

图的最优矩阵构建研究

袁廷磊, 吾守尔·斯拉木

(新疆大学 信息科学与工程学院, 新疆 乌鲁木齐 830046)

摘要:为了提高图的最优矩阵的构建效率,文中通过对 Floyd 算法的研究,进一步提出了对其进行四层优化的方法,通过对图的矩阵中的特殊元素的删除和在计算前的判断减少了不必要的计算,加入贪心算法使其减少中间结果的生成,使中间结果更加接近图的最优矩阵。优化后的 Floyd 算法在很大程度上提高了执行效率,使其在实际应用中更加可取,经过逐步的优化使改进后的算法在运行时间上平均时间最多减少为原来的四分之一,而且随着图顶点数目的增加,修改后的 Floyd 算法效率有显著的提高,因此,在实际应用中是一个切实可行的算法。

关键词:最短路径; Floyd 算法; 贪心算法; 最优矩阵

中图分类号: TP301.6

文献标识码: A

文章编号: 1673-629X(2013)07-0151-04

doi: 10.3969/j.issn.1673-629X.2013.07.039

Research on Optimal Matrix Construction of Diagram

YUAN Ting-lei, WUSHOUER Silamu

(College of Information Science and Engineering, Xinjiang University, Urumqi 830046, China)

Abstract: In order to improve the efficiency on construction of best matrix, through the Floyd algorithm, further propose its four optimization method, unnecessary calculation is reduced by the deletion of a particular element in FIG matrix and judgment before calculating, by adding a greedy algorithm to it reduce the generation of the intermediate result, the intermediate results are for closer to the best matrix. The optimized Floyd algorithm improves the efficiency of the implementation to a great extent, it is more desirable in practical applications, the improved algorithm through a step-by-step optimization in the average time on the running time reduced to a maximum of a quarter of the original, but also with the increase in the number of vertices by the figure, the modified Floyd algorithm efficiency is significantly improved. Therefore, it is a practical algorithm in the practical application.

Key words: the shortest path; Floyd algorithm; greedy algorithm; optimal matrix

0 引言

最短路径^[1]问题作为计算机领域中的一个重要的问题,在实际应用中有许多的算法,其中 Floyd^[2,3]算法和 Dijkstra^[1,4]算法是两个经典的算法,广泛为许多问题所应用,在这两个算法中 Dijkstra 算法是用来解决单源点到图^[5]中其它顶点之间的最短路径问题, Floyd 算法是用来解决图中各对顶点的最短路径问题。

Floyd 算法的主要思想^[2,3]:通过初始矩阵 $D^{(0)} = \{d_{ij} \mid i = 1, 2, 3, \dots, n; j = 1, 2, 3, \dots, n\}$,即点 V_i 和 V_j 不经过第三顶点可直接到达的点之间的权值 d_{ij} (也就是点 V_i 到点 V_j 的距离)。然后由 $D^{(0)}$ 求出 $D^{(1)}$,按照递归的思想依次求解出 $D^{(1)}, D^{(2)}, \dots, D^{(n)}$ 。矩阵 $D^{(n)}$ 的 i 行 j 列元素便是 i 号顶点到 j 号顶点的最短路径^[6]

长度,称 $D^{(n)}$ 为图的最短距离矩阵(在文中也称为最优矩阵),同时还可引入一个后继节点矩阵 path 来记录两点间的最短路径。

Floyd 算法的算法过程:

第一步:根据图构造出图的邻接矩阵,即文中所述的初始矩阵 $D^{(0)}$:

$$D_{ij}^{(0)} = \begin{cases} d_{ij} & \text{点 } i \text{ 和点 } j \text{ 相邻} \\ 0 & i = j \\ \infty & \text{点 } i \text{ 和点 } j \text{ 不相邻或无回路} \end{cases}$$

第二步:构造迭代矩阵 $D^{(k)} = d_{ij}^{(k)}$,其中:

$$d_{ij}^{(k)} = \min \{d_{ir}^{(k-1)} + d_{rj}^{(k-1)} \mid r = 1, 2, \dots, n\}$$

第三步:若 $D^{(k+1)} = D^{(k)}$,则迭代终止,否则返回第二步。

收稿日期:2012-09-21

修回日期:2012-12-27

网络出版时间:2013-03-05

基金项目:新疆自治区科技计划项目(200732143)

作者简介:袁廷磊(1986-),男,硕士研究生,CCF 会员,研究方向为嵌入式开发;吾守尔·斯拉木,博士生导师,教授,研究方向为多语种信息处理、计算机应用及网络技术研究与开发应用。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20130305.0819.052.html>

相应的算法伪代码(文中记为算法 1):

```

Floyd-shortest-paths( D )
1   n ← rows[ L ]
2   let L be an n × n matrix
3   for i ← 1 to n
4     do for j ← 1 to n
5       do l'ij ← ∞
6         for k ← 1 to n
7           do l'ij ← min( l'ij, l'ik + d'kj )
8   If L is equal to D
9   return L
10  else Floyd-shortest-paths( L )
    
```

Floyd 算法的运行效率^[7]:

由于该算法需要三重循环,遍历了矩阵中的所有点,所以每次迭代执行时间复杂度为 n^3 。如果图中的最优路径中间顶点为 k 个,则找到最优矩阵的执行时间是 $k * n^3$

1 Floyd 算法抽象为数学模型

Floyd 算法的算法模型^[8]可表示为一个数学矩阵^[9,10],为了更能说明问题,假定有五个顶点的图,则相应的矩阵如图 1。

$$\begin{pmatrix} d_{11} & d_{12} & d_{13} & d_{14} & d_{15} \\ d_{21} & d_{22} & d_{23} & d_{24} & d_{25} \\ d_{31} & d_{32} & d_{33} & d_{34} & d_{35} \\ d_{41} & d_{42} & d_{43} & d_{44} & d_{45} \\ d_{51} & d_{52} & d_{53} & d_{54} & d_{55} \end{pmatrix}$$

图 1 图的抽象矩阵

当 $i=j$ 时, $d_{ij} = 0$, 因此由图 1 转换成图 2, 在 Floyd 算法中, 当计算 d_{ij} 时, r 的值由 1 到 n 逐个扫描各个顶点, 相当于在矩阵中这样计算: 假设当前要计算的点为 d_{13} , 则在矩阵中相当于以 d_{13} 为矩形的一个顶点, 以矩阵的主对角线上的元素为相应矩形的对顶点, 则此时可以确定一个矩形, 然后图的最短路径的查找相当于计算矩形上的另外两个对顶点, 如果此时 $r=1$, 则计算

$$\begin{pmatrix} 0 & d_{12} & d_{13} & d_{14} & d_{15} \\ d_{21} & 0 & d_{23} & d_{24} & d_{25} \\ d_{31} & d_{32} & 0 & d_{34} & d_{35} \\ d_{41} & d_{42} & d_{43} & 0 & d_{45} \\ d_{51} & d_{52} & d_{53} & d_{54} & 0 \end{pmatrix}$$

图 2 转换后图的抽象矩阵

d_{11} 和 d_{13} 相加, 即相当于 Floyd - shortest - paths(L, D) 算法中的第 7 句, 如图 3 所示(虚线为接下两次要计算的情况)。

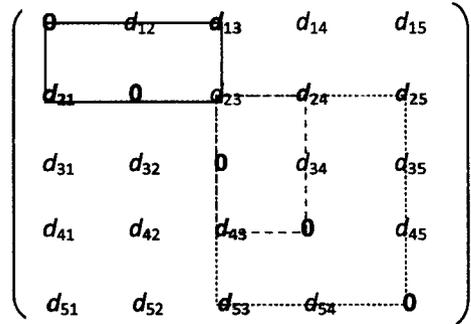


图 3 最短路径运算过程

图 3 所示的运算方式相当于 Floyd - shortest - paths(L, D) 算法中的内循环(伪代码中的第 6 行和第 7 行, 这里去除了当 $r=i, r=j$ 的两个矩阵中的元素), 可以发现图中只进行了三次加法运算, 对于这个问题下文将会给出相应的论证。

2 Floyd 算法的优化

2.1 Floyd 算法的第一层优化

在算法 1 中, 经过对该算法的分析可以发现, 在每次计算 d_{ij} 最小值时, 其中有两点是不必要考虑的, 即当 $r=i$ 或 $r=j$ 时, 每次求解过程都多进行了两次加法运算, 在程序运行过程中浪费了时间。证明如下:

如果 $r=i$, 算法 1 中的式子则变为: $d_{ij} = \min\{ d_{ii} + d_{ij} \}$; 很明显: $d_{ii} = 0$, 则 $d_{ii} + d_{ij}$ 的值就等于 d_{ij} 。所以不可能比 d_{ij} 的值小。因此此处的加法运算是多余的。同理证明当 $r=j$ 时, $d_{ii} + d_{ij}$ 的加法运算也是多余的。

因此可以将矩阵中相应的两个元素不进行加法运算。这点也在 Floyd 算法数学模型——矩阵中只画出了 $n-2$ 个矩形, 也就是进行了 $n-2$ 次加法运算。

很明显: 在这一次对算法 1 改进后, 内循环只进行了 $n-2$ 次加法运算。由于是三重循环, 外边的二重循环的次数为 n^2 , 所以经过第一层优化后算法的时间复杂度为: $(n-2) n^2$ 。小于原来的时间复杂度(n^3)。

2.2 Floyd 算法的第二层优化

在数学中, 可以得出:

如果 $a + b < c$, 则 $a < c$ 且 $b < c$ (其中 $a > 0, b > 0$);

如果 $a + b < c$, 则 $a < c$ 或 $b < c$ (其中 a, b 为实数)

根据数学中的这个思想, 由于在计算机中每执行一次加法的时间远远大于执行一次比较的时间, 因此对 Floyd 算法进行第二层优化: 当图中不存在负权值时, 使进入算法 1 的第 7 行的加法运算都满足: $d_{ii} < d_{ij}$

和 $d_{ij} < d_{ji}$; 假设大于和小于的概率相等, 则满足不等式 $d_{ij} < d_{ji}$ 的概率为 $1/2$ 。满足不等式 $d_{ij} < d_{ji}$ 的概率亦为 $1/2$ 。由于这两个条件要同时满足, 所以在 Floyd 算法的第一层优化的基础上, 每次的第三重循环执行加法运算的数学期望为: $(1/2)(1/2)(n-2)$, 即: $(n-2)/4$ 次加法运算, 则在经过第二层优化后 Floyd 算法的时间复杂度为: $(n-2)n^2/4$, 减小为原来的四分之一。同理可以推出当图中含有负权值的情况下, Floyd 算法的时间复杂度为: $(n-2)n^2/2$, 减小为原来的二分之一。

2.3 Floyd 算法的第三层优化

在 Floyd 算法抽象为数学模型的矩阵中, 可以发现当 $i=j$ 时 $d_{ij}=0$ 。该元素表示图中的第 i 个顶点到它本身的距离, 由于图中不含有负回路, 因此可以断定不会存在某一点使 d_{ij} 更小。所以可以把相应的满足 $i=j$ 的元素点从三重循环中直接去掉。由于矩阵的主对角线把矩阵元素分成对称的两份 (d_{ij} 和 d_{ji} 是以对角线为对称轴的一对对称元素), 因此利用矩阵的对称性只考虑一半的矩形, 就可以遍历整个矩阵上的元素, 如图 4 所示:

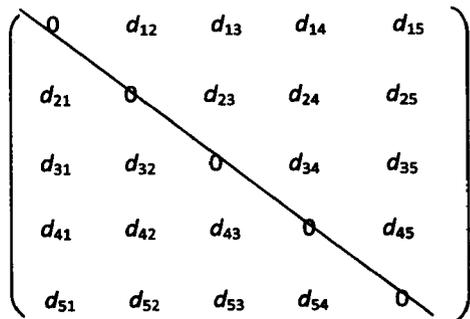


图 4 删除的矩阵元素

2.4 Floyd 算法的第四层优化

从 $D(0)$ 到最优矩阵 D' 的求解过程中, Floyd 算法总是由 $D(0)$ 迭代得出 $D(1)$, 然后由 $D(1)$ 迭代得出 $D(2)$ …… 最终得出问题的解 D' , 在这个过程中, 如果加入贪心算法^[2,11], 即在求解矩阵中的元素 d_{ij} , 假如是 d_{23} , 则可以利用它前面的元素 d_{13}, d_{14} 和 d_{15} 的值, 尽最大可能利用前面计算的结果, 经过一次迭代扫描完矩阵中的元素时得出的不一定是由 $D(0)$ 迭代得出 $D(1)$, 而是一个中间结果 D' 。因此此种方法迭代求出的图的矩阵一定会更接近图的最优矩阵 D' , 所以减少了中间结果的生成, 也就是使迭代次数尽可能地少于原来的次数。通过文献[2]的证明结果可以得出加入贪心算法后的时间复杂度为: $n^3 \leq T(n) \leq n^3 \log 2n$ 。因此加入贪心算法后将会使算法在寻找图的最优矩阵的求解过程中从时间执行效率上更加可

取。而且随着顶点元素的增多时间执行效率体现越明显(参见文中的实验结果)。

3 实验分析

3.1 实验示例

$$\begin{bmatrix} 0 & 13 & 18 & +\infty & 6 \\ +\infty & 0 & +\infty & 11 & 17 \\ +\infty & 14 & 0 & +\infty & +\infty \\ 12 & +\infty & 5 & 0 & +\infty \\ +\infty & +\infty & +\infty & 16 & 0 \end{bmatrix}$$

图 5 初始矩阵

通过上述四层优化方案, 下面给出一个实例计算步骤, 如图 5 是某图的初始矩阵(记为: $D(0)$), 首先按照原始的 Floyd 算法, 其计算如下: (为了方便计算, 文中将无穷大 $+\infty$ 在本示例计算过程中记为 1000)

$$D(0) = \begin{bmatrix} 0 & 13 & 18 & 1000 & 6 \\ 1000 & 0 & 1000 & 11 & 17 \\ 1000 & 14 & 0 & 1000 & 1000 \\ 12 & 1000 & 5 & 0 & 1000 \\ 1000 & 1000 & 1000 & 16 & 0 \end{bmatrix}$$

$$d_{11} = \min \{ d_{11}, d_{11}+d_{11}, d_{12}+d_{21}, d_{13}+d_{31}, d_{14}+d_{41}, d_{15}+d_{51} \}$$

$$\text{即: } d_{11} = \min \{ 0, 0+0, 13+1000, 18+1000, 1000+1000, 6+1000 \} = 0;$$

同理得矩阵中其它元素的值, 经过第 4 次迭代后得出:

$$D(4) = D(3) = \begin{bmatrix} 0 & 13 & 18 & 22 & 6 \\ 23 & 0 & 16 & 11 & 17 \\ 37 & 14 & 0 & 25 & 31 \\ 12 & 19 & 5 & 0 & 18 \\ 12 & 35 & 21 & 16 & 0 \end{bmatrix}$$

由于每计算一个元素的值都需要 5 次运算, 显然, 从 $D(0)$ 到 $D(1)$ 运算的次数为 $25 * 5 = 125$ 次, 由于 $D(4) = D(3)$ 所以最优矩阵为 $D(4)$, 共需要 4 次迭代, 由此可见, 在本示例中, 按照未优化的 Floyd 算法的经过 $4(125) = 800$ 次运算后得出图的最优矩阵, 下面运用文中经过四层优化后的方法计算, 此算法中, 记中间结果矩阵为 D' 其计算过程为:

$$\text{计算过程 (1): } d_{21} = \min \{ d_{21}, d_{24} + d_{41} \} = \min \{ 1000, 11+12 \} = 23, \text{修改的节点: } d_{21} = 23$$

$$\text{计算过程 (2): } d_{31} = \min \{ d_{31}, d_{32} + d_{21} \} = \min \{ 1000, 14+23 \} = 37, \text{修改的节点: } d_{31} = 37$$

$$\text{计算过程 (3): } d_{14} = \min \{ d_{14}, d_{12} + d_{24} \} = \min \{ 1000, 13+11 \} = 24, \text{修改的节点: } d_{14} = 24$$

$$\text{计算过程 (4): } d_{14} = \min \{ d_{14}, d_{15} + d_{54} \} = \min \{ 24, 6+16 \} = 22, \text{修改的节点: } d_{14} = 22$$

计算过程(5): $d_{51} = \min\{d_{51}, d_{54} + d_{41}\} = \min\{1000, 16 + 12\} = 28$, 修改的节点: $d_{51} = 28$

计算过程(6): $d_{23} = \min\{d_{23}, d_{24} + d_{43}\} = \min\{1000, 11 + 5\} = 16$, 修改的节点: $d_{23} = 16$

计算过程(7): $d_{42} = \min\{d_{42}, d_{43} + d_{32}\} = \min\{1000, 5 + 14\} = 19$, 修改的节点: $d_{42} = 19$

计算过程(8): $d_{52} = \min\{d_{52}, d_{54} + d_{42}\} = \min\{1000, 16 + 19\} = 35$, 修改的节点: $d_{52} = 35$

计算过程(9): $d_{34} = \min\{d_{34}, d_{32} + d_{24}\} = \min\{1000, 14 + 11\} = 25$, 修改的节点: $d_{34} = 25$

计算过程(10): $d_{35} = \min\{d_{35}, d_{32} + d_{25}\} = \min\{1000, 14 + 17\} = 31$, 修改的节点: $d_{35} = 31$

计算过程(11): $d_{53} = \min\{d_{53}, d_{52} + d_{23}\} = \min\{1000, 35 + 16\} = 51$, 修改的节点: $d_{53} = 51$

计算过程(12): $d_{35} = \min\{d_{35}, d_{34} + d_{45}\} = \min\{31, 25 + 18\} = 31$, 不需要修改

计算过程(13): $d_{53} = \min\{d_{53}, d_{54} + d_{43}\} = \min\{51, 16 + 5\} = 21$, 修改的节点: $d_{53} = 21$

计算的次数为: 13

修改中间结果次数为: 12

修改后的矩阵:

$$D' = \begin{bmatrix} 0 & 13 & 18 & 22 & 6 \\ 23 & 0 & 16 & 11 & 17 \\ 37 & 14 & 0 & 25 & 31 \\ 12 & 19 & 5 & 0 & 18 \\ 12 & 35 & 21 & 16 & 0 \end{bmatrix}$$

此时 D' 即为图的最优矩阵, 显然计算次数明显比原始 Floyd 算法降低了很多, 不失一般性, 3.2 节给出了一般性最优矩阵的构建模拟实验。

3.2 模拟实验

根据文中的四层优化方法, 优化后的 Floyd 算法(本算法适用于不存在负权值的路径)相应的伪代码可写为:

```

Floyd-shortest-paths( D )
1 n ← rows[ L ]
2 for k ← 1 to n
3   for i ← 1 to n
4     if i equal k
5       then i ← i + 1
6     for j ← i + 1 to n
7       if dik is smaller than dij and dkj is smaller than dij
8         if dij is smaller than dik + dkj
9           dij ← dik + dkj
10      if dki is smaller than dji and djk is smaller than dji
11        if dji is smaller than djk + dki
12          dji ← djk + dki
13 return D
  
```

根据上面优化后的算法代码, 使优化后的 Floyd 算法在运算过程中更加节省时间, 下面将以实验结果证明修改后的算法的可行性。在该实验中应用的实验平台是 Microsoft Visual C++ 6.0^[12]。计算机的主要配置是: CPU: 2.53GHz; 内存: 2G, 执行过程中所用时间(毫秒)的实验结果与时间对比如图 6、图 7。

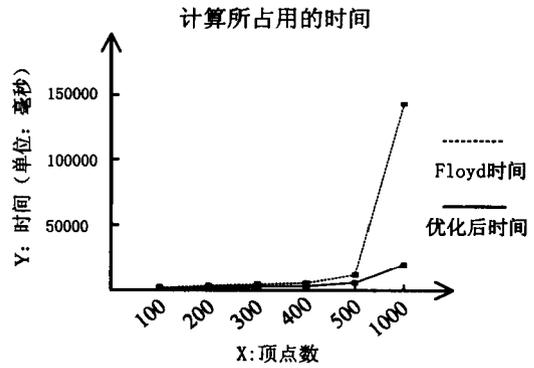


图 6 实验数据

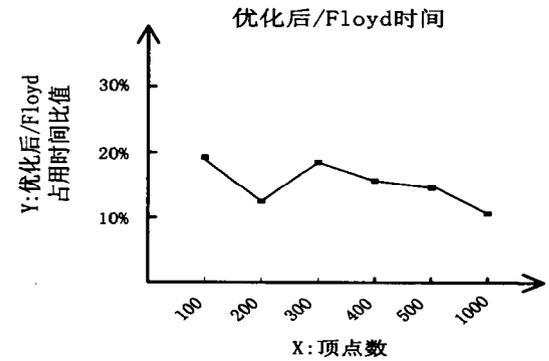


图 7 数据对比图

4 结束语

经过优化的 Floyd 算法, 由于在计算图的最短路径之前先进行了比较, 并且将图的矩阵中不必要的元素和顶点去掉, 使优化后的 Floyd 算法的时间复杂度为原来的四分之一, 当顶点过多达到一定数量时优化后的方法比原来的四分之一还要小。因此优化后的方法在计算量比 Floyd 原方法少了很多, 为构建图的最优矩阵提供了一个高效率的解决方案, 同时也是一种更加实用、高效的优化方法。

参考文献:

- [1] 徐凤生. 最短路径的求解算法[J]. 计算机应用, 2004(5): 88-90.
- [2] Cormen T H, Leiserson C E, Rivest R L, et al. Introduction to Algorithms [M]. 2nd ed. Beijing: Higher Education Press, 2002.
- [3] Floyd R W. Algorithm 97, shortest path[J]. Communications of the ACM, 1962, 5(6): 345-345.
- [4] Dijkstra E W. A note on two problems in connection with

(下转第 175 页)

4 结束语

现有方案很多是基于工业需求考虑的,结构一般较为复杂,需要较强的计算能力,依赖公共 CA 或家庭 CA,或 PKI,或过度依赖特有的平台或编程语言,需要专门的安全管理人员等;没有充分考虑家庭网络用户特点,家庭网络用户大多都是没有足够的专业知识背景,没有和企业一样的专职的和熟练的管理员,不太适合在居民家庭网络中使用;另外很多工作没有将体系结构和安全进行有机结合;很多都是理论上的研究成果,还没有开发出真正的产品。有关家庭网络安全方面的研究还处在起步阶段,离真正能够应用还有相当一段距离,还有大量的研究工作需要去做。为用户提供简单易用和安全的家庭网络环境是家庭网络安全研究的最终目标。

参考文献:

- [1] 彭海清,冯涛,童登金. 家庭网络的关键技术、业务及标准化[J]. 电信科学,2004(11):6-7.
- [2] Kim G W, Lee D G, Han J W, et al. Security Framework for Home Network: Authentication, Authorization, and Security Policy[C]//Proc of the 2007 International Conference on Emerging Technologies in Knowledge Discovery and Data Mining. Heidelberg:Springer,2007:621-628.
- [3] Hwang Jin-Bum, Lee Hyung-Kyu, Han Jong-Wook. Efficient and User Friendly Inter-domain Device Authentication/Access Control for Home Networks[C]//Proc. of the 2006 International Conference on Embedded and Ubiquitous Computing. Heidelberg:Springer,2006:131-140.
- [4] 瞿绍军,刘宏. 家庭网络安全的研究[J]. 航空计算技术,2007,37(2):112-115.
- [5] Shrestha A P, Choi D Y, Kwon G R, et al. Kerberos based authentication for inter-domain roaming in wireless heterogeneous network[J]. Computers and Mathematics with Applications,2010,60(2):245-255.
- [6] Jiang Zhefan, Kim S, Lee Kanghee, et al. Security Service Framework for Home Network[C]//Proceedings of the Fourth Annual ACIS International Conference on Computer and Information Science. South Korea: IEEE Computer Society,2005:233-238.
- [7] Cho E A, Moon C J, Park D H, et al. Access control policy management framework based on RBAC in OSGi service platform[C]//Proc. of 6th IEEE Int Conf. Korea: IEEE Computer Society,2006:161-166.
- [8] Lim H Y, Kim Y G, Moon C J, et al. Bundle authentication and authorization using XML security in the OSGi service platform[C]//Proc. of 4th Annu. ACIS Int. Conf. on Comput. Inf. Sci. . Washington: IEEE Computer Society,2005:502-507.
- [9] Kim I, Lee D, Kim K J, et al. Flexible authorization in home network environments[J]. Cluster Computing,2012,15(1):3-15.
- [10] You I. A One-time Password Authentication Scheme for Secure Remote Access in Intelligent Home Networks[J]. Knowledge-based Intelligent Information and Engineering Systems,2006,4252:785-792.
- [11] Vaidya B, Park J H, Yeo Sang-Soo, et al. Robust one-time password authentication scheme using smart card for home network environment [J]. Computer Communications,2011,34(3):326-336.
- [12] English C, Nixon P, Terzis S, et al. Security Models for Trusting Network Appliances[C]//Proc. of 5th IEEE International Workshop on Networked Appliances. Liverpool: IEEE Press,2002:39-44.
- [13] Mortensen K H, Schougaard K R, Schultz U P. Distance-based Access Modifiers Applied to Safety in Home Networks[C]//Proc. of Ambient Intelligence. Heidelberg:Springer,2004:315-326.
- [14] He Daojing, Ma Maode, Zhang Yan, et al. A strong user authentication scheme with smart cards for wireless communications[J]. Computer Communications,2011,34(3):367-374.
- [15] Helander J. Deeply Embedded XML Communication Towards an Interoperable and Seamless World[C]//Proc. of 5th ACM International Conference on Embedded Software. New York: ACM Press,2005:62-67.
- [16] Feamster N. Outsourcing Home Network Security[J]. Home-Nets,2010(6):37-42.
- [17] 俞银燕,汤帆. 数字版权保护技术研究综述[J]. 计算机学报,2005,28(12):1957-1968.

(上接第154页)

- graphs[J]. Numerische Mathematics,1959,1(1):269-271.
- [5] 方富贵. 图论的算法和应用研究[J]. 计算机与数字工程,2012(2):115-132.
- [6] 卢开澄,卢华明. 组合数学[M]. 第4版. 北京:清华大学出版社,2006.
- [7] Borgwardt K M, Kriegel H P. Shortest-path kernels on graphs [C]//Proceedings of the Fifth IEEE International Conference on Data Mining. [s. l.]:[s. n.],2005.
- [8] 李生彪,黄世华. 数学建模中矩阵的应用[J]. 甘肃联合大学学报,2012(1):22-26.
- [9] 戴华. 矩阵论[M]. 北京:科学出版社,2001.
- [10] 王朝瑞. 图论[M]. 第3版. 北京:北京理工大学出版社,2001.
- [11] 柴婷婷,齐璐,孙斌. 浅析贪心算法[J]. 中国科技博览,2011(20):182-183.
- [12] 潭浩强. VC++程序设计基础教程[M]. 北京:清华大学出版社,2010.

图的最优矩阵构建研究

作者: 袁廷磊, 吾守尔·斯拉木, YUAN Ting-lei, WUSHOUER Silamu
作者单位: 新疆大学信息科学与工程学院, 新疆乌鲁木齐, 830046
刊名: 计算机技术与发展 
英文刊名: Computer Technology and Development
年, 卷(期): 2013, 23(7)

本文链接: http://d.wanfangdata.com.cn/Periodical_wjfz201307039.aspx