ES 615 Nature Inspired Computing
Fall 2021-22

Instructor Prof. Nithin George

Assignment -3 Reservoir Design Optimization Using NSGA-II

Name - Devanshu Thakar

Roll no. – 18110174

Problem Statement :

As a Structural Engineer, you are tasked with designing the water reservoir for the water needs of the newly constructed hostel in IITGn.

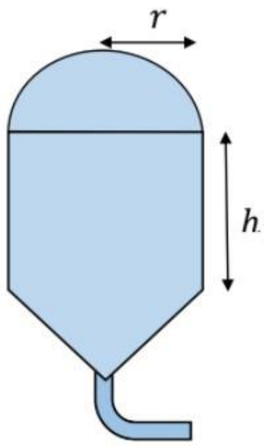
The geometric plan of the reservoir is available in the attached pdf (the connecting pipe is not part of the reservoir). The cost of the design is directly proportional to the surface area of the structure. Your objective as an engineer is to maximize the capacity/volume of the designed reservoir while minimizing the cost.

The limits of the parameters are:

5<= h <= 15

2<= r <= 10

The height of the conical section has to be one-third the height (h) of the cylindrical section.



Use the NSGA-II algorithm to generate the Pareto front for the two objectives.

Show the transitions of the Pareto fronts over iterations in a video.

In a separate pdf file, provide a write up of the design process and your interpretations of the result (what are the extreme designs on the final Pareto front, which design would you choose?)

Solution :

The design process involves the maximization of volume and minimizing area. The two input to the problem are radius r and height h. Two objective functions f_1(area) and f_2 (volume) can be expressed as :

height of cone = $h/3$

Surface Area $f_1(h, r)$= $2\pi r^2 + 2\pi r h + \pi r\sqrt{r^2 + \left(\frac{h}{3}\right)^2}$

Volume $f_2(h, r)$= $\frac{2}{3}\pi r^3 + \pi r^2 h + \frac{1}{3}\pi r^2 \frac{h}{3}$

The range for r and h are given as $5 \leq h \leq 15$ and $2 \leq r \leq 10$

A generic NSGA-II algorithm was implemented in python. NSGA-II is coded using NSGA2 class in python. A class of individual is also created for dealing with the particles.

The NSGA-II algorithm can be briefly described as follow :

1. Generate a set of random solutions, known as particles of individuals.
2. Perform the non-domination sorting over the set of individuals
    a. For sorting individuals inside a front select particles having highest crowding distance.
3. Generate offspring by doing crossover amongst the individuals.
4. Perform mutation on the offspring.
5. Combine the parent and the offspring population.
6. Perform a non-domination sorting on the combined population. Use crowding distance for sorting within a front.
7. Repeat this for the number of total generations.

Initially a random set of real valued solution is generated. To perform non-domination sorting, the fast-non-domination-sorting algorithm as proposed by Kalyanmoy Deb [1] is used in the code. Similarly to compute the crowding distance the algorithm proposed by[1] is used.

A crossover probability (CP) dictates the number of particles to be crossover after doing the non-domination based sorting. To carry crossover between real values, a blend cross-over as described in [2] is used. To perform mutation a random mutation is done. For every offspring to be mutated a random number if generated, if this random number is less the mutation probability (MP) then only the individual is mutated.

Blend crossover

Let x_1 and x_2 be two parent whose crossovers have to be made. Let $x_1 < x_2$. The blend cross-over randomly selects a child in the range $[x_1 - \alpha(x_2 - x_1), x_2 + \alpha(x_2 - x_1)]$. The value of $\alpha = 0.5$ was chosen in this problem.

Random mutation

Random mutation generates a solution randomly within the search space.

Parameter Values

Crossover probability $CP = 1.0$

$\alpha = 0.5$

Mutation probability $MP = 0.1$

Total number of particle = 500

Total generations = 5

The NSGA-II algorithm was run with 500 individuals and for 5 iterations. The evolution of pareto-front with the generation number is shown in video. The code also prints the number of solutions in pareto-front with each generation.

Running the algorithm

From the code it can be seen that the length of the pareto front quickly equals to the total number of particles. The length of pareto front for five generation and the initial random initialization is shown below –

Length of current pareto-front 224 (generation – 0)

Length of current pareto-front 347 (generation – 1)

Length of current pareto-front 500 (generation – 2)

Length of current pareto-front 500 (generation – 3)

Length of current pareto-front 500 (generation – 4)

Length of current pareto-front 500 (generation – 5)

From generation-2 onwards the length of pareto front becomes equal to the number of individuals. Therefore the algorithm attains convergence very quickly.

Results

The minimum and maximum value of area and volume in pareto front of the last generation is shown as follows :

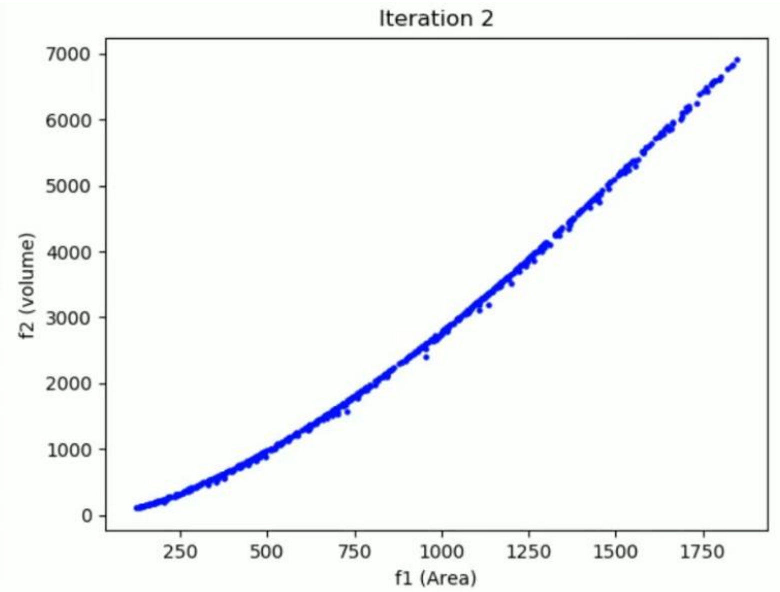
Minimum area f_1 = 121.179 Minimum volume f_2=106.289

For r =2.087 and h = 5.738

Maximum area f_1=1885.9475 Maximum volume f_2= 7138.6366

For r=9.974 and h=14.5727

The pareto front at 2nd generation is shown below :



In the final pareto-front all the possible solutions between the above mentioned extreme values are present. Since the cost is directly proportional to the surface area, based on our budget we can find the surface area of the tank which we could afford. From the surface area of tank we can select the optimal values of r and h from the final pareto front which would maximize volume.

References :

[1] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," in IEEE Transactions on Evolutionary Computation, vol. 6, no. 2, pp. 182-197, April 2002, doi: 10.1109/4235.996017.

[2] https://engineering.purdue.edu/~sudhoff/ee630/Lecture04.pdf