



证书号第 1446111 号



发明专利证书

P1200099

发明名称：基于粒子群优化算法的 XQuery 查询路径优化方法

发明人：李浩；赵伟；郑程光；孙伟丰；罗正海；李泉；李书淦；程仁波

专利号：ZL 2012 1 0193940.8

专利申请日：2012 年 06 月 13 日

专利权人：北大方正集团有限公司；方正信息产业控股有限公司
上海方正数字出版技术有限公司

授权公告日：2014 年 07 月 23 日

本发明经过本局依照中华人民共和国专利法进行审查，决定授予专利权，颁发本证书并在专利登记簿上予以登记。专利权自授权公告之日起生效。

本专利的专利权期限为二十年，自申请日起算。专利权人应当依照专利法及其实施细则规定缴纳年费。本专利的年费应当在每年 06 月 13 日前缴纳。未按照规定缴纳年费的，专利权自应当缴纳年费期满之日起终止。

专利证书记载专利权登记时的法律状况。专利权的转移、质押、无效、终止、恢复和专利权人的姓名或名称、国籍、地址变更等事项记载在专利登记簿上。



局长
申长雨

申长雨





(12) 发明专利

(10) 授权公告号 CN 102760167 B

(45) 授权公告日 2014. 07. 23

(21) 申请号 201210193940. 8

(22) 申请日 2012. 06. 13

(73) 专利权人 北大方正集团有限公司

地址 100871 北京市海淀区成府路 298 号方正大厦 5 层

专利权人 方正信息产业控股有限公司
上海方正数字出版技术有限公司

(72) 发明人 李浩 赵伟 郑程光 孙伟丰
罗正海 李泉 李书淦 程仁波

(74) 专利代理机构 北京英赛嘉华知识产权代理有限公司 11204

代理人 王达佐

(51) Int. Cl.

G06F 17/30 (2006. 01)

G06N 3/12 (2006. 01)

(56) 对比文件

CN 101211336 A, 2008. 07. 02, 全文 .

CN 101382945 A, 2009. 03. 11, 全文 .

审查员 苗伟

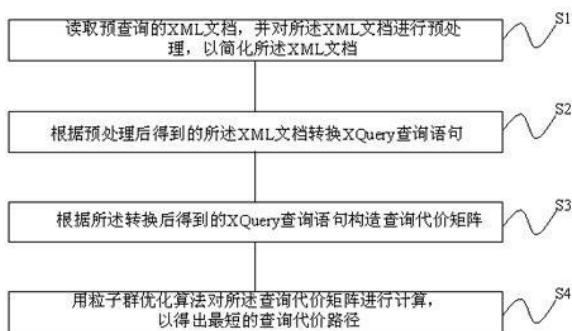
权利要求书2页 说明书9页 附图3页

(54) 发明名称

基于粒子群优化算法的 XQuery 查询路径优化方法

(57) 摘要

本发明公开了一种基于粒子群优化算法的 XQuery 查询路径优化方法, 包括 :S1 : 读取欲查询的 XML 文档, 并对所述 XML 文档进行预处理 ;S2 : 根据预处理后得到的所述 XML 文档转换 XQuery 查询语句 ;S3 : 根据所述转换后得到的 XQuery 查询语句构造查询代价矩阵 ;S4 : 用粒子群优化算法对所述查询代价矩阵进行计算, 以得出最短的查询代价路径。与现有技术相比, 本发明的基于粒子群优化算法的 XQuery 查询路径优化方法, 具有提供相应查询相应时间, 减少查询系统所需要的资源, 较好的利用查询中间结果并对中间结果进行有效的优化处理从而减少中间结果所消耗的系统资源等优点, 从而以较低的资源需求来完成相应的查询。



1. 一种基于粒子群优化算法的 XQuery 查询路径优化方法,其特征在于,包括:

S1:读取预查询的 XML 文档,并对所述 XML 文档进行预处理,以简化所述 XML 文档;

S2:根据预处理后得到的所述 XML 文档转换 XQuery 查询语句;

S3:根据所述转换后得到的 XQuery 查询语句构造查询代价矩阵;

S4:用粒子群优化算法对所述查询代价矩阵进行计算,以得出最短的查询代价路径;

所述步骤 S3 具体为:

S31:获取转换后得到的所述 XQuery 查询语句中的节点编号:MAX_ID 和 MIN_ID,其中,所述 MAX_ID、所述 MIN_ID 分别表示所述 XQuery 查询语句中的节点编号的最大值和最小值;

S32:根据 MAX_ID,MIN_ID 的取值范围建立一个 $(MAX_ID - MIN_ID) * (MAX_ID - MIN_ID)$ 的查询代价矩阵,并将所述查询代价矩阵中的各个元素值设置为一个节点到另一个节点所包含的节点数量,以表示所述两个节点之间的查询路径的权值;

所述步骤 S4 具体为:

步骤 S41:定义粒子群优化算法的目标函数为: $\min: f(x_1, x_2, \dots, x_n)$;

步骤 S42:随机生成 N 个个体,以生成初始种群;

步骤 S43:初始化 N 个个体的初始值,也即使用随机数生成器对所述 N 个个体的初始速度及位置生成初始速度和初始位置;

步骤 S44:计算所述各个个体的适应度值;

步骤 S45:若适应度值小于给定的阈值 d,则终止计算;

步骤 S46:输出最优值,即该算法所寻找到的最优路线,否则进行步骤 S47;

步骤 S47:对粒子个体进行更新;

其中, $f(x_1, x_2, \dots, x_n)$ 为 x_1, x_2, \dots, x_n 路径上的查询代价, min 为去所述目标函数的最小值;

d 为用户给定的阈值,由用户进行指定;

适应度值,用以表明个体的优秀度,其越小则表明所述个体越优秀;

在所述步骤 S47 中,具体按照下述的公式更新所述粒子个体:

$$v_i^{k+1} = w v_i^k + c_1 \text{rand}() (l_i^k - x_i^k) + c_2 \text{Rand}() (g^k - x_i^k) \quad (1)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (2);$$

其中, g^k 为在搜索空间中的优值, l_i^k 为第 i 个粒子在第 k 次迭代过程中所经历的最优位置; w 为算法惯性权值,此数值为经验值在 $[1, 1.3]$; c_1 、 c_2 为加速常数取值为 2; $\text{rand}()$ 、 $\text{Rand}()$ 为两个在 $[0, 1]$ 范围内变化的均匀分布函数。

2. 如权利要求 1 所述的基于粒子群优化算法的 XQuery 查询路径优化方法,其特征在于,所述步骤 S1 具体包括步骤 S11:对 XML 文档的所有节点标签进行预处理。

3. 如权利要求 1 所述的基于粒子群优化算法的 XQuery 查询路径优化方法,其特征在于,所述步骤 S1 具体包括步骤 S12:对 XML 文档中的冗余标签进行处理。

4. 如权利要求 2 所述的基于粒子群优化算法的 XQuery 查询路径优化方法,其特征在于,所述步骤 S11 具体包括:

S111:寻找出所述 XML 文档中所有节点标签信息,并将所述节点标签信息保存至一张节点数据表中,并对所述节点数据表中的每一项赋予唯一编号作为该项在表中的索引值;

S112 :在所述节点数据表的建立完成后,将所述节点在文档中的出现的位置使用其对应的节点索引号进行代替。

5. 如权利要求 3 所述的基于粒子群优化算法的 XQuery 查询路径优化方法,其特征在于,所述步骤 S12 具体包括删除所述 XML 文档的所有节点的右标签。

6. 如权利要求 1 所述的基于粒子群优化算法的 XQuery 查询路径优化方法,其特征在于,所述步骤 S2 具体为:根据所述的预处理后得到的 XML 文档转换原有的 XQuery 查询语句,也即使用所述 XML 文档中的各个节点的节点 ID 表示所述节点,从而将原 XQuery 语句进行转换。

基于粒子群优化算法的 XQuery 查询路径优化方法

技术领域

[0001] 本发明涉及通信技术领域,特别涉及一种基于粒子群优化算法的 XQuery 查询路径优化方法。

背景技术

[0002] XQuery 是用来从 XML 文档查找和提取元素及属性的一种语言,其业已被接纳为 W3 的标准。XQuery 被设计用来查询 XML 文档,但其不仅仅局限于对 XML 文档的查询,而且其还可以对任何以 XML 形式所呈现的数据,当然也包括数据库。XQuery 之于 XML,相当于 SQL 之于关系型数据集。故而,XQuery 被广泛应用于网络服务中的信息提取;信息摘要的生成;将 XML 转化为 HTML;XML 文档的搜索等应用场景下。

[0003] 因此,当需要对一 XML 文档进行查询时,对于所给出的 XQuery 语句的选择执行什么样的查询路径,直接影响查询语句的响应时间。现有技术中,对于查询路径的优化一般或多或少都是基于查询代价来完成查询路径的优化。在用户给出一条 XQuery 查询语句时,系统会根据所给出的 XQuery 查询语句,将该语句分解为数个不同的查询子任务,并根据关系代数或布尔代数做一些基本的等价变化,例如利用 De Morgan 定律对查询条件进行等价处理等。但当执行复杂查询时,或者包含嵌套查询时,现有技术对于此种类型的优化则办法不多。导致上述情形发生的因素包括:(1) 对于 XML 节点扫描消耗过多的资源,导致对于查询大文档时,资源消耗过大,查询时间相应增加;(2) 对于中间结果的处理不够理想。例如在查询语句中需要多次使用查询的中间结果,而系统对于查询结果要么没有进行缓存处理,要么进行缓存处理时没有对中间结果的内容进行处理,从而导致系统所消耗的资源在做查询时增长过快;(3) 对于查询路径没有进行有效编码和优化。

发明内容

[0004] 本发明提供一种基于粒子群优化算法的 XQuery 查询路径优化方法,以解决是现有技术中查询路径消耗资源过大的问题。

[0005] 为解决上述问题,本发明技术方案提供一种基于粒子群优化算法的 XQuery 查询路径优化方法,包括:

[0006] S1:读取预查询的 XML 文档,并对所述 XML 文档进行预处理,以简化所述 XML 文档;

[0007] S2:根据预处理后得到的所述 XML 文档转换 XQuery 查询语句;

[0008] S3:根据所述转换后得到的 XQuery 查询语句构造查询代价矩阵;

[0009] S4:用粒子群优化算法对所述查询代价矩阵进行计算,以得出最短的查询代价路径;

[0010] 所述步骤 S3 具体为:

[0011] S31:获取转换后得到的所述 XQuery 查询语句中的节点编号:MAX_ID 和 MIN_ID,其中,所述 MAX_ID、所述 MIN_ID 分别表示所述 XQuery 查询语句中的节点编号的最大值和最小值;

[0012] S32 :根据 MAX_ID, MIN_ID 的取值范围建立一个 $(MAX_ID - MIN_ID) * (MAX_ID - MIN_ID)$ 的查询代价矩阵,并将所述查询代价矩阵中的各个元素值设置为一个节点到另一个节点所包含的节点数量,以表示所述两个节点之间的查询路径的权值;

[0013] 所述步骤 S4 具体为:

[0014] 步骤 S41: 定义粒子群优化算法的目标函数为 $\min: f(x_1, x_2, \dots, x_n)$;

[0015] 步骤 S42: 随机生成 N 个个体,以生成初始种群;

[0016] 步骤 S43: 初始化 N 个个体的初始值,也即使用随机数生成器对所述 N 个个体的初始速度及位置生成初始速度和初始位置;

[0017] 步骤 S44: 计算所述各个个体的适应度值;

[0018] 步骤 S45: 若适应度值小于给定的阈值 d,则终止计算;

[0019] 步骤 S46: 输出最优值,即该算法所寻找到的最优路线,否则进行步骤 S47;

[0020] 步骤 S47: 对粒子个体进行更新;

[0021] 其中, $f(x_1, x_2, \dots, x_n)$ 为 x_1, x_2, \dots, x_n 路径上的查询代价, min 为去所述目标函数的最小值;

[0022] d 为用户给定的阈值,由用户进行指定;

[0023] 适应度值,用以表明个体的优秀度,其越小则表明所述个体越优秀;

[0024] 在所述步骤 S47 中,具体按照下述的公式更新所述粒子个体:

$$[0025] \quad v_i^{k+1} = w \cdot v_i^k + c_1 \cdot \text{rand}() \cdot (l_i^k - x_i^k) + c_2 \cdot \text{Rand}() \cdot (g^k - x_i^k) \quad (1)$$

$$[0026] \quad x_i^{k+1} = x_i^k + v_i^{k+1} \quad (2);$$

[0027] 其中, g^k 为在搜索空间中的优值, l_i^k 为第 i 个粒子在第 k 次迭代过程中所经历的最优位置; w 为算法惯性权值,此数值为经验值在 $[1, 1.3]$; c_1 、 c_2 为加速常数取值为 2; rand()、Rand() 为两个在 $[0, 1]$ 范围内变化的均匀分布函数。

[0028] 可选地,所述步骤 S1 具体包括:

[0029] S11 :对 XML 文档的所有节点标签进行预处理;

[0030] S12 :对 XML 文档中的冗余标签进行处理。

[0031] 可选地,所述步骤 S11 具体包括:

[0032] S111 :寻找出所述 XML 文档中所有节点标签信息,并将所述节点标签信息保存至一张节点数据表中,并对所述节点数据表中的每一项赋予唯一编号作为该项在表中的索引值;

[0033] S112 :在所述节点数据表的建立完成后,使用各个节点所对应的索引号对所述 XML 文档进行相应的处理,将所述节点在文档中的出现的位置使用其对应的节点索引号进行代替。

[0034] 可选地,所述步骤 S12 具体包括删除所述 XML 文档的所有节点的右标签。

[0035] 可选地,所述步骤 S2 具体为:根据所述的预处理后得到的 XML 文档转换原有的 XQuery 查询语句,也即使用所述 XML 文档中的各个节点的节点 ID 表示所述节点,从而将原 XQuery 语句进行转换。

[0036] 与现有技术相比,上述技术方案具有下优点:

[0037] 本发明的一种基于粒子群优化算法的 XQuery 查询路径优化方法,具有提供相应查询相应时间,减少查询是系统所需要的资源,较好的利用查询中间结果并对中间结果进

行有效的优化处理从而减少中间结果所消耗的系统资源等优点从而已较低的资源需求来完成相应的查询工作。

附图说明

[0038] 图 1 本发明的实施方式的基于粒子群优化算法的 XQuery 查询路径优化方法的流程图；

[0039] 图 2 是本发明的实施方式的基于粒子群优化算法的 XQuery 查询路径优化方法中的步骤 S1 的流程图；

[0040] 图 2a 是本发明的实施方式的基于粒子群优化算法的 XQuery 查询路径优化方法中的步骤 S1 中的步骤 S11 的流程图；

[0041] 图 3 是本发明的实施方式的基于粒子群优化算法的 XQuery 查询路径优化方法中的步骤 S3 的流程图；

[0042] 图 4 是本发明的实施方式的基于粒子群优化算法的 XQuery 查询路径优化方法中的步骤 S4 的流程图。

具体实施方式

[0043] 为使本发明的上述目的、特征和优点能够更为明显易懂，下面结合附图对本发明的具体实施方式做详细的说明。在以下描述中阐述了具体细节以便于充分理解本发明。但是本发明能够以多种不同于在此描述的其它方式来实施，本领域技术人员可以在不违背本发明内涵的情况下做类似推广。因此本发明不受下面公开的具体实施方式的限制。

[0044] 在对大 XML 文件进行复杂查询时，由于 XML 文件较大，且其执行的 XQuery 查询语句较为复杂，因此会导致了查询过程中所出现的中间结果较为庞大，从而影响到查询速度，且系统在进行复杂查询时的系统资源消耗过大，从而产生用户体验不佳的问题。

[0045] 故而，在对大型 XML 文件进行复杂的 XQuery 查询时候，首要考虑的因素是如何减少查询所产生的中间结果，而减少查询中间结果又涉及查询路径的编码以及所采用的优化算法。在此，我们主要关注两点，即如何对 XQuery 查询语句进行有效的编码，并产生出相对应的查询计划，其次是在对 XQuery 查询语句进行编码后，采取何种有效的算法来评估各个查询计划的代价，为执行器选择最优的查询计划提供支持。

[0046] 现有对于大型的 XML 文档进行复杂的查询时候，采用的朴素的优化方式进行，其中只涉及一些简单的等价处理，如 :Not, And, Or 等的等价变换。而且，现有技术对于查询计划没有进行相应的优化，只是简单的按照 XQuery 语句来生成相应的查询计划。

[0047] 例如，对于如下的 XML 文档：

[0048]

```
<books>

  <book id="book1">

    <title> Discovery in deep ocean.</title>

    <price>12.5</price>

    <author>Brad.Pitt</author>

  </book>

  <book id="book2">

    <title>anonymous</title>

    <price>21.4</price>

    <authro>annoy</author>

  </book>

</books>
```

- [0049] 我们使用如下的 XQuery 语句来查询出 books 下所有价格大于 12 美元的书：
- [0050] for\$x in doc(“books.xml”)/books/book where\$x/price>12return\$x;
- [0051] 则其执行的结果为：
- [0052]

```
<book id="book1">

  <title> Discovery in deep ocean.</title>

  <price>12.5</price>

  <author>Brad.Pitt</author>

</book>
```

- [0053] 现行的查询编译器对于上述查询语句的处理,按如下的步骤进行：
- [0054] 1、首先,会生产整个文档扫描的查询计划(Query Plan)。
- [0055] 2、进行查询条件的简单等价变换操作。
- [0056] 3、生产 where 条件的查询计划。
- [0057] 4、返回查询结果。
- [0058] 一个典型的查询计划如下：
- [0059]


```

<XQuery>
  <Return>
    <OrderByTuple direction="ascending" empty="least" stable="false" >
      <WhereTuple>
        ...
      <EffectiveBooleanValue>
        <ValueFilterQP comparison="eq" general="true">
          <StepQP axis="self" name="year" nodeType="element">
            <VariableQP name="book"/>
[0060]
            </StepQP>
          </ValueFilterQP>
        </EffectiveBooleanValue>
      </WhereTuple>
      ...
    </OrderByTuple>
    <VariableQP name="i"/>
  </Return>
</XQuery>

```

[0061] 由此可以看出，XQuery 查询类似于结构化的查询方法，以关系代数为基础对查询语句进行等价的变换。例如，通过交换投影和选择的执行顺序对查询树进行变换，从而减少查询的中间代价。另外，由于 XML 文档的非结构化特性，使得适用于结构化查询的某些代数变换无法应用到 XML 文档上。

[0062] 为解决现有技术中存在上述的技术问题，发明人经过研究，提出了一种基于粒子群优化算法的 XQuery 查询路径优化方法。参阅图 1，图 1 是本发明实施方式的基于粒子群优化算法的 XQuery 查询路径优化方法的流程图。所述基于粒子群优化算法的 XQuery 查询路径优化方法包括：

[0063] S1：读取预查询的 XML 文档，并对所述 XML 文档进行预处理，以简化所述 XML 文档；

[0064] 参见图 2、图 2a、图 3，它们示出了本发明的基于粒子群优化算法的 XQuery 查询路径优化方法的一种优选的实施方式。

[0065] 为了使系统能够减少在执行文档查询时所产生的中间冗余信息，同时也为了能够方便的对于查询语句的查询路径进行相应的编码，以便构造出一个适用于基于粒子群优化算法的查询路径编码。首先，要对整个要查询的 XML 文档进行预处理，使得在查询系统中对 XML 文档中的所有节点均给出一个数值 ID 来描述该节点并对该节点的属性及内容做同样的处理。

[0066] 以上面给出的 XML 文档为例，首先根据文档中的所有元素节点构造相应的元素容器，下表表示上述的 XML 文档的元素容器及属性容器和数据容器的情况。

[0067] 元素表

1	/books
2	/books/book
...	..
5	/books/book/a uthor

[0069] 属性表

1	/books/book/ @id
---	---------------------

[0071] /books/book/title

Discovery in deep ocean.
Anonymous

[0073] 在上述的元素容器构造完成的基础上，对于上述的 XML 文档作进一步的处理。

[0074] S2：根据预处理后得到的所述 XML 文档转换 XQuery 查询语句；

[0075] 参见图 2 所示，图 2 示出了本发明的实施方式的基于粒子群优化算法的 XQuery 查询路径优化方法中的步骤 S1 的流程图。所述步骤 S1 具体包括：

[0076] S11：对 XML 文档的所有节点标签进行预处理；

[0077] S12：对 XML 文档中的冗余标签进行处理。

[0078] 参见图 2a 所示，图 2a 示出了本发明的实施方式的基于粒子群优化算法的 XQuery 查询路径优化方法中的步骤 S11 的流程图。步骤 S11 的具体实现方式为：

[0079] S111：寻找出所述 XML 文档中所有节点标签信息，并将所述节点标签信息（例如：节点名称、节点属性值等信息）保存至一张节点数据表中，并对所述节点数据表中的每一项

赋予唯一编号作为该项在表中的索引值；

[0080] S112：在所述节点数据表的建立完成后，使用各个节点所对应的索引号对所述 XML 文档进行相应的处理，将所述节点在文档中的出现的位置使用其对应的节点索引号进行代替。

[0081] 例如，通过上述步骤处理后，对于前述的 XML 文档中出现的一对标签 <Name>Brad. Pitt</Name>，可将其简化成 <1>Brad. Pitt</1> 例如：对于出现的一对标签 <Name>Brad. Pitt</Name>，可将其简化成 <1>Brad. Pitt</1>，并可进一步简化为 <1>1001</1>，其中 1 为节点名称的变化，1001 为节点内容的编号。因而，可以得知，通过上述步骤的预处理后，可以在很大程度上减少 XML 文档中的冗余信息。

[0082] 优选地，步骤 S12 的具体实现方式为：删除所述 XML 文档的所有节点的右标签。因为，发明人经过研究发现：节点的右标签属于冗余信息，其标识了某节点的结束位置，而该位置我们可以从下一节点的开始标签获得其上一节点的右标签。删除节点的右标签后，相应的 XML 文档并没减少相关结构的信息，而 XML 文档却进一步得到了简化。

[0083] 优选地，步骤 S2 具体为：根据所述的预处理后得到的 XML 文档转换原有的 XQuery 查询语句，也即使用所述 XML 文档中的各个节点的节点 ID 表示所述节点，从而将原 XQuery 语句进行转换；

[0084] 例如：

[0085] 上述的查询语句：

[0086] `for$x in doc(“books.xml”)/books/book where$x/price>12return$x;`

[0087] 通过上述的处理，可简化为：

[0088] `for$9in doc(“1023”)/1/2where$x/4>12return$9;`

[0089] 其中，9 为字符 x 在系统中的编号，1023 则是系统为 book.xml 所分配的系统编号，1, 2, 4 分别为 books, book, price 在系统中的编号。同时，将 XQuery 查询语句的关键字使用特殊编码后（例如用 F 代表 for 操作，D 代码 doc 操作，等等），则简化后的查询语句进一步转换为：

[0090] `F|9|I|1023/1/2|W|9/4|R|9|。`

[0091] S3：根据所述转换后得到的 XQuery 查询语句构造查询代价矩阵；

[0092] 参见图 3，图 3 示出了本发明的实施方式的基于粒子群优化算法的 XQuery 查询路径优化方法中的步骤 S3 的流程图。步骤 S3 具体包括：

[0093] S31：获取转换后得到的所述 XQuery 查询语句中的节点编号：MAX_ID 和 MIN_ID；其中，所述 MAX_ID、所述 MIN_ID 分别表示所述 XQuery 查询语句中的节点编号的最大值和最小值；

[0094] 例如：

[0095] 上述的查询语句：

[0096] `doc(“books.xml”)/books/book where$x/price>12return$x;`

[0097] 则可以得到：MAX_ID 的值为 9，MIN_ID 的值为 1。

[0098] S32：根据 MAX_ID, MIN_ID 的取值范围建立一个 (MAX_ID - MIN_ID)*(MAX_ID - MIN_ID) 的查询代价矩阵（查询的路径必然在该范围内），并将所述查询代价矩阵中的各个元素值设置为一个节点到另一个节点所包含的节点数量，以表示所述两个节点之间的查询

路径的权值；其中，查询路径的权值越大，则说明在查询过程中所需要处理的节点元素个数越多，在进行选择最优查询路径的过程中，对该矩阵元素值进行相应的更新，新的元素值为该条路径上的节点数量。如果两个节点之间不存在有效的线路，则这两个节点之间的权值赋予 $+\infty$ 。

[0099] S4 :运用粒子群优化算法对所述查询矩阵进行编码，并计算出各个节点到其它节点的路径，并选择一条最短路径，从而使得执行所产生的中间结果最少。

[0100] 参阅图 4，图 4 示出了本发明的实施方式的基于粒子群优化算法的 XQuery 查询路径优化方法中的步骤 S4 的流程图。步骤 S4 具体包括：

[0101] 步骤 S41 :定义粒子群优化算法的目标函数为 $\text{min}:f(x_1, x_2, \dots, x_n)$ ；

[0102] 步骤 S42 :随机生成 N 个个体，以生成初始种群；

[0103] 步骤 S43 :初始化 N 个个体的初始值，也即使用随机数生成器对所述 N 个个体的初始速度及位置生成初始速度和初始位置；

[0104] 步骤 S44 :计算所述各个个体的适应度值；

[0105] 步骤 S45 :将适应度值与给定的阈值 d 进行比较，若小于所述的阈值 d，则终止计算；

[0106] 步骤 S46 :输出最优值，即计算得出的最短查询路径，否则，则

[0107] 步骤 S47 :更新粒子个体。

[0108] 其中， $f(x_1, x_2, \dots, x_n)$ 为 x_1, x_2, \dots, x_n 路径上的查询代价，min 为去所述目标函数的最小值；

[0109] d 为用户给定的阈值，由用户进行指定；

[0110] 适应度值，用以表明个体的优秀度，其越小则表明所述个体越优秀。

[0111] 优选地，按照下述的公式对粒子个体进行更新：

$$v_i^{k+1} = w v_i^k + c_1 \text{rand}() (l_i^k - x_i^k) + c_2 \text{Rand}() (g^k - x_i^k) \quad (1)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (2)$$

[0114] 其中， v_i^{k+1} 中的 x_i^{k+1} 等参数为所述的各个种群的所在搜索空间中第 k+1 次迭代后的位置和速度； g^k 为在搜索空间中的优值， l_i^k 为第 i 个粒子在第 k 次迭代过程中所经历的最优位置；w 为算法惯性权值，此数值为经验值，通常在 [1, 1.3]； c_1 、 c_2 为加速常数，通常取值为 2；rand()、Rand() 为两个在 [0, 1] 范围内变化的均匀分布函数。

[0115] 综上所述，本发明技术方案具有下优点：

[0116] 本发明的基于粒子群优化算法的 XQuery 查询路径优化方法，具有提供相应查询相应时间，减少查询是系统所需要的资源，较好的利用查询中间结果并对中间结果进行有效的优化处理，从而减少中间结果所消耗的系统资源，从而以较低的资源需求来完成相应的查询工作。

[0117] 应当理解的是，这里所描述的方法可以以各种形式的硬件、软件、固件、专用处理机或者它们的组合实现。尤其是，至少本发明的一部分包括程序指令的应用程序优选实现。这些程序指令被确实地包括在一个或者多个程序存储设备（包括但不限于硬盘，磁性软盘，RAM，ROM，CD，ROM 等）里，并且可由任何包括适当结构的设备或者机器，例如一种具有处理器、内存和输入 / 输出接口的通用数字计算机执行。还应当理解由于附图中描述的一些系统的组成部件和处理步骤优选地以软件实现，所以，系统模块（或者方法步骤的逻辑流程）

之间的连接可能不同,这取决于本发明的编程方式。根据这里给出的指导,相关领域的普通技术人员将能够设计出本发明的这些以及类似的实施方式。

[0118] 以上公开了本发明的多个方面和实施方式,本领域的技术人员会明白本发明的其它方面和实施方式。本发明中公开的多个方面和实施方式只是用于举例说明,并非是对本发明的限定,本发明的真正保护范围和精神应当以权利要求书为准。

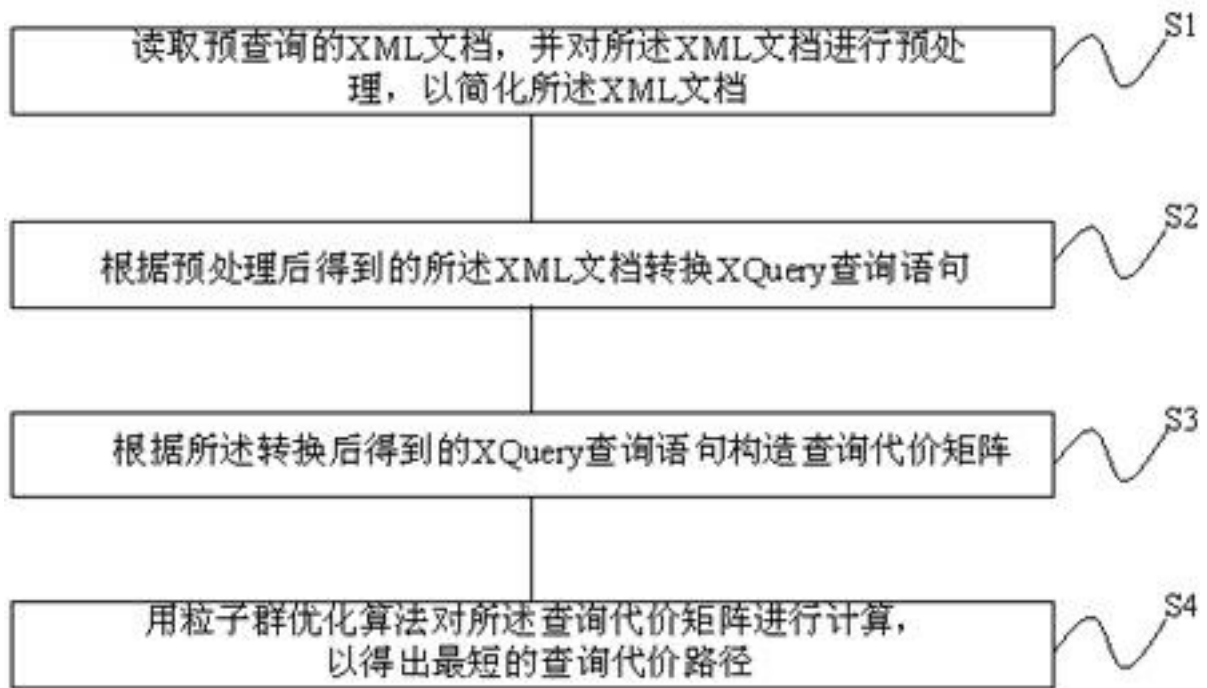


图 1

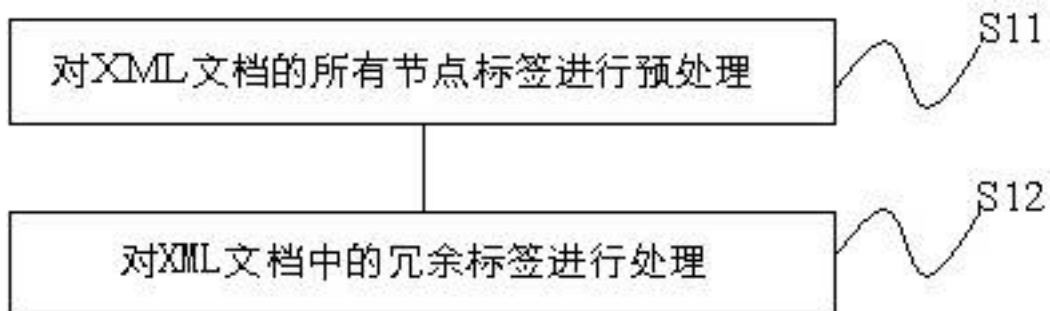


图 2

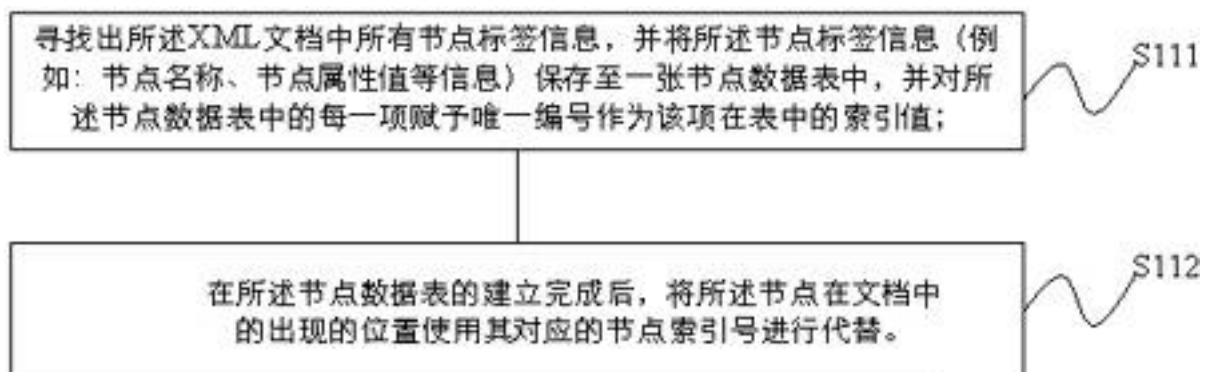


图 2a

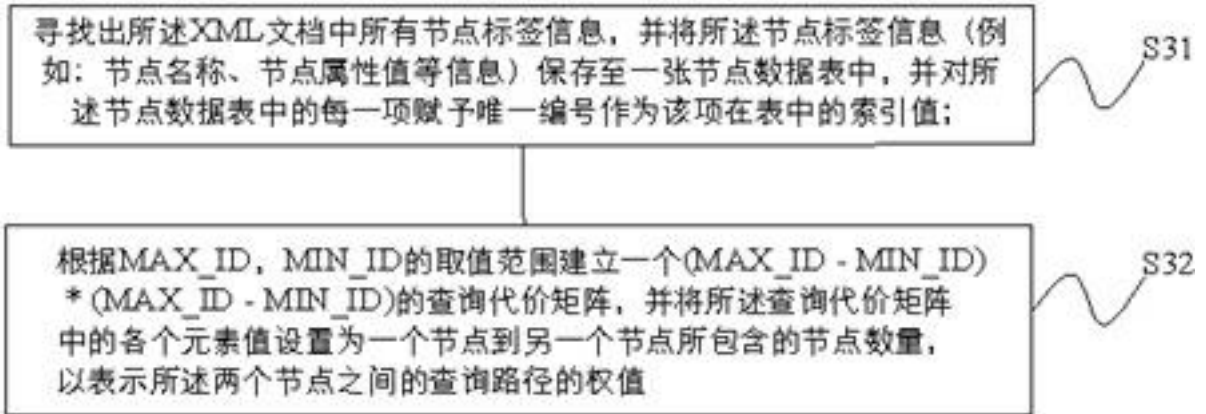


图 3



图 4