

SPN 中网状路径研究和性能解算

黄光球, 龚叶蓁

(西安建筑科技大学 管理学院, 陕西 西安 710055)

摘 要:以往的关于 SPN 中路径寻找的研究都基于 SPN 中仅存在链状路径的假设,事实上由于变迁之间复杂的逻辑关系 SPN 中存在着复杂的网状路径。关于随机 Petri 网中网状路径还未有深入研究。文中把链状路径看作网状路径的一个特例,给出了 SPN 中网状路径的定义和网状路径的寻找和路径性能解算的方法。大幅改进了 ACO 和 SPN,使变迁和蚂蚁都具有选择和记忆能力,增强了 ACO 的灵活性,使之成功找出网状路径;结合可靠性相关理论给出了变迁可靠性和效率服从指数分布的复杂路径性能和系统综合性能的求解的方法。该方法突破了模拟方法应用于大规模复杂系统的瓶颈,为模拟方法的广泛应用铺平了道路。

关键词:网状路径;随机 Petri 网;可靠性;ACO;大规模系统建模;计算机仿真

中图分类号:TP391.9

文献标识码:A

文章编号:1673-629X(2009)09-0058-04

Network Route Search and Capacity Analysis Method for Complex SPN

HUANG Guang-qi, GONG Ye-zhen

(School of Management, Xi'an University of Architecture & Technology, Xi'an 710055, China)

Abstract: In the past the researches about the SPN route search are based on the assumptions of chain route. In fact, due to the complex relationship between transitions in SPN there are network routes in SPN. The research about the route search and capacity calculating is still in the bland stage. The chain route was handled as a special case and the network definition and analysis method was put forward for the first time. Significantly improved the ACO and SPN, to make transitions and ants have the ability to choose and remember something, greatly enhance the flexibility of the ACO, to make it a successfully to find the network route; the combination of reliability theory gives the method to work out the route capacity and system capacity. This method fill the gaps in this field of research, and to bypass the state space explosion problem; with strong robust, suitable for a variety of complex systems modeling analysis.

Key words: network route; stochastic Petri net; reliability; ACO; large-scale modeling; computer simulation

0 引言

随机 Petri 网(Stochastic Petri Nets, SPN)^[1]是描述和分析离散不确定动态系统的有效建模工具。随着系统复杂程度不断提高,系统的 SPN 模型越来越庞大,对大规模复杂 SPN 模型性能的快速求解已成为系统性能分析无法回避的问题^[2]。SPN 的性能求解方法有两类:(1)以 Molloy^[3]为代表的解方程组的方法来求解模型的性能指标;(2)以文献[4]为代表的使用模拟技术来分析 SPN 模型的方法。模拟方法适合于解算大规模系统的性能指标。随着模型复杂度的提高,出现了复杂的网状路径。系统网状路径的定义和研究

成为模拟方法应用于大规模复杂系统的瓶颈。

目前在模拟方法的路径寻找方向引入的是蚁群算法,当前的研究将 SPN 中的路径均假设为简单的链状路径,没有针对 SPN 有向二部图的特殊性质给出网状路径的定义和分析方法。文献[4]虽然进一步将 Petri 网与蚁群算法相结合,根据蚁群算法和时间 Petri 网的特点提出了一种网络元素可以记录少量信息的记忆扩展时间 Petri 网(ETPN),但此文没有很好解决变迁多输入输出的问题。此外,值得借鉴的还有文献[5],该文探讨了智能 Petri 网实现最短路径问题,但文中只论述了智能的 Petri 网的最短路径问题,没有对更加复杂的 Petri 网类型进行讨论。

文中把链状路径看作网状路径的一个特例,首次给出了 SPN 中网状路径的定义和网状路径的寻找和路径性能解算的方法。针对上述问题给出多输入输出变迁的处理方法并采用正反馈的强化机制,提高了算法的稳定性和效率。

收稿日期:2008-12-27;修回日期:2009-03-14

基金项目:陕西自然科学基金项目(2007E217),陕西省教育专项基金项目(07JK076)

作者简介:黄光球(1964-),男,湖南桃源人,教授,博士,研究方向为计算智能、计算机仿真、系统工程。

1 LS-SPN模型和网状路径定义

Petri网以其优秀的描述异步并发的能力获得了长足的发展^[6]。文中构建一种扩展的SPN模型,为了标识这种特殊模型,称它为LS-SPN(Large-scale System SPN)。LS-SPN描述如下:

定义1 设LS-SPN= $\{P, T, Fl, W, \lambda, K, F_r, F_s, A\}$,其中, $P = \{p_1, p_2, \dots, p_n\}$ 是库所的有限集合, n 为库所的个数,且 $n > 0$; $T = \{t_1, t_2, \dots, t_m\}$ 是变迁的有限集合, m 表示变迁的个数,且 $m > 0$; $P \cup T = \emptyset$; A 是运行在SPN中的蚂蚁集合, $A = \{a_1, a_2, \dots, a_l\}$, l 表示蚂蚁数量,蚂蚁被看作是一种特殊的托肯; $Fl \subseteq (P \cdot T) \cup (T \cdot P)$ (\cdot 为笛卡儿积); $\text{dom}(Fl) \cup \text{cod}(Fl) = P \cup T$,其中 $\text{dom}(Fl) = \{x \mid \exists y: (x, y) \in Fl\}$, $\text{cod}(Fl) = \{y \mid \exists x: (x, y) \in Fl\}$ 即它们分别是 Fl 的定义域和值域; $W: Fl \rightarrow N$ 称为 N 上的权函数,对 $(x, y) \in Fl$, $W(x, y) = W((x, y))$ 称为 (x, y) 上的权; $\lambda = \{\lambda_1, \lambda_2, \dots, \lambda_m\}$ 与 T 中元素一一对应,当对系统的延时建模时 λ 为各个变迁的实施率,当对系统的可靠性建模时 λ 表示各变迁的失效率, λ 集合中的数字为事先测定的,在SPN的运行过程中不变; $K = \{k_1, k_2, \dots, k_n\}$ 定义为 $P \rightarrow N \cup \{\infty\}$ 为 P 上的容量函数。 F_r 用来计算系统中各条路径的性能; F_s 用来计算系统的总体性能。

定义2 当蚂蚁在Petri中运行时,路径是源库所到目的库所间的一个子网,称为网状路径 R , $R = \{r_1, r_2, \dots, r_q\}$, q 为系统中的网状路径数。它是蚂蚁为到达目的库所而触发的所有变迁和所经过的所有库所将构成的一个子网。

定义3 当蚂蚁按照算法运行规则到达目的库所时,所携带的一份路径序列称为部分路径,用 PR 表示。 $PR = \{pr_1, pr_2, \dots, pr_q\}$, q 为系统中的网状路径数,部分路径是网状路径的真子集。

定义4 一条路径上所有部件构成的子系统的总体性能称为路径性能。路径性能用 C 表示, $C = \{c_1, c_2, \dots, c_q\}$, q 为系统中的网状路径数。

定义5 与变迁相关的一组路径序列称为变迁最优路径。如果 t_j 有 k 个输入库所,那么 t_j 就有 k 个最优路径。变迁最优路径 $(t_j, \text{BestRoutes}_i, 1 \leq i \leq k)$ 保存从源库所到 t_j 的输入库所 i 的最高性能路径。

2 蚂蚁运行规则和网状路径寻找

蚁群在LS-SPN中的运行规则将不同于其在网路或无向图中的运行规则。为了寻找LS-SPN中的网状路径,文中给出扩展ACO^[7]算法LS-ACO。LS

-ACO用于分析LS-SPN时蚁群的运行规则如下。

2.1 蚂蚁前进的概率计算(规则1)

蚂蚁能够感知周围环境中的信息素,并根据信息素浓度决定是前进还是随机走一步^[8]。 a_k, Place 为蚂蚁所处的库所; $t_j \in a_k, \text{Place} \cdot, \rho_j = t_j, \text{Pheromone}$ ($1 \leq j \leq r, r$ 是 $a_k, \text{Place} \cdot$ 中变迁的数目)是 t_j 上的信息素浓度。当 $a_k, \text{Place} \cdot \in \emptyset$ 时, $r = 0$,此时蚂蚁只能停留在 a_k, Place 中。

蚂蚁前进的概率为

$$P_m = \frac{1}{\alpha} [\delta + \tanh(\frac{\rho_1 + \rho_2 + \dots + \rho_r}{\beta} - \chi)] = \frac{1}{\alpha} [\delta + \tanh(\frac{\rho_{\text{all}}}{\beta} - \chi)] (\rho_{\text{all}} = \rho_1 + \rho_2 + \dots + \rho_r) \quad (1)$$

蚂蚁随机走一步的概率为

$$P_n = 1 - P_m$$

式(1)中 $\alpha, \beta, \chi, \delta$ 为参数。

2.2 蚂蚁选择前进方向(规则2)

当蚂蚁决定前进时,它根据绝对和相对的信息素积累量从 $a_k, \text{Place} \cdot$ 中选择一个信息素浓度高的变迁作为前进方向。 $t_j \in a_k, \text{Place} \cdot, \rho_j = t_j, \text{Pheromone}$ ($1 \leq j \leq r, r$ 是 $a_k, \text{Place} \cdot$ 中变迁的数目)是 t_j 上的信息素浓度。蚂蚁选择变迁 t_j 的概率是:

$$P_j = (\phi + \rho_j)^\eta / \sum_{1 \leq i \leq r} (\phi + \rho_i)^\eta$$

其中 ϕ 和 η 是控制信息素影响程度的参数。变迁中的信息素浓度越高,蚂蚁选择该变迁的可能性就越大。

2.3 判断变迁是否可被触发(规则3)

蚂蚁根据SPN变迁的触发规则判定由规则2选择的变迁是否可被触发。假设 t_j 是被选中的变迁, $p_k \in \cdot t_j, p_l \in t_j \cdot$ ($1 \leq k \leq m, m$ 是 $\cdot t_j$ 中库所的数目, $1 \leq l \leq h, h$ 是 $t_j \cdot$ 中库所的数目),如果:(1) p_k 中蚂蚁数目大于 $W(p_k, t_j)$, (2) $t_j \cdot \in \emptyset$, (3) p_l 中蚂蚁数量与 $W(t_j, p_l)$ 之和小于 p_l 的容量,则变迁可被触发。

2.4 蚂蚁触发变迁(规则4)

蚂蚁触发被选中的变迁 t_j 时,不论变迁 t_j 有几个输入库所算法都将向该变迁的每个输出库所中输出一个蚂蚁。蚂蚁 a_i 触发变迁时与 $t_j, \text{BestRoutes}_i$ 比较并为其赋值,并把该变迁添加到它的路径队列 a_i, Route 中; a_i 到达 t_j 的后向库所时,同样把该库所添加到它的路径队列 a_i, Route 中,并将当前位置 a_i, Place 改为该库所。当变迁输入的蚂蚁数大于输出的蚂蚁数时,多余的蚂蚁将在变迁中死去;当变迁输出的蚂蚁数大于输入的蚂蚁数时,多余的蚂蚁将由变迁中产生。

2.5 信息素更新(规则5)

蚂蚁每触发一次变迁就在变迁上留下 $\Delta\tau$ 单位的信息素;如果某变迁上的信息素超过 μ 个单位,那么在

那里不再留下信息素。当 t 被触发时,有:

$$t.\text{Pheromone}_{\text{new}} = t.\text{Pheromone}_{\text{old}} + \Delta\tau(t) \\ \text{Pheromone}_{\text{old}} < \mu)$$

当蚂蚁到达目的库所时,蚂蚁找到一条路径 r ; 并以计算路径性能 c , 并在相关变迁(网状路径中包含的变迁)中增加 $\omega \times c$ 单位的信息素(ω 用来控制信息素的增量); 如果某变迁的信息素超过 λ 个单位, 那么在那里不再留下信息素。

$$t_j.\text{Pheromone}_{\text{new}} = t_j.\text{Pheromone}_{\text{old}} + \omega \times c(t_j) \\ \text{Pheromone}_{\text{old}} < \lambda, t_j \in r) \quad (2)$$

为了避免信息素堆积, 在每个时间步骤内, 每个变迁有固定比率 e 的信息素挥发。

$$t.\text{Pheromone}_{\text{new}} = t.\text{Pheromone}_{\text{old}} \times (1 - e)(t) \\ \text{Pheromone}_{\text{old}} > 0, t \in T) \quad (3)$$

2.6 清除蚂蚁(规则 6)

堆积在目的库所的蚂蚁将被清除。为避免蚂蚁堆积, 在更新路径信息素并保存路径之后目的库所中的蚂蚁将被清除。

每个时间步骤内, 算法遍历所有库所, 当发现一个库所中的蚂蚁达到 ϵ 个时, 算法将清除该库所中的所有蚂蚁。对于目的库所, 在清除蚂蚁之前将更新路径信息素并保存路径。

2.7 生成蚂蚁(规则 7)

由于蚂蚁在目的库所被清除, 它们不能返回源库所; 随着算法的运行 LS-SPN 中的蚂蚁不断减少, 算法将陷入僵局。为此, 在每个时间步骤内, 有 θ 只蚂蚁在源库所中生成。

2.8 网状路径寻找(规则 8)

当蚂蚁 $a_k(1 \leq k \leq l)$ 到达目的库所, 即 $a_k.\text{Place} = p_{\text{end}}$ 时, $pr_i = a_k.\text{Route}(1 \leq i \leq q), r_i = F_r(a_k.\text{Route}, T.\text{BestRoutes})$, 蚂蚁到达 p_{end} 时所携带的路径信息是一个部分路径, 网状路径的求解方法如下:

Step1: 标记 pr 中的变迁、库所和有向弧, 被标记的元素构成标记集。

Step2: 考查标记集中的第一个变迁, 并使 $\text{ifmark} = \text{false}$ 。

Step3: 如果变迁有最优路径没有被标记, 则标记这些路径中的变迁、库所和有向弧, 并使 $\text{ifmark} = \text{true}$ 。

Step4: 如果变迁的最优路径都被标记了, 则什么都不作。

Step5: 如果变迁是标记集的最后一项变迁, 则执行 Step6; 否则考查下一项变迁, 执行 Step3。

Step6: 如果 $\text{ifmark} = \text{true}$, 则执行 Step2, 否则算法结束。

3 网状路径性能以及系统性能的计算

3.1 部件可靠性服从指数分布的路径可靠性解算

系统中每条路径对应系统的一个子系统, 每个变迁都对应于子系统中的一个部件; 当对系统的可靠性建模分析时, 考虑该子系统是一个复合不可修系统的情况。下面给出根据各部件可靠性指标计算子系统可靠性指标的方法。

(1) 复合子系统的可靠性。

复合子系统由串联和并联子系统复合构成, 大部分路径都是串联复合而成, 每条复合路径据其具体情况计算平均寿命。对于此类并串联复合子系统, 计算路径的可靠性时只要把并联部分看作一个单元, 完全可按串联系统处理。此时

$$R(t) = \prod_{j=1}^n R_j(t) = \exp\left\{-\sum_{j=1}^n \lambda_j t\right\}, \lambda_j > 0, t \geq 0, 1 \leq j \leq n$$

$$\text{MTTF} = \int_0^{\infty} R(t) dt = \int_0^{\infty} \prod_{j=1}^n R_j(t) dt \quad (4)$$

式(4)中 $R_j(t)$ 可能是由多个部件并联构成的复合单元的可靠性函数, 在应用中根据实际情况计算出 $R_j(t)$ 。

(2) 路径性能的计算方法。

以路径的平均寿命作为反映路径可靠性的指标, MTTF 的值越大路径的性能越高, 路径性能的取值在 0 到 1 之间, 取值越接近 1 路径性能越高。下面给出分析系统可靠性时路径性能 c_r 的计算公式:

$$c_r = F_r(\text{MTTF}) = \frac{1}{2} [1 + \tanh(\text{MTTF}/10 - 1)]$$

3.2 部件延时服从指数分布的路径延时解算

当对系统的延时建模分析时, 与变迁对应的参数 λ_j 表示该部件的延时服从参数为 λ_j 的指数分布。假设部分路径中包含的变迁为 t_j , 其延时为一个随机变量 X_j , 服从参数为 λ_j 的指数分布 ($1 \leq j \leq n, n$ 是 pr 中库所的数目)。路径的平均速率为

$$v = \frac{1}{E(X)} = 1 / \left(\sum_{i=1}^n 1/\lambda_i \right) \quad (5)$$

路径性能的取值在 0 到 1 之间, 取值越接近 1 路径性能越高。下面给出分析子系统延时时路径性能 c_v 的计算公式(6):

$$c_v = F_r(v) = \frac{1}{2} \{1 + \tanh[1/(10 \sum_{i=1}^n 1/\lambda_i) - 1]\} \\ = \frac{1}{2} [1 + \tanh(v/(10 - 1))] \quad (6)$$

3.3 系统总体性能解算

可根据以上算法给出的路径以及各路径的性能计算系统性能。考虑由多条路径构成的系统为冷储备系

统的情况。设系统由 n 条路径组成,在初始时刻,一条路径开始工作,其余路径作冷储备。系统的平均寿命为

$$MTTF_s = E\{Y_{r,1} + Y_{r,2} + \dots + Y_{r,n}\} = \sum_{i=1}^n EY_{r,i} = \sum_{i=1}^n MTTF_{r,i} \quad (7)$$

路径 r_i 的平均速率为 v_i ,则系统的平均速率为

$$v_s = \sum_{i=1}^n v_i / n \quad (8)$$

工作路径的选择策略遵从以下规则:

如果路径转换时间相对于路径延时很小,系统在 t 时刻的工作路径 r_t 为

$$\text{if } v_j = \max\{v_1, v_2, \dots, v_m\} \text{ then } r_t = r_j (1 \leq j \leq m, m \leq n) \quad (9)$$

式(9)中, m 为 t 时刻系统未失效的路径数, n 为系统中的总路径数。

如果路径转换的时间相对较大,这种情况下系统在 t 时刻的工作路径 r_t 为

$$\text{if } w_i = l \cdot v_i + (1-l)MFFT_i, w_j = \max\{w_1, w_2, \dots, w_m\} \text{ then } r_t = r_j (1 \leq j \leq m, 1 \leq i \leq m, m \leq n, 0 \leq l \leq 1) \quad (10)$$

式(10)中, m 为 t 时刻系统未失效的路径数, n 为系统中的总路径数, l 为权重。

4 算法步骤及实例研究

下面给出系统性能分析算法步骤。

Step1:系统初始化: $\lambda_j = 0, t_j, \text{Pheromone} = 0, t_j, \text{BestRoute} = \text{Null} (1 \leq j \leq m, m \text{ 为系统中变迁数目}),$ 清除系统中所有蚂蚁,设定蚁群算法迭代次数。

Step2:将系统各部件的平均实施速率参数赋予 λ_j ,指定系统的源库所 p_{start} 目的库所 p_{end} ,开始寻找系统路径。

Step3:根据规则 7 在 p_{start} 中生成蚂蚁。

Step4:考查系统中的第一只蚂蚁。

Step5:根据规则 1,2 为蚂蚁选择一个变迁作为前进方向,如果根据规则 3 判断变迁可触发则转 Step6,否则转 Step7。

Step6:蚂蚁根据规则 4 触发变迁,并根据规则 5 为变迁添加信息素。

Step7:如果所有的蚂蚁都被考查过了则转 Step8;否则考查下一只蚂蚁,执行 Step5。

Step8:如果有蚂蚁到达 p_{end} 则根据规则 8 寻找该蚂蚁经过的网状路径,并根据式(2)为网状路径中的变迁增加信息素。之后将该蚂蚁保存的部分路径及其对应的网状路径保存到路径集中。

Step9:根据规则 6 清除系统中堆积的蚂蚁,并根据式(3)使所有变迁的信息素挥发。

Step10:如果未达到迭代次数转 Step3,否则转 Step11。

Step11:根据实际情况用式(4)计算系统中各路径平均寿命,用式(5)计算系统中各路径平均速率。

Step12:用式(7)计算系统平均寿命,用式(8)计算系统平均效率。

Step13:根据式(9)和式(10)选择当前工作路径。

Step14:解算结束,输出系统平均寿命、系统平均效率和当前工作路径。

5 结束语

文中致力于使用改进启发式算法在复杂的 SPN 环境中寻找网状路径,并解决大规模冗余系统的性能分析问题。给出了 Petri 网中路径的全新定义和在该定义之下寻找路径以及计算路径性能、系统性能的方法。文中提出的系统性能解算方法理论分析严密,通过计算机进行求解不存在理论上的缺陷,从而确保了解算结果的准确性。下一步的工作要考虑对部件性能服从其它分布的网状路径进行性能分析,并引入其它效率更高的算法进行路径寻找。

参考文献:

- [1] 赵不赅,景亮,严仰光. Petri 网的硬件实现[J]. 软件学报, 2002,13(8):1652-1657.
- [2] 郑晓东,崔志明,陈建明. 基于 Petri 网运行状态的组件化软件可靠性分析方法[J]. 计算机技术与发展, 2008,18(4):5-8.
- [3] Molloy M K. Fast bounds for stochastic Petri Nets[C]// In: Proc. Int. Workshop on Timed Petri Nets. Torino, Italy: [s. n.], 1985.
- [4] 黄光球,苏海洋,刘冠. 基于蚁群算法的 Petri 网最优路径序列寻找[J]. 计算机应用, 2007,27(4):932-935.
- [5] 贾再一,陈仕兵. 用智能 Petri 网实现最短路径问题[J]. 微机发展(现更名:计算机技术与发展), 2002,12(4):83-84.
- [6] 乐晓波,王中华,张春丽. 基于 Petri 网的哲学分析[J]. 计算机技术与发展, 2008,18(3):109-113.
- [7] Dorigo M, Caro G D. Ant colony optimization: a new meta-heuristic[C]// Proceeding of the 1999 Congress on Evolutionary Computation. Washington, DC, USA: [s. n.], 1999:1470-1477.
- [8] Kiritis D, Porchet M. A generic Petri net model for dynamic process planning and sequence optimization[J]. Advances in Engineering Software, 1996,25(1):61-71.