

# 钢坯入库路径优化模型与算法

陈小文<sup>1</sup>, 杨 静<sup>1</sup>, 杨观赐<sup>2</sup>

(1. 贵州大学 计算机科学与技术学院, 贵州 贵阳 550025;

2. 贵州大学 教育部现代制造技术重点实验室, 贵州 贵阳 550025)

**摘 要:**在钢铁工厂的车间里,钢坯入库是一道非常重要的工序,它可归结为装箱问题。文中根据某钢厂的实际情况建立了相应的数学模型,以减少天车的行走距离,提高库房的利用率。通过分析天车行走总距离与钢坯入库顺序的关系,提出并论证了单存储区的最小入库序列所满足的性质,并利用该性质设计了多存储区的入库算法。多组模拟实验数据测试表明,单存储区测试结果验证了最小入库序列性质的正确性,多存储区测试结果表明了文中算法可大量缩短天车行走总距离和提高库房利用率。

**关键词:**装箱问题;可变路径;固定路径;最小入库序列

中图分类号:O224;TP391

文献标识码:A

文章编号:1673-629X(2009)12-0196-05

## Models and Algorithms of Path Optimization for Loading of Steel

CHEN Xiao-wen<sup>1</sup>, YANG Jing<sup>1</sup>, YANG Guan-ci<sup>2</sup>

(1. Sch. of Institute of Computer Science and Technology, Guizhou University, Guiyang 550025, China;

2. Ministry of Education Key Laboratory of Advanced Manufacturing Technology, Guizhou University, Guiyang 550025, China)

**Abstract:** In steel factories, the loading of steel is a very important process and it can be attributed to bin - packing problem. To reduce the mobile distance of crane and improve the utilization of warehouse, a model to solve this problem is established according to the actual situation in a steel plant. Based on this model, analysing the contacts between walking distance of crane and storage orders of steel, a property which the minimal loading sequences of single - storage - area must satisfy is put forward and proved, then a kind of loading algorithm basis of that property for multi - storage - area is proposed. The test results of single - storage - area are verified that correctness of the property and the test results of multi - storage - area show that the algorithm can greatly reduce the crane walking distance and improve the utilization rate of the warehouse.

**Key words:** bin packing; variable path; fixed path; minimal loading sequences

## 0 引 言

在某钢铁工厂车间里,钢条通过连铸线的切割以后,按其切割长度与钢号的不同分为多种不同类型钢坯。在生产过程中,需要在库房中找到一个合适的位置,用天车将其调入库房进行临时存放。合格品库房分为六个相同长度的存放区,按照离火车道的远近分别命名为1到6区(见图1)。钢坯在库房中的摆放,必须遵循同钢号同长度的钢坯摆放在同一堆上,且由

于库房的高度、地面的承受能力等其它因素的限制,每一堆的最大叠放层数不能超过11层,所有的钢坯宽度与存放区的宽度相同。

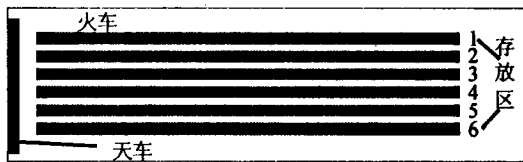


图1 库房示意图

深入研究钢坯入库问题,同时将其抽象简化,发现它在数学上是一个复杂的装箱(Bin Packing)问题<sup>[1]</sup>。在解决此类问题时,该问题又分为“在线”和“离线”两种情况:所谓“在线”,是指装入新到来的物品时,仅仅知道已装物品的信息,而不知将要到达物品及之后物品的信息,并且不允许移动已经放好的物品;所谓“离

收稿日期:2009-03-21;修回日期:2009-06-24

基金项目:国家自然科学基金项目(90718009);上海高可信计算实验室开放项目;贵州省科学技术基金(黔科合J字(2009,2123))

作者简介:陈小文(1985-),男,湖南浏阳人,硕士研究生,研究方向为智能计算、软件工程;杨 静,教授,博士,研究方向为可信软件、形式化方法。

线”,是指在进行装箱前,已经知道所有物品的信息。在装箱问题的算法研究中,对于 2 维的装箱问题研究较多<sup>[2,3]</sup>,3 维装箱问题由于其复杂性和难度大研究相应较少,而集装箱装入排放软件采用较多的是启发式方法,利用空间分解和组合的策略,如按层、块、三叉树空间分割等方法组合放置物体<sup>[4,5]</sup>。文献[6]利用柔性约束参数差异值和各库位到入口的距离设置匹配权值,并将最优适应(Best Fit)带上该权值提高了仓库的利用率,但没有从理论方面来支撑该算法,同时也忽略了路径优化。文献[7]提出复杂度为  $O(n^2)$  的随机适应算法(简称 RF 算法),在权衡复杂度和性能比时优于下次适应(Next Fit)、首次适应(First Fit)、最优适应(Best Fit),但并未考虑用 RF 算法时空间、路径优化情况。文中根据某钢厂的实际存放条件和入库流程,建立了数学模型。根据在钢坯入库时天车所移动的路径,提出了可变路径(Variable Path)和固定路径(Fixed Path)的概念,分析了入库时天车所移动的距离总和与两种路径的关系。在此基础上,从数学上提出并论证了单存放区时最小入库序列与钢坯入库顺序的关系,并设计了入库算法与算法验证方案。

## 1 数学模型

假设钢坯存放时,遵循自左至右的原则;再假设钢坯的生产线到存放区入口的距离相对于存放区的长度而言可以忽略不计。那么所研究问题的主要目标是:依据上述存放规则,按某种顺序将钢坯置入库中,使天车行走的路程总和较小并且库房利用率较高。

为了便于描述问题,给出如下的符号定义:

- 1)  $L$ : 每堆最大叠放层数;
- 2)  $B$ : 仓库中存放区间的数量,其中  $B \geq 1$ ;
- 3)  $T$ : 本次入库钢坯类型的总数量;
- 4)  $A_b$ : 本次入库前,第  $b$  存放区的已存放区的长度,其中  $A_b \geq 0$  且  $1 \leq b \leq B$ ;
- 5)  $H_k$ : 本次入库第  $k$  类型的钢坯数量,其中  $1 \leq k \leq T$ ;
- 6)  $a$ : 某一钢坯堆;
- 7)  $n$ : 本次入库钢坯堆的总数量,其中  $n \geq 1$ ;
- 8)  $t_{i,j}$ : 入库序列中第  $i$  堆第  $j$  层钢坯的钢号;
- 9)  $r_{i,j}$ : 入库序列中第  $i$  堆第  $j$  层钢坯的长度;
- 10)  $y_i$ : 入库序列中第  $i$  堆钢坯的层数,其中  $1 \leq i \leq n$  且  $y_0 = L$ ;

根据以上的符号定义,约束条件就可以表示为:

- 1)  $t_{i,j} = t_{i,k} \forall i, 1 \leq j, k \leq y_i$ ; 2)  $r_{i,j} = r_{i,k} \forall i, 1 \leq j, k \leq y_i$ ; 3)  $y_i \leq L \forall i$ 。有了约束条件,不妨令  $l_i$  为带上

述约束条件的序列中第  $i$  堆钢坯的长度,  $c_i$  为第  $i$  堆钢坯的层数。在上面约束条件中,条件 1) 规定每堆钢坯的钢号相同;在条件 2) 中,规定每堆钢坯的长度相同;在条件 3) 中,规定每堆钢坯层数不超过  $L$  层。对于某一要入库的钢坯,找到一个不违背约束条件又与入口具有最小距离的堆就能使天车的运动路程最短;因此当所有的钢坯都遵循此规则入库时,整体过程的天车运动路程就会比较小。

## 2 分析与定理证明

设天车的初始位置位于存放区的最左端,天车将一块钢坯置入存放区所行走的路程(Path,  $p_i$ ) 为天车从左向右移动到所置入钢坯的中点,然后返回到存放区的最左端。天车在入库的动作如图 2 所示。为了便于描述,不妨引入可变路径(Variable Path,  $v_i$ ) 和固定路径(Fixed Path,  $f_i$ ) 的概念。 $v_i$  是指存放区的最左端到入库后的第  $i$  堆钢坯的最左端的距离的两倍。 $f_i$  为第  $i$  堆钢坯的长度与该钢坯堆的层数的乘积,其值等于  $l_i \times c_i$ 。显然,  $\sum p_i = \sum v_i + \sum f_i$ 。对同一批钢坯不论以那种序列入库  $f_i$  都会相等。由于  $\sum f_i$  是定值,可以改变钢坯堆的入库顺序从而改变  $\sum v_i$ , 因此路径优化问题转向优化  $\sum v_i$ 。

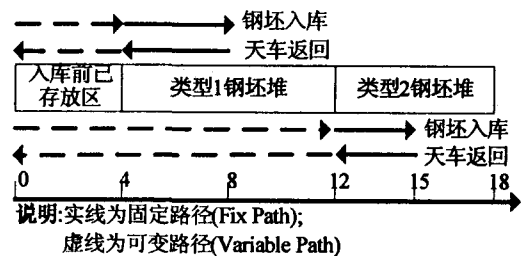


图2 钢坯入库天车行走示意图

当  $B = 1$  即只有一个存放区时,若将数量为  $n$  的钢坯堆序列置入空库中,则可变路径总和:

$$\begin{cases} \sum v_i = c_1 \times A_1, n = 1 \\ \sum v_i = \sum_{i=2}^n (c_i (A_1 + \sum_{j=1}^{i-1} l_j)), n \geq 2 \end{cases}$$

定理 1: 在只有一个存放区即当  $B = 1$  时,钢坯堆按某种顺序入库使  $\sum v_i$  最小的充分必要条件是序列满足:

$$c_l j - c_l i \geq 0 (1 \leq i < j \leq n) \quad (1)$$

必要性证明: \* 若有序列  $[(l_1, c_1), (l_2, c_2), \dots, (l_n, c_n)]$ , 假设以该序列依次置入库房使  $\sum v_i$  最小。这个序列的可变路径总和为:

$$S = c_2 l_1 + c_3 (l_1 + l_2) + \dots + c_n (l_1 + l_2 + \dots + l_{n-1}) \quad (2)$$

若交换其中第  $i$  堆与第  $i+1$  堆的位置,其序列变换为:

$$[(l_1, c_1)(l_2, c_2) \cdots (l_{i-1}, c_{i-1})(l_{i+1}, c_{i+1})(l_i, c_i) \cdots (l_n, c_n)] \quad (3)$$

该序列的可变路径总和:

$$S' = c_2 l_1 + c_3(l_1 + l_2) + \cdots + c_{i+1}(l_1 + \cdots + l_{i-1}) + c_i(l_1 + \cdots + l_{i-1} + l_{i+1}) + c_{i+2}(l_1 + \cdots + l_{i-1} + l_i + l_{i+1}) \cdots + c_n(l_1 + l_2 + \cdots + l_{n-1}) \quad (4)$$

由题设可知,  $S$  为最小, 此时有  $S' - S \geq 0$ , 得:

$$S' - S = c_i l_{i+1} - c_{i+1} l_i \geq 0 \quad (5)$$

观察发现对最小序列交换一次相邻项的顺序, 将它们的路程和相减, 都可以得到关系(5)。即得到了一个重要的关系: 最小序列当中, 相邻的钢坯之间满足关系:

$$c_i l_{i+1} - c_{i+1} l_i \geq 0 \quad (6)$$

通过对公式(6)进行变换得到:

$$\frac{c_i}{c_{i+1}} \geq \frac{l_i}{l_{i+1}} \quad (7)$$

令:  $a_i = \frac{c_i}{c_{i+1}}$  且  $b_i = \frac{l_i}{l_{i+1}}$  则有:

$$a_i \geq b_i \quad (8)$$

那么可以将这一序列进行分解为:

$$a_1 = \frac{c_1}{c_2}, b_1 = \frac{l_1}{l_2}; a_2 = \frac{c_2}{c_3}, b_2 = \frac{l_2}{l_3};$$

$$a_3 = \frac{c_3}{c_4}, b_3 = \frac{l_3}{l_4}; \cdots; a_i = \frac{c_i}{c_{i+1}}, b_i = \frac{l_i}{l_{i+1}}; \cdots;$$

$$a_{n-1} = \frac{c_{n-1}}{c_n}, b_{n-1} = \frac{l_{n-1}}{l_n} \quad (9)$$

根据公式(9)推导出:

$$c_1 = c_n \times \prod_{i=1}^{n-1} a_i, \quad l_1 = l_n \times \prod_{i=1}^{n-1} b_i \quad (10)$$

根据公式(8)、(9)、(10), 推导出:

$$c_1 l_i - c_i l_1 = c_1 l_1 \left( \frac{1}{\prod_{x=1}^{i-1} b_x} - \frac{1}{\prod_{x=1}^{i-1} a_x} \right) \geq 0 (2 \leq i \leq n),$$

$$c_2 l_i - c_i l_2 = c_2 l_2 \left( \frac{1}{\prod_{x=2}^{i-1} b_x} - \frac{1}{\prod_{x=2}^{i-1} a_x} \right) \geq 0 (3 \leq i \leq n),$$

$$c_3 l_i - c_i l_3 = c_3 l_3 \left( \frac{1}{\prod_{x=3}^{i-1} b_x} - \frac{1}{\prod_{x=3}^{i-1} a_x} \right) \geq 0 (4 \leq i \leq n),$$

.....,

$$c_i l_j - c_j l_i = c_i l_i \left( \frac{1}{\prod_{x=i}^{j-1} b_x} - \frac{1}{\prod_{x=i}^{j-1} a_x} \right) \geq 0 \quad (1 \leq i < j \leq n) \quad (11)$$

从公式(11)可以得到公式(1), 必要性得证。

充分性证明: 假设钢坯序列  $o = [(l_1, c_1), (l_2,$

$c_2), \cdots, (l_n, c_n)]$  满足公式(1) 且  $\sum v(o) = \sum c_i l_i (1 \leq i < j \leq n)$ 。当将  $o$  序列变换成任一种新序列  $o'$  后,  $o'$  中的任意  $i, j$  两项(其中  $1 \leq i < j \leq n$ ) 与  $o$  序列中的项只可能存在两种关系:

1) 两项前后顺序没有改变(即: 在  $o'$  序列中,  $i$  项在  $j$  项前);

2) 两个堆的前后顺序颠倒(即: 在  $o'$  序列中,  $i$  项在  $j$  项之后)。

在  $o'$  中, 若两项关系满足(1) 则在  $\sum v(o')$  中必会存在  $c_i l_i$  的项(其中  $1 \leq i < j \leq n$ ), 否则在  $\sum v(o')$  中必会存在  $c_j l_j$  的项(其中  $1 \leq i < j \leq n$ )。显然  $\sum v(o') - \sum v(o) = \sum (c_i l_j - c_j l_i)$  (其中  $1 \leq i < j \leq n$ ), 由公式(10)可以得出  $\sum (c_i l_j - c_j l_i) \geq 0$ 。因此在  $B = 1$  时, 满足公式(1) 的序列为最小序列, 充分性得证。

当  $B > 1$  时, 即只有多个存放区时, 若将数量为  $n$  钢坯堆序列置入空库中, 则可变路径总和:

$$\begin{cases} \sum v_i = A_b, n = 1 \\ \text{其中 } b \text{ 为该钢坯所置入的存放区号} \\ \sum v_i = \sum_{b=1}^B ((\sum v_j)_b), n > 1 \\ \text{其中 } (\sum v_j)_b \text{ 为第 } b \text{ 存放区的可变路径} \end{cases}$$

其中  $1 \leq b \leq B$ 。

在  $B > 1$  时, 多个存放区则可以表示为多个  $B = 1$  组成的存放区。根据定理 1 可知当  $\sum v_i$  最小时, 则每个存放区的钢坯堆都满足公式(1)。通过调整钢坯的入库顺序和选择存放区从而改变  $\sum v_i$ : 首先调整钢坯堆的入库顺序使其序列满足公式(1); 其次在钢坯入库时, 在  $B$  个存放区内寻找最近的存放区。通过以上两个步骤达优化  $\sum v_i$  目的。

### 3 算法设计与分析

#### 3.1 排序算法

根据上述分析知道当  $B = 1$  时, 若钢坯堆序列满足公式(1) 时, 则以该序列的顺序入库则使  $\sum v_i$  最小。通过调整钢坯的入库顺序, 使其入库序列满足公式(1), 从而缩短天车行走的距离。

该算法的形式化描述如下:

1)  $i := 1$ ; 2)  $j := i + 1$ ;

3) 若  $c_i l_j - c_j l_i < 0$  时, 交换  $i$  与  $j$  堆顺序, 否则转 4);

4)  $j := j + 1$ , 若  $j \leq n$  则转至 3), 否则转 5);

5)  $i := i + 1$ , 若  $i < n$  则转至 2), 否则终止。

从上面形式描述中可以看出,算法通过循环嵌套,两两比较是否满足  $c_i l_j - c_j l_i < 0 (1 \leq i < j \leq n)$ , 若满足则交换  $i$  与  $j$  堆的顺序。Sorting Operator 算法复杂度为  $O(n^2)$ 。

### 3.2 置入算法

按照存放规则,即遵循同钢号同长度的钢坯摆放在同一堆上,且每一堆的层数不能超过  $L$  (在文中所描述的问题中  $L = 11$ )。初始化时,按钢坯的类型分成若干堆满足  $n_k = \lceil H_k / L \rceil$  且  $y_d = L (\lceil \cdot \rceil$  为上取整), 其中  $1 \leq k \leq T, 1 \leq d \leq (n_k - 1), n = \sum_{k=1}^T n_k$ , 并将钢坯堆信息 (长度和层数) 赋给序列  $[(l_1, c_1), (l_2, c_2), \dots, (l_n, c_n)]$ , 则置入算法的形式化描述如下:

- 1) 初始化钢坯堆序列和  $n$ ;
- 2) 使用 Sorting Operator 算法生成新钢坯序列;
- 3)  $i := 1$ ;
- 4)  $a :=$  钢坯堆  $i$ ;
- 5) 查找离入口距离最小且满足堆  $a$  的区域;若找到,置入堆  $a$ , 否则转 6);
- 6)  $i := i + 1$ , 若  $i \leq n$  则转 4), 否则退出。

从以上的形式化描述中可以发现,初始化时对同类型的钢坯尽可能分成层数为  $L$  的堆,并使同类型的钢坯分成尽可能少的堆。因而在生产的全过程中,库房的利用率也高。利用 Sorting Operator 算子调整钢坯堆的入库顺序,使其入库序列满足公式(1)。置入算法主要是查找匹配与计算距离两个动作,它们的时间复杂度都为  $O(n)$ , 因此整个算法的时间复杂度仅为  $O(n^2)$ 。

## 4 数据测试

### 4.1 单存储区

实验在相应的参数约束下随机构造钢坯入库序列。实验的参数分为  $B = 1$  (即只有一个存放区),  $T = 3, A_1 = 0, n = 3, H_k \leq 11 (1 \leq k \leq T)$ 。为了验证定理 1, 实验将列出所有序列的  $\sum v_i$ 。实验将进行两组数据测试:

序列 1:  $[(6, 6), (5, 10), (4, 11)]$ ; 序列 2:  $[(5, 8), (7, 4), (3, 7)]$ ;

利用 Sorting Operator 算子将序列 1、2 分别置换成  $[(4, 11), (5, 10), (6, 6)]$  和  $[(3, 7), (5, 8), (7, 4)]$ 。其中表 1、表 2 中第一个序列分别为序列 1、序列 2 测试结果。

当  $B = 1$  时,利用 Sorting Operator 算法调整钢坯序列的顺序。序列 1、序列 2 的  $\sum_{i=1}^3 v_i$  分别为 94 和 56。实

验数据表明:Sorting Operator 可以得到入库的最小序列,缩短了天车行走的距离。表 1、2 中实验结果从数据上验证文中提出的定理 1。

表 1 序列 1 可变路径

入库顺序 序列编号	1	2	3	可变总路径 $\sum_{i=1}^3 v_i$
1	(4, 11)	(5, 10)	(6, 6)	94
2	(4, 11)	(6, 6)	(5, 10)	124
3	(5, 10)	(4, 11)	(6, 6)	109
4	(5, 10)	(6, 6)	(4, 11)	151
5	(6, 6)	(4, 11)	(5, 10)	166
6	(6, 6)	(5, 10)	(4, 11)	181

表 2 序列 2 可变路径

入库顺序 序列编号	1	2	3	可变总路径 $\sum_{i=1}^3 v_i$
1	(3, 7)	(5, 8)	(7, 4)	56
2	(3, 7)	(7, 4)	(5, 8)	92
3	(5, 8)	(3, 7)	(7, 4)	67
4	(5, 8)	(7, 4)	(3, 7)	104
5	(7, 4)	(3, 7)	(5, 8)	129
6	(7, 4)	(5, 8)	(3, 7)	140

### 4.2 多存储区

实验按参数的约束条件随机生成钢坯入库序列数据。当有多个存放区即  $B > 1$  时,钢坯入库序列的排列种数远远大于  $n!$ , 其中  $n$  为入库钢坯的总数量。实验将按每组参数进行 10 次数据测试,列出按某一随机顺序入库和 Loading Operator 算法所得到的  $\sum v_i$  进行比较,并引入缩短距离  $\Delta D$  和缩短百分比  $\Delta P$  的定义。

记随机顺序入库所得的随机序列的  $\sum v_i$  为  $S_r$ , 运用文中所提出的 Loading Operator 算法求出的  $\sum v_i$  记为  $S_l$ , 则:

$$\text{缩短距离 } \Delta D = (S_r - S_l)$$

$$\text{缩短百分比 } \Delta P = \Delta D / S_r \times 100\%$$

第 1 组参数:  $n = 10, B = 2, A_b = 0 (1 \leq b \leq B), T = 10, L = 11, H_k \leq L (1 \leq k \leq T)$  生成  $[(l_1, c_1), (l_2, c_2), \dots, (l_{10}, c_{10})]$ , 该序列满足  $1 \leq l_i \leq 11$  和  $1 \leq c_i \leq 11$ , 测试结果如表 3 所示。

第 2 组参数:  $B = 2, L = 11, A_b = 0 (1 \leq b \leq B), T$  为随机数 ( $10 \leq T \leq 100$ ) 生成的随机序列, 由该算法的初始化处理后得到  $[(l_1, c_1), (l_2, c_2), \dots, (l_{100}, c_{100})]$ , 该序列满足  $1 \leq l_i \leq 101$  和  $1 \leq c_i \leq 11$ , 测试结果如表 4 所示。

表 3 中是对第一组参数的约束下进行 10 次随机测试结果, 其中平均、最大、最小缩短比依次为: 33.60%、50.14%、20.89%。表 4 中是对第二组参数约束下进行

10 测试的结果,其平均、最大、最小缩短比依次为:44.38%、52.92%、27.38%,从表 3、表 4 的实验结果可以看出,当钢坯的长度增大和数量增加时  $\Delta D$  也越大,更能体现文中算法中的路径优化的高效性。

表 3 第 1 组参数生成的随机序列测试结果  
( $n = 10$ )

序号	$S_r$	$S_l$	$\Delta D$	$\Delta P(\%)$
1	1831	913	918	50.14
2	1765	1251	514	29.12
3	1290	840	450	34.88
4	1412	1117	295	20.89
5	1604	1146	458	28.55
6	2159	1472	687	31.82
7	1120	825	295	26.34
8	2114	1513	601	28.43
9	2233	1223	1010	45.23
10	1753	1042	711	40.56

表 4 第 2 组参数生成的随机序列测试结果  
( $n = 100$ )

序号	$S_r$	$S_l$	$\Delta D$	$\Delta P(\%)$
1	1474982	694410	780572	52.92
2	1358949	686325	672624	49.50
3	1485699	808629	677070	45.57
4	1532533	819342	713191	46.54
5	1459009	778368	680641	46.65
6	215345	156375	58970	27.38
7	854254	598251	256003	29.97
8	1311793	663942	647851	49.39
9	1618715	798554	820161	50.67
10	1435062	787925	647137	45.09

## 5 结束语

无论是产品生产单位还是物品运输中转港口都需

要解决无限的存储对象与有限的存储空间的问题。不依赖于装箱(或者入库)对象、能够最大限度优化运输路径、提高存储空间利用率的方法是提高工作效率与资源利用率的先决条件,也是规范作业程序、降低运营成本、提高企业竞争力的有效手段。文中正是针对上述目标,建立了钢坯入库路径优化问题的模型,从理论上推证出最小序列存在的性质,并以该性质为基础进行了算法设计。在此基础上,进行了单库房和多库房可变短路径的数据测试。仿真实验结果一方面佐证了文中提出并证明的性质定理,另一方面也表明了文中所设计算法的高效性。

## 参考文献:

- [1] 李 荣. 多重群体遗传算法在装箱问题中的应用研究[J]. 计算机技术与发展, 2007, 17(9): 247-249.
- [2] Mukhacheva E A, Mukhacheva A S. The rectangular packing problem: Local optimum search methods based on block structures[J]. Automation and Remote Control, 2004, 65(2): 248-257.
- [3] Imahori S, Yagiura M, Ibaraki T. Local search algorithms for the rectangle packing problem with general spatial costs[J]. Mathematical Programming, 2003, 97(3): 543-569.
- [4] 刘嘉敏, 马广娟, 黄有群. 基于组合的三维集装箱装入启发式算法的研究[J]. 工程图学学报, 2005, 26(1): 22-25.
- [5] Karabulut K, Inceoglu M. A hybrid genetic algorithm for packing in 3D with deepest bottom left with fill method[C]//Proc of the 3rd Int Conf on Advances in Information Systems. Turkey: Tatyana Yakhno, 2004: 441-450.
- [6] 刘志新, 李建国, 谢金星. 约束入库问题模型与算法研究[J]. 控制理论与应用, 2003, 20(1): 149-152.
- [7] 刘 辉. 装箱问题的概率近似算法[J]. 科学技术与工程, 2007, 7(13): 3279-3282.

(上接第 195 页)

视图,从而隔离应用与底层数据源,并通过统一的 API 存取所有数据源。建立业务数据模型的过程,就是数据服务抽象的过程;不同阶段、不同层次模型的转换和映射的过程,就是服务封装和组合的过程;而模型间的转换模式和映射模式,就是服务调用和响应的模式;而整个过程都是按照一定业务策略来进行的。如何以业务数据模型为基础和驱动,逐步推进虚拟学习社区协同系统的建设,将是继续研究的内容。

## 参考文献:

- [1] 何 玲, 刘惠芬. 网络学习社区研究动向辨析[J]. 现代远程教育研究, 2004(1): 45-50.
- [2] 张礼华, 卢道华. 基于多 Agent 的网络课程协作学习平台的构建与研究[J]. 计算机技术与发展, 2006, 16(9): 132-

134.

- [3] 叶 斌. 基于多 Agent 的高校教学评测系统模型研究[J]. 计算机技术与发展, 2007, 17(11): 225-227.
- [4] 谢幼如, 刘铁英. 网络课程的内容分析与评价研究[J]. 电化教育研究, 2003(11): 47-51.
- [5] Thomas E. Service-oriented architecture: a field guide to integrating XML and Web services[M]. [s.l.]: Prentice Hall PTR, 2004.
- [6] Brown A W, Delbaere M, Eeles P, et al. Realizing service-oriented solution with the IBM Software Development Platform[J]. IBM Systems Journal, 2005(10): 727-730.
- [7] Hubert R. Convergent architecture: Building model-driven J2EE systems with UML[M]. [s.l.]: Wiley, 2001.
- [8] 汪 琼, 李晓明. 网络课程实施问题分析[J]. 电化教育研究, 2002(1): 15-18.