

## 基于冲突延迟策略的约束群体自组装模型

窦全胜<sup>1,2,3</sup>, 安文<sup>1</sup>, 姜平<sup>1</sup>, 史忠植<sup>3</sup>

(1. 山东工商学院 计算机科学与技术学院, 山东 烟台 264005;

2. 山东工商学院 山东省高校智能信息处理重点实验室, 山东 烟台 264005; 3. 中国科学院 计算技术研究所, 北京 100080)

**摘要:**对二维网格空间上的群体自组装问题进行了较为深入的研究,给出了确保组装结构侧面连通性的充分必要条件,并进行了严格证明,以此为基础提出了基于最大相邻侧面封闭及冲突等待策略的群体自组装模型,该模型在并行的条件下,可以组装任意二维空间上的侧面连通结构,通过仿真和实验对所提出的模型进行了验证与分析。

**关键词:**群体智能; 自装配; 组块; 自组织

中图分类号: TP301.6

文献标识码: A

文章编号: 1000-436X(2012)12-0049-09

## Constrained collective self-assembly model based on collision delay strategy

DOU Quan-sheng<sup>1,2,3</sup>, AN Wen<sup>1</sup>, JIANG Ping<sup>1</sup>, SHI Zhong-zhi<sup>3</sup>

(1. School of Computer Science and Technology, Shandong Institute of Business and Technology, Yantai 264005, China;

2. Key Laboratory of Intelligent Information Processing in Universities of Shandong, Shandong Institute of Business and Technology, Yantai 264005, China;

3. Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080, China)

**Abstract:** The collective self-assembly problem on two dimensional grid space were studied, and put forward sufficient and necessary condition for ensuring the connectivity of the assembly structure side, which was proved strictly. Based on this condition, parallel assembly strategy was proposed for collective self-assembly model based on maximum adjacent side closing and collision delay. Under the parallel condition, the model could assembly arbitrary side connective structure on two dimensional grid space, and through simulation some properties of the model were analyzed and proved further.

**Key words:** collective intelligence; self-assembly; block; self-organization

### 1 引言

群体自组装(collective self-assembly)是指由仅具备局部能力的 Agent 完成某种复杂结构的过程,是当前群体智能领域研究的热点之一。关于群体自组装模型的研究主要包括: Werfel 采用正方形组块<sup>[1]</sup>对二维空间上指定的目标结构进行组装,并给出了

相应的组装算法,但不足的是算法缺少严格的理论支持。Mason 在文献[2]中阐述了这样一个系统,系统中的群体是由没有记忆但能够感知局部环境的 Agent 构成,个体能够产生“信息素”,并根据“信息素”浓度进行个体间交互。这个系统在二维空间上采用小正方形作为组块,最终成功地模拟了搭建“拱桥”的过程。Guo 等<sup>[3]</sup>将每一个组块看成是一

收稿日期: 2012-01-16; 修回日期: 2012-04-18

基金项目: 国家自然科学基金资助项目(60970088, 61272244, 60775035, 61173173, 60970105); 国家高技术研究发展计划(“863”计划)基金资助项目(2007AA01Z132)

**Foundation Items:** The National Natural Science Foundation of China (60970088, 61272244, 60775035, 61173173, 60970105); The National High-Technology Research and Development Program of China (863 Program) (2007AA01Z132)

个有限状态机，组块的状态在对其操作的过程中不断转变，根据不同的状态采取不同的动作。同时，系统还采用遗传算法对 Agent 所遵循的规则集进行动态修订。在自装配过程中，Agent 的行为是局部的，群体如何通过个体的局部行动来完成全局工作，即所谓的“局部到全局”(local to global)问题。为了解决这一问题，Jones and Matari 在文献[4]中提出了一种解决方案，设计出了一个规则转换器，这个转换器可以根据当前状态和所期望的目标生成一系列规则序列，Agent 遵循这个序列就能达到目标状态。另外还有一些有意义的工作<sup>[5-15]</sup>，在此不再逐一列举。

目前，由于研究背景不尽相同，在该领域的研究中缺乏统一的评判标准。Swell 等在文献[16]中提出一个二维空间上的自组装框架，将有界二维空间分割成  $n \times n$  个网格，称作网格空间 (grid space)，在网格空间上分布着若干正方形组块 (block)，可以看成是网格空间上的一个 Agent，其面积与网格大小相等，可以自行地在网格空间上移动 (但不能相互穿越) 或组装。指定某一位置作为组装的起始点，通过组块之间的连接产生该目标结构。

这里的主要问题在于：参与组装的组块没有全局能力，在目标结构远大于其感知范围的条件 下，如何并行地完成目标结构且不产生死锁。本文给出并严格证明了确保组装过程中结构侧面连通性的充分必要条件，提出基于最大相邻侧面封闭及冲突等待策略的群体并行自组装模型，该模型在并行条件下，可以组装任意二维空间上的侧面连通结构。

## 2 相关定义与符号

设  $G_n$  为二维  $n \times n$  网格空间，一个网格为  $G_n$  中的一个节点，组块  $b$  是  $G_n$  上占据一个网格节点的 Agent，用三元式  $\langle S, L, F \rangle$  表示，其中：

- 1)  $S$  为组块状态，包括 Idle、Mac、Pa、Waiting 和 Assembled 5 个状态，组块根据当前状态决定下一步的行动；
- 2)  $L = (i, j)$  为组块在  $G_n$  中的位置；
- 3)  $F$  为组块的侧面 (face) 集，每个组块包括  $f_E, f_S, f_W, f_N$  4 个侧面，侧面之间存在固定的方向，每个侧面有开放 (opened) 或关闭 (closed) 2 种不同状态。

设  $x$  为  $G_n$  中的某一节点，分别用  $(x)_i$  和  $(x)_j$  表

示  $x$  在  $G_n$  中的横坐标与纵坐标；设  $b$  是  $G_n$  上的一个组块，用  $L(b)$  表示组块  $b$  在  $G_n$  中的位置，即  $L(b) = ((b)_i, (b)_j)$ ，若  $f_x$  为组块的某一侧面， $f_{x+}$  为  $f_x$  逆时针方向的相邻侧面，与之相对  $f_{x-}$  为该侧面顺时针方向的相邻侧面， $\sim f_x$  为与  $f_x$  相对的另一个侧面，用  $Df_x(b)$  表示  $f_x$  所对的方向， $Df_x(b) + n$  表示  $G_n$  中侧面  $f_x$  所对应的方向上与该组块的距离为  $n$  的节点。组块  $b$  感知范围  $d(b)$  为  $G_n$  的一个子集：

$$d(b) = G_n \cap [(Df_E(b) + g)_i, (Df_W(b) + g)_i] \times [(Df_N(b) + g)_j, (Df_S(b) + g)_j]$$

其中， $g$  为组块  $b$  的感知半径。以上相关记号如图 1 所示。

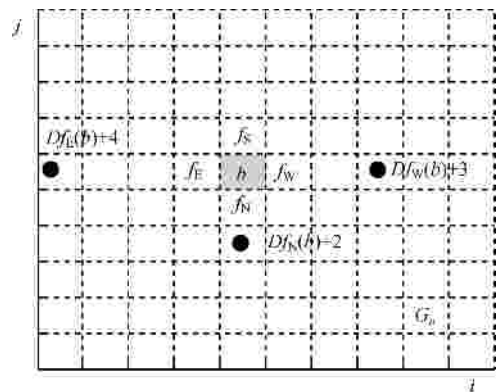


图 1 网格空间上的组块及相关符号的含义

若组块  $b$  的侧面  $f_S$  是开放的，且相邻的位置  $Df_S(b) + 1$  处存在侧面  $f_N$  开放的组块，或类似地，侧面  $f_W$  是开放的， $Df_W(b) + 1$  处有侧面  $f_E$  开放的组块，则 2 个组块可以相互连接，这个过程称为组装 (assembly)。

连接后的组块构成了  $G_n$  上的一个结构。若一个结构的某个侧面是开放的，且  $G_n$  上该侧面相对位置有组块存在，则这个结构可以和该组块进行连接，形成新的结构。称 2 个相互连接的侧面为内部侧面 (inside face)，本文以下所提到的侧面均指非内部侧面。

定义 1 设  $f_i, L, f_n, n=2,3$  为结构  $C$  相邻且开放侧面，若存在某一组块，可以与  $f_i, L, f_n$  全部连接，则称  $f_i, L, f_n$  为  $n$ -相关侧面 ( $n$ -related face)，记作  $f_i \sim L \sim f_n$ 。

在图 2 中，标注“=”的侧面为内部侧面，而用弧线标注的侧面为相关侧面。图 2 中包括 2 个 2-相关侧面和 1 个 3-相关侧面。

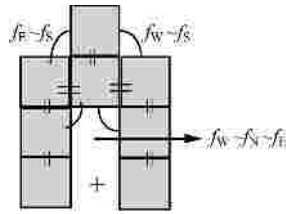


图 2 结构的内部侧面和相关侧面

定义 2 设  $f$  为结构  $C$  的侧面，称结构  $C$  是侧面连通的，当且仅当存在一个侧面序列  $f_0, f_1, \dots, f_n$ ，满足如下条件：

- $f_0 = f_n = f$ ；
- $f_k$  与  $f_{k+1}$  为相邻侧面， $0 \leq k \leq n-1$ 。

除  $f$  外， $C$  上所有侧面在序列  $f_0, f_1, \dots, f_n$  中出现且仅出现一次。

根据定义 2，图 3(a)中结构的任意侧面，只需按固定顺序沿结构边缘对侧面进行遍历，即可得到定义 2 所述的侧面序列，图 3(a)为侧面连通结构。而图 3(b)中结构内部的侧面与外部的侧面无法连通，为非侧面连通结构。

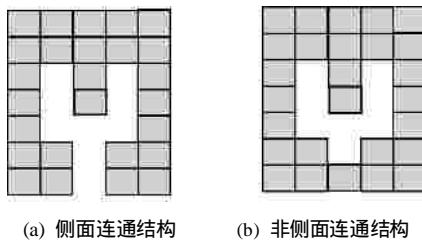


图 3 侧面连通结构举例

### 3 结构侧面连通的充要条件

本节将给出确保结构侧面连通性的相关定理。

定理 1 若  $C$  为  $G_n$  上的侧面连通结构， $C'$  为结构  $C$  与组块  $b$  连接得到的新结构， $C'$  为侧面连通结构的充分必要条件是  $b$  满足以下 2 个条件  $C_1$  和  $C_2$ 。

条件 1 ( $C_1$ ):  $b$  点的任意 2 个相对侧面  $f_x, \sim f_x$  封闭，与另外 2 个相对侧面  $f_{x+}, f_{x-}$  开放不同时成立；

条件 2 ( $C_2$ ):  $b$  点的  $f_x$  侧面封闭， $\sim f_x, f_{x-}$  (或  $f_{x+}$ ) 侧面开放，与  $t$  点存在固定组块不同时成立，其中， $t = Df_x - (D \sim f_x(p) + 1) + 1$  或  $t = Df_x + (D \sim f_x(p) + 1) + 1$ 。

证明 首先，证明必要性，若  $C'$  为侧面连通结构， $b$  满足约束  $C_1, C_2$ 。只需证明其逆否命题成立

即可，即若  $b$  不满足约束  $C_1$  或  $C_2$ ，则  $C'$  为非侧面连通结构。

用反证法，假设  $b$  不满足约束  $C_1$ ，即  $b$  的 2 个  $f_x$  和  $\sim f_x$  封闭，与  $f_{x+}, f_{x-}$  开放同时成立时， $C'$  为侧面连通结构。

不妨设  $C'$  中与  $f_{x+}$  左右相邻的 2 个侧面为  $f_l$  和  $f_r$ ，由于  $f_x$  和  $\sim f_x$  已经封闭，为内部侧面，故而  $f_l, f_r$  必为  $C$  的侧面，根据已知条件，若  $C$  为  $G_n$  上的侧面连通结构，则存在一个侧面序列  $S_1: f_l, \dots, f_r, \dots, f_l$ ，使  $C$  中所有侧面在序列中出现且只出现一次，由假设知  $C'$  为侧面连通结构，则存在侧面序列  $S_2: f_l, \dots, f_r, f_x, f_l$  使得  $C'$  中的所有侧面在序列中出现且只出现一次，由于  $C'$  是  $C$  与组块  $b$  连接得到的结构，因此 2 个序列  $S_1, S_2$  的前半部分  $f_l, \dots, f_r$  相同，而序列  $S_1$  的后半部分  $f_r, \dots, f_l$  必有侧面不在  $f_r, f_x, f_l$  中，与侧面连通结构的定义矛盾，故而假设错误，即当  $b$  不满足约束  $C_1$  时， $C'$  为非侧面连通结构。仿照上述过程不难得出当  $b$  不满足约束  $C_2$  时， $C'$  亦为非侧面连通结构。必要性得证。

以下证明充分性，即若  $b$  满足上述 2 个约束  $C_1, C_2$ ，则  $C'$  为侧面连通结构，若  $b$  满足  $C_1$ ，有以下 3 种情况。

1)  $f_x$  封闭， $f_{x+}, \sim f_x, f_{x-}$  开放，不妨设  $f_x$  使  $C$  中侧面  $f_c$  封闭，对于  $C$  中任意侧面  $f_0$ ，存在  $f_0, \dots, f_c, \dots, f_0$ ，使得除  $f_0$  外， $C$  上所有侧面在序列中出现且仅出现一次，此时若满足约束  $C_2$ ，即在点  $Df_x - (D \sim f_x(p) + 1) + 1$  及  $Df_x + (D \sim f_x(p) + 1) + 1$  处没有固定组块存在，则在  $C'$  中必有侧面序列  $f_0, \dots, f_{x+}, \sim f_x, f_{x-}, \dots, f_0$  (根据遍历方向不同也可能是  $f_0, \dots, f_{x-}, \sim f_x, f_{x+}, \dots, f_0$ )，使得除  $f_0$  外， $C'$  上所有侧面在序列中出现且仅出现一次，即  $C'$  为侧面连通结构。

2)  $f_x, f_{x+} (f_{x-})$  封闭， $\sim f_x, f_{x-} (f_{x+})$  开放；以  $f_x, f_{x+}$  封闭， $\sim f_x, f_{x-}$  开放为例，另一种情况类似。不妨设  $f_x, f_{x+}$  使  $C$  中侧面  $f_{c_1}, f_{c_2}$  封闭，显然这 2 个侧面在封闭前是相邻侧面，对于  $C$  中任意侧面  $f_0$ ，存在  $f_0, \dots, f_{c_1}, f_{c_2}, \dots, f_0$ ，使得除  $f_0$  外， $C$  上所有侧面在序列中出现且仅出现一次，此时若满足约束  $C_2$ ，即在点  $Df_x - (D \sim f_x(p) + 1) + 1$  处没有固定组块存在，则在  $C'$  中必有侧面序列  $f_0, \dots, \sim f_x, f_{x-}, \dots, f_0$  (根据遍历方向不同也可能是

$f_0, L, f_x, \sim f_x, L, f_0$ ), 使得除  $f_0$  外,  $C'$  上所有侧面在序列中出现且仅出现一次, 即  $C'$  为侧面连通结构。

3)  $f_x, f_x+, \sim f_x$  封闭,  $f_x-$  开放; 不妨设  $f_x, f_x+, \sim f_x$  使  $C$  中的 3-相关侧面  $f_{c_1} \sim f_{c_2} \sim f_{c_3}$  封闭, 对于  $C$  中任意侧面  $f_0$ , 存在  $f_0, L, f_{c_1}, f_{c_2}, f_{c_3}, L, f_0$ , 使得除  $f_0$  外,  $C$  上所有侧面在序列中出现且仅出现一次, 则在  $C'$  中必有侧面序列  $f_0, L, f_x-, L, f_0$ , 使得除  $f_0$  外,  $C'$  上所有侧面在序列中出现且仅出现一次, 即  $C'$  为侧面连通结构。

综上 3 种情况, 充分性得证。

证毕。

定理 1 为下面提出的自组装模型奠定了理论基础。

#### 4 基于冲突延迟策略的约束群体自组装模型

##### 4.1 组装点寻找算法

令  $fset1$ 、 $fset2$  和  $fset3$  分别表示在 Agent 感知范围  $d(b)$  内非相关侧面、2-相关侧面和 3-相关侧面集合。最优组装点的选取遵循所谓的“约束最大相邻侧面封闭法则”, 具体过程如下。

###### 算法 1

Searching Optimal Assembled Point

function  $e=FindOptimalPoint(b)$

If  $fset3$  isn't null

$tempf$ =select a face from  $fset3$  randomly ;

$e$ =position which is opposite to  $tempf$ ;

Else If  $fset2$  isn't null

While  $fset2$  isn't null do

$tempf$ =select a face from  $fset3$  randomly ;

Remove  $tempf$  from  $fset2$ ;

$t$ =position which is opposite to  $tempf$ ;

If  $t$  isn't satisfy  $C_1, C_2$

$e=t$ ;

End If;

End While;

Else If  $fset1$  isn't null

While  $fset1$  isn't null do

$tempf$ =select a face from  $fset1$  randomly ;

Remove  $tempf$  from  $fset1$ ;

$t$ =position which is opposite to  $tempf$ ;

If  $t$  isn't satisfy  $C_1, C_2$

$e=t$ ;

End If;

End While;

Else

$e=null$ ;

End If

Return  $e$

上述带组装点选择算法总是优先选择使结构相邻侧面封闭最多的位置作为组装点。在组装过程中, Agent 先不考虑结构的边界, 当组装完成后, 若对应的侧面为结构的边界, 则把该侧面的状态设置成封闭。

事实上, 算法 1 只需满足条件  $C_1$  和  $C_2$ , 即可避免死锁的产生。由于在组装过程中, 中间结构内部会产生一些“孔洞”, 这些“孔洞”仅能让较少的组块通过, 这在一定程度上影响组装的效率, 上述策略的主要目的是尽量减少组装过程中结构内部“孔洞”的产生, 进而提高组装效率。

##### 4.2 冲突延迟策略

事实上, 不难发现定理 1 仅保证了在串行条件下组装过程不会发生死锁。而在并行条件下, 多个组块可能在冲突位置同时组装, 造成不满足定理 1 的 2 个条件, 为解决这一问题, 本文提出了基于冲突延迟的并行组装策略, 规定任意组块  $b$  在不同状态的行为如下。

###### 算法 2

States and Action of Arbitrary Agent  $b$

STATE: Idle

If Agent Find Construction then

$STATE=Mac$  (Moving around construction);

Else

Moving on  $Gn$  randomly

End if

STATE: Mac

$e=findOptimalPoint(b)$

If  $e=null$

Moving Around Construction randomly

Else

If Current Position is  $e$

$STATE=Pa$  (Prepare Assembly)

Else

Moveing to  $e$

End If

```

End if
STATE: Pa
IF b satisfy C1 and C2
  Assembly;
  STATE= Assembled
Else
  IF STATE of Oppose(b) is Pa
    STATE= Waiting
  ELSE
    STATE= Mac ;
  END
End IF
STATE: Waiting
Waiting for some times randomly;
STATE= Pa
STATE : Assembled

```

在上述组块的状态和动作中,避免并行死锁的关键在于:当状态 Mac 组块搜索到最优组装点时,状态不直接转为 Assembled( 组装),而是转为 Pa,并将所有状态为 Pa 的组块视作固定组块,再次检查是否满足条件 C<sub>1</sub> 和 C<sub>2</sub>,若满足则组装;反之,若与之冲突的组块状态为 Assembled,则该组块状态由 Pa 转为 Mac,继续搜索合适的待组装点,若与之冲突的组块状态为 Pa,则这些相互冲突的组块状态由 Pa 转为 Waiting,进入随机等候状态,由于等候时间是随机的,最先结束等候的组块状态转为 Pa,而冲突的另一方(或多方)状态依然为 Waiting,此时,由算法 2 可知,状态为 Pa 的组块满足条件 C<sub>1</sub> 和 C<sub>2</sub>,可以组装,当其他组块结束等

候时,必不满足 C<sub>1</sub> 和 C<sub>2</sub>,状态由 Pa 转为 Mac,从而避免了死锁。综合算法 2 和以上论述,容易得到如下结论。

定理 2 若二维网格空间上的组块遵循算法 2 所述状态与规则,则能够组装完成二维网格空间上任意给定的侧面连通结构。

### 5 系统仿真

本节实验主要验证如下问题。

1) 本文中的组装模型可以并行组装任意二维网格空间上的侧面连通结构。

图 4~图 7 展示了 4 个侧面连通结构的组装过程,图 4 中包括了组装的中间结构和最终完成的结构。

初始时刻,图 4~图 7 中组块均匀分布在 90×90 网格空间内,4 个结构包含组块个数依次为 2 304、1 980、1 921、2 157,算法成功地完成了 4 个结构的组装。

下面仅以图 4 中的“方”形结构为例,对组装过程进行分析,图 8 展示了该结构 5 次组装的情况。

从图 8 中可以看出,在组装初期算法收敛较快,结构的大部分是在这期间完成的,而后来的小部分结构却占用了大部分时间。

表 1 更详细地列出了算法执行前 3 500 次完成结构的比例和结构最终的完成时间,从 5 次运行的平均结果中可以看出,在运行时间为 500、1 500、2 500 和 3 500 时,分别占最终完成时间的 0.63%、1.87%、3.12%和 4.38%,却完成了结构的 34%、53%、57%和 60%。不仅“方”形结构如此,其他结构的组装过程也表现出了早期收敛较快而后效率较

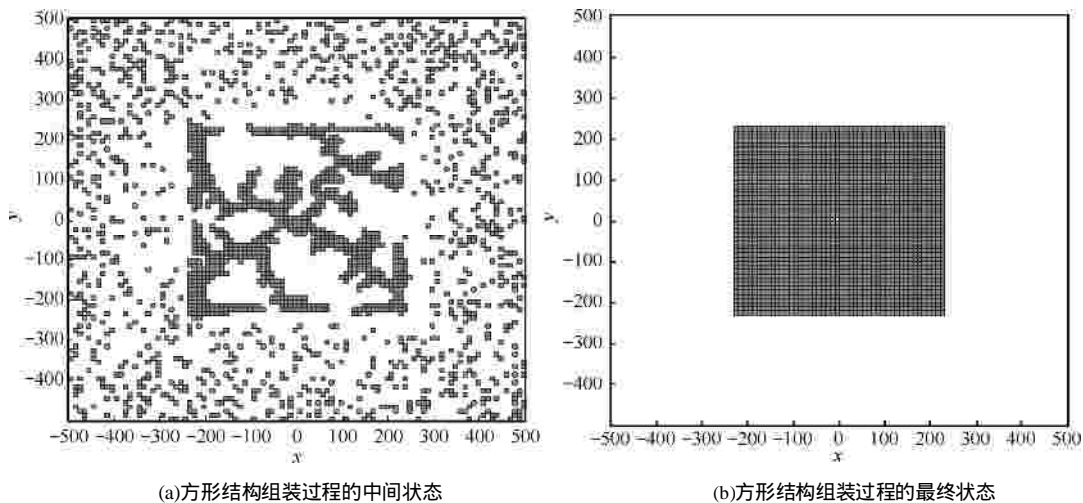


图 4 方形结构组装过程

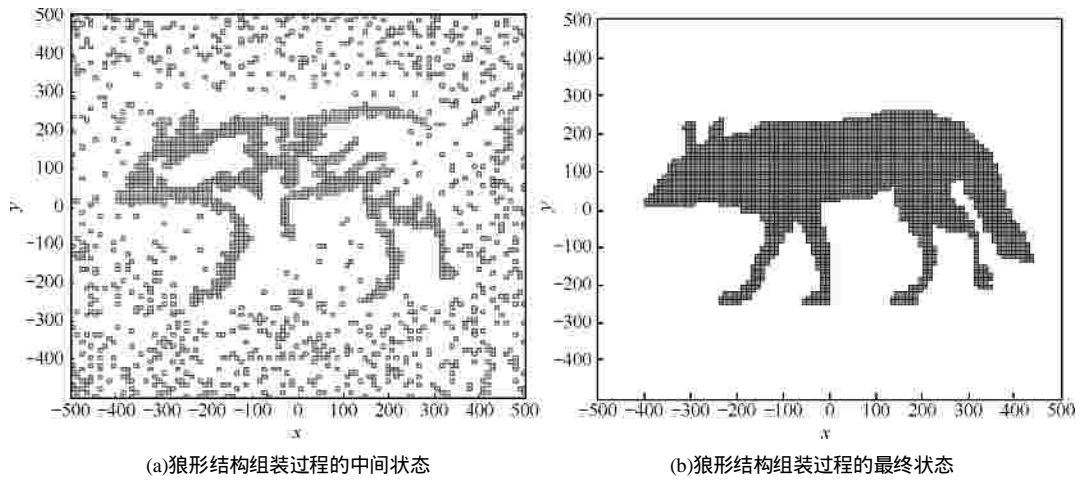


图 5 狼形结构组状过程

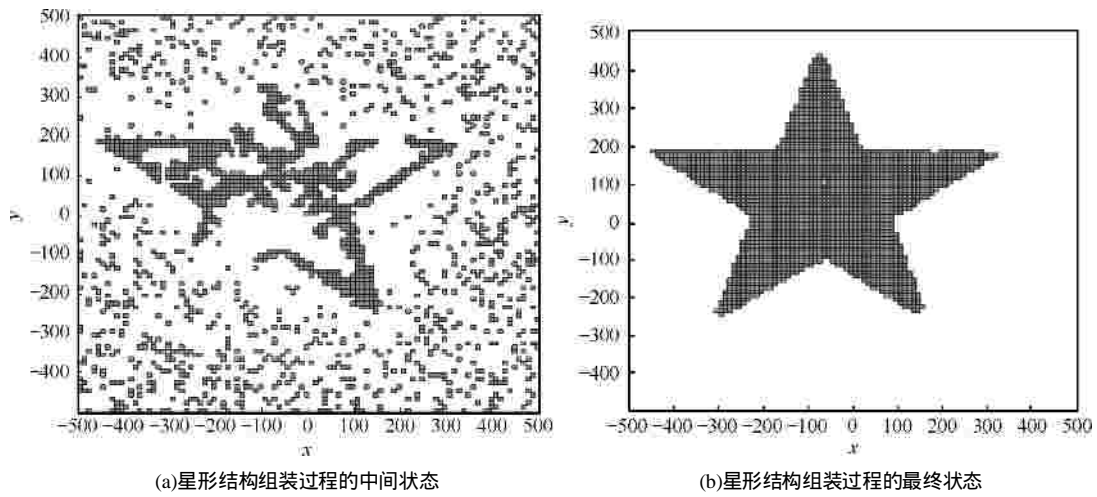


图 6 星形结构组状过程

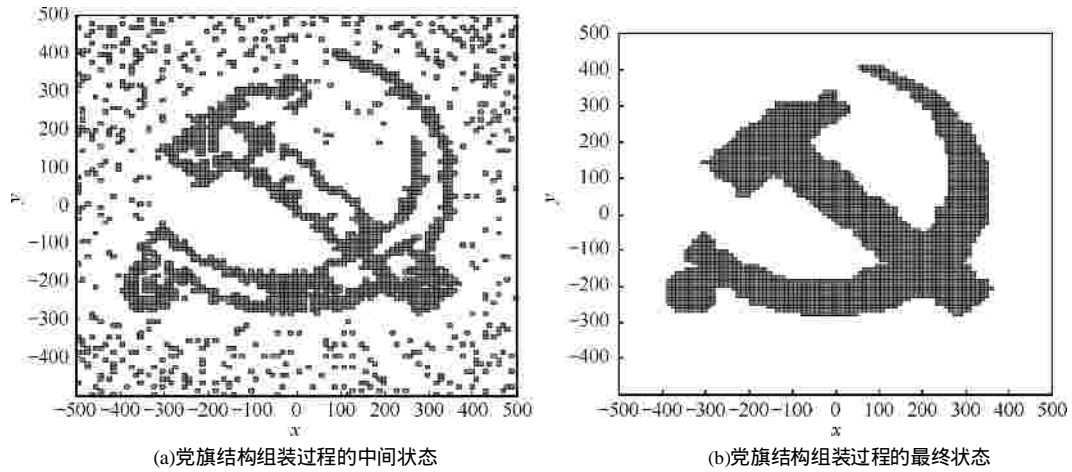


图 7 党旗结构组状过程

低的现象，其主要原因在于：一方面，在组装初期，组装的并行程度较高，而随着组块状态不断变成 Assembled，用于组装的组块数量大幅度减少，造

成组装的并行性逐渐减弱；另一方面，在组装的后期，结构大部分已经完成，由于组块只把握局部信息，且结构的体积远大于组块的感知区间，剩下的

组块需要在结构中寻找合适的组装位置，这一搜索过程造成了组装过程在后期效率偏低。

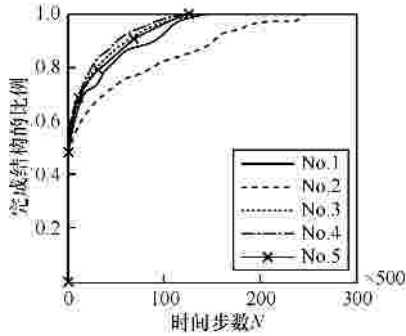


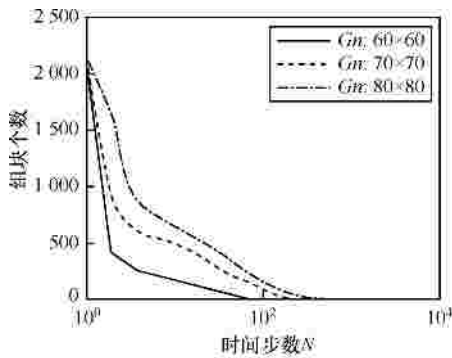
图 8 “方”形结构 5 次组装曲线

表 1 “方”形结构 5 次组装的初期情况及最终完成时间

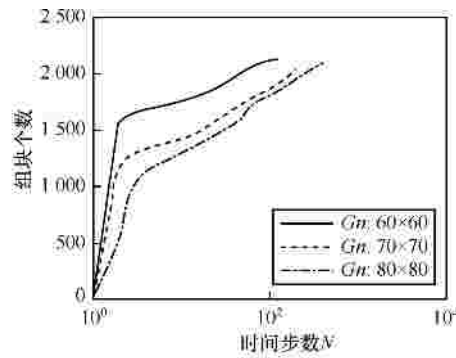
序号	时间步数 $N$				完成时间
	500	1 500	2 500	3 500	
1	0.32	0.55	0.59	0.61	75 812
2	0.37	0.50	0.54	0.55	125 919
3	0.34	0.53	0.58	0.61	58 900
4	0.35	0.52	0.57	0.60	74 796
5	0.33	0.55	0.60	0.64	64 424
平均	0.34	0.53	0.57	0.60	79 970

2) 组块状态变化特征及初始分布密度对其的影响。

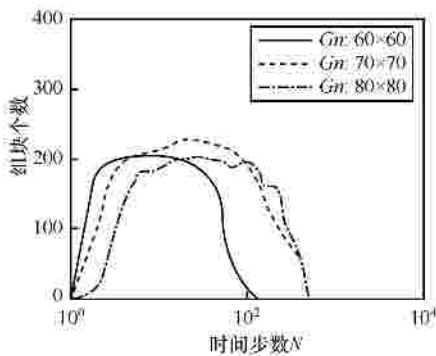
同样以“方”形结构为例，考察组块初始分布



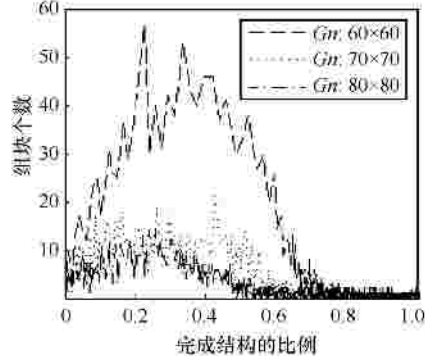
(a)Idle状态组块的变化曲线



(b)Assembled状态组块的变化曲线



(c)Mac状态组块的变化曲线



(d)Pa和Waiting状态的变化曲线

图 10 不同网格空间上，不同状态组块的变化曲线

密度  $D = \frac{\|C_g\|}{\|G_n\|}$  对并行性的影响，式中  $\|C_g\|$ 、 $\|G_n\|$  分别表示目标结构包含的组块个数和网格空间  $G_n$  的节点数。初始时组块均匀地分布在网格空间  $G_n$  上。在结构的初始组装点在所有条件均不变化的前提下，分别设置网格空间为  $60 \times 60$ 、 $70 \times 70$ 、 $80 \times 80$ ，从而组块的初始分布密度分别为 0.64、0.47 和 0.36，5 次运行所得平均曲线如图 9 所示。

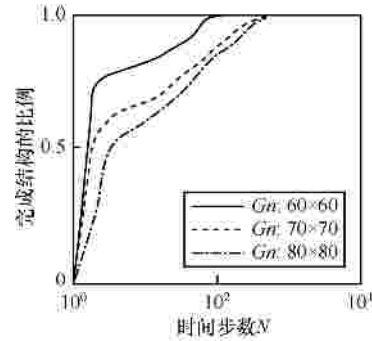


图 9 在不同网格空间上，“方”形结构的组装曲线

显然组块的初始分布密度对组装过程影响较大，组块在网格空间上的初始分布密度越大，其并行程度也就越高，收敛速度也就越快。为更详细地说明问题，图 10 考察了在上述 3 个不同网格空间上不同状态的组块变化情况。

图 10 分别是状态为 Idle、Assembled、Mac 的组块数量变化曲线。由图 10(a)~图 10(c)可以看出，

在最初时刻, 状态为 Idle、Assembled 和 Mac 的组块呈指数级下降和增长, 组块初始分布密度越大, 这种变化速率就越大; 由图 10(c)可知, 随着结构的逐渐完成, 状态为 Idle 的组块逐渐减少, 从而由 Idle 转成 Mac 的组块数量也不断减少, 因此状态为 Mac 的组块个数也逐渐减少; 图 10(d)为状态为 Pa 和 Waiting 的组块变化曲线, 为了考察结构完成到什么比例时, 算法的并行化程度最大, 这里的横坐标是完成结构的比例, 从图 10(d)中可以看出, 在 3 个不同的网格空间上, 初始分布密度越大, 并行化程度就越高, 从整体上看, 图 10(d)中曲线特征有相似之处, 在结构被完成 20%~40%之间, 并行组装的组块数量最多, 并行程度最大, 在此之后参与组装的组块逐渐减少, 并行性减弱。

图 11 展示了初始分布密度与最大并发数之间的关系。

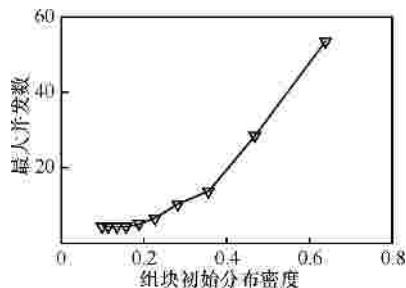


图 11 组块的初始分布密度与最大并发数之间关系

图 11 中, 网格空间为  $n \times n$ ,  $n=60, 70, \dots, 150$ , 在本例中, 当网格空间大于  $100 \times 100$  时, 即初始密度小于 0.23 时, 组装过程中的最大并行数不超过 10, 其组装的效率接近于串行。

## 6 结束语

本文提出了一种仅有局部能力的组块, 在并行且目标结构远大于其感知范围的条件下, 完成复杂全局结构的自组装模型, 在二维网格空间上, 提出了确保组装结构侧面连通性的充分必要条件, 并从理论上加以严格证明。提出并行组装策略用冲突延迟来避免并行条件下死锁现象的发生。通过实验和仿真, 成功地完成了 4 个不同形状的侧面连通结构的组装。并且以“方”形结构为例, 研究了组块分布密度对这类结构组装过程及并行性的影响, 同时研究了结构在组装过程中各状态组块的变化特征。以上研究对此领域的研究具有较强的促进意义。

## 参考文献:

- [1] WERFEL J. Building blocks for multi-agent construction[A]. Proceedings of Distributed Autonomous Robotic Systems 2004[C]. Toulouse, France, 2004.275-281
- [2] MASON Z. Programming with stigmergy: using swarms for construction[A]. Proceedings of Artificial Life VIII[C]. Sydney, Australia, 2002. 371-374.
- [3] JONES C, MATARIC M. From local to global behavior in intelligent self-assembly[A]. Proceedings of 2003 IEEE International Conference on Robotics and Automation[C]. 2003. 721-726.
- [4] BISHOP J, BURDEN S, KLAVINS E, *et al.* Self-organizing programmable parts[A]. Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems[C]. Edmonton, Canada, 2005.3684-3691.
- [5] WHITE P, KOPANSKI K, LIPSON H. Stochastic self-reconfigurable cellular robotics[A]. Proceedings of 2004 IEEE International Conference on Robotics and Automation[C]. New Orleans, Louisiana, 2004. 2888-2893.
- [6] WERFEL J, BAR-YAM Y, NAGPAL R. Building patterned structures with robot swarms[A]. Proceedings of Nineteenth International Joint Conference on Artificial Intelligence[C]. Edinburgh, Scotland, 2005. 1495-1502.
- [7] WERFEL J, NAGPAL R. Extended stigmergy in collective construction[J]. IEEE Intelligent Systems, 2006, 21(2):20-28.
- [8] WERFEL J, BAR-YAM Y, RUS D, *et al.* Distributed construction by mobile robots with enhanced building blocks[A]. Proceedings of 2006 IEEE International Conference on Robotics and Automation[C]. Orlando, USA, 2006.2787-2794.
- [9] STOY K, NAGPAL R. Self-repair through scale independent self-reconfiguration[A]. Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems[C]. Sendai, Japan, 2004. 2062-2067.
- [10] WHITE P, ZYKOV V, BONGARD J, *et al.* Three dimensional stochastic reconfiguration of modular robots[A]. Proceedings of Robotics: Science and Systems II[C]. Cambridge, MA, 2005.161-168.
- [11] DECUGNIERE A, POULAIN B, CAMPO A, *et al.* Enhancing the cooperative transport of multiple robots[A]. Proceedings of the Sixth International Conference on Ant Colony Optimization and Swarm Intelligence (ANTS-2008)[C]. Brussels, Belgium, 2008. 307-314.
- [12] GROß R, DORIGO M. Evolution of solitary and group transport behaviors for autonomous robots capable of self-assembling[J]. Adaptive Behavior, 2008, 16(5):285-305.
- [13] GROß R, DORIGO M. Self-assembly at the macroscopic scale[J]. Proceedings of IEEE, 2008, 96(19):1490-1508.

- [14] CHRISTENSEN A L, O'GRADY R, DORIGO M. A mechanism to self-assemble patterns with autonomous robots[A]. Proc of the 9th European Conference on Artificial Life (ECAL 2007)[C]. Berlin, Germany, 2007. 716-725.
- [15] YU C, NAGPAL R. A self-adaptive framework for modular robots in dynamic environments: theory and applications[J]. Journal of Robotics Research, 2010, 30(8):1015-1036.
- [16] WERFEL J, INGBER D, NAGPAL R. Collective construction of environmentally-adaptive structures[A]. IEEE Conference on Intelligent Robots and Systems[C]. Harvard Univ, Cambridge, 2007. 2345-2352.

#### 作者简介：



窦全胜(1971-),男,黑龙江大庆人,博士,山东工商学院副教授,主要研究方向为智能计算和数据挖掘。



安文(1966-),男,内蒙古呼和浩特人,博士,山东工商学院副教授,主要研究方向为智能计算和自动控制。



姜平(1979-),男,山东烟台人,山东工商学院讲师,主要研究方向为生物信息和医学图像处理。



史忠植(1941-),男,江苏无锡人,中国科学院计算技术研究所研究员、博士生导师,主要研究方向为智能科学理论与方法、知识工程、神经计算、数据挖掘、机器学习等。