

低功耗片上网络映射的遗传及蚂蚁融合算法

任向隆, 安建峰, 高德远, 樊晓桢

(西北工业大学计算机学院, 710072, 西安)

摘要: 针对带宽和时延约束下的低功耗片上网络映射问题, 提出了基于遗传和蚂蚁算法融合的映射算法. 该算法利用遗传算法的快速搜索能力, 获得若干优化解, 并按照这些优化解的最优顺序给蚂蚁路径赋初值, 以初始化蚂蚁算法的信息素分布. 然后, 借助具有交叉和变异操作的蚂蚁算法, 充分利用蚂蚁算法的正反馈特性, 搜索低功耗映射问题的更优解. 该算法具有收敛速度快、优化效果好的特点, 可用于求解大规模片上网络映射问题. 实验结果表明: 当系统规模扩大时, 该算法在搜索时间方面明显优于遗传类算法和蚂蚁类算法, 如系统规模为 64 处理单元时, 搜索速度提高率最高可达 220.3%, 在较快收敛的同时, 还保持了较好的优化效果, 与蚂蚁类算法的差别可保持在 9.1% 以内.

关键词: 映射; 遗传算法; 蚂蚁算法; 低功耗; 片上网络

中图分类号: TP302 **文献标志码:** A **文章编号:** 0253-987X(2012)08-0065-06

Fusion Strategy with Genetic and Ant Algorithms for Low-Power NoC Mapping

REN Xianglong, AN Jianfeng, GAO Deyuan, FAN Xiaoya

(School of Computer Science and Engineering, Northwestern Polytechnical University, Xi'an 710072, China)

Abstract: To solve low-power mapping with bandwidth and latency constraints of network-on-chip (NoC), a fusion mapping with genetic and ant algorithms is proposed. Several optimal solutions are obtained by quick searching of genetic algorithm, then the initial values of ant paths are assigned according to the order of these optimal solutions to initialize the pheromone distribution of ant algorithm. Resorting the ant algorithm with crossover and mutation operation, and taking full advantage of the positive feedback features of the ant algorithm, the exact solution of the low-power mapping is searched out. This fusion strategy can be used to solve the large-scale NoC mappings with good optimization precision and convergence. The experimental results indicate that this strategy obviously outperforms genetic and ant algorithms for larger scale systems, the improvement of search rate can be heightened up to 220.3% for system of 64 processing elements, and the optimization difference gets less than 9.1% compared with ant algorithm.

Keywords: mapping; genetic algorithm; ant algorithm; low-power; network-on-chip

随着半导体技术的快速发展, 设计者可在一块芯片上集成数百个核以满足日益增长的计算需求, 但庞大的计算资源则给通信系统造成了巨大的压力. 另外, 随着特征尺寸的快速缩小, 互连系统的延

收稿日期: 2011-12-22. 作者简介: 任向隆(1982—), 男, 博士生; 安建峰(通信作者), 男, 讲师. 基金项目: 国家自然科学基金资助项目(60736012, 60773223, 61003037, 61173047); 国家“863 计划”资助项目(2009AA01Z110); 西北工业大学基础研究基金资助项目.

网络出版时间: 2012-05-22

网络出版地址: <http://www.cnki.net/kcms/detail/61.1069.T.20120522.1033.002.html>

时和优化成为制约系统性能的关键因素之一,由于传统的总线及点对点互连结构在可扩展性、通信效率、功耗等方面的限制,已使通信成为突出的性能瓶颈,因此片上网络(NoC)被认为是解决上述问题的一个可行方案^[1],可有效地解决复杂的通信问题。

如何在系统的设计约束下,通过映射降低通信系统的功耗,已成为 NoC 设计的研究热点。文献[2]基于位能量模型,用分支限界法解决了规则 NoC 的映射问题。文献[3]采用两步遗传算法实现了任务图到 NoC 结构的映射。文献[4]用遗传算法完成了能量和带宽感知的规则 NoC 的多目标映射。文献[5]采用蚂蚁算法实现了面向能耗和延时的 NoC 映射。文献[6]提出了一种在延时约束条件下,基于遗传算法的片上网络通信链路的低功耗映射算法。其他关于映射的工作还有文献[7-9]等。上述工作虽在不同程度上降低了系统的通信能耗,但在算法的收敛速度和优化效果上未能做到较好的兼顾,仍有可改进之处。为获得更优的片上网络通信功耗,并提高映射算法的收敛速度,本文提出了一种基于遗传算法和蚂蚁算法融合的 NoC 映射(GAM)算法。

1 系统模型及问题描述

本文采用 2D Mesh 网络拓扑,不同节点间的数据传输采用确定性的 XY 路由。若进行数据传输的 2 个节点的曼哈顿距离为 n ,则一个数据包传输需要经过 $(n+1)$ 个路由器和 n 条链路,路由距离为 $(n+1)$ 跳。

1.1 功耗模型

根据文献[2]的位能量模型,通信链路中的路由距离为 $(n+1)$ 跳的资源节点 r_i, r_j 之间传输 1 位数据的平均能耗为

$$E_{i,j}(n+1) = (n+1)E_R + nE_L \quad (1)$$

式中: E_R 为单个路由器传输 1 位数据的平均动态能耗; E_L 为单条链路传输 1 位数据的平均动态能耗。根据功耗和能耗的关系,可得链路功耗^[6]为

$$P_{i,j}^C = (E_{i,j}V_{i,j})/T = (E_{i,j}V_{i,j})/(V_{i,j}/B_{i,j}) = E_{i,j}B_{i,j} \quad (2)$$

式中: $V_{i,j}$ 表示 r_i 到 r_j 的通信量; T 表示总量为 $V_{i,j}$ 的数据从 r_i 发送到 r_j 需要的时间; $B_{i,j}$ 表示 r_i 到 r_j 的通信带宽。

1.2 问题定义

参照文献[2],对 NoC 映射作如下的形式化定义。

定义 1 应用特征图 $A_{PG}(C, A)$ 是一个有向图,图

中每个顶点 $c_i \in C$ 表示一个处理单元(PE),每条有向边 $a_{i,j} \in A$ 表示从 c_i 到 c_j 的通信任务。每条有向边 $a_{i,j}$ 具有以下属性:①参数 $v(a_{i,j})$ 表示 $a_{i,j}$ 的数据流量;②参数 $b(a_{i,j})$ 是为满足性能需求, $a_{i,j}$ 所需要的最小带宽;③参数 $d(a_{i,j})$ 表示 $a_{i,j}$ 允许的最大传输延时,单位为跳数。

定义 2 体系结构特征图 $A_{RCG}(R, P)$ 是一个有向图,每个顶点 $r_i \in R$ 表示体系结构中的一个资源节点,每条有向边 $p_{i,j} \in P$ 表示从 r_i 到 r_j 的路由路径。每条 $p_{i,j}$ 具有以下参数: $e(p_{i,j})$ 表示从 r_i 到 r_j 传送 1 位数据的平均能耗,即 $e(p_{i,j}) = E_{i,j}; \forall p_{i,j} \in P, L(p_{i,j})$ 表示 $p_{i,j}$ 用到的链路集合;权重 $h(p_{i,j})$ 表示 r_i 与 r_j 间的路由距离,用跳数表示。

给定应用特征图 G_C 和体系结构特征图 G_R ,且 $|G_C| \leq |G_R|$,时延和带宽约束下的低功耗 NoC 映射问题就是要寻找映射函数 $\Phi(\cdot)$,使得最优功耗 $P_T = \sum_{i,j} P_{i,j}^C$ 最小且满足

$$\forall c_i \in C, \Phi(c_i) \in R \quad (3)$$

$$\forall c_i \neq c_j, \Phi(c_i) \neq \Phi(c_j) \quad (4)$$

$$\forall l_k \in L, \sum_{i,j} b(a_{i,j}) f(l_k, p_{\Phi(c_i), \Phi(c_j)}) \leq B(l_k) \quad (5)$$

$$\forall a_{i,j} \in A, h(p_{\Phi(c_i), \Phi(c_j)}) \leq d(a_{i,j}) \quad (6)$$

式中 $B(l_k)$ 是链路 l_k 的带宽,则

$$f(l_k, p_{m,n}) = \begin{cases} 0, & l_k \notin L(p_{m,n}) \\ 1, & l_k \in L(p_{m,n}) \end{cases} \quad (7)$$

式(3)、式(4)代表每个 PE 都应该且仅能被映射到一个资源节点上,并且每个资源节点仅能驻留一个 PE。式(5)限定映射结果中任意一条链路上通信流量的总和均不能超过该链路的可用带宽;式(6)限定映射结果必须保证 A_{PG} 中任意通信任务的时延需求。式(5)、式(6)通过限定映射必须满足每条通信流量的带宽和时延需求,定义了映射必须满足的性能约束。

2 遗传、蚂蚁融合的 NoC 映射

2.1 GAM 算法的思想

文献[10]提出了遗传算法(GA)与蚂蚁算法(AA)融合的 GAAA 算法,其基本思路是先采用遗传算法,利用其快速、随机及全局收敛的特性,产生求解问题的较优解,并用于初始化蚂蚁算法的信息素分布;然后采用蚂蚁算法,在该初始信息素分布的基础上,充分利用蚂蚁算法的正反馈特性,求解效果更优的解。

GAM 算法借鉴了 GAAA 算法,将遗传算法与蚂蚁算法融合,并应用于 NoC 低功耗映射问题的求解.在 GAM 算法中,为进一步加速算法的收敛,在较短时间内得到优化效果较好的映射方案,对遗传算法和蚂蚁算法进行了更为细粒度的融合.按照蚂蚁算法完成一次遍历后,再让蚂蚁算法作遗传算法的交叉操作和变异操作.

2.2 融合算法中 GA 的构造

根据 NoC 映射问题的特点,本文采用十进制整数编码,其位数由节点数目确定.以具有 9 个节点的网络为例,染色体编码(746512398)表示 PE 节点(746512398)分别映射到资源节点(123456789).本文主要对设计约束下的通信链路功耗进行优化,将适应函数直接取映射的目标函数.

采用随机函数生成一定数量的染色体种群,并采用最佳保留选择策略来选择适应度较好的个体进入下一代群体.

使用双点顺序交叉策略的方法如下:①在两父串上随机选择一个交叉区域,例如交叉区域选 $a = 123|4567|89, b = 987|6543|21$;②将 a, b 的交叉区域分别加到对方的交叉区域前, $a' = 123|6543|456789, b' = 987|4567|654321$;③删除 a', b' 中已在交叉区域出现过的基因,得到最终子串 $c = 126543789, d = 984567321$.

在一条染色体 c_0 中随机选取一个位置 s ,交换该染色体中处于 s 和 $s-1$ 的基因,其余不变,得到染色体 c_1 .例如, $c_0 = 123456789, i = 6$, 则 $c_1 = 123465789$.设遗传算法的最小、最大迭代次数为 $N_{\min}^{\text{GA}}, N_{\max}^{\text{GA}}$, 迭代次数等于最大迭代次数时算法终止.遗传算法迭代过程中统计子代群体的进化率,如果连续 $N_{\text{st}}^{\text{GA}}$ 次迭代,进化率都小于最小进化率 $m_{\text{st}}^{\text{GA}}$,说明此时遗传算法进化缓慢,则终止遗传算法.

2.3 融合算法中 AA 算法的构造

当一只蚂蚁完成 APCG 图中 PE 的周游后,得到一个 PE 访问顺序(一串十进制数字),该访问顺序对应的映射关系与 2.2 节中的染色体编码类似,可直接使用映射的目标函数.

参照文献[10],将各 PE 间路径的信息素初值设为

$$\tau_{i,j}^S(0) = \tau_{i,j}^C + \tau_{i,j}^G$$

式中: $\tau_{i,j}^C$ 为信息素常数,类似于 MMAS 算法中的 $\tau_{\min}$; $\tau_{i,j}^G$ 是根据遗传算法结果转换得到的信息素值.将 GA 算法中获得的一组较优解的群体 M 按照最优顺序给蚂蚁路径赋初始信息素,即

$$\begin{aligned} \tau_{i,j}^G &= \sum_{\forall m \in M} \Delta \tau_{i,j}^m \\ \Delta \tau_{i,j}^m &= \begin{cases} Q/P_m^\gamma, & (i,j) \in m \\ 0, & (i,j) \notin m \end{cases} \end{aligned} \quad (8)$$

式中: P_m^γ 为染色体 m (即一个映射编码)对应的功耗, γ 为常数,文中取 $\gamma=1$; Q 为常数,表示初始赋给节点信息素的大小.

蚂蚁 k ($k=1, \dots, n_a$) 在映射过程中,根据各路径上的信息量决定转移方向, $p_{i,j}^k(t)$ 表示在 t 时刻蚂蚁 k 由处理单元 i 转移到处理单元 j 的概率,即下一时刻映射处理单元于 j 的概率为

$$p_{i,j}^k = \begin{cases} \frac{\tau_{i,j}^\alpha(t) \eta_{i,j}^\beta(t)}{\sum_{s \in N_k} \tau_{i,s}^\alpha(t) \eta_{i,s}^\beta(t)}, & j \in N_k \\ 0, & j \notin N_k \end{cases} \quad (9)$$

式中: $\eta_{i,j}$ 为处理单元 i 与处理单元 j 之间的路径能见度,即路径的启发信息; N_k 为蚂蚁 k 尚未映射处理单元的集合,它随时间动态调整; α, β 为常数,表示信息素、启发信息的重要程度.

一个好的映射总是希望把最重要的处理单元分配到通信能力最强的资源节点上,参考文献[11]定义的启发信息如下

$$\eta_{i,j} = c(i')v(j) \quad (10)$$

$$c(i') = 1 / \sum_{j=1}^N h_{i',j}$$

$$v(j) = \sum_{i=1}^N w_{j,i} + \sum_{i=1}^N w_{i,j} \quad (11)$$

式中: i' 表示处理单元 i 在某映射编码中映射到的资源节点序号;将 $c(i')$ 定义为 i' 到所有其他资源节点的距离之和的倒数,该值越大,表示 i' 越靠近网络的中心; $v(j)$ 表示处理单元 j 与其他所有处理单元之间的通信量的大小,该值越大,表示处理单元越重要.

融合算法的 AA 中使用的交叉和变异策略与 GA 相同.根据交叉概率,选择若干组解,进行分组交叉操作,根据变异概率,对每只蚂蚁的处理单元访问顺序进行变异操作.在交叉和变异操作后,均进行适应度的判定,只有适应度提高的访问顺序才被接受下来,否则放弃操作.

蚂蚁完成一次周游后,其处理单元访问顺序对应一个映射方案,即仅在所有蚂蚁均完成所有处理单元的周游后才进行信息素更新.各路径上的信息素由下式进行调整

$$\tau_{i,j}(t+1) = \rho \tau_{i,j}(t) + \Delta \tau_{i,j}$$

$$\Delta\tau_{i,j} = \sum_{k=1}^m \tau_{i,j}^k \tag{12}$$

式中: ρ 为信息素的持久度; $\Delta\tau_{i,j}$ 为本次循环中路径 (i,j) 上的信息素增量; $\Delta\tau_{i,j}^k$ 为第 k 只蚂蚁本次周游中留在路径上的信息增量,其计算方法与式(8)的求解法相同.

3 实验结果及分析

为验证 GAM 算法的性能,对实际多媒体的应用进行了低功耗映射实验,为进一步比较 GAM 算法的优化质量和运行时间,还使用多个不同规模的随机应用进行了多组实验.实验中的多种算法均采用 Matlab 实现,并在 Intel Core(TM)2 Duo 2.2 GHz、2.0 GB RAM、Windows XP 的平台上运行.实验中 GAM 算法的主要参数如表 1 所示.

表 1 GAM 算法的主要参数

S	p_s	p_c	p_m	$N_{\min}^{\text{GA}}$	$N_{\max}^{\text{GA}}$
10	0.3	0.6	0.2	10	100
n_a	α	β	ρ	τ^c	Q
10	3	1	0.8	60	106
$N_{\text{IPV}}^{\text{GA}}$	$N_{\text{IPV}}^{\text{AA}}$	$N_{\text{die}}^{\text{GA}}$	$N_{\text{die}}^{\text{AA}}$	$N_{\min}^{\text{AA}}$	$N_{\max}^{\text{AA}}$
0.03	0.03	3	3	20	100

注: $N_{\min}^{\text{AA}}$ 、 $N_{\max}^{\text{AA}}$ 、 $N_{\text{IPV}}^{\text{GA}}$ 分别为 AA 算法的最小、最大进化代数及最小进化率; $N_{\text{IPV}}^{\text{GA}}$ 为 GA 算法的最小进化率; $N_{\text{die}}^{\text{GA}}$ 、 $N_{\text{die}}^{\text{AA}}$ 分别为 GA、AA 算法中小于最小进化率的最大次数.

3.1 实际多媒体系统

为评估 GAM 算法在真实应用上的潜力,将其应用到一个具有代表性的多媒体系统(MMS)中^[2].文献[2]将该系统划分成 40 个不同的任务,分配并调度到 25 个选定的处理单元上.将 APCG 作为输入,分别执行 GA、AA 和 GAM 算法,并对各算法的搜索过程进行了对比.

图 1 给出了搜索时各代的规格化平均功耗 P_{Ta} ,图 2 给出了各代的规格化 P_{T} .从图 1 中可以看出,在搜索初期,与 AA 算法相比,GAM 算法的收敛速度有很大提高,能够快速获得初步的优化结果,使得 AA 部分起点较好.从图 2 可以看出,与 GA、AA 算法相比,GAM 算法的优化效果优于 GA 算法、接近 AA 算法.在迭代次数方面,与 GA 算法相比,进化代数有较大减少,但比 AA 算法稍小.

在 GAM 算法中,GA 与 AA 算法的衔接及 AA 算法的具体步骤如下.

//GA 算法部分

1. 初始化 GA 算法参数(种群规模 S ,选择概率 p_s ,杂交概率 p_c ,变异概率 p_m);
2. 随机生成初始种群 P ;
3. 计算种群中个体的适应值;
4. 记录本代最佳个体及其适应值; //满足带宽、时延约束且功耗最小的方案;
5. 采用最佳保留选择策略,根据 p_s 进行种群选择;
6. 根据 p_c ,在种群中进行交叉操作;
7. 根据 p_m ,对种群的个体进行变异操作;
8. 若 GA 算法结束条件不满足,则回到 3.

//衔接部分

9. 从 P 中选择适应能力强的前 10% 的个体,放入集合 M ;
10. 对于 M 中的每个优化解 m ,按前面信息素的初始设置策略以及式(8),计算各 PE 间的路径信息素初值;
11. 利用 GA 的最优解,按式(10)、式(11)计算各 PE 间的路径启发信息;
- //AA 算法部分
12. 初始化 AA 算法参数(α,β,ρ,Q);
13. 将 n_a 只蚂蚁随机置于 n_p 个 PE 上,将各蚂蚁所在的 PE 加入各自的已映射集合 M_k 中;
14. 对每只蚂蚁 k ,按概率 $p_{i,j}^k$ 将 PE_j 置于 M_k ,重复本步骤,直到每只蚂蚁都完成所有 PE 的映射;
15. 选择若干组蚂蚁的映射方案,根据 p_c ,分组进行交叉,并进行适应度判断;
16. 对每只蚂蚁的映射方案,根据 p_m ,进行变异操作,并进行适应度判断;
17. 记录本轮迭代的最佳映射,及其对应功耗;
18. 根据本轮迭代的最佳映射,按式(12)更新信息素;
19. 若 AA 算法结束条件不满足,则回到步骤 13;
20. 输出目前最好解.

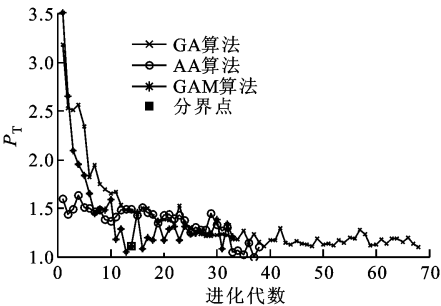


图 1 MMS 搜索时各代的规格化平均功耗

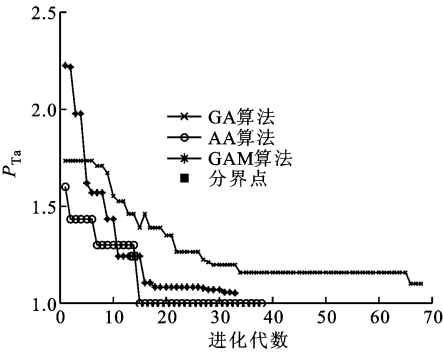


图 2 MMS 搜索时各代的规格化最优功耗

3.2 系统规模变化时的情况

为进一步比较算法的优化质量和运行时间,针对不同的资源规模,分别在 3×3 到 6×6 的 Mesh 结构上,采用 TGFF 软件^[12]产生相应规模的应用特征图、随机生成的通信带宽和延迟约束进行了多组实验,如图 3、图 4、图 5 所示.使用 TGFF 软件产生的 6 个随机 PE 应用特征图用 $a\sim f$ 表示,包含的 PE 数目分别为 9~64 个.将 6 个应用特征图及随机生成的带宽、延迟约束作为 GA 算法、文献[6]算法、AA 算法、文献[5]算法和 GAM 算法的输入,进行低功耗映射方案的求解.

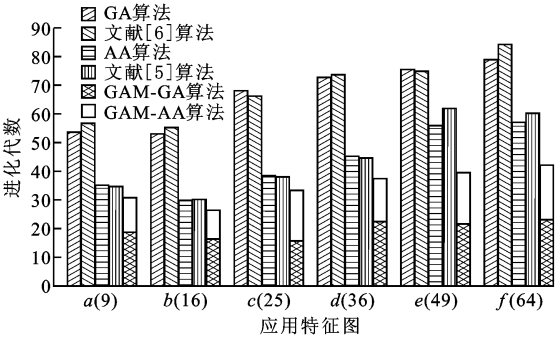


图 3 不同算法的进化代数

图 3 为 GAM 算法与其他 4 种算法的进化代数的比较,其中,GAM-GA、GAM-AA 算法分别表示 GAM 算法中 GA、AA 算法的进化代数.可以看出,GA 算法与文献[6]算法的进化代数差别不大,这是因为它们同属于遗传类算法. AA 算法与文献[5]算法的进化代数基本相同,都同属于蚂蚁类算法.在总体趋势上,随着问题规模的扩大,各类算法的进化代数都呈增加趋势.本文提出的 GAM 算法与遗传、蚂蚁算法相比,进化代数比较小.

将系统规模扩大时 GAM 算法与其他 4 种算法的运行时间 T 进行了对比,如图 4 所示.可以看出,

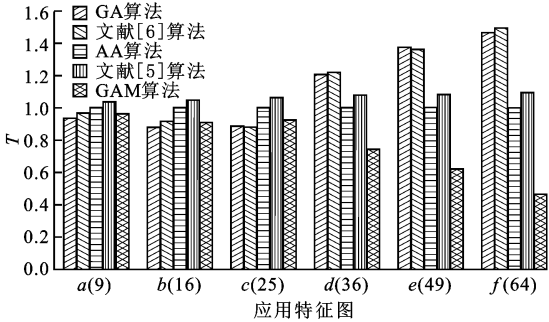


图 4 不同算法的运行时间

当任务图节点较少(9~25)时,GAM 算法与 GA 算法、文献[6]算法的搜索时间基本相同,均小于 AA 算法和文献[5]算法的搜索时间.当节点数超过 25 时,GA 算法和文献[6]算法的搜索时间开始显著延长,此时 GAM 算法优于其他 4 种算法,且随系统规模的扩大,搜索时间缩短效果较为明显.当系统规模分别为 49、64 PE 时,与其他 4 种算法相比,GAM 算法的搜索速度提高率如表 2 所示.

将 GAM 算法的搜索速度提高率定义为

$$\left(\frac{\text{用于比较的其他算法的规格化搜索时间}}{\text{GAM 算法的规格化搜索时间}}-1\right)\times 100\%$$

表 2 GAM 算法的搜索速度提高率

PE 数	GAM 搜索速度提高率/%			
	GA 算法	文献[6]算法	AA 算法	文献[5]算法
49	120.8	119.0	60.9	74.2
64	214.9	220.3	115.1	135.3

图 5 对不同系统规模下的三类算法的优化效果进行了比较,从图中可以看出,系统规模较小时,三类算法的功耗优化效果基本相同,但系统规模较大时,蚂蚁类算法的优化效果最好,GAM 算法次之,遗传类算法最差.当系统规模扩大时,遗传类算法与蚂蚁类算法的优化结果差别呈快速扩大的趋势,而 GAM 算法的优化结果与 AA 算法的差别则保持在 6.5%以内,与文献[5]算法的差别保持在 9.1%以内.综合图 3~图 5 可得出:在进化代数和运行时间

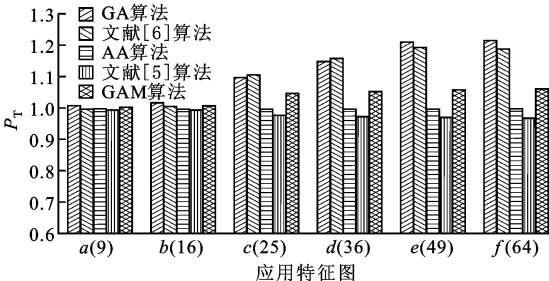


图 5 优化后的规格化功耗

方面,与遗传类算法和蚂蚁类算法相比,GAM算法均有较大优势,而在优化效果方面,GAM算法与蚂蚁类算法保持了较好的逼近度,具有更好的竞争力。

4 结 论

本文提出了一种基于遗传、蚁群融合的映射算法,以解决带宽和时延约束下的低功耗 NoC 映射问题。GAM 算法综合了遗传算法、蚂蚁算法的优点,采用核生成信息素分布,利用蚂蚁算法求解更优解,优势互补,并在蚂蚁算法中加入交叉及变异操作以加快收敛速度。实验结果表明,在具有相似优化效果的情况下,与遗传类算法和蚂蚁类算法相比,GAM 算法极大地加快了收敛速度,使搜索时间缩短,并随着 NoC 规模的增大效果更加显著。由于 GAM 算法具有优化性能好、收敛速度快的显著优点,因此适用于大规模 NoC 映射问题的求解。

参考文献:

- [1] DALLY W J, TOWLES B. Route packets, not wires: on-chip interconnection networks [C]//Proceedings of DAC 2001. Las Vegas, USA: ACM, 2001: 684-689.
- [2] HU Jingcao, MARCULESCU R. Energy-aware mapping for tile-based NoC architectures under performance constraints [C]//Proceedings of ASP-DAC 2003. Piscataway, NJ, USA: IEEE, 2003: 233-239.
- [3] TANG Lei, KUMAR S. A two-step genetic algorithm for mapping task graphs to a network on chip architecture [C]//Proceedings of Symposium on Euro Micro Digital System Design. Piscataway, NJ, USA: IEEE, 2003: 180-187.
- [4] BHARDWAJ K, JENA R K. Energy and bandwidth aware mapping of IPs onto regular NoC architectures using multi-objective genetic algorithms [C]//Proceedings of SOC 2009. Piscataway, NJ, USA: IEEE, 2009: 27-31.
- [5] 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 Shengguang, LI Li, GAO Minglun, et al. An energy- and delay-aware mapping method of NoC [J]. Acta Electronica Sinica, 2008, 36(5): 937-942.
- [6] 董文箫, 沈海斌, 全励, 等. 基于遗传算法的片上网络低功耗映射 [J]. 浙江大学学报: 理学版, 2010(6): 654-656.

DONG Wenxiao, SHEN Haibin, QUAN Li, et al. Power-aware mapping based-on genetic algorithm for network-on-chip [J]. Journal of Zhejiang University: Science Edition, 2010(6): 654-656.

- [7] 林桦, 李险峰, 佟冬, 等. 保证 QoS 的片上网络低能耗映射与路由方法 [J]. 计算机辅助设计与图形学学报, 2008, 20(4): 425-431.
LIN Hua, LI Xianfeng, TONG Dong, et al. A low energy mapping and routing approach for network-on-chip with QoS guarantees [J]. Journal of Computer Aided Design & Computer Graphics, 2008, 20(4): 425-431.
- [8] 常政威, 谢晓娜, 桑楠, 等. 片上网络映射问题的改进禁忌搜索算法 [J]. 计算机辅助设计与图形学学报, 2008(2): 155-160.
CHANG Zhengwei, XIE Xiaona, SANG Nan, et al. An improved tabu search algorithm for network-on-chip mapping [J]. Journal of Computer-Aided Design & Computer Graphics, 2008(2): 155-160.
- [9] 张剑贤, 周端, 杨银堂, 等. 一种低能耗的片上网络映射算法 [J]. 西安电子科技大学学报, 2011, 38(4): 95-100.
ZHANG Jianxian, ZHOU Duan, YANG Yintang, et al. Low energy consumption mapping algorithm for the network-on-chip [J]. Journal of Xidian University, 2011, 38(4): 95-100.
- [10] 丁建立, 陈增强, 袁著祉. 遗传算法与蚂蚁算法的融合 [J]. 计算机研究与发展, 2003, 40(9): 1351-1356.
DING Jianli, CHEN Zengqiang, YUAN Zhuzhi. On the combination of genetic algorithm and ant algorithm [J]. Journal of Computer Research and Development, 2003, 40(9): 1351-1356.
- [11] 易伟, 王佳文, 潘红兵, 等. 基于蚁群混沌遗传算法的片上网络映射 [J]. 电子学报, 2011, 39(8): 1832-1836.
YI Wei, WANG Jiawen, PAN Hongbing, et al. Ant colony chaos genetic algorithm for mapping task graphs to a network on chip [J]. Acta Electronica Sinica, 2011, 39(8): 1832-1836.
- [12] DICK R P, RHODES D L, WOLF W. TGFF: task graphs for free [C]//Proceedings of CODES/CASHE. Seattle, USA: ACM, 1998: 97-101.

(编辑 管咏梅 武红江)