MOMA Optimization

Team Optimistic

Group 40

Mechanical Engineering Department
Indian Institute of Technology Bombay

March 7, 2025

What is Muti Objective Optimization?

In general we are interested in the following mathematical problem type:

$$
\begin{aligned}
\text{minimize/maximize} \quad & f_m(x), m = 1, 2, ..., M \\
\text{subject to} \quad & g_j(x) \geq 0, j = 1, 2, ..., J \\
& h_k(x) = 0, k = 1, 2, ..., K \\
& x_i^{(L)} \leq x_i \leq x_i^{(U)}, i = 1, 2, ..., n
\end{aligned}
$$

A solution is a vector of n decision variables:

$$x = (x_1, x_2, ..., x_n)^T$$

Feasible Solution

A solution that satisfies all constraints and variable bounds. The set of all feasible solutions is called the feasible region, or S

Domination

1. $x^{(1)}$ is strictly better than $x^{(2)}$ in at least one objective
2. $x^{(1)}$ is no worse than $x^{(2)}$ for all objectives

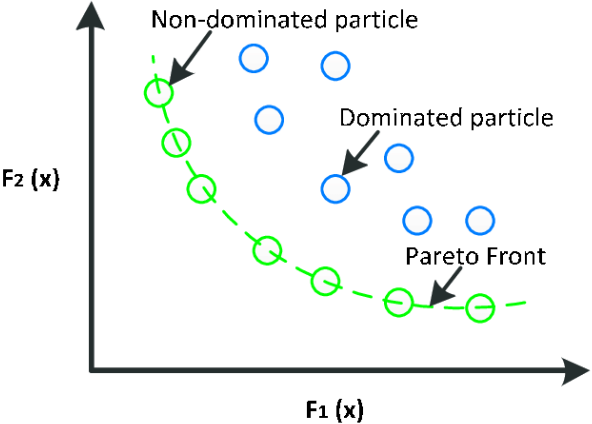
Non-dominated set

Solutions that are not dominated by any member of given set P

Globally Pareto-optimal set

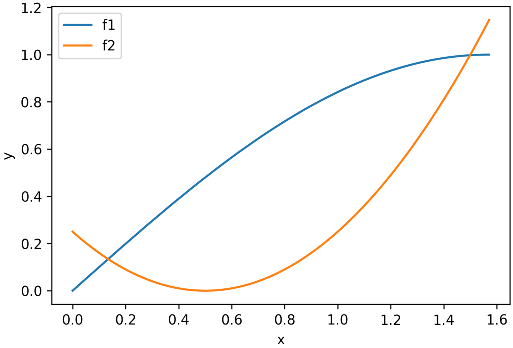
The non-dominated set of the entire feasible search space S is the globally Pareto-optimal set

Pareto Optimal Solutions



Synthetic Experiments

$$f_1 = \sin(x)$$
$$f_2 = (x - 0.5)^2$$
$$x \in [0, \pi/2]$$



Synthetic Experiments

Algorithm 1: Naive Gradient-Based Algorithm

Input: f_1, f_2, η, epochs

$x \sim U(a, b)$

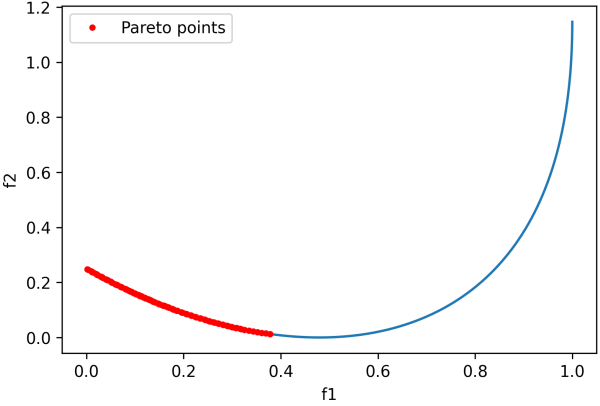
$P \leftarrow \phi$

for $i = 0$ **to** epochs **do**

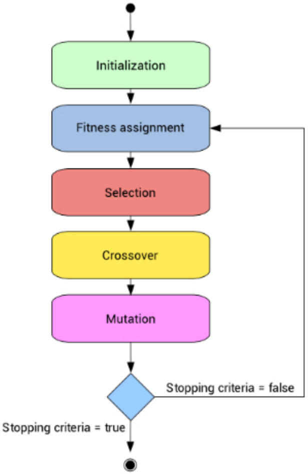
- $x_i = x_{i-1} - \eta \nabla_x f_1(x_{i-1})$
- **if** $f_2(x_i) > f_2(x_{i-1})$ **and** $x_i \notin P$ **and** $x_i \in [a, b]$ **then**
  - $P \leftarrow P \cup \{x_i\}$
- **else**
  - $x_i \sim U(a, b)$

return P

Synthetic Experiments



What are Genetic Algorithms?

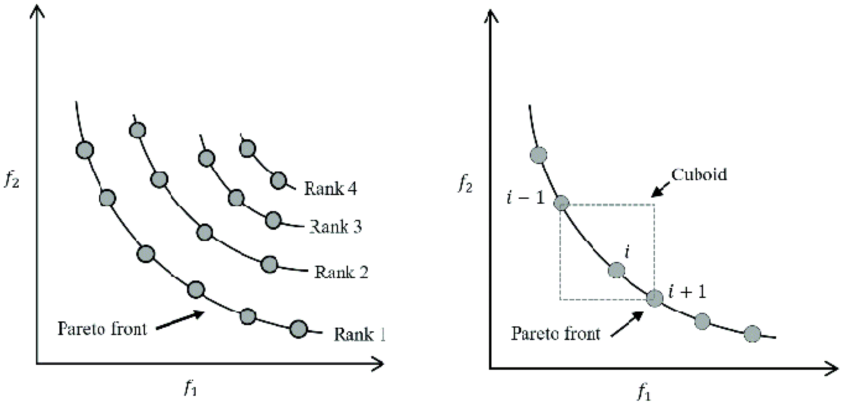


NSGA - II

Nondominated Sorting Genetic Algorithm II

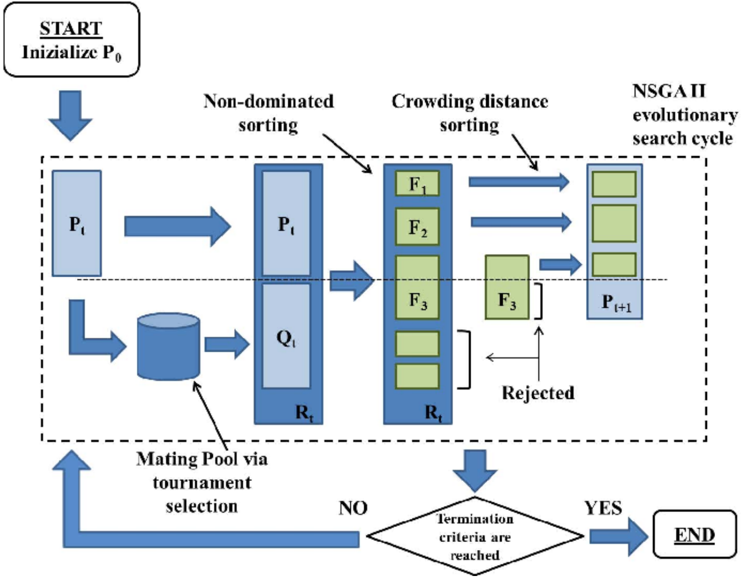
- Emphasis the non-dominated solutions
- Explicit diversity preserving mechanism (Crowding distance)
- Elitist principle

NSGA-II



a) Non-dominated sorting

b) Crowding distance calculation



START
Inizialize P0
Non-dominated sorting
Crowding distance sorting
NSGA II evolutionary search cycle
Pt
Pt
Qt
Rt
F1
F2
F3
F3
Rt
Pt+1
Rejected
Mating Pool via tournament selection
NO
Termination criteria are reached
YES
END

Choosing solutions from a Pareto set

TOPSIS

- Let decision matrix after normalization be denoted as $\mathrm{r_{ij}}$, $\mathrm{i} = 1, 2, 3 \ldots \mathrm{n}$, $\mathrm{j} = 1, 2, 3 \ldots \mathrm{m}$.
- The weighted matrix is calculated as : $\mathrm{v_{ij}} = \mathrm{w_j r_{ij}}$.
- Best solution $\mathrm{A^+} = (\mathrm{v_1^+}, \mathrm{v_2^+}, \ldots, \mathrm{v_m^+}) = \max_{\mathrm{i}}(\mathrm{v_{ij}})$, $\mathrm{i} = 1, 2, 3 \ldots \mathrm{n}$
- Worst solution $\mathrm{A^-} = (\mathrm{v_1^-}, \mathrm{v_2^-}, \ldots, \mathrm{v_m^-}) = \min_{\mathrm{i}}(\mathrm{v_{ij}})$, $\mathrm{i} = 1, 2, 3 \ldots \mathrm{n}$
- Euclidean distance between weighted matrix and the best and worst solutions are $\mathrm{S_i^+} = \sqrt{\sum_{\mathrm{j}=1}^{\mathrm{m}} (\mathrm{v_{ij}} - \mathrm{v_j^+})^2}$ and $\mathrm{S_i^-} = \sqrt{\sum_{\mathrm{j}=1}^{\mathrm{m}} (\mathrm{v_{ij}} - \mathrm{v_j^-})^2}$ respectively.
- Closeness index is calculated as $\frac{\mathrm{S_i^-}}{\mathrm{S_i^-} + \mathrm{S_i^+}}$.

Steps of TOPSIS

1) Normalize the decision matrix

→

2) Multiply the weights with the normalized matrix

→

3) Calculate Euclidean distances between the weighted matrix to the best solution and the worst solution

→

4) Find the point having the highest closeness index

Multi-Agent extension

`Agent 1`: $x_1^1, x_1^2, ...x_1^n$

$f_1^1(x_1^1, x_1^2, ..., x_2^1, ..)$

$f_1^2(x_1^1, x_1^2, ...x_2^1, ..), ...$

$f_1^m(x_1^1, x_1^2, ...x_2^1, ..)$

`Agent 2`: $x_2^1, x_2^2, ...x_2^n$

$f_2^1(x_1^1, ..x_2^1, x_2^2, ..., x_3^1, ..)$

$f_2^2(x_1^1, ..x_2^1, x_2^2, ...x_3^1, ..), ...$

$f_2^m(x_1^1, ..x_2^1, x_2^2, ...x_3^1, ..)$

`... k agents`

`Single Agent`: $x_1^1, x_1^2, .., x_k^n$

$f_1^1(x_1^1, x_1^2, ..., x_2^1, ..), ...$

$f_2^1(x_1^1, ..x_2^1, x_2^2, ..., x_3^1, ..), ...$

$f_k^m(x_1^1, ..x_2^1, x_2^2, ...x_3^1, ..)$

Single agent

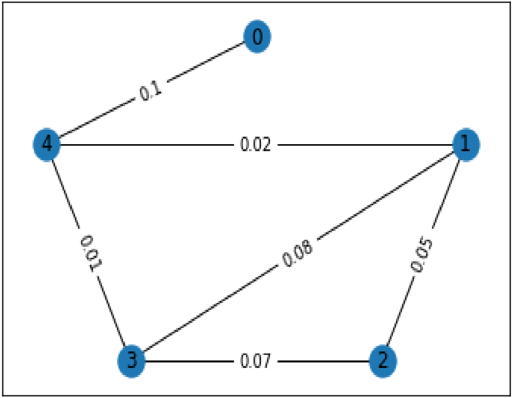
- kn Decision variables
- km Objective functions

Problem Definition

I. Group of k undergrad students, each with m_i objective functions
II. Each student has n decision variables representing the time given to certain activities
III. Further, each student in the group is influenced by some other student(s) in the group.

Here we analyse a group of 5 students with a total of 6 objective functions

Graph



Objective Functions

Table 1: List of objective functions of different students

Student ID	Objective Functions
1	job
2	gradstudy
3	health
4	social
5	explore, social

Decision Variables

We have the following decision variables with the unit of `hours per week`

- Academic Activities (`ac`)
- Sports (`sp`)
- Research Work (`rw`)
- PoRs (`pr`)
- Tech Teams (`tt`)
- Tech Clubs (`tc`)
- Non-Core Clubs (`nc`)
- Culturals (`cu`)
- Leisure (`le`)
- Sleep (`sl`)

Constraints

1. $\texttt{ac + sp + rw + pr + tt + tc + nc + cu + le + sl} \leq 168$
2. $\texttt{ac + sp + rw + pr + tt + tc + nc + cu + le + sl} \geq 120$
3. $\texttt{18} \leq \texttt{ac} \leq 54$
4. $\texttt{pr + tt + tc + nc} \leq 50$
5. $\texttt{sl} \geq 35$
6. $\text{le} \geq 10$

Influence Methods

(1)
$$\lambda_{\mathrm{i}} \leftarrow \lambda_{\mathrm{i}} + \sigma_{\mathrm{ji}}\lambda_{\mathrm{j}}$$

(2)
$$\lambda_{\mathrm{i}} \leftarrow \lambda_{\mathrm{i}} + \sigma_{\mathrm{ji}}(\lambda_{\mathrm{j}} - \lambda_{\mathrm{i}})$$

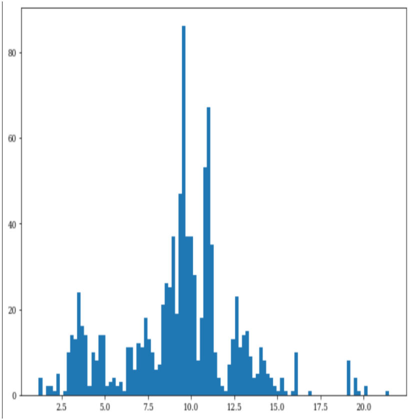
(3)
$$\lambda_{\mathrm{i}} \leftarrow \frac{\lambda_{\mathrm{i}} + \sigma_{\mathrm{ji}}\lambda_{\mathrm{j}}}{1 + \sigma_{\mathrm{ji}}}$$

- Here, $\sigma_{\mathrm{ji}} \in [0, 1]$ is the influence of agent i on j, and λ_{i} is the i^{th} decision variable
- Drawbacks when using Method (1)
- Comparative study between methods (2) and (3)

MaxHist: New Method for Choosing Solutions

Given a pareto set,
For each decision variable,

- Choose its most occurring value that decision variable in the pareto set



Sample Solution

One of the obtained solution is shown below to understand

{1: [job], 2: [gradstudy], 3: [health], 4: [social], 5: [explore, social]}

wtsu_1500_500

Student ID	Acads	Sports	Research	PORs	Tech Teams	Tech Clubs	Non-Core Clubs	Cult	Leisure	Sleep
1	36	2	10	2	0	6	0	1	10	48
2	59	0	57	1	0	0	0	0	10	48
3	18	21	0	2	0	0	0	9	11	49
4	17	0	14	1	0	0	5	1	13	42
5	21	0	0	1	0	3	1	21	14	37

Figure 1: Sample Solution

Experiments Conducted

1. Experimental Analysis for Different Influence Methods
2. Analysis of varying number of generations on the result
3. Choosing Effective Selection Method from Pareto
4. Extending number of objectives to 15
5. Effect of Influence on Optimal Solutions

1. Different Influence Methods

Student ID	Acads	Sports	Research	PORs	Tech Teams	Tech Clubs	Non-Core Clubs	Cult	Leisure	Sleep
1	36	2	10	2	0	6	0	1	10	48
2	59	0	57	1	0	0	0	0	10	48
3	18	21	0	2	0	0	0	9	11	49
4	17	0	14	1	0	0	5	1	13	42
5	21	0	0	1	0	3	1	21	14	37

Figure 2: Method 2 - Influence

Student ID	Acads	Sports	Research	PORs	Tech Teams	Tech Clubs	Non-Core Clubs	Cult	Leisure	Sleep
1	32	2	10	1	11	1	8	4	17	45
2	58	0	57	0	0	0	0	0	10	46
3	18	12	1	4	3	1	7	0	10	49
4	19	24	0	13	0	1	4	10	11	46
5	23	2	3	0	0	4	1	22	15	47

Figure 3: Method 3 - Influence

2. Effect Of Number of Generations

abswtsf_1500_150

Student ID	Acads	Sports	Research	PORs	Tech Teams	Tech Clubs	Non-Core Clubs	Cult	Leisure	Sleep
1	38	10	8	0	3	5	0	12	10	38
2	53	8	7	8	7	1	4	0	11	47
3	21	59	2	5	7	0	2	6	9	48
4	19	15	1	1	9	3	11	22	10	53
5	22	0	5	6	6	9	1	7	10	47

Figure 4: Ngen = 150

abswtsf_1500_1000

Student ID	Acads	Sports	Research	PORs	Tech Teams	Tech Clubs	Non-Core Clubs	Cult	Leisure	Sleep
1	29	1	2	1	0	2	17	26	10	47
2	59	0	59	0	0	0	0	0	10	45
3	23	18	2	0	2	0	2	7	12	51
4	15	1	33	3	0	0	1	5	10	51
5	19	0	0	1	0	1	0	19	14	47

Figure 5: Ngen = 1000

3. Topsis vs MaxHist

wtsu_1500_300_IMP

Student ID	Acads	Sports	Research	PORs	Tech Teams	Tech Clubs	Non-Core Clubs	Cult	Leisure	Sleep
1	53	0	37	1	0	11	1	0	10	48
2	57	0	6	1	3	0	0	4	10	43
3	18	30	2	0	0	6	5	5	11	45
4	17	2	2	3	0	14	6	1	10	42
5	53	0	3	0	2	0	1	10	11	47

Student ID	Acads	Sports	Research	PORs	Tech Teams	Tech Clubs	Non-Core Clubs	Cult	Leisure	Sleep
1	56	0	0	0	0	4	0	0	10	49
2	59	0	47	1	0	0	0	0	10	45
3	18	59	0	0	0	0	0	5	11	49
4	18	0	2	0	0	0	1	59	10	49
5	16	0	0	0	0	0	0	59	10	48

Figure 6: Topsis vs MaxHist

4. More number of Objectives

Student ID	Objective Functions
1	job, health, social
2	gradstudy, explore, social
3	job, health, social
4	social, explore, job, health
5	explore, gradstudy

largeobj_3000_100

Student ID	Acads	Sports	Research	PORs	Tech Teams	Tech Clubs	Non-Core Clubs	Cult	Leisure	Sleep
1	31	9	8	9	3	9	17	11	10	45
2	21	9	7	4	9	9	14	6	11	36
3	27	10	6	10	10	8	11	0	16	46
4	17	18	1	9	2	3	9	9	11	34
5	19	11	15	14	9	10	10	9	16	39

Figure 7: 15 Objective functions

5. Effect Of influence On Optimal Solution

largeobj_3000_100

Student ID	Acads	Sports	Research	PORs	Tech Teams	Tech Clubs	Non-Core Clubs	Cult	Leisure	Sleep
1	31	9	8	9	3	9	17	11	10	45
2	21	9	7	4	9	9	14	6	11	36
3	27	10	6	10	10	8	11	0	16	46
4	17	18	1	9	2	3	9	9	11	34
5	19	11	15	14	9	10	10	9	16	39

Figure 8: Effect Of Influence in Students 1 and 3

Stochasticity

We identify sources of uncertainty in our proposed framework:

- Naive Algo:
  - Random restarts
- NSGA-II:
  - Random mutations in the NSGA-II algorithm
  - Cross-over occurs with probability 0.9

Sensitivity Analysis

Here we analyse sensitivity of our framework with respect to the influence weights.

abswtsinfluence_1000_500

Student ID	Acads	Sports	Research	PORs	Tech Teams	Tech Clubs	Non-Core Clubs	Cult	Leisure	Sleep
1	43	0	2	1	2	1	0	0	9	48
2	59	0	57	0	1	0	0	0	10	47
3	18	13	1	6	1	2	2	1	13	45
4	18	0	9	7	1	0	3	10	11	46
5	17	4	3	0	0	0	5	21	11	47

Figure 9: Results with $\sigma_{25} = 0.02$

abswtsinfluence_1000_500

Student ID	Acads	Sports	Research	PORs	Tech Teams	Tech Clubs	Non-Core Clubs	Cult	Leisure	Sleep
1	31	0	2	4	13	13	6	9	16	38
2	49	0	59	1	0	1	0	1	10	47
3	21	22	1	7	0	0	4	4	16	49
4	31	2	13	4	0	7	2	0	13	46
5	18	0	0	2	7	1	0	35	13	51

Figure 10: Results with $\sigma_{25} = 0.022$

Limitations

- Competitive nature of some objective functions not considered
- The algorithm depends on the number of generations in two ways :
  - If the number of generations is low, then the preferred gene may not appear by mutation and sub optimal solutions may be obtained
  - If the number of generations is high, the the population will be saturated with preferred decision variables

Conclusion

- Pareto Solutions
- Synthetic Experiments, Naive Algorithm
- Genetic algorithms
- Selection from Pareto
- Multi-agent extension
- Insti-Life Problem

- We can take up any queries now
- The following slides contain(for reference) :
  - Appendix A : Some Code Snippets
  - Appendix B : More on Genetic Algorithms

Appendix A: Important Code Snippets

```python
## objective functions

def job(s):
    job = (9*(s.acads)**2 + 4*(s.research)**2 + 3*(s.pors)**2 + 5*(s.tech_team)**2
          +  5*(s.tech_club)**2)*np.exp(-( (s.sleep-49)**2 / ( 2.0 * 14**2 ) ) ) # mu = 49, sigma = 14
    return job

def gradstudy(s):
    gradstudy = (np.exp(-( (s.sleep-49)**2 / ( 2.0 * 14**2 ) ) ))*(8*(s.acads)**2
                    +10*(s.research)**2+5*(s.tech_team)**2+5*(s.tech_club)**2)
    return gradstudy

def health(s):
    health = (np.exp(-( (s.sleep-49)**2 / ( 2.0 * 7**2 ) ) ))*
            (1 + s.sports**2/5 + (1 - np.exp(-s.leisure/15)))
    return health

def social(s):
    social = (np.exp(-( (s.sleep-49)**2 / ( 2.0 * 14**2 ) ) ))*((s.sports**2)
                +7*(s.pors**2)+3*(s.tech_team)**2+3*(s.tech_club)**2+7*(s.nc_club**2)+7*(s.cult**2))
    return social

def explore(s):
    explore = (np.exp(-( (s.sleep-49)**2 / ( 2.0 * 14**2 ) ) ))*((5*(s.sports**2)+5*(s.pors**2)
    +5*(s.tech_team)**2+5*(s.tech_club)**2 +5*(s.nc_club**2)+5*(s.cult**2)+5*(s.acads**2)+5*(s.research**2)))
    return explore
```

Figure 11: Objective Functions

Appendix A: Important Code Snippets

```
from pymoo.algorithms.moo.nsga2 import NSGA2
from pymoo.optimize import minimize
from pymoo.visualization.scatter import Scatter
import matplotlib.pyplot as plt

import time

problem = CollegeLife(student_list, g)

n_var = 50
X = 10*np.ones((1000, n_var))
algorithm = NSGA2(pop_size=1000, sampling=X)

# algorithm = NSGA2(pop_size=2000)

t1 = time.time()
res = minimize(problem,
               algorithm,
               ('n_gen', 150),  ## was 200 ## 120 giving good results
               seed=1,
               verbose=True)
```

Figure 12: PYMOO's NSGA2

Appendix A: Important Code Snippets

```python
def TOPSIS(pareto_matrix, weights):
    normalized_mx = normalize(pareto_matrix)
    weighted_mx = normalized_mx*weights
    best_sol = np.max(weighted_mx, axis=0)
    worst_sol = np.min(weighted_mx, axis=0)
    diff_best = weighted_mx - best_sol
    diff_worst = weighted_mx - worst_sol
    sq_diff_best = np.square(diff_best)
    sq_diff_worst = np.square(diff_worst)
    dist_best = np.sqrt(np.sum(sq_diff_best, axis=1))
    dist_worst = np.sqrt(np.sum(sq_diff_worst, axis=1))
    final_cost = dist_worst/(dist_best + dist_worst)
    ##print(final_cost)
    return np.argmax(final_cost)
```

Figure 13: TOPSIS

Appendix A: Important Code Snippets

```
# This is using max Distribution
ans=np.linspace(1,1,50)
for i in range(0,50):
 n, bins, patches = ax.hist(res.X[:,i], bins = 100)
 max_occ = bins[np.argmax(n)]
 ans[i]=max_occ
ans
```

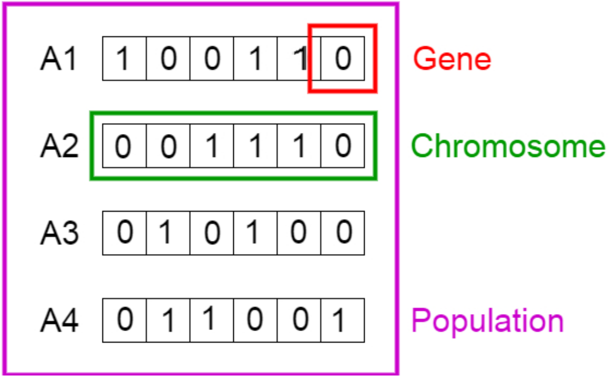
Figure 14: MAXHist

Appendix B: Genetic Algortihms

Example: Knapsack Problem

ITEM	WEIGHT	SURVIVAL POINTS
SLEEPING BAG	15	15
ROPE	3	7
POCKET KNIFE	2	10
TORCH	5	5
BOTTLE	9	8
GLUCOSE	20	17

1. Initialization



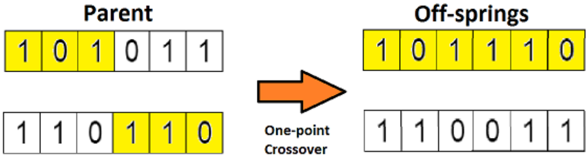
2. Fitness Function

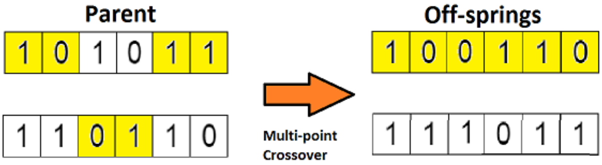
ITEMS	WEIGHT	SURVIVAL POINTS
Sleeping bag	15	15
Torch	5	5
Bottle	9	8
TOTAL	29	28

ITEMS	WEIGHT	SURVIVAL POINTS
Pocket Knife	2	10
Torch	5	5
Bottle	9	8
TOTAL	16	23

3. Selection

4. Crossover





5. Mutation