DISTRIBUTION PACKAGE

PARETO ARCHIVED DYNAMICALLY DIMENSIONED SEARCH (PA-DDS) ALGORITHM FOR SOLVING MULTI OBJECTIVE OPTIMIZATION PROBLEMS (MOPs)

MATLAB PC (WINDOWS) Version 1.3
Works with R2013a and later … we hope
Convex Hull Contribution selection is added in this version

by: Masoud Asadzadeh, January 2015
PhD
Department of Civil & Environmental Engineering
University of Waterloo
200 University Avenue West
Waterloo, Ontario, Canada, N2L 3G1
email: masadzad@uwaterloo.ca OR ma.asadzadeh@gmail.com

REFERENCES FOR THIS ALGORITHM:

- Asadzadeh, M., B. A. Tolson, and D. H. Burn (2014), A new selection metric for multiobjective hydrologic model calibration, Water Resour. Res., 50, 7082–7099, doi:10.1002/2013WR014970.
- Asadzadeh, M. and B. A. Tolson (2013). Pareto archived dynamically dimensioned search with hypervolume-based selection for multi-objective optimization, *Engineering Optimization*. DOI:10.1080/0305215X.2012.748046.
- Asadzadeh M. and B. A. Tolson (2011). Hybrid Pareto archived dynamically dimensioned search for multi-objective combinatorial optimization: application to water distribution network design. *Journal of Hydroinformatics*, 14(1), pp. 192-205.

README_1.3 FILE CONTENTS:

- INTRODUCTION
- DISCLAIMER AND LICENSE INFORMATION
- GENERAL PA-DDS ALGORITHM DESCRIPTION (To be written)
- FILES & SUBDIRECTORIES INCLUDED IN THIS PACKAGE (1.1)
- USER INSTRUCTIONS
- MODIFICATIONS FROM V.1.0 TO V.1.1 TO V.1.2 TO V.1.3

INTRODUCTION:

This file contains the disclaimer and user instructions for the Pareto Archived Dynamically Dimensioned Search (PA-DDS) Algorithm version 1.3 by Masoud Asadzadeh. PA-DDS is an n-dimensional (in decision space), m-dimensional (in objective space), heuristic, stochastic global multi-objective optimization algorithm for box-constrained (bound-constrained) multi-objective optimization problems (MOPs). PA-DDS can optimize continuous, integer or mixed integer problems. PA-DDS is designed to find good Pareto Approximate front quickly (with limited number of solution evaluations) and requires essentially no algorithm parameter tuning.

MATLAB Distribution package 1.3 is a PC (Windows Platform) version of PA-DDS that can be linked with any user-supplied objective function that is coded as an mfile that takes the decision variable values as input and outputs the objective function values.

WHO SHOULD TRY PA-DDS?

- Those who are optimizing computationally expensive Multi-objective Optimization Problems (MOPs) such as spatially distributed environmental simulation model calibration problem (e.g. watershed model) and do not want to spend any time to fine tune the optimization algorithm parameters.
- Convex Hull Contribution option is the best option if uses expect a convex Pareto front, e.g. model calibration problems; otherwise, Hypervolume Contribution option is the best selection metric.
- PA-DDS performance has not yet been thoroughly assessed for optimizing low-dimensional problems (5 or fewer decision variables).
- PA-DDS is not designed to solve quick to evaluate many objective optimization problems with very large number of solution evaluations (10^6 or more) efficiently. Users who solve these problems by PA-DDS may find PA-DDS slows down as the search moves forward, because PA-DDS has an unbounded archive size and uses a complex selection metric based on hypervolume. However, PA-DDS has shown relative high performance compared to other MO algorithms when solving problems with limited number of solution evaluations in the order of 10^4 or fewer.

DISCLAIMER AND LICENSE INFORMATION:

Please report to me any bugs/problems you find with this package. For companies wishing to link this optimization code with an existing code, I am available for some consulting work.

GENERAL PA-DDS ALGORITHM DESCRIPTION:

To Be Written

PA-DDS FILES INCLUDED IN THIS PACKAGE (1.3):

1- bounds.m
2- Crowd_dist.m
3- CHC.m
4- DDS.m
5- HVC.m
6- MainPADDS.m
7- ND_check.m
8- neigh_value_continuous.m
9- neigh_value_discrete.m
10- neigh_value_mixed.m
11- Plot_tradeoff.m
12- read_DDS_inp.m
13- roulette_wheel.m

14- Hypervolume_MEX.mexw32
15- Hypervolume_MEX.mexw64

16- DDS_inp.txt

17- README_1.3.rtf & README_1.3.pdf

OTHER files/Subdirectory:

There is an example (ZDT1) to help the users easily learn how to run PA-DDS and to code their problems:

20- ZDT Folder with ZDT1.m and ZDT1.inp files
21- Example output folder containing a ZDT1_iter1000_HVC subfolder

USER INSTRUCTIONS:

In order to learn how to use this version of the PA-DDS algorithm, an example PA-DDS program application is described below. The instruction below will take you through the Steps for running PA-DDS. You should follow the Steps to run the PA-DDS program for ZDT1. You will be familiar with the PA-DDS program inputs and outputs after working through this example.

ZDT1 Example & Discussion of PA-DDS Inputs:

All you need to do to run this example problem is enter the full directory path for the location of the ZDT1.m file on your PC on line 11 of DDS_inp.txt. Run MainPADDS. If this does not work in your version of MATLAB see point 3.j below. If it does work, skip to point 5 below for output file description.

Discussion below tells you what input value is exactly and how to change it for a new problem. After successful first run of PA-DDS, read carefully below and consider changing some of the inputs, inserting new objective function etc.

1. Open “ZDT1.inp” in the ZDT1 folder with any text editor. For any new problem (e.g. called myfunc.m) in any arbitrary folder (e.g. “myfunc” or ZDT1 folder in this example), a similar file (e.g. with the name myfun.inp) must be created in the same folder. This file contains the name of each decision variable (e.g. 1, 2, … in ZDT1.inp), the lower bound for each decision variable (e.g. 0, 0, … in ZDT1.inp), the upper bound for each decision variable (e.g. 1, 1, … in ZDT1.inp) and a flag for each decision variable to define it as an integer (flag = 1) or continuous (flag = 0) decision variable.

2. Open “ZDT1.m” with MATLAB or any text editor. Similar to this mfile, your objective function should be an mfile that takes a solution (set of decision variables in a row or column vector) as input and returns the objective function variables in a row vector as output. PA-DDS can handle minimization and/or maximization of each objective function. Also, the objective function directory (e.g. ZDT1 folder in this example) can be anywhere on your computer, so you do not need to move it to the PA-DDS directory.

3. Go back to the PA-DDS directory and open the “DDS_inp.txt”. You should modify this file to be able to run PA-DDS for your problem. If you want to run PA-DDS for “ZDT1” example, you only need to modify line 11 in “DDS_inp.txt” to point to the ZDT1 directory. Then run “MainPADDS.m”. The results will be stored in the “ZDT1_iter1000_CVC” subdirectory in the PA-DDS directory. The full directory path or just the subdirectory name can be used on line 11.

If you want to run PA-DDS for your problem (e.g. myfunc.m) you need to make the following changes to the “DDS_inp.txt”:

3.a. In line 3 and instead of ZDT1, name your problem main file (e.g. myfunc) note that “.m” is not required.

3.b. In line 4, name the result directory. PA-DDS will create a sub-directory based on this name in the PA-DDS directory and moves all the results into it.

3.c. In line 5, type the number of independent trials of PA-DDS solving your problem.

3.d. In line 6, type the desired number of solution evaluations per each trial.

3.e. In lines 7 and 8, type two integer numbers to initialize the random number generation of MATLAB.

3.f. In line 9, choose flag 1 to see the evolution of the Pareto front during the search or 0 to avoid plotting the Pareto front at all. Note that plotting progress slows the code down: I observed a factor of 4 increase in computation time for quick to run ZDT1 problem. Plotting flag can be 1 for problems with Up to 6 objective functions where bi-objective sub-spaces of the m-D space will be plotted.

3.g. In line 10, name the file that contains initial solution(s) if you want PA-DDS start with those solutions. PA-DDS assumes that each row of the corresponding file contains one solution. If you have the objective function values for each of these solutions include them in the beginning of each row (e.g. obj1 obj2 … DV1 DV2 …). PA-DDS automatically figures out the objectives and decision variables based on the number of decision variables in the “myfun.inp” for example. If you want PA-DDS starts from randomly generated solutions leave this part empty.

3.h. In line 14, type a flag 1 or -1 for each objective function to be minimized or maximized respectively. Note that the number of flags must be the same as the number of objective functions returned by myfunc.m for example.

3.i. On line 15, is the only algorithm parameter (r) of PA-DDS with the default value of 0.2. In all the previous experiments with PA-DDS, this default value is not changed, and it is not recommended to be fine-tuned due to the acceptable quality of the PA-DDS results for a variety of problems. However, users can fine tune this parameter if they want to assess its effect on the results. If users are looking for best Approximate Pareto front, it is recommended they simply solve multiple trials with r=0.2 and aggregate the fronts.

3.j. In line 16, choose the selection metric from 0: Random, 1: Crowding distance, 2: Hypervolume Contribution, 3: Convex Hull Contribution based selection. Based on my experience option 3 is the best when use expects a convex Pareto front; otherwise, option 2 is the best. However, since not all versions of MATLAB can run “Hypervolume_MEX.mexw32”, “Hypervolume_MEX.mexw64”, or qhull (depending on the operating system), I still include the other options for the users to run PA-DDS. Crowding distance (1) is next best option.

4. Run MainPADDS.m. The result will be stored in a sub-directory in the PA-DDS directory based on the name that you provided in line 4 of “DDS_inp.txt”.

5. Examining PA-DDS algorithm output folder. Open this new subdirectory. If you have only changed line 11 of *DDS_inp.txt* then from the distributed file then the subdirectory is called *ZDT1_iter1000_CVC* and in it you will find:

- copies of the input files used (*DDS_inp.txt* and problem specific decision variable info file, *ZDT1.inp* in this example)
- a screen capture file
- The ****_nondom_sol*.out* output files are for each optimization trial and give the decision variable (DV) values of each final nondominated solution from PA-DDS. Line 1 is values for DV1, DV2, etc. of solution 1. Line 2 is values for DV1, DV2, etc. of solution 2. etc... In this example there are two of these since two optimization trials performed and they are called *ZDT1_nondom_sol1.out* (trial 1) and *ZDT1_nondom_sol2.out* (trial 2)
- The ****_nondom_pts*.out* output files are for each optimization trial and give the objective function values corresponding to each final nondominated solution from PA-DDS. Line 1 is values for objF1, objF2, etc. corresponding to solution 1. Line 2 is values for objF1, objF2, etc. corresponding to

solution 2. etc... In this example there are two of these since two optimization trials performed and they are called *ZDT1_nondom_pts1.out* (trial 1) and *ZDT1_nondom_pts2.out* (trial 2)

- Note that the file pairs (e.g., *ZDT1_nondom_sol1.out* and *ZDT1_nondom_pts1.out)* are intrinsically linked in that the their line numbers correspond to the nondominated solution reference number … Line 1 solution (DV values) yield Line 1 objective function values.

USING PRE-EMPTION CONCEPT w PA-DDS:

To be written. See for example:

Asadzadeh, M., Razavi, S. S., Tolson, B. A., and D. Fay. Pre-emption Strategies for Efficient Multi-Objective Optimization: Application to the Development of Lake Superior Regulation Plan. Accepted Dec. 2013 to *Environmental Modelling and Software*.

MISC. PA-DDS PROGRAM NOTES:

- In the event the program crashes, or is killed early, or the power goes off, etc., the *.mat files are present in the PA-DDS directory AND whatever optimization trials that were previously completed are already saved in this directory. If the program termination is normal these *.mat files will be deleted.
- For solving quick to evaluate problems, users may want to turn off the plotting.
- For simple test functions like ZDT1, this code may run noticeably slower than other MO algorithms applied with the same number of solution evaluations. However, as soon as objective function evaluation time approaches 1 second per solution, algorithm runtime differences become negligible. See for example discussion in Asadzadeh and Tolson (2013) surrounding Figure 6.

PA-DDS for Integer (Discrete Valued) Decision Variable Problems:

I used the earlier version of PA-DDS with the crowding distance selection metric to solve water distribution networks design problems with discrete decision variables (pipe size options). This work is published in the Journal of Hydroinformatics:

Asadzadeh M. and B. A. Tolson (2011). Hybrid Pareto archived dynamically dimensioned search for multi-objective combinatorial optimization: application to water distribution network design. Journal of Hydroinformatics, 14(1), pp. 192-205.

Results show good performance of PA-DDS compared to the benchmark MOEAs.

DDS ALGORITHM MODIFICATIONS/IMPROVEMENTS/VERSIONS:

\- I am currently testing the performance of PA-DDS with a novel selection metric that I designed for PA-DDS when solving MOPs with convex Pareto front.

\- I am also testing the performance of PA-DDS against other MO algorithms that are designed to solve MOPs with limited number of solutions evaluations.

Modifications from version 1.0 to Version 1.1 are listed below:
----

i. The following files are deleted:
   1. cleanInput.m
   2. textread_bt.m
   3. DDS_inout.m

ii. knnsearch in the "Crowd_dist.m" and "HVC.m" files are now changed to the new function "dsearchn" which is embedded in MATLAB.

iii. Changes are made in "Plot_tradeoff.m" function to increase the readability of the Pareto front.

Modifications from version 1.1 to Version 1.2 are listed below:
----

i. MainDDS.m is changed to MainPADDS.m throughout the algorithm code.

ii. Old comments are removed and new comments are added to help interested users understand the code more quickly.

iii. Plot_tradeoff.m is modified to:
   1. Plot up to 6 objective functions in the 2-D sub-plots.
   2. Dock the figure in the MATLAB window to avoid interruptions with any other open window.

iv. Pareto Front figures (if out_plot == 1) are now saved in the output directory.

Modifications from version 1.2 to Version 1.3 are listed below:
----

i. Option 3 is now added in line 16 of "DDS_inp.txt" for Convex Hull Contribution selection metric. Make sure that the Pareto front of the problem of your interest has a known convex Pareto front or is expected to have one; otherwise, HVC (option 2 in line 16 of "DDS_inp.txt") is the selection metric for general problems.
This distribution version of PA-DDS, calls the compiled executable file called qhull.exe in the MS Windows platform. See http://www.qhull.org/ for the source code of qhull.

ii. An error in crowding distance code is fixed. Now, this metric is 1 for all archived solutions in case that number of archived solutions is less than or equal to the number of objective functions + 1.