

## 赋权超图划分问题的多水平迁移优化算法研究

冷明<sup>1,2</sup>, 孙凌宇<sup>1</sup>, 朱平<sup>1</sup>, 李金忠<sup>1</sup><sup>1</sup>(井冈山大学 计算机科学系, 江西 吉安 343009)<sup>2</sup>(加州大学 洛杉矶分校 电子工程系, 美国 加利福尼亚州 90095)

E-mail: lzylmsly@gmail.com

**摘要:** 伴随着赋权超图划分问题的规模不断地增长, 对赋权超图划分的处理效率和能力、准确性有了更高程度的要求. 本文采用形式化符号描述了赋权超图优化划分问题, 阐明了基于多水平思想求解该问题的粗化、初始划分和迁移优化三个阶段; 提出了基于离散粒子群和多水平思想的迁移优化算法, 在 $|V|$ 维空间中每个粒子所处的位置对应于超图划分方案; 伴随着投影优化阶段的细化, 每个粒子对应的划分方案映射到下一水平层的粗化超图; 随着划分问题解空间的维度 $|V|$ 增加, 粒子的自由度增大; 将结点迁移的收益值作为粒子的启发式搜索信息, 利用粒子之间聚集协同表现出的智能行为有效地搜索解空间; 给出了赋权超图的改进压缩存储格式, 提出了割切值和收益值的快速计算方法, 有效降低了划分算法的时空复杂度. 基于 ISPD98 测试基准的对比实验数据表明: 与基于迁移方法的赋权图、赋权超图优化划分算法相比, 本文算法有效地增强了迁移优化算法逃离局部最优的能力.

**关键词:** 赋权超图; 划分问题; 多水平方法; 优化算法

中图分类号: TP391

文献标识码: A

文章编号: 1000-1220(2016)06-1293-05

## Research on Multi-level Refinement Algorithm of Weighted Hypergraph Partitioning Problem

LENG Ming<sup>1,2</sup>, SUN Ling-yu<sup>1</sup>, ZHU Ping<sup>1</sup>, LI Jin-zhong<sup>1</sup><sup>1</sup>(Department of Computer Science, Jinggangshan University, Ji'an 343009, China)<sup>2</sup>(Department of Electrical Engineering, University of California, Los Angeles, CA, 90095, USA)

**Abstract:** With the scale of hypergraph partitioning problem continues to grow so far as to several millions of vertices, it leads to higher performance requirements for speed, accuracy and partitioning capability. The formal description of weighted hypergraph partitioning problem is presented. We describe the solution based on the multi-level method and propose the multi-level discrete particle swarm optimization refinement algorithm, which each particle's position in  $|V|$ -dimensional can be considered as the corresponded partitioning. During the refinement process of the uncoarsening phase, the algorithm project successively each particle's partitioning back to the next level finer hypergraph and the degree of particle's freedom increases with the increase of solution space's dimension. The algorithm also regards the gain of vertex as particle information for the heuristic search and successfully finds the solution spce based on the intelligent behavior between individuals' collaboration. Furthermore, the ICSF format of weighted hypergraph is presented and the two-dimensional auxiliary array is designed for counting the vertices of each hypergraph in different partitions. The rapid method of calculating the vertex's gain and the cut's size are proposed to avoid traversing each vertex of hypergraph and reduce the algorithm's space and time complexity. Experimental results show that the algorithm not only can find the better partitioning of weighted hypergraph than the move-based method, but also can improve the search capability of the refinement algorithm.

**Key words:** weighted hypergraph; partitioning algorithm; multilevel method; refinement algorithm

## 1 引言

经典的非线性结构图, 其结点之间是多对多的关系, 被广泛地应用在 VLSI 设计<sup>[1]</sup>、云计算任务调度等众多研究领域<sup>[2]</sup>. 以云计算任务调度中的任务优化划分为例, 其目的是将大量分散的任务按照特定的约束条件划分为多个规模较小的任务子集, 使划分后的各个任务子集计算规模相当, 且任务子集之间的依赖关系数目达到最少, 从而最大程度地实现云计算

平台的负载均衡和缩短整个任务完成的时间跨度<sup>[2]</sup>. 与元任务调度的数学模型—有向无环图相比, 赋权有向超图作为依赖任务的调度提供了更为精确的模型, 全面地表示云计算环境的异构性、分布性、广域性等特点, 从而提高任务调度的准确性和执行效率. 其中, 任务表示为赋权有向超图的结点, 任务结点间的先后依赖关系表示为赋权有向超图中的有向超边.

由于在合理的时间内不可能求解出给定赋权超图的最优划分, 导致迁移方法<sup>[5,6]</sup>、几何方法<sup>[7]</sup>、组合方法<sup>[8]</sup>、潜方

收稿日期: 2015-04-16 收修改稿日期: 2015-06-12 基金项目: 国家自然科学基金项目(61363014, 61163062)资助; 江西省青年科学家培养对象计划项目(20153BCB23003)资助; 江西省科技厅支撑项目(20132BBE50048)资助; 江西省自然科学基金项目(20132BAB201035)资助; 江西省教育厅科学技术研究项目(GJJ14561)资助. 作者简介: 冷明, 男, 1975年生, 博士, 副教授, 研究方向为算法分析与设计、组合分析与优化等; 孙凌宇(通信作者), 女, 1976年生, 硕士, 教授, 研究方向为云计算、任务调度优化等; 朱平, 男, 1955年生, 硕士, 教授, 研究方向为算法分析与设计; 李金忠, 男, 1976年生, 硕士, 副教授, 研究方向为组合分析与优化等.

法<sup>[9]</sup>、元胞自动机方法<sup>[10]</sup>等大量的启发式方法用于计算该问题的近似最优解。然而,随着划分问题的结点数规模也不断地增长,甚至达到数百万个,对赋权超图划分技术的处理效率和能力、准确性等提出了进一步的要求。

基于多水平思想的赋权超图划分方法包含粗化、初始划分和迁移优化三个阶段<sup>[11-13]</sup>,其实现步骤如图1所示。在粗化阶段通过合并结点得到粗化赋权超图,重复合并的过程直至得到最小赋权超图;在初始划分阶段计算最小赋权超图的初始划分;在迁移优化阶段逐层将划分从最小赋权超图投影到初始赋权超图,并对划分的投影进行迁移优化。

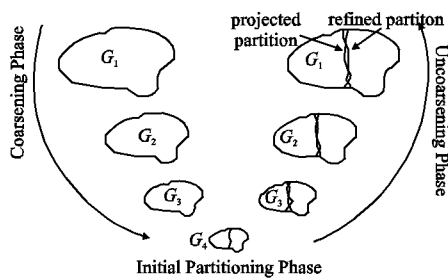


图1 基于多水平方法的赋权超图划分的实现步骤

Fig.1 Multi-level method of weighted hypergraph partitioning

本文首先采用形式化符号描述了赋权超图优化划分问题,阐明了基于多水平思想求解该问题的粗化、初始划分和迁移优化三个阶段;接着,提出了基于离散粒子群的多水平方法的主要框架,以及该框架下的基于离散粒子群的多水平迁移优化算法;然后,给出了赋权超图的改进压缩存储格式和快速的划分割切值和结点收益值计算方法,有效地降低了划分算法的空间和时间复杂度;最后,基于ISPD98的18组测试基准,进行了基于迁移方法的赋权图、赋权超图优化划分算法和本文算法的对比实验,数据表明该算法有效地增强了迁移优化算法逃离局部最优搜索的能力。

## 2 赋权超图优化划分问题的形式化描述

为方便后续讨论,本文对赋权超图优化划分问题及其相关术语进行了如下的数学定义。

**定义1.** 对于给定的赋权超图  $H^G(V, E^h)$ , 其中  $V = \{v_i | 1 \leq i \leq n\}$ ,  $E^h = \{e_j^h | e_j^h \subseteq V, 1 \leq j \leq m\}$ . 现将赋权超图  $H^G$  的结点  $V$  划分为两个结点子集  $V^1$  和  $V^2$ , 满足  $V^1 \cup V^2 = V$  和  $V^1 \cap V^2 = \emptyset$ , 称其为赋权超图  $H^G(V, E^h)$  的划分  $P = \{V^1, V^2\}$ .

**定义2.** 定义边的非负权值为  $W: E \rightarrow R$ , 即边  $e_j^h$  的权值为  $W(e_j^h)$ . 定义结点的非负权值为  $S: V \rightarrow R$ , 即结点  $v_i$  的权值为  $S(v_i)$ .

**定义3.** 对于给定的赋权超图  $H^G(V, E^h)$ , 若  $v \in e^h$ , 则称  $e^h$  是结点  $v$  的邻接超边, 且记  $N(v) = \{e^h | v \in e^h \wedge e^h \in E^h\}$  为结点  $v$  的邻接超边集。

**定义4.** 对于给定赋权超图  $H^G(V, E^h)$  的划分  $P = \{V^1, V^2\}$ , 设结点  $v \in V^1$ , 则结点  $v$  的内联度为其邻接超边中所有结点全部处在结点子集  $V^1$  的边权值之和, 记为  $ID(v) = \sum_{e^h \in N(v) \wedge e^h \subseteq V^1} w(e^h)$ . 结点  $v$  的外联度为其邻接超边中结点处于不同结点子集的边权值之和, 记为  $ED(v) = \sum_{e^h \in N(v) \wedge e^h \not\subseteq V^1} w(e^h)$ . 该

划分割切为属于不同结点子集的超边权值之和, 记为  $cut(P)$

$$= \sum_{e^h \in E^h \wedge e^h \not\subseteq V^1 \wedge e^h \not\subseteq V^2} w(e^h).$$

**定义5.** 对于给定的赋权超图  $H^G(V, E^h)$  的划分  $P = \{V^1, V^2\}$ , 任意结点从当前结点子集  $V^1$  迁移到另一结点子集  $V^2$ , 则迁移前后的划分割切差称为该结点的收益  $gain(v)$ .

**定义6.** 给定平衡约束参数  $r$ , 对于给定赋权超图  $H^G(V, E^h)$  的划分优化问题是寻找割切最小化的划分  $P = \{V^1, V^2\}$ , 且同时满足约束条件  $\frac{(1-r)S(V)}{2} \leq S(V^1)$ ,  $S(V^2) \leq \frac{(1+r)S(V)}{2}$ .

## 3 基于离散粒子群的多水平迁移优化算法

在多水平迁移优化阶段, 一般采用基于迁移的 FM<sup>[13]</sup>、1-FM<sup>[13]</sup>、EE-FM<sup>[13]</sup>、LIFO-FM<sup>[14]</sup>、FIFO-FM<sup>[14]</sup> 等 FM 类结点迁移算法对每一水平层粗化赋权超图的划分进行优化. 本文分析上述 FM 类结点迁移算法的搜索能力, 观察到当划分陷入局部最优后, FM 类结点迁移算法简单地采用贪心原则来选择最小代价的结点进行迁移, 暂时允许划分割切的增加来解决局部最优问题, 导致其逃离局部最优能力受到限制。

**Input:** original hypergraph  $H_0^G(V_0, E_0^h)$

**Output:** the partitioning  $P_0$  of hypergraph  $H_0^G$

01 **Begin**

02  $k=0; H_k^G(V_k, E_k^h) = H_0^G(V_0, E_0^h);$

03  $H_{k+1}^G(V_{k+1}, E_{k+1}^h) = RM(H_k^G(V_k, E_k^h));$

04 **While** ( $|V_{k+1}| \geq \psi$ ) **Do Begin**

05  $k=k+1; H_{k+1}^G(V_{k+1}, E_{k+1}^h) = SHEM(H_k^G(V_k, E_k^h));$

06 **End While**

07  $P_k = GGGP(H_k^G);$  /\*  $P_l = P_k$  and  $H_l^G = H_k^G$  \*/

08 **For**  $i=1$  to  $N$  **Do**

09  $\vec{X}_{ki} = P_k;$

10  $\vec{Q}_{ki} = \vec{X}_{ki};$

11  $\vec{Y}_{ki} = \vec{U}[-y_{max}, +y_{max}];$

12 **End For**

13 **While** ( $k \geq 1$ ) **Do Begin**

14  $\vec{Q}_{k0} = MDPSOR(H_k^G, \vec{Y}_{k0}, \vec{X}_{k0}, \vec{Q}_{k0});$

15 /\* **Prepare for the next call of MDPSOR** \*/

16 **For**  $i=1$  to  $N$  **Do**

17 **Project the  $i^{th}$  particle' velocity  $\vec{Y}_{ki}$  to  $\vec{Y}_{(k-1)i}$**

18 **Project the  $i^{th}$  particle' position  $\vec{X}_{ki}$  to  $\vec{X}_{(k-1)i}$**

19 **Project the  $i^{th}$  particle' best position  $\vec{Q}_{ki}$  to  $\vec{Q}_{(k-1)i}$**

20 **End For**

21 **Project the swarm's best position  $\vec{Q}_{k0}$  to  $\vec{Q}_{(k-1)0}$**

22  $k=k-1;$

23 **End While**

24  $P_0 = MDPSOR(H_0^G, \vec{Y}_0, \vec{X}_0, \vec{Q}_0);$

25 **End**

图2 基于离散粒子群的多水平方法的主要框架

Fig.2 Framework of the multi-level method based on the discrete PSO

进而, 本文提出了基于离散粒子群的结点迁移优化算法 (简称 MDPSOR 算法), 在  $|V|$  维空间中每个粒子所处的位置对应于超图划分方案; 伴随着投影优化阶段的细化, 每个粒

子对应的划分方案映射到下一水平层的粗化超图;随着划分问题解空间的维度 $|V|$ 增加,粒子的自由度增大;将结点迁移的收益值作为个体的启发式搜索信息,利用粒子之间聚集协同表现出的智能行为,有效地搜索解空间。

### 3.1 基于离散粒子群的多水平方法的主要框架

基于离散粒子群的多水平方法的主要框架如图2所示。在初始划分阶段,第7行代码计算最小赋权超图 $H_1^G(V_1, E_1^h)$ 的初始划分 $P_1$ ,第9到10行代码初始化每个粒子的历史最优位置向量 $\vec{Q}_{ki}$ 和初始位置向量 $\vec{X}_{ki}$ 为 $P_1$ 。第11行代码利用随机

函数初始化每个粒子的速度向量 $\vec{Y}_{ki}$ 。在投影优化阶段,第14行代码通过MDPSOR算法得到当前水平层的粗化赋权超图的近似最优划分 $\vec{Q}_{k0}$ 。第16到20行代码将 $k$ 水平层第 $i$ 粒子的当前速度向量 $\vec{Y}_{ki}$ 投影到 $\vec{Y}_{(k-1)i}$ ,当前位置向量 $\vec{X}_{ki}$ 投影到 $\vec{X}_{(k-1)i}$ ,历史最优位置向量 $\vec{Q}_{ki}$ 投影到 $\vec{Q}_{(k-1)i}$ 。第21行代码投影粒子群近似最优划分方案 $\vec{Q}_{k0}$ 到 $k-1$ 水平层,作为 $k-1$ 水平层的近似最优划分方案初始值 $\vec{Q}_{(k-1)0}$ 。第24行代码通过在原始图 $H_0^G(V_0, E_0^h)$ 上运行MDPSOR算法,计算得到整个粒子群的近似最优划分 $P_0$ 。

**Input:** hypergraph  $H_k^G(V_k, E_k^h)$ , inertia weight  $\alpha$ , scale of swarm  $N$ , cognitive parameter  $\phi_1$ , social parameter  $\phi_2$ , balancing constraint  $\beta$ , maximum velocity  $y_{max}$ , the maximum cycles  $\delta$

**Output:** swarm's best position  $\vec{Q}_{k0}$

```

01 Begin
02   For  $t = 1$  to  $\delta$  Do
03     For  $i = 1$  to  $N$  Do
04       Initial two-dimensional auxiliary array  $EDG[2][m]$ 
05       For  $d = 1$  to  $|V_k|$  Do /* each vertex  $v$  in  $V_k$  */
06         fast calculate the  $d^{th}$  vertex's gain;
07         If the  $d^{th}$  vertex's gain  $> 0$  Do
08           move the  $d^{th}$  vertex to the other side;
09         Else
10            $y_{id}^{t+1} = \alpha \cdot y_{id}^t + random(0,1) \cdot \phi_1 \cdot (q_{id}^t - x_{id}^t) + random(0,1) \cdot \phi_2 \cdot (q_{i0}^t - x_{id}^t)$ ;
11            $y_{id}^{t+1} = sign(y_{id}^{t+1}) \cdot \min(abs(y_{id}^{t+1}), y_{max})$ ;
12            $x_{id}^{t+1} = \begin{cases} 1 & \text{if } random(0,1) < 1.0 / (1.0 + \exp(-1.0 \times y_{id}^{t+1})) \\ 0 & \text{otherwise} \end{cases}$ ;
13           If  $x_{id}^{t+1} \neq x_{id}^t$  Do
14             move the  $d^{th}$  vertex to the other side;
15           End If
16         End If
17       End For
18       Apply the EE-FM algorithm to refine  $\vec{X}_{ki}$  which is satisfied with the balancing constraint  $\beta$ 
19       Update the  $i^{th}$  particle's position  $\vec{Q}_{ki}$ 
20     End For
21   Update the swarm's best position  $\vec{Q}_{k0}$ 
22 End For
23 End

```

图3 基于离散粒子群的多水平迁移优化算法伪代码

Fig. 3 Pseudo-code of the multi-level discrete particle swarm optimization refinement algorithm

### 3.2 基于离散粒子群的多水平迁移优化算法

图3给出了基于离散粒子群的多水平迁移优化算法伪代码。第2到22行是算法的第一层循环,控制变量为 $t$ ,直至满足规定的迭代次数 $\delta$ 。第3到20行是第二层循环,控制变量为 $i$ ,直至处理整个粒子群 $N$ 。第5到17行是第三层循环,控制变量为 $d$ ,直至处理粒子的所有维度 $|V_k|$ ,即每个维度对应的结点。第6行,运用“快速结点收益值计算方法”计算出第 $i$ 个粒子的第 $d$ 个结点在当前划分的迁移收益。如果收益值大于零,则执行第8行的结点迁移,否则执行第10到15行。第10行遵循粒子群算法的速度更新公式,更新第 $i$ 个粒子在第 $d$ 维的当前速度。如果对第 $i$ 个粒子的加速导致它在第 $d$ 维的速度超过该维的最大速度 $y_{max}$ ,则第11行将第 $d$ 维的速度限制为最大速度 $y_{max}$ 。第12行遵循粒子群算法的位置更新公

式,更新第 $i$ 个粒子在第 $d$ 维的当前位置。第13到15行,判断第 $i$ 个粒子在第 $d$ 维的当前位置是否发生变化。如果变化则将第 $d$ 个结点迁移。经过迁移一连串结点后,为避免第 $i$ 个粒子当前位置向量 $\vec{X}_{ki}$ 对应的划分违背平衡约束参数 $\beta$ ,第18行代码结合“快速的划分割切值计算方法”,运行EE-FM算法优化当前位置向量 $\vec{X}_{ki}$ 。在第二层循环中第19行的代码,更新第 $i$ 个粒子的历史最优位置向量 $\vec{Q}_{ki}$ 。在第一层循环中第21行的代码,更新粒子群的全局最优位置 $\vec{Q}_{k0}$ 。

## 4 算法的实现和时空复杂度分析

### 4.1 赋权超图的压缩存储格式

本文采用了改进的(ICSF: Improved compressed storage



format) 压缩存储格式。ICSF 存储每个结点所有邻接超边的列表信息在全局数组 *adjncy*; 存储每个结点所有邻接超边列表的起始位置信息在全局数组 *xadj*; 存储每条超边的结点列表信息在全局数组 *eind*; 存储每条超边的结点列表的起始位置信息在全局数组 *eptr*. 存储结点的权值信息在使用全局数组 *vwgts*; 存储超边的权值信息在使用全局数组 *hewgts*.

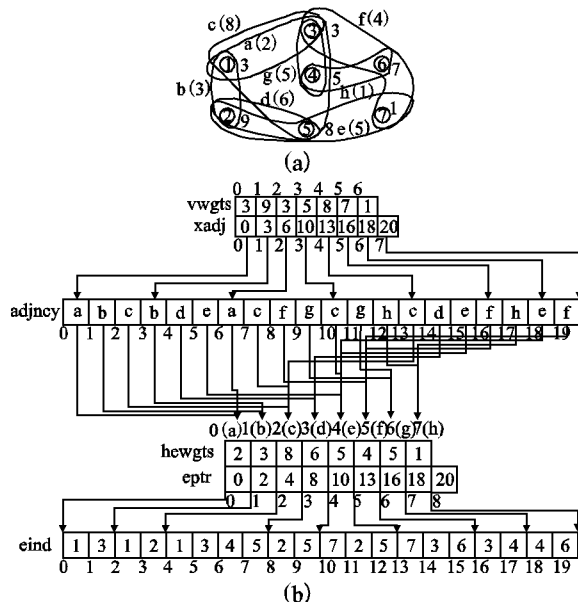


图4 赋权超图及其 ICSF 存储格式的示例

Fig. 4 An example of weighted hypergraph and its improved compressed storage format

#### 4.2 快速的结点收益值计算方法

为降低 MDPSOR 算法时间复杂度,快速的结点收益值计算方法统计每条超边在划分子集的结点个数并存储在二维辅助数组  $EDG[2][m]$ , 相应的结点收益值计算步骤如下:

**初始化步骤:**在图3算法的步骤4中,初始化二维辅助数组  $EDG[2][m]$  值为零,并基于初始划分统计每条超边在两个划分子集的结点个数.

**更新步骤:**在图3的步骤8或步骤14中,当结点从原划分子集迁移到新划分子集(设结点原划分子集为 *from*,新划分子集为 *to*),访问其邻接超边 *eId*,  $EDG[from][eId]$  执行减1操作和  $EDG[to][eId]$  执行加1操作.

**计算收益值步骤:**初始化结点收益值为零;读取结点的所有邻接超边的权值,访问其邻接超边 *eId*,如果  $EDG[to][eId]$  值等于0,则收益值减邻接超边的权值;如果  $EDG[from][eId]$  值等于1,则收益值加邻接超边的权值.

#### 4.3 快速的划分割切值计算方法

通过遍历每条超边,根据其结点所处的划分子集判断是否为两栖边;划分割切值为所有两栖边的权值之和.为了避免遍历每条超边的所有结点,有效降低划分割切值计算方法的时间复杂度,两栖边的判断依据是其在两个划分子集都有结点,即当  $(EDG[to][eId] \geq 1) \&\& (EDG[from][eId] \geq 1)$  条件表达式为真时,则该超边判定为两栖边.

#### 4.4 时间复杂度分析

设超边数为 *m*,结点数为 *n*,超边的结点最大数目为 *y*(*y*

$< n$ ),结点的邻接超边最大数目为 *x*(*x*  $< m$ ),迭代次数为  $\delta$ ,粒子群规模为 *N*.

整个算法为三层嵌套循环体,第一层循环体的循环次数为  $\delta$ ,第二层循环体的循环次数为  $N \times \delta$ ,第三层循环体的循环次数为  $n \times N \times \delta$ . MDPSOR 算法的空间复杂度主要包括5部分:1) 第二层循环中辅助数组 EDG 的初始化,其时间复杂度是  $O(N \times \delta \times m \times y)$ . 2) 第三层循环中每个维度对应结点的收益值快速计算,其时间复杂度是  $O(n \times N \times \delta \times x)$ . 3) 第三层循环中全部粒子的速度更新和位置更新,其时间复杂度是  $O(n \times N \times \delta \times 2)$ . 4) 第二层循环中 EE-FM 算法,其算法时间复杂度相对离散粒子群算法的时间复杂度可以忽略不计. 5) 第二层循环中全部粒子的历史最优位置的更新,其时间复杂度是  $O(N \times \delta \times m)$ . 忽略低阶因素的影响,MDPSOR 算法的时间复杂度为  $O(N \times \delta \times m \times y + n \times N \times \delta \times x)$ ,从而表明整个算法的计算主要集中在迭代过程中辅助数组 EDG 的初始化和结点收益值的计算.

#### 4.5 空间复杂度分析

设粒子群规模为 *N*,超边中结点个数总和为 *z*. MDPSOR 算法的空间复杂度主要包括4部分:(1) 粒子群的空间,大小为  $3 \times n \times N$ . (2) *eind* 和 *adjncy* 全局数组空间,大小为  $2 \times z$ . (3) *eptr* 和 *adjwgt* 全局数组空间,大小为  $2 \times m$ . (4)  $EDG[2][m]$  数组空间,大小为  $2 \times m$ . MDPSOR 算法的空间复杂度为  $\Omega(3 \times n \times N + 4 \times m + 2 \times z)$ .

### 5 实验结果与分析

#### 5.1 ISPD98 电路测试基准

文献[16]给出了学术界和工业界电路划分测评的基准工具 ISPD98. 本文分别按照电路到赋权图转换算法<sup>[17]</sup>和电路到赋权超图转换算法<sup>[15]</sup>的伪代码,将 ISPD98 电路测试基准转换到赋权图和赋权超图. 本文在粗化阶段,选用 RM 与 SHEM 组合粗化策略<sup>[12]</sup>;在初始划分阶段,选用 GGGP 初始剖分算法<sup>[13]</sup>;在迁移优化阶段,选用基于 FM 的赋权图迁移优化划分算法 FM-Graph<sup>[6]</sup>、基于 FM 的赋权超图迁移优化划分算法 FM-HyperGraph<sup>[8]</sup>和本文算法(简称 PSO-HyperGraph)进行对比实验. 在 PSO-HyperGraph 算法中,设定的粒子数 *N* 为40,迭代次数  $\delta$  为40,惯性权值参数  $\alpha$  为1,认知参数  $\varphi_1$  为0.5,社会参数  $\varphi_2$  为0.5.

下页表2给出了三种算法的割切最小值和平均值对比实验数据. 其中,FM-Graph 算法的划分割切值来源于文献[9],FM-HyperGraph 的划分割切值来源于文献[10].

表2的实验数据分析表明:FM-Graph 算法的割切值远大于 PSO-HyperGraph 和 FM-HyperGraph 算法的割切值,从而表明赋权超图相比赋权图提供了精确的数学模型. PSO-HyperGraph 算法得到的割切值优于 FM-HyperGraph 算法,从而表明 PSO-HyperGraph 算法相比 FM-HyperGraph 算法具备更强的逃离局部最优能力.

### 6 结束语

本文运用群智能理论,针对赋权超图优化划分问题提出了基于离散粒子群的多水平迁移优化算法,其中粒子所处

$|V|$ 维空间位置对应于一个超图划分方案;伴随着投影优化阶段的细化,每个粒子对应的划分方案映射到下一水平层的粗化超图;随着划分问题解空间的维度 $|V|$ 增加,粒子的自由度增大;将结点迁移的收益值作为个体的启发式搜索信息,利

表2 FM-Graph,FM-HyperGraph 和 PSO-HyperGraph 算法的对比实验数据

Table 2 Experimental result comparison under FM-Graph, FM-HyperGraph and PSO-HyperGraph

电路基准	FM-Graph		FM-HyperGraph		PSO-HyperGraph	
	Min	Ave	Min	Ave	Min	Ave
ibm01	517	1091	469	602	235	241
ibm02	4268	11076	421	477	338	344
ibm03	10190	12353	1 338	2 080	962	1088
ibm04	2273	5716	894	1 969	635	794
ibm05	12093	15058	2 319	3 552	2024	2801
ibm06	7408	13586	892	1 792	829	1114
ibm07	3219	4140	995	1 840	921	1259
ibm08	11980	38180	1 492	2 713	1275	1427
ibm09	2888	4772	1 700	3 029	667	730
ibm10	10066	17747	1 788	2 806	1682	1822
ibm11	2452	5095	1 482	3 833	945	1168
ibm12	12911	27691	3 582	5 929	3318	3693
ibm13	6395	13469	1 551	2 287	1179	1642
ibm14	8142	12903	2 657	6 942	2413	3275
ibm15	22525	46187	6 020	9 697	3175	3594
ibm16	11534	22156	2 533	5 425	2426	2977
ibm17	16146	26202	3 208	5 898	2762	3169
ibm18	15470	20018	2 076	3 420	1853	2354

用粒子之间聚集协同表现出的智能行为,有效地搜索解空间。本文对该算法进行了时间复杂度和空间复杂度分析,分析表明其时间花费集中在迭代过程中的结点收益值计算上。进而通过统计每条超边在划分子集的结点个数并存储在二维辅助数组,提出了快速的划分割切值和结点收益值的计算方法。本文针对 ISPD98 的 18 组测试基准,进行了基于迁移方法的赋权图、赋权超图优化划分算法和基于离散粒子群的赋权超图优化划分算法,对比实验数据分析表明,本文算法有效地增强了迁移优化算法逃离局部最优的能力。

#### References:

- [1] Nan Guo-fang. Research on algorithm solving key problem in VLSI physical design[D]. Tianjing: Tianjing University, 2004.
- [2] Shi Heng-liang. Research on job scheduling on cloud computing[D]. Nanjing: Nanjing University of Science and Technology, 2012.
- [3] Lengauer T. Combinatorial algorithms for integrated circuit layout [M]. Berlin, John Wiley & Sons, Inc., 1990.
- [4] Alpert C J, Kahng A B. Recent directions in netlist partitioning, integration[J]. The VLSI Journal, 1995, 19(2): 1-81.
- [5] Kernighan B W, Lin Song. An efficient heuristic procedure for partitioning graphs[J]. Bell System Technical Journal, 1970, 49(2): 291-307.
- [6] Fiduccia C, Mattheyses R. A linear-time heuristics for improving network partitions[C]. Proc. of the 19th Design Automation Conference, Los Alamitos, USA: IEEE Computer Society Press, 1982: 175-181.
- [7] Pilkington J, Baden S. Partitioning with space filling curves[D]. San Diego, USA: University of California, 1994.
- [8] Dongarra J, Fox G, Kennedy K, et al. Sourcebook of parallel computing[M]. San Francisco, USA: Morgan Kaufmann Publishers, 2003.
- [9] Sun Ling-yu, Leng Ming, Yu Song-nian. Spectral-based weighted undirected graph partitioning algorithm[J]. Applications Research of Computers, 2009, 26(6): 2086-2089.
- [10] Leng Ming, Sun Ling-yu, Bian Ji-nian, et al. Research on cellular automata model of hypergraph partitioning problem and its algorithm[J]. Computer Engineering, 2012, 38(15): 23-27.
- [11] Soufiane R. Hypergraph cuts and unsupervised representation for image segmentation[J]. Fundamenta Informaticae, 2009, 96(1-2): 153-179.
- [12] Karypis G, Aggarwal R, Kumar V, et al. Multilevel hypergraph partitioning: applications in VLSI domain[C]. Proc. of the 34th Design Automation Conference, New York, USA: ACM Press, 1998.
- [13] Karypis G, Aggarwal R, Kumar V, et al. Multilevel hypergraph partitioning: applications in VLSI domain[J]. IEEE Transactions on Very Large Scale Integration Systems, 1999, 7(1): 69-79.
- [14] Hagen L, Huang D, Kahng A. On implementation choices for iterative improvement partitioning algorithms[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 1997, 16(10): 1199-1205.
- [15] Leng Ming, Sun Ling-yu, Bian Ji-nian, et al. An  $O(m)$  algorithm for cores decomposition of undirected hypergraph[J]. Journal of Chinese Computer Systems, 2013, 34(11): 2568-2573.
- [16] Alpert C J. The ISPD98 circuit benchmark suite[C]. Proc. of ACM International Symposium of Physical Design, New York, USA: ACM Press, 1998: 80-85.
- [17] Leng Ming, Sun Ling-yu, Zhu Ping, et al. Comparative experiment of the core property of weighted hypergraph based on the various matching strategies[J]. Computer Engineering, 2013, 39(6): 85-90.

#### 附中文参考文献:

- [1] 南国芳. VLSI 物理设计中关键问题求解的算法研究[D]. 天津: 天津大学, 2004.
- [2] 史恒亮. 云计算任务调度研究[D]. 南京: 南京理工大学, 2012.
- [9] 孙凌宇, 冷 明, 郁松年. 基于谱方法的无向赋权图划分算法[J]. 计算机应用研究, 2009, 26(6): 2086-2089.
- [10] 冷 明, 孙凌宇, 边计年, 等. 赋权超图划分优化问题的元胞自动机模型及算法研究[J]. 计算机工程, 2012, 38(15): 23-27.
- [15] 冷 明, 孙凌宇, 边计年, 等. 一种时间复杂度为  $O(n)$  的无向超图核值求解算法[J]. 小型微型计算机系统, 2013, 34(11): 2568-2573.
- [17] 冷 明, 孙凌宇, 朱 平, 等. 基于结点匹配策略的赋权超图核值属性的实验比较[J]. 计算机工程, 2013, 39(6): 85-90.