OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG

INF FAKULTÄT FÜR INFORMATIK

Evolutionary Algorithms

Application: Multi-Critera Optimization

Prof. Dr. Rudolf Kruse   Pascal Held

{kruse,pheld}@iws.cs.uni-magdeburg.de

Otto-von-Guericke University Magdeburg

Faculty of Computer Science

Institute of Knowledge and Language Engineering

Outline

1. Multi-Critera Optimization

Simple Approach

Pareto-Optimal Solutions

Solution with Evolutionary Algorithms

2. Example: Setting up Annennas

Multi-Criteria Optimization

in many everyday problems: optimization of more than one variable

achieve **different objectives** as good as possible

Example: requirements when bying a car

- low price,
- low fuel consumption, low emissions
- best comfort (electric windows, air conditioning)

often many different objectives are not independent, but **conflicting**

Example: bying a car

- additional charge for most comfort features
- air conditioning or large inner space require a big engine, thus higher pricing and more fuel consumption

Multi-Criteria Optimization

technical description: k critera given, with one objective function each:

$$f_i : \Omega \to \mathbb{R}, \qquad i = 1, \ldots, k$$

simple approach: put together k objective functions into one aggregated objective function, e.g. with

$$f(s) = \sum_{i=1}^{k} w_i \cdot f_i(s)$$

choosing the weights:

- *sign:* if $f \to \max$, then all weights of f $w_i > 0$, otherwise $w_i < 0$
- *absolute value:* relative significance of these criteria (take fluctuation into account!)

Multi-Criteria Optimization

problems with this approach:

- relative significance of these criteria needs to be fixed before starting the search
- choice of weights not always that simple, thus preferences between criteria might be inadequate

problems occuring with linear combination of the f_i are even more fundamental:

- in general: problem of **aggregation of preference**
- also occurs in people's elections (preferences of voters for nominees need to be aggregated)
- **Arrow's impossibility theorem** [Arrow, 1951]: there is no choice function that possesses all the desired features

Multi-Criteria Optimization

- Arrow's impossibility theorem [Arrow, 1951] may be avoided by using a **scaled order of preferences**
- **however:** scaled order of preferences is a new degree of freedom
- the task of finding a suitable scaling is most likely to be even more complex than finding suitable weights for the linear combination

Pareto-Optimal Solutions

- **alternative approach:** try to find as much **Pareto-optimal** solutions as possible

Definition

An element $s \in \Omega$ is called **Pareto-optimal** regarding the objective functions f_i, $i = 1, \ldots, k$, if there is no such element $s' \in \Omega$ for which

$$\forall i, 1 \leq i \leq k : \quad f_i(s') \geq f_i(s) \quad \text{and}$$
$$\exists i, 1 \leq i \leq k : \quad f_i(s') > f_i(s) \quad \text{holds.}$$

- **illustration:** no value of an objective function can get better without the value of another function getting worse.

Definition of “Pareto-Optimum“

element $s_1 \in \Omega$ **dominates** element $s_2 \in \Omega$, if

$$\forall i, 1 \leq i \leq k : \quad f_i(s_1) \geq f_i(s_2)$$

element $s_1 \in \Omega$ **dominates** element $s_2 \in \Omega$ **properly**, if s_1 dominates s_2 and

$$\exists i, 1 \leq i \leq k : \quad f_i(s_1) > f_i(s_2)$$

element $s_1 \in S$ is called **Pareto-optimal**, if it is not properly dominated by any other element $s_2 \in \Omega$

set of Pareto-optimal elements is called **Pareto-front**

Multi-Criteria Optimization

Advantages when searching for Pareto-optimal solutions:

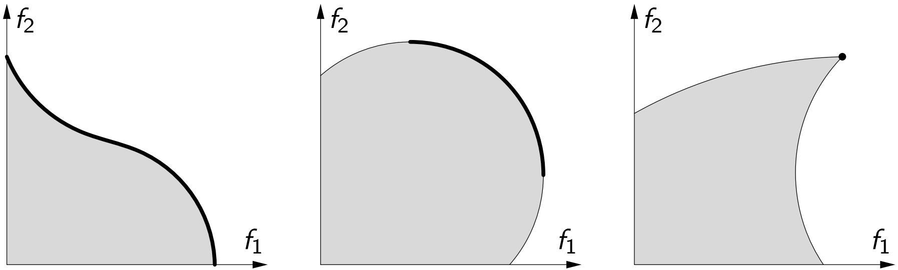
- no need for aggregating objective functions

  i.e. no need to choose weights

- Even for different preferences the search has to be performed only once.

  It is after this search that the solutions are chosen.

Pareto-Optimal Solution / Pareto-Front



- all points of Ω are located within the gray zone
- Pareto-optimal solution = bold part of the border
- note: Pareto-optimal solution can be unique (depending on the location of the candidate solutions)

Solution with Evolutionary Algorithms

- objective: spreading the population along the Pareto-Front as widely as possible
- challenge: without previously definded weights

  many different, equivalent solutions

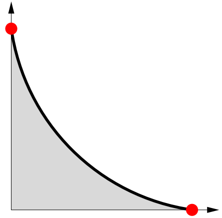
- **simplest approach:** use weighted sum of the objective functions as fitness function

Solution with Evolutionary Algorithms

obvious alternative: so-called **VEGA-method**

- given k criteria with assigned objective functions f_i, $1 \ldots, k$
- $\forall i$, $1, \ldots, k$: choose $\frac{|P|}{k}$ individuals according to the fitness function f_i

- **advantage**: simple, without much computational effort
- **disadvantage:** clear handicap for solutions that satisfy every criterion good, but none perfectly
- **consequence:** search concentrates on marginal solutions



Solution with Evolutionary Algorithms

better approach: use concept of dominance for selection

building a **rating scale** of the individuals of one population:

- find all non-dominated solutions of a population
- assing solution candidates to the best rank, remove them from the population
- repeat identification and removal of non-dominated solution candidates fot other ranks, until population is empty

perform a **rank-based selection** according to the ranking scale

problem: all individuals of the Pareto-Front are assessed as equally good

$\Rightarrow$ genetic drift: Pareto-Front converges at a random point, because of random effects

Preventing the Genetic Drift

aim: spread along the Pareto-front as equally as possible

solution: **niche techniques** to be able to decide between individuals with same rank

- e.g. *power law sharing:* individuals with frequent combination of function values get a low fitness score

  combinations occuring isolated are as probable as solution candidates of frequent occuring combinations

- sharing as for evaluation functions, but with distance measure for function values

problem: calculating the ranking scale is costly

NSGA-Selection

Non-Dominated Sorted Genetic Algorithm

alternatively: **tournament selection**, with the winner being determined by the dominance concept and niche techniques

method:

- choose reference individual
- select non-dominated individual
- otherwise: individual with less individuals in niche

here: niche defined by radius ε

Algorithm 1 NSGA-SELECTION

Input: goodness $\langle A^{(i)}.F_j \rangle_{1 \le i \le r, 1 \le j \le k}$, sample size N_{dom}

$I \leftarrow \{\}$
for $t \rightarrow 1, \ldots, s$ {
  $a \leftarrow U(\{1, \ldots, r\})$
  $b \leftarrow U(\{1, \ldots, r\})$
  $Q \leftarrow$ subset of $\{1, \ldots, r\}$ of size N_{dom}
  $d_a \leftarrow \exists i \in Q : A^{(i)} >_{\text{dom}} A^{(a)}$
  $d_b \leftarrow \exists i \in Q : A^{(i)} >_{\text{dom}} A^{(b)}$
  if d_a **and not** d_b {
    $I \leftarrow I \cup \{b\}$
  } **else** {
    if not d_a **and** d_b {
      $I \leftarrow I \cup \{a\}$
    } **else** {
      $n_a \leftarrow \left| \left\{ 1 \le i \le r \mid d(A^{(i)}, A^{(a)}) < \varepsilon \right\} \right|$
      $n_b \leftarrow \left| \left\{ 1 \le i \le r \mid d(A^{(i)}, A^{(b)}) < \varepsilon \right\} \right|$
      if $n_a > n_b$ {
        $I \leftarrow I \cup \{b\}$
      } **else** {
        $I \leftarrow I \cup \{a\}$
      }
    }
  }
}
return I

NSGA-Selection

nonetheless poor approximation of the Pareto-Front.

reasons:

parameter setting of ε

population used for two purposes

- as storage for non-dominated individuals (Pareto-Front)
- as living population (for searching the search space)

solution: separate archive for non-dominated individuals from the population

- archive often finite
- testing all individuals for dominance by archive individuals
- for newbies: remove dominated individuals from the archive

Strength Pareto EA (SPEA2)

- is a simple EA
- evaluation function: two components:
  1. how many individuals dominate individuals dominating this individual
  2. distance to the $\sqrt{n}$-th closest individual
- the archive influences the calculation of fitness and contains non-dominating individuals
  - if too few: fittest individuals additionally
  - replace in archive, because of distance to other archived individuals

Algorithm 2 SPEA2

Input: objective function $F_1, \ldots, F_k$, population size μ, archive size $\tilde{\mu}$

$t \leftarrow 0$
$P(t) \leftarrow$ create population with μ individuals
$R(t) \leftarrow \emptyset$
while termination condition not satisfied {
  evaluate $P(t)$ by $F_1, \ldots, F_k$
  for each $A \in P(t) \cup R(t)$ {
    numDom(A) $\leftarrow |\{B \in P(t) \cup R(t) \mid A >_{\mathsf{dom}} B\}|$
  }
  for each $A \in P(t) \cup R(t)$ {
    $d \leftarrow$ distance of A and its $\sqrt{\mu + \tilde{\mu}}$ closest individuals in $P(t) \cup R(t)$
    $A.F \leftarrow \frac{1}{d+2} + \sum_{B \in P(t) \cup R(t), B >_{\mathsf{dom}} A} \mathrm{numDom}(B)$
  }
  $R(t+1) \leftarrow \{A \in P(t) \cup R(t) \mid A$ is non-dominated$\}$
  while $|R(t+1)| > \tilde{\mu}$ {
    remove the individual with the smalles / second smallest distance from $R(t+1)$
  }
  if $|R(t+1)| < \tilde{\mu}$ {
    fill $R(t+1)$ with the best individuals from $P(t) \cup R(t)$
  }
  if termination condition not satisfied {
    select from $P(t)$ via TOURNAMENT SELECTION
    $P(t+1) \leftarrow$ apply recombination and mutation
    $t \leftarrow t + 1$
  }
}
return non-dominated individuals from $R(t+1)$

Pareto-Archived ES (PAES)

- $(1 + 1)$-evolution strategy
- condition of acceptance: archived individual is dominated or codomain is not frequented enough
- niches: follow from organization of the archive as a multi-dimensional hash table

Algorithm 3 PAES

Input: objective function $F_1, \ldots, F_k$, archive size $\tilde{\mu}$

```
t ← 0
A ← generate a random individual
R(t) ← {A} organized as a multi-dimensional hash-table
while termination condition not satisfied {
    B ← mutation on A
    assess B by F_1, ..., F_k
    if ∀C ∈ R(t) ∪ {A} : not (C >_dom B) {
        if ∃C ∈ R(t) : (B >_dom C) {
            remove all individuals being dominated by B from R(t)
            R(t) ← R(t) ∪ {B}
            A ← B
        } else {
            if |R(t)| = μ̃ {
                g* ← hash entry with most entries
                g ← hash entry for B
                if entries in g < entries in g* {
                    remove entries from g*
                    R(t) ← add B to R(t)
                }
            } else {
                R(t) ← add B to R(t)
                g_A ← hash entry for A
                g_B ← hash entry for B
                if entries in g_B < entries in g_A {
                    A ← B
                }
            }
        }
    }
    t ← t + 1
}
return non-dominated individuals from R(t + 1)
```

Summary

Even up-to-date methods got problems in approximating the Pareto-front with more than 3 criteria.

reason: massive computation time is needed for detection

solution: iterative presentation of current solutions found so far

user decides on the field the search should concentrate on

Outline

Task: Setting Up Antennas

- base antennas for mobile networks
- highest priority: high network availability
- second aim: low costs
- common method:
  - place base antennas and configure their size/scope ⇒ satisfy the demand
  - assign frequencies ⇒ minimize interferences

Initial Situation

- both problems are $\mathcal{NP}$-hard
- the placing might restrain the frequency assignment a lot
- within one iteration the results of frequency assignment can only partially be taken into account

basic policy decision:

- both problems are processed in parallel!

Formalization

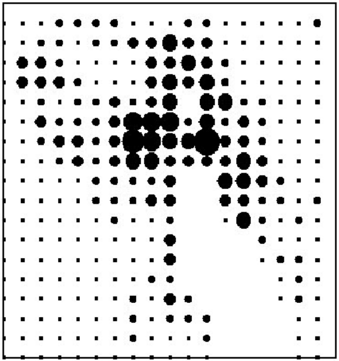
rectangular space $(x_{\text{min}}, y_{\text{min}})$ and $(x_{\text{max}}, y_{\text{max}})$ with grid *res*

set of all (possible) positions:

$$Pos = \left\{ (x_{\text{min}} + i \cdot res, y_{\text{min}} + j \cdot res) \mid 0 \leq i \leq \frac{x_{\text{max}} - x_{\text{min}}}{res} \right. \\ \left. \text{and } 0 \leq j \leq \frac{y_{\text{max}} - y_{\text{min}}}{res} \right\}$$

Call Demand Zurich

statistically determined call demands $demand(cell) \in \mathbb{N}$ for some $cell \in pos$



Formalization: Antenna

antenna $t = (pow, cap, pos, frq)$

transmission / signal strength $pow \in [MinPow, MaxPow] \subset \mathbb{N}$

call capacity $cap \subset [0, MaxCap] \subset \mathbb{N}]$

frequencies/channels $frq \subset Frequ = \{f_1, \ldots, f_k\}$ mit $|frq| \leq cap$

all possible antenna configurations:

$$T = [MinPow, MaxPow] \times [0, MaxCap] \times Pos \times Frequ$$

Genotype

problem-related genotype

$$\Omega = \mathcal{G} = \{\{t_1, \ldots, t_k\} \mid k \in \mathbb{N} \text{ and } \forall i \in \{1, \ldots, k\} : t_i \in T\}$$

variable length

Constraints

highest priority on network availability $\Rightarrow$
coded as a hard constraint

reachable positions according to wave propagation model:
$wp : pos \times [MinPow, MaxPow] \to \mathcal{P}(pos)$

$A.G = (t_1, \ldots, t_k)$ is called legal, if there is a mapping $serves(t_i, cell) \in \mathbb{N}$ (with $cell \in pos$) for all t_i, such that

- $serves(t_i, cell) > 0 \Rightarrow \; cell \in wp(t_i)$
- $\sum_{i=1}^{k} serves(t_i, cell) \geq demand(cell)$
- $\sum_{cell \in pos} serves(t_i, pos) \leq cap$ with $t_i = (pow, cap, pos, frq)$

Evaluation Functions

interference with antennas of same (or close) frequencies within one cell

$$f_{interference}(A) = \frac{\sum_{i=1}^{k} \#\text{disruptedCalls}(t_i)}{\sum_{cell \in pos} demand(cell)}$$

costs $costs(pow_i, cap_i)$ per antenna

$$f_{costs}(A) = \sum_{i=1}^{k} costs(t_i)$$

„Design Patterns“

Only legal individuals! Thus a repair function is needed.

Every antenna configuration needs to be accessible.

Extending and shortening operators are balanced.

Fine tuning and exploration are balanced:
problem specific operators and random operators

Repair Function

consider cells in a random order

if the demand is not satisfied:

1. if there is at least one antenna with some availabe capacity: choose the most powerful antenna, and assign a frequency
2. if necessary, determine the antenna that could satisfy the demand when increasing its power with minimal costs
3. if necessary, check the costs for setting up a new antenna within close proximity
4. if necessary, deploy solution (2) or (3)

Repair Function

use:

for every new generated individual

for initialization of the initial population

- repair function for an empty individual
- max. $2^{|pos|}$ individuals by (preferably) random order of the demand cells

mutation operators

- 6 „directed“ mutations, following certain concepts
- 5 „random“ mutations

Directed Mutation Operators

name	effect
DM1	if antenna got unsused frequencies ⇒ reduce capacity accordingly
DM2	if antenna uses maximal capacity ⇒ place another antenna nearby
DM3	if some antennas share large overlapping areas ⇒ remove one antenna
DM4	if some antennas share large overlapping areas ⇒ reduce intensity of 1 antenna (still satisfying the demand!)
DM5	if there are interferences ⇒ change affected frequencies
DM6	if antenna serves only little amount of calls ⇒ remove this antenna

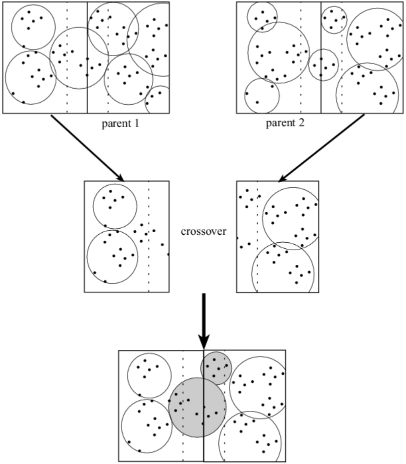
Random Mutation Operators

name	effect
RM1	change position of an antenna (intensity and capacity unchanged, frequencies new according to repair function)
RM2	fully random individual (as during initialization)
RM3	change intensity of an antenna randomly ⇒ compensation for *DM4*
RM4	change capacity of an antenna randomly ⇒ compensation for *DM1*
RM5	change assigned frequencies of an antenna ⇒ compensation for *DM5*

Recombination

- split the antennas in two halves (vertically or horizontally)
- exchange the two halves between the individuals
- fill corridor arount the split with a repair function

Recombination: Example



Selection

modern multi-objective selection is needed

problem of existing algorithms (e.g. SPEA):

- individual is integrated with $\mathcal{O}(\mu^2)$ in archive of size μ
- unfavorable because of „steady state"-approach (see basic policy decision!)

parental selection as tournament selection, based on

- $\textit{dominates}(A)$ = set of individuals (in the population) being dominated by A
- $\textit{isDominated}(A)$ = set of individuals dominating A

assign rank

$$\textit{rank}(A) = \#\textit{isDominated}(A) \cdot \mu + \#\textit{dominates}(A)$$

only problem: genetic drift, if all individuals are equivalent

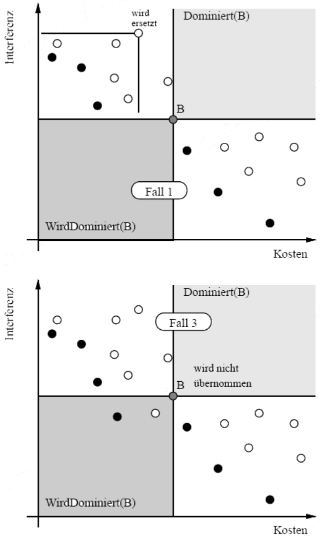
Selection
Four Options

Will the new individual be included in the next generation?

Which one will be replaced?

1. both sets empty $\Rightarrow$ include and delete individual with poorest rank
2. $dominates(B) \neq \emptyset \Rightarrow$ include and delete worst individual from $dominates(B)$
3. $dominates(B) = \emptyset \wedge isDominated(B) \neq \emptyset$
   $\Rightarrow$ B is unconsidered
4. both sets empty and no individual is dominated by another one
   $\Rightarrow$ include and delete according to measure of niche technique

Selection



Interferenz
wird ersetzt
Dominiert(B)
B
Fall 1
WirdDominiert(B)
Kosten
Interferenz
Dominiert(B)
wird ersetzt
B
Fall 2
WirdDominiert(B)
Kosten
Interferenz
Dominiert(B)
Fall 3
wird nicht übernommen
B
WirdDominiert(B)
Kosten
Interferenz
Dominiert(B)
Fall 4
B
ein Individuum mit vielen Nachbarn wird ersetzt
WirdDominiert(B)
Kosten

Selection

- data structure for population: 2D range tree
- ranges according to both objective function values
- searching, inserting and deleting in $\mathcal{O}(\log^2 \mu)$
- 2D range queries (all individuals within this range) in $\mathcal{O}(k + \log^2 \mu)$ with number k of found individuals

Altorithm

Algorithm 4 Antenna Optimization

Input: antenna problem

$t \leftarrow 0$
$P(t) \leftarrow$ initialize μ individuals with repair function
calculate rank for individuals in $P(t)$
while $t \leq G$ { /* maximum number of generations G */
  $A, B \leftarrow$ select from $P(t)$ according to rank and TOURNAMENT SELECTION
  $C \leftarrow$ apply operator to A (and for recombination to B)
  calculate sets *dominates*(C) and *isDominated*(C)
  $P(t+1) \leftarrow$ integrate C in $P(t)$ and update ranks
  $t \leftarrow t + 1$
}
return non-dominating individuals from $P(t)$

Specific Problem Data

- $9 \times 9\ km^2$ area within Zurich
- creating grid
  - demand $500m$
  - placing antennas $100m$
- 505 calls in total
- $\#frequ = 128$
- maximum capacity $MaxCap = 64$
- intensity between $MinPow = 10dBmW$ and $MaxPow = 130dBmW$

Cost Function and Parameters

cost function

- costs of one antenna:

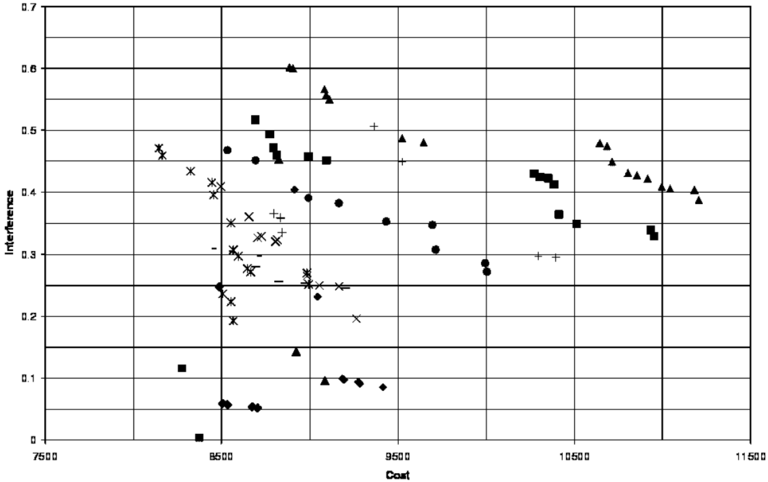
$$costs(pow_i, cap_i) = 10 \cdot pow_i + cap_i$$

parameter adjustment

- population size $\mu = 80$
- 64000 evaluations
- archive size of 80 individuals (SPEA)

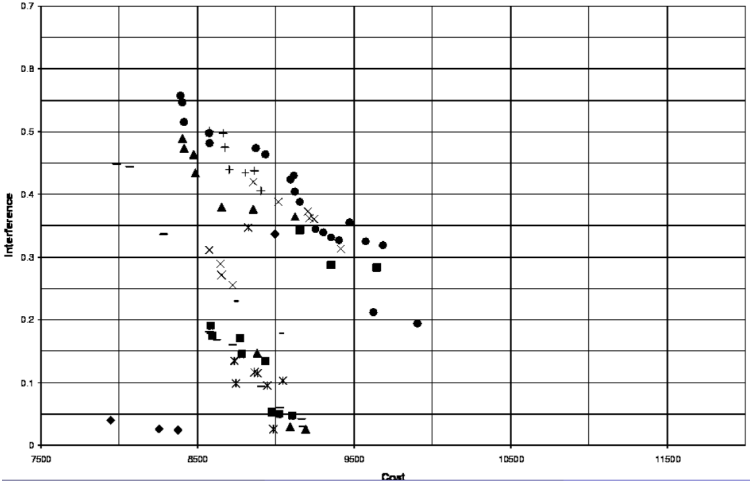
Pareto-Front

SPEA2, $p_{RM} = p_{DM} = 0.5$ and $p_{Rek} = 0$



Pareto-Front

eigene Selektion, $p_{RM} = p_{DM} = 0.5$ and $p_{Rek} = 0$



Sa Fronts roughly convex:

$$\begin{aligned}
\widehat{f_{inference}}(A) &= \frac{f_{inference}}{0.7} \\
\widehat{f_{costs}}(A) &= \frac{f_{costs} - 7500}{4500} \\
qual(P) &= \min_{A \in P} \left(\alpha \cdot \widehat{f_{inference}}(A) + (1 - \alpha) \cdot \widehat{f_{costs}}(A) \right)
\end{aligned}$$

t-test for values of 16 experiments

positive only, if significant for all $\alpha \in \{0.1, 0.2, \ldots, 0.9\}$

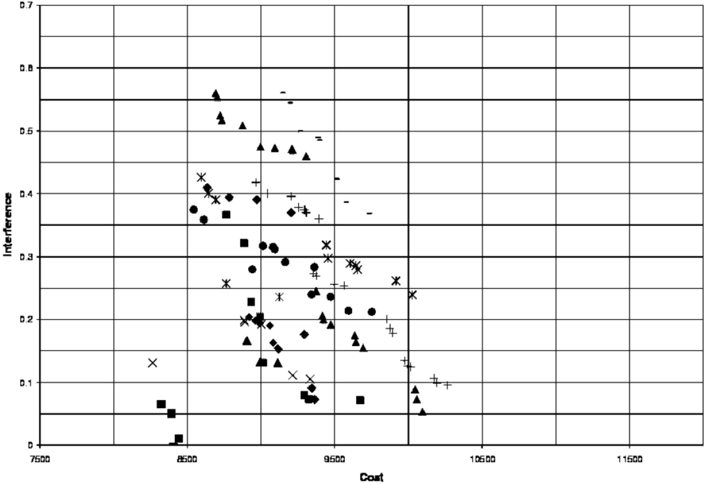
significant: combination better than just chance

no difference: previous figures

best result: next slide

Pareto-Front

own selection, $p_{RM} = p_{DM} = 0.3$ and $p_{Rek} = 0.4$



Literature for This Lecture I

Arrow, K. J. (1951).
Social Choice and Individual Values.
PhD thesis, Wiley, New York, USA.