta_L} &= (2u)^{1/(1+\eta_m)} - 1, \qquad for\ u \le 0.5, \\ \overline{\delta_R} &= 1 - (2(1-u))^{1/(1+\eta_m)}, \qquad for\ u > 0.5. \end{aligned} \tag{1.5}$$

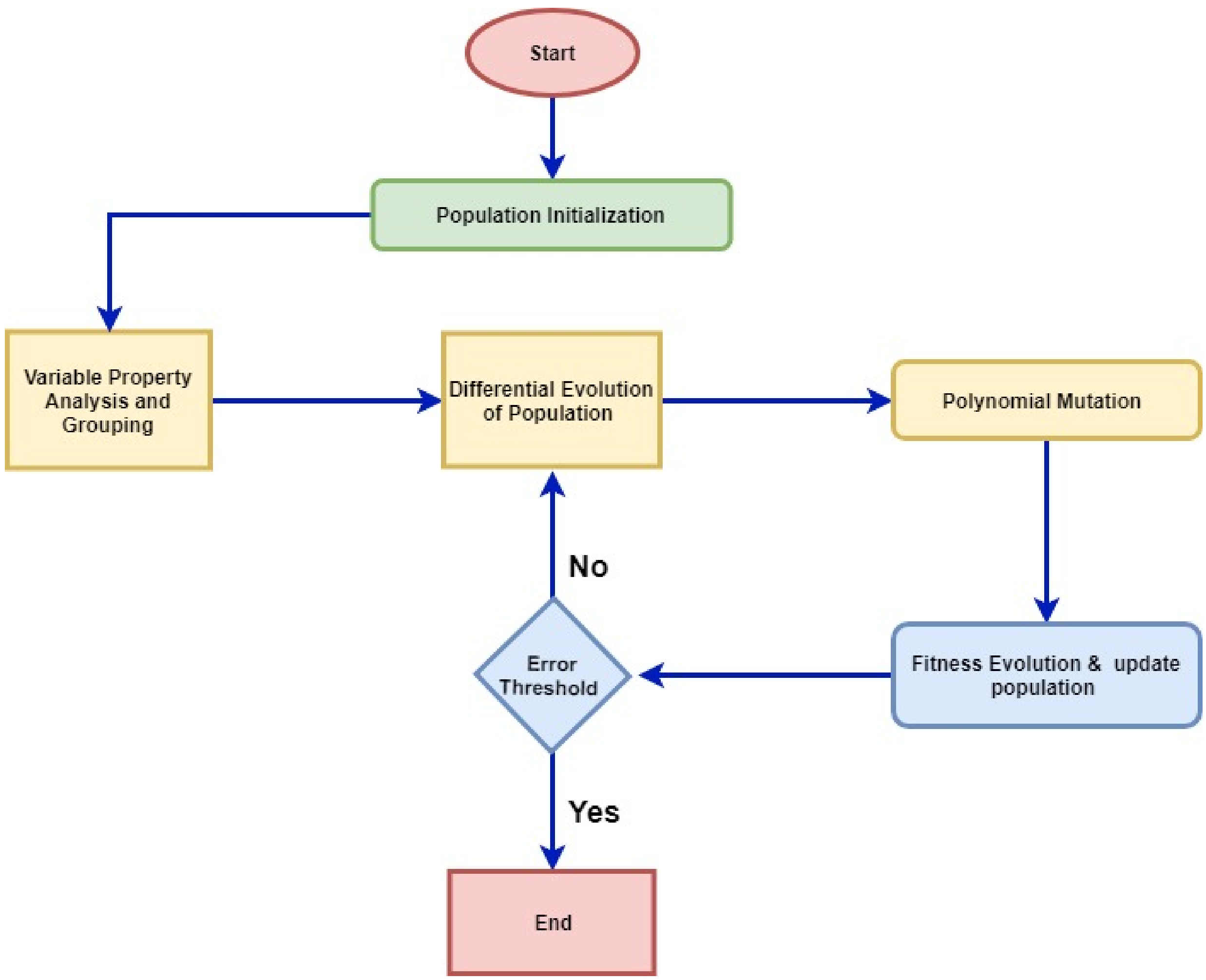
1.4 Fitness Evaluation

Fitness function is used to evaluate the performance of the obtained solution as to how close our solution is to the desired optima.

Chapter 2

Implementation

2.1 Algorithm Flowchart



2.2 Explanation

This report proposes a message passing interface MPI -based distributed parallel cooperative coevolutionary multiobjective evolutionary algorithm (DPCCMOEA).

Algorithm :-

/* Initialization;
Set generation number G=0, set parameters , and allocate memory;

/*Grouping variables;
Analysis and group decision variables as discussed in 1.1;

/* Species Initialization;
Initialize individuals of the sets in each species.;

while $G < G_{MAX}$ **do**

/ Population Evolution;*
Differential evolution();
polynomial mutation() ;

/ Population Updating;*
Fitness function();

G = G+1;

end

Chapter 3

Experiments and Results

DTLZ problem test suite is used for evaluating the DPCCMOEA-LSOP algorithm implemented. DTLZ problem set is used for multi-objective problems with scalable fitness parameters where DTLZ stands for Deb, Thiele, Laumanns and Zitzler.
Deb, Thiele, Laumanns and Zitzler proposed test problems which are used to evaluate the performance of MOEA and compare them against other MOEAs. They are considered a standard to understand the principles and evaluate any MOEA. These test problems are scalable with respect to the number of decision variables and objectives with known pareto-optimal fronts.

3.1 Results

The results for DPCCMOEA-LSOP with Population size 300 and Maximum error 0.0001 are tabulated in the table given below.
The tabulated results show that :

- The time taken for DTLZ 1 to 4 increases gradually. The number of evaluations required to converge also increases.

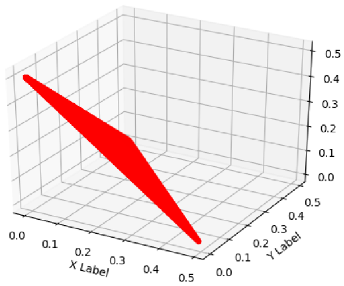
- DTLZ5 and DTLZ6 exhibits similar nature and takes less computation time to converge.

- DTLZ7 with 2 objective functions converges in less number of generations and 3, 5 objective functions take more number of evaluations to converge compared to other DTLZ problem sets.
- DTLZ[1-4] forms concave surface as Pareto Front(PF). DTLZ[5-6] gets 3d curve as PF. DTLZ7 forms multiple PFs.
- The percentage of computation time of Interdependence analysis(serial code) is small compared to overall time.

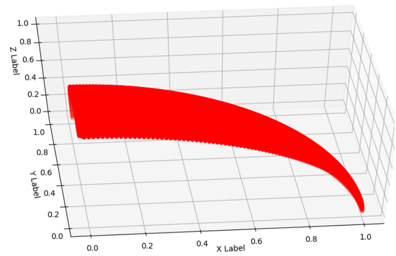
Table 3.1 DPCCMOEA-LSOP with Population size 300 and Maximum error 0.0001

Type of DTLZ	No.of variables	No.of Objective functions	No.of evaluations	Time taken in seconds
DTLZ1	20	2	801	9.03395
		3	1201	16.349
		5	801	9.29606
DTLZ2	20	2	2001	29.8495
		3	2801	44.5869
		5	1801	28.184
DTLZ3	20	2	2401	36.6756
		3	1201	14.2478
		5	1201	16.7836
DTLZ4	20	2	2401	29.9509
		3	3401	53.7689
		5	1201	16.5433
DTLZ5	20	2	401	0.69061
		3	801	9.08874
		5	1001	13.701
DTLZ6	20	2	601	4.66957
		3	601	5.41483
		5	801	9.75401
DTLZ7	20	2	601	2.06806
		3	2201	33.6167
		5	2201	36.6451

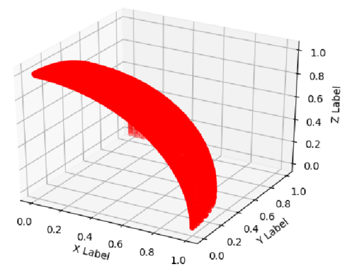
3.2 Pareto front solutions for 3 objective functions



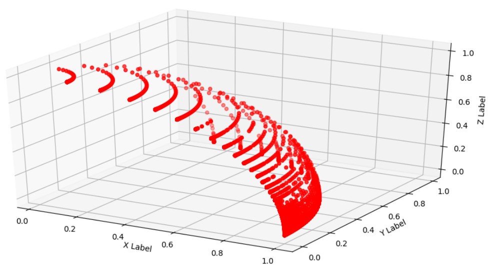
DLTZ 1



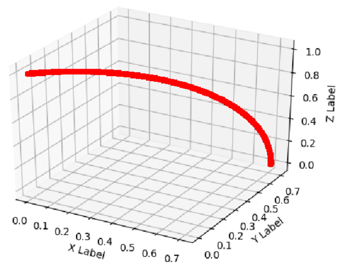
DLTZ 2



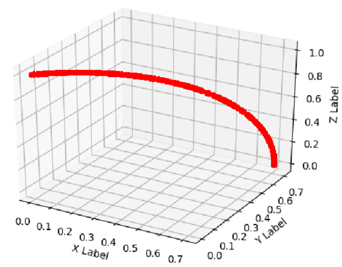
DLTZ 3



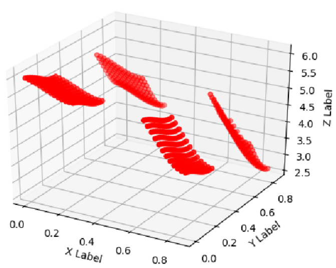
DLTZ 4



DLTZ 5



DLTZ 6



DLTZ 7

Chapter 4

Conclusion and Future Work

Multiobjective Evolutionary Algorithms for large-scale optimization is successfully implemented and tested on DTLZ [1-7] problem sets. The pareto front solutions obtained are shown in Results section. The DTLZ variables have no interaction among them. DTLZ7 contains mixed variables and Distance variables but no postion variables. DTTLZ[1-4] has both position variables and distance variables.

For 3 Objective functions, DTLZ7 forms multiple pareto fronts as solutions and DTLZ[1-4] forms a concave shaped surface as pareto front soution. DTLZ[5-6] forms a curve in 3d as the solution.

4.1 Future work

Message Passing Interface(MPI) based distributed parallel cooperative coevolutionary implementation of this evolutionary algorithm to be done. Control Variable analysis and Grouping, Differential evolution crossover, Polynomial Mutation sections can be parallelized. Interdependence analysis can not be parallelized. For a population set of size 1000, 97.5% of the running time can be parallelized which gives a speed up factor above 200 for DTLZ problem sets.

References

[1] Bin Cao, Jianwei Zhao, Zhihan Lv, and Xin Liu. A distributed parallel cooperative coevolutionary multiobjective evolutionary algorithm for large-scale optimization. *IEEE Transactions on Industrial Informatics*, 13(4):2030–2038, 2017.

[2] Kalyanmoy Deb and Debayan Deb. Analysing mutation schemes for real-parameter genetic algorithms. *International Journal of Artificial Intelligence and Soft Computing*, 4(1):1–28, 2014.

[3] Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. Scalable test problems for evolutionary multiobjective optimization. In *Evolutionary multiobjective optimization*, pages 105–145. Springer, 2005.

[4] Xiaoliang Ma, Fang Liu, Yutao Qi, Xiaodong Wang, Lingling Li, Licheng Jiao, Minglei Yin, and Maoguo Gong. A multiobjective evolutionary algorithm based on decision variable analyses for multiobjective optimization problems with large-scale variables. *IEEE Transactions on Evolutionary Computation*, 20(2):275–298, 2016.

[5] Kenneth Price, Rainer M Storn, and Jouni A Lampinen. *Differential evolution: a practical approach to global optimization*. Springer Science & Business Media, 2006.

[6] Rainer Storn and Kenneth Price. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341–359, 1997.