**第二十三届大学生科技节数学建模大赛**

**承　诺　书**

**我们仔细阅读了第二十三届大学生科技节数学建模大赛的有关注意事项。**

我们完全明白，在竞赛开始后参赛队员不能以任何方式（包括电话、电子邮件、网上咨询等）与队外的任何人（包括教师）研究、讨论与赛题有关的问题。

我们知道，抄袭别人的成果是违反竞赛规则的, 如果引用别人的成果或其他公开的资料（包括网上查到的资料），必须按照规定的参考文献的表述方式在正文引用处和参考文献中明确列出。

我们郑重承诺，严格遵守竞赛规则，以保证竞赛的公正、公平性。如有违反竞赛规则的行为，我们将受到严肃处理。

**参赛队伍信息表（用电子格式填写）**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **姓名** | **学校** | **院系** | **手机** | **电子邮箱** |
| 朱晨宇 | 华中科技大学 | 电子信息与通信学院 | 15262488922 | 2792357190@qq.com |
| 周羽涵 | 华中科技大学 | 电气与电子工程学院 | 18936475506 | 613051212@qq.com |
| 毛晟杰 | 华中科技大学 | 材料科学与工程学院 | 18001559097 | 2661561089@qq.com |
| **队伍名** | **朱晨宇 15262488922** | | **组别** | **兴趣组** |
| **注意事项**   1. **队伍名**为“队长姓名+队长手机号”，如“张三12345678910”**。** 2. **组别**为“专业组/兴趣组”   3、**请将这两页粘贴到论文的最前面，作为论文的封面（非常重要）。** | | | | |

**以下内容请用黑色签字笔填写**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **本队选择的赛题为：□D题□E题□F题（专业组）**  **□A题□B题□C题（兴趣组）** | | | | |
| **队员签字** |  |  |  |  |

**第二十三届大学生科技节数学建模大赛**

**编　号　专　用　页**

**本页用于论文评阅，只需打印出来，不要在本页填写任何内容！**

|  |  |  |  |
| --- | --- | --- | --- |
| **评委** | **评委1** | **评委2** | **评委3** |
| **评分** |  |  |  |
| **备注** |  |  |  |
| **最终得分** | |  | |
| **评价** |  | | |

**基于整数线性规划模型的机舱集装箱运输问题求解**

# 摘要

在实际的航空货运中，如何高效安排货物的装载是航空货运规划的一个重要环节，通过高效而合理的装箱配载可以提高运输量，保证较高效率，在保证安全的同时极大降低运输成本，节约能源保护环境。本文针对机舱集装箱装箱运输问题进行了对载重与体积，飞机飞行的稳定性方面等限制因素的探讨，并建立了相应数学物理模型，并用Lingo和MATLAB软件进行了分析与求解。该问题的研究能更好的指导实际中的装箱与配载的方案，提高运输效率。

**针对问题一，**本文运用了混合型整数线性规划的方法，运用MATLAB进行自定义代码实现，并利用了对货物是否装进集装箱内运用了矩阵与0-1规划的方案进行实现，最后达到在体积与重量均满足题目限制的条件下，实现所装货物的最大值。求解得到一架飞机最多能实现运载货物数为345，具体货物的选取与放置的集装箱编号将在模型求解中给出。

**针对问题二，**本文借用上一问中矩阵与0-1规划的方案进一步解决多架飞机运送所给定的货物问题，首先根据在各架飞机的总重量与各个集装箱的载重和体积均符合要求的前提下，确定所需的飞机架数，然后进一步运用Lingo实现优化的配载分配方案。求解得到所需飞机共3架，所需集装箱总数为80，具体方案将在模型求解中给出。

**针对问题三，**本文根据题目要求对模型一进行了优化处理，依照对飞行稳定性的要求，使用MATLAB进行整数线性规划，并定义了重心的三维空间向量实现装载后与空载时的重心偏移量最小化，从而制定出最优的装载和配载计划。本文再进一步利用遗传算法进行对解法一的优化。求解所得的运载货物数为345，针对飞机的飞行稳定性调整得到的货物分配方案将在附录中给出。

**关键词：多目标优化 0-1规划 整数线性规划 遗传算法**

# 问题重述

## 问题背景

货运的配载计划包括装箱和配载两部分工作，在实际航空货运充满了种种限制，如飞机所能装的集装箱数量，集装箱内部的重量和体积限制，飞行的稳定性与安全性需要得到保证等等。因此，需要利用数学建模给出一种优化的配载方案。

## 问题提出

题目给出了一种飞机，飞机内有36个集装箱分上下左右四块区域，每块区域有9个尺寸为长318cm，宽244cm，平均高163cm（即体积约为12.64m³）的集装箱，且每个集装箱载重不超过6804kg，总载重量为102300kg。现给出特定的货物清单（共700件货物），已知对应的质量与体积，请回答下列问题：

**问题一：**在只有一架飞机的情况下，请给出具体的装箱方式使得飞机所装的货物量达到最大值；

**问题二：**在货物清单中的货物均需运送时，请给出所需飞机的最少架数并给出最优的分配方案；

**问题三：**在问题一的模型基础上，考虑使得运载的货物较多，飞机重心偏移量较低多目标的最优装载与配载的计划。

# 问题分析

**问题一：**对于题目给出的货物清单进行分析后，由于货物量较多，并且存在部分货物因为体积和重量的限制无法由一架飞机完成运送，故考虑0-1整数线性规划。对货物进行编号，其中选入箱中的赋值为1，未选入箱中的赋值为0。其中问题的约束条件为：装箱总数不能超过总个数，总载重不能超过飞机载重，对每个单独的集装箱也不能超过最大载重和体积限制，一件货物只能至多装入一个集装箱内。

**问题二：**问题要求实现用最少的飞机架数运送借用问题一的思路，同样的进行0-1整数线性规划。步骤一：先确立所需的飞机的架数，步骤二：①出于效率考虑，减少装箱数；②对每架飞机，装箱量尽可能多以增大体积利用率。考虑到这是一个多目标问题，要转化为单目标求解，我们的实现方案是使用主目标法，以最小装箱量为主目标，给体积利用率赋予一个下界来限制。

**问题三：**要求最小重心偏移量，需要引入位置参数求最小值。这仍旧是一个多目标问题，要转化为单目标，首先我们要设定一定的重心偏移范围下界保证飞机稳定性，本文使用的是三维空间向量进行实现，并同时希望所装货物数最多，利用整数线性规划进行实现。

# 模型假设与符号说明

## 模型基本假设

1. 由题目表述可知不考虑集装箱中货物的实际长宽高带来的影响，不考虑实际货物摆放顺序的实际需求。
2. 在实际第三问的求解中可知，不考虑飞机飞行过程中耗油所带来的重心偏移，并假设飞机重心即飞机中心。

## 符号说明

符号说明

|  |  |  |
| --- | --- | --- |
| 符号 | 含义 | 单位 |
|  | 第k个集装箱装入第i个货物 |  |
|  | 第i个货物的质量 |  |
|  | 第i个货物的体积 |  |
|  | 集装箱在某飞机上是否空载 |  |
|  | 货物重心相对机舱重心的位置 |  |
|  | 货物重心相对位置的坐标x,y,z |  |

# 模型建立与求解

## 问题一模型建立与求解

考虑如下图给出的装配矩阵y：



其中记行数为,列数为,行数表示集装箱的编号，列数表示所给的货物的编号，若第个集装箱内装有第个货物，则对赋值为1，否则则为0。

1.目标函数：

所需的装载量达到最大值：



2.约束条件：

 (1)

 (2)

 (3)

 (4)

###### 式表示对货物总数的约束，(2)式表示对飞机运载的总质量的约束，(3)式表示对每个集装箱的总质量和总体积的约束，(4)式表示在不同集装箱中每个货物的赋值之和至多为1的约束。所求得的货物总数为345，所得结果如下表所示：



## 问题二模型建立与求解

步骤一：先确立飞机的总架数，分别计算出体积与重量的最值与总和：

|  |  |  |  |
| --- | --- | --- | --- |
| 体积平均值 | 体积总和 | 体积最大值 | 体积最小值 |
| 1.099971904 | 769.980333 | 1.199554937 | 1.00040913 |
| 重量平均值 | 重量总和 | 重量最大值 | 重量最小值 |
| 302.1742857 | 211522 | 314 | 290 |

由问题一的解答可知，两架飞机最多只能装690件货物，显然至少需要3架飞机。按照体积和重量的最大值计算，无论是集装箱还是飞机均符合体积与装载的条件，故确定所需的飞机数量为3架。

步骤二：计算较优的分配方案：

在第一问的基础之上，拓展装配矩阵，加入判定列e表示是否空载，并将行数扩充至108行，其中1~36行表示飞机A，37~72行表示飞机B，73~108行表示飞机C，得到增广矩阵：



1.目标函数：

和

2.约束条件

 (5)

 (6)

 (7)

 (8)

(5)式表示对货物总数的约束，(6)式表示对飞机运载的总质量的约束（式中只列出了对第一架飞机的载重约束，二三同理，(7)式表示对每个集装箱的总质量和总体积的约束，(8)式表示在不同集装箱中每个货物的赋值之和至多为1的约束。

使用MATLAB进行求解，得到结果如下表：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 集装箱编号 | 货物编号 | | | | | | |
| 1 | 3 | 200 | 262 | 355 | 514 | 574 |  |
| 2 | 26 | 57 | 246 | 322 | 436 | 494 | 687 |
| 3 | 72 | 159 | 263 | 379 | 531 | 587 |  |
| 4 | 1 | 120 | 226 | 334 | 463 | 585 | 691 |
| 5 | 45 | 117 | 229 | 424 | 470 | 619 | 665 |
| 6 | 133 | 191 | 387 | 456 | 592 | 673 |  |
| 7 | 27 | 202 | 336 | 417 | 523 | 631 |  |
| 8 | 41 | 74 | 217 | 314 | 457 | 593 | 666 |
| 9 | 93 | 151 | 263 | 429 | 536 | 588 |  |
| 10 | 52 | 154 | 229 | 390 | 498 | 605 | 696 |
| 11 | 118 | 225 | 333 | 431 | 546 | 594 |  |
| 12 | 36 | 107 | 213 | 313 | 465 | 607 |  |
| 13 | 23 | 31 | 138 | 341 | 355 | 502 | 611 |
| 14 | 72 | 136 | 259 | 302 | 438 | 541 | 689 |
| 15 | 99 | 233 | 309 | 459 | 524 | 604 |  |
| 16 | 69 | 140 | 241 | 369 | 497 | 603 |  |
| 17 | 27 | 100 | 219 | 329 | 433 | 538 | 577 |
| 18 | 73 | 118 | 236 | 348 | 448 | 558 |  |
| 19 | 36 | 132 | 238 | 352 | 467 | 556 | 660 |
| 20 | 34 | 135 | 247 | 354 | 467 | 571 | 652 |
| 21 | 100 | 212 | 320 | 381 | 528 | 641 |  |
| 22 | 3 | 154 | 293 | 358 | 492 | 611 |  |
| 23 | 10 | 113 | 217 | 361 | 436 | 545 | 622 |
| 24 | 1 | 184 | 236 | 413 | 506 | 627 | 700 |
| 25 | 76 | 235 | 269 | 423 | 573 | 676 |  |
| 26 | 11 | 196 | 299 | 407 | 517 | 619 |  |
| 27 | 28 | 135 | 264 | 304 | 463 | 584 | 642 |
| 28 | 98 | 212 | 312 | 380 | 534 | 567 | 667 |
| 29 | 73 | 202 | 275 | 401 | 595 | 634 |  |
| 30 | 106 | 206 | 239 | 345 | 452 | 571 | 669 |
| 31 | 167 | 194 | 286 | 400 | 511 | 632 |  |
| 32 | 103 | 145 | 261 | 428 | 473 | 577 | 683 |
| 33 | 93 | 177 | 327 | 437 | 536 | 641 |  |
| 34 | 104 | 161 | 289 | 366 | 477 | 617 |  |
| 35 | 61 | 149 | 178 | 381 | 432 | 546 | 657 |
| 36 | 63 | 125 | 323 | 401 | 463 | 577 |  |
| 37 | 44 | 92 | 275 | 391 | 454 | 612 |  |
| 38 | 19 | 91 | 184 | 305 | 418 | 512 | 569 |
| 39 | 37 | 125 | 273 | 325 | 482 | 589 |  |
| 40 | 18 | 107 | 229 | 319 | 426 | 535 | 664 |
| 41 | 44 | 138 | 280 | 387 | 424 | 601 |  |
| 42 | 10 | 66 | 227 | 324 | 460 | 564 | 670 |
| 43 | 48 | 85 | 291 | 399 | 505 | 613 |  |
| 44 | 20 | 135 | 163 | 280 | 379 | 483 | 595 |
| 45 | 1 | 196 | 303 | 313 | 434 | 567 |  |
| 46 | 18 | 104 | 185 | 348 | 404 | 480 | 608 |
| 47 | 7 | 119 | 271 | 323 | 503 | 588 | 694 |
| 48 | 98 | 209 | 356 | 433 | 523 | 672 | 695 |
| 49 | 180 | 287 | 358 | 499 | 630 |  |  |
| 50 | 10 | 46 | 180 | 306 | 401 | 513 | 624 |
| 51 | 200 | 246 | 414 | 444 | 569 |  |  |
| 52 | 4 | 63 | 205 | 319 | 444 | 483 | 624 |
| 53 | 74 | 148 | 306 | 392 | 458 | 548 | 645 |
| 54 | 108 | 232 | 324 | 468 | 591 | 642 |  |
| 55 | 44 | 113 | 285 | 389 | 497 | 604 |  |
| 56 | 9 | 124 | 224 | 337 | 462 | 520 | 668 |
| 57 | 112 | 213 | 324 | 427 | 507 | 633 | 654 |
| 58 | 67 | 189 | 370 | 472 | 573 |  |  |
| 59 | 8 | 70 | 199 | 306 | 427 | 540 | 647 |
| 60 | 54 | 145 | 216 | 382 | 482 | 589 | 609 |
| 61 | 109 | 136 | 231 | 434 | 551 | 608 |  |
| 62 | 52 | 109 | 210 | 287 | 494 | 521 | 631 |
| 63 | 42 | 125 | 252 | 378 | 468 | 557 | 679 |
| 64 | 91 | 211 | 315 | 424 | 532 | 644 |  |
| 65 | 54 | 156 | 264 | 375 | 479 | 597 |  |
| 66 | 8 | 116 | 228 | 332 | 437 | 548 | 657 |
| 67 | 55 | 170 | 328 | 399 | 531 | 611 |  |
| 68 | 19 | 126 | 211 | 380 | 444 | 553 | 678 |
| 69 | 86 | 197 | 306 | 414 | 515 | 628 |  |
| 70 | 27 | 134 | 266 | 374 | 479 | 523 | 644 |
| 71 | 125 | 228 | 278 | 459 | 528 | 634 |  |
| 72 | 62 | 171 | 276 | 391 | 476 | 599 | 677 |
| 73 | 144 | 218 | 324 | 375 | 558 | 673 |  |
| 74 | 73 | 166 | 299 | 412 | 500 | 608 |  |
| 75 | 13 | 139 | 249 | 356 | 385 | 563 | 657 |
| 76 | 83 | 191 | 297 | 314 | 430 | 586 | 653 |
| 77 | 70 | 154 | 361 | 374 | 478 | 587 |  |
| 78 | 95 | 123 | 285 | 405 | 433 | 625 | 663 |
| 79 | 94 | 161 | 292 | 460 | 571 | 683 |  |
| 80 | 18 | 126 | 240 | 321 | 460 | 626 |  |
| 81 | 35 | 145 | 207 | 343 | 416 | 524 | 623 |
| 82 | 29 | 216 | 248 | 358 | 470 | 542 |  |
| 83 | 44 | 61 | 208 | 315 | 458 | 559 | 692 |
| 84 | 10 | 158 | 314 | 432 | 498 | 605 |  |
| 85 | 23 | 98 | 242 | 283 | 433 | 582 | 672 |
| 86 | 17 | 143 | 302 | 341 | 524 | 606 |  |
| 87 | 18 | 124 | 280 | 285 | 468 | 583 | 631 |
| 88 | 101 | 188 | 284 | 404 | 522 | 572 |  |
| 89 | 51 | 157 | 213 | 388 | 397 | 588 | 700 |
| 90 | 99 | 218 | 315 | 368 | 549 | 565 |  |
| 91 | 63 | 170 | 279 | 289 | 491 | 598 |  |
| 92 | 4 | 30 | 134 | 326 | 378 | 533 | 563 |
| 93 | 47 | 180 | 288 | 374 | 422 | 568 | 638 |
| 94 | 45 | 206 | 256 | 370 | 477 | 574 |  |
| 95 | 44 | 114 | 216 | 334 | 438 | 544 | 621 |
| 96 | 59 | 157 | 270 | 374 | 522 | 651 |  |
| 97 | 25 | 127 | 235 | 330 | 438 | 581 | 660 |
| 98 | 68 | 204 | 299 | 407 | 507 | 618 |  |
| 99 | 31 | 131 | 293 | 405 | 456 | 580 | 690 |
| 100 | 123 | 239 | 312 | 415 | 568 | 675 |  |
| 101 | 33 | 191 | 277 | 310 | 485 | 539 | 696 |
| 102 | 60 | 167 | 317 | 400 | 572 | 664 |  |
| 103 | 68 | 192 | 236 | 383 | 513 | 610 |  |
| 104 | 31 | 64 | 235 | 262 | 459 | 555 | 672 |
| 105 | 79 | 202 | 295 | 393 | 439 | 534 |  |
| 106 | 44 | 149 | 163 | 364 | 443 | 516 | 684 |
| 107 | 92 | 198 | 304 | 323 | 489 | 539 | 645 |
| 108 | 55 | 267 | 330 | 484 | 588 | 686 |  |

## 问题三模型建立与求解

### 问题三求解思路

在问题一的基础上，以货舱中心为原点建立空间直角坐标系，通过给每个货舱赋予一个位置矢量坐标，量化重心偏移位置，以容纳最多数量的货物、沿三个轴方向重心偏移量最小为目标，构建整数0-1线性规划模型求解。

### 问题三模型建立

对于问题一中的装配矩阵，我们加入新的条件：

对每个集装箱，定义一个位置向量，即在要求的同时，使得。

这样问题就变成了一个多目标优化问题。解决问题有两种方法，第一种是对于设定上界，以为主目标，将多目标转化为单目标优化问题。

为了线性化和方便求解，我们将向量按三个方向分解，根据舱体位置，以货舱总中心为原点，建立空间直角坐标系，每个坐标的取值范围如下：



为了得到更加精确的结果，我们设置上界为一个极小值，可以将重心偏移关系表示为：

；；

我们取极小值为0.1，考虑到每个货物的重量和定义的坐标乘积0.1，基本上可以求出偏移最优解。

同时，还要满足问题一中提出的有关条件：

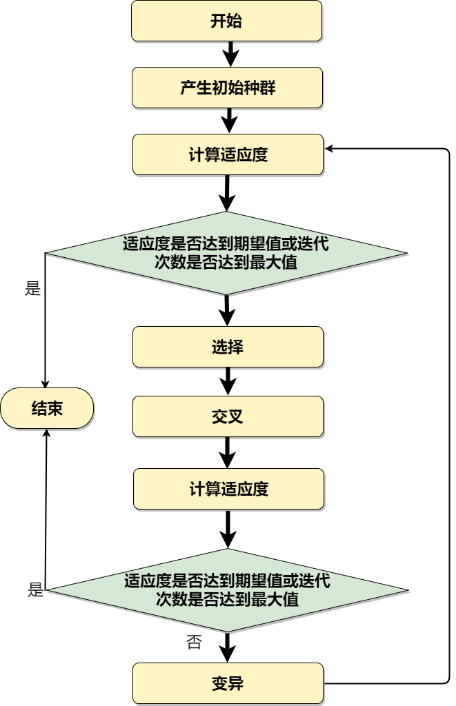
；

；

。

第二种方法就是运用遗传算法直接进行多目标优化。我们使用的是NSGA-II算法。

这种算法利用进化算法的思想，比较拥挤度，拥挤度大的点（实际为和周围点相隔远的点）更能保持物种多样性，更容易发展出新种群。具体算法表现为首先随机产生规模为N的初始种群，非支配排序后通过遗传算法的选择、交叉、变异三个基本操作得到第一代子代种群；其次从第二代开始将父代种群与子代种群合并进行快速非支配排序，同时对每个非支配层中的个体进行拥挤度计算，根据非支配关系以及个体的拥挤度选取合适的个体组成新的附带种群；最后，通过遗传算法的基本操作产生新的自带种群，以此类推直到满足程序结束条件[1]。



遗传算法流程图

### 问题三模型求解与分析

对于问题三，我们采用MATLAB解决整数线性规划问题。将由y组成的二维变量矩阵转化为一维向量进行规划求解，采用intlinprog函数得到最终结果。

结果输出的初始0-1矩阵详见附件T\_03\_origin\_result.csv。

对输出矩阵进行进一步处理可以得到最终结果如下表：



可见该结果在重心偏移尽可能小的情况下实现了装箱货物的最大化。

第二种方法我们采用了NSGA-Ⅱ遗传算法，同样在MATLAB中使用gamultiobj函数求解，代码详见附录。在求解过程中，由于需要的种群数量过大（至少为36\*700 = 25200个目标），以及在针对一维向量编写约束条件时没有向量化，使用了较大的循环，导致出现了内存不足的问题，代码无法跑通。通过其他较小的模型进行验证，证明不是代码书写错误的问题。遗传算法未能跑通是比赛期间的一个遗憾，故将代码附在附录中供参考。

# 模型评价与推广

## 模型的优点

1. 模型充分结合实际，简化了货物装箱的条件和飞机重心的条件，考虑了绝大部分题目所给出的条件得到合理的模型，最优化最高效地将货物装载到集装箱中并分配在飞机货舱中，尽可能保证飞机飞行的稳定性安全性，同时在保证运力的前提下动用最少的运输工具数。
2. 模型通过数据分析合理设置参数，又创新性地使用了引入空载矩阵的办法，模型的输出结果符合题目要求，能解决实际问题；
3. 算法采用0-1整数规划，将复杂的问题线性化便于求解，结果也更为直观；
4. 本文得到的装配方案策略具有输出稳定的特点，在现有条件下能有效提高装配方案解决效率。

## 模型的不足

1. 实际应用中，货物的长宽高、实际摆放顺序（如易碎物品不能置于底层）也是重要的因素，但本文未能考虑到这些因素的影响，一定程度上影响了模型的准确性；
2. 在实际运用中，pcg，即重心处弦长百分比（计算公式为 ，其中表示以机翼弦长的参考点为基准飞机重心的纵向位置，是飞机的空气动力学中心的纵向位置，是机翼的弦长，即飞机重心相对空气动力学中心的偏移并化为百分比）反应了飞机重心相对于机翼弦长的位置，会对飞机飞行产生稳定性方面的影响，但是本模型并没有考虑；飞机的实际重心也不一定在中心点，模型也没有考虑飞行过程中耗油会带来的重心偏移的问题等等。需要优化这些模型需要更多相关数据，如机翼长度，货物的长宽高数值等等，同时也会带来更多空间方面的建模挑战。

## 模型的推广

更进一步讲，基于本文给出的模型，可以通过添加更多约束条件进一步讨论装箱过程中的优化问题，譬如针对不同批次的货物，对装箱次序提出约束；同时，还可以引入非线性不等式和实数变量更加灵活地解决问题。若进一步考虑装箱物品体积和尺寸，本文所涉及的遗传算法很适合求出最优解。

# 参考文献

1. “科学猛兽”, 遗传算法之NSGA-Ⅱ原理分析和代码解读[P/OL]，https://zhuanlan.zhihu.com/p/144807879

# 附录

|  |
| --- |
| 附录1 |
| 基于MATLAB intlinprog混合整数线性规划的第一问求解代码 |
| clc  clear  %导入数据  data = readmatrix("\*\*\*\B.csv");  count = data(: , 1)';  weight = data(: , 2)';  volume = data(: , 3)';  %设置规划  %约束条件lb <= x <= ub  %约束为0-1变量  lb = zeros(1, 36\*700);  ub = ones(1, 36\*700);  %约束条件A \* x <= b  %对于总体的约束矩阵  weight\_y = repmat(weight, 1, 36);  volume\_y = repmat(volume, 1, 36);  ones\_vector = ones(1, 36\*700);  A = [ones\_vector; weight\_y];  b = [700; 102300];  %对于每行的约束矩阵  for i = 1:36  container\_row = zeros(1, 36\*700);  container\_row((i-1)\*700 + 1 : i\*700) = 1;  container\_weight = container\_row .\* weight\_y;  container\_volume = container\_row .\* volume\_y;  A = [A; container\_weight; container\_volume];  b = [b; 6804; 12.647496];  end  %对于每列的约束矩阵  for j = 1:700  container\_col = zeros(1, 36\*700);  container\_col(j:700:end) = 1;  A = [A; container\_col; - container\_col];  b = [b; 1; 0];  end  %Aeq\*x = beq  Aeq = [];  beq = [];  x0 = [];  % 目标函数和整数变量  f = -ones(1 , 36\*700);  intcon = 1 : 36\*700;  % 整数线性规划  [x,fval,exitflag,output] = intlinprog(f,intcon,A,b,Aeq,beq,lb,ub,x0);  % 存储和输出结果矩阵  x = res(x,36,700);  resultMatrix = zeros(size(x));  for i = 1:size(x, 1)  indices = find(x(i, :));  resultMatrix(i, indices) = indices;  end  csvFileName\_01 = 'T\_01\_result.csv';  for i = 1:size(resultMatrix, 1)  currentRow = resultMatrix(i, :);  currentRow(currentRow == 0) = [];  if i == 1  writematrix(currentRow, csvFileName\_01);  else  writematrix(currentRow, csvFileName\_01, 'WriteMode', 'append');  end  end  csvFileName\_02 = 'T\_01\_origin\_result.csv';  writematrix(x, csvFileName\_02);  % 显示结果矩阵  disp(resultMatrix);  disp(-fval);  disp(x);  % 转置一维行向量为二维矩阵  function a = res(A,a,b)  a = reshape(A.',b,a).';  end |
| 基于Lingo多维变量整数0-1非线性规划的第二问求解代码 |
| MODEL:  SETS:  CONTAINER\_ROWS /1..108/;  CONTAINER\_COLS /1..700/;  PART\_ROWS/1..36/;  ONE\_ROWS/1..72/;  TWO\_ROWS/1..108/;  link(CONTAINER\_COLS, CONTAINER\_ROWS): y;  CARGO\_ROWS/1..700/: COUNT, WEIGHT, VOLUME;  NUMBER\_ROWS/1..108/:m;  ANOTHER\_ROWS/1..108/:k;  ENDSETS  DATA:  COUNT, WEIGHT, VOLUME = @OLE( '\*\*\*\B.xls', 'count', 'weight', 'volume');  @OLE('\*\*\*\T\_02\_origin\_result.xlsx','container')=y;  ENDDATA  min = @sum(NUMBER\_ROWS(i):k(i));  @for(link: @bin(y));  @for(CONTAINER\_ROWS(i):@sum(CONTAINER\_COLS(j):y(j,i))=m(i));  @for(ANOTHER\_ROWS(i):k(i)=@if(m(i)#ne#0,1,0));  @for(PART\_ROWS(i):@sum(CARGO\_ROWS(j):WEIGHT(j) \*y(j,i))<= 102300);  @for(ONE\_ROWS(i):@sum(CARGO\_ROWS(j):WEIGHT(j) \*y(j,i))<=204600);  @for(TWO\_ROWS(i):@sum(CARGO\_ROWS(j):WEIGHT(j) \*y(j,i))<= 306900);  @for(CONTAINER\_ROWS(i): @sum(CONTAINER\_COLS(j): WEIGHT(j) \* y(j,i)) <= 6804);  @for(CONTAINER\_ROWS(i): @sum(CONTAINER\_COLS(j): VOLUME(j) \* y(j,i)) <= 12.647496);  @for(CONTAINER\_COLS(j): @sum(CONTAINER\_ROWS(i): y(j,i)) =1);  END |
| 基于MATLAB intlinprog整数线性规划的第三问求解代码 |
| clc  clear  %导入数据  data = readmatrix("\*\*\*\B.csv");  count = data(: , 1)';  weight = data(: , 2)';  volume = data(: , 3)';  %初始化箱体位置矢量坐标  x\_axis = ones(1, 36);  x\_axis([1:9, 19:27]) = -1;  x\_axis = kron(x\_axis, ones(1, 700));  y\_axis = (1:36) - 5 - 9 \* floor((0:35) / 9);  y\_axis = kron(y\_axis, ones(1, 700));  z\_axis = ones(1,36);  z\_axis(19:36) = -1;  z\_axis = kron(z\_axis, ones(1, 700));  %设置规划  %约束条件lb <= x <= ub  %约束为0-1变量  lb = zeros(1, 36\*700);  ub = ones(1, 36\*700);  %约束条件A \* x <= b  %对于总体的约束矩阵  weight\_y = repmat(weight, 1, 36);  volume\_y = repmat(volume, 1, 36);  ones\_vector = ones(1, 36\*700);  A = [ones\_vector; weight\_y; weight\_y .\* x\_axis; weight\_y .\* y\_axis; weight\_y .\* z\_axis];  b = [700; 102300; 0.1; 0.1; 0.1];  %对于每行的约束矩阵  for i = 1:36  container\_row = zeros(1, 36\*700);  container\_row((i-1)\*700 + 1 : i\*700) = 1;  container\_weight = container\_row .\* weight\_y;  container\_volume = container\_row .\* volume\_y;  A = [A; container\_weight; container\_volume];  b = [b; 6804; 12.647496];  end    %对于每列的约束矩阵  for j = 1:700  container\_col = zeros(1, 36\*700);  container\_col(j:700:end) = 1;  A = [A; container\_col; - container\_col];  b = [b; 1; 0];  end  %对于  %Aeq\*x = beq x  Aeq = [];  beq = [];  x0 = [];  % 目标函数和整数变量  f = -ones(1 , 36\*700);  intcon = 1 : 36\*700;  % 整数线性规划  [x,fval,exitflag,output] = intlinprog(f,intcon,A,b,Aeq,beq,lb,ub,x0);  % 存储和输出结果矩阵  x = res(x,36,700);  resultMatrix = zeros(size(x));  for i = 1:size(x, 1)  indices = find(x(i, :));  resultMatrix(i, indices) = indices;  end  csvFileName\_01 = 'T\_01\_result.csv';  for i = 1:size(resultMatrix, 1)  currentRow = resultMatrix(i, :);  currentRow(currentRow == 0) = [];  if i == 1  writematrix(currentRow, csvFileName\_01);  else  writematrix(currentRow, csvFileName\_01, 'WriteMode', 'append');  end  end  csvFileName\_02 = 'T\_01\_origin\_result.csv';  writematrix(x, csvFileName\_02);  % 显示结果矩阵  disp(resultMatrix);  disp(-fval);  disp(x);  % 转置一维行向量为二维矩阵  function a = res(A,a,b)  a = reshape(A.',b,a).';  end |
| 基于MATLAB gamultiobj的NSGA-Ⅱ遗传算法的第三问求解代码 |
| clc  clear  %导入数据  data = readmatrix("\*\*\*\B.csv");  count = data(: , 1)';  weight = data(: , 2)';  volume = data(: , 3)';  %初始化箱体位置矢量坐标  x\_axis = ones(1, 36);  x\_axis([1:9, 19:27]) = -1;  x\_axis = kron(x\_axis, ones(1, 700));  y\_axis = (1:36) - 5 - 9 \* floor((0:35) / 9);  y\_axis = kron(y\_axis, ones(1, 700));  z\_axis = ones(1,36);  z\_axis(19:36) = -1;  z\_axis = kron(z\_axis, ones(1, 700));  %设定NSGA参数  fitnessfcn = @getFit; %定义适应函数  nvars = 36\*700; %定义变量个数  %设置规划  %约束条件lb <= x <= ub  %约束为0-1变量  lb = zeros(1, 36\*700);  ub = ones(1, 36\*700);  %约束条件A \* x <= b  %对于总体的约束矩阵  weight\_y = repmat(weight, 1, 36);  volume\_y = repmat(volume, 1, 36);  ones\_vector = ones(1, 36\*700);  A = [ones\_vector; weight\_y];  b = [700; 102300];  %对于每行的约束矩阵  for i = 1:36  container\_row = zeros(1, 36\*700);  container\_row((i-1)\*700 + 1 : i\*700) = 1;  container\_weight = container\_row .\* weight\_y;  container\_volume = container\_row .\* volume\_y;  A = [A; container\_weight; container\_volume];  b = [b; 6804; 12.647496];  end  %对于每列的约束矩阵  for j = 1:700  container\_col = zeros(1, 36\*700);  container\_col(j:700:end) = 1;  A = [A; container\_col; - container\_col];  b = [b; 1; 0];  end  %Aeq\*x = beq x  Aeq = [];  beq = [];  nonlcon = [];  intcon = 1 : 36\*700;  %设置求解器  % 创建遗传算法的选项  options = optimoptions('gamultiobj', ...  'ParetoFraction', 0.3, ...  'PopulationSize', 36\*700\*5, ...  'Generations', 200, ...  'StallGenLimit', 200, ...  'TolFun', 1e-5, ...  'PlotFcn', @gaplotpareto);  % 遗传算法求解  [x,fval,exitflag,output,population,scores] = gamultiobj(fitnessfcn,nvars,A,b,Aeq,beq,lb,ub,nonlcon,intcon,options);  % 存储和输出结果矩阵  x = res(x,36,700);  resultMatrix = zeros(size(x));  for i = 1:size(x, 1)  indices = find(x(i, :));  resultMatrix(i, indices) = indices;  end  csvFileName\_01 = 'T\_01\_result.csv';  for i = 1:size(resultMatrix, 1)  currentRow = resultMatrix(i, :);  currentRow(currentRow == 0) = [];  if i == 1  writematrix(currentRow, csvFileName\_01);  else  writematrix(currentRow, csvFileName\_01, 'WriteMode', 'append');  end  end  csvFileName\_02 = 'T\_01\_origin\_result.csv';  writematrix(x, csvFileName\_02);  % 显示结果矩阵  disp(resultMatrix);  disp(-fval);  disp(x);  % 转置一维行向量为二维矩阵  function a = res(A,a,b)  a = reshape(A.',b,a).';  end  % 适应函数  function objectives = getObj(x, weight, x\_axis, y\_axis, z\_axis)  obj1 = sum(x .\* weight .\* x\_axis)^2 + sum(x .\* weight .\* y\_axis)^2 + sum(x .\* weight .\* z\_axis)^2;  obj2 = -sum(x);  objectives = [obj1, obj2];  end  function fitness = getFit(x)  fitness = getObj(x, weight, x\_axis, y\_axis, z\_axis);  end |