

基于遗传算法的 Unity3D 装箱系统设计

赵治羽,陈倩云,李婷婷,张 娟

(西南科技大学信息工程学院,四川 绵阳 621010)

摘要:针对物流配送的装箱问题,本文设计一款新型的基于遗传算法和 Unity3D 物理引擎软件系统。与传统的遗传算法相比,该系统可以方便快捷地计算出基于多条件的合理装箱方案,即如何摆放货物才能最大限度地利用集装箱空间。该系统应用 Unity3D 添加的手动装载功能和装箱方案导出功能,通过 JS 脚本设计物体的运动及基本操作过程。仿真结果表明,该系统既能高效地计算出一套完整的装箱方案,又能通过人为的经验加以调整,从而实现更好的装箱方案。

关键词:Unity3D; 遗传算法; 装箱

中图分类号:TP39

文献标识码:A

doi: 10.3969/j.issn.1006-2475.2014.06.032

Design of Unity3D Packing System Based on Genetic Algorithm

ZHAO Zhi-yu, CHEN Qian-yun, LI Ting-ting, ZHANG Juan

(School of Information Engineering, Southwest University of Science and Technology, Mianyang 621010, China)

Abstract: Aiming at the problem of the container logistics, this paper designs a software system based on Genetic algorithm and Unity3D physics engine. The system calculates reasonable packing scheme based on multi conditions through the genetic convolution algorithm, namely how to place the goods in order to maximize the use of container space. The system adds manual loading and packing plan derived function through the Unity 3D, so that the final scheme more reasonable, humanization, greatly improve the packing efficiency and container space utilization.

Key words: Unity3D; Genetic algorithm; packing

0 引 言

三维装箱问题在飞机装仓、码头装货、一般容器装载等诸多物流配送领域有着广泛的应用。三维装箱问题属于 NP-hard 问题,国内外专家和学者提出了很多启发式算法和智能优化算法,如 Fuellerer 等分别采用双层禁忌搜索算法和蚁群与启发式算法结合的方法优化路径成本和货物装载的顺序和方向^[1]。张德富等采用启发式算法与模拟退火相结合的方法对三维装载进行优化^[2]。孙隼华采用逐次放置货物的放置方式和遗传算法相结合的方式实现了货物三维装载的装载率优化^[3],但并没有考虑货物的放置方向、重心、承重等因素。

本文采用遗传算法和 Unity3D 物理引擎设计三维装箱软件系统,该系统特有的手动装箱功能和装箱方案导出功能,极大地提高了装箱效率和集装箱的空

间利用率。

1 基于遗传算法的装箱算法

三维装箱问题属于不确定多项式问题,随着问题规模的不断扩大,不能在有限的时间中找到精确解。遗传算法是解决搜索问题的一种通用算法,对于各种通用问题都可以使用。基于遗传算法的装箱问题,最核心的部分是遗传因子的设计,该设计直接关系到算法的收敛速度,即程序的运行效率。另外,适应度函数设计的好坏直接影响最终方案的优劣。

1.1 遗传因子的选择

在遗传因子的设计当中,基本策略是优先将同种货物合并堆放,由于同种货物尺寸相同,所以不会出现较多的空间浪费,更容易收敛。但这种方式生成遗传因子的过程比较麻烦,如果遇到货物种类较多,每种货物个数较少则无法发挥遗传因子的特点,相反,

收稿日期:2014-03-20

基金项目:西南科技大学博士基金资助项目(122X7127);西南科技大学实验室开放基金资助项目(13xnf43)

作者简介:赵治羽(1993-),男,四川绵阳人,西南科技大学信息工程学院本科生,研究方向:智能学习;陈倩云(1991-),女,四川绵阳人,本科生,研究方向:智能学习。

会导致计算过程繁琐。在本设计中,由于装箱货物种类较少,每种货物个数较多,为了减少空间浪费,提高空间利用率,采用如下方案设计遗传因子:

$\{[kind_1, dir_1, num_1], [kind_2, dir_2, num_2], \dots, [kind_k, dir_k, num_k]\}$

其中,每一对括号的内容为一组基因信息, $kind_1 \sim kind_k$ 代表当前这组中使用货物的种类,即初始化时不是按照货物个数进行编号,而是按照货物种类编号,并记录每种货物的个数; $dir_1 \sim dir_k$ 代表当前组内货物的放置方向; $num_1 \sim num_k$ 代表当前组内共有多少个同种货物。

1.2 适应度评价函数设计

适应度评价函数设计的优劣将直接影响遗传算法的优劣。在本设计中,最终目标是使集装箱空间利用率尽可能高,所以目标函数为放入集装箱货物的总体积与集装箱体积之比,即:

$$y = \frac{\sum_{i=1}^n v_i}{V}$$

其中, n 代表放入集装箱的货物总数, v_i 代表第 i 件货物的体积, V 代表集装箱体积, y 为所求得的适应度。

1.3 罚函数设计

目前对于约束条件的处理(包括单目标和多目标优化问题),主要采用各种罚函数方法。这种方法通过对不可行解的惩罚将约束问题转化为无约束问题。任何对约束的违反都要在目标函数中添加惩罚项。通过这一机制,迫使决策方案朝可行解区域进化。

本设计中罚函数实现的功能是针对不可行的方案进行空间惩罚,即在原适应度函数的基础上减去一个罚值,所以属于加形式的罚函数,适应度函数可改写成如下形式:

$$y = \frac{\sum_{i=1}^n v_i}{V} + \partial g(x)$$

1.4 三维装箱算法设计

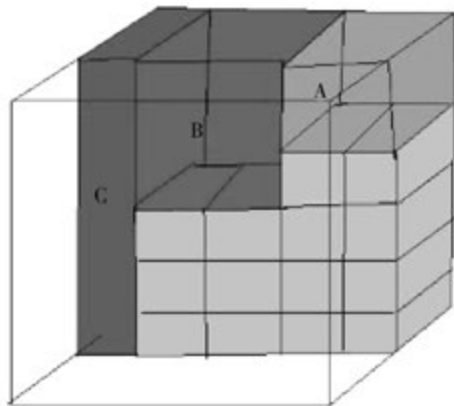


图1 算法原理示意图

当某货物入库后,可能剩余空间 A,B,C(如图1所示)。通过计算,获取剩余空间 A,B,C 的长、宽、高和某一角的坐标,并以这一点为放置货物的起点坐标。由于已知货物的长宽高,所以能获取每个货物的具体位置。

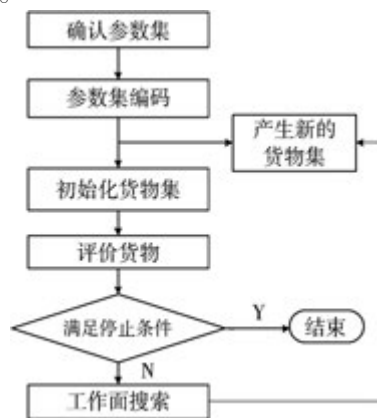


图2 算法过程

算法过程如图2所示。

(1)工作面拆分搜索。将仓库分割成 n 个工作面,第一个工作面的起点坐标为 $(0,0,0)$,往后工作面起点坐标为 $(start_x + goods_x, 0, 0)$, $start_x$ 为上一工作面的坐标, $goods_x$ 为上一行货物的长度。

(2)当在某工作面放入第一种货物时,便会有剩余工作面。根据放入的货物数量,计算剩余工作面的起点坐标和尺寸(即长、宽、高)。

(3)剩余空间拆分搜索。调用剩余工作面函数,若没有可以放入剩余工作面的货物,则返回,进入下一工作面;若有则将货物放入,不断递归调用剩余工作面函数,直到没有可以放入的货物。

2 基于 Unity3D 的装箱功能实现

Unity3D 是由 Unity Technologies 公司开发的一款多平台的游戏开发工具,由于其优秀的性能以及丰富的开发功能,Unity3D 已被广泛运用于虚拟现实的研究领域。本文设计的装箱系统使用 Unity3D 物理引擎软件。

2.1 装箱功能设计流程

装箱功能设计流程如图3所示。

(1)在场景中创建集装箱及其缩略图,调整主摄像头以及缩略图的摄像头置其对应的集装箱立方体位置,并设置其方向。

(2)读取文件中货物的数据,同时创建货物对象,并根据读取的数据,将其摆放在集装箱中合适的位置,再判断读取的内容为空值后停止读取文件。

(3)静态场景生成后,在程序运行过程中判断是否有鼠标点击,若是右键点击,则根据其滑动的轨迹

旋转摄像头的角度;若是左键点击货物,则根据其运动的轨迹移动货物。

(4) 保存过程中,将原状态数据写入新建的 `outputold.txt` 文件中,作为上一次的状态数据,然后将修改后的货物位置存入 `output.txt` 文件中。下次运行程序时,将自动读取修改后的数据,即 `output.txt` 中的文件数据。

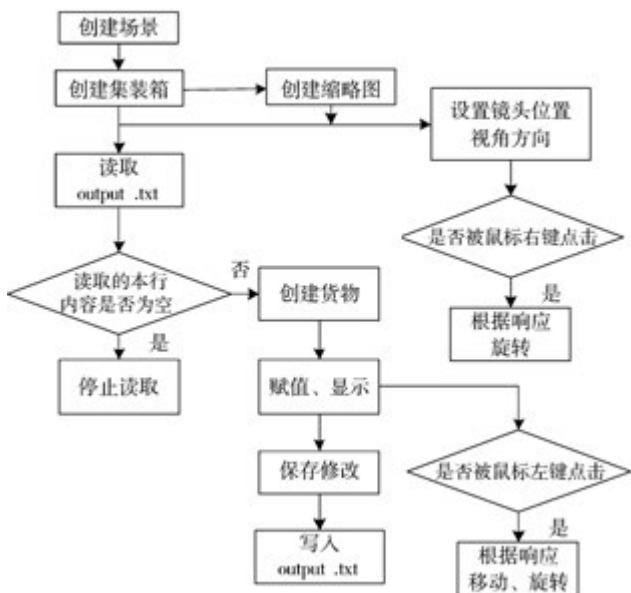


图3 装箱功能设计流程



图4 方向块

2.2 货物的拖动以及旋转

(1) 货物的拖动。

货物点击方向块如图4所示,拖动步骤如下:

Step1 创建 `Move_Goods` 脚本,将其添加到货物体上。脚本主要实现鼠标左键的点击监听、方向轴的显示,并可拖动货物。

Step2 创建 `Move_Way` 脚本,将其添加到货物体上。脚本主要实现方向块的显示和隐藏,并设置方向块的位置,方向块是指鼠标点击的货物体,六边显示代表 X, Y, Z 轴的圆柱形物体。

Step3 重写 `OnMouseDown()` 方法。在鼠标点

击时,找到 `MoveCenter` 游戏体下的 `Move_Way` 脚本,调用 `getMoveWay()` 方法,将此被点击的货物体作为参数传入。

Step4 获取当前帧鼠标坐标位置,并转换为三维坐标,再加上物体位置与鼠标之间的距离,赋值给变量 `curPosition`。判断是否为初始位置,若是则 `curPosition` 的 x, y, z 为当前点击的货物体坐标,然后将 `curPosition` 的坐标位置赋值给当前点击的货物体实现移动。

(2) 货物旋转。

货物旋转步骤如下:

Step1 创建 `GoodsMessage` 脚本,将其添加到货物体上。该脚本实现了鼠标左键和键盘按键的点击监听。

Step2 在 `OnMouseDown()` 方法中不断循环判断获取鼠标是否为点击状态,同时判断获取键盘 X, Y, Z 键的按下状态,如果按下则向其相应方向旋转,如绕 X 轴旋转:

```
if (Input.GetKeyUp(KeyCode.X)) { this.transform.Rotate(90, 0, 0); }
```

2.3 镜头旋转缩放功能

镜头的旋转缩放功能实现方法:创建 `MyCamera` 脚本,将其添加在 `Main Camera` 摄像头上。在 `Update()` 方法中实现滚轮缩放和镜头旋转功能。

(1) 镜头缩放。

获取鼠标滚轮的滑动再乘以 400 表示距离:

```
Input.GetAxis("Mouse ScrollWheel") * 400
```

获得滚轮的距离后重新设置镜头的位置:

```
transform.Translate(Vector3.forward * distance)
```

(2) 镜头旋转。

在鼠标右键点击时获取点击的位置,并记为 `p1`,在鼠标移动时记录下当前帧鼠标的位置 `p2`,将 `p2` 的 X 和 Y 坐标减 `p1` 的 X 和 Y 坐标,将它们的差 `dx`, `dy` 作为参数传入 `RotateAround()`:

```
transform.RotateAround(Center.transform.position, Vector3.up, dx * 0.5);
```

```
transform.RotateAround(Center.transform.position, Vector3.back, dy * 0.5);
```

使镜头以 `Center` 的位置为中心点,横向或者竖向旋转 `dx × 0.5` 或者 `dy × 0.5` 的距离。

监听获得键盘 4 个方向键的输入,并实现向相应方向的旋转,如向上旋转:

```
if (Input.GetKey("up")) {
```

```
transform.RotateAround(Center.transform.position, Vector3.back, 1); }
```

3 仿真实验结果及分析

实验环境为:Pentium (R) Dual-Core CPU 2.8GHz, 1.96G 内存,采用面向对象的 C#语言实现遗传算法,并采用 Unity3D 技术对算法的结果进行展示,实现三维装箱结果的可视化显示。

3.1 参数设置

所选的容器信息如表 1 所示,待装箱货物信息如表 2 所示。货物自身的约束条件如表 3 所示,这些约束在计算时均要获取,加入判断依据。

表 1 选择的容器信息

选择	类型	长 (mm)	宽 (mm)	高 (mm)	载重 (g)
√	Htt 12/501 40-Binnencontainer	12000	2440	2440	12020197

表 2 待装箱货物信息

选择	名称	长 (mm)	宽 (mm)	高 (mm)	重量 (g)	数量
√	DemoBox1	2000	1200	500	35000	20
√	DemoBox2	320	270	250	6000	200
√	DemoBox3	400	400	400	25000	2
√	DemoBox4	300	250	150	800	60
√	DemoBox5	800	1200	1200	86000	10
√	DemoBox6	290	240	240	1000	90

3.2 仿真结果

对于选择不同的装箱策略,其空间利用率和耗时间如表 3 所示。

表 3 策略选择

装载情况	策略选择	利用率 (%)	耗时间 (s)
情况 1	宽度优先	95.697	422
情况 2	长度优先	91.6808	365
情况 3	数量优先	88.866	389
情况 4	体积优先	99	413

经过测试,该系统可以成功创建场景、集装箱、缩略图,并显示 X,Y,Z 方向轴以及集装箱入口,如图 5 所示。鼠标右键滑动、方向键均可以实现旋转镜头,鼠标可以拖动货物体,并且可以通过 X,Y,Z 按键使其分别围绕 X,Y,Z 轴旋转。系统通过读取文件中货物的数据,创建货物箱并将其摆放在正确的位置,可以在界面通过控件显示仓库的信息,将鼠标移动到货物体上时可以显示货物的数据,点击缩略图上仓库的 6 个面可以实现主界面上对应平面的显示和隐藏。



图 5 集装箱入口

3.3 导出装箱步骤

测试系统中总共 6 种货物,由里往外,分批次装箱。生成快照的过程中,每将一种货物装箱后间隔大概一秒钟的时间,在时间内截取屏幕,生成一张 PNG 图片,如图 6 所示。将图片保存在 store_Data \Resources 目录下,完成快照功能。装箱步骤如图 7 所示。

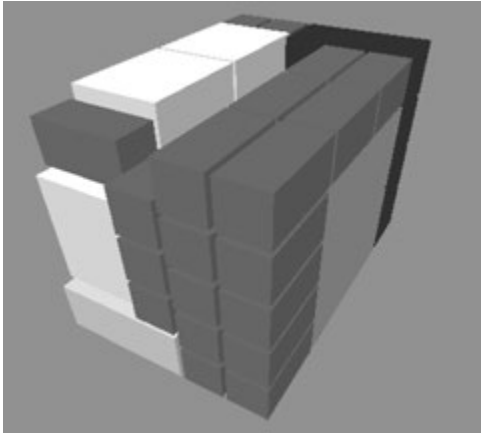


图 6 货物装箱情况

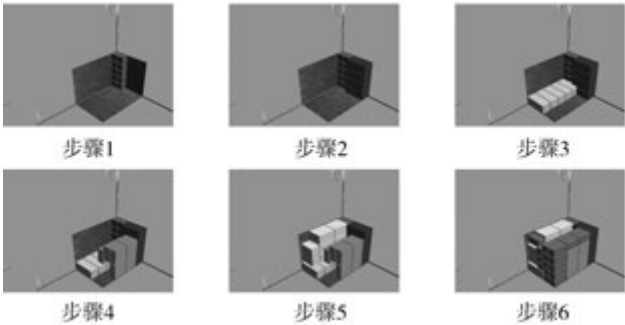


图 7 装箱步骤

4 结束语

本文针对物流配送的装箱问题,设计一款新型的基于遗传算法和 Unity3D 物理引擎装箱系统。该系统用遗传算法算出装箱方案后,添加手动装箱功能和装箱方案导出功能。为了简化手动装箱功能的设计,本文采用 Unity3D 物理引擎作为设计基础,通过 JS 脚本设计物体的运动及基本操作过程,3D 场景中则综合运用 3D 模型、刚体、光源、镜头等重要组成元素,实现手动装箱功能。通过测试表明,该系统既能通过计算机高效计算出一套完整的装箱方案,又能通过人为的经验加以调整,从而实现更好的装箱方案。

参考文献:

[1] Fuellerer G, Doemer K F, Hartl R F, et al. Metaheuristics for vehicle routing problems with three-dimensional loading constraints[J]. European Journal of Operational Research, 2010,201(3): 751-759. (下转第 149 页)

耗效应做优化而得到,优化传输中的尾能耗百分比约为普通传输的一半。由于优化方案加载在云端,节省了由手机端优化所带来的额外电量消耗;设计的尾能耗聚合模型和数据预取决策方案简单可行,可以作为未来云端与智能手机相互配合以节省能耗的模型。

参考文献:

- [1] Pathak A, Hu Y C. Where is the energy spent inside my app?: Fine grained energy accounting on smartphones with eprof [C]// Proceedings of the 7th ACM European Conference on Computer Systems. 2012;29-42.
- [2] 王安,刘丹琦,于炯,等. 移动终端无线数据传输能耗的测量研究[DB/OL]. <http://www.paper.edu.cn/>, 2012-12-27.
- [3] Niranjana B, Aruna B, Arun V. Energy consumption in mobile phones: A measurement study and implications for network applications[C]// Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement Conference. 2009;280-293.
- [4] Falaki H, Lymberopoulos D, Mahajan R, et al. A first look at traffic on smartphones [C]// Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement. 2010;281-287.
- [5] Aggarwal B, Chitnis P, Dey A, et al. Stratus: Energy-efficient mobile communication using cloud support [J]. ACM SIGCOMM Computer Communication Review, 2010, 40(4):477-478.
- [6] Singhal A. Energy Efficient Applications for Low Powered Devices[R]. M Tech Project Stage 1 Report: Indian Institute of Technology Bombay, 2013.
- [7] Balasubramanian N, Balasubramanian A, Venkataramani A. Energy consumption in mobile phones: A measurement study and implications for network applications [C]// Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement Conference. 2009;280-293.
- [8] Liu H, Zhang Y, Zhou Y. TailTheft: Leveraging the wasted time for saving energy in cellular communications [C]// Proceedings of the Sixth International Workshop on MobiArch. 2011;31-36.
- [9] Yu F, Xue G, Zhu H, et al. Cutting without pain: Mitigating 3G radio tail effect on smartphones [C]// INFOCOM, 2013 Proceedings IEEE. 2013;440-444.
- [10] Deng S, Balakrishnan H. Traffic-aware techniques to reduce 3G/LTE wireless energy consumption [C]// Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies. 2012;181-192.
- [11] Zhao B, Zheng Q, Cao G, et al. Energy-aware Web browsing in 3G based smartphones [C]// 2013 IEEE 33rd International Conference on Distributed Computing Systems (ICDCS). 2013;165-175.
- [12] 张磊,康乐,李巍,等. 智能终端的快速休眠特性研究[J]. 广东通信技术, 2012,32(11):69-72.
- [13] Qian F, Wang Z, Gerber A, et al. Characterizing radio resource allocation for 3G networks [C]// Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement. 2010;137-150.
- [14] Huang Y, Song B, Soliman S S. Adaptive fast dormancy for energy efficient wireless packet data communications [C]// 2013 IEEE International Conference on Communications (ICC). 2013;6194-6199.
- [15] Esmailzadeh R, Nakagawa M, Jones A. TDD-CDMA for the 4th generation of wireless communications [J]. IEEE Wireless Communications, 2003,10(4):8-15.
- [16] Shuo Deng. Reducing 3G Energy Consumption on Mobile Devices [DB/OL]. <http://people.csail.mit.edu/shuo-deng/papers/master-thesis.pdf>, 2012-02-03.
- [17] GSM Association. Network Efficiency Task Force Fast Dormancy Best Practices V1.0 [DB/OL]. <http://zh.scribd.com/doc/64209207/Fast-Dormancy-Best-Practices>, 2010-05-26.
- [18] Sarukkai R. Link prediction and path analysis using Markov Chains [J]. Computer Networks: The International Journal of Computer and Telecommunications Networking, 2000,33(1-6):377-386.
- [9] 赵鸿飞,张琦,朱春生,等. 基于改进自适应和声遗传算法的装配序列优化研究[J]. 计算机应用研究, 2013,30(8):2357-2359,2363.
- [10] 陈建新,杨宇航,龚玲,等. 两种准在线装箱算法[J]. 计算机工程, 2006,32(13):4-5,17.
- [11] 刘传领. 改进的蚁群遗传优化算法及其应用[J]. 计算机应用, 2013,33(11):3111-3113.
- [12] 梁剑,王强,安俊英. 铁路集装箱中心站内部集卡调度和箱位配置[J]. 计算机工程, 2012,38(8):213-216.
- [13] 何琨,黄文奇,胡骞. 基于动作空间的求解三维矩形装箱问题的穴度算法[J]. 计算机科学, 2010,37(10):181-183,220.
- [14] 郭晶,陈贤富. 面向装箱问题的量子遗传优化算法[J]. 计算机科学, 2013,40(Z6):67-69,102.
- [15] 郑帅丽,李成严. 基于改进模糊模拟的混合遗传算法[J]. 计算机工程, 2013,39(5):169-173.
- [16] 朱钰,韩昌佩. 一种种群自适应收敛的快速遗传算法[J]. 计算机科学, 2012,39(10):214-217.

(上接第144页)

- [2] 张德富,魏丽军,陈青山,等. 三维装箱问题的组合启发式算法[J]. 软件学报, 2007,18(9):2083-2089.
- [3] 孙棣华,涂平,彭光含,等. 基于遗传算法的单车运输配载研究[J]. 计算机仿真, 2008,25(3):285-288.
- [4] Leung S C H, Zhou X, Zhang D. Extended guided tabu search and a new packing algorithm for the two-dimensional loading vehicle routing problem [J]. Computers and Operations Research, 2011,38(1):205-215.
- [5] 钟石泉,王雪莲. 多箱型三维装箱问题及其优化研究[J]. 计算机工程与应用, 2009,45(22):197-199.
- [6] 贺同智. 自主学习神经网络算法研究[D]. 武汉:华中师范大学, 2011.
- [7] 宁爱兵,熊小华,马良. 城市物流配送中的三维装箱算法[J]. 计算机工程与应用, 2009,45(9):207-209.
- [8] 杨超,陆志强. 基于遗传算法的出口集装箱送箱时间窗优化[J]. 计算机应用研究, 2013,30(6):1643-1646.
- [9] 杜立宁,张德珍,陈世峰. 蚁群算法求解复杂集装箱装载问题[J]. 计算机应用, 2011,31(8):2275-2278.