

基于动态自适应离散粒子群算法的 3D NoC 低功耗映射方法

刘勤让, 戴启华, 沈剑良, 赵博

(国家数字交换系统工程技术研究中心, 河南 郑州 450000)

摘 要: 相对于 2D NoC, 3D NoC 具有更好的集成度和系统性能, 是解决低功耗映射的一个可靠途径。在传统粒子群算法(PSOA, particle swarm optimization algorithm)的基础上, 提出了一种动态自适应离散粒子群算法(DADPSOA, dynamic adaptive discrete particle swarm optimization algorithm)。该算法基于早熟收敛程度和个体适应度值变化动态调整参数 ω , 不断靠近最优解; 同时对粒子进行合理的解构造, 减小了算法时间复杂度。仿真结果表明, 与随机映射、遗传算法(GA, genetic algorithm)、PSOA 和动态蚁群算法(DACA, dynamic ant colony algorithm)相比, DADPSOA 可以缩短执行时间, 减小映射结果通信功耗; 在面向任务图映射的时候, 其通信功耗下降。

关键词: 3D NoC; 低功耗映射; 解构造; 自适应离散粒子群算法

中图分类号: TP393.03

文献标识码: A

Dynamic adaptive discrete particle swarm optimization algorithm based method on low-power mapping in network-on-chip

LIU Qin-rang, DAI Qi-hua, SHEN Jian-liang, ZHAO Bo

(National Digital Switching System Engineering & Research Center, Zhengzhou 450000, China)

Abstract: Compared to 2D NoC, 3D NoC has better integrated density and system performance, which was a reliable method to solve the problem about low-power mapping. On the basis of the traditional particle swarm optimization algorithm (PSOA), a dynamic adaptive discrete particle swarm optimization algorithm (DADPSOA) was proposed. Parameter in this algorithm was adjusted dynamically based on the degree of early convergence and the charge of individual adaptive value to approach the optimal solution. At the same time, the reasonable structure of the particles was made aiming at reducing the time complexity of this algorithm. Experimental results show that comparing with the random mapping, genetic algorithm (GA), PSOA and dynamic ant colony algorithm (DACA), DADPSOA can save the execution time, reduce the communication power consumption of mapping results. The power consumption of the task graph is reduced.

Key words: 3D NoC, low-power mapping, deconstruction, adaptive discrete particle swarm optimization algorithm

1 引言

随着 CMOS(complementary metal oxide semiconductor)技术的发展, 单个芯片上集成的晶体管数量越来越多。传统总线架构已经无法满足日益增长的数据需求^[1,2]。NoC (network-on-chip)凭借其可扩展架构和并行通信的特点在一段时间内缓解数据

传输物理限制^[3,4]。但随着芯片集成度的持续提高, 2D NoC Mesh 物理链路拥塞、传输延迟等问题越发突出。而 3D IC(integrated circuit) 技术的逐渐成熟, 使复杂度高、通信量大的任务图能够从 2D Mesh 映射到 3D Mesh 上, 促成了 3D NoC 的产生, 增强了映射平台处理多样化任务图的能力^[5,6]。相对于 2D Mesh, 3D NoC Mesh 具有更好的集成度和性能, 是

收稿日期: 2015-11-02; 修回日期: 2016-11-08

基金项目: 国家高技术研究发展计划 (“863” 计划) 基金资助项目(No. 2014AA01A704); 国家自然科学基金创新群体基金资助项目 (No. 61521003); 国家自然科学基金面上基金资助项目 (No. 61572520)

Foundation Items: The National High Technology Research and Development Program of China (863 program) (No. 2014AA01A704), The Innovation Group Program Project of National Natural Science Foundation of China (No. 61521003), The General Program of National Natural Science Foundation of China (No. 61572520)

解决 2D NoC 困境的一个可靠途径。

在实际应用中, 能量消耗是阻碍 NoC 系统发展的最主要因素。而合理的映射可以使系统整体性能尤其是在能耗上得到显著提升。国内外许多学者都对 NoC 映射算法给予了关注, 如采用整数线性规划 (ILP, integer linear programming) 策略, 通过遍历所有潜在解, 确定最佳映射^[7]; 针对遗传算法易早熟收敛的缺点, 提出了改进遗传算法来避免掉入早熟陷阱^[8]; 文献^[9]采用 DACA 解决 3D NoC 映射问题, 动态调整蚁群参数, 获取映射结果; 采用蚁群混沌遗传算法, 实现了面向能耗和延时的 NoC 映射^[10]。

上述算法各有优劣, 本文针对粒子群算法收敛性差、易陷入局部极值和搜索时间长等缺陷进行改进, 提出了一种基于动态自适应离散粒子群算法的 3D NoC 映射算法。DADPSOA 在算法迭代过程中动态调整参数 ω , 使算法在初期收敛速度快, 全局搜索能力强, 而在迭代后期算法局部搜索能力增强, 保证算法的寻优能力和收敛速度符合映射要求。针对 3D NoC 架构的特殊性, 构建新的能耗模型, 确保映射结果通信功耗的正确性; 同时, 对粒子进行合理的解构造, 在经典粒子群优化算法的基本思想框架下将 PSO 算法映射到离散空间, 减小了算法的时间复杂度。算法仿真结果表明, DADPSOA 可以有效地减小算法时间复杂度和最佳映射的功耗值, 适用于 3D NoC 映射问题的求解。

2 问题定义和模型描述

2.1 3D NoC Mesh 架构

3D NoC 映射平台是由资源节点、路由和物理链路组成。本文采用规则 3D NoC 拓扑结构, 如图 1 所示。3D NoC 每一层的大小一致, 垂直方向上的路由通过 TSV(through silicon vias)连接^[5,11,12]。3D NoC 映射平台大小为 $N \times N \times L$, $N \times N$ 表示每一层映射平台的资源节点数量, L 表示层数, $N \times N \times L$ 为 3D NoC 可映射的资源节点数目。图 1 的 3D NoC 架构大小为 $3 \times 3 \times 3$, 可映射的资源节点有 27 个。根据相连链路数量的多少, 可以将资源节点分为 4 类, 相连链路数量分别为 6、5、4、3。如 14 号资源节点有 6 条链路, 1 号资源节点有 3 条链路。这些数据链路的存在将水平层面上或者垂直层面上的相邻资源节点连接起来, 从而保证资源节点之间的信息交互。

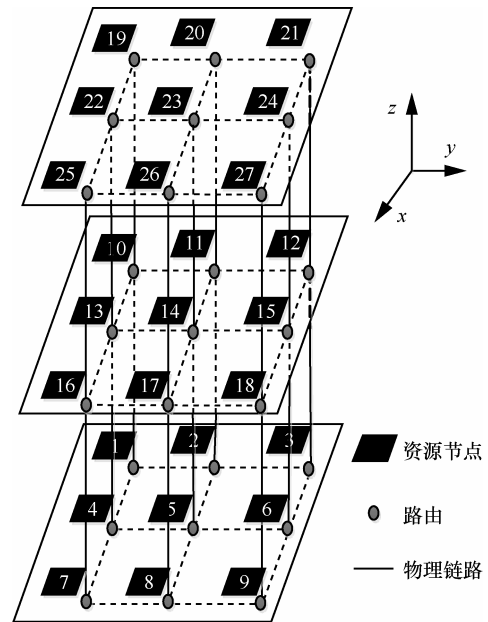


图 1 结构特征 (ARCG)

2.2 NoC 映射

片上网络映射指的是在指定系统约束条件下, 将给定任务图 TG (task graph) 上已知通信关系和通信量的逻辑 IP 核分配到结构特征图 (ARCG, architecture characteristic graph) 相应的资源节点上, 使各种应用在满足约束条件下顺利高效地完成。映射过程如图 2 所示。

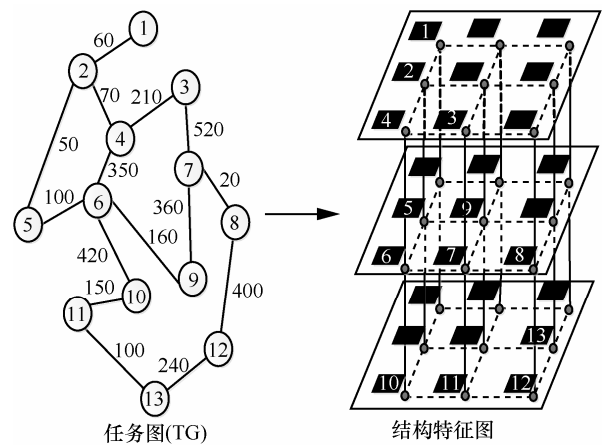


图 2 3D NoC 映射

用数学公式表示 3D NoC 映射过程, 需要给出 2 个定义。

定义 1 任务图 $TG(n)$ 是一个非循环有向图, 如图 2 中的任务图所示。顶点 $n_j \in N$ 表示一个逻辑 IP 核, 相连 IP 核连接链路上的值为这 2 个节点之间的通信量, 记为 W_{n_i, n_j} 。

定义 2 结构特征图 $G_{\text{ARCG}}(p, d)$ 也是有向图, 如图 1 所示。顶点 $p_i \in P$ 表示规则 3D Mesh 上的一个资源节点, $d_{i,j}$ 为资源节点 p_i 、 p_j 之间的物理链路, 其值为两资源节点之间的曼哈顿距离。 $B_{i,j}$ 为 p_i 、 p_j 节点之间最大通信带宽。

映射函数 $\text{map}()$ 将 TG(n) 中逻辑 IP 核分配到 $G_{\text{ARCG}}(p, d)$ 资源节点上, 记为

$$\forall n_i \in N, \text{map}(n_i) \rightarrow p_m \in P \quad (1)$$

$$\forall n_j \in N, \text{map}(n_j) \rightarrow p_s \in P \quad (2)$$

$$n_i \neq n_j \quad (3)$$

$$p_m \neq p_s \quad (4)$$

$$W_{n_i, n_j} \leq B_{i,j} \quad (5)$$

$$\text{Size}(\text{TG}) \leq \text{Size}(\text{ARCG}) \quad (6)$$

式(1)~式(4)表示逻辑 IP 核和资源节点的一一对应, 每个资源节点承担一个逻辑 IP 核的计算任务。式(5)和式(6)表明物理链路带宽满足逻辑 IP 核之间的通信需求, ARCG 中资源节点数必须大于等于 TG 中的逻辑 IP 核数。

2.3 功耗模型

3D NoC Mesh 层与层之间的距离非常短, 仅为水平相邻资源节点距离的 1%^[13]。单位比特数据在 TSV 上传输能量消耗为 9.56×10^{-3} pJ, 水平链路上为 0.127 pJ, 同样的数据传输 TSV 上的消耗仅为水平传输的 7.5%^[12]。因而 2D Mesh 上的数据传输所消耗的能量公式明显不适宜 3D NoC 映射。传统的 2D NoC 功耗模型已不适用于 3D NoC 通信功耗的计算, 从文献[12~14]得到启发, 单位数据传输能耗模型为

$$E_{\text{bit}}^{n_i, n_j} = E_{\text{horizontal}} d_h + E_{\text{vertical}} d_v + (d_h + d_v + 1) E_{\text{switch}} \quad (7)$$

其中, $E_{\text{horizontal}}$ 为单位比特在水平方向上相邻资源节点物理链路传输所消耗的能量; d_h 为资源节点 n_i 到 n_j 之间水平方向上的跳变数; E_{vertical} 为单位比特在垂直方向上相邻资源节点物理链路传输所消耗的能量; d_v 为资源节点 n_i 到 n_j 之间垂直方向上的跳变数; E_{switch} 为单位比特数据在交换结构所消耗的能量。 d_h 、 d_v 通常用曼哈顿距离表示, 假设 3D NoC 处于一个 XYZ 坐标系中, 则

$$d_h = |X_{n_i} - X_{n_j}| + |Y_{n_i} - Y_{n_j}| \quad (8)$$

$$d_v = |Z_{n_i} - Z_{n_j}| \quad (9)$$

3D 片上网络信息交互所消耗的总能量为

$$E_{\text{NoC}} = \sum_{i=1}^M \sum_{j=1}^M W_{n_i, n_j} E_{\text{bit}}^{n_i, n_j} \quad (10)$$

其中, M 为节点数, W_{n_i, n_j} 为资源节点 n_i 到 n_j 之间的通信量。目标函数即为

$$E = \min \left\{ E_{\text{NoC}} = \sum_{i=1}^M \sum_{j=1}^M W_{n_i, n_j} E_{\text{bit}}^{n_i, n_j} \right\} \quad (11)$$

3 自适应离散粒子群算法

粒子群算法是一种群体智能的优化算法, 最早由 Kennedy 和 Eberhart 在 1995 年提出^[15]。PSOA 算法源于对鸟类捕食行为的研究, 鸟类捕食时, 获取食物最简单有效的方法就是搜寻距离食物最近的鸟的周围区域。算法中每一个粒子都代表问题的一个潜在解, 每一个粒子对应一个适应度值。粒子的速度决定了粒子移动的方向和距离, 速度随着粒子位置移动进行动态调整, 从而快速获取解空间的最优解。

针对粒子群算法易陷入局部极值、过早收敛和收敛性能差等缺点, 本文提出一种基于早熟收敛程度和个体适应度值的动态自适应离散粒子群最优优化算法。

3.1 解结构

假设规则 3D NoC Mesh 中的资源节点从最底层到最上层依次排序, 每一个资源节点都有唯一的序列号。如图 1 中 $3 \times 3 \times 3$ 的 3D NoC 架构资源节点的最小序列号为 1, 最大序列号为 27。将图 2 任务图的映射结果按照逻辑 IP 核所对应的资源节点组合成一个一维数组, 如图 3 所示, 数组长度为 TG 逻辑 IP 核数, 即 13。其中, TG 中的 7 号 IP 核占用 17 号资源节点, 10 号 IP 核占用 7 号资源节点。当所有 IP 核映射到资源节点上, 且资源节点不重复的时候, 获取一个有效解。

3.2 自适应粒子群优化算法

1) 基本粒子群算法架构

在粒子群算法基本架构中, 每个粒子都代表问题空间的一个潜在解, 用位置、速度和适应度值来表征粒子特征, 粒子更新就是不断地调整自己在解空间中的位置来实现的^[16]。适应度值由目标函数计算得到, 其值的大小表示粒子的优劣。在 n 维解空

间中, 第 i 个粒子在第 t 次进化中的位置和速度可表示为

$$P_i^t = \langle P_{i,1}^t, P_{i,2}^t, \dots, P_{i,j}^t, \dots, P_{i,n}^t \rangle \quad (12)$$

$$V_i^t = \langle V_{i,1}^t, V_{i,2}^t, \dots, V_{i,j}^t, \dots, V_{i,n}^t \rangle \quad (13)$$

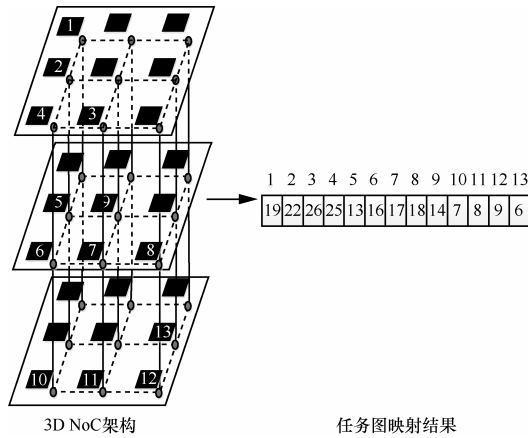


图 3 3D NoC 映射结果解构造

在算法进化中, 粒子借助个体极值和群体极值的变化来更新自身的速度和位置。个体极值 $zbest_i$ 是指个体所经历位置中适应度值最优位置, 群体极值 $gbest_i$ 是指种群中所有粒子搜索到的适应度值最优位置。更新后的速度和粒子位置如下

$$V_i^{t+1} = \omega V_i^t + c_1 r_1 rand_1(zbest_i - P_i^t) + c_2 r_2 rand_2(gbest_i - P_i^t) \quad (14)$$

$$P_i^{t+1} = P_i^t + V_i^{t+1} \quad (15)$$

其中, t 为当前迭代次数, $i=1, 2, \dots, n$ 。 c_1, c_2 为非负常数, 称为加速度因子; r_1, r_2 是分布在 $[0,1]$ 之间的随机数。在算法迭代的过程中, 根据适应度值变化进行 $zbest$ 和 $gbest$ 的更新。 ω 为惯性权重, 表示当前粒子继承先前粒子速度的能力, 是扩展或收缩解空间的重要参数, 对算法收敛有重要影响。

2) 自适应惯性权重策略

惯性权重 ω 是 PSOA 算法中调节局部搜索和全局搜索的关键参数。 ω 值较大时, 收敛速度快, 全局搜索能力强而局部搜索能力弱; ω 较小时, 收敛速度慢, 局部搜索能力强而全局搜索能力弱, 所以选择一个合适的 ω 是提高算法的寻优能力和收敛能力的关键。在解决实际问题中, 惯性权重如何设置, 在解空间内何时需要加强全局搜索, 何时需要加强局部搜索与许多因素有关。其中, 早熟收敛程度和个体适应度值是影响 ω 最重要的 2 个因素。基

于此, 针对粒子群算法存在早熟收敛、易陷入局部最优等缺点, 本文提出了一种基于早熟收敛程度和个体适应度值对惯性权重 ω 进行自适应调整策略。该策略基本步骤如下。

① 计算第 t 次迭代时粒子群的平均适应度值 f_{avg}^t 。选择适应度值小于平均适应度值的粒子, 再次计算其平均适应度值 ${}^1f_{avg}^t$ 。

$$f_{avg}^t = \frac{\sum_{i=1}^n f_i^t}{n} \quad (16)$$

其中, f_i^t 是第 i 个粒子的适应度值。定义 $\Delta = |f_{best}^t - {}^1f_{avg}^t|$, f_{best}^t 是最优粒子的适应度值。

② 若 f_i^t 大于 f_{avg}^t , 则表明该粒子为较差的粒子, 惯性权重值应当较大, 加强全局搜索, 迅速靠近较优解区域。其中, K_1 主要影响惯性权重的上限, K_2 主要影响惯性权重的调节能力, K_1, K_2 均为常数。

$$\omega = \omega_{max} - \frac{1}{1 + K_1 \exp(-K_2 \Delta)} \quad (17)$$

③ 若 f_i^t 大于 ${}^1f_{avg}^t$ 但小于 f_{avg}^t , 则表明该粒子为一般, ω 如式(18)所示, 其中, T_{max} 为最大迭代次数。

$$\omega = \omega_{max} - (\omega_{max} - \omega_{min}) \left(\frac{t}{T_{max}} \right)^2 \quad (18)$$

④ 若 f_i^t 小于 ${}^1f_{avg}^t$, 则表明该粒子是较为优秀的粒子, 应该加强局部寻优。

$$\omega = \omega_{max} - (\omega_{max} - \omega_{min}) \left| \frac{f_i^t - {}^1f_{avg}^t}{\Delta} \right| \quad (19)$$

自适应离散粒子群算法具体步骤如下。

Step1 设置参数, 随机初始化粒子的位置和速度。

Step2 计算粒子适应度值。

Step3 更新个体极值 $zbest$ 和群体极值 $gbest$ 。

Step4 根据适应度值的不同采取不同的动态自适应调整策略, 依照式(17)~式(19)调整惯性权重。

Step5 根据式(14)和式(15), 更新粒子速度, 获得新的粒子并计算适应度值。

Step6 判断是否结束, 结束则显示结果; 否则转到 Step2。

在 Step2 中, 以目标函数, 即映射结果通信能量损耗为适应度值, 适应度值越小, 映射结果越好;

适应度值越大，映射结果越差。

在 Step4 中，每次迭代整个粒子群都会进行速度和空间位置的更新，为减小算法时间复杂度，惯性权重只在整体迭代中进行调整，即 ω 调整次数为算法迭代次数。

在 Step5 中，经过式(14)和式(15)处理之后，新的粒子位置可能会超出映射限制。如 $P_i = \{19, 22, 26, 25, 13, 16, 17, 18, 14, 7, 8, 9, 6\}$ ，此时粒子速度为 $V_i = \{1, 2, -2, 5, 3, -4, 5, 1, -5, 0, 3, -1, 5\}$ 。下一次迭代过程中， $P_{i+1} = \{20, 24, 24, 30, 16, 12, 22, 19, 9, 7, 11, 8, 11\}$ 。可以发现， $P_{i+1}[2]$ 和 $P_{i+1}[3]$ ， $P_{i+1}[11]$ 和 $P_{i+1}[13]$ 重复，且 $P_{i+1}[4]$ 超出映射范围，由于图 1 ARCG 资源节点号最大为 27，根据一一对应映射规则，可知 P_{i+1} 无效。因此，需要对不合理的新粒子进行解修复操作，确保后续算法的有效执行，如式(20)所示。

$$P_{i+1} = \{20, 24, 24, 30, 16, 12, 22, 19, 9, 7, 11, 8, 11\} \xrightarrow{\text{repair}} P_{i+1} = \{20, 24, 23, 26, 16, 12, 22, 19, 9, 7, 11, 8, 10\} \quad (20)$$

4 算法时间复杂度分析

为了便于计算 DADPSOA 的时间复杂度，本文设置参数如下：

- 1) 迭代次数为 N_1 ；
- 2) 粒子数目为 N_2 ；
- 3) 任务图逻辑 IP 核数目为 N_3 ；
- 4) 结构特征图资源节点数目为 N_4 。

算法 ILP、GA、DACA、PSOA 和 DADPSOA 的复杂度对比如表 1 所示。

算法	算法复杂度
ILP	$\frac{N_4!}{(N_4 - N_2)!}$
GA	$N_1 N_2^2 N_4$
DACA	$N_1 N_2^2 N_3$
PSOA	$N_4^5 \ln N_4 + N_1 N_2 N_4^2 \ln N_4$
DADPSOA	$N_1 N_2 N_3$

ILP 由于要穷尽所有的映射结果，其算法复杂度往往最高，实际可操作性最低。后文仿真实验中由于 ILP 时间复杂度过于庞大，无法获取其有效映射结果进行对比。根据一一对应的映射关系所得，ARCG 资源节点数是大于任务图逻辑 IP 数的，即

$N_4 > N_3$ ，可得

$$O(N_1 N_2 N_3) < O(N_1 N_2^2 N_4) \quad (21)$$

$$O(N_1 N_2 N_3) < O(N_1 N_2^2 N_3) \quad (22)$$

$$O(N_1 N_2 N_3) < O(N_4^5 \ln N_4 + N_1 N_2 N_4^2 \ln N_4) \quad (23)$$

从算法时间复杂度上来看，DADPSOA 算法计算量小于 GA、DACA 和 PSOA。当 ARCG 规模较大且待映射逻辑 IP 核数目较多的时候，DADPSOA 计算量远远小于其余几种算法。综上所述，对于算法时间复杂度，本文所提算法性能提升最明显。

5 算法仿真与分析

为了验证 DADPSOA 算法的优势，本文采用 C++ 语言编写算法源代码，在 Linux 环境下进行编译和仿真。测试样本同时采用随机生成任务图和实际任务通信图，确保数据的多样性和代表性。其中，MPEG-4、VOPD、263 Enc 为典型实际应用图，属性如表 2 所示。映射平台 ARCG 大小为 $3 \times 3 \times 3$ 。获取每种任务图在不同映射算法下 20 次映射结果，并计算 20 次映射结果的功耗平均值。仿真参数 ω_{\max} 、 ω_{\min} 、 c_1 、 c_2 的设置参考文献[15]，另外，需要声明的是粒子速度范围设定为 $[-0.25N_4, 0.25N_4]$ 。当粒子速度超出上下限的时候，将违法粒子速度更改为边界速度。这样做的原因是减小更新后粒子是非法解的概率，防止粒子解超出 n 维解空间范围，影响后续的粒子寻优趋势。

任务图	逻辑 IP 核数	通信边	通信总量/(Mbit·s ⁻¹)
MPEG-4	12	13	3 466
VOPD	16	21	3 471
263 Enc	12	12	230.214

图 4 是 DADPSOA 在计算 3 种实际任务通信图时算法迭代过程中惯性权重值 ω 的变化过程。从图中可以明显地看出，在算法的起始阶段 ω 较大，可以继承前次迭代过程中速度的 85% 左右，确保了算法前期较强的全局搜索能力，收敛速度快，求解精度低；算法的过渡阶段，惯性权重 ω 随着迭代次数的增加，逐渐减小，表明粒子向全局最优区域靠近，空间距离越近， ω 越小；在算法后期，惯性权重值 ω 减小速度变缓，在 0.2~0.3 趋于一个较小的稳定值，此时算法全局搜索能力

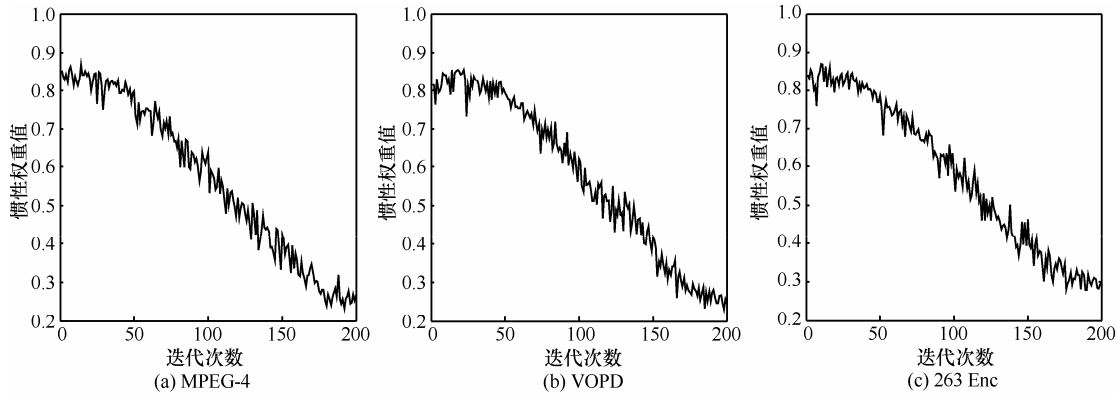


图 4 惯性权重

弱，收敛速度变慢，局部搜索能力强，利于算法跳出局部最优而求得全局最优。另外，3 种任务图的 ω 值变化都不是一个光滑的曲线，是随着迭代次数、全局极值和局部极值一同作用下不断动态调整的曲线。惯性权重值 ω 的走势符合算法搜索过程的变化状态。

记录 3 种实际通信任务图每次迭代过程中 DADPSOA 搜寻到的全局最优值，如图 5 所示。从图中可以明显看到，算法初始阶段，MPEG-4、VOPD、263 Enc 映射结果最优个体适应度值收敛速度很快，以较少的迭代次数迅速靠近最优解区域；在后期局部寻优中，算法收敛速度明显放慢，需要较长的迭代次数完成最优解的获取。3 种实际通信任务图全局最优值寻优过程印证了惯性权重值 ω 前期大、后期小的变化顺序。

图 6 为 4 种算法映射结果相对于随机映射结果功耗下降比例。可以发现，DADPSOA 算法映射结果优于其他算法，映射结果通信功耗下降更加明显。相对于随机映射、GA、PSOA 和 DACA 算法，DADPSOA 在通信功耗上分别下降了 49.56%、

13.63%、11.03%和 5.72%。

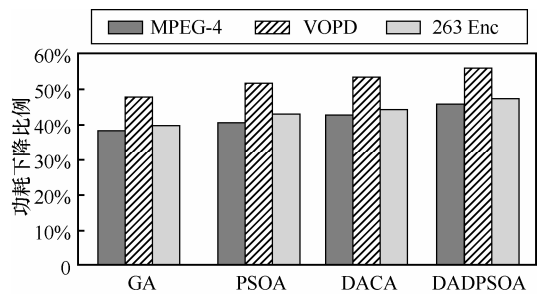


图 6 DADPSOA 功耗下降对比

考虑到上文测试的实际任务通信图逻辑 IP 核数目相对于 ARCG 规模较小，为进一步体现本文所提算法优势，利用 TGFF^[17]随机生成具有 27 个逻辑 IP 核的通信任务图 G1，并将其经 DADPSOA 处理之后的结果映射到 3×3×3 的 3D NoC 架构上，记录其通信功耗值，如表 3 所示。可以看到，DADPSOA 对随机生成且具有较大逻辑 IP 核数的任务图仍然有效。DADPSOA 映射结果通信功耗明显优于随机映射，且相对于 GA、PSOA、DACA 映射结果也分别有 21.32%、15.69%、6.74% 的功耗下降。

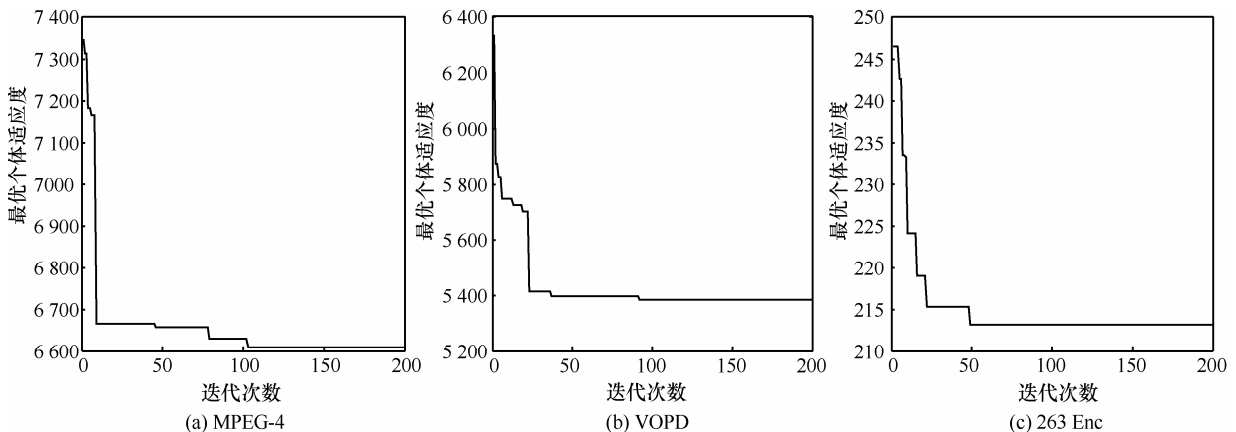


图 5 全局最优值

表 3

任务图 G1 映射结果对比

TGFF 任务图	逻辑 IP 核数	ARCG 大小	通信功耗比例			
			DADPSOA Random	DADPSOA GA	DADPSOA PSOA	DADPSOA DACA
G1	27	3×3×3	42.53%	78.68%	84.31%	93.26%

通信任务图在映射平台实际运行的时候，其执行时间与多种因素有关。利用链路负载方差来衡量通信时延是可信度比较高的一个指标^[10]。

$$load = \frac{\sum_{i=1}^L \left(load_i - \frac{\sum_{j=1}^L load_j}{L} \right)^2}{L} \quad (23)$$

其中， L 为链路总数， $load_i$ 为链路 i 的通信负载量，

$\frac{\sum_{j=1}^L load_j}{L}$ 为平均链路负载量。以 MPEG-4 和 G1 为例，分别记录 2 种任务图在 GA、PSOA、DACA、DADPSOA 算法下的映射结果。根据式(23)获取映射结果链路负载方差，并采用 NoCmap^[18]仿真平台获取 3 种映射结果的运行时间。结果如表 4 和图 7 所示，DADPSOA 映射结果运行时间和链路负载方差都明显优于 GA、PSOA 和 DACA。同时可以看到，映射结果链路负载方差的下降和实际运行时间的减小保持了一致性。持续大量的链路负载会引起传输通路堵塞，延长了数据传输时间，进而影响任务图运行时间。故 DADPSOA 算法性能优于 PSOA、DACA。

表 4 链路负载方差对比

任务图	链路负载方差		
	DADPSOA GA	DADPSOA PSOA	DADPSOA DACA
MPEG-4	84.64%	88.16%	93.12%
G1	86.83%	91.45%	96.62%

6 结束语

针对 3D NoC 环境下的通信任务图低功耗映射问题，本文提出了一种基于动态自适应离散粒子群的新型映射算法。通过合适的解构造，搭建了任务图逻辑 IP 核到 ARCG 资源节点正确高效的对应关系，减小了算法的时间复杂度；根据 3D NoC 架构

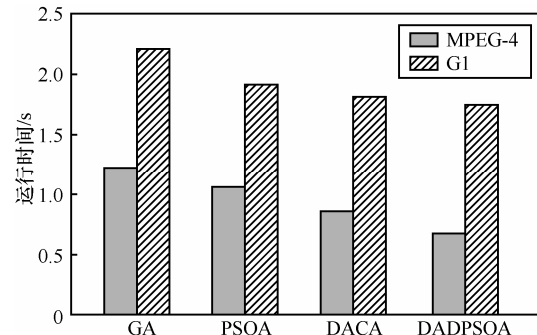


图 7 MPEG-4 和 G1 在不同映射算法下映射结果的通信时间

特性，重新定义了 NoC 功耗模型，保证了映射结果通信功耗的可靠性和准确性；利用粒子群算法早熟收敛程度和个体极值动态调整惯性权重值，防止算法陷入局部陷阱，阻止算法停滞不前。仿真结果表明，与随机映射、GA、PSOA、DACA 映射算法相比，DADPSOA 映射结果通信功耗下降了 49.56%、13.63%、11.03%和 5.72%，寻优效果更好，算法时间复杂度更小。

本文着重研究 3D NoC 一一对应的映射关系，技术的提升使资源节点对于逻辑 IP 核承载量加大。在下一步的研究过程中，可以进一步考虑多对一映射关系的实现，同时，可以丰富映射算法的目标函数，建立一个更为精准的 3D NoC 映射算法模型。

参考文献：

- [1] LE Q, YANG G, HUNG W N N, et al. Pareto optimal mapping for tile-based network-on-chip under reliability constraints[J]. International Journal of Computer Mathematics, 2015, 92(1): 41-58.
- [2] KUMAR S, JANTSCH A, SOININEN J P, et al. A network on chip architecture and design methodology[C]//VLSI, 2002. IEEE Computer Society Annual Symposium. IEEE, Pittsburgh, Pennsylvania, 2002: 105-112.
- [3] SAHU P K, CHATTOPADHYAY S. A survey on application mapping strategies for network-on-chip design[J]. Journal of Systems Architecture, 2013, 59(1): 60-76.
- [4] BJERREGAARD T, MAHADEVAN S. A survey of research and practices of network-on-chip[J]. ACM Computing Surveys (CSUR), 2006, 38(1): 1.
- [5] ELMILIGI H, EL-KHARASHI M W, GEBALI F. Power consumption

- of 3D networks-on-chips: modeling and optimization[J]. *Microprocessors and Microsystems*, 2013, 37(6): 530-543.
- [6] CHENG Y, ZHANG L, HAN Y, et al. Thermal-constrained task allocation for interconnect energy reduction in 3D homogeneous mpsoes[J]. *Very Large Scale Integration (VLSI) Systems*, IEEE Transactions, 2013, 21(2): 239-249.
- [7] TOSUN S, OZTURK O, OZEN M. An ILP formulation for application mapping onto Network-on-Chips[C]//*Application of Information and Communication Technologies*, 2009. AICT 2009. International Conference IEEE, Baku, Azerbaijan.2009: 1-5.
- [8] PLANIVELOO V A, AMBROSE J A, SOWMYA A. Improving GA-based NoC mapping algorithms using a formal model[C]//*VLSI (ISVLSI)*, 2014 IEEE Computer Society Annual Symposium. IEEE, Florida, USA, 2014: 344-349.
- [9] 王佳文, 李丽, 易伟, 等. 3D NoC 映射问题的动态蚁群算法[J]. *计算机辅助设计与图形学学报*, 2011, 23(9): 1614-1620.
WANG J W, LI L, YI W, et al. A dynamic ant colony optimization algorithm for 3D NoC mapping[J]. *Journal of Computer-Aided Design & Computer Graphics*. 2011, 23(9): 1614-1620.
- [10] 杨盛光, 李丽, 高明伦, 等. 面向能耗和延时的 NoC 映射方法[J]. *电子学报*, 2008, 36(5): 937-942.
YANG S G, LI L, GAO M L, et al. An energy-and delay-aware mapping method of NoC[J]. *Acta Electronic Sinica*, 2008, 36(5): 937-942.
- [11] RAHMANI A M, LILJEBERG P, PLOSILA J, et al. Developing a power-efficient and low-cost 3D NoC using smart GALS-based vertical channels[J]. *Journal of Computer and System Sciences*, 2013, 79(4): 440-456.
- [12] ZHANG T, WU N, ZHOU F, et al. A traffic equilibrium mapping method with energy minimization for 3D NoC-Bus mesh architecture[J]. *IAENG International Journal of Computer Science*, 2015, 42(1): 1-7.
- [13] WANG X, PALESI M, YANG M, et al. Power-aware run-time incremental mapping for 3-D networks-on-chip[C]//*Network and Parallel Computing*. Springer Berlin Heidelberg, 2011: 232-247.
- [14] 李东生, 刘琪. 面向通信能耗的 3D NoC 映射研究[J]. *半导体技术*, 2012, 37(7): 504-507.
LI D S, LIU Q. Research on mapping 3D network on chip for communication energy-aware[J]. *Semiconductor Technology*, 2012, 37(7): 504-507.
- [15] 史峰, 王辉, 郁磊, 等. *Matlab 智能算法 30 个案例分析*[M]. 北京航空航天大学出版社, 2011.
SHI F, WANG H, YU L, et al. *Matlab intelligent algorithm analysis of 30 cases*[M]. Beihang University Press, 2011.
- [16] SAHU P K, SHAH T, MANNA K, et al. Application mapping onto

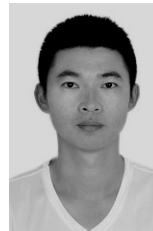
mesh-based network-on-chip using discrete particle swarm optimization[J]. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2014, 22(2): 300-312.

- [17] DICK R P, RHODES D L, WOLF W. TGFF: task graphs for free[C]//*The 6th International Workshop on Hardware/Software codesign*. IEEE Computer Society. 1998: 97-101.
- [18] Available online[EB/OL].<http://www.ece.cmu.edu/~sld/software/nocmap.php>, 2015. 4.29

作者简介:



刘勤让 (1975-), 男, 河南睢县人, 博士, 国家数字交换系统工程技术研究中心研究员, 主要研究方向为系统体系结构、网络安全与芯片设计。



戴启华 (1990-), 男, 浙江湖州人, 国家数字交换系统工程技术研究中心硕士生, 主要研究方向为片上网络映射与芯片设计。



沈剑良 (1982-), 男, 浙江德清人, 博士, 国家数字交换系统工程技术研究中心助理研究员, 主要研究方向为嵌入式系统、可重构计算与片上网络。



赵博 (1981-), 男, 吉林公主岭人, 博士, 国家数字交换系统工程技术研究中心助理研究员, 主要研究方向为软件定义网络。