

基于遗传算法的集装箱后方堆场箱位分配策略*

王志明, 符云清

(重庆大学 计算机学院, 重庆 400044)

摘要: 根据重庆港实际情况建立了以提箱时间为制约因素,以最小化翻箱率为目标的集装箱后方堆场箱位分配模型,并针对模型提出了基于遗传算法的解决方案。方案对一次卸船或者进港的一批箱进行全局优化,并考虑其分配对后续集装箱的影响。遗传算法迭代过程中采用适应度函数指数变换防早熟,采用可行解替换法处理约束,并设计最优解保存策略保证最终的优化效果。最后针对实际堆场的不同规模,对方案的优化结果同文献中的其他遗传算法方案进行比较,证明了本文优化策略的优越性和实用性。

关键词: 遗传算法; 集装箱; 优先级; 翻箱率

中图分类号: TP391; U656.1 + 35

文献标志码: A

文章编号: 1001-3695(2010)08-2939-03

doi:10.3969/j.issn.1001-3695.2010.08.035

Location allocation strategy for rear yard based on genetic algorithm

WANG Zhi-ming, FU Yun-qing

(School of Computer, Chongqing University, Chongqing 400044, China)

Abstract: Based on the practice application of Chongqing harbour, proposed a assignment model for container in rear yard firstly, which was limited by time of picking up container and aimed at the minimal container-turned rate. According to the above model, then addressed a genetic algorithm-based solution. A batch of containers were global optimized in the solution, and its effect on follow-up containers was also considered. In iterative process of genetic algorithms, used the fitness function exponential transform to avoid early-mature, used a replacement method to solve constraint, and used a strategy of saving best solution to ensure optimization of the final result. Finally, proved the proposed solution advantage and practicability by comparison with other genetic algorithms to which referred in different actual size of yard.

Key words: genetic algorithm; container; priority; rate of turning container

集装箱堆场是船舶装卸作业与内陆提箱作业间的一个缓冲区,起到临时储存集装箱的作用。随着码头集装箱吞吐量不断增加,堆场系统对整个集装箱码头作业能力和效率瓶颈的作用越来越明显,优化堆场系统作业已成为重要的研究课题。

码头堆场一般分为前方堆场和后方堆场。前方堆场是用来暂存出口重箱的,要进行装箱作业的集装箱会先堆放到靠近泊位的前方堆场;后方堆场则是进行重箱或空箱交接、保管和堆存的场所,卸船的集装箱一般堆存在后方堆场,以方便货主进行提箱。以前的研究大部分以海港为模型,主要集中于前方堆场在不同箱流量下采用无缓冲区(动态)和建立缓冲区(静态)两种策略下出口箱箱位分配问题;Kim^[1]通过考察不同码头堆场的布局及其对堆场作业过程中平均倒箱次数的影响,建立了回归方程对集装箱堆存情况进行简化求解;Kim和Park^[2]研究了提高堆场空间利用率和装船作业效率的出口箱堆放空间动态分配方法,提出了基本的混合整数规划模型,并通过算例比较了近视算法和次梯度法两种启发式算法;国内的刘丽芸^[3]对出口箱堆场倒箱操作的落箱位置进行研究,对该问题建立了数学模型,提出了新的启发式算法,求解得到了较好的结果;陈庆伟^[4]以前方堆场出口集装箱贝位堆存为模型,考虑集装箱重量、目的港因素,以装船最小化倒箱数为目标,通过遗传算法求解箱位分配。

本文将集中于目前研究比较少的集装箱后方堆场堆存策略,以重庆港为例,考虑提箱时间因素,以最小化翻箱率为目标,利用遗传算法求解到场箱的分配箱位;并且区别于文献[4,5]的每次只对一个待分配箱进行优化的遗传算法,结合重庆港的卸船实际,将每次优化的目标扩大为一批待分配箱。

1 集装箱后方堆场箱位分配模型

以重庆寸滩港为例,堆场被划分为若干区,每区又被划分为若干段。每段包括20~30贝,每贝有20列5层共100个箱位。具体情况如图1所示,与屏幕垂直方向的是贝(bay),与屏幕平行的左右方向是行(row),与屏幕平行的上下方向是层(tier)。实际工作中,翻箱是不可避免的。考虑到跨贝翻箱会大大增加作业成本,一般堆场都会在一贝预留2~3列空位作为缓冲,保证翻箱都在同一贝中。本文的后方堆场箱位分配模型就是对一次卸船或者进港的一批箱子,在指定贝中分配箱位,力求最终整贝翻箱率最低。

1.1 模型假设条件

集装箱后方堆场箱位分配模型基于以下假设建立:a)倒箱只发生在同一贝中;b)同一贝中只放同一类箱型的箱,即不考虑20尺和40尺箱混堆的情况;c)在为新到来的箱安排箱位的时候,对于已经堆存好的集装箱不进行倒箱;d)假定货主必

收稿日期: 2010-02-02; 修回日期: 2010-03-02

作者简介: 王志明(1984-),男,山西朔州人,硕士研究生,主要研究方向为集装箱堆场作业优化(259277277@qq.com);符云清(1969-),男,教授,博士,主要研究方向为计算机协同、网络安全、远程教育、行业信息化。

须在约定的时间内提箱,并且由港口根据约定时间安排车辆运送。

1.2 集装箱优先级设定

以重庆港的实际,影响堆场翻箱的因素是提箱时间,需早一点提走的箱子尽量放到上层可以减少翻箱,所以考虑将集装箱的提箱时间反映为集装箱的优先级,具体方案如下:

a) 以每一贝集装箱的进场时间 $t_0, t_1, t_2, \dots, t_N$ 中最小值 t_{\min} 为基准,当前贝内集装箱的优先级为该箱提箱时间 t_p 与 t_{\min} 的差值 Δt ,当基准箱被提走后,更新 t_{\min} 为剩下箱中最早到场的时刻;b) 考虑重庆港实际情况以及遗传算法效果,本文决定以半天 12 h 为一个单位,即差半天为差一个优先级;c) 重庆港允许的集装箱最大堆存时间为 15 d,所以堆场内集装箱优先级的取值为 $[1, 60]$ 。表 1 为优先级对照表, t_p 表示提箱时间。

表 1 集装箱优先级表

t_p 与 t_{\min} 差 Δt	$\Delta t < 12h$	$12h < \Delta t < 24h$...	$708h < \Delta t < 720h$
优先级	1	2	...	60

1.3 数学模型

a) 整个贝位为一个矩阵 $M[\text{row}][\text{line}]$, 矩阵元素 $M[i][j] \in \{0, c\}, 0 \leq i < \text{row}, 0 \leq j < \text{line}, 0$ 表示空位, c 为该位置集装箱的优先级;

b) $\text{class} = \{1, 2, 3, \dots, c, \dots, 60\}$, 集装箱优先级集合, 数值越小, 优先级越高;

c) $\text{stack} = \{1, 2, 3, \dots, c, \dots, \text{row} - 1\}$, 贝内列的集合;

d) $x(d, r) \in (0, 1)$, 取值为 1 表示从 d 列到 r 列发生一次倒箱;

e) $M[i][j] = c, M[i+1][j] \neq 0$, 约束条件, 表示集装箱不能悬空;

f) $\min \sum_{d \in \text{stack}} \sum_{r \in \text{stack}} x(d, r)$ 为目标函数, 表示倒箱量最少。

2 集装箱后方堆场箱位分配遗传算法

本文的遗传算法希望得到的是一个全局最优的结果,所以在考虑对待分配箱位时,要同时考虑其分配策略对其后随机到来的集装箱的影响。优化的目标是,整个贝位满箱时翻箱量最少。

2.1 编码及染色体描述

本算法采用实数编码,基因就用集装箱的优先级来表示。每一个染色体对应一批待分配箱的一个分配方案。

染色体描述如下:a) 染色体长度为矩阵 M 中 0 的个数,即贝内空位的数量;b) 对初始矩阵按先行再列遍历,每个为 0 个元素与应染色体中的基因位顺序对应;c) 当待分配箱的数量小于矩阵中 0 位的数量时,染色体中基因由待分配箱优先级以及随机生成的 1~60 的随机数构成。

2.2 适应度函数

参与运算的是初始状态矩阵与染色体相结合产生的矩阵,对该矩阵计算精确翻箱量比较困难,因为翻出的箱子不确定会放到哪一列,对新的列产生如何的影响不好估计,所以本文采取一种近似的计算,忽略翻箱对其他列的影响,也可以考虑为有一个专门的缓冲区用来翻箱,其实际的情况当整个贝位列较多、堆存被优化后翻箱量较小的情况下,往往对于翻出的箱大多可以找到一个不增加翻箱量的位置来。

本文计算压箱量近似为翻箱量,翻箱量计算方式如下:

```
f(x) = 0;
for(s = 1; s <= maxline; s++) {
    while(h(s) > 1) {
        f(x) += (h(s) - h(maxclass));
        h(s) = h(maxclass) - 1; }
}
```

其中: $h(s)$ 表示第 S 堆垛列的高度, $h(\text{maxclass})$ 表示该列中最高优先级箱子的高度。

为了保证目标函数的优化方向对应于适应度值增大的方向,取适应度函数为目标函数的倒数。

为了防止遗传算法迭代过程中过早收敛(早熟),对适应度函数进行指数变换,最后确定的适应度函数为 $F(x) = 1 / (1 + \exp(f(x)/1000))$, 其中 $f(x)$ 为目标函数。

2.3 初始化种群

本遗传算法是单目标、多约束的,在迭代过程中会产生不少的不可行解,为了提高遗传算法的计算效果,一般在初始化种群的时候采取一定的措施使得随机产生的染色体都为可行解,本文也将采用这种方式。

具体产生可行解算法描述如下:a) 计算染色体长度 = 矩阵 0 元素数量,初始化染色体基因全为 0;b) 随机选取一个待分配箱,将其分配至随机的一个可分配位置(不悬空),并将该位置对应的基因更新为该箱的优先级,循环执行 b),直至分配完所有的待分配箱;c) 对染色体中为 0 的位置分配一个 1~60 的随机数。

2.4 遗传操作

本文采用比例选择法中的轮盘赌选择,即适应度越高的个体被选择的概率越大。个体被选择的概率为 $P_i = f_i / \sum_{i=1}^N f_i$ 。其中: P_i 为第 i 个个体被选择概率, f_i 为第 i 个成员的适应度。

选择后的个体进行两两配对,配对后的个体进行交叉,考虑本文约束复杂,采用对个体破坏性较小的单点交叉。

变异方式:针对每个基因,产生一个 $(0, 1)$ 的随机数,若随机数小于变异概率,就将该基因与染色体中随机的另一个基因对换。

2.5 重要算子的选择

根据参考文献以及多次的实验,决定算子数值如下:

种群规模 $N = (50, 100)$, 进化代数 $M = 30 \sim 50$, 交叉概率 = 0.6~0.8, 变异概率 = 0.02。

2.6 约束处理

集装箱后方堆场箱位分配模型是一个多约束的组合优化问题,遗传进化过程中可能产生不符合约束条件的不可行解。本文将采用可行解替换法处理不符合约束的不可行解,即对每次遗传操作后产生的不可行解用随机产生的可行解替换。

2.7 最优解保存策略

遗传算法进化过程中,由于选择、交叉、变异等遗传算子操作的随机性,虽然随着种群的进化过程会产生越来越多的优良个体,但它们也有可能导致当前种群中适应度最高的个体丢失。针对这种情况,可以使用最优保存策略进行优胜劣汰选择,使历史最优个体总保存在当前种群中。具体做法:每次遗传操作后,将种群中最优个体与历史最优个体进行比较,若适应度强过历史最优个体,则更新历史最优个体;否则,用历史最优个体替换本种群中适应度最差的个体。

2.8 遗传算法流程

遗传算法流程如图 2 所示。

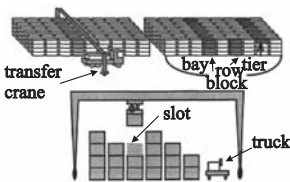


图1 实际堆场贝位示意图

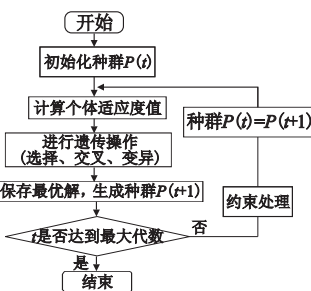


图2 遗传算法流程

3 算例分析及优化对比

3.1 算例分析

本文在 J2EE 环境下实现了此遗传算法,并编写程序模拟堆场堆箱的实际情况。具体从空贝开始,每次随机到达 1~20 个箱子,利用遗传算法优化分配箱位,最终得到满箱时的贝位情况,并计算其翻箱量。

这里展示的算例为 5 层 10 列的贝的运行结果,如图 3 所示。

3.2 对比文献中的遗传算法

文献[4,5]的前方堆场动态模型与本文的后方堆场模型类似,但遗传算法的应用上有区别。文献[4,5]采用的是并行遗传算法,每次只针对一个待分配箱进行优化,整个算法过程大致如下:假设待分配箱优先级为 C,预分配的贝位有 F 个空位,待分配箱可选的位置有 K 个(1,2,...,i,...,K),那么子种群数为 K,染色体长度为 F;第 i 个子种群代表待分配箱将放到第 i 个待分配位置,该子种群染色体构成为第 i 个待分配位置对应的基因位为 C,其他基因位为随机的优先级;这样每个子种群就代表了一个可选位置,各子种群经过遗传迭代之后,比较各子种群个体的平均适应度,第 i 个子种群平均适应度最高,就代表待分配箱放到第 i 个待分配位置最优。

文献[4,5]的遗传算法与本文的区别主要是其每次只针对一个待分配箱进行优化,对于卸船的一批箱只能拆分为一个的集装箱来考虑,而本文对一批待分配箱是进行一次优化,而且具体的算法过程也因为优化实体的不同区别很大。

文献[4]中有一个算例:6 列 4 层的贝位初始为空,随机到达了 21 个重箱为 {11, 23, 42, 31, 23, 12, 23, 41, 32, 42, 52, 23, 21, 42, 52, 33, 53, 12, 51, 41, 21}, 文献[4]的遗传算法优化的结果翻箱量为 3,本文也对这个算例进行了优化,其中 100 次优化中,81 次得到的翻箱量为 2。图 4 为其中一次的优化结果。针对这一个算例,本文遗传算法优于文献[4]。

到场箱列表: {23 21 38 37 25 43 31 51 10
 58 4 46 16 51 35 7 41 9 18}
 到场箱列表: {39 35 7 27 16 45 3 15}
 到场箱列表: {16 22 58 10 4 46 42 39}
 到场箱列表: {9 33 38 10}
 到场箱列表: {49 18 25 19 49 54}
 到场箱列表: {32 39 41 16 19}
 最终状态:
 10 38 49 42 16 4 19 15 9 10
 16 25 18 3 19 22 32 39 33 46
 16 7 58 46 16 35 49 41 27 25
 18 45 9 4 39 7 38 41 39 37
 31 54 10 21 23 43 58 51 51 35
 最终翻箱量:15

最终状态:—————
 0 42 12 31 0 11
 0 23 12 21 23 41
 21 23 33 32 23 51
 53 41 42 42 52 52
 最终翻箱量:2

图4 优化结果为2的例子

图3 5层10列运行结果

本文同样实现了文献[4,5]的遗传算法。这里分别将文献[4,5]的遗传算法与本文的遗传算法应用于 3.1 节的模拟

堆场环境中,在与重庆港各码头实际对应的不同贝位规模下,对比两种遗传算法的优化效果,结果如表 2 所示。定义翻箱率为:为移动目标箱而多移动的非目标箱的次数除以该贝位总的箱数。表 2 中的翻箱率均为执行 100 次优化后的平均值。

表 2 两种遗传算法优化效果对比

实际码头	贝位规模	文献遗传算法	本文遗传算法
		优化后的翻箱率 (算法 1)	优化后的翻箱率 (算法 2)
涪陵	4 层 6 列	44.50%	21.79%
万州	4 层 10 列	44.03%	25.75%
九龙坡	4 层 16 列	44.29%	29.23%
寸滩	4 层 20 列	43.58%	30.73%
寸滩	5 层 20 列	50.65%	39.12%

本文将文献[4,5]的遗传算法叫做算法 1,将本文的遗传算法叫做算法 2。由表 2 的对比可得出算法 2 的优化效果在堆场实际的规模内是优于算法 1 的,其缘由不难得出。假设对一批 M 个待分配箱进行优化,算法 1 会将 M 个箱拆分为一个的箱子来考虑,这样在对第 i 个箱子的箱位进行优化时,贝位里的 N 个空位有 N-i-1 个将被放入随机的优先级,但对于这批具体的待分配箱,这 N-i-1 个空位中有 M-i 个位置的优先级其实是确定的。这无形中将本来确定的优先级用不确定的优先级来代替,必然影响最终的优化效果。算法 2 是一次对全部 M 个待分配箱进行优化,这样贝位里 N 个空位中 M 个空位是确定优先级,N-M 个空位用不确定的随机优先级来代替,这样最后优化的结果明显会优于算法 1。

当然,算法 1 也有它明显的优点,就是优化效果稳定。在表 2 的对比中就可以看出,随着贝位规模的增长,算法 1 始终保持稳定的翻箱率,而算法 2 却有明显的跨度。但总体在堆场实际的规模中,算法 2 都是优于算法 1 的,所以在实用性上肯定是算法 2 更好。

4 结束语

本文首先对后方堆场箱位分配进行了数学建模。然后在模型的基础上提出了一种基于遗传算法的优化方案,并在其后对遗传算法各个步骤进行了优化。最后通过一个算例的展示以及与其他遗传算法的比较,说明了本文遗传算法的有效性和实用性。

参考文献:

[1] KIM K H. Evaluation of the number of rehandles in container yard [J]. Computers and Industrial Engineering, 1997, 32(4): 701-711.

[2] KIM K H, PARK Y M. Deriving decision rules to locate export containers in container yards [J]. European Journal of Operational Research, 2000, 124(2): 89-101.

[3] 刘丽芸. 集装箱码头堆场倒箱问题优化算法及仿真 [D]. 大连:大连海事大学, 2008: 5-50.

[4] 陈庆伟. 基于遗传算法的堆场贝位分配优化问题研究 [D]. 青岛:青岛大学, 2007: 8-42.

[5] 周留井. 基于遗传算法的进口箱堆场 BAY 位分配优化模型研究 [D]. 广州: 中山大学, 2008: 1-54.

[6] 陈庆伟, 王继荣. 集装箱堆场出口箱堆存模型及其算法 [J]. 物流科技, 2007, 40(11): 350-352.

[7] 计三有, 高悦文. 集装箱堆场减少倒箱率方法研究 [J]. 水运工程, 2006, 30(8): 53-61.

[8] 周明, 孙树栋. 遗传算法原理及应用 [M]. 北京: 国防工业出版社, 1999: 32-64.