

基于改进蚁群算法的数据邮递 QoS 模型

龚 跃 展兴华 吴 航 陈纯义

(长春理工大学软件学院 长春 130022)

摘 要 分布式数据交换网络系统中,海量数据传输一直是数据交换的难题。在进行路由选择时通常考虑跳数和时延,不提供 QoS 保证,致使数据在传输过程中因阻塞或中断长时间延时、严重的长时间延迟而丢弃。提出了改进的蚁群算法构造数据邮递过程的 QoS 模型,以实现满足 QoS 约束、均衡链路负载、快捷高效的海量数据传递。

关键词 分布式系统,蚁群算法,数据邮递,QoS 路由

Model of Data Mail QoS Based on Improved Ant Colony Algorithms

GONG Yue ZHAN Xing-hua WU Hang CHEN Chun-yi

(College of Software, Changchun University of Science and Technology, Changchun 130022, China)

Abstract In the network system of distributed data exchange, the exchange of massive data is a key problem. The employed hop and time delay are considered during a routing selection, and there is not QoS guarantee. The data packages in the transmission process are delayed a long time for obstruction or interrupt, and some of them are discarded for seriously delay. In this paper, the model of data mail QoS based on improved ant colony algorithms realizes the data mailing with satisfaction of QoS routing, load balance and efficiency.

Keywords Distributed system, Ant colony algorithm, Data mail, QoS routing

随着网络技术的飞速发展,网络用户对网络服务质量(QoS, Quality of Service)有了新的要求。而 QoS 着重于多约束路径选择问题,因此,多约束的路径选择问题是服务质量路由研究的重要课题。

目前,对 QoS 路由模型的研究都是将算法应用与路由问题有机结合起来,去解决具体的路由选择问题。如,分布式启发式算法解决时延受限的路由问题^[1,2],模糊逻辑方法解决 QoS 路由问题^[3],遗传算法解决优化网络资源的 QoS 路由选择问题^[4,5]。

蚁群算法是一种由 Dorigo M 等人提出的基于种群模拟进化、用于解决复杂优化问题的启发式算法^[6,7]。蚁群算法从提出后有了快速的发展,也用于解决 QoS 的路由问题,蚁群算法在网络通信方面具有很多成果^[8,14],但在广域网络远程数据邮递方面可参考的成果很少^[15-17]。本文对蚁群算法进行改进,提出了一种保证服务质量并能快速邮递海量数据的策略。

1 数据邮递路径的 QoS 模型

QoS 路由的基本问题就是在源和目的节点之间,如何找到一条能够同时满足多个约束条件且具有最小代价的路径。广域网络中分布式数据库系统节点拓扑结构组成一有向图,设 $G = \{V, E\}$ 表示一个网络模型, V, E 分别表示节点集和链路集, $|V| = n$ 和 $|E| = m$ 分别表示网络的节点数和链路数。设 $(u, v) \in E$, 从源节点 s 到目的节点 d 的路径为 P 。

QoS 模型在选定路径 P 时考虑的参数主要有:

(1) 数据传输率

设 $R(u, v)$ 表示链路 (u, v) 上的数据传输率, R_0 是传输率门阈值,要求选定路径 P 上任意 2 个相邻节点之间的数据传输率满足:

$$\min\{R(u, v) \mid v \in J_k(u)\} \geq R_0 \tag{1}$$

式中, $J_k(u)$ 为与节点 u 相邻的节点集合。

(2) 网络节点负载率

$$S = \{S(u, v) \mid v \in J_k(u)\} \leq S_0 \tag{2}$$

式(2)表示网络模型中任意传输主机节点的负载率,传输主机经过 SOCKET 通信可以获知子机的总数量以及空闲的子机数目,子机空闲得越多则网络负载越小。

(3) 数据量

设 D 表示数据传输量,根据任务配置进行数据解析,即可得到传输任务发送的数据量。

(4) 传输时间

在网络传输中所需要的端到端点的传输时间,可由数据量和传输的即时传输速率得出:

$$T(u, v) = D/R(u, v), v \in J_k(u), (u, v) \in P \tag{3}$$

(5) 链路故障率

设 $L(u, v)$ 表示链路 (u, v) 上的故障率, L_0 是故障率门阈值,要求选定路径 P 的成功率满足:

$$\prod_{(u, v) \in P} (1 - L(u, v)) \geq (1 - L_0), v \in J_k(u) \tag{4}$$

(6) 时间延迟

时间延迟是各个相邻点之间的传输时间 $T(u, v)$ 以及通信时间 T_c 之和。

本文受国家自然科学基金项目(61007046),国家中小企业科技创新基金项目(05C26212200378)资助。

龚 跃(1960-),男,教授,主要研究方向为网络通信。

$$T(u, v) + T_c \leq T_0, v \in J_k(u) \quad (5)$$

数据交换链路 QoS 问题可转化为以下多约束优化路径问题,即寻找一条由源节点到目的节点的路径 P 满足以下条件:

$$\begin{aligned} \min\{R(u, v) | v \in J_k(u)\} &\geq R_0 \\ S = \{S(u, v) | v \in J_k(u)\} &\leq S_0 \\ T(u, v) + T_c &\leq T_0, v \in J_k(u) \end{aligned} \quad (6)$$

求满足 QoS 约束式(6)要求下的可行路径,然后在可行路径中选择所用时间 $T_d(r, s) = \sum_{(u, v) \in P} (T(u, v) + T_c)$ 最少的路径。

2 改进的蚁群算法^[16]

蚁群算法是模拟蚂蚁的行为而提出的一种启发式算法。此改进的蚁群算法是首先将 m 只蚂蚁放置于开始的网络结点上,在算法中位于城市 i 上的蚂蚁 k 以某种规则选择下一个网络结点 j ,若 $U < U_0$,则蚂蚁 k 会从下一个可走的结点集合 $allowed_k$ 中随机地选择一个,若 $U \geq U_0$,则蚂蚁 k 以 $p_{ij}^k(t)$ 为概率函数选择下一个可走的城市 j ,其公式为:

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{jk}(t)]^\beta}{\sum_{s \in allowed_k} [\tau_{is}(t)]^\alpha \cdot [\eta_{sk}(t)]^\beta}, & \text{若 } j \in allowed_k \\ 0, & \text{否则} \end{cases} \quad (7)$$

式中, $\tau_{ij}(t)$ 为 t 时刻路径 (i, j) 上的信息量, $allowed_k$ 表示蚂蚁 k 下一步允许选择的网络结点集; α 为信息启发式因子,表示轨迹的相对重要性,反映了蚂蚁在运动过程中所积累的信息在蚂蚁运动时所起的作用,其值越大,该蚂蚁越倾向于选择其他蚂蚁经过的路径,蚂蚁之间协作性越强; β 为期望启发式因子,表示能见度的相对重要性,反映了蚂蚁在运动过程中启发信息在蚂蚁选择路径中的受重视程度,其值越大,则该状态转移概率越接近于贪心规则; $\eta_{jk}(t)$ 为启发函数,其表达式如下:

$$\eta_{ij}(t) = \frac{1}{d_{ij}} \quad (8)$$

式中, d_{ij} 表示两个城市之间的距离。

为了避免残留信息素过多引起残留信息淹没启发信息,在每只蚂蚁走完一步或者完成对所有 n 城市的遍历后,要对残留信息素进行更新处理。这种更新策略模仿了人类大脑记忆的特点,在新信息不断存入大脑的同时,存储在大脑中的旧信息随着时间的推移逐渐淡化,甚至忘记。由此, $t+n$ 时刻在路径 (i, j) 上的信息素可按如下规则进行调整:

$$\tau_{ij}(t+n) = (1-\rho) \cdot \tau_{ij}(t) + \Delta\tau_{ij}(t) \quad (9)$$

$$\Delta\tau_{ij}(t) = \sum_{k=1}^m \Delta\tau_{ij}^k(t)$$

式中, ρ 表示信息素挥发系数,则 $1-\rho$ 表示信息素残留因子,为了防止信息的无限积累, ρ 的取值范围为: $\rho \in [0, 1)$; $\Delta\tau_{ij}(t)$ 表示本次循环中路径 (i, j) 上的信息素增量,其值在局部更新与全局更新时有所不同,分别是:

在局部更新时:

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{D_{ij}^k}, & \text{若第 } k \text{ 只蚂蚁在 } t \text{ 和 } t+1 \text{ 中经过 } (i, j) \\ 0, & \text{否则} \end{cases} \quad (10)$$

式中, D_{ij}^k 表示第 k 只蚂蚁经过路径 (i, j) 时的时间延迟; Q 表

示信息素强度,对算法的影响不大,只是影响收敛的速度,一般 Q 取值大于 0。

在全局更新时分为两种情况:

第一种情况是,第 k 只蚂蚁所走的 (i, j) 属于局部最优路径 S 中的路径,则 $\Delta\tau_{ij}^k(t)$ 用下式来更新:

$$\Delta\tau_{ij}^k(t) = F_1 \cdot F_2 \cdot \frac{Q}{\sum_{(i, j) \in S} D_{ij}^k} \quad (11)$$

其中,上述公式中参数含义为:

$$F_1 = 1 + \frac{D_{pre} - \sum_{(i, j) \in S} D_{ij}^k}{\sum_{(i, j) \in S} D_{ij}^k} \quad (12)$$

$$F_2 = 1 + \frac{\frac{m}{2} - m_s}{m} \quad (13)$$

D_{ij}^k 表示第 k 只蚂蚁经过路径 (i, j) 时的时间延迟;

D_{pre} 表示在历次迭代中局部最优的解中的最小值,即时延最小的解;

m_s 表示通过局部最优路径 S 的蚂蚁数量;

F_1 函数用来抑制在信息素全局更新时次优解的信息素增加,提高最优解的信息素增加。此函数能有效地避免最终结果出现次优解。

F_2 函数是对每次局部最优路径上信息素量增加的控制。若在某次迭代中通过局部最优路径 S 的蚂蚁数量超过一半,则对该路径上的信息素增加量进行抑制,让其增加少些。此函数能有效地控制信息素的增长速度,避免出现早熟。

第二种情况是,第 k 只蚂蚁所走的 (i, j) 不属于局部最优路径 S 中的路径,此时 $\Delta\tau_{ij}^k(t)$ 用下式来更新:

$$\Delta\tau_{ij}^k(t) = \begin{cases} 0, & m_{ij} < \frac{m}{2} \\ -F_3 \cdot \frac{Q}{D_{ij}^k}, & m_{ij} \geq \frac{m}{2} \end{cases} \quad (14)$$

$$F_3 = 1 + \frac{m_{ij} - \frac{m}{2}}{m}$$

m_{ij} 表示通过路径 ij 的蚂蚁数量。

F_3 函数是对每次非局部最优路径上信息素量增加的控制。若在某次迭代中通过非局部最优路径 ij 的蚂蚁数量超过一半,则对该路径上的信息素增加量进行抑制,此举可以分流链路负载,平衡相关链路流量,控制阻塞。

3 实验模拟

模拟验证改进的蚁群算法在此模型基础上的传输模型,同时与 Dijkstra 作比较。网络结点原始数据参数如图 1 所示。

首先预处理,即 QoS 条件过滤。其理论是读取数据传输率阈值、结点间的传输时延阈值以及传输节点的故障率阈值,删除那些数据传输率小于传输率阈值和数据故障率大于故障率阈值的链路,得到:

$$G'(V', E') = \{V', E', \{(u, v) \in P, R(u, v) | v \in J_k(u)\} \geq R_0, \sum_{(u, v) \in P} T(u, v) + T_c \leq T_0, \{\prod_{(u, v) \in P} (1-L(u, v)) \geq (1-L_0)\}\} \quad (15)$$

式中, $J_k(u)$ 为与节点 r 相邻的节点集合。这样过滤路径不仅可以满足 QoS 约束,而且减少了算法的搜索空间。在 G' 中,对于每条边 V' ,置其权重 $w(u, v) = D/R(u, v) + T_c$,其中, D

为传输任务中每个数据包的数据量, 结点之间的通信时间极短, 因此, T_c 可以忽略不计, 所以时间延迟都在 $D/R(u, v)$ 部分。

其次是条件设定, 即设定 QoS 约束条件的值。图 1 中网络结点的权值是模拟在 1MB/s 带宽的路由器上的网络节点间的最大传输率, 单位为 kB/s, QoS 的 R_0 约束条件 $R_0 = 200\text{ kB/s}$ 。在多任务传输的实时系统中, 每个定制的任务对于给定的一对源节点和目的节点, 如果每次选择同一路径, 那么会造成数据链路的网络带宽急剧缩减, 形成链路拥塞。附加 QoS 约束条件, 在网络节点的最优路径选择中引入网络负载率、故障率等因子, 进行平衡过滤。加入负载率和故障率的网络结点。模拟网络节点负载率与故障率参数如表 1 所列。假设结点承载的能力无限大, 即无论有多少数据传输过来, 结点都可以接收; 满足以上 QoS 约束条件之后网络结点模型演化为图 2。QoS 的负载率和故障率约束条件分别设为: $S_0 = 60\%$, $L_0 = 0.4\%$ 。

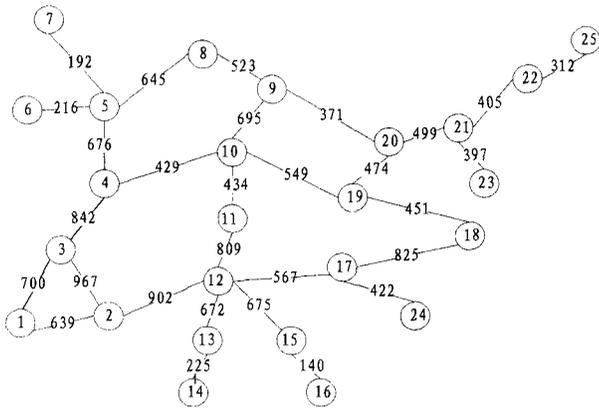


图 1 传输拓扑网络

表 1 模拟网络结点负载率和故障率参数

节点	负载率 (%)	故障率 (%)	节点	负载率 (%)	故障率 (%)	节点	负载率 (%)	故障率 (%)
1	40	0.1	10	50	0.3	19	50	0.2
2	50	0.1	11	40	0.2	20	40	0.2
3	60	0.2	12	30	0.3	21	70	0.2
4	50	0.3	13	30	0.3	22	30	0.1
5	40	0.2	14	60	0.5	23	40	0.1
6	20	0.1	15	30	0.1	24	80	0.5
7	20	0.5	16	20	0.6	25	20	0.1
8	40	0.1	17	50	0.2			
9	20	0.2	18	60	0.1			

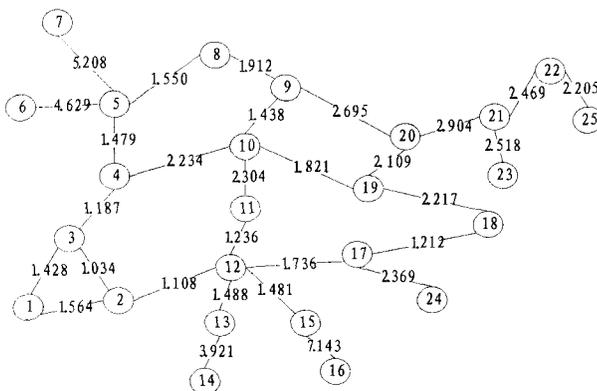


图 2 加入负载率和故障率的拓扑网络

最后为 QoS 条件下的实验模拟。在相同的条件下分别运行改进的蚁群算法和 Dijkstra 算法, 从结点 1 到结点 20 传输等量的数据量, 并记录各自总的传输时延。其中蚁群算法中, 蚂蚁的数目为 50 只, $\alpha=2, \beta=5, \rho=0.5$ 。蚁群算法是先训练传输数据的网络, 因蚁群算法前期蚂蚁的随机性, 而不是一个稳定的网络, 通过前期的训练, 信息素在训练过程中趋于稳定, 即找出了最优路径、次优路径或者次次优路径。而 Dijkstra 算法考虑到网络的稳定等问题, 在每个数据包传递之前都先选择最优路径, 以免出现网络中断现象。

在传输的数据量方面, 考虑到现在数据量大型化, 所以选择了相对大的数据量, 而传输的数据量分为若干个数据包, 每个数据包依次为 2M, 4M, 6M, ..., 20M, 改进的蚁群算法中每只蚂蚁运动时模拟携带一个数据包, 算法的完成时间就是所有的蚂蚁从开始结点走到目的结点。而 Dijkstra 算法没有蚂蚁这种运动的对象只能是模拟数据包的运动, 每个结点主机负责数据包的收发。最后结果如表 2 所列, 图 3 为表 2 的折线图。

表 2 改进的蚁群算法和 Dijkstra 算法所用时间对比

数据总量(M)	改进的蚁群算法所用时间(s)	Dijkstra 算法所用时间(s)
100	274.34	252.56
200	383.11	499.09
300	557.95	745.42
400	735.77	991.42
500	874.50	1238.15
600	1056.28	1483.73
700	1254.19	1730.69
800	1371.93	1975.33
900	1509.86	2222.64
1000	1712.78	2468.39

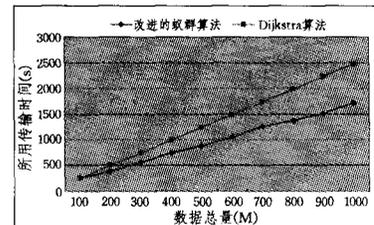


图 3 改进的蚁群算法与 Dijkstra 算法在传输时间上的对比

由上述实验可知, 在数据量比较小的情况下, Dijkstra 算法都是选择最优路径, 传递时间必然最短。而改进的蚁群算法中数据包是在训练好的网络中传递, 当在最优链路被占用时, 会选择其他的次优路径, 这必然耗费一定的额外时间, 造成总传输时间相对 Dijkstra 算法较长。但是随着数据量的加大, 最优链路会出现严重阻塞, 蚁群算法搜索路径机理表现出优越性, 因为大数据包在一段链路中就要耗费大量时间, 如果有多条路径时, 整体传递的数据流量自然增加。所以在海量数据交换时改进的蚁群算法优于 Dijkstra 算法。

结束语 在广域网络分布式海量数据库系统数据交换中 (动态异地备份或迁移), 由于任务堆积经常造成网络瘫痪, 因此, 根据网络节点不同的运行状况利用改进蚁群算法 QoS 模型会大大减少网络的阻塞, 提高传输任务的成功率。从实验仿真结果可以看出, 在改进的算法中, 动态网络负载均衡有利于网络的稳定性, 可以尽可能避免网络拥塞。

参考文献

[1] 刘莹, 吴建平. 求解带时延约束组播路由问题的启发式遗传算法

- [J]. 计算机研究与发展, 2003, 40(3): 381-386
- [2] 陈曦, 柳林. 基于遗传算法的时延受限多播路由研究[J]. 计算机工程与应用, 2002(17): 170-171
- [3] Aboelela E, Douligeris C. Fuzzy multiobjective routing model in B-ISDN. *Computer Communications*, 1998, 21(17): 1571-1584
- [4] 孙宝林, 李腊元. 基于遗传算法的 QoS 多播路由优化算法[J]. 计算机工程, 2005, 31(14): 70-73
- [5] 韩院彬, 张京军, 王立国. 基于改进遗传算法的满足可靠性 QoS 约束的组播路由算法[J]. 计算机应用与软件, 2009, 29(5): 98-100
- [6] Dorigo M, Colomia M V. Ant system: Optimization by a colony of cooperating agents[J]. *IEEE Transactions on Systems, Man and Cybernetics*, 1996, 26(1): 29-41
- [7] Dorigo M, Carog D, Gambarsellal M. Ant algorithms for discrete optimization[J]. *Artificial Life*, 1999, 5(2): 137-172
- [8] Ke Xiong. Multi-constrained Shortest Disjoint Paths for Reliable QoS Routing[J]. *ETRI Journal*, 2009, 31(5): 534-544
- [9] Liu C H. Cross-Layer Design for QoS in Wireless Mesh Networks[J]. *Wireless Pers Commun*, 2009, 51: 593-613
- [10] Mantar H A. A QoS Routing Architecture for Multi-Domain Networks[C]//*Proceedings of the Seventh IEEE International Symposium on Computer Networks (ISCN'06)*. IEEE, 2006
- [11] Huang han, Hao Zhi-feng. The convergence speed of ant colony optimization[J]. *Chinse journal of computers*, 2007, 30(8): 1345-1353
- [12] Mantar H A. A QoS Routing Architecture for Multi-Domain Networks[C]//*Proceedings of the Seventh IEEE International Symposium on Computer Networks (ISCN'06)*. IEEE, 2006
- [13] Ke Xiong. Multi-constrained Shortest Disjoint Paths for Reliable QoS Routing[J]. *ETRI Journal*, 2009, 31(5): 534-544
- [14] Liu C H. Cross-Layer Design for QoS in Wireless Mesh Networks[J]. *Wireless Pers Commun*, 2009, 51: 593-613
- [15] Gong Yue, Song Ying-ying, Wang Yu-zhuo. Data Mail Delivery QoS Model Based on Distributed System[J]. *Computer Engineering*, 2010, 36(4): 103-104
- [16] 龚跃, 宋瑛瑛, 付慧霞, 等. 基于数据邮递的并行传输技术研究. 长春理工大学学报: 自然科学版, 2008, 31(3): 158-159
- [17] Gong Yue, Wu Hang, Bao Jie, et al. Research on Data Routing Model Based on Ant Colony Algorithms[J]. *Journal of China Ordnance*, 2010, 6(4): 269-272

(上接第 290 页)

可以通过求像系统的约简来获得原系统的约简。

如果不对上述例子中的一致覆盖决策系统 (U, Δ, D) 进行数据压缩处理, 即在给定的系统上直接进行属性约简, 根据定义 9, 常见算法如下: 首先观察覆盖 C_1 是否必要。然后根据定义 3, 计算出论域 U 中 15 个对象在覆盖集 $\{C_2, C_3\}$ 中的邻域。就计算邻域这一步, 我们在观察任意一个覆盖是否必要时, 计算量都是比较大的。而运用同态函数对系统进行数据压缩后, 论域中的对象变为 5 个, 明显提高了约简效率。

结束语 近年来, 属性约简一直是粗糙集理论中的热门话题。就覆盖粗糙集, 各式各样的属性约简方法相继被提出, 本文也是关于约简方法的讨论。与其他约简方法不同的是, 本文无需提出新的约简算法就实现了高效性。运用同态函数将一个数据量较大的一致覆盖决策系统压缩成一个相对规模较小但与之等价的像决策系统, 从而可以将对原决策系统的属性约简转移到压缩产生的像决策系统上, 这大大提高了约简效率。在今后的工作中, 我们将对非一致覆盖决策系统做出相应的研究, 并进一步探讨新的属性约简方法。

参 考 文 献

- [1] Palwak Z. Rough set[J]. *International computer and Information Sciences*, 1982, 11: 341-356
- [2] Zakomski W. Approximations in the space (U, II) [J]. *Demonstratio Mathematica*, 1983, 16: 761-769
- [3] Yao Y Y. Relational interpretations of neighborhood operators and rough set approximation operators[J]. *Information Sciences*, 1998, 111: 239-259
- [4] Zhu W, Wang F Y. Reduction and axiomization of covering generalized rough sets[J]. *Information Sciences*, 2003, 152: 217-230
- [5] Zhu W. Relationship among basic concepts in covering-based rough sets[J]. *Information Sciences*, 2009, 179(14): 2478-2486
- [6] Yao Y Y. Constructive and algebraic methods of the theory of rough sets[J]. *Information Sciences*, 1998, 109: 21-47
- [7] 祝峰, 何华灿. 粗集的公理化[J]. 计算机学报, 2000, 23: 331-333
- [8] 杨勇, 朱晓钟, 李廉. 覆盖粗糙集的公理化[J]. 计算机科学, 2009, 36(5): 181-182
- [9] Zhu W. Topological approaches to covering rough sets[J]. *Information Sciences*, 2007, 177(6): 1499-1508
- [10] Chen D G, Wang C Z, Hu Q H. A new approach to attribute reduction of consistent and inconsistent covering decision systems with covering rough sets[J]. *Information Sciences*, 2007, 177: 3500-3518
- [11] 张亚军, 王艳平, 付上金. 基于覆盖粗糙集理论中的约简与求核[J]. 模糊系统与数学, 2007, 21(6): 152-156
- [12] Li D Y, Ma Y C. Invariant characters of information systems under some homomorphisms[J]. *Information Sciences*, 2000, 129: 211-220
- [13] Wang C Z, Chen D G, Wu C, et al. Data compression with homomorphism in covering information systems[J]. *International Journal of Approximate Reasoning*, 2011, 52: 519-525
- [14] Graymala-Busse J W. Algebraic properties of knowledge representation systems [C]//*Proceedings of the International Symposium on Method for Intelligent Systems*. 1986: 432-440
- [15] Wang C Z, Wu C X, Chen D G, et al. Some properties of relation information systems under homomorphisms[J]. *Applied Mathematics Letters*, 2008, 21: 940-945
- [16] Wang C Z, Chen D G, Zhu L K. Homomorphisms between fuzzy information systems[J]. *Applied Mathematics Letters*, 2009, 22: 1045-1050