

文章编号: 1006-4710(2014)02-0180-07

晶体在立方体晶格移动的最短路径问题研究

张娜¹, 王家民¹, 陈小军², 窦忠发³

(1. 西安理工大学 机械与精密仪器工程学院, 陕西 西安 710048;

2. 陕西省地方电力(集团)有限公司, 陕西 西安 710061; 3. 中航工业 618 所, 陕西 西安 710065)

摘要: 通过模型转换, 可使立方体晶格中晶体移动模型转变为一个包含多个移动体的拓扑图, 在明确多个约束条件的基础上, 求解最短路径问题可用典型最短路径算法来解决。本文在模型构建的基础上, 提出了基于魔方结构的立方体晶格中晶体移动的多源多路最短路径算法(CMCLA), 该算法通过对每次移动步数最小的晶体的挑选和对晶格状态的动态更新, 使全体晶体从初始状态逼向最终状态。实验结果表明, 本文算法可行且有效。

关键词: 立方体晶格; 晶体; 魔方结构; 最短路径; 模型转换

中图分类号: TP391 **文献标志码:** A

Research on shortest path problem for crystal mobile model in cube lattice

ZHANG Na¹, WANG Jiamin¹, CHEN Xiaojun², DOU Zhongfa³

(1. Faculty of Mechanical and Precision Instrument Engineering, Xi'an University of Technology,

Xi'an 710048, China; 2. Shaanxi Regional Electric Power Group Co Ltd, Xi'an 710061, China;

3. Avic Xi'an Flight Automatic Control Research Institute, Xi'an 710065, China)

Abstract: The model transfer can make the crystal mobile model in cube lattice transfer into a topological graph including many moving bodies. On the basis of understanding of several constraint conditions, such shortest path problem is transferred into the typical shortest path algorithm to solve. Depending on the model construction, a shortest path algorithm (CMCLA) with multiple sources and paths based on the magic square for crystal mobile model in cube lattice is suggested. In this algorithm, the crystal with the least steps is to select each moving process and the state of cube lattice is updated after a moving process, thus, the crystals would tend to be compelled to final state from initial state. The experiment results indicate that the algorithm in this paper is feasible and efficient.

Key words: cube lattice; crystal; magic square; shortest path; model transfer

很多场景都涉及晶体在立方体晶格中的移动问题。例如, 现代物流业的立体仓库, 立体仓库可以看成是一个立方体晶格, 其中存放货物的货架相当于晶格中的晶体^[1]。当满载货物的立体仓库中的各个货架上的物品需要调整时, 通过货位移动, 完成位置调整, 这个过程可以抽象为晶体在立方体晶格中的移动。为了降低成本并提高工作效率, 晶体在立方体晶格中的移动往往通过最短路径来实现^[2]。

很多空间最小值问题经过数学模型转换, 都可以转化成一个最短路径问题^[3-7]。最短路径问题最初是为解决交通运输而提出的。随着我国国民经济

的快速发展, 除交通运输领域, 物流仓储、工业生产等基础领域也出现了最短路径问题。最短路径问题与社会经济生活的联系越来越紧密。求解最短路径的算法主要有 Dijkstra 算法、A* 算法、D* 算法、Floyd-Warshall 算法、Bellman-Ford 算法、SPFA 算法等^[8-16]。

晶体在晶格中移动的最短路径求解实际上是一个多源多路径的动态环境问题, 改进型的 D* 算法和 Floyd-Warshall 算法可以求解, 但这两种算法设计较为复杂, 且效率不高^[17-19]。本文在求解晶体在晶格中移动时, 将晶格抽象成魔方结构, 通过模型转

收稿日期: 2014-01-13

基金项目: 陕西省自然科学基金资助项目(2011JM7005)。

作者简介: 张娜, 女, 博士生, 研究方向为计算机辅助设计。E-mail: zhangna830204@163.com。

王家民, 男, 教授, 博导, 研究方向为计算机辅助与产品造型设计。E-mail: wjm3550@xaut.edu.cn。

换,将魔方结构变成一个包含多个移动体的拓扑结构图,该最短路径问题可转化为典型最短路径算法来解决,算法在单路径寻找时以 Dijkstra 算法为基础,加入当前最佳移动晶体的选择功能及晶格结构变化调整的操作,来完成最短多源多路径选择。该算法的贡献在于通过对每次移动步数最小的晶体挑选和对晶格状态的动态更新,使全体晶体从初始状态逼向最终状态为总体步数最小。

1 晶体在晶格中移动图模型的构建

1.1 晶体移动的规则

晶体在晶格中移动模型可以用图 1 所示的魔方结构来描述,利用各类颜色代表不同元素方位的类别,晶体在晶格中移动的过程就是使不同颜色的单元都归在魔方的 6 个方向的过程。

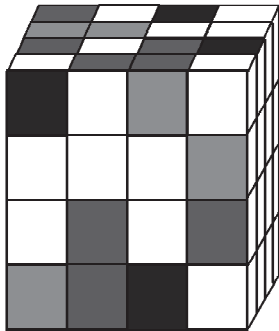


图 1 基于魔方结构的晶格

Fig. 1 Based cube lattice structure

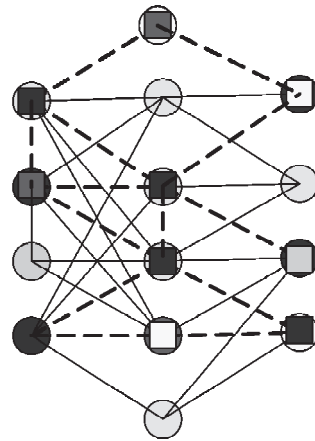
与魔方结构单元的运动方式不同的是,魔方结构的单元只能在单面将所有单元整体移动,而在晶格模型中,只要所有路径的晶格不存在障碍晶格,每个晶体都可以在晶格中向其他相邻晶格自由移动。由于障碍晶格直接影响了路径选择,而晶格中的每个晶体在移动过程中又会对其他晶体的移动产生影响,故每个移动晶体的起点和终点均互相交叉,晶格所走的路径将是动态变化的,而不是固定存在的路径。由此可见,求晶体在晶格中移动的最短路径问题必须进行模型转换,将动态的路径选择转化为相对静止的路径选择,将多源晶体的移动转为多个单一晶体的相对移动。

1.2 晶体移动的图模型

从图拓扑结构的角度看,晶格的信息可以转化为顶点的信息和描述顶点之间关系的信息。将晶格中的每个晶体所在的位置理解为拓扑图结构中的一个顶点,晶体所在的位置与周围晶体所在位置的移动关系理解为顶点之间连接关系的信息。当相邻晶体位置不存在障碍晶格时,每个晶体与周围的晶体

存在连接关系,而当存在障碍晶格时,则不存在连接关系。由此,在多个晶格都处于运动时,每个晶体与周围的晶体的连接关系是动态的。当晶体在图结构上运动后,需要立即更新顶点之间关系的信息。

典型的最短路径算法,主要用于计算一个节点到其他所有节点的最短路径,以起始点为中心向外层层扩展,直到扩展到终点为止。与单一物体在图结构上从起点经过连续路径到终点的移动轨迹不同,晶格中所有的晶体都需要经过从起点到终点的路径,每次移动的可以是当前任何一个不存在任何障碍的晶体。由此可知,最短路径算法不仅仅要考虑选择当前晶体走哪条路径,而且要考虑先走哪个晶体的问题。构造的模型如图 2 所示。



- 一个不存在晶体的顶点
- 蓝色晶体终点的顶点
- 黄色晶体终点的顶点
- 红色晶体终点的顶点
- 绿色晶体终点的顶点
- 代表当前路径可走
- - 代表当前路径不可走
- 蓝色晶体
- 黄色晶体
- 红色晶体
- 绿色晶体

图 2 晶格的网络拓扑图

Fig. 2 Lattice network topology

图 2 的模型用不同的颜色代表不同的晶体类型,所有晶体最终要运动到颜色一致的顶点,例如,“■”晶体最终要到达“●”顶点,其它颜色的晶体类似。

2 算法设计

2.1 数据结构

对于一个有 M 个晶体、 N 个位置的晶格,对其数据结构描述如下。

1) 晶格六面体的方向描述。可将六面体的元

素方位分为 $U_i[6] = \{U_1, U_2, U_3, U_4, U_5, U_6\}$ 。每个方位都可以用图 2 模型中的不同颜色来表示。

2) 顶点的描述。顶点是一个数组 $V_i[N] = \{V_1, V_2, \dots, V_N\}$, 其中的元素 V_i 具有属性 U , 代表所属方位, $V_i, U \in \{U_1, U_2, U_3, U_4, U_5, U_6\}$, 即顶点数组中的元素都可以归类到其中一个方位。

3) 边之间关系描述。任意两个顶点之间存在路径即为边, 边之间的关系实际上是一个 $N \times (N-1)$ 的二维矩阵的计算机存储表示问题, 边是一个二维数组 $C_s[N][N-1] = \{\{V_1, V_2\}, \{V_1, V_3\}, \dots, \{V_N, V_{N-1}\}\}$ 。二维数组 $C_s[N][N-1]$ 的取值分二种情况: ①由于每个顶点只可能和周围的最多 6 个顶点成为邻接顶点, 当某顶点与另一顶点不是邻接顶点时, $C_s[N][N-1]$ 的取值为 0; ②当某顶点与另一顶点是邻接顶点, 且二者当前的连接关系为“—”时, $C_s[N][N-1]$ 的取值为 1; 二者当前的连接关系为“- -”时, $C_s[N][N-1]$ 的取值为 0。

4) 晶体描述。晶体集合是一个数组 $S[M] = \{S_1, S_2, \dots, S_M\}$, 其中的元素 S_i 具有属性 U , 代表所属方位, 取值为 $S_i, U \in \{U_1, U_2, U_3, U_4, U_5, U_6\}$, 即晶体集合数组中的元素都可以归类到其中一个方位。

5) 晶体位置状态描述。令晶体位置是一个二维数组 $L[M][M]$, 其元素为 $\langle v, s \rangle$, v 代表顶点的信息, s 代表顶点上的晶体信息。例如, 某一个晶体位置状态可以描述为 $L[M][M] = \{\{V_1, S_1\}, \{V_2, S_2\}, \dots, \{V_M, S_M\}\}$, 代表 M 个晶体所在的位置。当对所有的晶体元素, 都满足 $S_i, U = V_i, U$ 条件时, 该状态就是晶体的终止位置状态, 记为 E 。晶体的终止位置状态是最短路径算法终止的条件。晶体的起始位置状态记为 B , 在该状态至少有一个晶体不满足 $S_i, U = V_i, U$ 。

6) 当前位置到源点路径的描述。晶格中每个晶体都要完成从起始顶点到终止顶点的运动路径, 对于一个具有 M 个晶体 ($M < N$) 的晶格, 共有 M 条路径。用二维数组 $dist[M][maxnum]$ 来保存当前状态下所有晶体所在的位置到起始顶点的最短路径长度, 其中 $dist[i]$ 代表 S_i 晶体所在的位置到起始顶点的最短路径长度; 用二维数组 $prev[M][maxnum]$ 来保存当前状态的所有晶体所走过的顶点信息, 其中, $prev[i][maxnum]$ 代表 S_i 晶体走过的顶点集合, $maxnum$ 代表前一个结点的序号, 该值在 S_i 移动过程中处于递增的状态。

7) 晶体在晶格中的移动路径描述: 晶体在晶格中移动的路径集合是一个数组 $pathNodes[]$, 数组

中的元素为 $\langle s, v \rangle$, s 代表每一步所移动的晶体信息, v 代表晶体所移动到的顶点信息。

2.2 算法描述

在此提出一种晶体在晶格中移动的最短路径算法(CMCLA), 其伪代码描述如图 3 所示。

```

输入: 晶体的起始位置状态 B、晶体的终止位置状态 E。
输出: 晶体的最短运动路径 pathNodes 和最短路径值 sp,
全局变量: (S[], Cs[N][N-1], L[M][M], dist[M][maxnum],
           pathNodes[]);
(step1: 依据 B 初始化状态集合 prev[M][maxnum], 每个
prev[M] 仅含有对应晶体的起始位置状态;
step2: 挑选当前需移动的晶体
CurrentCrystal ← SelectCurrentCrystal(S[], prev[M][maxnum]);
step3: 查找当前要移动到的位置
v ← searchPath(CurrentCrystal, prev[M][maxnum]);
step4: 添加 <CurrentCrystal, v> 进入 pathNodes[];
step5: 更新 Cs[N][N-1] 和 L[M][M];
step6: 判断所有的晶体元素都有 S_i, U = V_i, U (即晶体的终止
位置状态 E) goto step 5, 否则 goto step2;
step7: 计算最短路径
sp ← computeShortestPath(dist[M][maxnum]);
step8: 输出 pathNodes 及最短路径 sp;
}
//定义挑选当前需移动的晶体的函数
Function SelectCurrentCrystal(S[], prev[M][maxnum])
{ 通过 Cs[N][N-1] 和 L[M][M] 确定 S[] 中的可移动晶体,
  并列入集合 s[m];
  令 maxstep = 1, CurrentCrystal = s[1];
  For(i = 0; i < M; i++)
  { 从 prev[i][maxnum] 中计算晶体的移动步数为 step;
    If(step < maxstep) { maxstep = step; CurrentCrystal = s[i]; }
  }
  Return CurrentCrystal;
}
//定义查找当前要移动到的位置的函数
Function searchPath(currentCrystal, prev[M][maxnum])
{ 依据 currentCrystal, 从 prev[M][maxnum] 确定对应晶体序号 i;
  从 Vs-prev[i][j] 中取出与 currentCrystal 具有最短路径长度
  度顶点 v, 作为本次移动的目标;
  将 v 添加到 prev[i][j] 中, 更新数组 dist[i][j];
  Return v;
}
//定义计算最短路径的函数
Function computeShortestPath(dist[M][maxnum])
{ shortestPath = 0;
  For(i = 1; i < M; i++) { For(j = 1; j < N; j++) {
    shortestPath += dist[i][j];
  } }
}

```

图 3 CMCLA 算法描述

Fig. 3 CMCLA algorithm description

CMCLA 算法的基本思想为: 设置顶点集合 $prev[M][maxnum]$ 并不断地作贪心选择来扩充这个集合, 最终求出最短路径。其中, 当前晶体集合 $S[M]$ 的每一个晶体都要完成从起始位置状态到晶体终止位置状态的路径移动。

例如在晶体 $S[1]$ 的路径移动中, 一个顶点属于

集合 $prev[1][j]$, 当且仅当从源到该顶点的最短路径长度已知。初始时, $prev[1][j]$ 中仅含有 $S[1]$ 所在的起始位置状态。设 v 是晶格中的某一个顶点, 把从起始顶点到 v 且中间只经过 $prev[1][j]$ 中顶点的路径称为从源到 d 的特殊路径, 并用数组 $dist[i][j]$ 记录当前每个顶点所对应的最短特殊路径长度。每次从 $V-prev[1][j]$ 中取出具有最短特殊路径长度的顶点 v , 将 v 添加到 $prev[1][j]$ 中, 同时对数组 $dist[1][j]$ 作必要的修改。该过程持续进行直到满足全局条件: 当且仅当所有的晶体元素都有 $S_i \cdot U = V_i \cdot U$ 。其他晶体的移动与 $S[1]$ 的移动的处理完全一致。

CMCLA 算法通过每次移动步数最小的晶体挑选和晶格状态的动态更新, 使全体晶体从初始状态通向最终状态。对于一个 M 个晶体、 N 个位置的晶格, 令 d 为采用 d 路堆实现优先队列 ADT, 函数 $SelectCurrentCrystal$ 的复杂度为 $(M * N)^2$, $searchPath$ 的时间复杂度为 $M * \lg N$, $computeShortestPath$ 的时间复杂度为 $M * N$, 主函数的时间复杂度为 $N * M$, 在最坏的条件下 CMCLA 算法的时间复杂度是 $O(N^3 M^4 * \lg N * \lg d(M))$ 。CMCLA 算法中变量 $prev[M][maxnum]$ 、 $dist[M][maxnum]$ 、 $Cs[N][N-1]$ 使用的空间最大为 $O(N * N)$, 其他局部变量使用的空间 $O(M * N)$, 由此 CMCLA 算法的空间复杂度是 $O(N^2)$ 。

CMCLA 算法通过每次对移动步数最小的晶体的挑选和对晶格状态的动态更新, 使全体晶体从初始状态通向最终状态。算法中有以下两点与典型的最短路径算法不一样。

1) M 个晶体的同时移动, 每个时刻只能选择一个晶体移动, 处理时首先挑选 $S[M]$ 中可移动的晶体进入集合 $s[m]$, 然后从 $prev[M][maxnum]$ 中选择 $s[m]$ 中移动步数最少的晶体命名为 $currentCrystal$, 作为当前移动的晶体。

2) 每次从 $V-prev[i][j]$ 中取出具有最短路径长度的顶点 u 后, 将 u 添加到 $prev[i][j]$ 中, 在对数组 $dist[i][j]$ 作必要修改的同时, 更新晶体位置状态 $L[M][M]$ 和边之间的关系 $Cs[N][N-1]$ (即当相关顶点当前的连接关系为“—”时, $Cs[N][N-1]$ 的取值为 1; 相关顶点当前的连接关系为“- -”时, $Cs[N][N-1]$ 的取值为 0)。

3 实验研究

通过仿真实验验证晶体在晶格中移动的最短路径算法(CMCLA)。基于图 1 所示魔方结构, 在进

行模型转换后, 利用 C 语言实现上述算法。实验数据分成两组。

1) 第一组: 采用立方体晶格边分别为 4(顶点数为 64)、5(顶点数为 125)、6(顶点数为 216) 和 7(顶点数为 343) 的四种晶格为基础, 晶体的数量为 18 (最终状态是六面体的每个方位都有 3 个晶体), 起始状态为晶体在其晶格六面体每个方位的反面。

2) 第二组: 以边为 7(顶点数为 343) 的晶格为基础, 晶体的数量分别为 18(最终状态是六面体的每个方位都有 3 个晶体)、24(最终状态是六面体的每个方位都有 4 个晶体)、30(最终状态是六面体的每个方位都有 5 个晶体)、36(最终状态是六面体的每个方位都有 6 个晶体) 四种情况, 起始状态为晶体在其晶格六面体每个方位的反面。

算法在 2.8 GHZ Intel core2 CPU、带有 PAE 的 2G 内存和 160G 硬盘的 PC 机上运行, 经过运行, 所有设定的晶格都从初始状态到达了最终状态, 算法的运行结果如图 4 和图 5 所示。

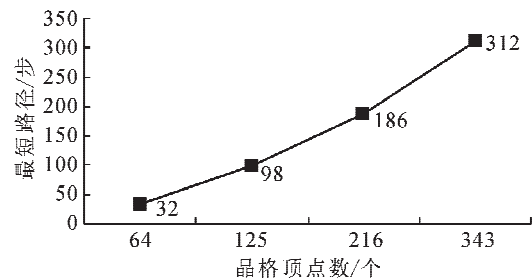


图 4 最短路径与立方体晶格顶点数量之间的关系
Fig. 4 The shortest path between the lattice and the number of vertices of a cube

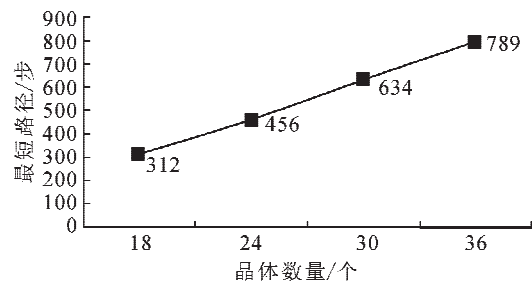


图 5 最短路径与晶体数量之间的关系
Fig. 5 The shortest path between the crystal and the number

图 4 是晶体数量为 18 时, 四组输入数据状态下的最短路径与立方体晶格顶点数量之间的关系。图 5 是立方体晶格顶点数量是 343 时, 四组输入数据状态下的最短路径与晶体数量之间的关系。从图 4 可以看出, 随着立方体晶格顶点数量的增加, 最短路径的长度也增加, 二者增长幅度大致相同。从图 5 可以看出, 随着晶体数量的增加, 最短路径的长度也

是增加的,但二者的增长幅度不一致,晶体数量增加的曲线较为平缓,但最短路径的长度呈快速增长的趋势。

对第二组运行数据,进一步分析晶格顶点数量是 343 时的四组数据中的晶体运行轨迹。四组数据从起点到终点的变化中,已完成晶体数量与移动步数之间的关系如图 6 所示。

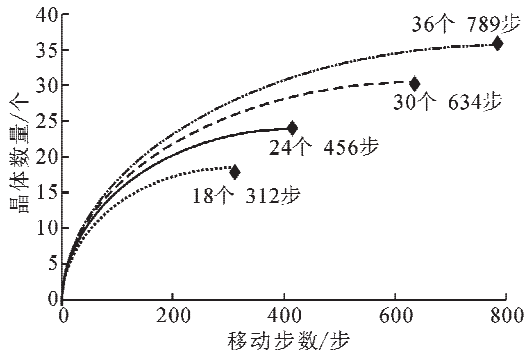


图 6 完成晶体数量与移动步数之间的关系
Fig. 6 The relationship between the finished crystal and moved steps

由图 6 可以看出,对于四组输入,算法总的趋势是随着时间进度的推移,完成晶体数量从 0 逐渐逼近了最终状态,分别是 312 步完成 18 个晶体移动,456 步完成 24 个晶体移动,634 步完成 30 个晶体移动,789 步完成 36 个晶体移动。但该过程不成线性关系,而是接近负指数函数关系,即前述少量时间完成了大部分晶体的移动,而后续大部分时间均用在少数晶体的移动。这是因为算法在后续存在路径迂回现象。算法未加入启发式规则,使得位置查找过程存在重复。这也说明了算法还有很大的改进余地。

CMCLA 是一个在动态环境中解决多源多路径问题的最短路径算法,在本文相关工作中列出的最短路径算法中 D-Star 和 Floyd-Warshall 算法也可以解决此类多源多路径问题。采用 C 语言实现 D-Star 和 Floyd-Warshall 算法,并将这两种算法及 CMCLA 算法以下列输入数据在同样的硬件机器上分别运行 4 次后,记录每次的运行时间,然后取平均值,其运行完成时间的如图 7 和 8 所示。

图 7 表明,在晶体相同的条件下,对于 CMCLA、D-Star 和 Floyd-Warshall 三种算法,随着晶格数据的增加,运行时间均呈现增加的趋势,而且晶格越多,增长的幅度越大。在晶体数量相同条件下,三种算法除了在晶格顶点数量为 64 时,D-Star 运行时间最短(这是因为 D-Star 算法在晶格数量比较少

时,启发式搜索使得收敛速度快),在其他状态下,CMCLA 算法的运行时间均比 D-Star 和 Floyd-Warshall 短。这说明晶格数量较少时,CMCLA 算法与 D-Star 算法相比无优势,而当晶格数量较多时,CMCLA 算法具有明显的优势。

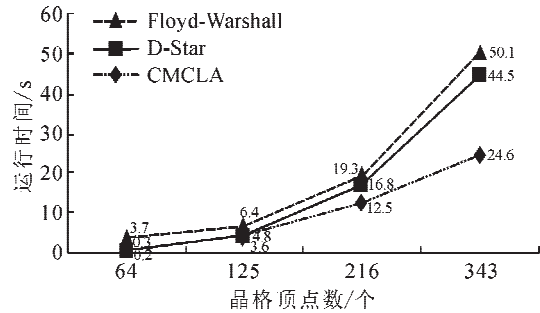


图 7 晶体相同、晶格不同条件下的运行时间比较
Fig. 7 The comparison of completing time in the same crystals and different lattices

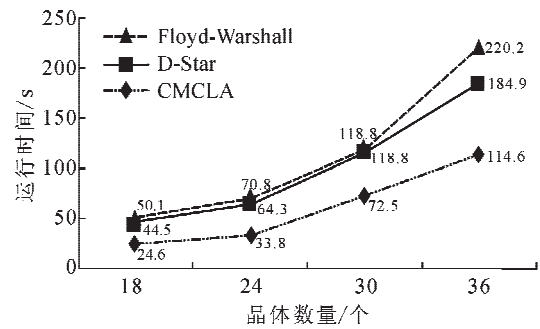


图 8 晶体不同、晶格相同条件下的运行时间比较
Fig. 8 The comparison of completing time in different crystals and the same lattices

从图 8 可看出,在晶格相同的条件下,对于 CMCLA、D-Star 和 Floyd-Warshall 三种算法,随着晶体数据的增加,运行时间均呈现增加的趋势。三种算法中,在四种状态下,CMCLA 运行时间均比 D-Star 和 Floyd-Warshall 短。

从上述实验可以看出,当晶体和晶格数量较少时,CMCLA 无法体现出在运行时间上的优势,而当晶格的规模达到较大数量时,特别是晶体数量也较多时,CMCLA 的运行效率将充分体现出来。这证明,在大规模晶体晶格结构中,CMCLA 算法能实现利用最低成本完成移动,提高工作效率、节约时间,具有重要的应用意义。

从空间复杂度方面分析,CMCLA 算法由于是在立方体模型的基础上实现的,并且算法设计思路是寻求立方体中多源多路径的最短路径,而 D-Star 和 Floyd-Warshall 算法主要针对平面图查找最短路径,因此,CMCLA 算法空间复杂度要大于 D-Star

和 Floyd-Warshall 算法空间复杂度。

从时间复杂度方面分析,由于在魔方结构的立方体晶格中需要进行多源多路径的最短路径查找,使得 D-Star 和 Floyd-Warshall 实现时,算法设计较为复杂,运行次数和运行时间受到多源多路径因素和算法设计自身的影响,造成算法时间复杂度的增加。而 CMCLA 算法自身考虑到晶体结构,将魔方结构变成一个包含多个移动体的拓扑结构图,通过对每次移动步数最小的晶体的挑选和路径晶格状态的动态更新,使全体晶体以总体步数最小从初始状态通向最终状态,从而节约了算法的执行时间。因此,当立方体晶格数量较少时,多源多路因素对三种算法的影响较小,三者的执行时间较为接近。而当立方体晶格数量较大时,CMCLA 算法的时间复杂度相对于 D-Star 和 Floyd-Warshall 算法的时间复杂度越来越小。

4 结 语

寻找立方体晶格中晶体移动的最短路径与典型的最短路径有重要区别,但通过模型转换,它就变成了一个包括多个约束条件的典型最短路径问题。本文在模型构建的基础上,分析并提出了基于魔方的立方体晶格中晶体移动的最短路径算法 CMCLA,算法的创新在于通过对每次移动步数最小的晶体挑选和对晶格状态的动态更新,使全体晶体以总体步数最小从初始状态通向最终状态。实验结果表明,算法能有效解决晶体在立方体晶格中的移动问题,能以最低成本完成移动,提高工作效率、节约时间。

在未来的工作中,笔者将致力于以下三个方面的工作:①优化现有算法,减少算法的空间复杂度;②引入智能计算思想和启发式规则,使算法在挑选当前需移动的晶体和查找当前要移动到的位置时依据规则,有目标地缩小挑选和查找范围,提高算法的效率;③将现有算法进行扩展和延伸,以解决多维数据的最短路径查找问题。

参考文献:

- [1] 卢香清,李洪安,康宝生,等. 图最短路径并行化及其应用研究[J]. 计算机工程与应用,2012,48(14):38-43.
Lu Xiangqing, Li Hongan, Kang Baosheng, et al. Research on parallelization of shortest path for graph and its application[J]. Computer Engineering and Applications, 2012, 48(14): 38-43.
- [2] 杨帆,杨晓光,云美萍. 考虑信号交叉口等待时间的最短路径算法[J]. 同济大学学报:自然科学版,2013,41(5):680-686.
Yang Fan, Yang Xiaoguang, Yun Meiping. Shortest path algorithm with a consideration of waiting time at signalized intersections[J]. Journal of Tongji University (Natural Science), 2013, 41(5): 680-686.
- [3] 王红梅,胡明. 基于直接/间接邻边概念的最短路径算法[J]. 计算机应用,2010,30(5):1297-1299,1303.
Wang Hongmei, Hu Ming. Shortest-path algorithm based on direct/indirect adjacent edge concept[J]. Journal of Computer Applications, 2010, 30(5): 1297-1299, 1303.
- [4] 王树西,吴政学. 改进的 Dijkstra 最短路径算法及其应用研究[J]. 计算机科学,2012,39(5):223-228.
Wang Shuxi, Wu Zhengxue. Improved dijkstra shortest path algorithm and its application[J]. Computer Science, 2012, 39(5): 223-228.
- [5] Thouti, Krishnahari, Sathe S R. Performance analysis of single source shortest path algorithm over multiple GPUs in a network of workstations using open CL and MPI[J]. International Journal of Computer Applications, 2013, 9(77): 31-36.
- [6] Jasika N, Alspahic N, Elma A, et al. Dijkstra's shortest path algorithm serial and parallel execution performance analysis[C]//Proceedings of the 35th International Convention, MIPRO, 2012:1811-1815.
- [7] Huang Yizhen, Yi Qingming, Shi Min. An improved dijkstra shortest path algorithm[C]//Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering(ICCSEE 2013), 2013:26-29.
- [8] 刘明周,赵志彪,凌先姣,等. 基于最短路径的复杂机械产品装配过程质量控制点公差带在线优化方法[J]. 机械工程学报,2012,48(10):173-177.
Liu Mingzhou, Zhao Zhibiao, Ling Xianjiao, et al. Research on online tolerance optimization for complex mechanical products assembly process based on shortest path[J]. Chinese Journal of Mechanical Engineering, 2012, 48(10): 173-177.
- [9] 徐涛,郑英,周念. 物流最短路径规划最优化计算方法研究[J]. 物流技术,2013,32(7):236-238,262.
Xu Tao, Zheng Ying, Zhou Nian. Study on optimal computation method of shortest logistics route[J]. Logistics Technology, 2013, 32(7): 236-238, 262.
- [10] Guerrero W J, Velasco N, Prodhon C, et al. On the generalized elementary shortest path problem: a heuristic approach[J]. Electronic Notes in Discrete Mathematics, 2013, 41(5): 503-510.
- [11] Alireza Nazemi, Farahnaz Omid. An efficient dynamic model for solving the shortest path problem [J]. Transportation Research Part C: Emerging Technologies, 2013, 26: 1-19.

- [12] Keyur Rana, Mukesh Zaveri. A-star algorithm for energy efficient routing in wireless sensor network[J]. Trends in Network and Communications, 2011, 197: 232-241.
- [13] Fangfang T, Shiyang Y, Xiaofeng Z, et al. Research on two-tiered A-star algorithm in game path-finding[J]. Microcomputer Applications, 2010, 1: 011.
- [14] Yao Junfeng, Lin Chao, Xie Xiaobiao, et al. Path planning for virtual human motion using improved A* star algorithm[C]//2010 Seventh International Conference on Information Technology: New Generations (IT-NG), 2010:12-14.
- [15] Deng Yong, Chen Yuxin, Zhang Yajuan, et al. Fuzzy dijkstra algorithm for shortest path problem under uncertain environment [J]. Applied Soft Computing, 2012, 12(3): 1231-1237.
- [16] 武苏里,王湘桃,张阳,等. 易阵游戏中两子交换算法的研究[J]. 计算机应用与软件, 2009, 26(2): 272-274.
- Wu Suli, Wang Xiangtao, Zhang Yang, et al. On two-piece exchange algorithm in Yizhen game[J]. Computer Applications and Software, 2009, 26(2): 272-274.
- [17] Garg H, Rawat P. An improved algorithm for finding all pair shortest path[J]. International Journal of Computer Applications, 2012, 47(25): 35-37.
- [18] 任小康,吴尚智,苟平章. 基于动态状态树的回溯算法[J]. 计算机工程与设计, 2007, 28(4): 755-756, 759. Ren Xiaokang, Wu Shangzhi, Gou Pingzhang. Backtracking algorithm of dynamic state space tree [J]. Computer Engineering and Design, 2007, 28(4): 755-756, 759.
- [19] 杨勇虎,李楠. 平面魔方游戏中算法的设计和优化[J]. 计算机应用与软件, 2010, 27(8): 273-275. Yang Yonghu, Li Nan. Designing and optimising algorithm for plane magic cube game[J]. Computer Applications and Software, 2010, 27(8): 273-275.

(责任编辑 王卫勋)

(上接第 174 页)

- [10] 孙悦,李栋梁,朱拥军. 渭河径流变化及其对气候变化与人类活动的响应研究进展[J]. 干旱气象, 2013, 31(2): 396-405. Sun yue, Li dongliang, Zhu yongjun. Advances in study about runoff variation of the Weihe River and its response to climate change and human activities[J]. Journal of Arid Meteorology, 2013, 31(2): 396-405.
- [11] 潘雅婧,王仰麟,彭建,等. 基于小波与 R/S 方法的汉江中下游流域降水量时间序列分析[J]. 地理研究, 2013, 31(5): 811-820. Pan yajing, Wang yanglin, Peng jian, et al. Precipitation change in middle and lower reaches of hanjiang river, based on wavelet analysis and R/S analysis[J]. Geographical Research, 2013, 31(5): 811-820.
- [12] Mandelbrot B B, Wallis J R. Computer experiments with fractional gaussian noises(part 1 and part 2) [J]. Water Resources Research, 1969, 5(1): 228-259.
- [13] 赵琛,党兴华. 基于分形理论的城市群最优空间结构模型与应用[J]. 西安理工大学学报, 2012, 28(2): 240-246. Zhao jing, Dang xinghua. An optimal spatial structure model of urban agglomeration and its application based on fractal theory [J]. Journal of Xi'an University of Technology 2012, 28(2): 240-246.
- [14] 徐绪松,马莉莉,陈彦斌. R/S 分析的理论基础: 分数布朗运动[J]. 武汉大学学报: 理学版, 2004, 50(5): 547-550. Xu Xusong, Ma Lili Chen Yanbin. Theoretical basis of R / S analysis; fractional brownian motion [J]. Wuhan University Journal (Natural Science Edition), 2004, 50(5): 547-550.
- [15] 马晓超,栗晓玲,薄永占. 渭河生态水文特征变化研究[J]. 水资源与水工程学报, 2011, 22(1): 16-21. Ma Xiaochao, Li Xiaoling, Bo Yongzhan. Research on eco-hydrological alternation of Weihe River basin [J]. Journal of Water Resources & Water Engineering, 2011, 22(1): 16-21.

(责任编辑 杨小丽)