

一种改进的基于粒子群的三维片上网络优化布局算法

宋国治 张大坤 涂 遥 黄 翠 王莲莲

(天津工业大学计算机科学与软件学院 天津 300387)

摘 要 提出了一种改进的基于粒子群算法的优化布局算法(Improved Particle Swarm Optimization, IPSO)来替换原有的基于模拟退火(Simulated Annealing, SA)算法的优化布局算法,使其更加适用于大型三维片上网络的仿真。通过比较这两种算法的基本思想,给出了这两种算法的实现步骤并详细介绍了 IPSO 算法的改进思路。最后利用一款现有的三维片上网络仿真器进行了仿真验证。结果表明,提出的 IPSO 算法比原来的 SA 算法更适用于大型三维片上网络的仿真。

关键词 三维片上网络,布局算法,退火算法,粒子群优化算法

中图分类号 TP301.6 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.7.024

Improved Algorithm for 3D NoC Floorplan Based on Particle Swarm Optimization

SONG Guo-zhi ZHANG Da-kun TU Yao HUANG Cui WANG Lian-lian

(School of Computer Science & Software Engineering, Tianjin Polytechnic University, Tianjin 300387, China)

Abstract We presented an improved particle swarm optimization algorithm based algorithm to optimize the floorplan, called improved particle swarm optimization(IPSO) algorithm, and replaced the original floorplan optimization algorithm based on simulated annealing algorithm (SA algorithm) to make it more suitable for large-scale three-dimensional networks-on-chip simulation. We compared the basic idea of these two algorithms. The details of the implementation steps of the two algorithms and improvement ideas of IPSO algorithm were given. We verified the performance improvement using an existing 3D NoC simulator. The simulation results show that the proposed IPSO algorithm is more suitable for large-scale three-dimensional simulation of 3D NoCs than the original SA algorithm.

Keywords 3D NoCs, Floorplan algorithm, Simulated annealing, Particle swarm optimization

1 引言

随着集成电路工艺的飞速发展和设备的计算复杂度愈来愈高,单芯片上的集成度也会愈来愈大,这样二维 NoC 虽然解决了 SoC 面临的全局同步等一系列问题,但是从根本上依然不可避免关键部件路径无法最短、功耗开销和连线延迟等相关问题。在这种背景下,三维片上网络的概念被提出。三维片上网络因其第三维的引入,与二维片上网络相比较有着先天的几何优势,这也使得 3D NoC 系统在性能上有了显著提升^[1]。3D NoC 因其具有提升系统抗干扰能力,最大程度上缩短关键部件路径长度,降低延时,减小功耗等优势,而使其一经问世便引起了学术界和产业界的强烈关注^[2]。可以预见,3D NoC 将在未来的一段时间内仍将是集成电路领域的研究热点。

2005 年,第一篇有关 3D NoC 的科研论文^[3]才发表出来,说明三维片上网络相对而言是一个全新的研究课题。目前从事三维片上网络研究的研究机构主要包括美国宾夕法尼亚州

立大学、加州大学等。宾夕法尼亚州立大学的 Yuan Xie 等人 在 3D IC 设计及系统架构方面做了不小的贡献^[4],他们主要研究多维映射和布局布线的影响等方面。美国加州大学的 Nandakumar Vivek S 教授等人提出了多层 3D NoC 架构^[5],在 3D 片上网络架构的设计,以及路由算法的设计与实现方面做了不少工作。Srinivasan Murali 和 Giovanni De Micheli 的主要研究方向在于 3D NoC 架构下的路由算法。他们设计出一种命名为 NMAP 的算法,该 NMAP 算法提出了一种单个路径路由和分段运输路由选择。此外,有 6 个视频处理的实验应用表明,比起现有的路由算法,NMAP 算法显著地节省了带宽和通信成本^[6]。Walid Lafi 等人 在其研究论文^[7]中提出了一种新的路由器,相比其他经典的 3D mesh 结构下的路由器,在吞吐量和延迟方面它都有着不错的收益,尤其是在比较大的 mesh 架构下。他们提出的路由器是由两个完全分离的模块组成,一个用于层间通信,一个用于层内通信,从而很好地增强了 3D NoC 的通信性能。K. Somasundaram 在研究网络拓扑和路由技术上做出了很大贡献,他在一篇研究论

到稿日期:2014-07-01 返修日期:2014-10-20 本文受国家自然科学基金:基于柏拉图立体多级裂变模型的 3D NoC 拓扑结构的研究(61272006)资助。

宋国治(1977—),男,博士,副教授,主要研究领域为三维片上网络及复杂系统与结构数值仿真计算,E-mail: guozhi.song@googlemail.com;张大坤(1960—),女,博士,教授,主要研究领域为网络拓扑结构,E-mail: zhangdakun2002@163.com;涂 遥(1992—),男,硕士生,主要研究方向为三维片上网络布图规划,E-mail: 372149290@qq.com;黄 翠(1991—),女,硕士生,主要研究方向为三维片上网络映射,E-mail: 1220093354@qq.com;王莲莲(1989—),女,硕士,主要研究方向为三维片上网络,E-mail: 584870224@qq.com。

文中^[8]设计了一种 NS-29 的仿真环境和两个三维网络拓扑结构(GBT 和 CBT),还提出了一种相应的基于集群的路由算法。通过仿真实验及分析丢包率、能源损耗和分组延时表面发现,作者提出的两种架构比起现有的经典架构在性能上有显著的改善。

从 2003 年开始,国内陆续开始关注片上网络的研究,而对三维片上网络的研究起步就更晚,且很大一部分研究还处于初步阶段,研究成果与国外相比较而言还是有比较大的差距的。目前国内大部分研究三维片上网络的方向主要是在拓扑结构和路由技术这两个方面。合肥工业大学的研究员董少周在其硕士论文中,针对 Turn Model 模型设计了无死锁的确定性的路由算法^[9]。实验表明,该算法由于避免了以前经典的 XY 路由算法会在一个相同的方向上传输数据时出现堵塞的问题,有效地提升了通信的吞吐量。电子科技大学的博士生陈亦欧主要针对片上网络拓扑结构和通信方法两个方向上的研究,总结了原来经典的几种 3D NoC 拓扑结构^[10],而且还提出了 3 种适用于 3D NoC 上的拓扑结构,并针对每种架构设计出与之对应的路由算法。

本文针对一款现有的三维片上网络仿真的布局算法(模拟退火算法)进行了详细的分析后,提出了一种改进的基于粒子群优化算法的优化布局算法,并用该仿真器对两个算法进行了仿真验证,证明本文提出的 IPSO 算法更适用于大型三维片上网络的仿真。

本文第 2 节讨论了本文所采用的仿真器原有的优化布局的算法(模拟退火算法),然后提出了一种改进的基于粒子群优化算法优化布局的算法,并对该算法的改进思路进行了详细的说明;第 3 节通过使用该仿真器对两个算法进行了多次仿真验证,对仿真结果的分析证明了本文提出了 IPSO 算法更适用于大型三维片上网络的仿真;最后总结全文。

2 3D NoC 布局优化算法

本文研究所采用的是一款目前主流的片上网络的仿真平台 vnoc3,它是由美国 North Dakota State 大学的 Cristinel Ababei 开发的。该仿真平台采用了模拟退火算法作为它的优化布局的算法^[11]。本文对此模拟退火算法进行了详细的分析后,提出了一种改进的基于粒子群优化算法优化布局的算法,并通过仿真实验对两个算法进行了比较。

2.1 模拟退火算法

要想明白模拟退火算法的原理就要先弄清楚物理退火过程。物理上退火的意思就是说当把固体加热升温到足够高的温度下,固体内部所有的分子在状态空间里自由运动,然后逐步降温使其冷却,分子的运动也会逐渐趋于有序,最后分子会以低能状态排列,固体最终达到一种稳定的状态。这种热处理过程就是所谓的退火。

模拟退火(Simulated Annealing, SA)算法的思想就是通过模仿物理退火过程,使其与一般优化问题结合。SA 算法是一种普遍适用的随机搜索算法,是拓展化的局部搜索算法。在介绍 SA 算法思想之前,首先要知道 Metropolis 准则。它是在 1953 年由 Metropolis 提出的一种重要性采样法,也就是以概率来接受新的状态^[12]。SA 算法的基本思想是以给出的一个初始高温为开始,采用包含概率突跳特性的 Metropolis

准则的抽样策略在解空间内随机地搜索,随着温度的逐渐下降重复进行抽样过程,最终会得出问题的全局最优解。由此可以看出 SA 算法是种启蒙式的随机寻找最优解的算法。

模拟退火算法的具体步骤如下:

1) 给出一个模型的每一个参数的变化范围,然后在这个范围内选一个初始解 m_0 ,且计算其目标函数值 $E(m_0)$ 。

2) 随机产生一个领域解 m ,计算其相应的目标函数值 $E(m)$,算出 $\Delta E = E(m) - E(m_0)$ 。

3) 如果 $\Delta E < 0$,则新的解被接受,如果 $\Delta E > 0$,则新解按概率 $P = \exp(-\Delta E/T)$ 来进行接受, t 为温度。当新解被接受时,重置 $m_0 = m, E(m) = E(m_0)$ 。

4) 在温度 t 下,若达到热平衡则转向步骤 5),否则重复步骤 2)、3)。

5) 降低温度。如果温度达到给定的最低温度则算法停止,不然转向步骤 2)。

2.2 改进的粒子群优化算法

粒子群优化(Particle Swarm Optimization, PSO)算法将优化问题的每一个潜在的解想象成 D 维搜索空间的一个粒子(Particle),每一个粒子有一个适应值(Fitness Value),这个值是由目标函数唯一确定的。这些粒子在 D 维搜索空间中飞行。每个粒子有一个速度值来决定它们的位置(状态,也就是解)。每个粒子在搜索过程中会考虑两个方面:1) 自己搜索到的历史最优值;2) 群体内或者领域内的其它粒子的历史最优值。用这两方面的数据来进行自身位置(也就是解)的变化。而 PSO 算法搜索的关键在于粒子速度的更新。基本粒子群优化算法中,粒子的速度更新主要受 3 方面影响:(1) 前一次迭代的自身速度;(2) 粒子会考虑自身的经验,往曾经搜索到的最好点靠近,称为自我认知的部分;(3) 粒子向群体学习,考虑到领域内其他粒子搜索到的最好点,往所有粒子曾经搜索到的最好点靠近,称为社会经验部分。粒子位置更新方式可以用图 1 来描述。

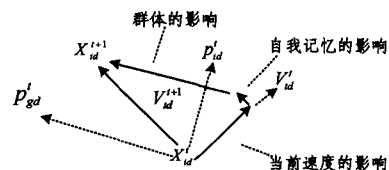


图 1 粒子速度更新方式

在基本粒子群优化算法中,种群 $X = \{x_1, x_2, \dots, x_i, \dots, x_m\}$ 由解空间里的 m 个粒子组成。第 i 个粒子的速度和位置分别表示为 $v_i = (v_{i1}, v_{i2}, \dots, v_{in})$, $x_i = (x_{i1}, x_{i2}, \dots, x_{in})$ 。

第 i 个粒子搜索过程中经过的历史最优值为 $P_g = (p_{g1}, p_{g2}, \dots, p_{gm})$, 种群或领域内的所有粒子经过的全局历史最优值为 $P_g = (p_{g1}, p_{g2}, \dots, p_{gm})$ 。在迭代过程中粒子的速度更新公式和位置更新公式分别为:

$$v_{id}^{t+1} = w * v_{id}^t + c_1 r_1 (p_{id}^t - x_{id}^t) + c_2 r_2 (p_{gd}^t - x_{id}^t) \quad (1)$$

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1} \quad (2)$$

式(1)中 w 为惯性权重,惯性权重的大小决定了粒子对此刻速度继承的多少,惯性权重的合理取值可以决定粒子的探索能力和开发能力的均衡。 c_1, c_2 为学习因子或加速系数,学习因子的加入使粒子具有总结自身和向群体或领域内优秀粒子学习的能力,学习因子的合理取值可以使粒子能够往自身历

史最好点和群体内的全局历史最好点靠拢。

本文提出一种改进的粒子群优化算法(以下简称 IPSO 算法),使其能更加适用于大型架构的优化布局。具体的改进策略是:在学习因子方面,本算法采用学习因子异步时变的方式。这是一种由 Ratnaweera 等提出的一种时变的学习因子设置方法^[12]。之所以称为异步时变,是因为学习因子在优化的过程中会随着时间的不同而变化,其目的在于在优化的初期阶段全面增强全局搜索,而在后期阶段更趋向于局部寻优,使粒子不会陷入局部最优解而无法逃脱。通过异步时变可以让粒子在搜索初期趋向于探索整个搜索空间,而不会很快陷入局部最优,在优化后期,可以使粒子趋向于局部寻优,而不会长时间找不到全局最优解。可通过随着时间的增加逐渐减小自我学习的因子 c_1 ,并逐渐增加社会学习因子 c_2 来实现这种想法。 c_1 和 c_2 随时间增加的更新公式如下:

$$c_1 = (c_{1f} - c_{2f}) \frac{iter}{Iter_max} + c_{1i} \quad (3)$$

$$c_2 = (c_{2f} - c_{2i}) \frac{iter}{Iter_max} + c_{2i} \quad (4)$$

其中, $c_{1i} = 2.5$, $c_{1f} = 0.5$, $c_{2i} = 0.5$, $c_{2f} = 2.5$ 。 $Iter_max$ = 最大迭代数。 $iter$ = 当前迭代数。

在权重 w 方面,在 PSO 算法速度更新公式中,权重 w 对寻优的结果有重要的作用。数值大的 w 更趋向于全局搜索,算法收敛速度快。数值较小的 w 趋向于局部搜索,有利于局部寻优,算法收敛速度慢。所以本文决定通过判断该粒子目前是适合全局搜索还是局部搜索来将粒子分类^[13]。分类的方法在于,先收集出本轮迭代所有粒子的适应值,计算出本轮迭代单个粒子适应值的平均值 f_{avg} ,将每个粒子的适应值 f_i 与平均值比较大小而分为两类。

(1)当 $f_i \geq f_{avg}$ 时,认为这个粒子在群体中为较优的粒子,应减小权重,充分发挥它的局部寻优的能力。此时权重更新公式为:

$$w = w_{max} - (w_{max} - w_{min}) * i / Iter_max \quad (5)$$

其中, i 为当前迭代次数, $Iter_max$ 为最大迭代次数, w_{max} 和 w_{min} 为最大权重和最小权重,具体值根据实际的问题来确定,在本文中取最大权重为 3,最小权重为 0.5。

(2)当 $f_i < f_{avg}$ 时,认为这个粒子在群体中为较差的粒子,应增大权重,充分发挥它的全局搜索的能力。此时,权重更新公式为:

$$w = 1.5 - 1 / (1 + k_1 * \exp(k_2)) \quad (6)$$

其中, k_1 是用来调整 w 的上限, k_2 用来平衡公式在全局寻优和局部寻优上的能力。在本文中的 k_1 和 k_2 的值分别取 1.5 和 1。

IPSO 算法具体步骤:

1)初始化一群粒子,包括随机的位置 X_0 和随机速度 V_0 ,范围均在 $[-3, 3]$ 之间,群体规模设为 shu 。

2)把最大迭代数定义为 N , $N = times * fp_p - > size()$ 。定义当前迭代次数为 $iter$,定义粒子本身经过的最优位置为 pre_cost ,种群内所有粒子经过的最优位置定义为 $best$ 。

3)判断 $iter$ 是否大于等于 N ,如果是则执行 5),否则执行 3)。

①计算粒子的适应度,适应度公式为:

$$Cost\ function = \alpha * Area + (1 - \alpha) * WireLength \quad (7)$$

这个公式是计算布置一个布局所需的成本即 $Cost\ function$ 。 α 是个常数,本文中取 0.25,它主要用于平衡全部线长和单元片占的面积。

②当 $pre_cost \geq best$ 时,把 pre_cost 的值赋给 $best$ 。当 $pre_cost < best$,统计这样的粒子数 $badnum$ 为多少。

③用式(3)和式(4)计算此次迭代中的学习因子,通过比较每个粒子的适应值与平均值的大小来决定使用式(5)或式(6)计算权重,用式(1)和式(2)来更新粒子的速度和位置。

4) $iter = iter - 1$,执行步骤(3)。

5)将 $best$ 输出,结束。

3 仿真实验与结果分析

本文实验使用了 6 个测试用例^[11],它们的特性如表 1 所列。在这个表中,列出了直接拓扑结构的大小。

表 1 6 个测试用例的特性表

测试用例	IP 核数目	核的纵横比的平均值	核的纵横比的标准差	直接拓扑 R×R
apte	8	4324/2499	27/4	3×3
xerox	10	2114/2872	335/1290	4×4
hp	11	4533/924	2498/386	4×4
ami25	25	1770/1408	1201/896	5×5
ami33	33	1581/1573	830/865	6×6
ami49	49	1089/1123	768/651	7×7

Cristinel Ababei 是从典型的 MCNC 测试用例构造这些测试用例,MCNC 测试用例的面积按一定比例缩小到平均约 1cm×1cm,这也是文献[14]中提到的 NoC 的典型面积。模块之间的初始连接被用来计算与每个测试用例布局相关联的通信任务图之间的通信量。因为在 $vnoc3$ 提供的仿真方法中,网络路由之间的物理链路的长度随网络大小不同而不同,Cristinel Ababei 通过使用文献[15]中一个简单的埃尔莫尔伦公式推算物理链路延迟来估计链路延迟。同样的延迟估计技术被应用在 IP 核和路由之间的额外链路上,它被假设为 L 形(通过金属层之间的延迟可以忽略不计)。

通过观察这 6 个测试用例的 IP 核数目和直接拓扑结构的大小发现,apte, xerox 和 hp 这 3 个测试用例相比 ami25、ami33、ami49 来说架构偏小,所以本文决定将 apte, xerox 和 hp 分为一组,用来代表架构偏小的三维网络拓扑结构;而 ami 系列分为另一组,代表架构偏大的三维网络拓扑结构。通过比较两种算法下的仿真器 $vnoc3$ 在用同样的命令进行测试后所得结果中的 CPU 运行时间(CPU processor time(s))、平均微片延迟(average flit latency(s))和吞吐量(throughput(kb/s))这 3 方面性能来判断两种算法的优劣。显然对于同样一条命令,相比较来看 CPU 运行时间较短、延迟较低、吞吐量较高的就代表其性能更优。由于篇幅原因,本文只选择注入负载为 100%时两种算法的比较图作为代表。在 CPU 运行时间方面,因为两种算法在注入负载为 50%和 100%两种情况下的变化趋势基本一致,所以本文只选择注入负载为 100%时两种算法运行时间的比较图作为代表,如图 2 所示。

从图 2 可以看出,在拓扑结构较小的测试用例如 apte、xerox 中,算法 IPSO 和 SA 在运行时间上基本上保持一致,但当测试用例的拓扑规模大于 4×4 的时候,也即从图中的 hp 测试用例开始,本文提出的改进了的 IPSO 算法在 CPU 的运

行时间上要明显的少于原 SA 算法。

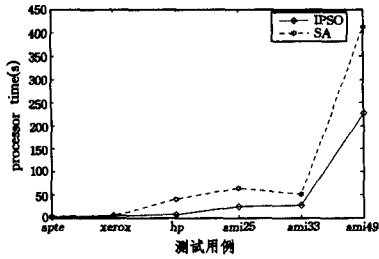


图 2 满负载的情况下两种算法的运行时间变化

从图 2 中曲线的趋势也可以看出,当拓扑结构越大,IP-PSO 的这一优势就越明显。因此可以得出结论,就运行时间这一性能来说,改进了的 IP-PSO 算法是要优于原来的 SA 算法。

通过观察图 3 发现,注入负载为 50% 的时候,虽然在拓扑结构较小时两种算法平均延迟相差不多,增长趋势也基本一致,但是可以观察到,当随着测试用例的拓扑架构开始偏大时,IP-PSO 算法的延迟增长率要小于原 SA 算法的。

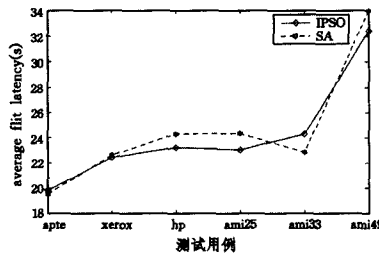


图 3 注入负载为 50% 情况下两种算法延迟变化

图 4 中,满负载的情况下,在 Mesh 尺寸较小的测试用例中,就增长趋势来说,IP-PSO 算法的延迟增长率不稳定,而原 SA 算法延迟增长率缓慢且较稳定,所以 Mesh 尺寸较小时 SA 算法的性能更优。但是当拓扑结构大于 6×6 后,原 SA 算法的延迟增长率增大且明显高于 IP-PSO 算法的增长率。

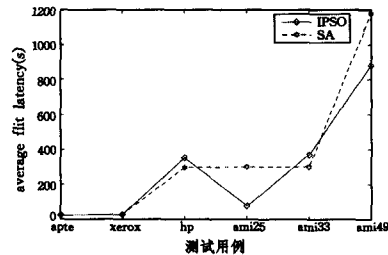


图 4 满负载情况下两种算法延迟变化

因此可以得出结论,就延迟方面来看本文提出的 IP-PSO 算法比起原 SA 算法要更适用于规模较大的网络架构。

负载为 50% 和 100% 情况下两种算法的吞吐量变化如图 5、图 6 所示。

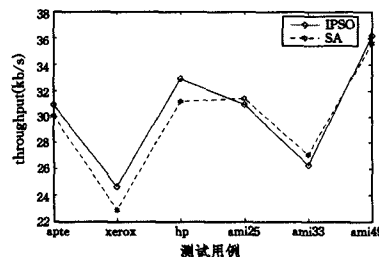


图 5 负载为 50% 情况下的两种算法的吞吐量的变化

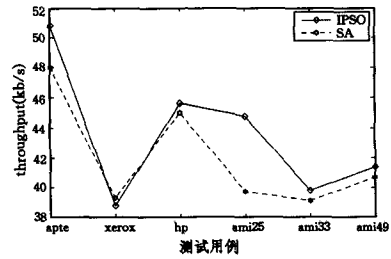


图 6 满负载情况下的两种算法的吞吐量的变化

从图 5 和图 6 可以看出,注入负载为 50% 和满负载的两种情况下的两种算法的吞吐量都没有一种稳定的变化趋势,而是上下起伏地变化着。

从表 1 中 6 个测试用例的特性可以看出,吞吐量的变化会受到多个属性值的影响,如 IP 核数目的变化、核的纵横比的变化等。

由图 5 和图 6 可得出结论,从整体效果上来看,IP-PSO 算法下的吞吐量要高于 SA 算法。

结束语 本文提出了一种改进的适用于规模较大的三维片上网络架构的 IP-PSO 算法,且详细介绍了该算法的改进之处。本文针对 IP-PSO 算法和原 SA 算法进行了一系列的仿真实验,从实验结果分析中发现,IP-PSO 算法在测试用例规模较大时的性能比较中总是明显优于原 SA 算法的,而规模较小时两算法的优劣性不明显,说明本文提出的 IP-PSO 算法比起原来的 SA 算法更适用于布局规模较大的网络拓扑结构。

参考文献

- [1] Wang Jia-wen. Research of Key Issues on Three Dementional Network on Chip[D]. Nanjing: Nanjing University, 2012
- [2] Jeang Y L, Wey T H, Wang H Y, et al. Mesh-Tree Architecture for Network-on-chip Design[C]// International Conference on Innovative Computing Information and Control(ICICIC). 2007; 1-4
- [3] Kangmin L, Se-Joong L, Donghyun K, et al. Networks-on-chip and networks-in-package for high-performance SoC platforms [C]// 2005 IEEE Asian Solid-State Circuits Conference. 2005; 485-488
- [4] Ouyang Jin, Xie Jing, Matthew P, et al. Evaluation of using inductive/capacitive-coupling vertical interconnects in 3D network-on-chip[C]// IEEE/ACM International Conference on Computer-Aided Design(ICCAD). 2010; 477-482
- [5] Nandakumar Vivek S, Marek-Sadowska M. Low power, high throughput network-on-chip fabric for 3D multicore processors [C]// IEEE International Conference on Computer Design(IC-CD). 2011; 453-454
- [6] Murali S, De Micheli G, et al. Bandwidth-Constrained Mapping of Cores onto NoC Architectures[C]// Proc. DATE'04. 2004; 896-901
- [7] Lafi W, Lattard D, Jerraya A, et al. An efficient hierarchical router for large 3D NoCs[C]// 2010 21st IEEE International Symposium on Rapid System Prototyping(RSP). 2010; 1-5
- [8] Somasundaram K, Viswanathan N, et al. Performance Analysis of Cluster based 3D Routing Algorithms for NoC[C]// 2011 IEEE Recent Advances in Intelligent Computational Systems (RAICS). 2011; 157-162

不同参数下运用本文提出的基于截断边扩展图近似算法得出的网络可靠度近似值;“BDDSize1”栏列出了算法 1 生成的 BDD 尺度;“BDDSize2”栏列出了算法 2 生成的 BDD 尺度。“误差”栏列出本文所提算法与 Kuo 算法关于网络可靠度的相对误差。

由表 1 可得:

1)对于大型网络,本算法效果更好。比如表 1 中的网络 5,BDD 尺度减小 80.1%,并且误差在 5%之内。

2)对于同一网络,需不断地调整 $Length$ 的值,直至选定一个合适值,从而得到一个合适的近似值,使得在允许性能条件下误差最小。比如表 1 中网络 2, $\{s,t\}$ 为 $\{0,8\}$ 时,在误差允许范围内,当设置 $Length$ 的值为 6 时,BDD 尺度减小 13%,而当设置 $Length$ 的值为 4 时,BDD 尺度减小 34.7%。

3)对于同一网络,要选择合适的 $\{s,t\}$,比如网络 3,在误差允许范围内,当 $\{s,t\}$ 为 $\{0,15\}$ 时,BDD 尺度减小 25.5%,而 $\{s,t\}$ 为 $\{1,10\}$ 时,BDD 尺度减小 62.6%。

表 1 可靠度分析

网络	$\{s,t\}$	算法 1	BDDSize1	Length	算法 2	BDDSize2	误差(%)
网络 1	$\{0,3\}$	0.97848	8	2	0.96389	6	1.5
网络 2	$\{0,8\}$	0.97250	46	4	0.96992	30	0.3
	$\{0,8\}$	0.97250	46	6	0.97183	40	0.06
	$\{1,6\}$	0.98216	39	4	0.95287	11	3.1
	$\{2,7\}$	0.98216	41	4	0.95287	11	3.1
网络 3	$\{0,15\}$	0.97504	247	6	0.96556	184	1.0
	$\{1,10\}$	0.99686	225	5	0.99270	84	0.4
	$\{2,5\}$	0.99709	273	6	0.99508	63	0.2
	$\{3,11\}$	0.98540	141	6	0.98190	51	0.3
网络 4	$\{0,5\}$	0.99997	141	2	0.99986	46	0.01
	$\{0,1\}$	0.99997	147	2	0.99986	31	0.01
	$\{0,3\}$	0.99997	147	2	0.99986	34	0.01
网络 5	$\{0,9\}$	0.99999	46256	3	0.99999	9860	≈ 0
	$\{0,3\}$	0.99999	49724	3	0.99999	9860	≈ 0
	$\{0,7\}$	0.99999	48248	3	0.99999	9872	≈ 0

结束语 本文提出基于截断边扩展图的网络可靠度近似分析算法。利用该近似算法可以快速计算出大型网络的可靠度,减少了计算递归网络的次数,其计算量要比精确算法小得多,从而缩短了计算二端网络可靠度的时间,为进一步研究大规模网络的可靠度提供依据,这也较为直观地表明该近似算法具有很好的优越性。

参 考 文 献

[1] Xing L D. An efficient binary-decision-diagram-based approach for network reliability and sensitivity analysis [J]. IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans, 2008, 38(6): 105-115

[2] Rai S, Aggarwal K K. An efficient method for reliability evalua-

tion of a general network [J]. IEEE Transactions on Reliability, 1978, 27(3): 206-211

[3] Medhi D. A unified approach to network suevivability for tele-traffic networks: models, algorithms and analysis [J]. IEEE Transactions on Communications, 1994, 42(2/3/4): 534-548

[4] Doverspike R. Trends in layered network management of ATM, SONET, and WDM technologies for network survivability and fault management[J]. Journal of Network and Systems Management, 1997, 5(2): 215-220

[5] Hui K P. Monte carlo networks reliability ranking estimation [J]. IEEE Transactions on Reliability, 2007, 56(1): 50-57

[6] Shier D R. Network reliability and algebraic structures[M]. Oxford New York: Clarendon Press, 1991

[7] Lin Y K. Spare routing reliability for a stochastic flow network through two minimal paths under budget constraint [J]. IEEE Transactions on Reliability, 2010, 59(1): 2-10

[8] Yeh W C. A simple minimal path method for estimating the weighted multi-commodity multistate unreliable networks reliability [J]. Reliability Engineering and System Safety, 2008, 93(1): 125-136

[9] Jane C C, Lai Y W. A dynamic bounding algorithm for approximating multi-state two-terminal reliability [J]. European Journal of Operational Research, 2010, 205(3): 625-637

[10] Shooman A M. Algorithms for network reliability and connection availability analysis [C]// Electro/95 International Professional Program Proceedings. Boston, USA, 1995: 309-333

[11] Resnde L I P. Implementation of a factoring algorithm for reliability evaluation of undirected networks [J]. IEEE Transactions on Reliability, 1988, 37(5): 462-468

[12] Kuo S Y, Lu S K, Yeh F M. Determining terminal-pair network reliability based on edge expansion diagrams using OBDD [J]. IEEE Transaction on Reliability, 1999, 48(3): 234-246

[13] Bryant R E. Symbolic Boolean Manipulation with Ordered Binary-Decision Diagrams [J]. ACM Computing Surveys, 1992, 24(3): 293-318

[14] Friedman S J, Supowit K J. Finding the Optimal Variable Ordering for Binary Decision Diagrams [J]. IEEE Transactions on Computers, 1990, 39(5): 710-713

[15] Yeh W C, Lin Y C, Chung Y Y. Performance analysis of cellular automata Monte Carlo simulation for estimating network reliability [J]. Expert Systems with Applications, 2010, 37(5): 3537-3544

[16] Doguc O, Ramirez-Marquez J E. A generic method for estimating system reliability using Bayesian networks [J]. Reliability Engineering and System Safety, 2009, 94(2): 542-550

(上接第 117 页)

[9] Dong Shao-zhou. The Desigh and Research of Routing algorithm and Simulation Model on Network on Chips, master dissertation [D]. Hefei: Hefei University of Technology, 2009

[10] Chen Yi-ou. Research on the network on chip architecture of oriented real-time complex systems and the mapping of technology [D]. Chengdu: University of Electronic Science and Technology of China, 2012

[11] Ababei C, de Paulo V, et al. 3D Network-on-Chip Architectures Using Homogeneous Meshes and Heterogeneous Floorplans

[J]. International Journal of Reconfigurable Computing, 2010

[12] Wang Ding-wei, Wang Jun-wei, Wang Hong-feng. Intelligent Optimization Methods[M]. Higher Education Press, 2007

[13] Yu Gang. The improved PSO algorithm and its application[M]. Chengdu: Chengdu University of Technology

[14] Mineo C, Jenkal R, Melamed S, et al. Interdie signaling in three dimensional integrated circuits[C]// Proceedings of IEEE Custom Integrated Circuits Conference (CICC '08). 2008: 655-658

[15] Rabaey J M. Digital Integrated Circuits: A Design Perspective (2nd edition) [M]. Upper Saddle River, NJ, USA: Prentice-Hall, 2003