EPFL

Online Movie Streaming

Dorsa Abdolhamidi
Onur Demiray
Jun Han
Philipp Schneider
Andrea Xu

@Netflix

June 1, 2021

Outline

Introduction

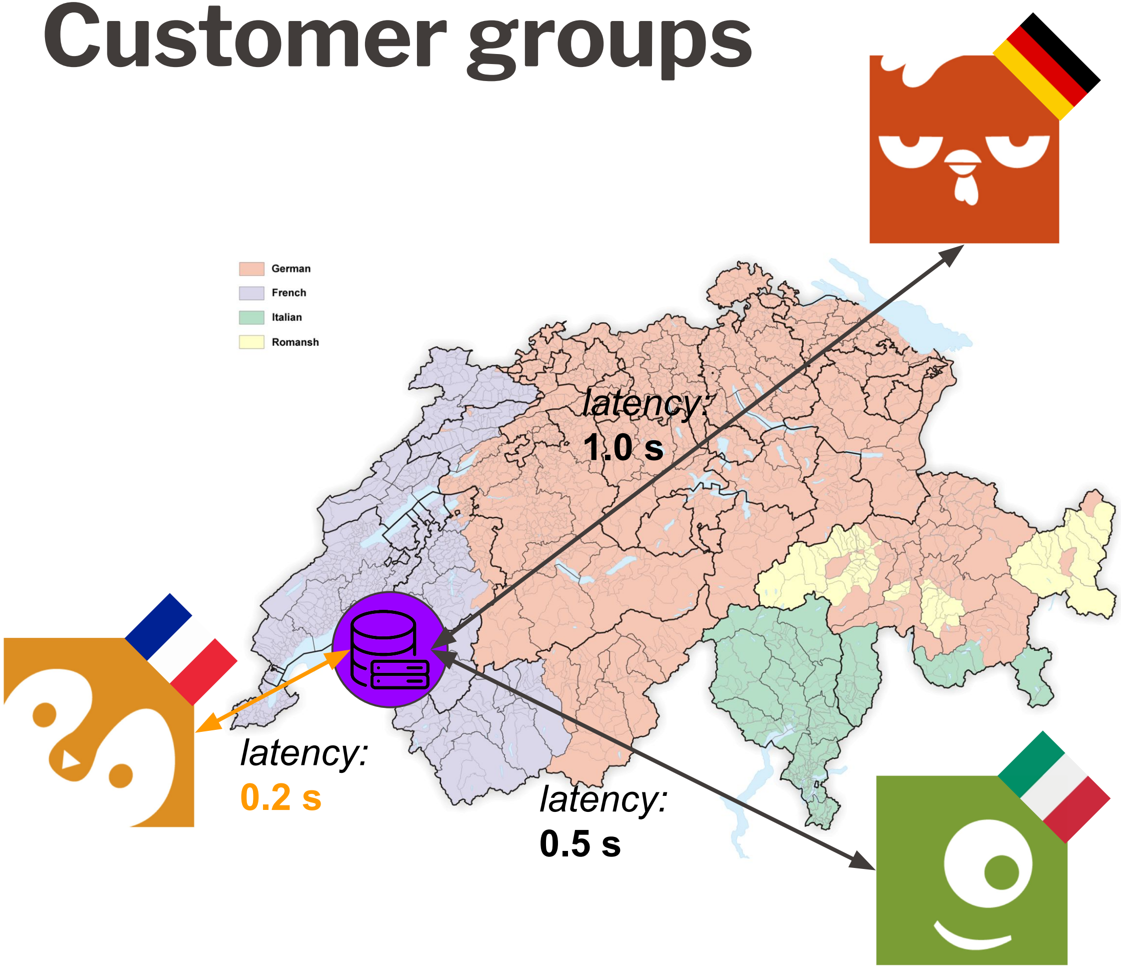
Simulation

- Discrete event simulation
- Initial system evaluation
- Bootstrapping, variance reduction

Optimization

- Problem formulation
- Single-objective - algorithms
- Neighborhood definitions
- Multi-objective optimization problem
- Computational experiments

Customer groups



Provided for each group:

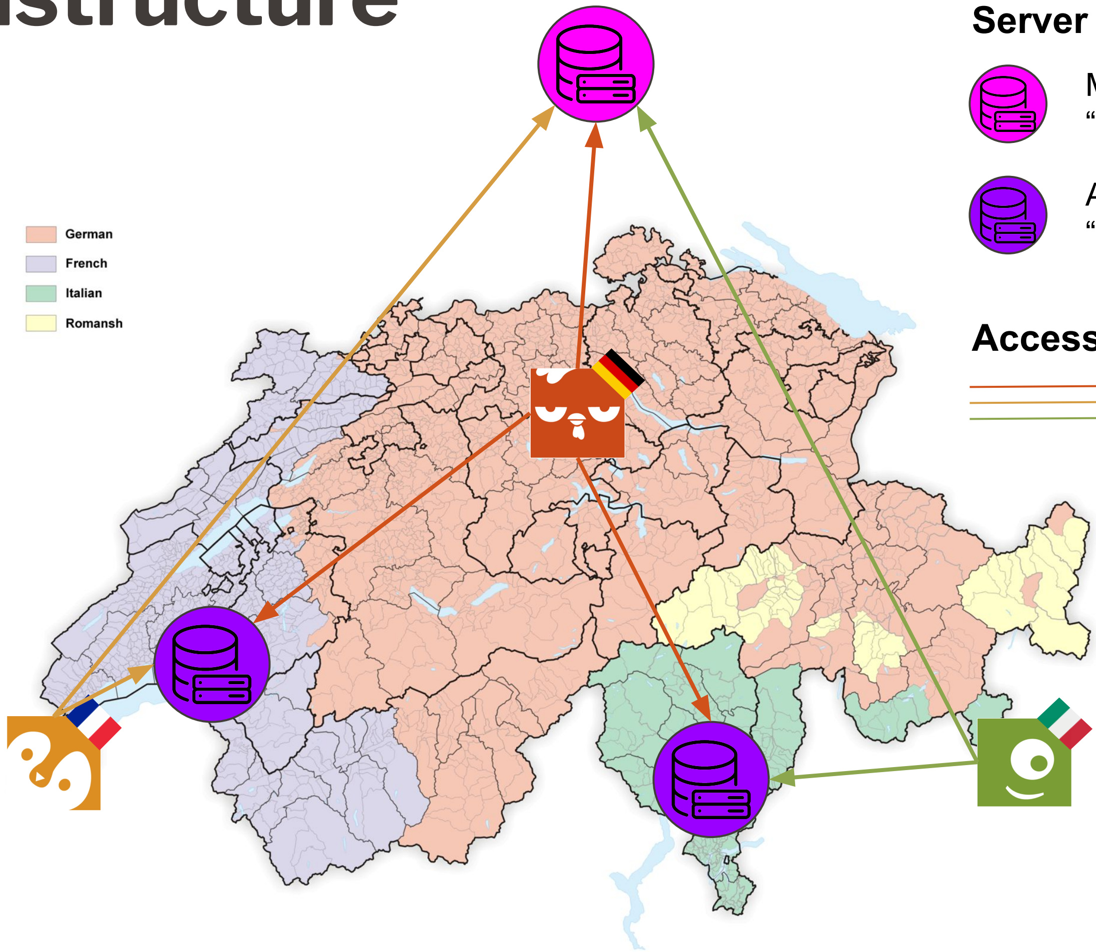
- movie popularity
- activity pattern
- latency information

waiting time

$$\omega = t_{\text{send}} + t_{\text{handle}} + t_{\text{serve}}$$

EPFL

Infrastructure



Server types

Main storage node "MSN" (unlimited storage)

Additional storage node "ASN" (limited storage)

Access constraints

Problem Description

Infrastructure

10 movies

MSN

0	1	2	3	4	5	6	7	8	9

ASN1

3500 MB
storage space

ASN2

What **movies** do we **store** at **each server**?

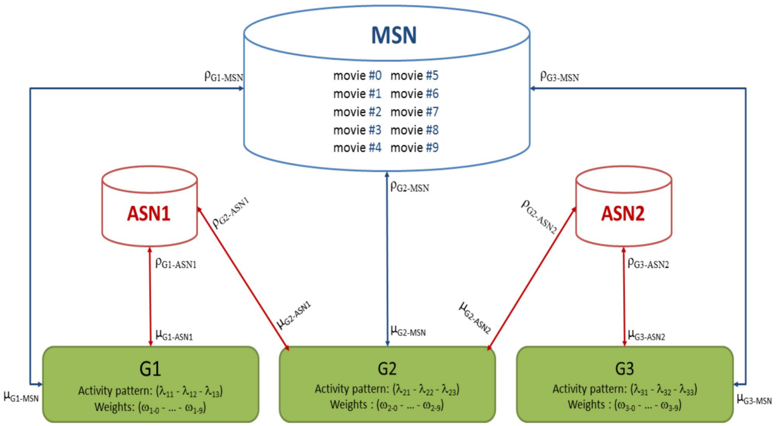
Customer groups

Objective

Maximize customer satisfaction → **Minimize waiting time**

Storage Nodes, Movies, Customer Groups

1. 1 MSN (outside Switzerland), 2 ASNs (ASN with the capacity of 3500 MB), prefer to store in ASNs
2. 10 movies
3. 3 customer groups



What We Know, Our Aim

Given:
1. Size of movies
2. Movie popularity in different groups
3. Nonhomogeneous Poisson process for arrival of customers

Waiting time:
1. Send a request (Table 4) *closest storage node*
2. Handle a request (Exp(2)) *Storage Node Occupied*
3. Serve a movie: Deterministic (Table 5) + Random (U(0.3,0.7))

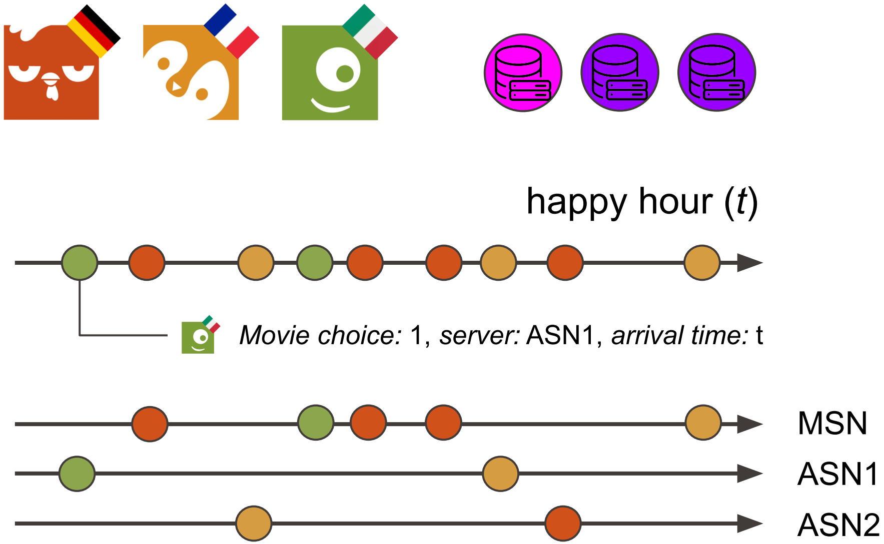
Aim:
1. Store which movies to which ASN
2. Minimize waiting time
3. Multi-objective: Consider the cost of storage in ASNs

Simulation

Discrete event simulation

Scenario Setup

- Customer groups
- Movie storage / allocation
- Server latency



Customer generation

- Generate customers
- Assign movie
- Assign server

Process customers

Quality of Service (QoS) - Statistics overall system:

- Average waiting time
- Max waiting time
- 75th percentile waiting time

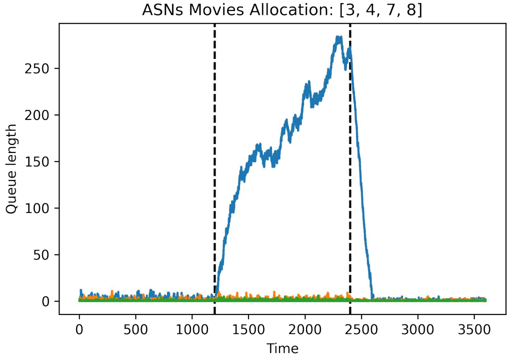
Additional statistics per server / per customer group: (Max, min, avg., std., var.)

Computation time:

- One run: 4.5s

Evaluation of system performance

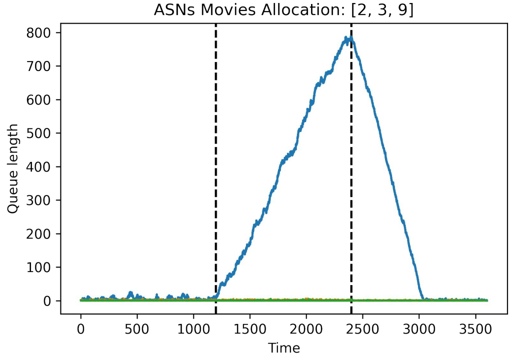
Allocation 1



Waiting time

Average: 30.77s
Max: 240.55s
75th percentile: 46.51s

Allocation 2



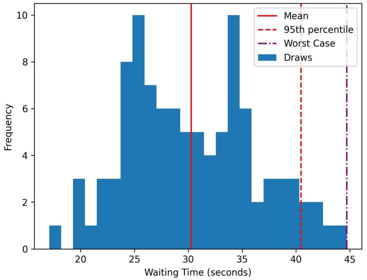
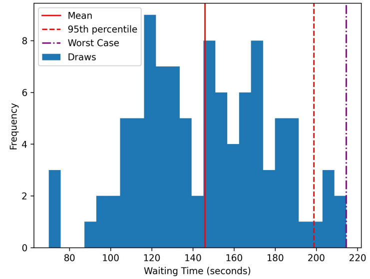
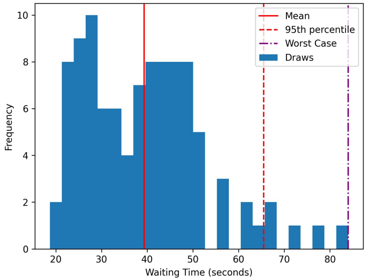
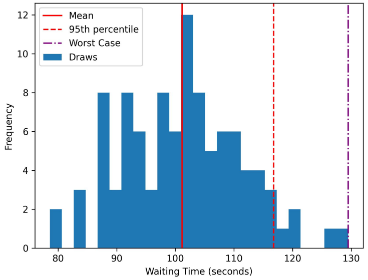
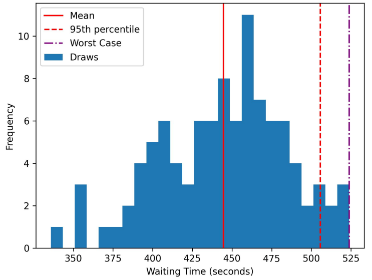
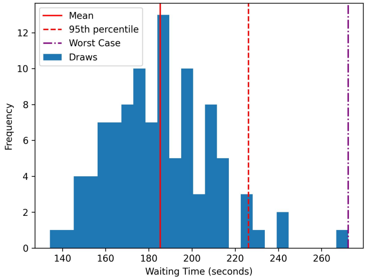
Waiting time

Average: 102.75s
Max: 549.19s
75th percentile: 207.08s

Non-optimized allocation of movies to servers results in **customer dissatisfaction**

Statistics & Bootstrapping

Speaker

	Average	Max	75th percentile
Allocation 1	 MSE: 0.37s	 MSE: 7.67s	 MSE: 2.22s
Allocation 2	 MSE: 0.92s	 MSE: 12.68s	 MSE: 11.19s

Variance Reduction

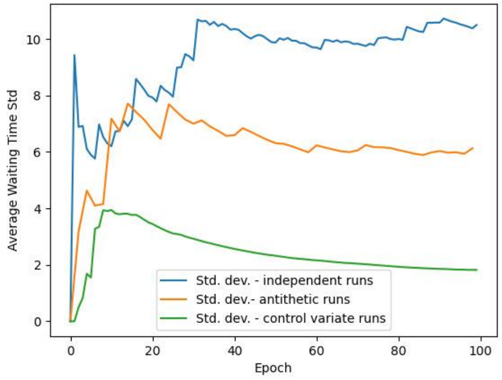
Insight: **MSN**

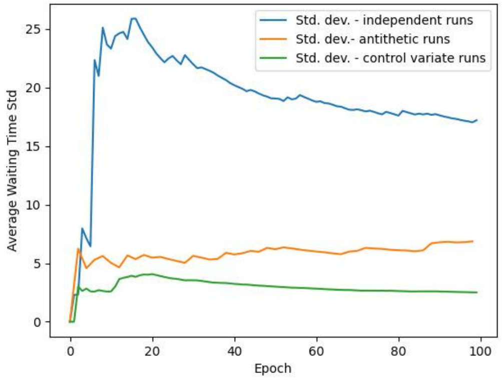
Allocation 1

Allocation 2

Variable:
Average waiting time

Control Variable:
Maximum queue length





Variance Reduction

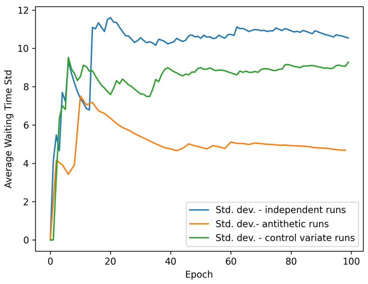
Overall system

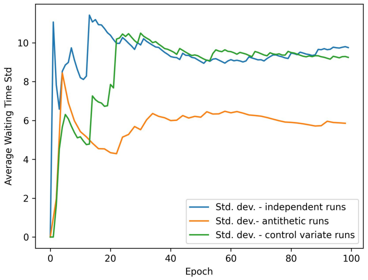
Variable:
Average waiting time

Control Variable:
Total customers

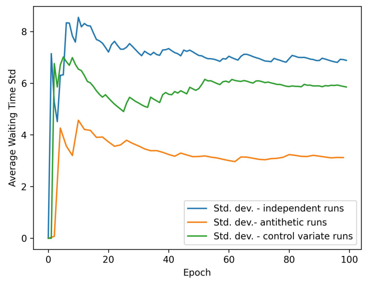
Control Variable:
Total customers MSN

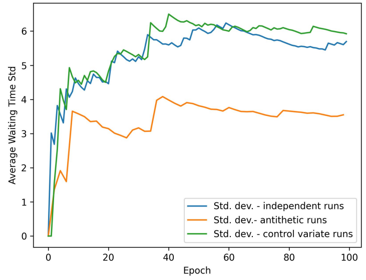
Allocation 1





Allocation 2





Antithetic runs for variance reduction, 30 runs are sufficient

Optimization

Problem Formulation

Sets

- $\mathcal{M}$: The set of movies
- $\mathcal{A}$: The set of additional storage nodes

Variables

- $x_{ij} = \begin{cases} 1, & \text{if movie } i \in \mathcal{M} \text{ is stored in ASN } j \in \mathcal{A} \\ 0, & \text{otherwise} \end{cases}$
- $\omega\left(\{x_{ij}\}_{\substack{i \in \mathcal{M} \\ j \in \mathcal{A}}}\right)$: Average waiting time of customers

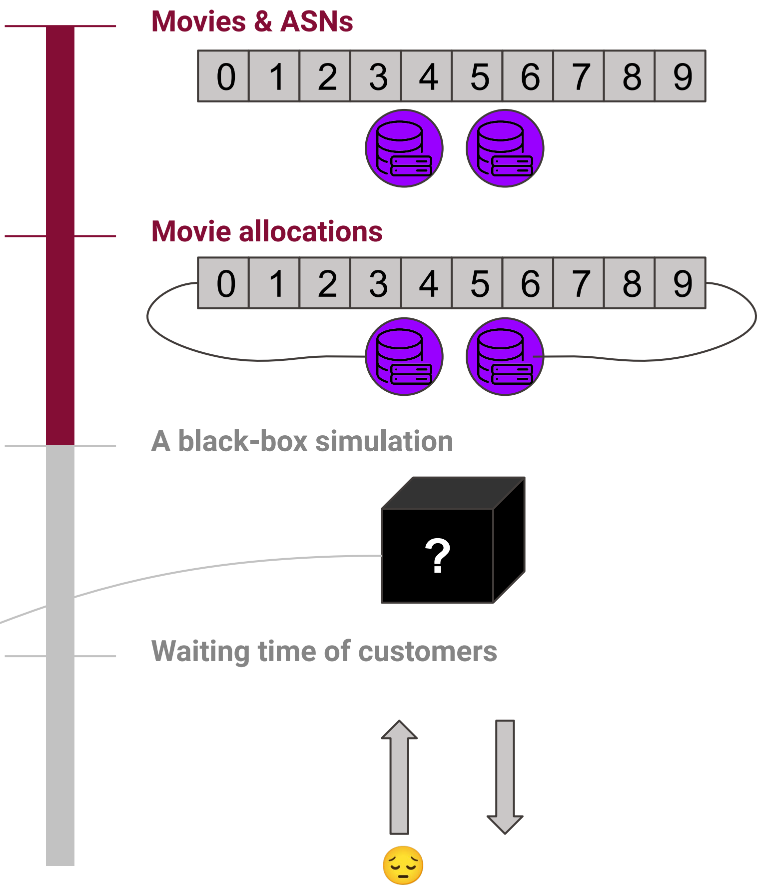
Model

$$\min \mathbb{E}[\omega]$$

$$\text{s.t.} \sum_{i \in \mathcal{M}} x_{ij} \leq 3500, \quad \forall j \in \mathcal{A}$$

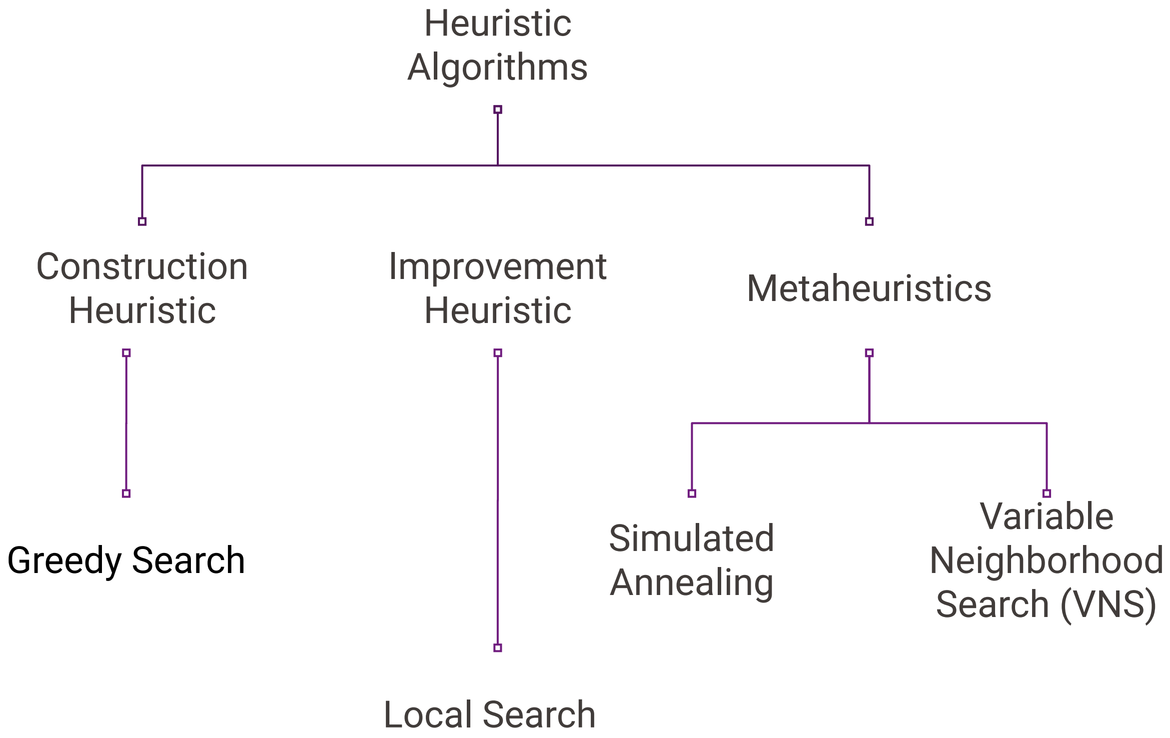
$$\boxed{\omega = f(\{x_{ij}\}_{\substack{i \in \mathcal{M} \\ j \in \mathcal{A}}})}$$

$$x_{ij} \in \{0, 1\} \qquad \forall i \in \mathcal{M}, j \in \mathcal{A}$$



Single Objective - Algorithms

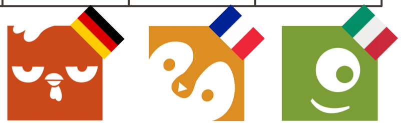
- Combinatorial simulation optimization problem



Greedy Search

- Popularity & distance based movie allocation
- Illustrative example with 5 movies

movie	size	popularities		
0	1600	3	4	3
1	1000	2	1	4
2	900	4	5	2
3	1500	3	3	5
4	1000	4	2	1



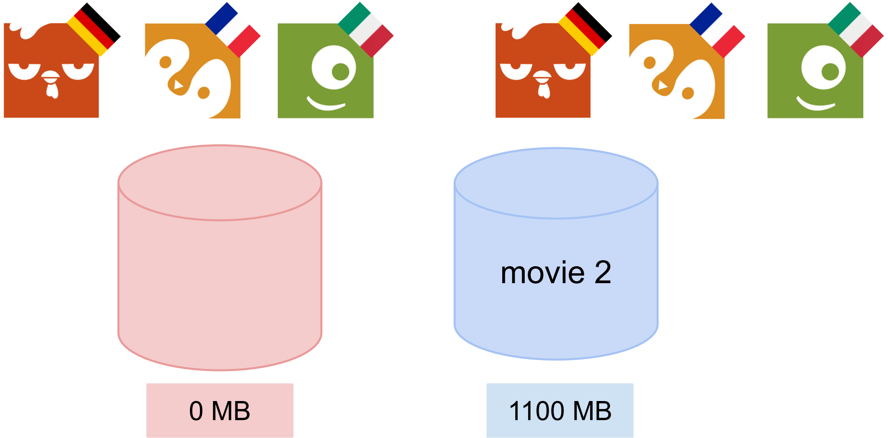
distances			
40	70	55	ASN 1
60	40	65	ASN 2

Greedy Search

- Iteration 1

movie	size	popularities		
0	1600	3	4	3
1	1000	2	1	4
2	900	4	5	2
3	1500	3	3	5
4	1000	4	2	1

distances			
40	70	55	ASN 1
60	40	65	ASN 2

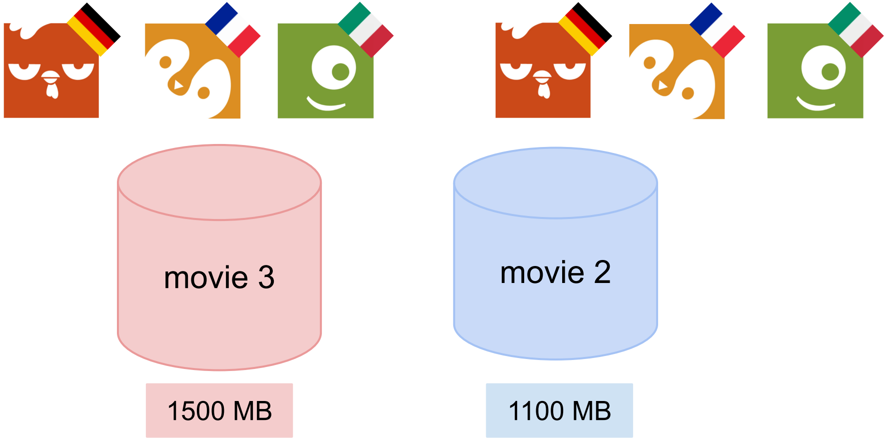


Greedy Search

- Iteration 2

movie	size	popularities		
0	1600	3	4	3
1	1000	2	1	4
2	900	4		2
3	1500	3	3	5
4	1000	4	2	1

distances			
40	70	55	ASN 1
60	40	65	ASN 2

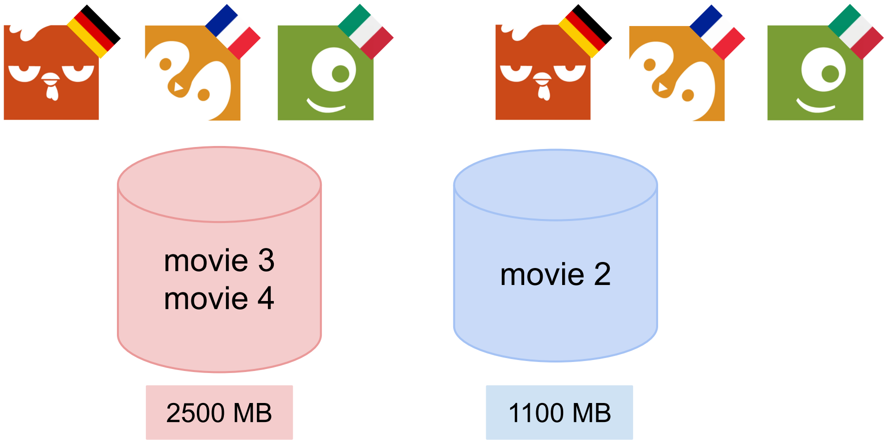


Greedy Search

- Iteration 3

movie	size	popularities		
0	1600	3	4	3
1	1000	2	1	4
2	900	4		2
3	1500	3	3	
4	1000	4	2	1

distances			
40	70	55	ASN 1
60	40	65	ASN 2

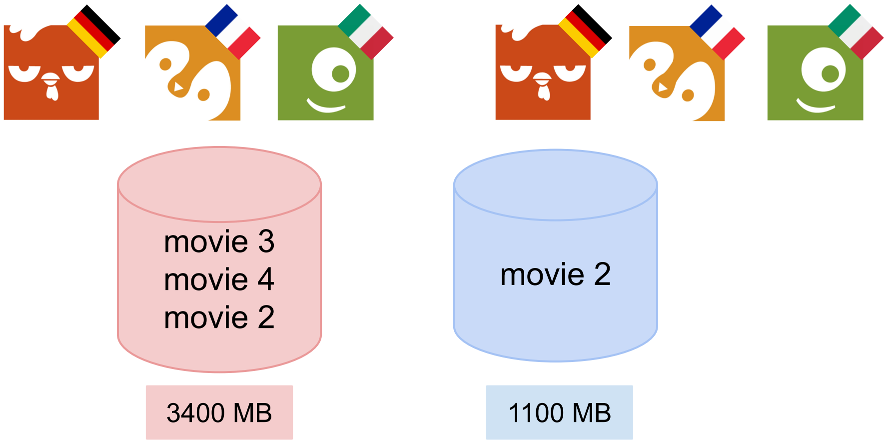


Greedy Search

- Iteration 4

movie	size	popularities		
0	1600	3	4	3
1	1000	2	1	4
2	900	4		2
3	1500	3	3	
4	1000		2	1

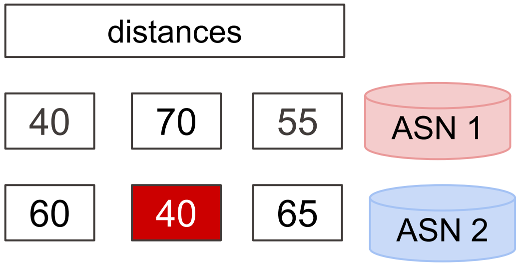
distances			
40	70	55	ASN 1
60	40	65	ASN 2

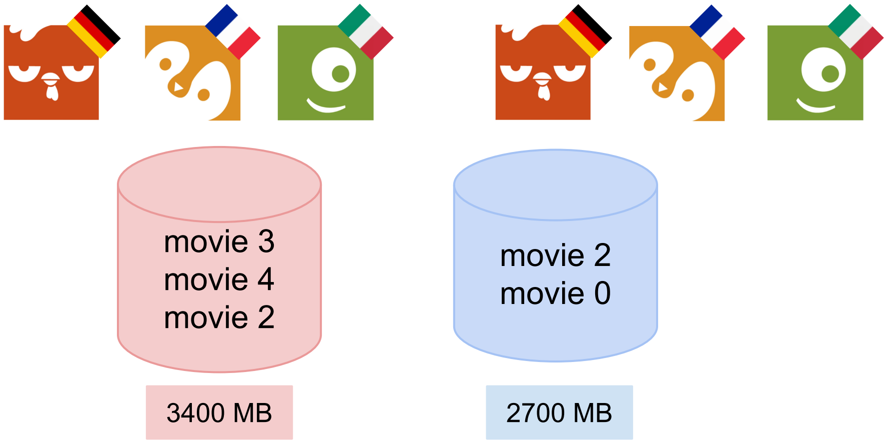


Greedy Search

- Iteration 5

movie	size	popularities		
0	1600	3	**4**	3
1	1000	2	1	4
2	900			2
3	1500	3	3	
4	1000		2	1

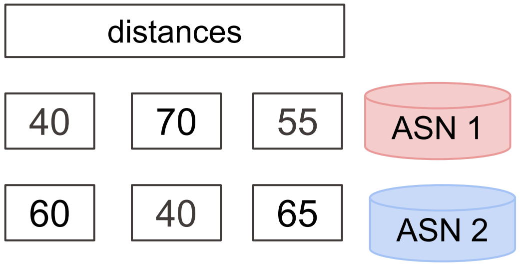


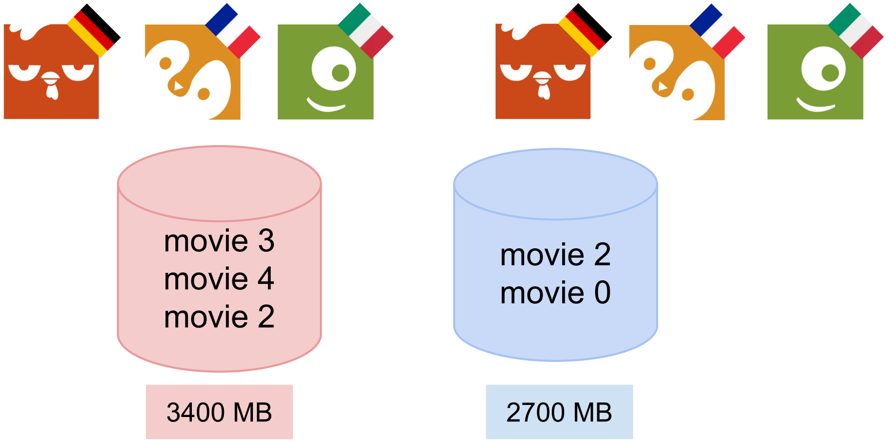


Greedy Search

- Stopping criteria

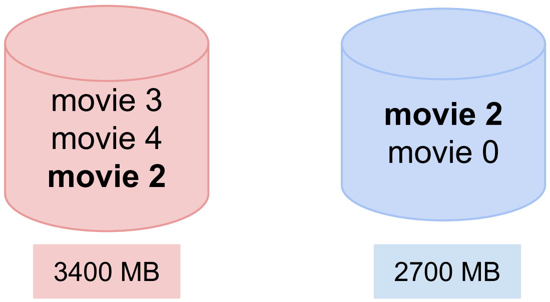
movie	size	popularities		
0	1600	3		3
1	1000	2	1	4
2	900			2
3	1500	3	3	
4	1000		2	1





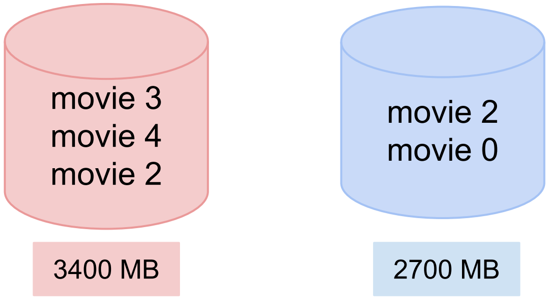
Greedy Search

- Caution!!!



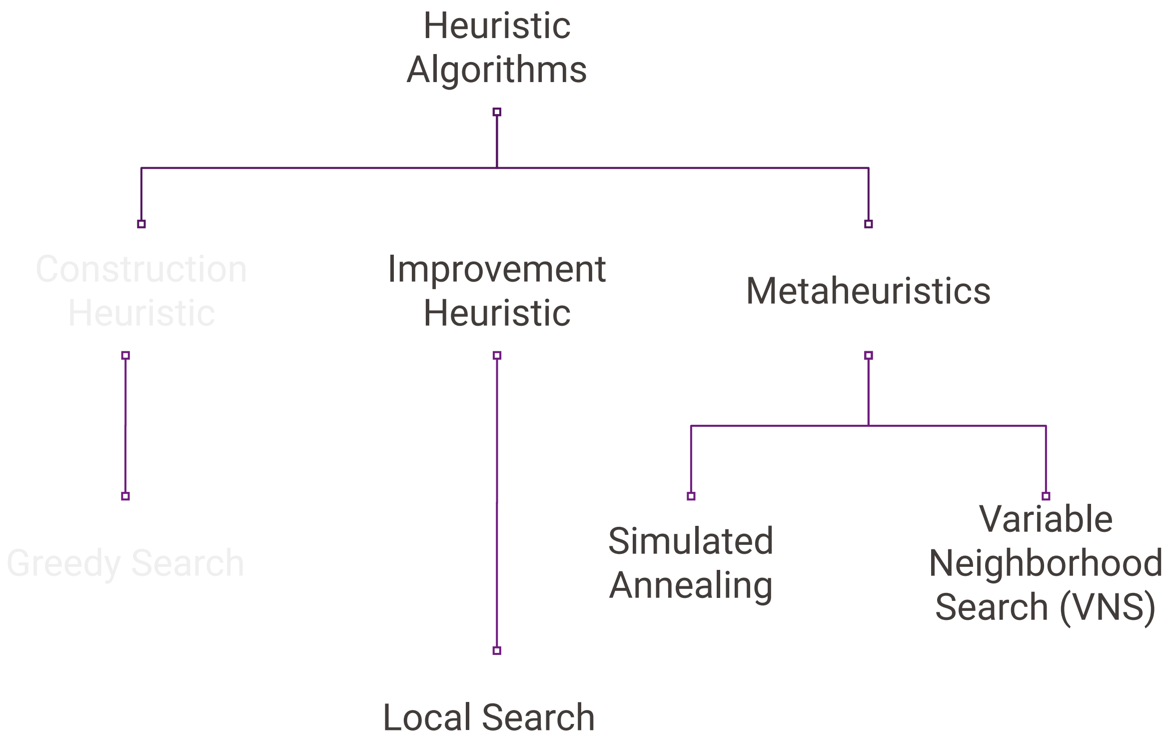
- A movie can be stored in both ASNs.
- If it could be stored in at most 1 store:
  - $\sum_{j \in \mathcal{A}} x_{ij} \leq 1, \forall i \in \mathcal{M}$

Greedy Search



- Generates a feasible solution in short computational time
- Provides a good initial solution due to the problem size

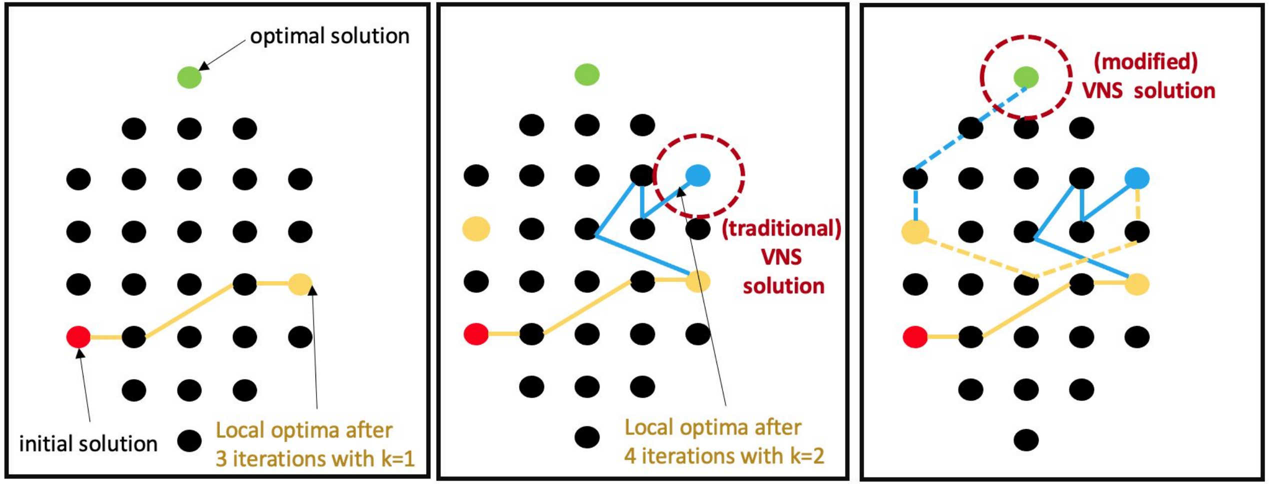
Improvement heuristics & Metaheuristics



- Local search for intensification
- Metaheuristics for diversification
- 4 neighborhood definitions for all
- Simulated annealing with non-linear temperature updates
- A modification to VNS to **increase diversification** further

Modification in VNS

- Set k=1 whenever k=K and number of iterations does not reach the maximum number of iterations
- Increases diversification, thus the chance of getting global optima
- Illustrative example with 2 neighborhood definitions:



Neighborhood Definitions
3 random neighbors

1. Randomly remove 1 movie, randomly and sequentially add 2 (not in the previous list), check size
   Example: (0,1,2) $\longrightarrow$ (0,1) $\longrightarrow$ (0,1,5) $\underset{\text{Over capacity!}}{\rightleftarrows}$ (0,1,5,6)
2. Randomly remove 2 movies, randomly and sequentially add 3 (not in the previous list), check size
3. Randomly remove 3 movies, randomly and sequentially add 4 (not in the previous list), check size

Comments:

a. The "distance" of the first method is the closest
b. After sufficient runs, the movie list always contains at least 3 movies

Neighborhood Definitions

ASNs Binary Change neighbor

4. Randomly choose at most 2 movies in ASN1 and ASN2
   For ASN1, check:
   (1) include the movies in ASN2, remove
   (2) not include, add, check capacity
   Apply similarly to ASN2

   Example:
   ASN1: (0,1,2)   ASN2: (1,3,4)
   Choose (1,2) for ASN1 and (1,3) for ASN2
   ASN1 (0,2) → (0,2,3)
   ASN2 (3,4) → (2,3,4)

Comments:

a. when both ASNs contain the same movies

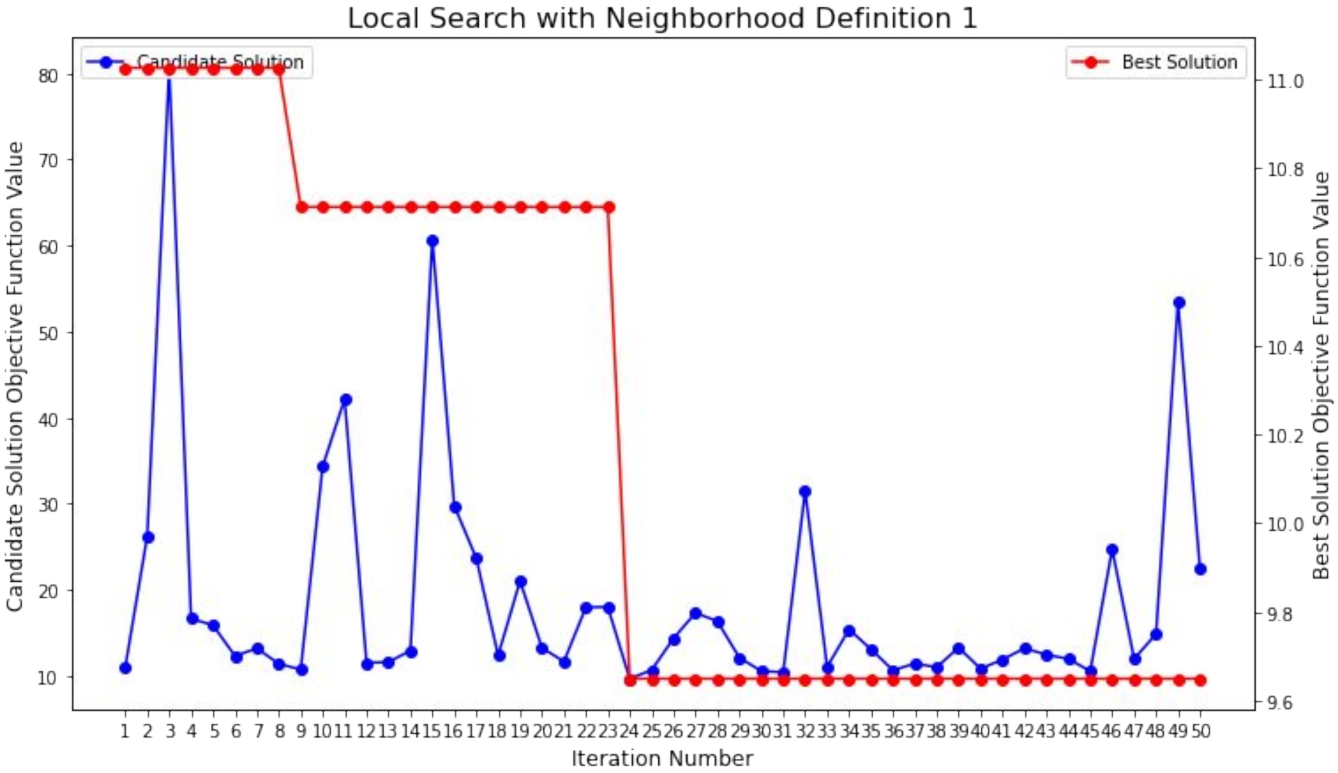
Multi-objective Optimization Problem

- Total size (MB) in ASNs **vs** waiting time of customers
- Cost of storing movies in ASNs.
- Two conflicting objectives:
  - Minimize average waiting time of customers
    - $f_1 : \min \mathbb{E}[\omega]$
  - Minimize the total size of movies stored in ASNs
    - $f_2 : \min \sum_{i \in \mathcal{M}} \sum_{j \in \mathcal{A}} x_{ij} c_i$ where c_i is the size in MB of movie $i \in \mathcal{M}$
- Two algorithms:
  - Local Search
  - Variable Neighborhood Search (VNS)

Computational Experiments

- Comparison among neighborhood methods
- Random vs. greedy initial solution
- Comparison among single objective algorithms
- Comparison among multiobjective algorithms
- Different objectives

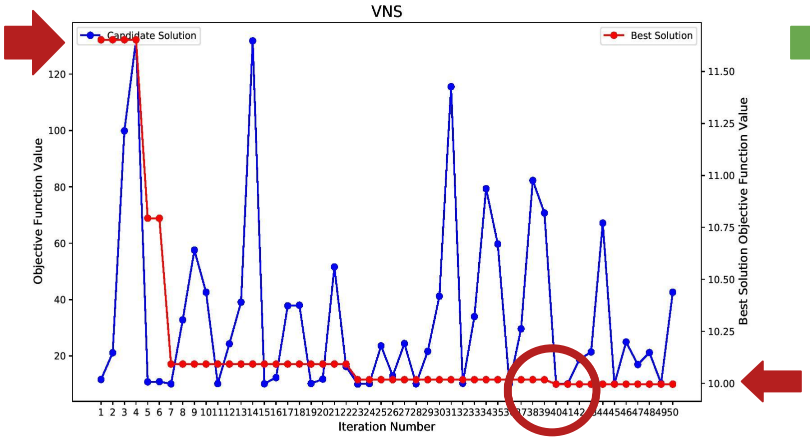
Comparison among Neighborhood Methods



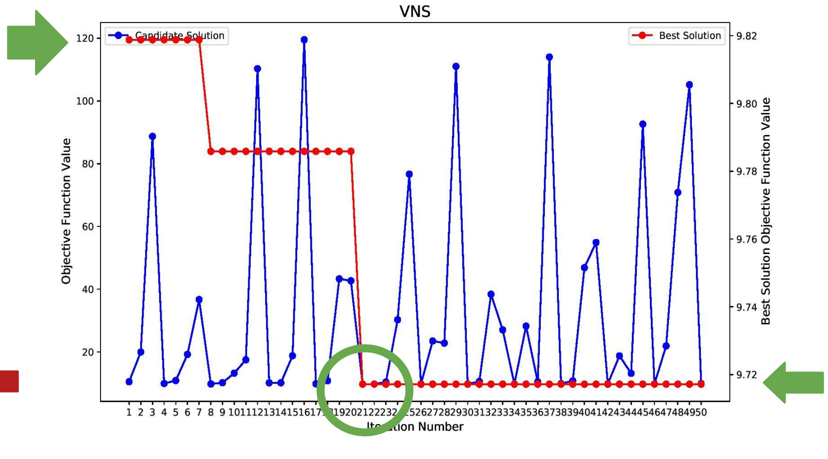
Random vs. Greedy Initial Solution

Speaker

Random initial Solution



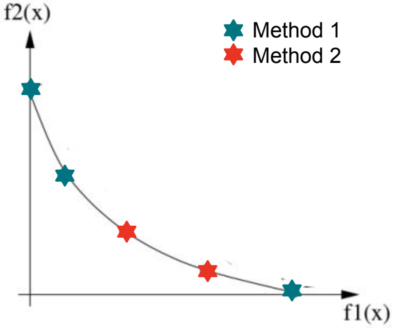
Greedy initial Solution



Comparison among Single Objective Algorithms

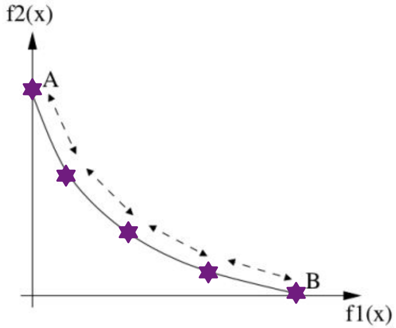
Algorithm	Local Search (N=1)	Simulated Annealing (N=1)	Variable Neighborhood Search
Time	*1:24*	2:26	3:10
Best objective	*9.6497*	9.8894	9.6847

Metrics for Comparing Multiobjective Algorithms



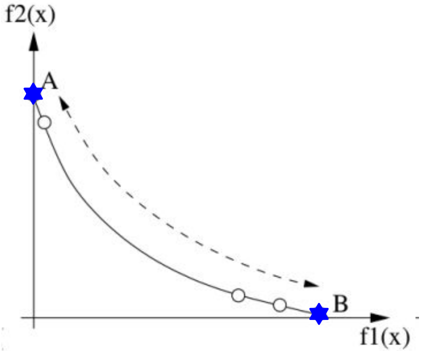
- Time
- Quality metric [1]
- Spacing metric [2]

$$SM = \sqrt{\frac{\sum_{k=1}^{m}\left(d_k - \bar{d}\right)^2}{m-1}} \qquad d_k = \min_{i \neq j} \sum_{h=1}^{n} \left| \frac{f_h^j - f_h^i}{f_h^{max}} \right|$$



- Diversity metric [3]

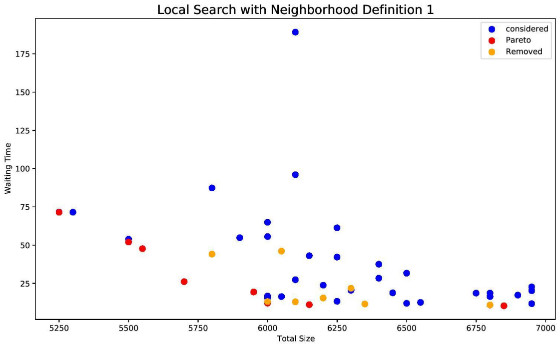
$$DM = \sqrt{\sum_{h=1}^{n} \max_{i,j} \left(\frac{f_h^j - f_h^i}{f_h^{max}} \right)^2}$$

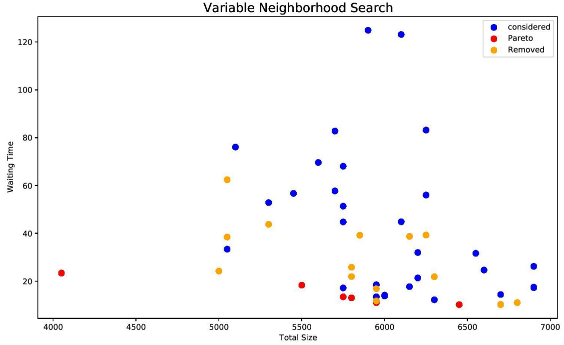


Speaker

Comparison among multiobjective algorithms

	Local Search (N=1)	Variable Neighborhood Search
Spacing Metric	**0.1043**	0.1956
Diversity Metric	**0.8775**	0.6788
Quality Metric (%)	16.6667	**83.3333**
Time	**1:46**	3:57





Speaker

Different objectives

Local Search (N=1)

Objective	Average	Maximum
Time	2:21	2:10
Average	9.6497	9.8090
Maximum	29.3452	28.2079

References

[1] Rabbani, M., Heidari, R., Farrokhi-Asl, H., & Rahimi, N., 2018. Using metaheuristic algorithms to solve a multi-objective industrial hazardous waste location-routing problem considering incompatible waste types. Journal of Cleaner Production, 170, pp.227-241,

[2] Jason R Schott. Fault tolerant design using single and multicriteria genetic algorithm optimization. Technical report, DTIC Document, 1995

[3] Zitzler, E., 1999. Evolutionary algorithms for multiobjective optimization: Methods and applications (Vol. 63). Ithaca: Shaker.

Acknowledgement

Thank you for your listening!

Any questions?