

基于蚁群混沌遗传算法的片上网络映射

易 伟,王佳文,潘红兵,李 丽

(南京大学微电子设计研究所,江苏南京 210093)

摘 要: 蚁群算法可以在兼顾功耗和负载平衡的情况下进行任务映射,但是由于传统蚁群算法对初始化参数的设置比较敏感,所以使用遗传算法来调整蚁群算法参数,在蚁群算法陷入局部最优时引入混沌模型来修改蚁群参数.修改后的算法在能耗方面相较于传统算法改善了 11%,在负载平衡方面改善了 1%,两者联合优化改善了 4%.

关键词: 片上网络; 蚁群算法; 遗传算法; 混沌模型; 映射算法

中图分类号: TN47 **文献标识码:** A **文章编号:** 0372-2112 (2011) 08-1832-05

Ant Colony Chaos Genetic Algorithm for Mapping Task Graphs to a Network on Chip

YI Wei, WANG Jia-wen, PAN Hong-bing, LI Li

(Institute of VLSI Design, Nanjing University, Nanjing, Jiangsu 210093, China)

Abstract: Ant colony algorithm is one of the solutions to task mapping, which optimizes the communication energy and the distribution of link load. Because the ant colony algorithm is very sensitive to the initialization of the parameters, we use genetic algorithm to set the parameter. To avoid getting the local optimal solution, we use chaos module to optimizing the parameter. The algorithm we improved gives a solution, which is 11% lower than the traditional algorithm on power, 1% better than the traditional algorithm on load balance and 4% better when optimizing both of them.

Key words: network-on-chip; ant colony algorithm; genetic algorithm; chaos module; mapping algorithm

1 引言

由于片上硬件架构越来越复杂,传统拓扑架构效率越发低下,因此人们将计算机网络移植到芯片设计中来解决传统总线带来的问题, NoC (network on chip) 概念也因此被提出. 功耗, 通信和速度等问题就一直是 NoC 设计中的主要约束, 然而如何将多个任务映射到 NoC 网络中的各个节点上, 并安排各个任务的执行顺序, 成为 NoC 领域需要解决的首要问题.

本文主要基于蚁群算法^[1]来解决带宽约束和低功耗问题, 但是由于传统的蚁群算法收敛性很大程度上依赖于初始化参数^[2]的设置, 怎样选取合适的初始化参数, 加速蚁群算法收敛也就显得相当重要. 本文在蚁群算法的迭代过程中通过使用遗传算法^[3]选择合适的种群, 通过杂交和变异来更新蚁群参数. 同时, 由于蚁群算法容易导致局部最优, 所以在达到局部最优值时通过引入混沌模型^[4]并增大变异概率, 避免算法停滞.

2 相关工作

如何使功耗最小化并满足带宽约束问题正逐渐成为 NoC 领域的研究热点, 很多研究人员都在研究相关技术. 文献[5,6]分析了通讯功耗对整个系统功耗的影响, 并构建了功能分析模型. 文献[7]构建了系统延时模型, 并采用两步式遗传算法进行任务图映射. 首先执行粗粒度映射, 用任务类型进行映射, 然后将第一步得到的集合进行细粒度映射, 将它们映射到各个节点. 文献[8]采用混沌遗传算法 CGMAP, 通过混沌操作克服了遗传算法的早熟现象, 得到更低的通信能耗, 节能效果比现有的其他方法优越. 文献[9]使用蚁群算法实现了面向能耗和延时的 NoC 映射.

3 NoC 映射模型

NoC 映射就是在给定 NoC 体系结构和 IP 核之间通信量的基础上, 按照一定的优化规则将 IP 核分配到

NoC 中一个唯一的资源节点上,实现特定应用,并且使目标成本最小化的过程。

3.1 映射过程

映射过程的示意图如图 1 所示.图中, P_i 表示执行第 i 个任务的 IP 核, R_i 表示第 i 个路由器节点.映射就是在保证目标成本最优的情况下,将不同的 P_i 挂接到不同的 R_i 上,实现特定的应用。

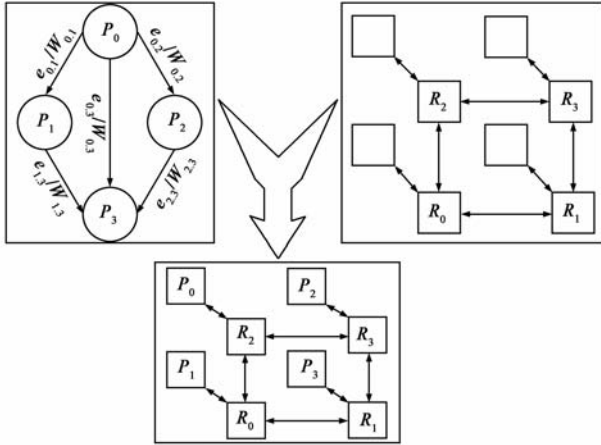


图1 映射过程示意图

在不同的优化目标下会产生不同的映射结果,而映射结果影响着系统方案的性能和成本,因此映射要根据具体需求定义不同的目标函数,下面对目标函数进行简单介绍。

3.2 能耗和延时建模

由文献[5,9]可知系统能耗为：

$$E(C) = \sum_{i=1}^N \sum_{j=1}^N w_{i,j} \times h_{i,j} \quad (1)$$

式(1)中 $w_{i,j}$ 为通信量, N 为节点数, $h_{i,j}$ 表示从节点 r_i 到节点 r_j 经过的跳数(hop).从图 2 可以看出,存在多条路径使得从 A 到 B 的距离等于曼哈顿距离。

为了简化设计模型,本文假设 $h_{i,j}$ 为两节点之间的曼哈顿距离($|x_n - x_j| + |y_n - y_j|$).所以优化能耗的目标是最小化各节点间的加权曼哈顿距离之和,就是将通信任务较重的几个 IP 核分配到紧邻的节点,实现通信近邻化。

文献[7]指出从源节点 i 到目标节点 j 的延时如式

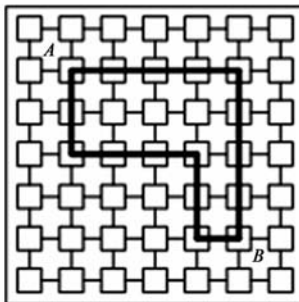


图2 从A到B的最短路径

(2)所示：

$$T_{i,j} = (T_b + T_w) \times h_{i,j} + T_b(B - 1) \quad (2)$$

式(2)中, T_b 为无拥塞时一帧数据通过一个开关和一条链路所需要的时间, T_w 为存在拥塞时包头在开关节点处平均等待时间, B 为数据包中包含的帧数.其中 T_b 和 B 为常数.由式(2)可以看出, $T_{i,j}$ 依赖于参数 T_w 和 $h_{i,j}$.在带宽约束下,减小数据的传输延时 T_w 可以通过缓解数据拥塞实现,而缓解拥塞的关键就在于平衡链路负载.平衡链路负载就是最小化链路负载方差.所以优化延时以链路负载方差 ($VAR(L)$)作为指标：

$$VAR(L) = \sum_{i=1}^M [Load(l_i) - Load(l_{avg})]^2 / M \quad (3)$$

式(3)中 M 为链路总数, $Load(l_i)$ 为链路 l_i 的负载量, $Load(l_{avg})$ 为平均链路负载量.显然,通信延时优化就是将通信任务均匀分布。

从式(1)、(3)看出,一方面希望优化系统能耗 $E(C)$,另一方面希望平衡链路负载 $VAR(L)$.为了达到联合优化的目的定义目标函数如下：

$$cost = \lambda \times E(C) + (1 - \lambda) VAR(L) \quad (4)$$

式(4)中, λ 是比例系数,用于调节通信能耗和延时在函数中的比重,取值范围为 $(0, 1)$.当 $\lambda = 1$ 时,优化通信能耗; $\lambda = 0$ 时,优化通信延时.在实际使用中需要通过具体需要调整 λ 值,比如在通信能耗的优化更重要时, λ 取值 $(0.5, 1]$; 通信延时更重要时, λ 取值 $[0, 0.5)$, λ 的具体取值根据实际需要进行调整。

3.3 蚁群遗传算法

Maniezzo, V 等在文献[10]使用蚁群算法解决问题以来,很多研究人员在此基础上提出改进算法.文献[11]对各种算法的性能进行了比较.本文对蚁群算法进行修改后用于 NoC 映射。

基于遗传的蚁群算法为了解决对初始化参数的依赖,通过选择合适种群的参数进行杂交,并以一定的概率变异从而不断对参数进行更新.为了避免算法陷入局部最优,引入混沌模型^[4],加大突变概率.下面简单介绍算法主要步骤。

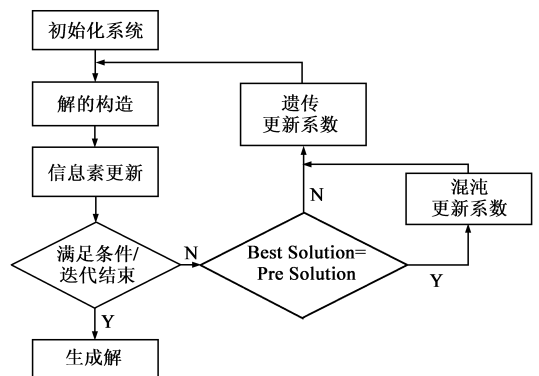


图3 流程图

3.3.1 初始化系统

本文的初始参数值在文献[2]的基础上根据具体应用做了简单调整. 由于在以后的迭代中会根据遗传算法选择合适的种群更新参数, 所以这里的简单设置主要为了避免结果过于发散.

3.3.2 解的构造

在构造解的过程中, 假设第 k 只蚂蚁在第 t 次循环时以概率 $p_{i,j}^k(t)$ 将 IP 核 j 分配到资源节点 i 上.

$$p_{i,j}^k(t) = \begin{cases} \frac{[\tau_{i,j}(t)]^\alpha \times [\eta_{i,j}(t)]^\beta}{\sum_{j \notin tabu_k} [\tau_{i,j}(t)]^\alpha \times [\eta_{i,j}(t)]^\beta}, & j \notin tabu_k \\ 0, & j \in tabu_k \end{cases} \quad (5)$$

$tabu_k (k = 1, 2, \dots, M)$ 用于记录蚂蚁 k 已经分配过的 IP 核. $\tau_{i,j}(t)$ 表示在第 t 次循环时, 将 IP 核 P_j 分配到资源节点 R_i 上的信息素强度, 信息素的更新将在 3.3.3 详细介绍. $\eta_{i,j}(t)$ 是指 IP 核 P_j 分配到资源节点 R_i 上的启发信息. 式中 α, β 分别表示信息素强度与启发信息的重要程度, α, β 在第一次迭代时由初始化系统设置初值, 以后在蚁群算法每执行一次迭代后由遗传算法更新.

一个好的映射总是希望把最重要的 IP 核分配到通信能力最强的资源节点上. 因此, 定义启发信息如下所示:

$$\eta_{i,j}(t) = VIP(j) / Comm(i) \quad (6)$$

$\eta_{i,j}(t)$ 体现出将 IP 核 v_j 分配给资源节点 R_i 的合理程度. $Comm(i)$ 表示 R_i 在 NoC 中的通信能力, $VIP(i)$ 代表 P_i 在应用特征图中的重要程度.

$$Comm(i) = \sum_{j=1}^N h_{i,j} \quad (7)$$

$$VIP(i) = \sum_{j=1}^N w_{i,j} + \sum_{j=1}^N w_{j,i} \quad (8)$$

求解步骤如下:

(1) 从 IP 核集合 P 中按概率 $p_{i,j}^k(t)$ 选择一个未分配的核 P_i 分配到 R_1 上, 并将该核添加到 $tabu$ 中.

(2) 重复执行 N 步, 直到所有的核都分配到相应的资源上, $tabu$ 满.

一次蚁群迭代循环完成后, 在所有的蚂蚁中选择最优解, 然后进入下一次循环迭代.

3.3.3 信息素更新

随着程序运行, 以前留下的信息素逐渐消逝, 同时在每只蚂蚁选择的过程中又有新的信息素加入. 用参数 ρ 表示信息素的持久程度 ($0 < \rho < 1$), $\Delta\tau_{i,j}$ 为信息素增量.

$$\Delta\tau_{i,j} = \sum_{k=1}^M \Delta\tau_{i,j}^k \quad (9)$$

$\Delta\tau_{i,j}^k$ 表示本次循环中蚂蚁 k 在分配路径 ($P_j \rightarrow R_i$) 上留下的信息量, 计算公式为:

$$\Delta\tau_{i,j}^k = \begin{cases} \frac{1}{\text{cost}(k)}, & \text{map}(k) \text{ includes } (P_j \rightarrow R_i) \\ 0, & \text{else} \end{cases} \quad (10)$$

式中, $\text{cost}(k)$ 为蚂蚁 k 完成的分配方案的成本. 最优解具有最小的成本, 所以对信息量的贡献最大. 在所有蚂蚁完成一次循环以后, 根据下式对各分配路径 ($P_j \rightarrow R_i$) 上的信息量作更新.

$$\tau_{i,j}(t+1) = \rho \times \tau_{i,j}(t) + \Delta\tau_{i,j} \quad (11)$$

3.3.4 遗传更新系数

当蚁群算法得到的解不能满足要求时, 使用遗传算法更新蚁群算法的参数. 为了避免遗传算法复杂的编码解码过程, 这里使用实数 α, β, Q 对各个蚂蚁进行基因编码. 那么每个蚂蚁的染色体即用 (α, β, Q) 表示.

$$\alpha = x \times \alpha_f + (1-x) \times \alpha_m \quad (12)$$

$$\beta = x \times \beta_f + (1-x) \times \beta_m \quad (13)$$

$$Q = x \times Q_f + (1-x) \times Q_m \quad (14)$$

初始状态下杂交比例系数 x 取值为 0.5, 随着程序的运行, 当达到局部最优或者是变异的时候, 根据混沌模型对 x 值进行调整, 如 3.3.5 所示.

在下面伪代码中简单介绍更新系数的遗传算法.

for i varying from 0 to $M/2$

begin

dad = Roulette();

mom = Roulette();

Crossover(mom, dad);

Mutate(mom);

Mutate(dad);

Save(mom);

Save(dad);

End

通过轮盘概率 (*Roulette*) 实现优胜劣汰, 选择两个合适的种群, 将它们的参数 (α, β, Q) 按照一定的比例进行杂交 (*Crossover*). 然后再按照一定的概率对它们的参数进行变异 (*Mutate*), 将结果保存. 重复执行 $M/2$ 次后, 所有蚂蚁的参数都被更新, 然后进入下一轮蚁群算法的迭代过程.

3.3.5 混沌更新系数

当本次的最优解与上一次的最优解相等时, 意味着可能陷入了局部最优, 为了避免算法停滞, 利用混沌对初值变化极度敏感的特性, 将此时的系数进行混沌操作. 如文献[4]提出:

$$cx_i^{k+1} = 4cx_i^k(1-x_i^k) \quad (15)$$

式中, cx_i^k 表示 k 次迭代得到的第 i 个混沌变量, $cx_i^k \in (0, 1)$ 而且 $cx_i^k \notin \{0.25, 0.5, 0.75\}$. 按照式(15)对遗传算法的杂交比例系数 x 进行更新.

4 实验与分析

为了证明好的映射算法可以有效的降低通信功耗和平衡负载,本文进行了一系列测试.程序源代码用 C++ 编写,在 Linux gcc 环境下运行.

4.1 MPEG4 解码器任务图

为了评价算法的实际应用,本文参考文献[12],将算法应用在 MPEG4 解码器上.通过将 MPEG4 解码器分解为 12 个任务,然后把它们交给与 12 个 IP 核分别执行.图 4 为他们的通信任务图.

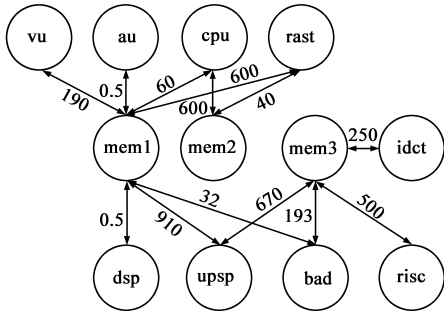


图4 MPEG4 解码器任务图

从表 1 可以看出,当 $\lambda = 0.5$ 时,对通信能耗的优化明显弱于 $\lambda = 0$ 时的通信能耗优化,而对链路负载的优化弱于 $\lambda = 1$ 时的链路负载优化. $\lambda = 0.5$ 的优化在于联合考虑通信能耗和链路负载平衡.也就是说可以通过对

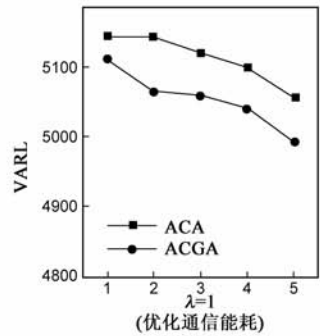
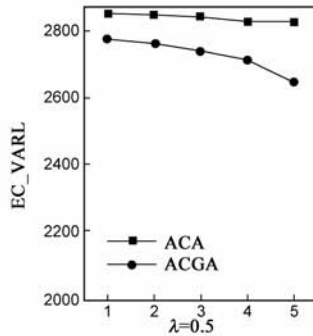
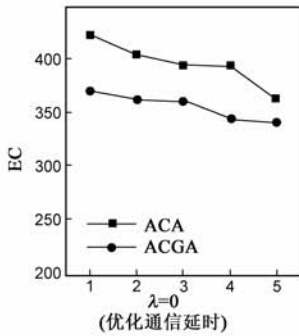


图6 蚁群遗传算法与传统蚁群算法结果比较

5 结论

本文采用蚁群遗传算法来解决负载平衡和低功耗下的 NoC 映射问题.文中提出的算法通过引入遗传算法来解决蚁群算法对参数过于敏感的问题,并引入混沌模型来解决算法停滞问题.在单一目标优化时,其映射结果在通信能耗和链路负载平衡方面分别比参考模型优化 11% 和 1%,而混合目标优化则可以取得 4% 的提高.在下一步的研究过程中,需要进一步考虑如何建立一个更为精确的模型,同时根据实际应用中通信延时和通信能耗的重要程度建立一个更好的目标函数.

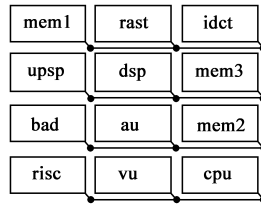


图5 NoC映射结果

表 1 MPEG4 解码器不同映射结果

	$E(C)$	$VAR(L)$
$\lambda = 1$	8082	98918.6
$\lambda = 0.5$	10744	95666.3
$\lambda = 0$	11044	77172.6

λ 值进行调整,在满足通信能耗要求的情况下,优化负载平衡.或者是在满足带宽要求的情况下,优化通信能耗.

图 5 为 $\lambda = 0.5$ 时的映射结果.比图 4 的直接放置各个任务图在通信能耗上降低了 31%,链路负载上优化了 56%.

4.2 随机任务图

为了充分体现本文所提出的算法的优势,使用 TGFF^[13]生成一系列的随机任务图,并侧重不同的优化目标($\lambda = 1, 0.5, 0$)对 NoC 进行了映射优化,其将结果同文献[9]进行了比较.

图 6 所示为 $\lambda = 1, 0.5, 0$ 时各映射方案的通信能耗和链路负载方差与参考方案的对比结果.

从图中可以看出蚁群遗传算法(ACGA)明显优于蚁群算法(ACA). $\lambda = 1$ 时映射方案比参考方案 cost 降低 11%, $\lambda = 0.5$ 时映射方案比参考方案降低 4%, $\lambda = 0$ 时映射方案比参考方案降低 1%.

参考文献

[1] 段海滨. 蚁群算法原理及其应用[M]. 北京: 科学出版社, 2005. 24 - 42.
Haibin Duan. Ant Colony Algorithms: Theory and Applications [M]. Beijing: Science Press, 2005. 24 - 42. (in Chinese)

[2] 吴春明, 陈治, 姜明. 蚁群算法中系统初始化及系统参数的研究[J]. 电子学报, 2006, 34 (8): 1530 - 1533.
Wu Chun-ming, Chen Zhi, Jiang Ming. The research on initialization of ants system and configuration of parameters for different TSP problems in ant algorithm [J]. Acta Electronica Sinica, 2006, 34(8): 1530 - 1533. (in Chinese)

- [3] Abbattista F, Abbattista N, Caponetti L. An evolutionary and cooperative agents model for optimization [A]. International Conference on Evolutionary Computation [C]. Perth: ICEC, 1995.2. 668 – 671.
- [4] Xuefeng F. Yan, Dezhao Z. Chen, Shangxu X. Hu. Chaos-genetic algorithms for optimizing the operating conditions based on RBF-PLS model [J]. Computers & Chemical Engineering, 2003, 27(10): 1393 – 1404.
- [5] Hu J, Marculescu R. Energy-aware communication and task scheduling for network-on-chip architectures under real-time constraints [A]. Design, Automation and Test in Europe Conference and Exhibition [C]. Paris: DATE, 2004. 234 – 239.
- [6] Ye T T, Benini L, Micheli G De. Analysis of power consumption on switch fabrics in network routers [A]. 39th Design Automation Conference [C]. New Orleans, LA: ACM Press, 2002. 524 – 529.
- [7] Tang Lei, Shashi Kumar. A two-step genetic algorithm for mapping task graphs to a network on chip architecture [A]. Euromicro Symposium on Digital System Design [C]. Antalya, Turkey: DSD, 2003. 180 – 187.
- [8] Moein-Darbari F, Khademzade A, Gharooni-Fard G. CGMAP: a new approach to network-on-chip mapping problem [J]. IEICE Electronics Express, 2009, 6(1): 27 – 34
- [9] 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 Sheng-guang, Li Li, Gao Ming-lun, et al. An energy and delay-aware mapping method of NoC [J]. Acta Electronical Sinica, 2008, 36(5): 937 – 942. (in Chinese)
- [10] Maniezzo, V.; Colomi, A. The ant system applied to the quadratic assignment problem [J]. IEEE Transactions on Knowledge and Data Engineering, 1999, 11(5): 769 – 778.
- [11] T Stützle, M Dorigo. New Ideas in Optimization [M]. UK Maidenhead: McGraw-Hill, 1999. 33 – 55.
- [12] J Hu, R Marculescu. Energy and performance-aware mapping for regular NoC architectures [J]. IEEE Trans On Computer-Aided Design of Integrated Circuits and Systems, 2005, 24 (4): 551 – 562.

- [13] R P Dick, D L Rhodes, W Wolf. TGFF: Task graphs for free [A]. 1998 6th International Workshop on Hardware/Software Codesign [C]. Seattle: CODES, 1998. 97 – 101.

作者简介



易 伟 男, 1986 年生, 现为南京大学电子科学与工程学院在读硕士研究生, 主要从事多核系统中固件的实现以及并行任务的调度和映射问题研究。

E-mail: sc.yiwei@gmail.com



王佳文 男, 1985 年生, 现为南京大学电子科学与工程学院在读博士研究生, 主要从事多核处理器体系架构、片上网络通信架构及通信协议研究。

E-mail: jiawenwang@nju.edu.cn



潘红兵 男, 1971 年生, 现为南京大学电子科学与工程学院副教授, 主要从事多核系统中并行软件的任务划分及实现技术的研究。

E-mail: phb@nju.edu.cn



李 丽 女, 1975 年生, 现为南京大学电子科学与工程学院教授, 主要从事多核处理器芯片体系结构及实现方法、SoC-IP 设计方法的研究。

E-mail: lili@nju.edu.cn