الگوریتم های ژنتیک چند هدفه

Multi-Objective Genetic Algorithms

استاد: دکتر مصطفی زندیه
دانشگاه شهید بهشتی- دانشکده مدیریت و حسابداری

مفاهیم اولیه

- مقدمه ای بر بهینه یابی چند هدفه
  - تفاوت بین بهینه سازی تک هدفه و بهینه یابی چند هدفه
  - فضای توابع هدف
  - مرز/ جبهه کارا [1]
  - جواب های مغلوب[2] و نامغلوب [3]
  - مجموعه جوابهای پارتو [4]
  - تقریب پارتو [5]

1- Efficient Front 2- Dominated 3- Non-Dominated 4- Pareto 5- Pareto Approximation

روش های حل مسائل بهینه یابی چند هدفه

- جستجو بدون تقدم
  - روش مین – ماکس
    - کمینه سازی فاصله نسبی از جواب ایده آل

$$\begin{aligned} &Min \quad \left[\sum_{i=1}^{m}\left(\frac{f_i(\underline{x})-f_i^*}{f_i^*}\right)^p\right]^{\frac{1}{p}} \\ &s.t: \quad \underline{x}\in S \\ &\qquad 1\le p\le\infty \end{aligned}$$

- ابتدا تصمیم گیری سپس جستجو
  - اولویت بندی اهداف توسط تصمیم گیرنده و یافتن یک یا چند جواب قانع کننده
- جستجو و تصمیم گیری همزمان
  - تصمیم گیرنده در طول فرآیند جستجو دخالت کرده و با تغییر اولویت اهداف، جستجو برای یافتن جواب های مورد نظر ادامه می یابد.
- ابتدا جستجو و سپس انتخاب
  - مجموعه جواب هایی بدست آمده و تصمیم گیرنده براساس ترجیحاتش انتخاب می کند.

1- Enumeration 2- Branch-and-Bound

ابتدا جستجو و سپس انتخاب

- ترکیب وزنی
  - تبدیل اهداف به صورت مجموع وزنی اهداف

$$\begin{aligned} &Min \quad \sum_{i=1}^{m} w_i f_i(\underline{x}) \\ &s.t: \quad \underline{x}\in S \\ &\qquad \sum_{i=1}^{m} w_i = 1 \end{aligned}$$

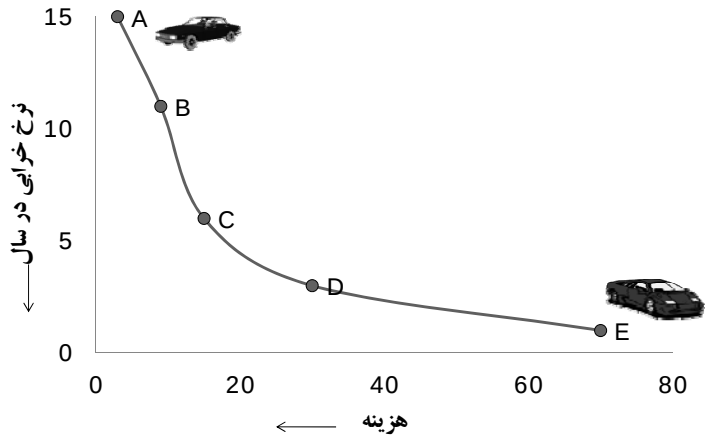
- روش محدودیت
  - بهینه سازی یک تابع هدف و قرار دادن بقیه اهداف در محدودیت ها

$$\begin{aligned} &Min \quad f_k(\underline{x}) \\ &s.t: \quad f_i \le e_i, \quad i\ne k, \ i=1,2,\ldots,m \\ &\qquad \underline{x}\in S \end{aligned}$$

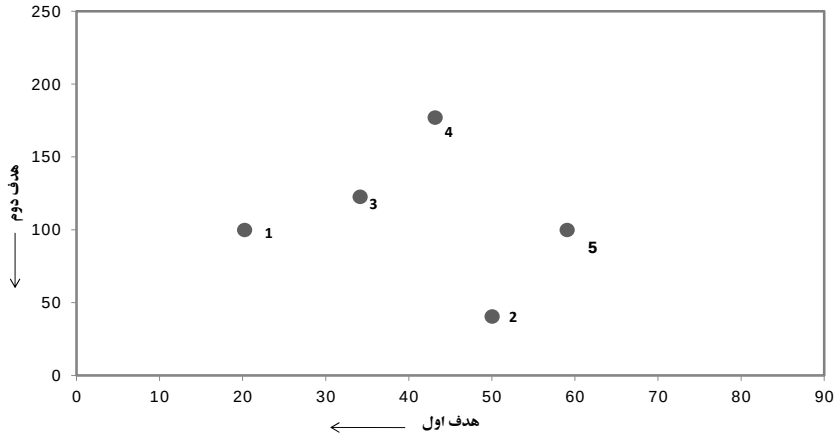
- روشهای مبتنی بر پارتو
  - براساس مفهوم جواب های نامغلوب، مجموعه ای از جواب ها به نام مجموعه ی پارتو ارائه می شود.

1- Enumeration 2- Branch-and-Bound

بهینه یابی چند هدفه

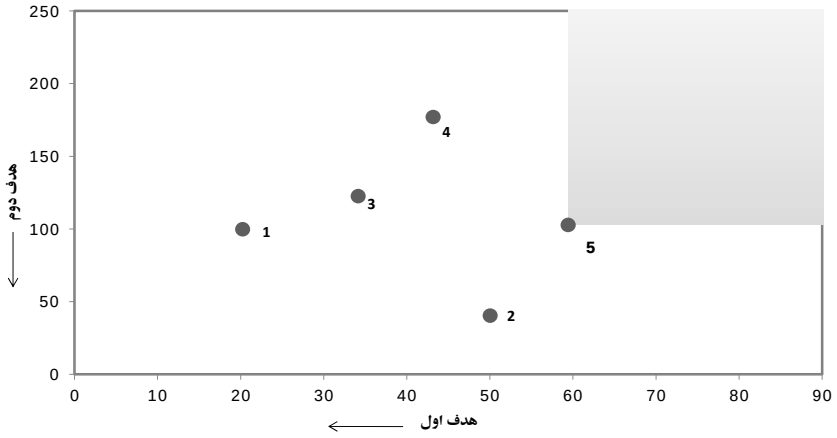


جواب مغلوب و نامغلوب



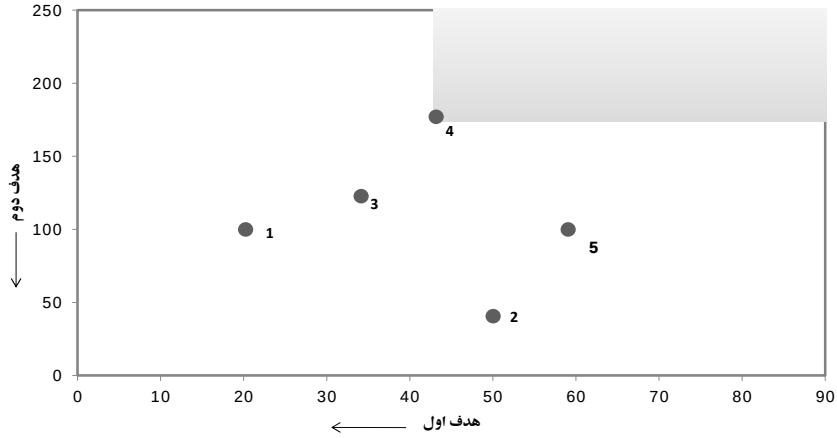
1- Dominated Solution 2- Non-dominated Solution

جواب مغلوب و نامغلوب



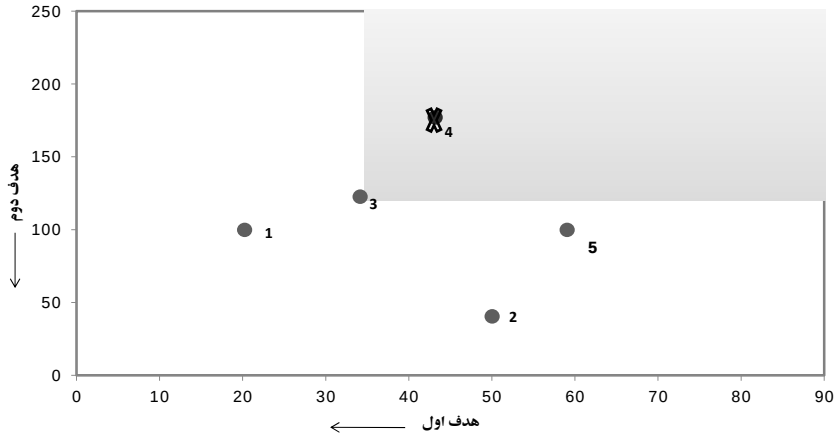
1- Dominated Solution 2- Non-dominated Solution

جواب مغلوب و نامغلوب



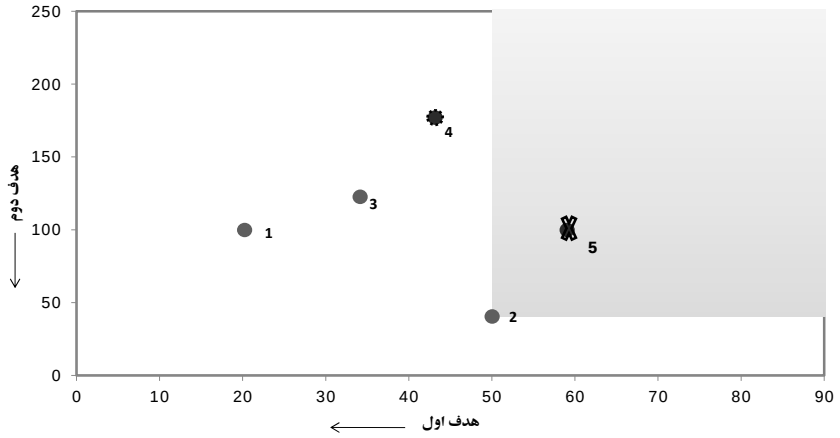
1- Dominated Solution 2- Non-dominated Solution

جواب مغلوب و نامغلوب



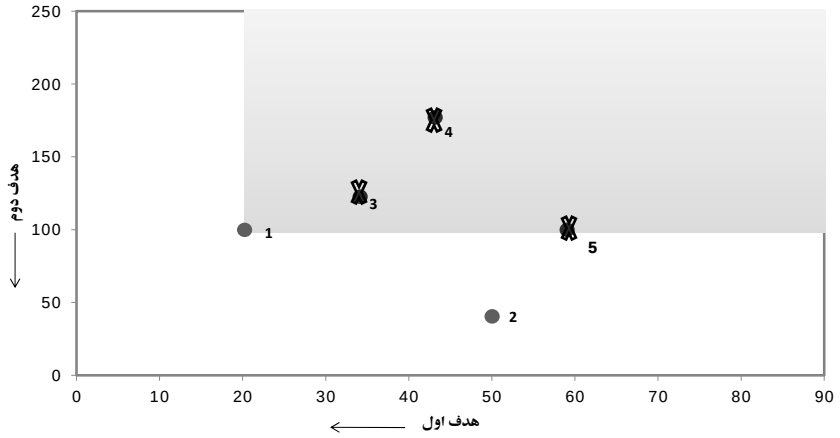
1- Dominated Solution 2- Non-dominated Solution

جواب مغلوب و نامغلوب



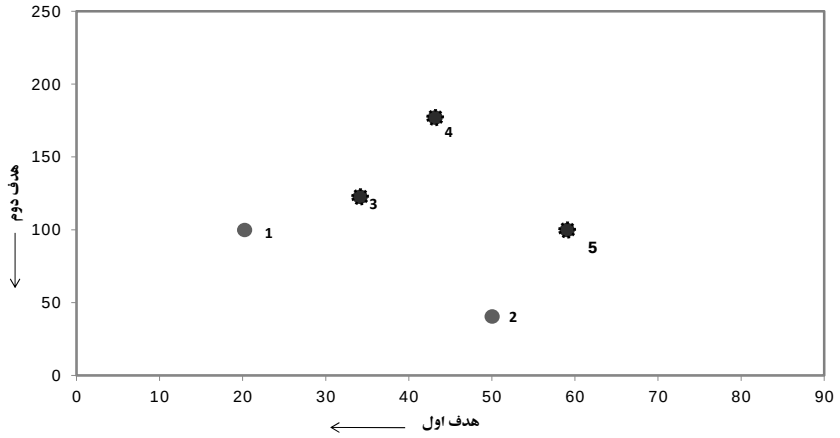
1- Dominated Solution 2- Non-dominated Solution

جواب مغلوب و نامغلوب



1- Dominated Solution 2- Non-dominated Solution

جواب مغلوب و نامغلوب



1- Dominated Solution 2- Non-dominated Solution

الگوریتم های ژنتیک چند هدفه

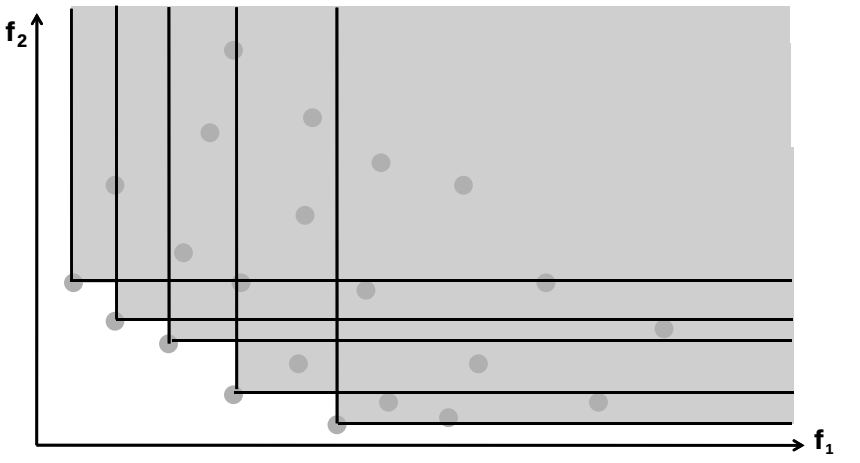
- NSGA-II
- SPGA
- MOGA
- NRGA
- VGA

الگوریتم NSGA-II

- رویکرد سریع مرتب سازی نامغلوب
- تخصیص فاصله ازدحام
- عملگر انتخاب مسابقه ای ازدحام
- رویه الگوریتم

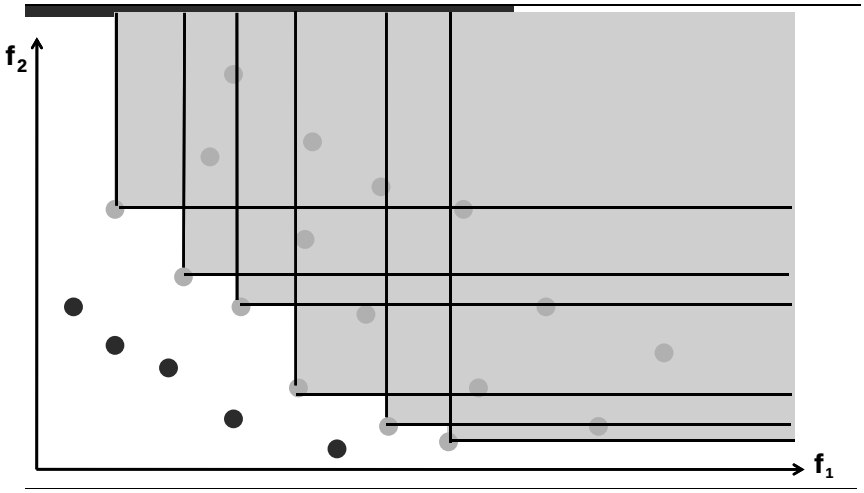
1- Non-dominated Sorting Genetic Algorithm

رویکرد سریع مرتب سازی نامغلوب

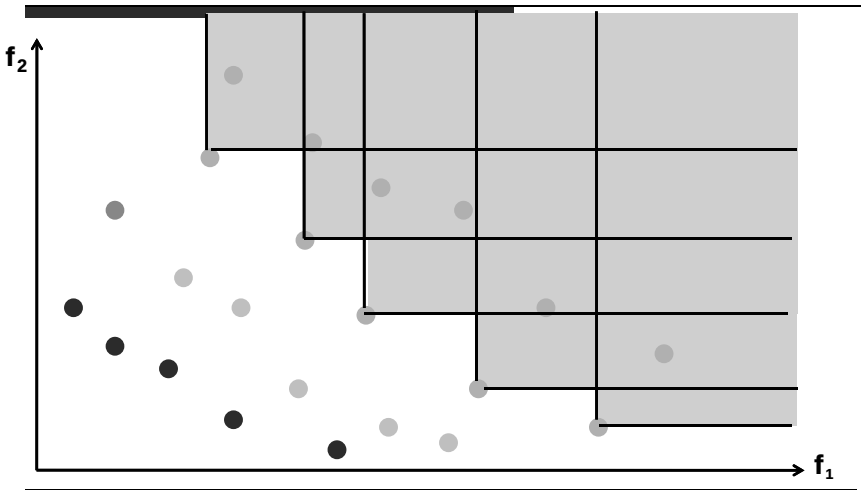


1- Non-dominated Sorting

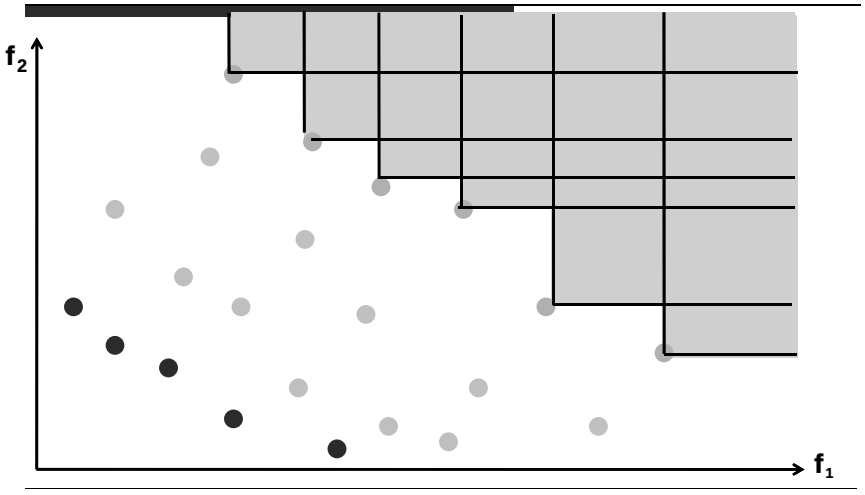
رویکرد سریع مرتب سازی نامغلوب



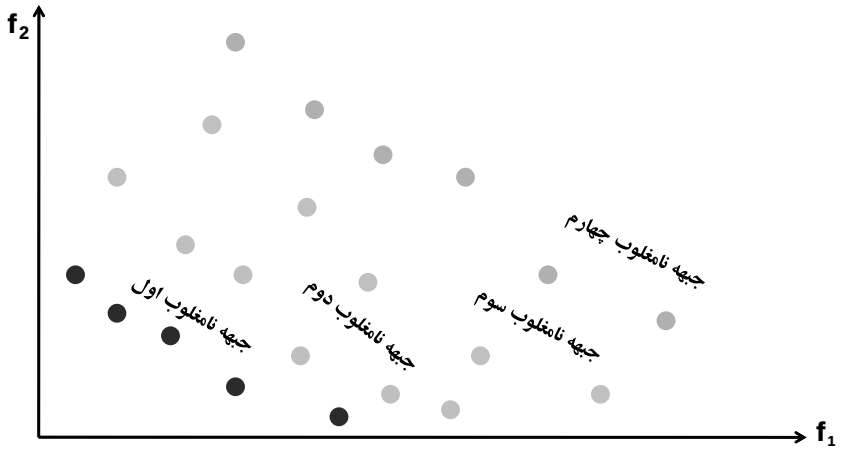
رویکرد سریع مرتب سازی نامغلوب



رویکرد سریع مرتب سازی نامغلوب



رویکرد سریع مرتب سازی نامغلوب

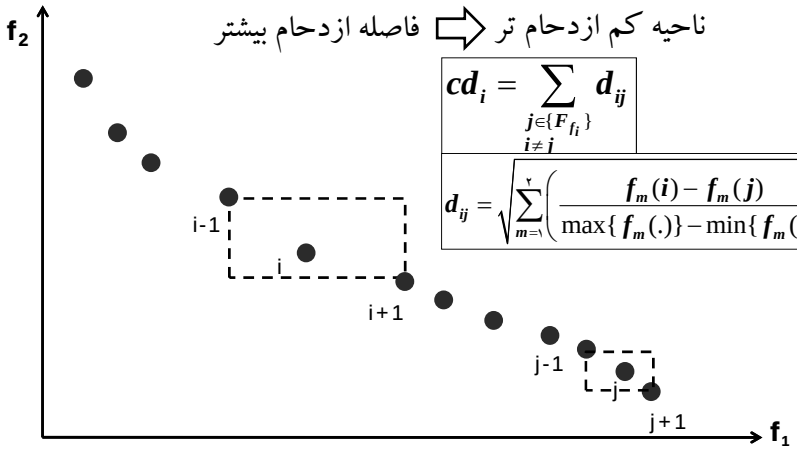


تخصیص فاصله ازدحام

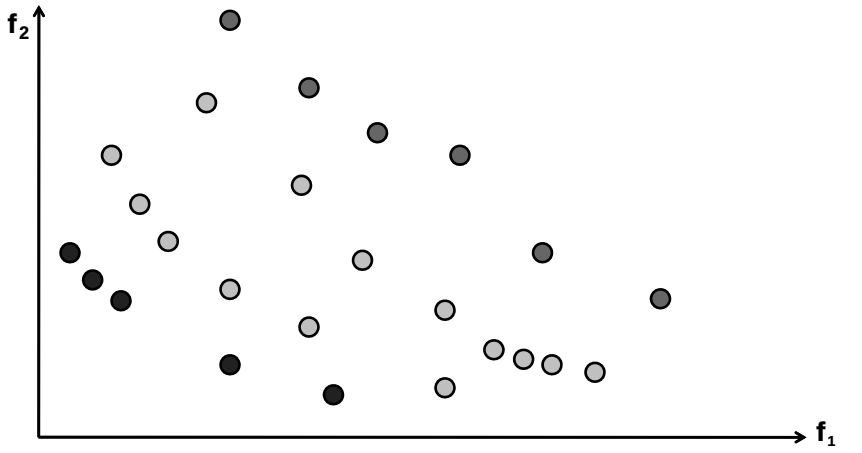
ناحیه کم ازدحام تر ⇨ فاصله ازدحام بیشتر

$$cd_i = \sum_{\substack{j \in \{F_{f_i}\} \\ i \neq j}} d_{ij}$$

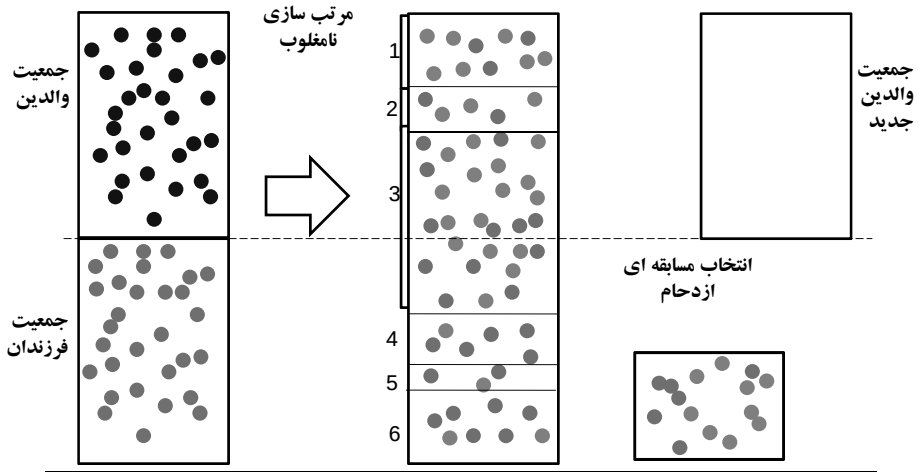
$$d_{ij} = \sqrt{\sum_{m=1}^{2} \left(\frac{f_m(i) - f_m(j)}{\max\{f_m(.)\} - \min\{f_m(.)\}} \right)^2}$$



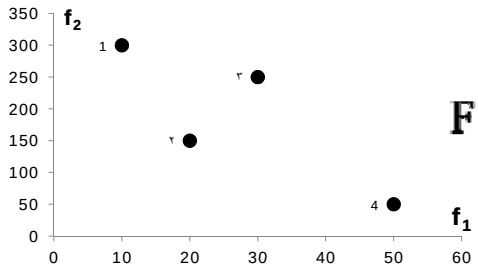
عملگر انتخاب مسابقه ای ازدحام



رویه الگوریتم



الگوریتم SPGA



$$F = W_1 f_1 + W_2 f_2$$

MOGAW

1. **[Initialization]**
   (a) **[Initialize Parameters] Set** *PopSize, Numgen, Pc, Pm, StopCriteria,..*
   (b) **[Initialize Population] Generate** *PopSize* chromosomes, randomly
2. **[Evaluation] Evaluate** the fitness of each chromosome by a weighted sum of objective functions.
3. **[Selection] Select** Parents based on selection strategy. **Selection: Roulette wheel selection, Elitist selection and Purely random selection are employed to reproduce the next generation.**
4. **Crossover operation: Select *PopSize* × *Pc* pairs of parents from current population (Roulette wheel selection), and perform crossover on the parents.**
5. **Mutation operation: Select *PopSize* × *Pm* chromosome from current population (Purely Random Selection), and mutate the individual bits.**
6. **Update Archive: Efficient solutions of current population are copied into an archive that stores the best non-dominated front obtained.**
7. **Generate next generation: Select *np* × (1-*Pm*-*Pc*) solutions from current population and select *PopSize* × (*Pc*+*Pm*) solutions from crossover and mutation operation .**
8. **Stopping criteria: Terminate the algorithm if the stopping criterion is met; else return to Step 2.**

MOGA-C

1. Initialization:
   (a) Parameters setting: Set the population size (*PopSize*), number of generation (*ng*), Probability of crossover and mutation (*Pc, Pm*).
   (b) Initial population generation: Randomly generate initial population.
2. Non-dominated sorting and ranking: The population of solutions is classified into successive non-dominated fronts and computing crowding distance of each solution.
3. Selection: Binary-tournament selection is employed to reproduce the next generation.
4. Crossover and Mutation operation: Select (*PopSize*×*Pc*) / 2 pairs of parents from current population, and perform crossover on the parents. Perform mutation with the probability of *Pm* on the child.
5. Update Archive: All chromosomes on Front 1 (efficient solutions) are copied into an archive that stores the best non-dominated front obtained
6. Generate next generation: Select *PopSize*×*Pmc* solutions from crossover and mutation operation and select *PopSize*×*Pn* solutions from neighborhood search structure.
7. Stopping criteria: Terminate the algorithm if the stopping criterion (number of generation) is met; else return to step 2.

MOGA-T

1. Initialization:
   (a) Parameters setting: Set the number of initial population (*np*), Number of generation (*ng*), Probability of crossover (*Pc*), Probability of mutation (*Pm*), Probability of reproduction (*Pr*).
   (b) Initial population generation: Randomly generate an initial population.
2. Evaluation: Evaluate the fitness function for each of the initial population by considering ideal and negative-ideal solutions (evaluation process carried out by application of the TOPSIS approach).
3. Selection: Roulette wheel selection, Elitist selection and Purely random selection are employed to reproduce the next generation.
4. Crossover operation: Select *np* × *Pc* pairs of parents from current population (Roulette wheel selection), and perform crossover on the parents.
5. Mutation operation: Select *np* × *Pm* chromosome from current population (Purely random selection), and mutate the individual bits.
6. Update Archive: efficient solutions of current population are copied into an archive that stores the best non-dominated front obtained.
7. Generate next generation: Select *np* × *Pr* solutions from current population and select *np* × (*Pc*+*Pm*) solutions from crossover and mutation operation.
8. Stopping criteria: Terminate the algorithm if the stopping criterion (number of generation) is met; else return to step 2.

TOPSIS

- Normalization of both the weights vector and the criteria evaluation vectors

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^{m} x_{ij}^2}} \qquad V = (r_{ij} w_j)$$

- Determined the ideal (A+) and negative-ideal (A-) solutions

$$A^+ = \{(\max_i v_{ij} \mid j \in J), (\min_i v_{ij} \mid j \in J') \mid i = 1,2,\ldots,m \;,\; j = 1,2,\ldots,n\} = \{v_1^+, v_2^+, \ldots, v_n^+\}$$

$$A^- = \{(\min_i v_{ij} \mid j \in J), (\max_i v_{ij} \mid j \in J') \mid i = 1,2,\ldots,m \quad j = 1,2,\ldots,n\} = \{v_1^-, v_2^-, \ldots, v_n^-\}$$

- Calculated the separation distances from the ideal and negative-ideal solution

$$S_i^- = \sqrt{\sum_{j=1}^{n} (v_{ij} - v_j^-)^2} \qquad S_i^+ = \sqrt{\sum_{j=1}^{n} (v_{ij} - v_j^+)^2}$$

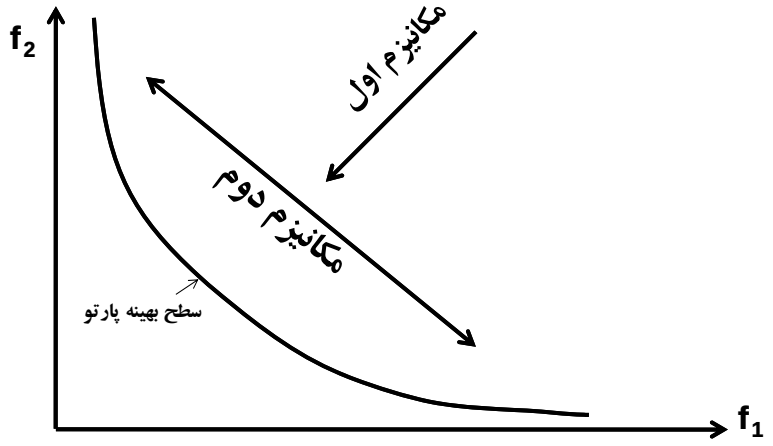
- Finally, the relative closeness to the ideal solution is calculated as follows

$$CL_i = \frac{S_i^-}{S_i^+ + S_i^-}$$

MOGA-S

1. Initialization:
   (a) Parameters setting: Set the number of initial population (*np*), Number of generation (*ng*), Probability of crossover (*Pc*), Probability of mutation (*Pm*), Probability of reproduction (*Pr*).
   (b) Initial population generation: Randomly generate an initial population.
2. Evaluation: Divides the initial population into two groups (*k1, k2*) and evaluate the fitness function for each group of individuals similar sample GA.
3. Selection: Roulette wheel selection, Elitist selection and Purely random selection are employed to reproduce the next generation.
4. Crossover operation: Select *size (k1)× Pc* pairs of parents from group 1 (Roulette wheel selection), and perform crossover on the parents. Select *size (k2)× Pc* pairs of parents from group 2 (Roulette wheel selection), and perform crossover on the parents.
5. Mutation operation: Select *size(k1)× Pm* chromosome from group 1 (Purely Random Selection), and mutate the individual bits. Select *size (k2)× Pm* chromosome from group 2 (Purely Random Selection), and mutate the individual bits
6. Update Archive: Aggregate childs then all chromosomes on Front 1(efficient solutions) are copied into an archive that stores the best non-dominated front obtained
7. Generate next generation: select childs solutions from crossover and mutation operation.
8. ~~Stopping criteria: Terminate the algorithm if the stopping criterion (number~~ of generation) is met; else return to step 2.

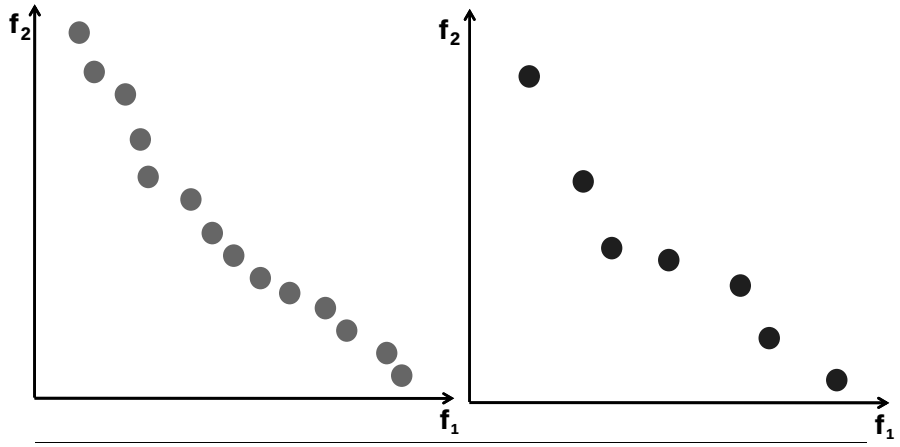
مکانیزم های لازم در الگوریتم های بهینه یابی چند هدفه



معیارهای مقایسه کارایی الگوریتم ها

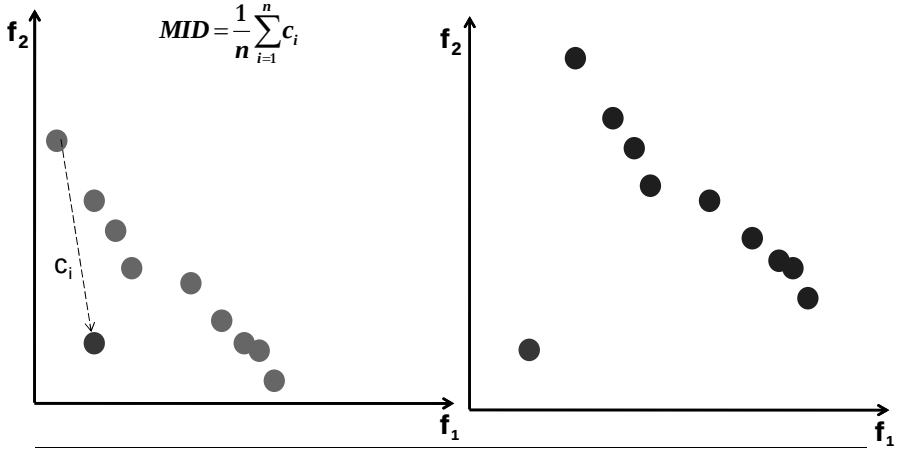
- تعداد جواب های آرشیو پارتو
- میانگین فاصله از نقطه ایده آل
- یکنواختی
- معیار پوشش مجموعه
- بیشترین گسترش

تعداد جواب های آرشیو پارتو



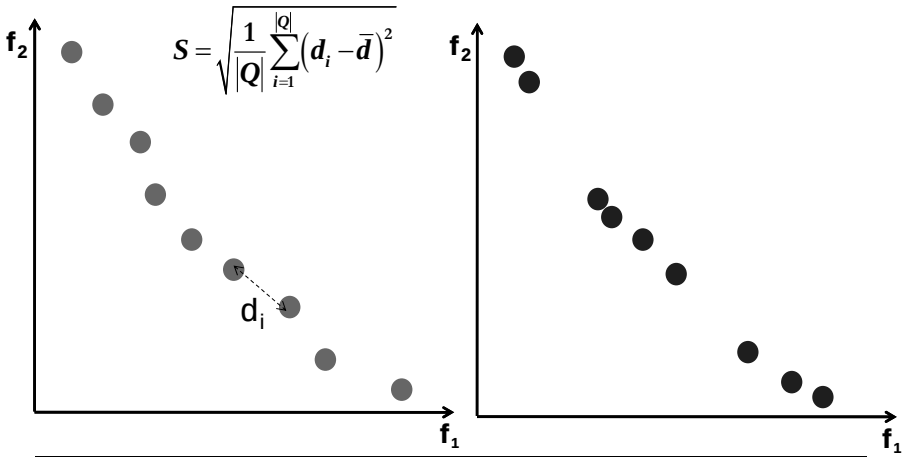
میانگین فاصله از نقطه ایده آل

$$MID = \frac{1}{n}\sum_{i=1}^{n} c_i$$



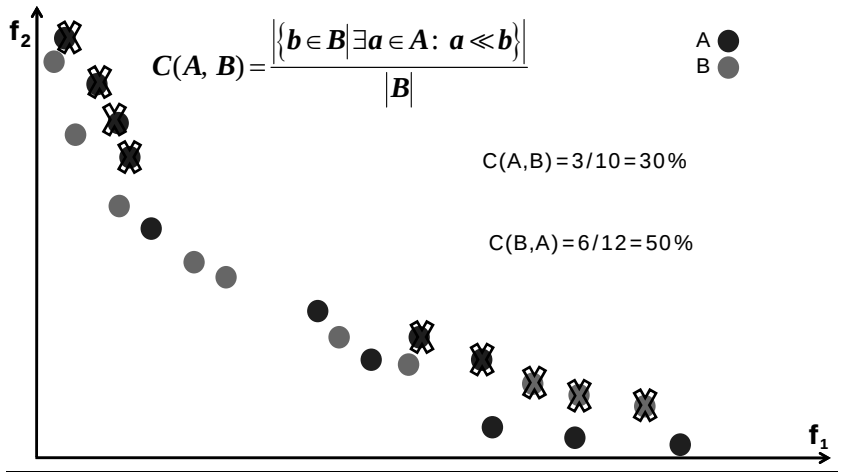
یکنواختی

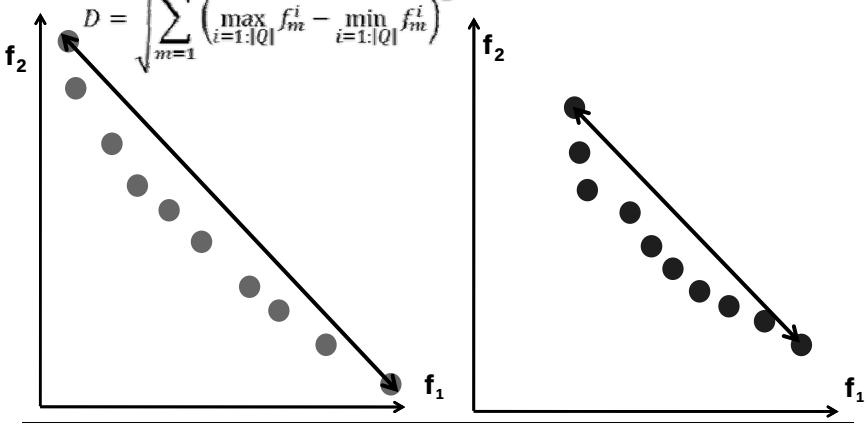
$$S = \sqrt{\frac{1}{|Q|}\sum_{i=1}^{|Q|}\left(d_i - \bar{d}\right)^2}$$



معیار پوشش مجموعه

$$C(A, B) = \frac{\left|\left\{b \in B \,\middle|\, \exists a \in A:\ a \ll b\right\}\right|}{|B|}$$



بیشترین گسترش

$$D = \sqrt{\sum_{m=1}^{M} \left(\max_{i=1:|Q|} f_m^i - \min_{i=1:|Q|} f_m^i \right)^2}$$