

基于 GA-MMAS 算法的片上网络低能耗映射

穆屹峰, 吴 宁, 葛 芬

(南京航空航天大学 电子信息工程学院, 江苏 南京 210016)

摘 要: 针对片上网络的低能耗 IP 映射问题, 提出了一种基于该混合算法的映射算法——GA-MMAS 算法. 该算法首先利用优先映射通讯量大的核的方法代替启发因子来确定启发信息, 以改善最大最小蚁群算法(MMAS)的最优解, 然后将 MMAS 与 GA 结合, 利用 GA 的快速性得到 MMAS 初始信息素, 来弥补 MMAS 的前期信息素匮乏的缺点, 提高了最优解的精确性, 从而得到通信能耗更低的映射结果. 实验表明, 该映射算法与初始映射相比, 能耗可以节省 36%~60%, 与已有的 GA, AS 和 MMAS 相比分别能节省 3%~25%, 10%~30% 和 3%~30% 的能耗.

关键词: 能耗; 混合算法; 映射; 片上网络

中图分类号: TP302

文献标识码: A

文章编号: 1000-7180(2011)12-0025-05

Low Energy Mapping for Network-on-Chip Using GA-MMAS Algorithm

MU Yi-feng, WU Ning, GE Fen

(College of Electronic and Information Engineering, Nanjing University of
Aeronautics & Astronautics, Nanjing 210016, China)

Abstract: Aiming at the problem of low energy IP mapping for Network-on-Chip, an algorithm named GA-MMAS is proposed by studying the hybrid algorithm which combines the Genetic Algorithm (GA) with Ant System Algorithm (AS). Firstly in this algorithm, instead of using heuristics to obtain the elicitation information in MMAS, we implement this behavior via priority mapping of IP core with larger communication volume, which improves the optimal solution of MMAS. Then with the combination of MMAS and GA, the advantage of speed in GA makes compensation to the lack of pheromone in the early stage of MMAS, and in turn enhancing the accuracy of optimal solution, which leads to lower energy consumption. Experimental results show that the algorithm saves about 36%~60% of energy consumption compared to initial mapping. Compared to GA, Ant System Algorithm (AS) and MMAS, the algorithm saves about 3%~25%, 10%~30% and 3%~30%.

Key words: energy consumption; hybrid algorithm; mapping; network-on-chip

1 引言

在片上网络(Network-on-Chip, NoC)设计中, NoC 的映射是一个非常重要的步骤. 不同的映射结果, 对于系统的执行时间、通信延时、通讯能耗等性能有着重要的影响^[1].

NoC 映射实质上是一个二次分配问题, 属于 NP-完全问题, 寻找一种算法能在合理的时间内得

到一个最优的结果是设计者面临的一个难题. 在已有的相关研究中, 文献[2-3]中采用遗传算法(Genetic Algorithm, GA)优化系统的通讯能耗, 但是 GA 对于系统中反馈信息利用不够, 因此会导致求解到一定范围时往往做大量无为的冗余迭代. 文献[4-5]则采用蚁群算法(Ant System Algorithm, AS)和最大最小蚁群算法(MAX-MIN Ant System Algorithm, MMAS), 以最小化能耗和延时为目标

实现 IP 核映射,但是蚁群算法在初期信息素匮乏,因此会导致求解速度较慢。

基于对上述映射算法和 GA 与 AS 融合的混合算法的研究,本文提出一种 GA-MMAS 算法,在改进 MMAS 算法的基础上,将 GA 和 MMAS 结合,实现 NoC 的低能耗 IP 映射,并通过实验验证了算法性能。

2 NoC 低能耗 IP 映射问题

NoC 低能耗 IP 映射是指:在给定 IP 通讯图和 NoC 拓扑结构后,将 IP 核分配到 NoC 的资源节点上,以使整个系统的通讯能耗最小^[6]。本文采用二维网孔(2Dmesh)结构和确定性 XY 路由^[7]。能耗模型采用改进的比特数据能耗模型:

$$E_{\text{bit}}^{i,j} = n_{\text{switch}} \times E_{\text{sbit}} + (n_{\text{switch}} - 1) \times E_{\text{Lbit}} \quad (1)$$

式中, $E_{\text{bit}}^{i,j}$ 表示从节点 i 到节点 j 传输一位数据的平均能耗。

为了形式化描述映射问题,首先给出以下定义:

定义 1 应用特征图(APCG)。图中每个顶点 c_i 代表一个已选的 IP 核,每条边 $a_{i,j}$ 代表从 c_i 到 c_j 的通信。 $v(a_{i,j})$ 表示顶点 c_i 到 c_j 的通讯量。 $b(a_{i,j})$ 表示顶点 c_i 到 c_j 的带宽要求。

定义 2 结构特征图(ARCG)。图中每个顶点 t_i 代表结构中的一个节点,每条边 $p_{i,j}$ 代表从 t_i 到 t_j 的路由路径。 $e(p_{i,j})$ 表示从节点 t_i 到 t_j 发送一个比特数据的能耗(焦耳)。 $L(p_{i,j})$ 表示组成路径 $p_{i,j}$ 的线路集合。

使用上述定义后,低能耗的 IP 核映射问题就是指,已知 APCG 和 ARCG 满足 $\text{size}(\text{APCG}) \leq \text{size}(\text{ARCG})$,找出一种从 APCG 到 ARCG 的映射函数 $\text{map}()$ 实现:

$$\min \{ \text{Energy} = \sum_{\forall a_{i,j}} v(a_{i,j}) \times e(P_{\text{map}(c_i), \text{map}(c_j)}) \} \quad (2)$$

同时要满足:

$$\forall c_i \in C, \text{map}(c_i) \in T \quad (3)$$

$$\forall c_i \neq c_j \in C, \text{map}(c_i) \neq \text{map}(c_j) \quad (4)$$

$$B(l_k) \geq \sum_{\forall a_{i,j}} b(a_{i,j}) \times f(l_k, P_{\text{map}(c_i), \text{map}(c_j)}) \quad (5)$$

$B(l_k)$ 表示路线 l_k 的带宽,而且:

$$f(l_k, P_{m,n}) = \begin{cases} 0 & l_k \notin L(P_{m,n}) \\ 1 & l_k \in L(P_{m,n}) \end{cases} \quad (6)$$

条件(3)和条件(4)说明一个 IP 核只能被映射到一个节点上,而且一个节点也只能放一个 IP 核。

不等式(5)保证每条线路的负载都不会超过它的带宽。

3 GA-MMAS 映射算法

GA-MMAS 算法的基本思想是在前期采用具有精英保留的 GA,充分利用它的快速性、随机性得到映射结果,然后用该结果产生 MMAS 的初始信息素分布;在后期采用改进启发信息的 MMAS,在有一定初始信息素分布的情况下充分利用它的正反馈性、精确性等特点得到最终映射结果。算法主要的三个步骤是:(1)遗传算法;(2)初始化信息素;(3)最大最小蚁群算法。

3.1 GA-MMAS 中的遗传算法

算法的开始使用遗传算法,得到一些较好的结果。

(1)染色体编码和初始群体

NoC 映射问题中的染色体表示的是 IP 核,它的长度应与 IP 通讯图中的 IP 核的数量相同。每一个基因值表示 IP 核映射到 NoC 中哪个资源节点上^[3]。图 1 所示为一个染色体编码例子。

5	1	3	4	0	2	8	7	6
---	---	---	---	---	---	---	---	---

图 1 染色体编码

图中,染色体长度为 9,表示 9 个 IP 核。对于 IP 核 0,其基因值为 5 表示 IP 核 0 映射在位置为 5 的节点上。每一个个体所表示的映射结果都必须满足式(5),如果不满足则为非法个体,适应度设为 0。

初始群体是由根据编码形式随机生成的染色体组成。

(2)适应度计算

适应度定义为通讯能耗的倒数,即:

$$\text{fitness} = \frac{1}{\sum_{\forall a_{i,j}} v(a_{i,j}) \times e_{\text{map}(c_i), \text{map}(c_j)}} \quad (7)$$

(3)交叉和变异

根据条件(3)和(4),本文的染色体编码形式中,所有基因值不能相同,因此为防止一个基因值重复出现,交叉操作采用改进的顺序交换法,变异操作采用对位变换法^[3]。

改进的顺序交换法如图 2 所示。

先要在要交叉的染色体上随机找到交换区域(灰色区域),将 B 的交换区加到 A 前面,然后将 A 中与 B 交换区中相同的基因值去掉得到新的染色体 A*。同理,可以得到新染色体 B*。

对位变换法是在染色体上随机找到两个基因,

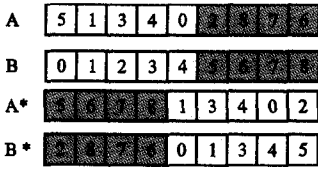


图2 改进的顺序交换法

然后对换基因值。

(4) 保留精英

每一代群体形成后,都会用式(7)计算每一个个体的适应度,然后选出适应度最大的和最小的作为当代最佳个体和最差个体。当代最佳个体的适应度与当前最佳个体的适应度比较,选出适应度大的作为新的当前最佳个体,并用其代替当代最差个体,参与下一次繁殖。

(5) 参数设置

遗传算法的参数包括种群大小,染色体长度,最大进化代数,交叉概率和变异概率。通过测试,其中,种群大小设为 100,最大进化代数为 1 000,交叉概率为 0.95,变异概率为 0.05,染色体长度与 IP 核的数目相同。

3.2 初始化信息素

遗传算法结束后,根据 2Dmesh 结构的大小($x \times y$)在最后一代中选择适应度为前 $x \times y/2$ 的个体组成矩阵 T_1 ,矩阵的每一行是一个个体。然后再随机生成 $x \times y/2$ 个个体组成矩阵 T_2 。 T_1 和 T_2 合起来组成矩阵 $P^{[8]}$ 。例如一个 2×2 的 2Dmesh 结构,假设遗传算法的最后一代中适应度最大的两个个体为 2013 和 2031,则 $T_1 = \begin{bmatrix} 2 & 0 & 1 & 3 \\ 2 & 0 & 3 & 1 \end{bmatrix}$;然后随机生成

两个个体 1302 和 1230,则 $T_2 = \begin{bmatrix} 1 & 3 & 0 & 2 \\ 1 & 2 & 3 & 0 \end{bmatrix}$;最

$$\text{后 } P = \begin{bmatrix} 2 & 0 & 1 & 3 \\ 2 & 0 & 3 & 1 \\ 1 & 3 & 0 & 2 \\ 1 & 2 & 3 & 0 \end{bmatrix}.$$

P 中核 i 映射节点 j 出现的次数记为 $p_{i,j}$,最后将 $p_{i,j} / (x \times y)$ 作为信息素 $\tau_{i,j}$ 的初值。例如在上面的 P 中核 0 映射在节点 2 上出现了两次,则 $p_{0,2} = 2$, $\tau_{0,2} = 0.5$ 。例子中的信息素初值矩阵应为

$$\tau = \begin{bmatrix} 0 & 0.5 & 0.5 & 0 \\ 0.5 & 0 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0 & 0.5 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{bmatrix}.$$

3.3 最大最小蚁群算法

算法的最后,使用 MMAS 得到最终映射结果。

本文对 MMAS 也做了改进,去掉启发因子,转而为优先映射通讯量大的 IP 核的方法确定启发信息。

(1) 蚂蚁的设置

每只蚂蚁都有一个路径表,记录 IP 核的映射节点(类似于遗传算法中的染色体)。路径表的位置表示 IP 核,位置上的数值表示 IP 核映射在哪个节点上。每只蚂蚁都有一个节点状态表,用于存放未映射过的节点位置。初始化时,状态表应为满,每映射一个 IP 核,状态表中就要去掉该 IP 核映射的节点,直到本只蚂蚁完成此次循环。再次循环时,应将状态表填满。

(2) 参数设定

MMAS 的参数包括蚂蚁的个数 m ,信息素的重要程度 α ,信息素的持久度 ρ 和最大循环次数 M 。经过测试,其中, m 设为 IP 核的数量, α 为 0.8, ρ 为 0.8, M 为 1 000。

(3) 选择策略

设 $P_{i,j}^k(t)$ 表示在第 t 次循环时蚂蚁 k 将 IP 核 i 映射到节点 j 上的概率。

$$P_{i,j}^k(t) = \begin{cases} \frac{(\tau_{i,j(t)})^\alpha}{\sum_{r \in \text{allowed}} (\tau_{i,r(t)})^\alpha} & j \in \text{allowed} \\ 0 & j \notin \text{allowed} \end{cases} \quad (8)$$

式中, $\tau_{i,j}$ 表示 (i,j) 对上的信息素强度,其初值由 3.2 节给出。 allowed_k 为第 k 只蚂蚁的节点状态表。这里没有用到启发因子 η ,原因后面给出。

本文引入如下转移规则来选择映射节点:

设 q 为 $(0,1)$ 的随机数, q_0 为 $[0,1]$ 的常数。若 $q < q_0$,则根据式(9)选择映射节点,即概率最大的节点被选择;

$$j = \arg \max_{s \in \text{allowed}_k} (\tau_{i,s}^\alpha) \quad (9)$$

否则,按照式(8)采用轮盘赌方法选择,即由式(8)计算概率做累计统计,然后产生随机数,该随机数在哪一个累计概率中,则该概率对应的节点就被选为映射节点。这样做,增加了搜索的多样性,避免过早的陷入搜索停滞。同时,为了减小陷入局部最优的概率,对 q_0 做出如下设定^[8]:当循环次数小于 300 时, q_0 为 0.8;当循环次数在 300 到 800 之间, q_0 为 0.6;当循环次数大于 800 时, q_0 为 0.3。

(4) 解的构造

首先解释概率公式中为什么没有 η 。根据文献[4-5],IP 核 i 映射到节点 j 的启发因子 $\eta_{i,j}$ 的计算公式为

$$\eta_{i,j} = \frac{f_i}{d_j} \text{ 或者 } \eta_{i,j} = \frac{1}{f_i \times d_j} \quad (10)$$

式中, f_i 表示 IP 核 i 与其他 IP 核的通讯量和, d_j 表示节点 j 与其他节点的距离和. 由式可知 η 表明通讯量大的核应该映射到 NoC 结构的中间位置, 使用 η 是为了更快地找到最优解. 但是蚁群算法在构造映射问题的解时, 是比较同一个核映射到不同位置的概率, 并没有在不同的核之间比较, 因此在计算 $\eta_{i,j}$ 时, f_i 是一个定值, 变化的只有 d_j , 这样 η 就表示每一个核都尽量放在中间位置, 就非常有可能造成通讯量小的核放在了中间位置, 而通讯量大的核放在了边缘位置, 反而减慢了求解速度, 甚至找不到最优解, 这样并没有达到 η 预期的目的. 所以算法没有使用 η , 而是采用优先映射通讯量大的核的方法代替它确定启发信息, 因为通讯量大的核的位置对能耗的影响比其他核要大, 先确定它的位置, 有利于尽快找到最优解, 加快收敛速度. 因此求解步骤如下:

① 根据 IP 通讯图将每个 IP 核与其他 IP 核的通讯量和求出.

② 定义一个优先级队列, 将 IP 核按通讯量和非升序加入优先级队列中.

③ 蚂蚁 k 按照优先级队列选择 IP 核, 然后根据 (3) 的选择策略为该核选择映射节点, 将该节点计入路径表中, 并将状态表中该节点删除. 重复操作, 直到所有的 IP 核都映射完毕.

④ 重复执行步骤 ③, 直到所有的蚂蚁都完成映射. 这样一次循环完成, 得到 m 个映射结果.

(5) 局部搜索

一次循环结束后, 利用式 (7) 计算适应度, 选出适应度最大的作为本次循环的最优解, 并对该解采用两交换法 (2-Opt) 进行局部搜索. 两交换法是指将路径表中的任意两点交换得到新解. 首先对本次的最优解做多次两交换, 得到多个新解, 然后比较多个新解与原解的适应度, 找到适应度最大的作为本次循环的最优解. 由于多次交换会使算法时间增长, 因此, 设定交换次数最多为 1 200 次.

(6) 信息素更新

MMAS 在一次循环后, 只允许本次循环中构造的解最优的蚂蚁更新信息素, 其解所包含的各 (i, j) 对上的信息素根据式 (11) 调整:

$$\tau_{i,j} = (t + 1) = \rho + \eta_{i,j}(t) + \Delta\tau \tag{11}$$

$$\Delta\tau = \text{fitness}(\text{best solution}) \tag{12}$$

式 (12) 中, $\text{fitness}(\text{best solution})$ 是指本次最优解的适应度.

同时为了防止搜索出现停滞现象, MMAS 对任意 (i, j) 对规定了信息素的范围 $[\tau_{\min}, \tau_{\max}]$. 但是考

虑到遗传算法对信息素初始化作用, 这里只对最优解的 (i, j) 对的信息素限定了范围, 其中:

$$\tau_{\max} = \frac{1}{1 - \rho} \text{fitness}(\text{best solution}) \tag{13}$$

$$\tau_{\max} = \frac{1}{5} \tau_{\max} \tag{14}$$

4 实验分析

为验证 GA-MMAS 算法的性能, 本文进行了 11 组实验. 表 1 中给出了这 11 组实验 IP 通讯图的具体信息和 2Dmesh 结构 NoC 的大小.

表 1 各组实验具体信息

应用	IP 核数量	有向弧	通讯总量	NoC 尺寸
1	10	12	1 634	4×3
2	12	12	3 466	4×3
3	16	33	609 257	4×4
4	20	24	2 640	5×4
5	20	24	2 640	5×5
6	20	37	4 176	5×4
7	20	37	4 176	5×5
8	25	33	684 271	5×5
9	40	72	80 951	7×6
10	48	95	98 840	7×7
11	96	194	215 274	10×10

表中, IP 核数量为 12 的应用是 MPEG4, IP 核数量为 25 的应用是 H. 263. 其他则是用随机任务流程图生成器 TGFF 产生的.

针对上述的 11 组应用, 分别采用 GA-MMAS, GA(采用精英保留), AS^[4] 和 MMAS^[5] 实现映射, 并比较映射后的通讯能耗. 所有的算法都采用 C++ 编写, 采用 Intel P IV 3. 4 GHz CPU, 1 GB 内存, 在 Windows XP 操作系统下运行.

以初始映射(一一映射)^[5]的结果为标准(该映射下 NoC 的通讯能耗为 1), 图 3 所示为各种算法的求解性能比较.

从图中可以看出, 在不同规模的应用下, 相比初始映射, 各种算法均不同程度地减少了通讯能耗, 且 GA-MMAS 算法性能优于其他算法. GA-MMAS 算法与初始映射相比, 最高可节省 60% 的能耗, 平均节省 45% 左右; 与 GA 相比, 最高可节省 25% 的能耗, 平均节省 20% 左右; 与 AS 比较, 最高可节省 30% 的能耗, 平均节省 20% 左右; 与 MMAS 相比,

最高可节省 30% 的能耗,平均节省 20% 左右。

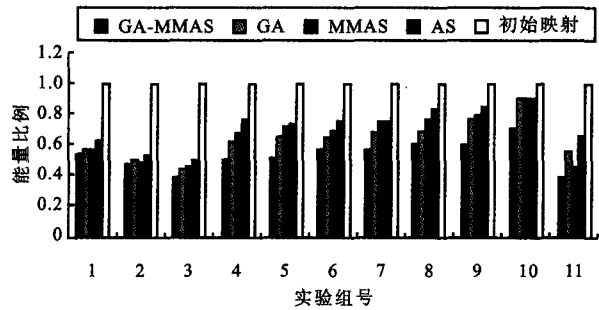


图 3 各种算法性能比较

同时从实验中还可以得出,GA-MMAS 算法比 GA 和 AS 的计算时间长,但与 MMAS 相近. 这是由于虽然算法由 GA 和 MMAS 结合,但 GA 的计算时间很短,算法的时间主要花费在 MMAS 中的局部搜索,因此才会有这样的结果. 对于表 1 中应用,GA-MMAS 算法均可以在合理的时间(十秒到十几分钟内完成),如表 2 所示.

表 2 各种算法的时间比较 s

应用	GA	AS	MMAS	GA-MMAS
1	1.8	1.8	8.5	10
2	2	2	9	11
3	3	5.5	10	12
4	3	5	13	14
5	5	7	20	21
6	7	10	25	27
7	6	8	21	23
8	7	12	30	31
9	18	72	198	168
10	24	132	368	321
11	48	148	1 196	1 205

随着 NoC 结构和 IP 核的增多,算法的计算时间会增加很多. 规模最大的 96 核应用,大约在 20 分钟内完成,这样的时间是可以接受的.

5 结束语

本文提出了一种 GA-MMAS 算法来解决 2Dmesh 结构 NoC 的 IP 核映射问题. 算法采用带有

精英保留策略的 GA 来初始化 MMAS 的信息素,弥补了 MMAS 前期信息素匮乏的缺点,同时去掉了 MMAS 中启发因子,转而在 IP 通讯量和优先级来确定启发信息,从而得到更小的通讯能耗. 为验证算法的性能,进行了多组实验,其中包括 MPEG4 和 H. 263 两个实际应用. 实验结果显示,该算法与初始映射相比,能耗可以节省 36%~60%,与已有的 GA,AS 和 MMAS 相比分别能节省 3%~25%,10%~30%和 3%~30%的能耗. 由此可以看出,GA-MMAS 算法对于解决 2Dmesh 结构 NoC 的低能耗映射问题是非常有效的.

参考文献:

[1] Ogras U Y, Hu Jingcao, Marchlescu R. Key research problems in NoC design: a holistic perspective [C]// Proceedings of the 3rd IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis. Jersey City, 2005: 69-74.

[2] 陈延仓,谢伦国,李晋文. 一种基于遗传算法的片上网络映射算法[J]. 计算机工程与科学,2009,31(A1):14-16.

[3] 葛芬,吴宁. 功耗优化的片上网络协同映射[J]. 应用科学学报,2008,26(6):606-216.

[4] 杨盛光,李丽,高明伦,等. 面向能耗和延时的 NoC 映射[J]. 电子学报,2008,36(5):937-942.

[5] 周干民,尹勇生,胡永华,等. 基于蚁群优化算法的 NoC 映射[J]. 计算机工程与应用,2005(18):7-10.

[6] 欧阳一鸣,张岚,梁华国. 低功耗优先的片上网络映射优化方法[J]. 微电子学与计算机,2010,27(3):54-58.

[7] 万玉鹏,吴宁. 基于自适应算法的 NoC 路由单元的系统级设计[J]. 微电子学与计算机,2010,27(5):42-45.

[8] 武交锋. 应用遗传算法提高蚁群算法性能的研究[D]. 太原:太原理工大学,2007.

作者简介:

穆屹峰 男,(1986-),硕士研究生. 研究方向为电子系统集成与专用集成电路设计.

吴宁 女,(1956-),教授,博士生导师. 研究方向为数字系统理论与技术、电子系统集成与专用集成电路设计.

葛芬 女,(1981-),博士. 研究方向为片上网络设计.