MRPGA

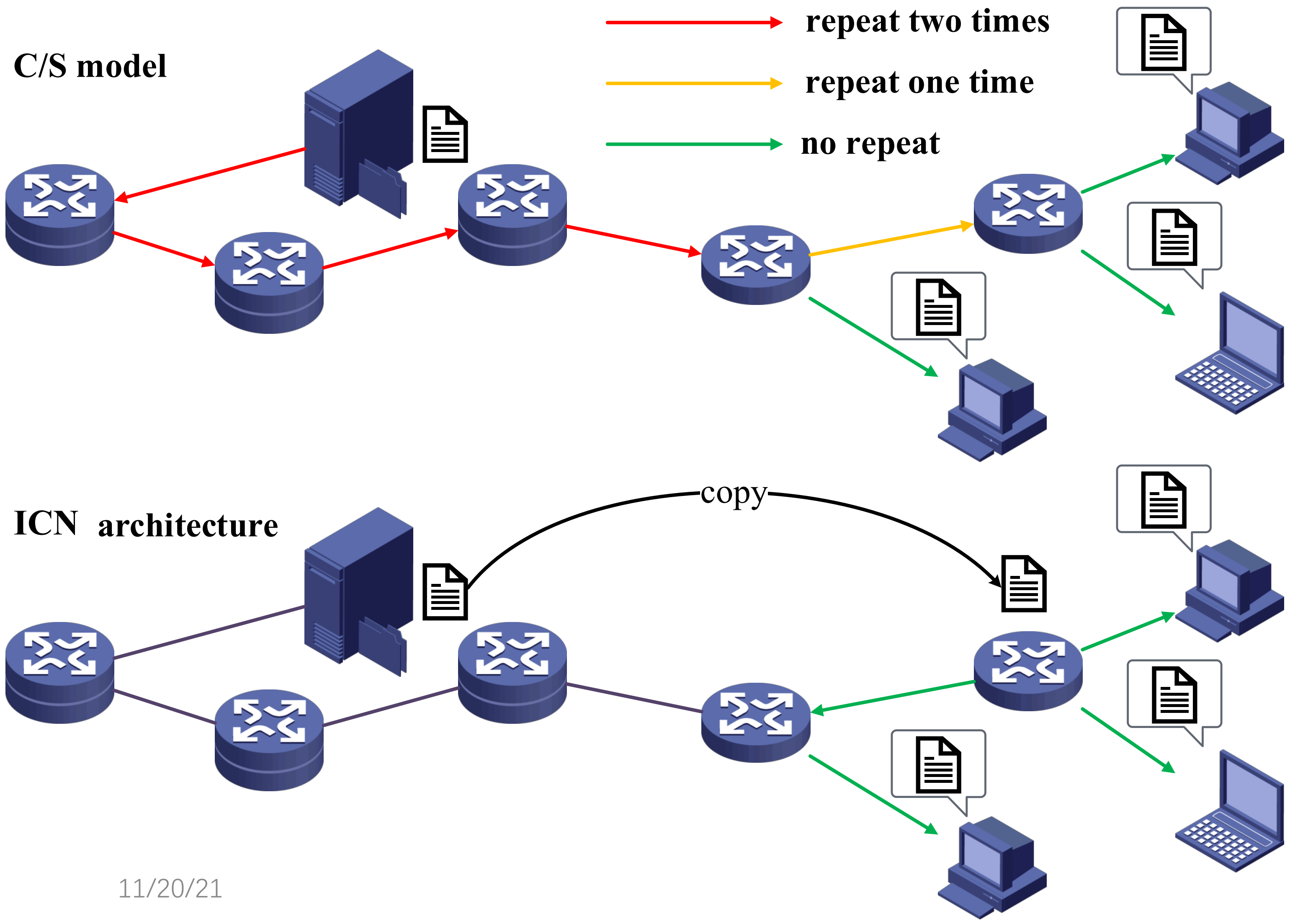
A Genetic-Algorithm-Based In-network Caching for Information-Centric Networking

Authors: Fan Yang, **Zerui Tian**

CONTENTS

- Background
- Mathematical Model
- Details of MRPGA
- Simulation & Conclusion
- Q&A

BACKGROUND: Information-Centric Networking (ICN)

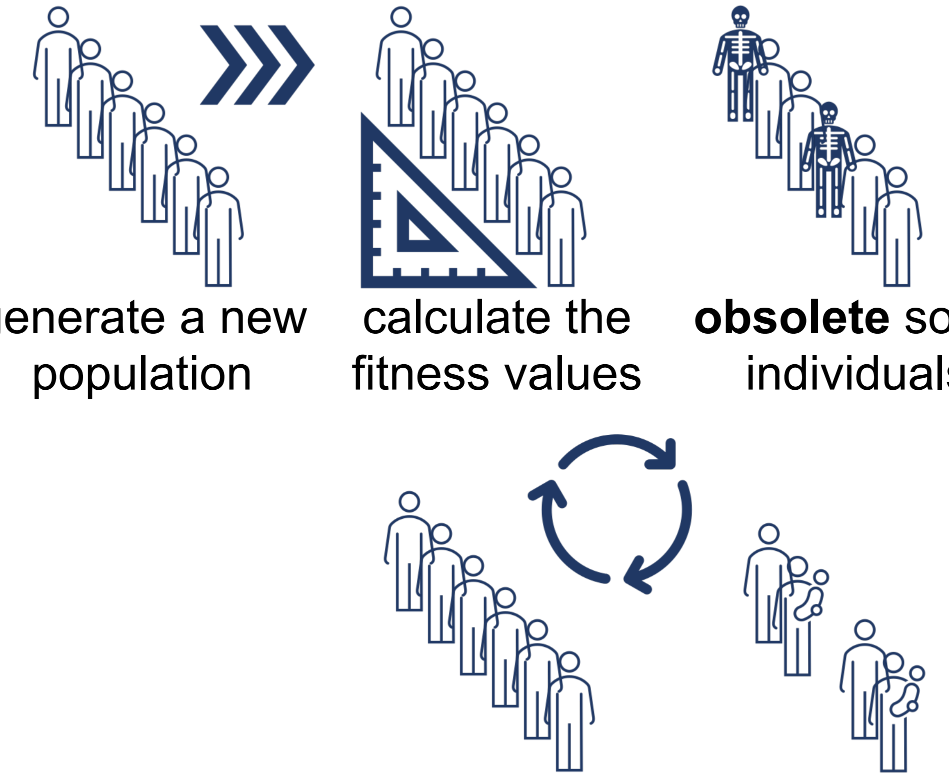


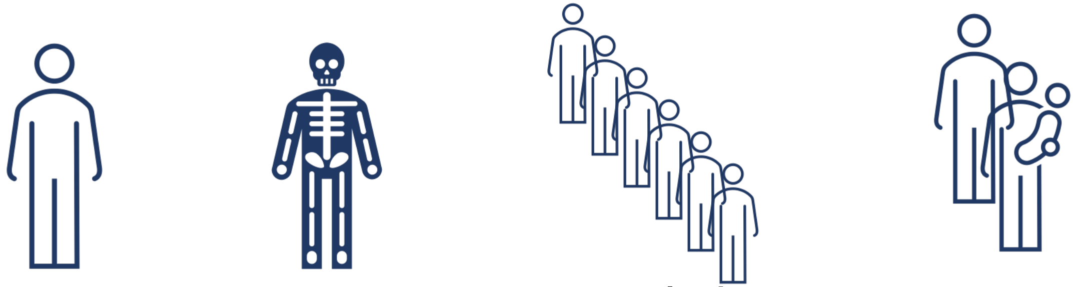
Classic Distributive Cache Strategies

- LCE
- Betw
- ProbCache

Centralized Cache Allocation is an NP-hard 0-1 programming problem

BACKGROUND: Genetic Algorithm (GA)





Algorithm 1: Genetic Algorithm

Input: Iteration time $n_{iteration}$, the size of population $n_{individual}$, mutation rate $p_{mutation}$

Output: The best individual $winner$

Create a random population $population$

for i=1:$n_{iteration}$ **do**

  $survival$ = Obslete($population$)

  $population$ = Produce($survival$,$n_{iteration}$,$p_{mutation}$)

end

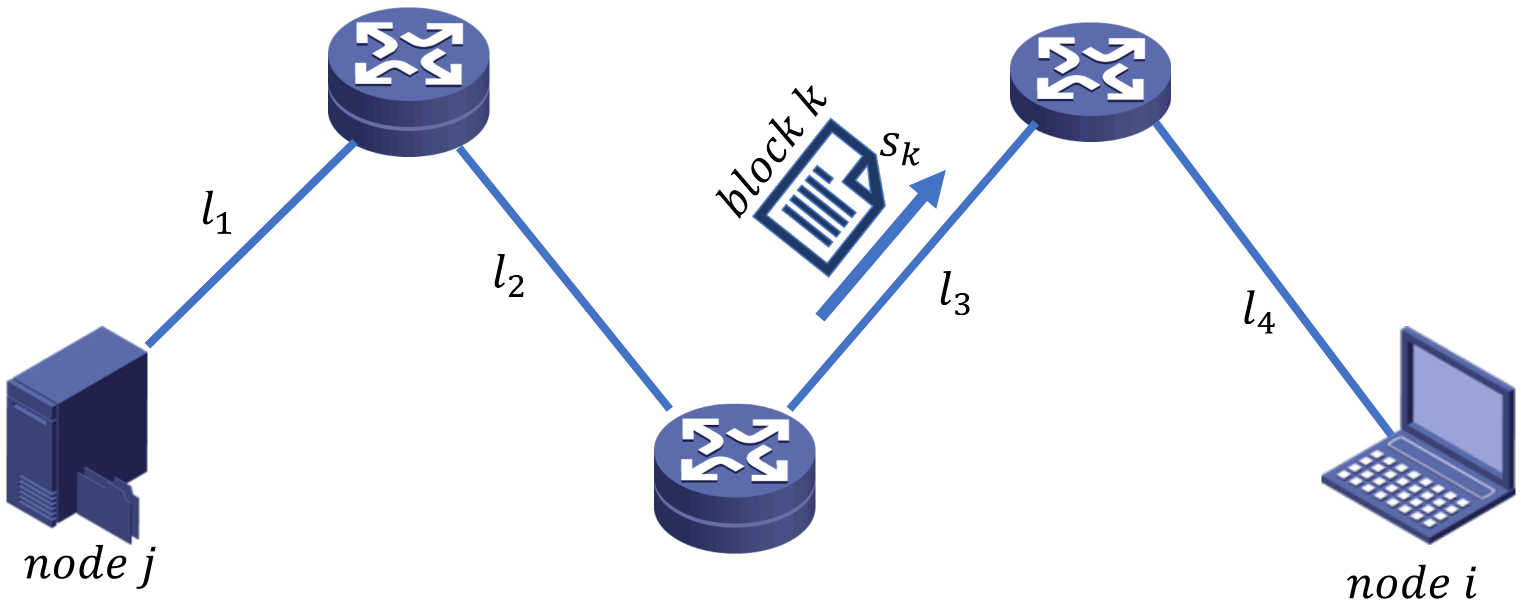
return $winner$=*The best individual in population*

MATHEMATICAL MODEL

Symbol	Name	Description
G	Topology graph	The topology of network, content routers -> nodes, links -> edges
D_{n*n}	Distance Matrix	d_{ij} = the length of the shortest path between node i and node j
C_{m*n}	Cache Matrix	c_{ij} = whether node j caches block i
v_{n*1}	Volume vector	v_i = the residual volume of node i
F_{m*n}	Frequency Matrix	f_{ij} = the request frequency of node j for block i
s_{m*1}	Size vector	s_i = the size of block i

MATHEMATICAL MODEL

the sum of router throughputs caused by the requests of node i for block k toward node j $s_k f_{ki} d_{ij}$

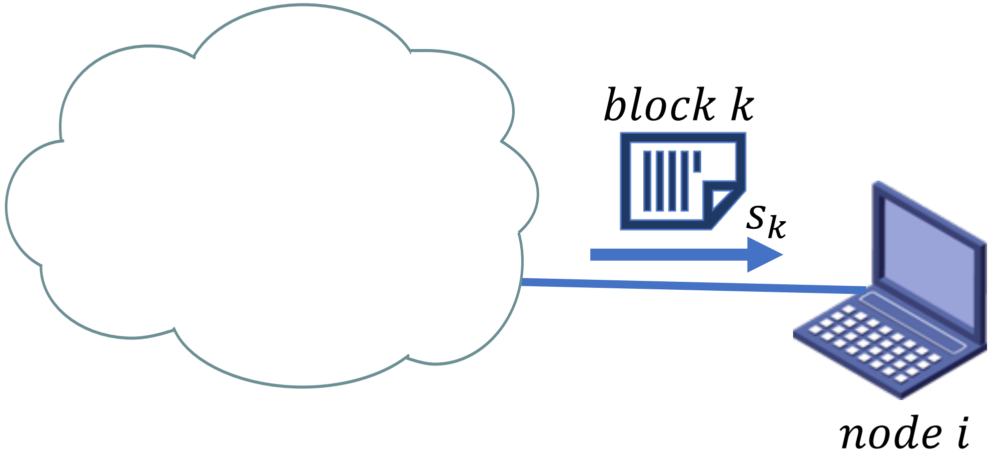


Symbol	Name
G	Topology graph
D_{n*n}	Distance Matrix
C_{m*n}	Cache Matrix
v_{n*1}	Volume vector
F_{m*n}	Frequency Matrix
s_{m*1}	Size vector

MATHEMATICAL MODEL

the sum of router throughputs caused by the requests of node i for block k

$$s_k f_{ki} \min_{j \in \{j | c_{kj} \neq 0\}} d_{ij}$$



Symbol	Name
G	Topology graph
D_{n*n}	Distance Matrix
C_{m*n}	Cache Matrix
v_{n*1}	Volume vector
F_{m*n}	Frequency Matrix
s_{m*1}	Size vector

MATHEMATICAL MODEL

the sum of router throughputs caused by the requests of node i for bock j

$$s_k f_{ki} \min_{j \in \{j | c_{kj} \neq 0\}} d_{ij}$$

the sum of router throughputs caused by all the requests (optimization target)

$$TC = \sum_{(i,k)=(1,1)}^{(n,m)} s_k f_{ki} \min_{j \in \{j | c_{kj} \neq 0\}} d_{ij}$$

Symbol	Name
G	Topology graph
D_{n*n}	Distance Matrix
C_{m*n}	Cache Matrix
v_{n*1}	Volume vector
F_{m*n}	Frequency Matrix
s_{m*1}	Size vector

MATHEMATICAL MODEL

the sum of router throughputs caused by the requests of node i for bock j

$$s_k f_{ki} \min_{j \in \{j | c_{kj} \neq 0\}} d_{ij}$$

the sum of router throughputs caused by all the requests (optimization target)

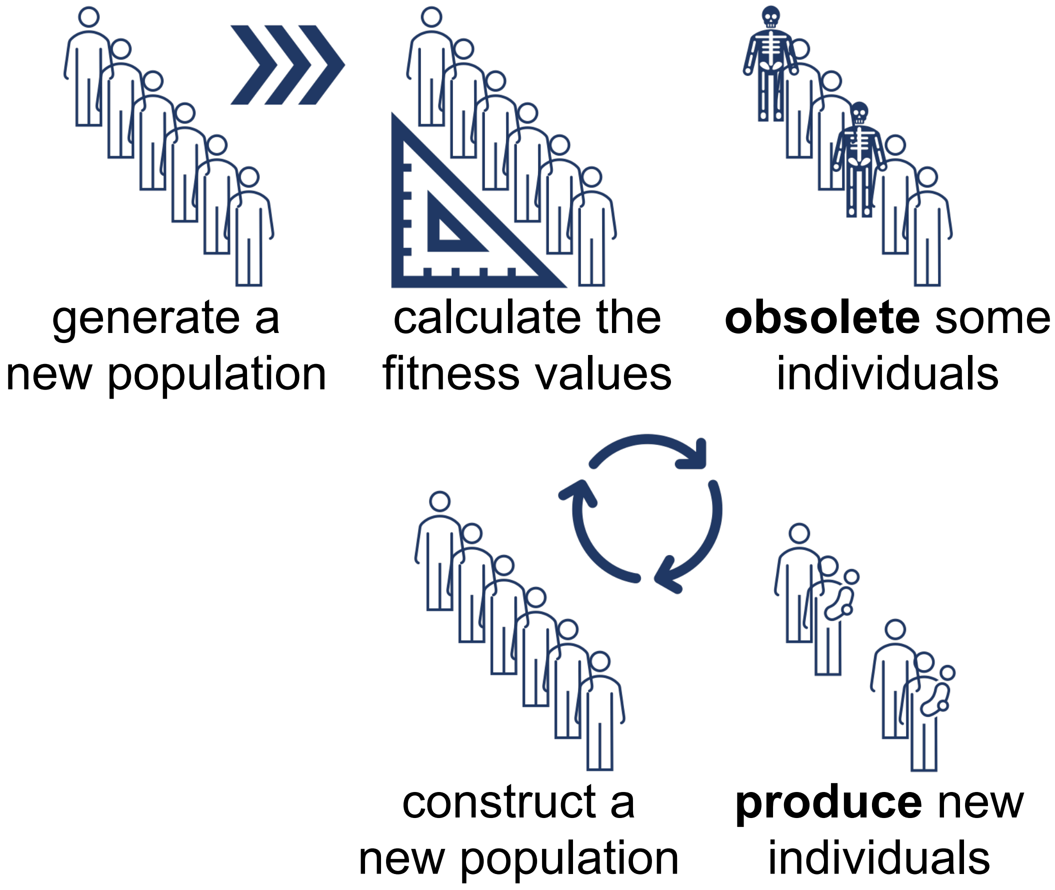
$$TC = \sum_{(i,k)=(1,1)}^{(n,m)} s_k f_{ki} \min_{j \in \{j | c_{kj} \neq 0\}} d_{ij}$$

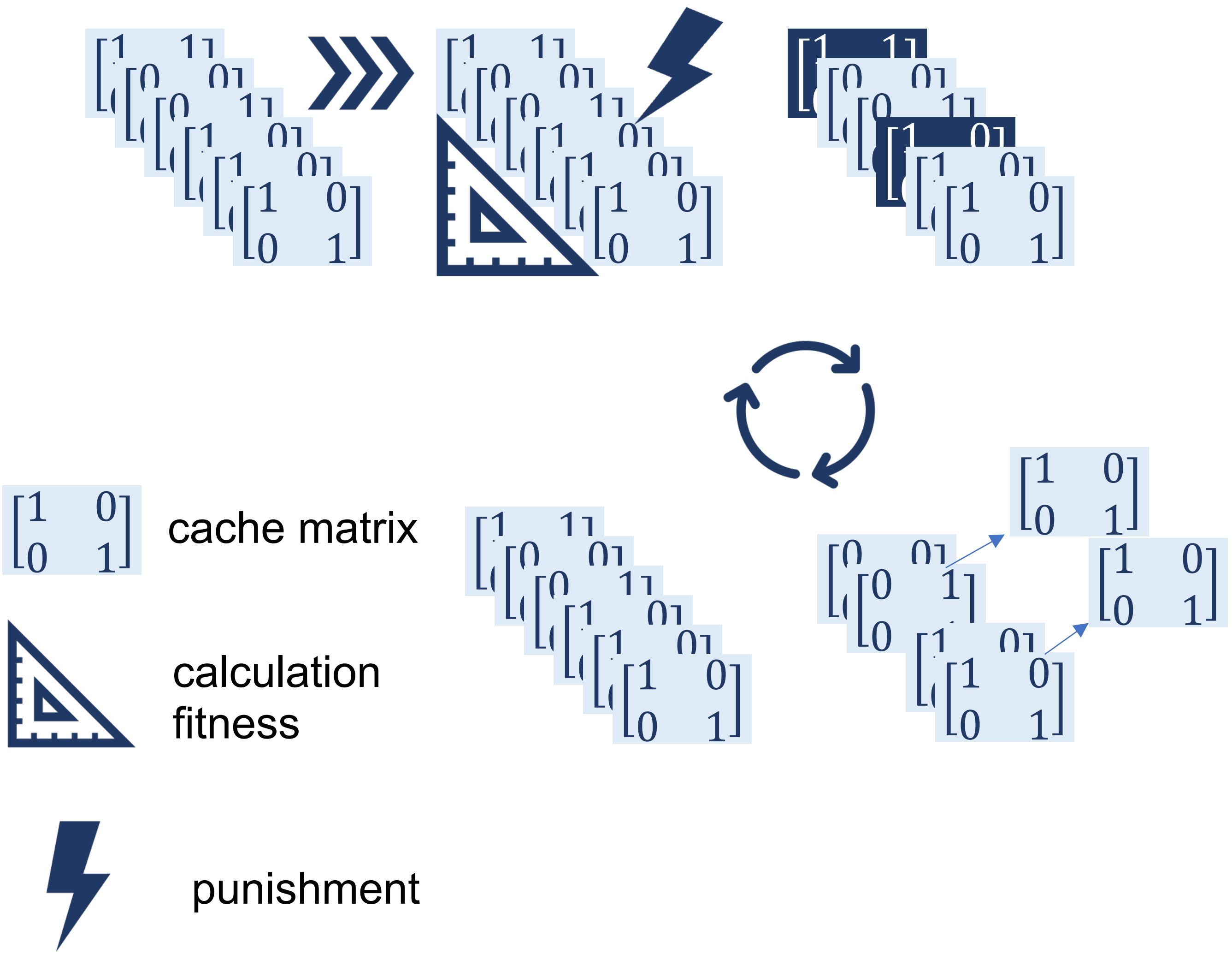
mathematical model of cache allocation problem

$$\begin{cases} C_{opt} = \arg\min_{C} \left[\sum_{(i,k)=(1,1)}^{(n,m)} s_k f_{ki} \min_{j \in \{j | c_{kj} \neq 0\}} d_{ij} \right] \\ s.t. (s^T C)_i \leq v_i, i = 1,2, \dots, n \end{cases}$$

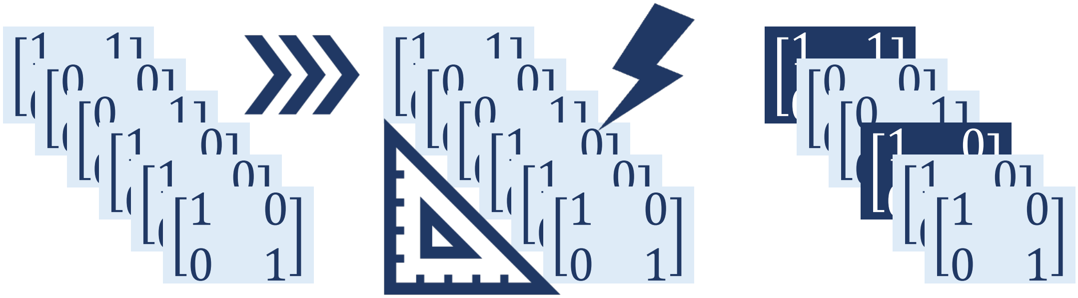
Symbol	Name
G	Topology graph
D_{n*n}	Distance Matrix
C_{m*n}	Cache Matrix
v_{n*1}	Volume vector
F_{m*n}	Frequency Matrix
s_{m*1}	Size vector

DETAILS OF MRPGA: GA APPLICATION ON CACHING

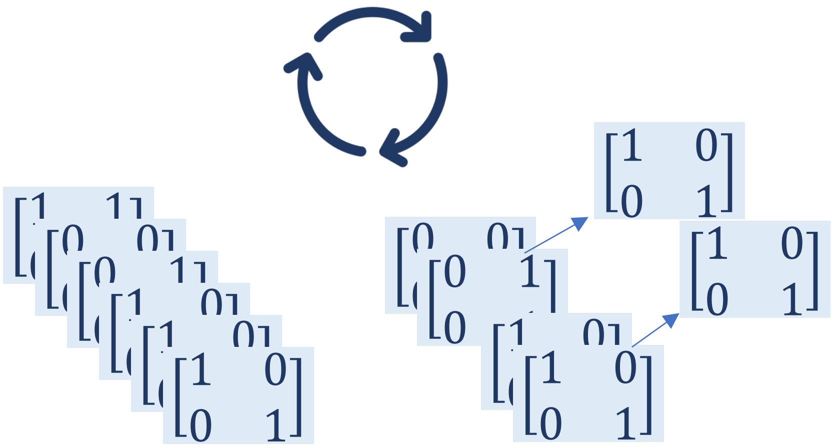




DETAILS OF MRPGA: GA APPLICATION ON CACHING



$$fitness(C) = \begin{cases} \left(\sum_{(i,k)=(1,1)}^{(n,m)} s_k f_{ki} \min_{j \in \{j | c_{kj} \neq 0\}} d_{ij} \right)^{-1}, (s^T C)_i \leq v_i, i = 1,2, \ldots, n \\ 0, otherwise \end{cases}$$



DETAILS OF MRPGA: PROBLEM DECOMPOSITION

$$\begin{cases} C_{opt} = \arg\min\limits_{C} \left[\sum\limits_{(i,k)=(1,1)}^{(n,m)} s_k f_{ki} \min\limits_{j \in \{j | c_{kj} \neq 0\}} d_{ij} \right] \\ s.t. (s^T C)_i \leq v_i, i = 1,2,\dots,n \end{cases}$$

$$\begin{cases} (c_1)_{opt} = \arg\min\limits_{c_1} \left[\sum\limits_{i=1}^{n} s_1 f_{1i} \min\limits_{j \in \{j | c_{1j} \neq 0\}} d_{ij} \right] \\ s.t. s_1 c_{1i} \leq v_i, i = 1,2,\dots,n \end{cases}$$

for block 1 allocation

$$\begin{cases} (c_2)_{opt} = \arg\min\limits_{c_2} \left[\sum\limits_{i=1}^{n} s_2 f_{2i} \min\limits_{j \in \{j | c_{2j} \neq 0\}} d_{ij} \right] \\ s.t. s_2 c_{2i} \leq v_i, i = 1,2,\dots,n \end{cases}$$

for block 2 allocation

$$\vdots$$

$$\begin{cases} (c_m)_{opt} = \arg\min\limits_{c_m} \left[\sum\limits_{i=1}^{n} s_m f_{mi} \min\limits_{j \in \{j | c_{mj} \neq 0\}} d_{ij} \right] \\ s.t. s_m c_{mi} \leq v_i, i = 1,2,\dots,n \end{cases}$$

for block m allocation

DETAILS OF MRPGA: FEASIBLE SOLUTION GUARANTEE

for block k allocation

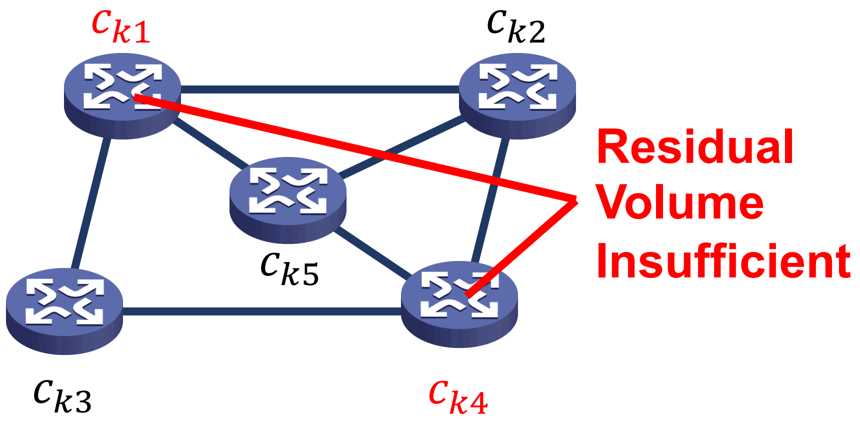
$$\begin{cases} (c_k)_{opt} = \arg\min\limits_{c_k} \left[\sum\limits_{i=1}^{n} s_k f_{ki} \min\limits_{j \in \{j | c_{kj} \neq 0\}} d_{ij} \right] \\ \boxed{s_k \leq v_i, i \in \{i | c_{ki} = 1\}} \end{cases} \rightarrow s_k \leq v_i, i \in \{i | c_{ki} = 1\}$$

TARGET: remove the limit

DETAILS OF MRPGA: FEASIBLE SOLUTION GUARANTEE

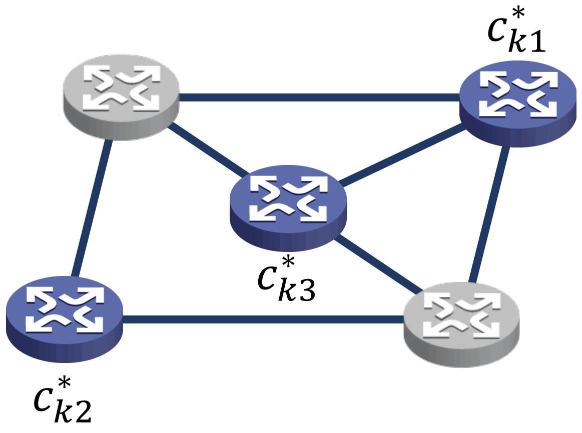
for block k allocation

$$\begin{cases} (c_k)_{opt} = \underset{c_k}{\arg\min} \left[\sum_{i=1}^{n} s_k f_{ki} \min_{j \in \{j | c_{kj} \neq 0\}} d_{ij} \right] \\ s_k \leq v_i, i \in \{i | c_{ki} = 1\} \end{cases}$$



$$c_k = \begin{bmatrix} c_{k1} \\ c_{k2} \\ c_{k3} \\ c_{k4} \\ c_{k5} \end{bmatrix}$$

$$(c_k^*)_{opt} = \underset{c_k^*}{\arg\min} \left[\sum_{i=1}^{n} s_k f_{ki} \min_{j \in \{j | c_{kj}^* \neq 0\}} d_{ij} \right]$$

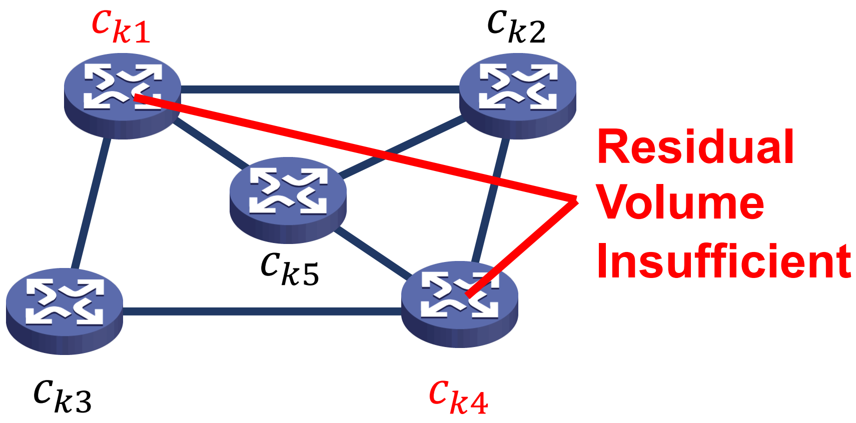


$$c_k^* = \begin{bmatrix} c_{k1}^* \\ c_{k2}^* \\ c_{k3}^* \end{bmatrix}$$

DETAILS OF MRPGA: FEASIBLE SOLUTION GUARANTEE

for block k allocation

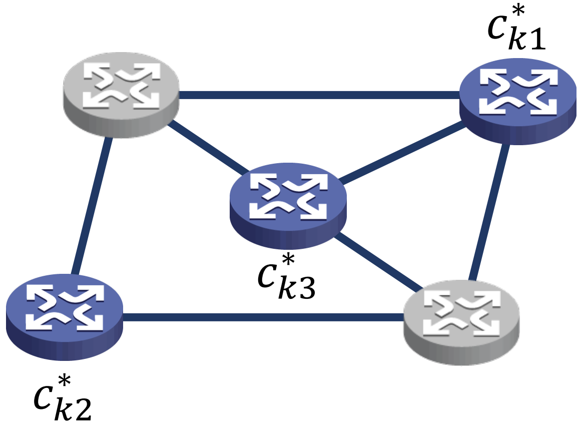
$$\begin{cases} (c_k)_{opt} = \arg\min\limits_{c_k} \left[\sum\limits_{i=1}^{n} s_k f_{ki} \min\limits_{j \in \{j | c_{kj} \neq 0\}} d_{ij} \right] \\ s_k \leq v_i, i \in \{i | c_{ki} = 1\} \end{cases}$$



$$c_k = \begin{bmatrix} c_{k1} \\ c_{k2} \\ c_{k3} \\ c_{k4} \\ c_{k5} \end{bmatrix}$$

$$(c_k^*)_{opt} = \arg\min\limits_{c_k^*} \left[\sum\limits_{i=1}^{n} s_k f_{ki} \min\limits_{j \in \{j | c_{kj}^* \neq 0\}} d_{ij} \right]$$

the limit has been removed



$$c_k^* = \begin{bmatrix} c_{k1}^* \\ c_{k2}^* \\ c_{k3}^* \end{bmatrix}$$

the lengths of individuals are shortened

DETAILS OF MRPGA: SOLUTION COMBINATION

$$(c_1^*)_{opt} = \arg\min_{c_1^*} \left[\sum_{i=1}^{n} s_1 f_{1i} \min_{j \in \{j | c_{1j}^* \neq 0\}} d_{ij} \right] \longrightarrow (c_1)_{opt}$$

$$(c_2^*)_{opt} = \arg\min_{c_2^*} \left[\sum_{i=1}^{n} s_2 f_{2i} \min_{j \in \{j | c_{2j}^* \neq 0\}} d_{ij} \right] \longrightarrow (c_2)_{opt}$$

⋮

$$(c_m^*)_{opt} = \arg\min_{c_m^*} \left[\sum_{i=1}^{n} s_m f_{mi} \min_{j \in \{j | c_{mj}^* \neq 0\}} d_{ij} \right] \longrightarrow (c_m)_{opt}$$

?

C_{opt}

DETAILS OF MRPGA: SOLUTION COMBINATION

$$(c_1^*)_{opt} = \arg\min_{c_1^*}\left[\sum_{i=1}^{n} s_1 f_{1i} \min_{j\in\{j|c_{1j}^*\neq 0\}} d_{ij}\right] \longrightarrow (c_1)_{opt}$$

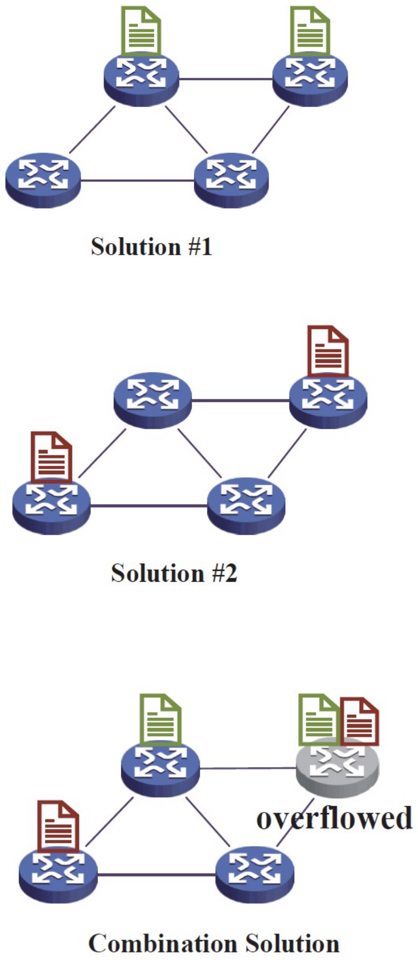
$$(c_2^*)_{opt} = \arg\min_{c_2^*}\left[\sum_{i=1}^{n} s_2 f_{2i} \min_{j\in\{j|c_{2j}^*\neq 0\}} d_{ij}\right] \longrightarrow (c_2)_{opt}$$

$\vdots$

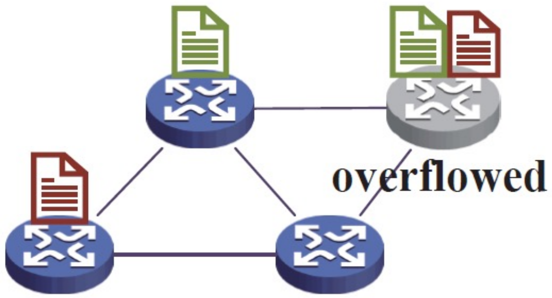
$$(c_m^*)_{opt} = \arg\min_{c_m^*}\left[\sum_{i=1}^{n} s_m f_{mi} \min_{j\in\{j|c_{mj}^*\neq 0\}} d_{ij}\right] \longrightarrow (c_m)_{opt}$$

?

$$C_{opt}$$

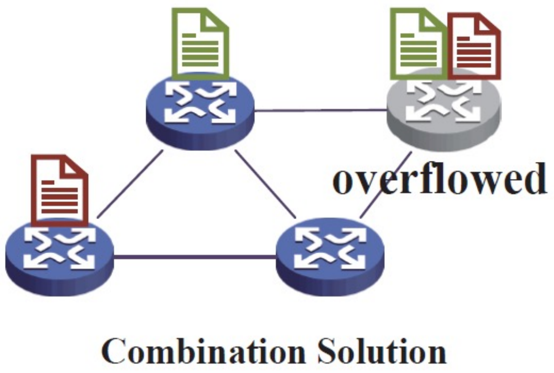


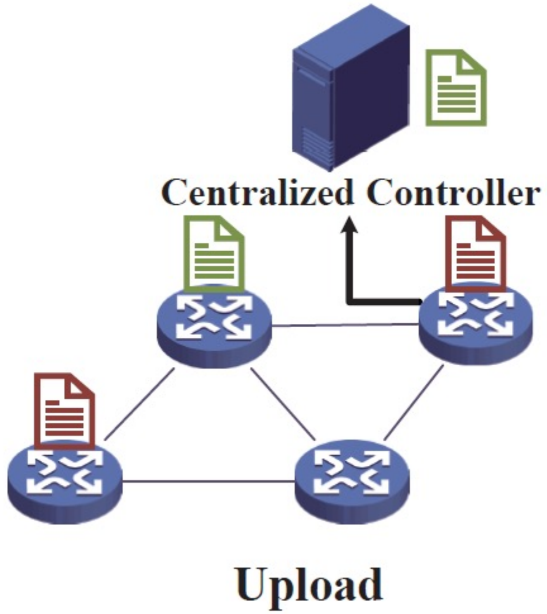
DETAILS OF MRPGA: SOLUTION COMBINATION



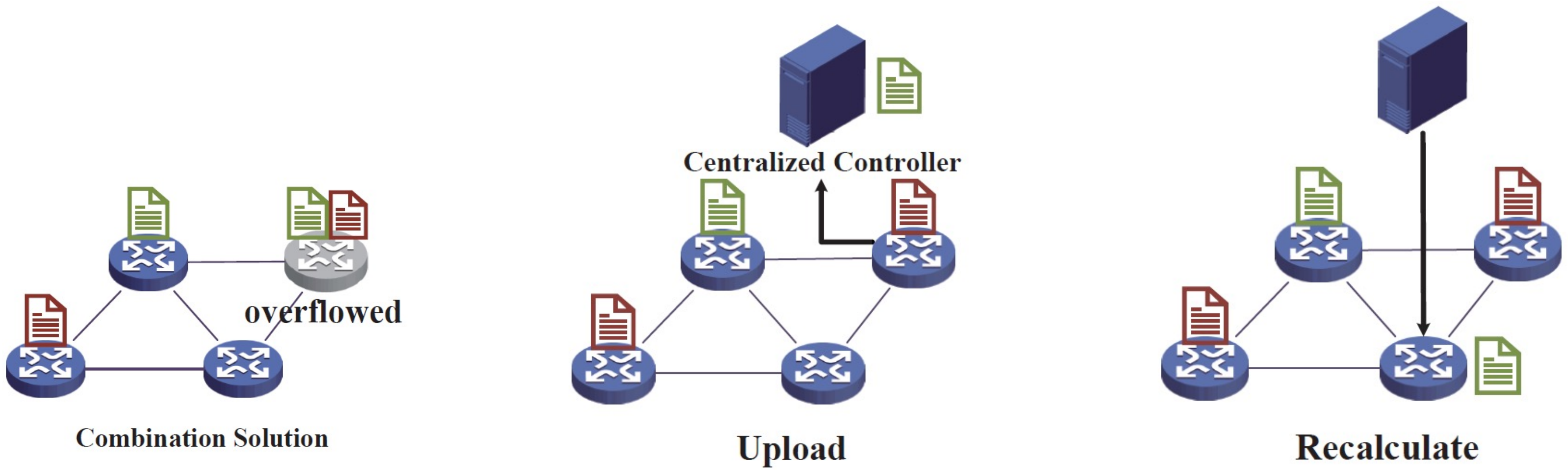
Combination Solution

DETAILS OF MRPGA: SOLUTION COMBINATION





DETAILS OF MRPGA: SOLUTION COMBINATION



DETAILS OF MRPGA：AMOUNT PREALLOCATION

Algorithm 4: Multiple-Round Parallel Genetic Algorithm

Input: The adjustable parameter K, size of population $n_{individual}$, mutation rate $p_{mutation}$, iteration time $n_{iteration}$

Output: A feasible solution C

Calculate the amount of every category of blocks perviously with K and obtain the amount vector a

while $\sum_{i=1}^{m} a_i \neq 0$ **do**

  for i = $1{:}m$ **do**

    if $a_i \neq 0$ **then**

      if $\exists j \in \{1, 2, \ldots, n\}, v_j \geq s_i, c_{ij} = 0$ **then**

        $(C_{recommand})_i$= GeneticAlgorithm(a_i,$n_{individual}$,$p_{mutation}$,$n_{iteration}$)

      end

      else

        $a_i = 0$

      end

    end

  end

  Distribute $C_{recommand}$ to the date plane

  Update C,v,a

end

return C

DETAILS OF MRPGA: AMOUNT PREALLOCATION

Algorithm 4: Multiple-Round Parallel Genetic Algorithm

Input: The adjustable parameter K, size of population $n_{individual}$, mutation rate $p_{mutation}$, iteration time $n_{iteration}$

Output: A feasible solution C

Calculate the amount of every category of blocks perviously with K and obtain the amount vector a

while $\sum_{i=1}^{m} a_i \neq 0$ **do**
  for i = $1{:}m$ **do**
    if $a_i \neq 0$ **then**
      if $\exists j \in \{1, 2, \dots, n\}, v_j \geq s_i, c_{ij} = 0$ **then**
        $(C_{recommand})_i$= GeneticAlgorithm(a_i,$n_{individual}$,$p_{mutation}$,$n_{iteration}$)
      end
      else
        $a_i = 0$
      end
    end
  end
  Distribute $C_{recommand}$ to the date plane
  Update C,v,a
end
return C

K is an adjustable parameter

$$a_i = \min\left\{K\sum_{j=1}^{n} f_{ij}, h_i\right\}, i = 1 \cdots m$$

$$\sum_{i=1}^{m} a_i s_i < \sum_{i=1}^{n} v_i$$

$$\begin{cases} (c_k^*)_{opt} = \arg\min\limits_{c_k^*}\left[\sum_{i=1}^{n} s_k f_{ki} \min\limits_{j\in\{j|c_{kj}^*\neq 0\}} d_{ij}\right] \\ \boxed{s.t. \sum_{i=1}^{n} c_k^* = a^i} \end{cases}$$

violating this limit does not mean the solution is infeasible

DETAILS OF MRPGA: AMOUNT PREALLOCATION

Algorithm 4: Multiple-Round Parallel Genetic Algorithm

Input: The adjustable parameter K, size of population $n_{individual}$, mutation rate $p_{mutation}$, iteration time $n_{iteration}$

Output: A feasible solution C

Calculate the amount of every category of blocks perviously with K and obtain the amount vector a

while $\sum_{i=1}^{m} a_i \neq 0$ **do**

  for i = $1{:}m$ **do**

    if $a_i \neq 0$ **then**

      if $\exists j \in \{1, 2, \ldots, n\}, v_j \geq s_i, c_{ij} = 0$ **then**

        $(C_{recommand})_i$= GeneticAlgorithm(a_i,$n_{individual}$,$p_{mutation}$,$n_{iteration}$)

      end

      else

        $a_i = 0$

      end

    end

  end

  Distribute $C_{recommand}$ to the date plane

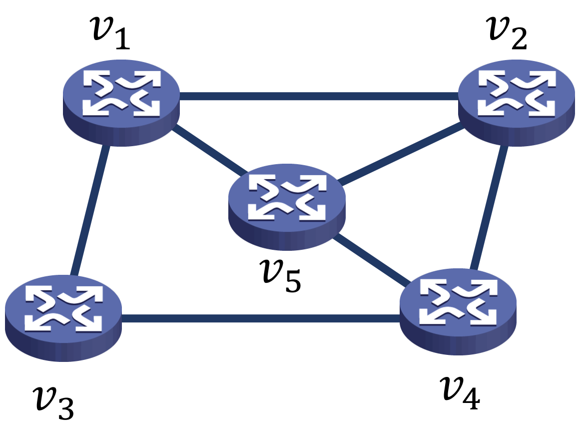
  Update C,v,a

end

return C

$$a_i = \min\left\{K\sum_{j=1}^{n} f_{ij}, h_i\right\}, i = 1 \cdots m$$

$$\sum_{i=1}^{m} a_i s_i < \sum_{i=1}^{n} v_i$$



$$\begin{cases} v_1 < s_i \\ v_2 < s_i \\ v_3 < s_i, but, \sum_{k=1}^{5} v_k \geq s_i \\ v_4 < s_i \\ v_5 < s_i \end{cases}$$

endless loop

DETAILS OF MRPGA: AMOUNT PREALLOCATION

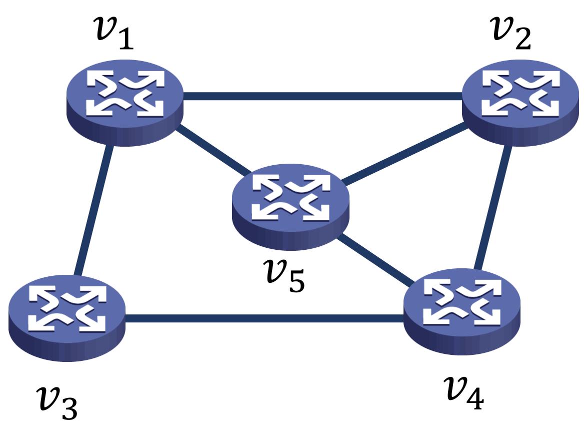
Algorithm 4: Multiple-Round Parallel Genetic Algorithm

Input: The adjustable parameter K, size of population $n_{individual}$, mutation rate $p_{mutation}$, iteration time $n_{iteration}$
Output: A feasible solution C

Calculate the amount of every category of blocks perviously with K and obtain the amount vector a
while $\sum_{i=1}^{m} a_i \neq 0$ **do**
  for i = $1{:}m$ **do**
    if $a_i \neq 0$ **then**
      if $\exists j \in \{1, 2, \ldots, n\}, v_j \geq s_i, c_{ij} = 0$ **then**
        $(C_{recommand})_i$= GeneticAlgorithm(a_i,$n_{individual}$,$p_{mutation}$,$n_{iteration}$)
      end
      else
        $\boxed{a_i = 0}$
      end
    end
  end
  Distribute $C_{recommand}$ to the date plane
  Update C,v,a
end
return C

$$a_i = \min\left\{K\sum_{j=1}^{n} f_{ij}, h_i\right\}, i = 1 \cdots m$$

$$\boxed{\sum_{i=1}^{m} a_i s_i < \sum_{i=1}^{n} v_i}$$

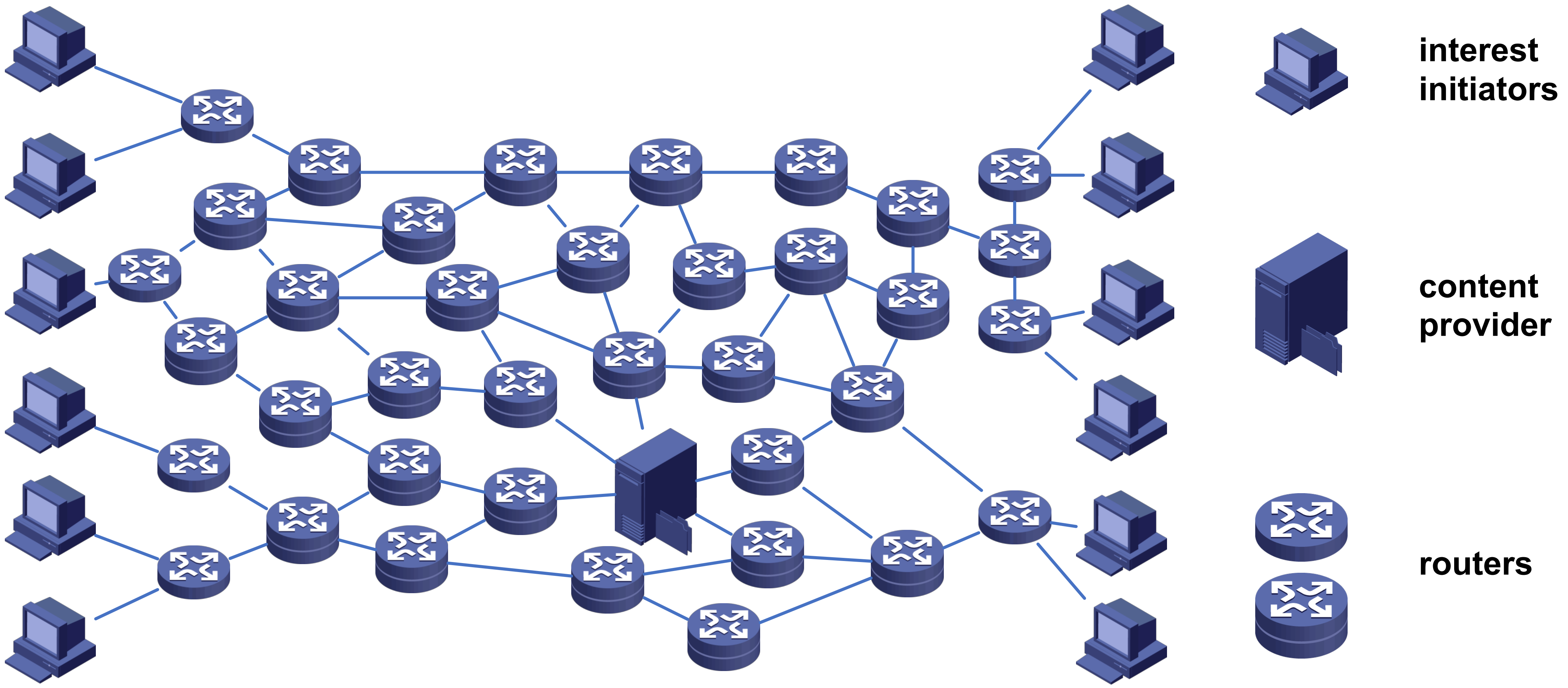


$$\begin{cases} v_1 < s_i \\ v_2 < s_i \\ v_3 < s_i, but, \sum_{k=1}^{5} v_k \geq s_i \\ v_4 < s_i \\ v_5 < s_i \end{cases}$$

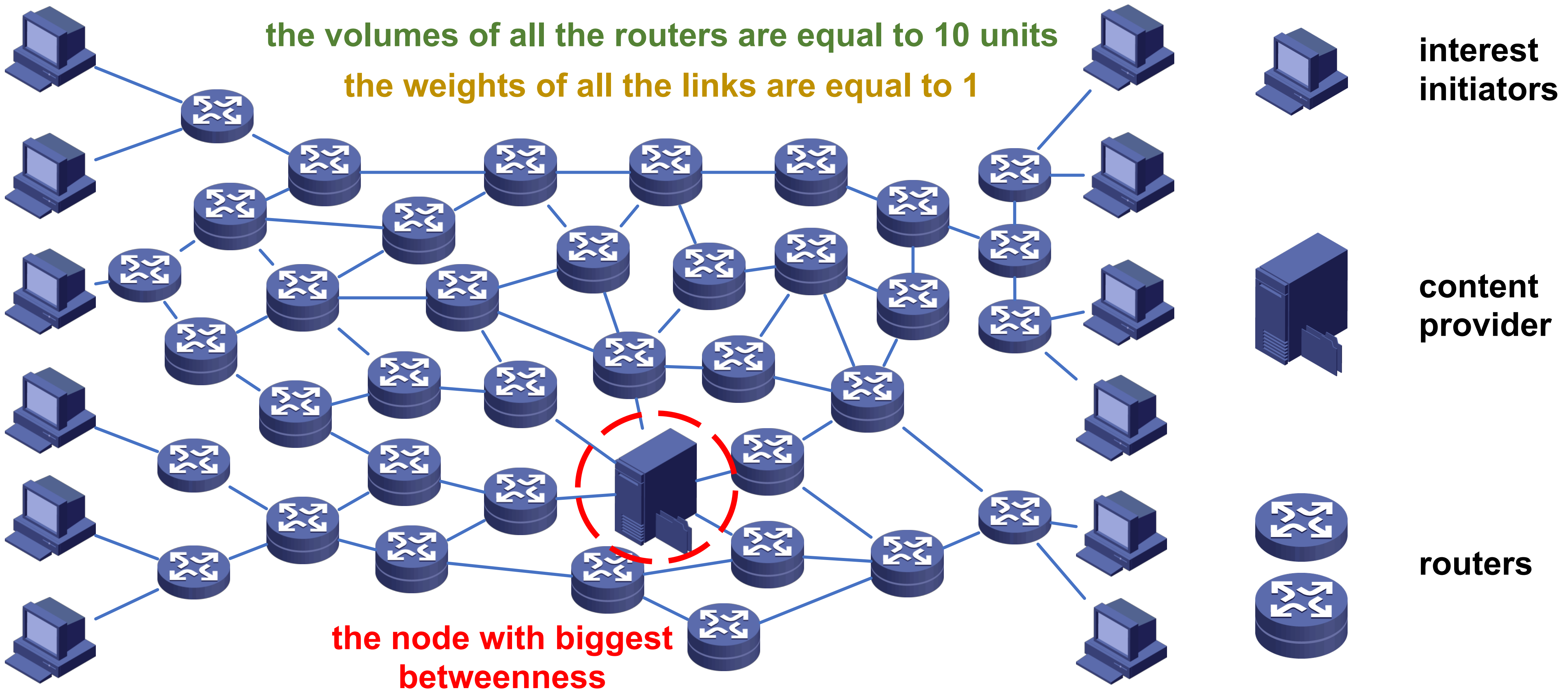
endless loop

If there is no router can cache block i, then the algorithm should not try to distribute block i anymore

SIMULATION: CONDITION



SIMULATION: CONDITION



SIMULATION: CONDITION

Number	Size	Number	Size	Number	Size
#1	1	#6	2	#11	3
#2	1	#7	2	#12	5
#3	1	#8	2	#13	5
#4	1	#9	2		
#5	1	#10	3		

$$p(node = i, block = j) = \begin{cases} \frac{C}{j^{\alpha}}, if\ node\ i\ is\ an\ interest\ initiator \\ 0, otherwise \end{cases}$$

zipf's distribution

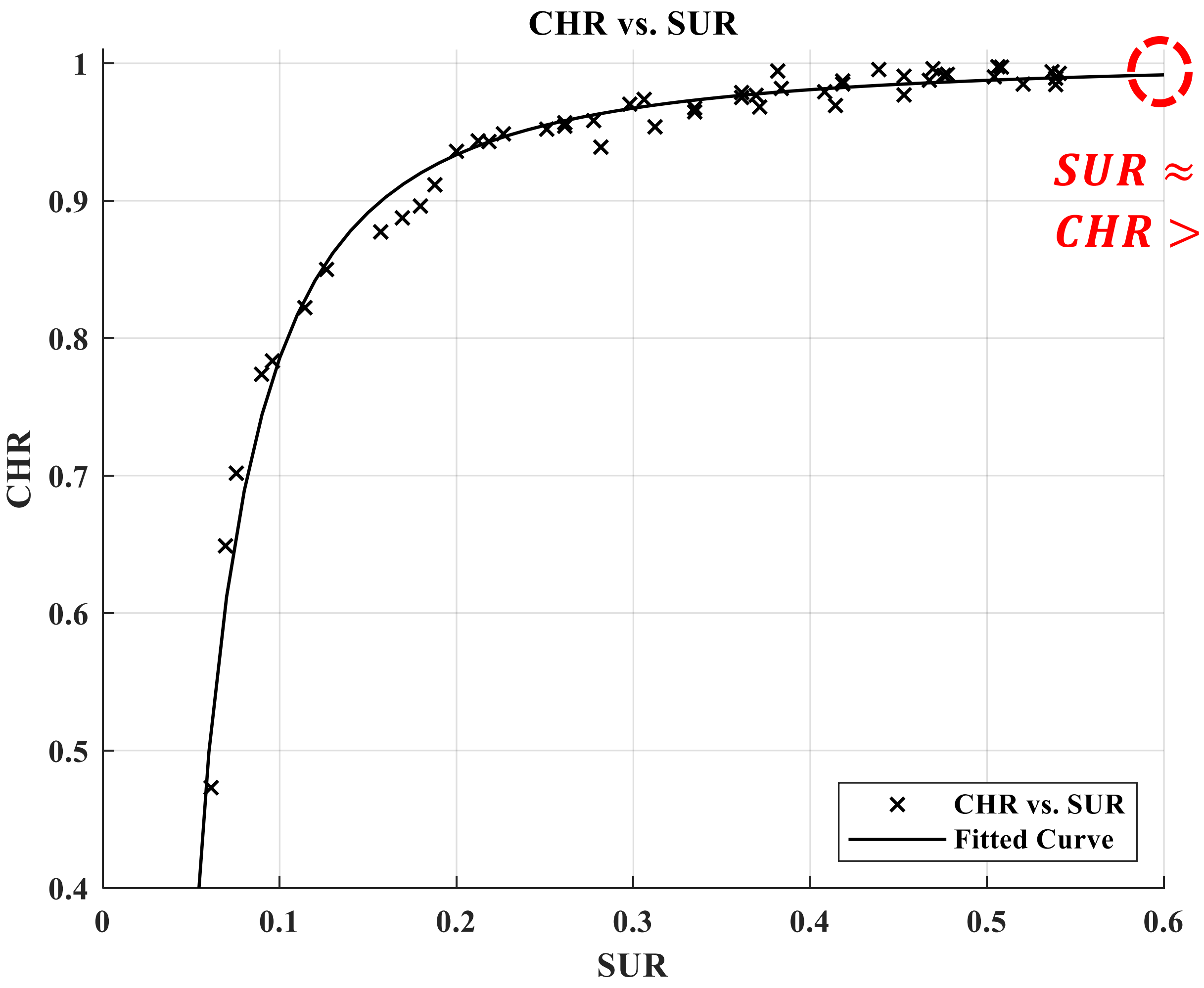
$$\sum_{(i,j)=(1,1)}^{(n,m)} p(i,j) = 1$$

SIMULATION: PARAMETER K

parameter K

$$SUR = \frac{\sum_{i=1}^{n}(s^T C)_i}{\sum_{i=1}^{n} v_i}$$

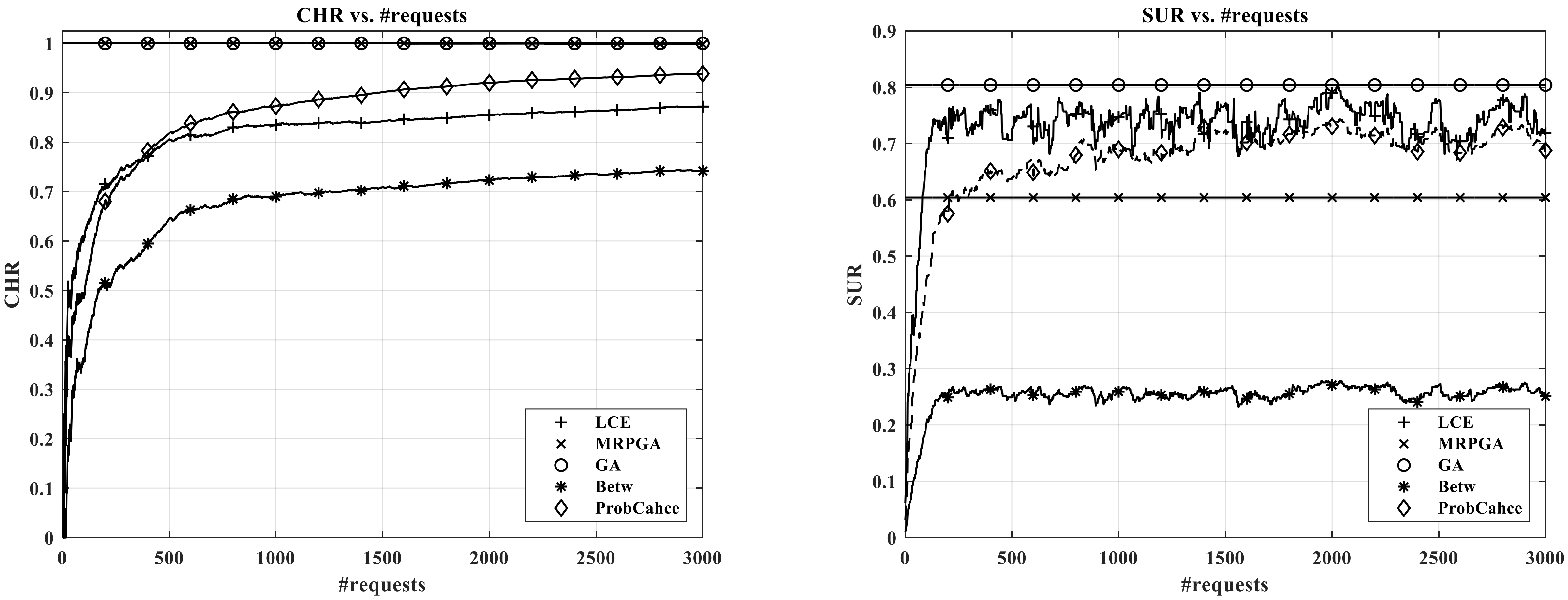
$$CHR = \frac{\#hits}{\#requests}$$



$SUR \approx 0.6$

$CHR > 0.99$

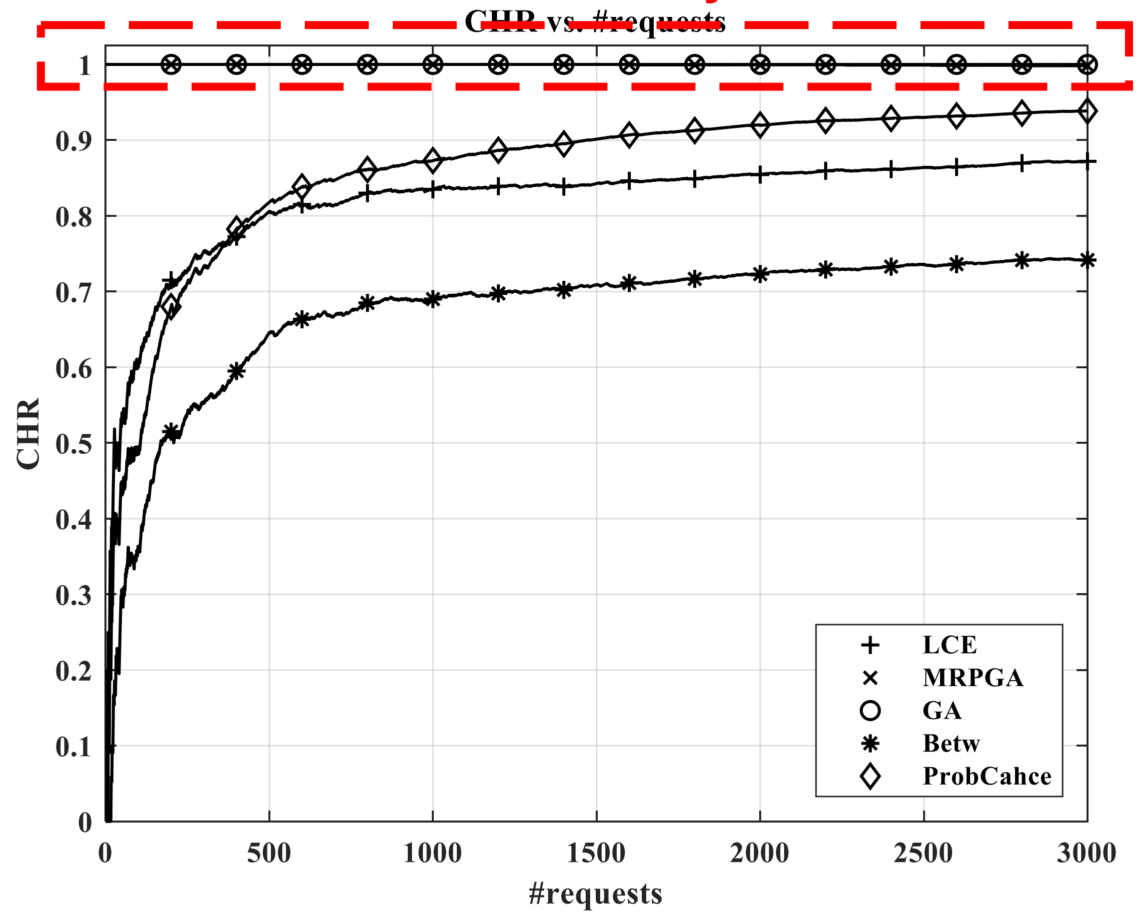
SIMULATION: CHR & SUR



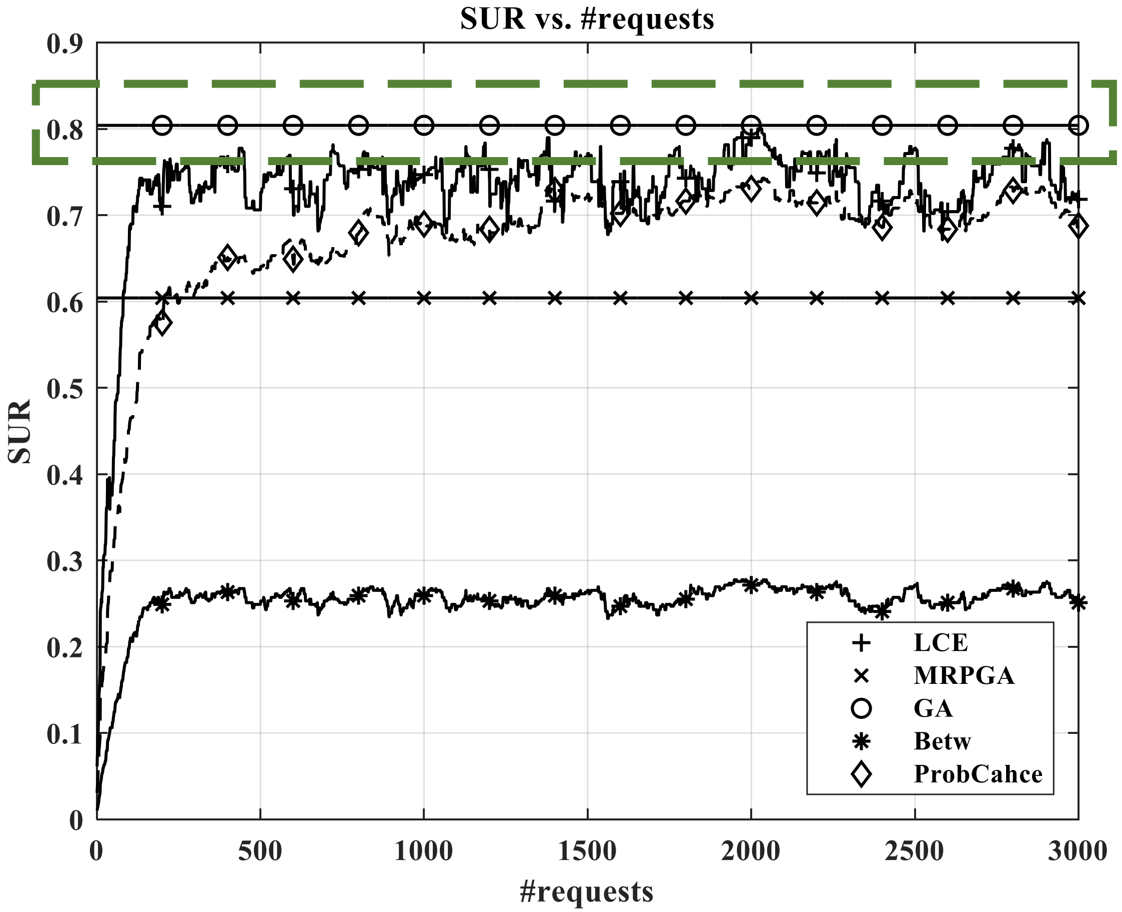
the total iteration times of GA and MRPGA are same

SIMULATION: CHR & SUR

almost hit every time

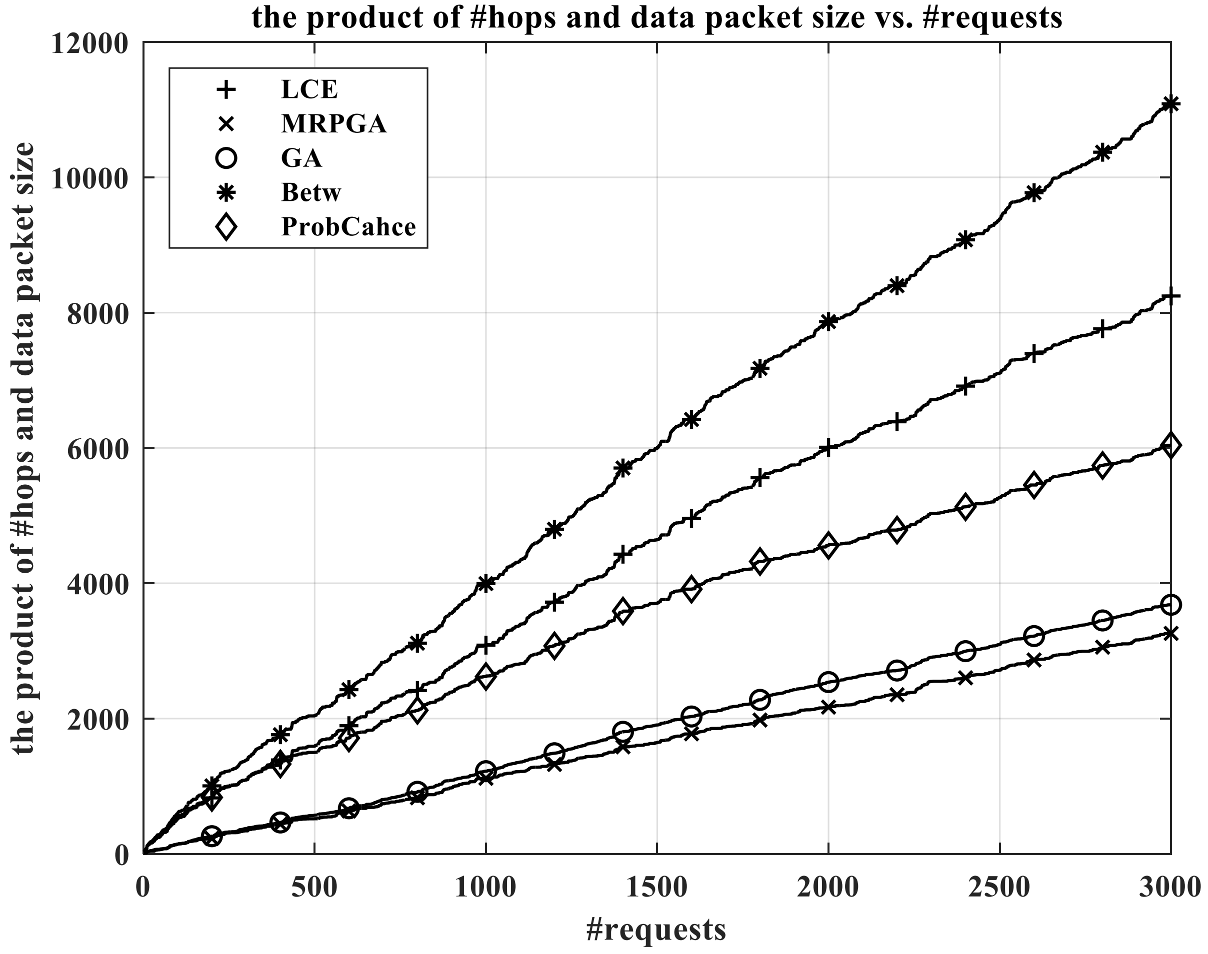


SUR of GA is around 0.8



the total iteration times of GA and MRPGA are same

SIMULATION: OPTIMIZATION TARGET



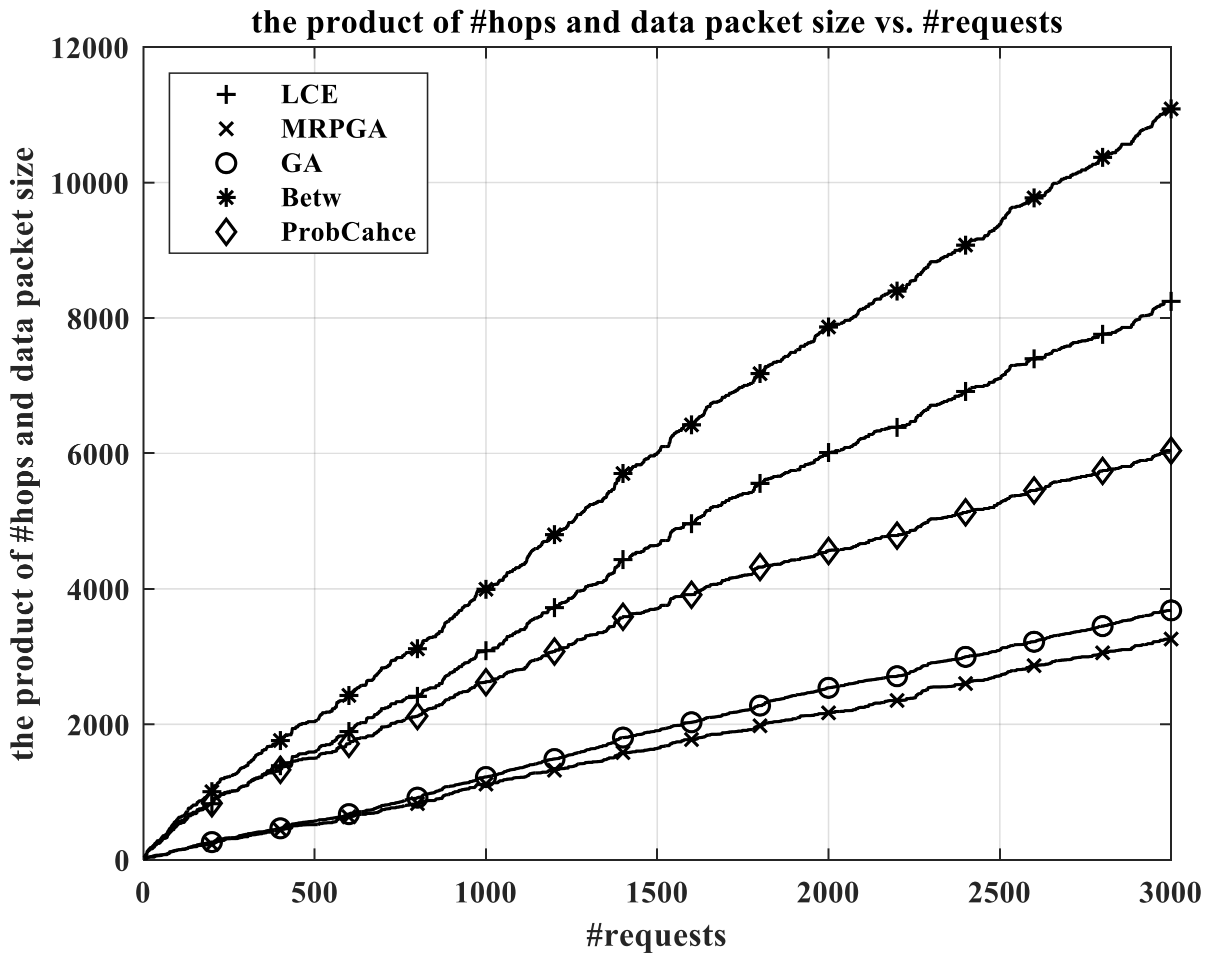
$$TC = \sum_{(i,k)=(1,1)}^{(n,m)} s_k f_{ki} \min_{j\in\{j|c_{kj}\neq 0\}} d_{ij}$$

$$AH = \frac{TC}{\#requests}$$

Algorithm	AH
MRPGA	1.0870
GA	1.2520
ProbCache	1.9003
LCE	2.7315
Betw	3.6612

the total iteration times of GA and MRPGA are same

SIMULATION: OPTIMIZATION TARGET



$$TC = \sum_{(i,k)=(1,1)}^{(n,m)} s_k f_{ki} \min_{j\in\{j|c_{kj}\neq 0\}} d_{ij}$$

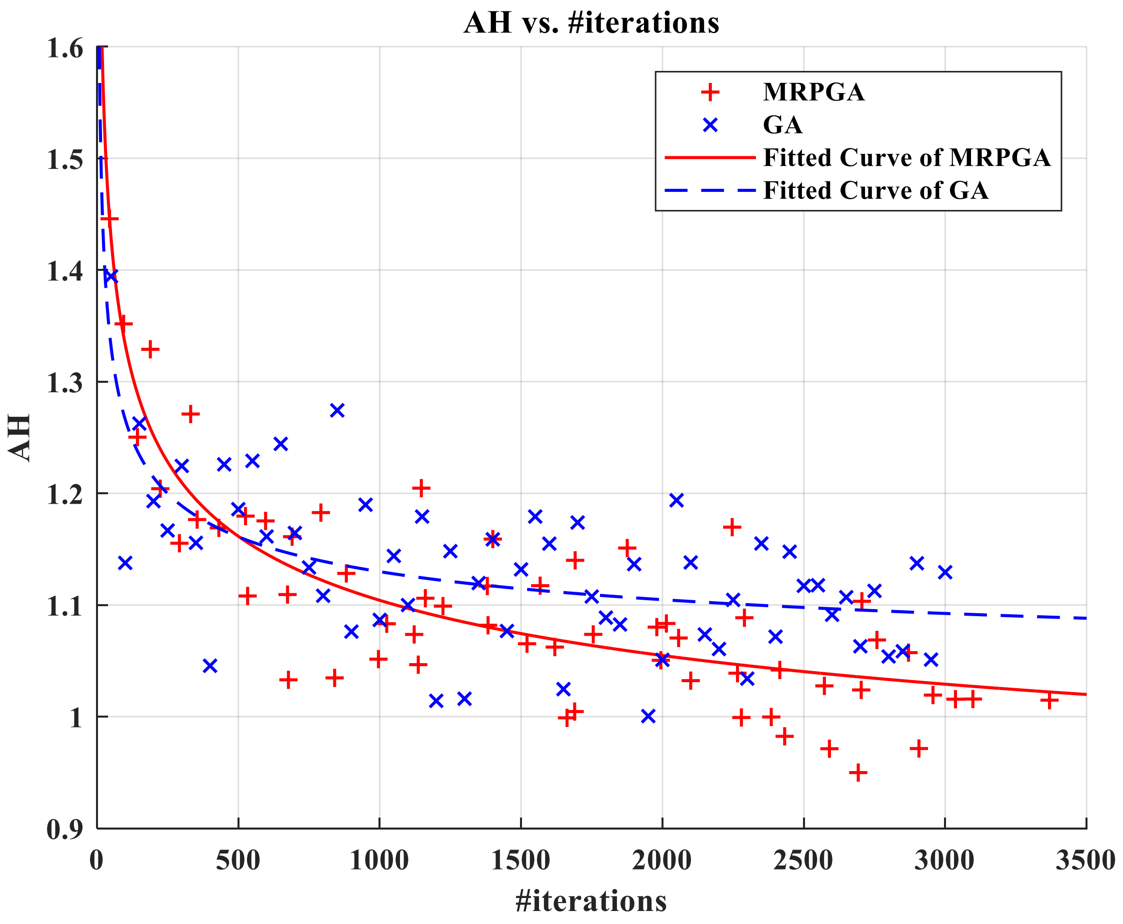
$$AH = \frac{TC}{\#requests}$$

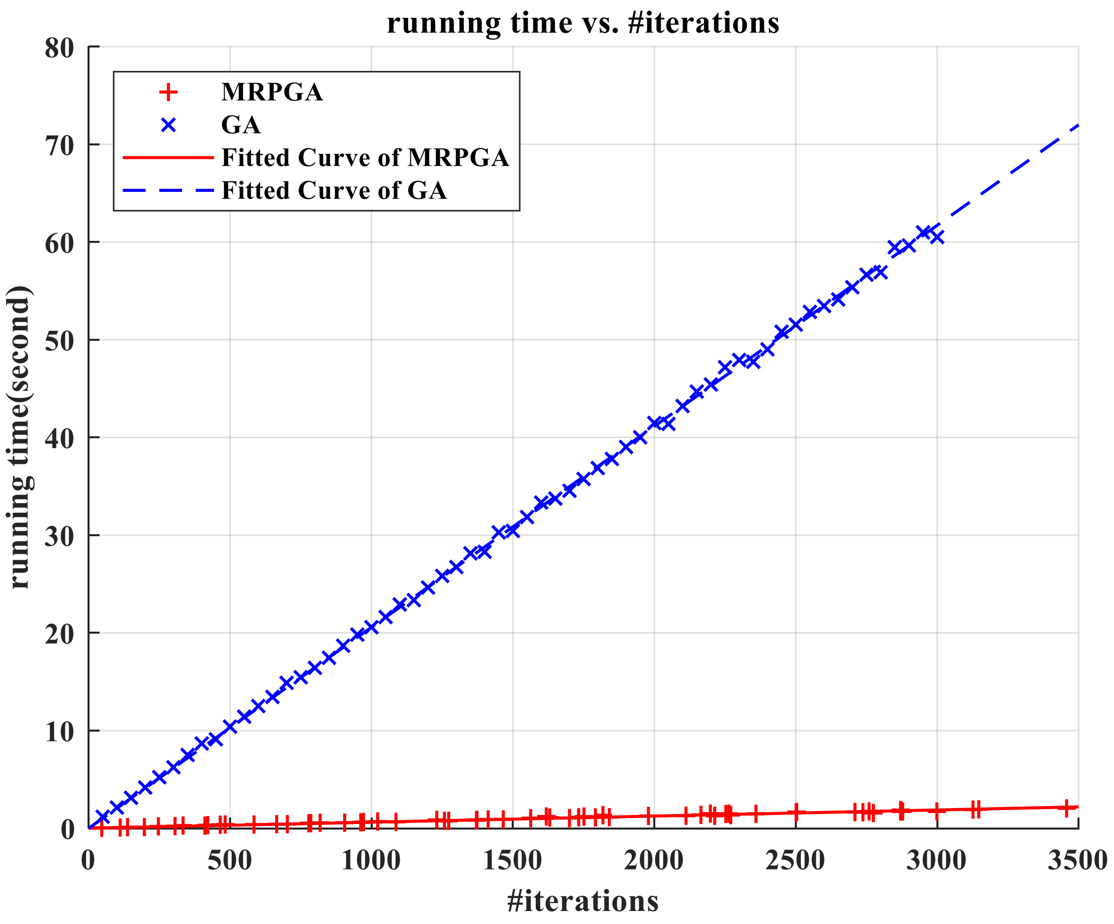
MRPGA is slightly better

Algorithm	AH
MRPGA	1.0870
GA	1.2520
ProbCache	1.9003
LCE	2.7315
Betw	3.6612

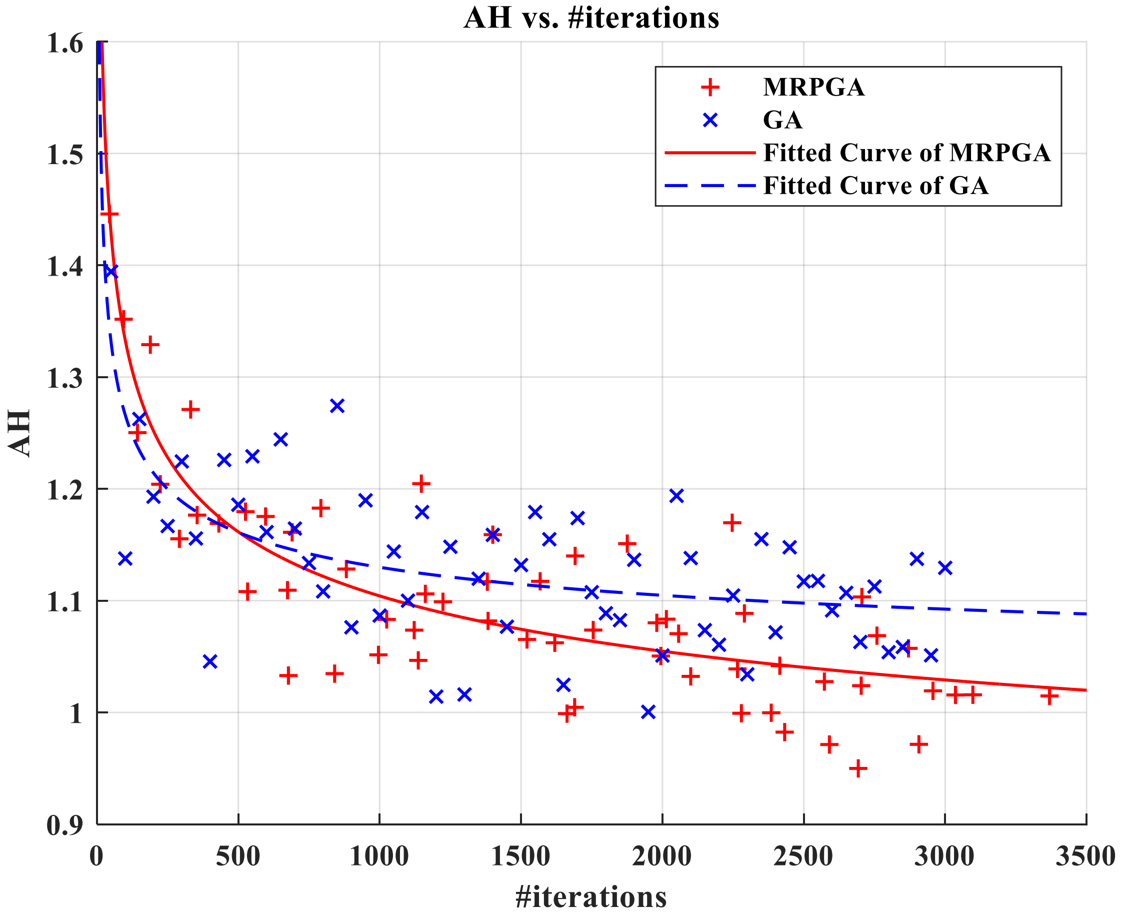
the total iteration times of GA and MRPGA are same

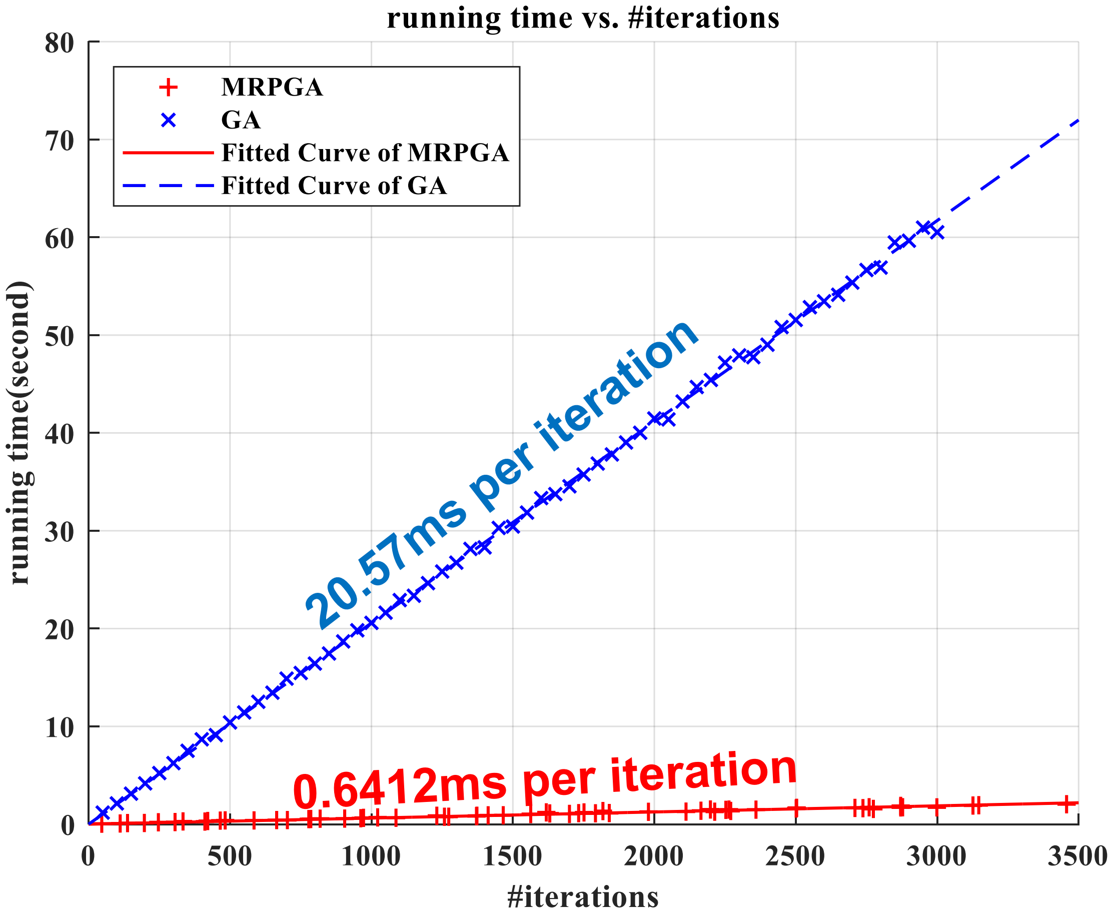
SIMULATION: RUNNING TIME



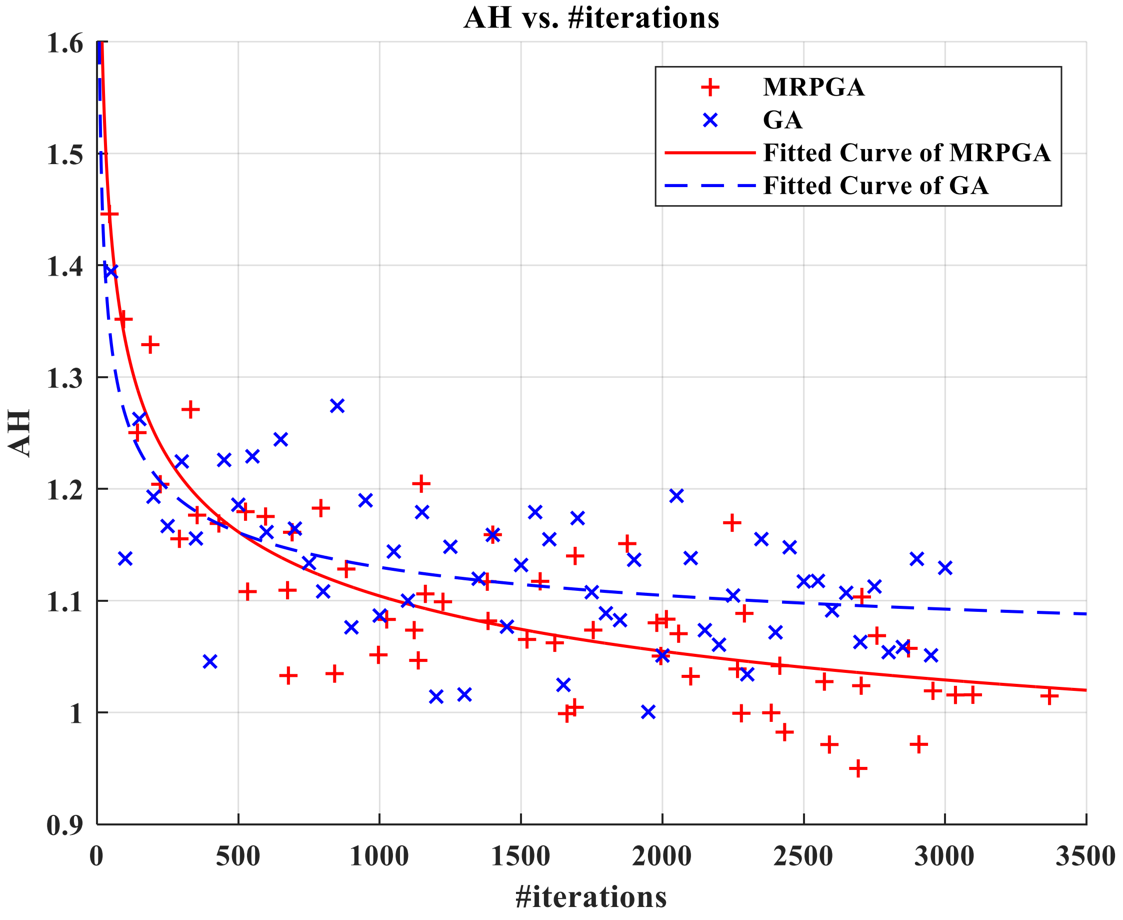


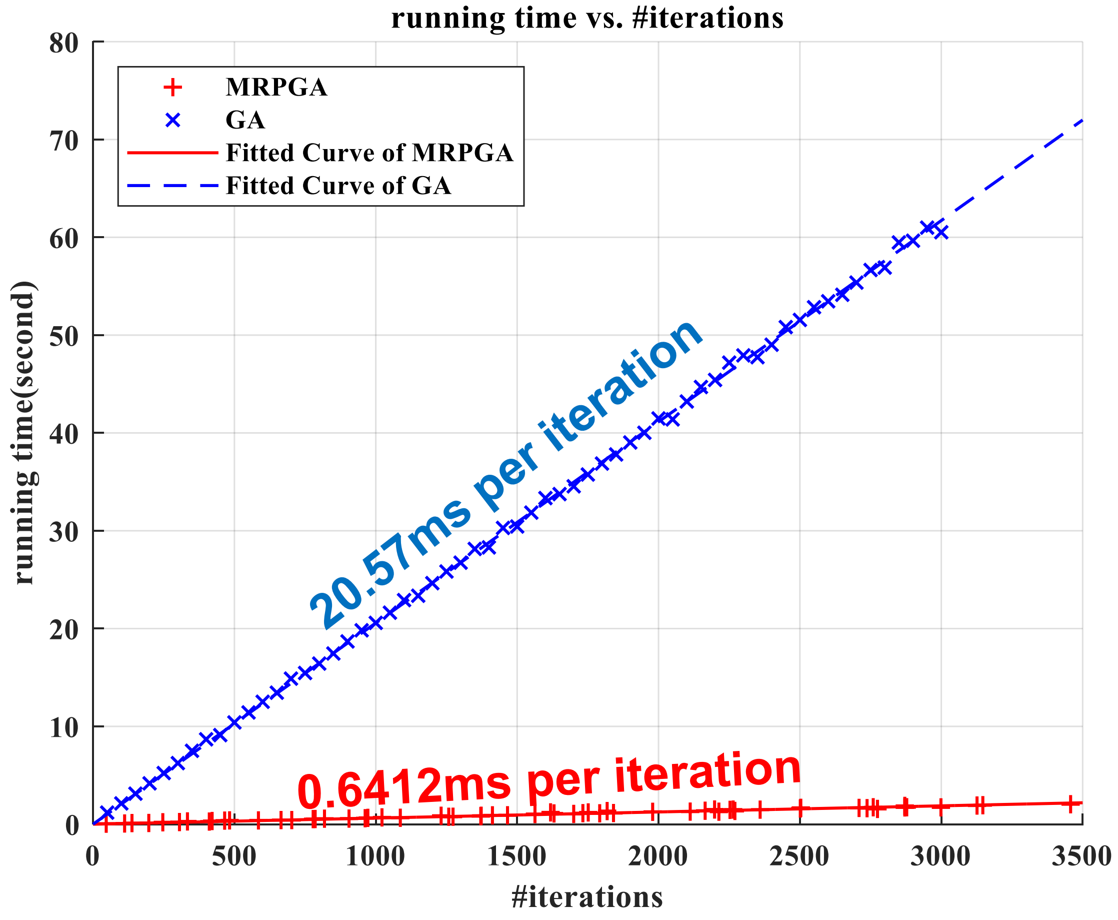
SIMULATION: RUNNING TIME





SIMULATION: RUNNING TIME





the performance of solutions are approximate to each other, but MRPGA is much faster than GA

Q&A

Thank you for your listening.

Authors: Fan Yang, **Zerui Tian**