

文章编号: 2095-2163(2020)07-0029-05

中图分类号: TP332

文献标志码: A

# 一种众核处理器拓扑重构优化近似算法

祁刚<sup>1</sup>, 覃志东<sup>1</sup>, 肖芳雄<sup>2</sup>

(1 东华大学 计算机科学与技术学院, 上海 201620; 2 金陵科技学院 软件工程学院, 南京 211169)

**摘要:** 用于众核处理器拓扑重构的行波列借算法是一种基于物理拓扑规整性贪心原则的局部搜索算法。由于其缺乏对处理器物理拓扑全局考虑, 拓扑重构时易连锁列借操作, 导致虚拟拓扑性能不高。针对这种情况, 本文首先求解处理器物理拓扑最优 AMD 解集, 充分获取了处理器物理拓扑的优化全局信息, 在此基础上以物理拓扑规整性为贪心原则, 设计出了一种更加优化的拓扑重构近似算法。实验表明, 失效核心越多, 该算法的优越性越显著。

**关键词:** 众核处理器; 物理拓扑; 虚拟拓扑; 拓扑重构

## An approximate optimization algorithm for topology reconfiguration of many-core processors

QI Gang<sup>1</sup>, QIN Zhidong<sup>1</sup>, XIAO Fangxiong<sup>2</sup>

(1 School of Computer Science and Technology, Donghua University, Shanghai 201620, China;

2 School of Software Engineering, Jinling Institute of Technology, Nanjing 211169, China)

**【Abstract】** The row rippling column stealing (RRCS) algorithm, which is used for topology reconfiguration of many-core processors, is a local searching algorithm based on the greedy principle of maintaining physical topology regularity. Without considering the overall situation of a physical topology, it is easy to result in the serious chain column stealing operations that will produce a virtual topology with lower performance. In view of this situation, the optimal AMD solution set of the physical topology of a processor is solved firstly. After obtaining the optimal global information by that, a more optimal approximate algorithm for topology reconfiguration is designed based on the principle of greedy physical topology regularity. Experiments show that the more failure cores, the performance of a virtual topology reconfigured by the new algorithm is better.

**【Key words】** Many-core processors; Physical topology; Virtual topology; Topology reconfiguration

## 0 引言

随着晶体管特征尺寸进入纳米时代, 处理器设计也进入到众核 (Many-Core) 时代<sup>[1]</sup>。由于众核芯片上集成的晶体管的数量和密度都在增大, 制造缺陷等因素导致芯片的成品率 (Yield) 越来越低<sup>[2]</sup>。为了改善成品率, 核级冗余是一种在众核芯片上采用的容缺陷设计技术<sup>[3]</sup>。该技术导致芯片拓扑结构发生了改变, 使用前需要进行拓扑重构。

已证明众核芯片的拓扑重构问题是一个 NP-complete 问题实例, 并给出了一种多项式时间复杂度的近似算法——行波列借算法 (Row rippling column stealing, RRCS)<sup>[4]</sup>。RRCS 算法基于保持处理器核心分布的行向和列向的规整性这一贪心规则, 进行局部状态空间搜索, 重构速度非常快。但其解的质量受限于所选择的初始物理拓扑状态, 且贪心规则也没有直接考虑相关的性能参数, 导致对于

某些物理拓扑结构, 重构后的虚拟拓扑的性能很差。显然, 基于性能参数优化的贪心规则, 选择较好的初始物理拓扑结构, 将有利于改善局部搜索算法的解的质量。

本文引入平均最小距离 (Average Minimum Distance, AMD) 这一性能参数, 并以该参数最优为贪心原则, 进行初始物理拓扑结构选择, 并限定了搜索状态空间, 发展了一种更优化的众核拓扑重构近似算法。

## 1 拓扑重构问题描述

### 1.1 众核核级冗余与拓扑重构思想

众核处理器分为同构和异构两种, 二者各有优点。如图 1 所示, 同构众核处理器上大量的同构核心由 2D-Mesh 结构片上网络 (Network-on-Chip, NoC) 互联<sup>[5]</sup>。当核心越来越多的时候, 要保证处理器制造, 全部核心正常可用的概率会越来越低, 这也是导致众核处理器的成品率很低的一大原因。

**基金项目:** 国家自然科学基金 (6126200)。

**作者简介:** 祁刚 (1996-), 男, 硕士研究生, 主要研究方向: 计算机容错技术; 覃志东 (1974-), 男, 博士, 副教授, 主要研究方向: 嵌入式系统与人工智能; 肖芳雄 (1972-), 男, 博士, 教授, 主要研究方向: 人工智能与数据工程。

**通讯作者:** 覃志东 Email: zdqing@dhu.edu.cn

收稿日期: 2020-04-19

为了提高成品率,针对这种同构众核处理器提出了“ $G+R$ ”的核级冗余方案<sup>[4]</sup>。即当需要  $G$  个核心正常工作时,额外集成  $R$  个冗余核心;当失效核数量不多于  $R$  个时,剩下的正常核心可以重新配置成  $G$  核处理器使用。例如,图1中的#4、#8、#12为3个设计的冗余核,制造出来的芯片,其上失效核数目及分布是不定的,每种情况都是一种具体的处理器物理拓扑结构。把图1中屏蔽结构细节,呈现出与最初设计目标拓扑相似的  $3 \times 3$  排列的这种结构,定义为虚拟拓扑结构<sup>[4]</sup>;而研究如何把具体的物理拓扑结构向虚拟拓扑结构重新进行优化配置,就是带冗余核的众核处理器的拓扑重构问题。

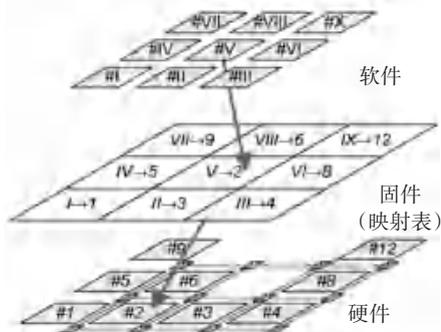


图1 众核处理器拓扑重构思想示意

Fig. 1 The schematic diagram of the topology reconfiguration of many-core processors

## 1.2 性能评价参数

为了衡量重构后的虚拟拓扑的性能,引入了两个评价指标:距离因子(Distance Factor, DF)以及拥塞因子(Congestion Factor, CF)<sup>[4]</sup>。下面对这两个因子进行形式化描述。

首先,把虚拟拓扑中两相邻核心之间的距离( $DF_{node-node'}$ )定义为它们在物理拓扑中的最短路径。而把一个核心的距离参数( $DF_{node}$ )定义为它在虚拟拓扑中与其相邻的所有核心的距离的均值,如式(1)所示:

$$DF_{node} = \frac{1}{i} \sum_{i \in \{2,3,4\}} DF_{node-node'} \quad (1)$$

距离因子 DF 则定义为芯片上  $N$  个核心的距离参数的均值,如式(2)所示:

$$DF = \frac{1}{N} \sum_{node=1}^N DF_{node} \quad (2)$$

为描述物理链路是否出现拥塞,流量分布的均匀程度,对链路  $l$  定义了拥塞参数( $CF_{link}$ ):该参数首先初始化为 0,而当虚拟拓扑内两相邻核心按照某种路由机制经过该链路  $l$  通信时,则其  $CF_{link}$  值加 1。拥塞因子 CF 则定义为虚拟拓扑中所涉通信链

路的拥塞参数的标准偏差,如式(3)所示:

$$CF = \sqrt{\frac{\sum_{link=1}^L (CF_{link} - \bar{CF}_{link})^2}{L-1}} \quad (3)$$

其中,  $L$  为虚拟拓扑结构中通信所涉物理链路的总数目。

## 1.3 现有算法的不足

假设物理拓扑有  $N$  个核心,虚拟拓扑有  $M(N < M)$  个核心位置,则物理拓扑上核心到虚拟拓扑上映射时,映射方式的状态空间大小为  $P_N^M$ 。例如,当  $M = 105, N = 100$  时,映射方式多达  $P(105, 100) = 9.01e + 165$  种;这是一个天文数字,当物理拓扑中核心增多时,将导致映射的状态空间急剧增长。已证明物理拓扑到虚拟拓扑映射的拓扑重构问题是一个二次分配问题实例,属于 NP-Complete 类<sup>[6]</sup>。解决此类问题,通常采用元启发式算法和近似算法。近似算法的运行效率高,能够迅速地获得满足问题需求的一个较优解,相对于元启发式算法具有确定的最坏执行时间。

虚拟拓扑与目标拓扑相比,性能降级的主要原因是虚拟拓扑在物理上变得不规整,因而在拓扑重构时,应该尽可能地保证虚拟拓扑结构的规整性。以此作为贪心规则,针对众核重构问题设计了一种近似算法:行波列借算法(RRCS)<sup>[4]</sup>。RRCS 在操作上以行和列为单位维护拓扑的规整性,但是这种启发式规则并没有直接考虑 DF 和 CF 这两个参数。此外,对于某些物理拓扑,RRCS 易导致连锁列借操作;这样的连锁操作不仅影响算法的计算速度而且将会导致计算所得的虚拟拓扑结构性能很差。本质上,RRCS 是一种局部搜索算法,其初始状态的选取没有考虑物理拓扑的全局信息,贪心规则也只是局部搜索,所以很难保证解的质量。

一个新的 NoC 拓扑性能评价标准:平均最小距离(AMD)<sup>[7]</sup>。AMD 计算了每个节点到其他节点的距离,通过计算节点间的距离来衡量拓扑结构中节点之间通信的质量。由于 AMD 从全局考虑了拓扑的性能,因此先以 AMD 最优,构造出初始物理拓扑搜索状态,同时直接利用性能参数限定虚拟拓扑搜索空间,最后利用 RRCS 算法的贪心规则进行邻域搜索,便可以实现较优的拓扑重构。

## 2 优化 AMD 行波列借重构近似算法

### 2.1 基于平均最小距离的搜索空间构造

如图2所示,在 2D-Mesh 网格中,从节点  $a$  到节点  $b$  的最小距离是数据包从节点  $a$  出发到达节点  $b$  所经过

的最小链路数, 最短链路如虚线所示, 最小距离为 4。在一个  $i$  行  $j$  列的 2D-Mesh 网格中, 总共有  $2 * (i * (j - 1) + j * (i - 1))$  条链路。因此, 图 2 所示的 3 行 4 列的 2D-Mesh 网格, 共有 34 条通信链路。

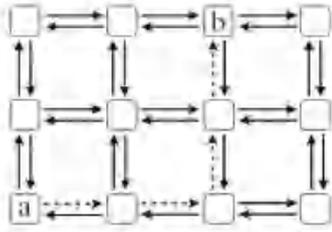


图 2 最小距离

Fig. 2 Minimum distance

假设虚拟拓扑中共有  $k$  个节点, 每个节点记为  $N_i, i \in \{1, 2, 3, \dots, k\}$ 。节点的平均最小距离记为式(4):

$$AMD_{N_i} = \frac{1}{k-1} \sum_{x=1}^k L_{N_i-N_x} \quad (4)$$

其中,  $L_{N_i-N_x}$  表示节点  $N_i$  到节点  $N_x$  在物理拓扑中的最小距离。

整个拓扑的 AMD 表示为式(5):

$$AMD_{topo} = \frac{1}{k} \sum_{i=1}^k AMD_{N_i} \quad (5)$$

使用 AMD 作为虚拟拓扑的性能评价标准, 可以发现虽然物理拓扑到虚拟拓扑的映射方式有多种, 但一些映射方式下虚拟拓扑对应的 AMD 值是一样的。如图 3 所示, 物理拓扑(a)可以映射为虚拟拓扑(b)和虚拟拓扑(c), 但二者的 AMD 值是一样的。这是因为虚拟拓扑整体的 AMD 是由单个核心的 AMD 所决定, 当构成虚拟拓扑的核心相同时, 则整体的 AMD 值相同。

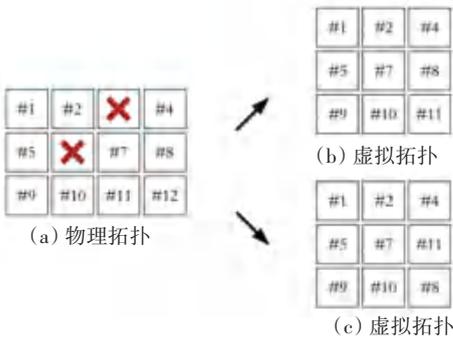


图 3 同一物理拓扑映射为不同的虚拟拓扑示意

Fig. 3 The same physical topology is mapped to different virtual topologies

虚拟拓扑映射的状态空间非常庞大。使用排列数表示拓扑映射问题的解空间大小可表示为式(6):

$$P(N, M) = \frac{N!}{(N - M)!} \quad (6)$$

其中,  $N$  为物理拓扑完好核心的个数,  $M$  为虚拟拓扑所需核心个数, 且  $N > M$ 。

由于 AMD 用来衡量核心在物理拓扑上的通信质量, 对于虚拟拓扑上的  $M$  个核心, 不论  $M$  个核心在虚拟拓扑上如何排列, 其 AMD 值是一样的。因此对于被选中的  $M$  个核心, 其共有  $P(M, M)$  种排列方式, 且其 AMD 值一致。此时, 将具有相同 AMD 值的拓扑归为一个子状态空间, 则拓扑映射问题的子状态空间共有  $X$  个, 即式(7):

$$X = \frac{P(N, M)}{P(M, M)} = \frac{N!}{(N - M)! * M!} = C(N, M) \quad (7)$$

显然, 最小 AMD 值所对应子状态空间是这其中一个, 便是需要确定的虚拟拓扑搜索空间。

由式(4)和(5)可得式(8):

$$AMD_{topo} = \frac{1}{k(k-1)} MD_{topo} \quad (8)$$

其中:

$$MD_{topo} = \sum_{i=1}^k \sum_{x=1}^k L_{N_i-N_x} \quad (9)$$

式(9)中的  $\sum_{x=1}^k L_{N_i-N_x}$  描述了一个核心到其它核心的最短距离之和。分别计算出每个核心到其它核心的最短距离之和, 找出最短距离之和最大的一个核心, 从系统中剔除, 以保障整体 AMD 值的下降, 重复此工作直至系统中核心的个数与虚拟拓扑所需核心个数相同。此时, 物理拓扑中具体的  $M$  个核心便确定了, 则初始物理拓扑搜索状态和虚拟拓扑搜索状态空间便确定了。

## 2.2 基于优化状态空间的行波列借算法

在确定好初始物理拓扑搜索状态和虚拟拓扑搜索状态空间后, 采取 RRCS 基于完整性的贪心规则, 对物理拓扑进行重构, 便可以得到较优的虚拟拓扑结构。

基于优化状态空间的行波列借算法 (AMD-RRCS) 步骤如下:

- (1) 对物理拓扑进行 RRCS, 求得虚拟拓扑;
- (2) 计算物理拓扑每个核心到其他核心的最短距离之和;
- (3) 删除一个最短距离之和最大的节点;
- (4) 重复步骤(2)与步骤(3)至物理拓扑中节点数与虚拟拓扑所需节点数一致;

(5)对物理拓扑中剩余节点使用 RRCS,求得虚拟拓扑;

(6)对比步骤(1)与步骤(5)中获取的虚拟拓扑,得出较优解。

AMD-RRCS 的伪代码如下:

基于优化状态空间的行波列借算法

```

/*
RRCS()           行波列借算法
sumMinDistance() 计算节点的最小距离之和
T               物理拓扑
M               虚拟拓扑的核心数
deleteNode()    删除最短距离之和最大的节点
maxSMD()       查找最短距离之和最大的节点
*/
R1 ← RRCS(T)
tmpT ← T
for var x ← 0 to tmpT.NodeNum - M
  for var i ← 0 to tmpT.NodeNum
    SMDij ← sumMinDistance(T, i)
  tmpT ← deleteNode(tmpT, maxSMD(SMDij))
R2 ← RRCS(tmpT)
R ← min(R1, R2)

```

显然,AMD-RRCS 较 RRCS 需要更多的计算时间,来获取较优的结果。

### 3 实验与结果分析

为了验证 AMD-RRCS 的性能,实验选择典型

的 12 行×13 列的 2D-Mesh 物理拓扑结构进行仿真,其中第 13 列表示冗余核。实验分别对不同的失效核个数  $i$  ( $1 \leq i \leq 12$ ) 进行 30 次比较。实验中失效核的位置使用线性同余算法生成。每次试验通过计算两种算法产生的虚拟拓扑的 DF 和 CF 来对比算法的效果,距离因子和拥塞因子的值越小,表示虚拟拓扑结构的性能越好。实验首先分别对比 DF 和 CF,来比较两种算法的效果,然后使用线性加权的方式将两个参数叠加为统一时间性能指标(UM),通过对比 UM 来比较两种算法的效果。实验结果如图 4~6 所示。

通过图 5 与图 6 可以发现。当失效核数目较少时,失效核在物理拓扑上的分布较为零散,RRCS 算法在计算过程中几乎不会出现连锁列借操作。因此 AMD-RRCS 算法与 RRCS 算法重构出的虚拟拓扑性能大多一致,少数情况下得到了优化。但随着失效核数目增多,AMD-RRCS 算法从全局考虑,避免了连锁列借操作,使得更多的物理拓扑重构后的虚拟拓扑性能得到较大改善。

图 6 为距离因子和拥塞因子通过线性加权方式计算出的 UM 在不同失效核数目下的对比,其中 DF 的权重为 0.1,CF 的权重为 0.9。可以看出,当失效核数目较少时,AMD-RRCS 算法重构出的虚拟拓扑的 UM 比 RRCS 算法略有提升。当失效核数目较多时,AMD-RRCS 算法能在更多的物理拓扑情况下重构出性能较优的虚拟拓扑。

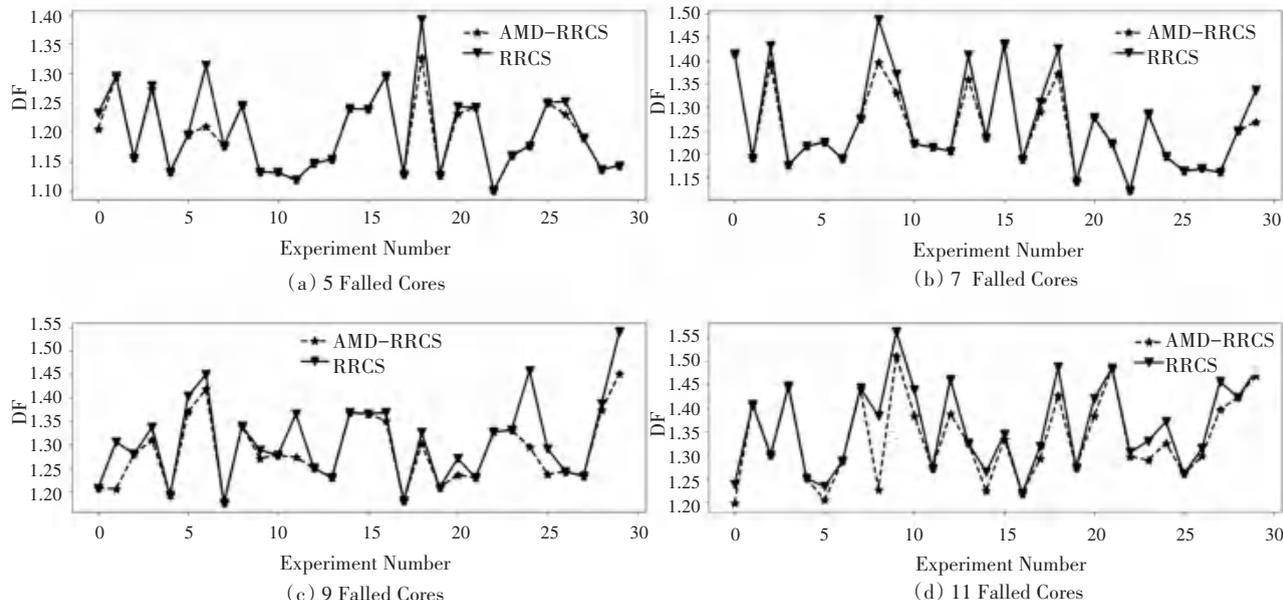


图 4 不同算法下的 DF 对比

Fig. 4 Comparison of DF under different algorithms

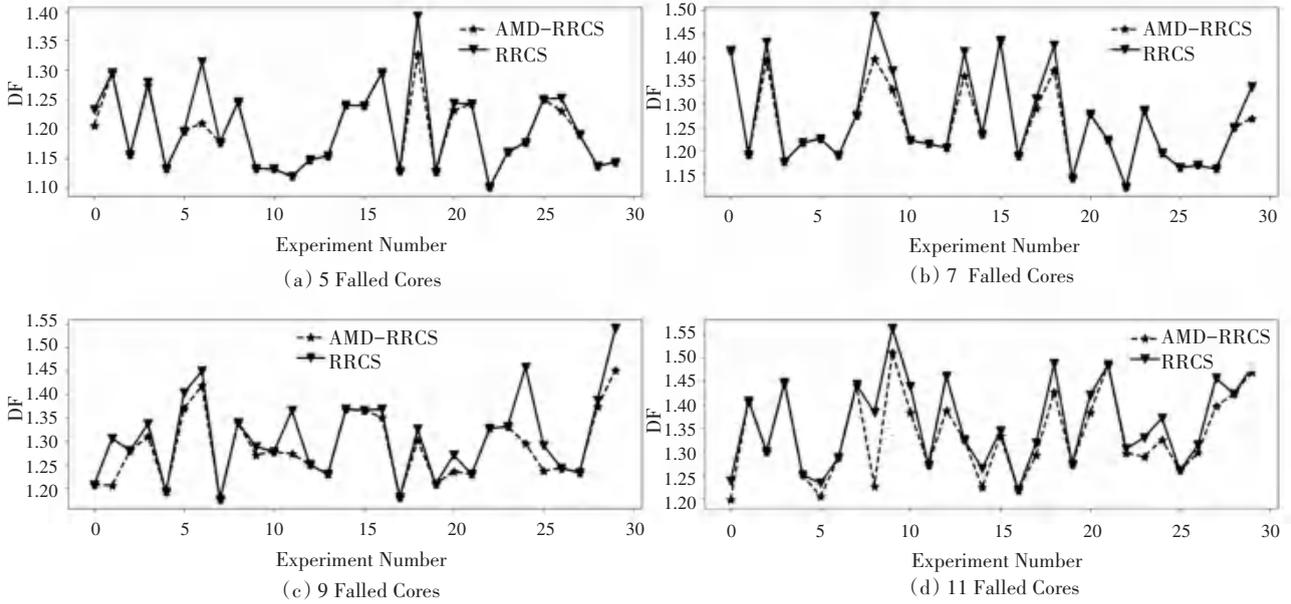


图 5 不同算法下的 CF 对比

Fig. 5 Comparison of CF under different algorithms

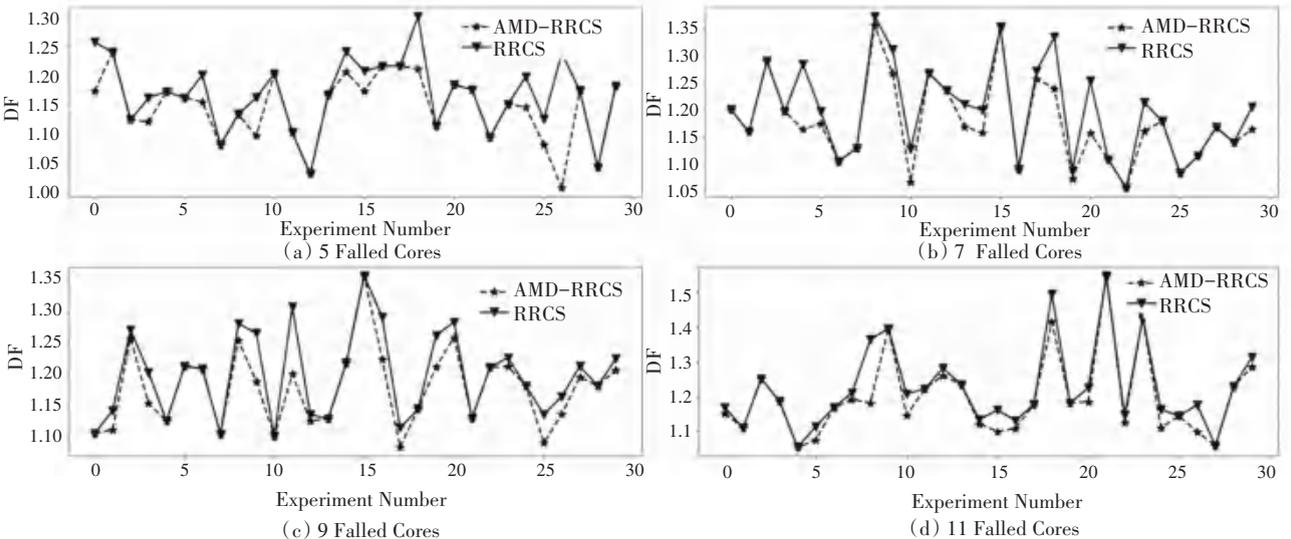


图 6 不同算法下的 UM 对比

Fig. 6 Comparison of UM under different algorithms

### 4 结束语

RRCS 算法本质上是局部搜索算法,但其未能从全局考虑。本文提出了基于优化状态空间的行波列借算法,该算法从全局考虑,构造最优 AMD 子状态空间,基于此子状态空间使用 RRCS 算法可以重构出性能更优的虚拟拓扑。

### 参考文献

[1] Jeong-Gun Lee, Sanghoon Kwak. A performance-aware yield analysis and optimization of manycore architectures [J]. Computers and Electrical Engineering, 2016, 54(8): 40-52.

[2] IsraelKoren, Dhiraj K. Pradhan. Yield and performance enhancement through redundancy in VLSI and WSI multi-processor systems[J]. Proceedings of the IEEE, 1986, 74(5): 699-711.

[3] Koren I, Koren Z. Defect tolerance in VLSI circuits; techniques and yield analysis [J]. Proceedings of the IEEE, 1998, 86(9): 1819-1838.

[4] Zhang L, Han Y, Xu Q, et al. On topology reconfiguration for defect-tolerant NoC-based homogeneous manycore systems [J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2009, 17(9): 1173-1186.

[5] Dally W J, Towles B. Route packets, not wires; On-chip interconnection networks [C]//Proceedings. IEEE Design Automation Conference, 2001: 684-689.

[6] Kuo S, Fuchs W K. Efficient spare allocation for reconfigurable arrays[J]. IEEE Design & Test of Computers, 1987, 4(1): 24-31.

[7] Saeed Shamsiri. Yield, Cost, Reliability, and Availability of Multi-Core System-on-Chips [D]. Santa Barbara: University Of California, 2011.