حل مسئله کوله پشتی با الگوریتم مورچه ACO در متلب

فرض کنید یک کوله‌پشتی با حجمی ثابت و مجموعه‌ای از اشیاء دارید که هر کدام از آن ها حجمی و ارزشی دارند. می‌خواهید کوله‌پشتی خود را به نحوی پرکنید که حجم اشیا برداشته شده از حجم کوله‌پشتی بیشتر نباشد و مجموع ارزش اشیا بیشینه باشد.

صورت مسئله:

یک کوله پشتی به حجم ۵۰۰ داریم و ۲۰ تا شی داریم که ارزش اشیابه صورت زیر است:

```
v = [391 444 250 330 246 400 150 266 268 293 471 388 364
493 202 161 410 270 384 486];
```

و وزن اشیا به صورت زیر است:

```
w = [55 52 59 24 52 46 45 34 34 59 59 28 57 21 47 66 64 42
22 23];
```

می خواهیم این اشیا را به نحوی در کوله پشتی قرار دهیم که ارزش اشیا در کوله پشتی ماکزیمم شود و حجم اشیا درون کوله پشتی از حجم کل کوله پشتی بیشتر نشود.

شرح کد:

این سورس کد شامل ۵ فایل می باشد.

ابتدا تابع CreateModel() را پیاده سازی می کنیم که اطلاعات مسیله داخل یک مدل پیاده سازی می شود برای اینکه به پارامترهای مسیله به صورت یکجا دسترسی داشته باشیم.

```
function model=CreateModel()

ارزش هر کدام از اشیا

    v = [391 444 250 330 246 400 150 266 268 293 471 388
364 493 202 161 410 270 384 486];

وزن هر کدام از اشیا
```

```
    w = [55 52 59 24 52 46 45 34 34 59 59 28 57 21 47 66 64
42 22 23];

                                                تعداد کل اشیا
     n = numel(v);

                        حداکثر وزنی که کوله پشتی می تواند تحمل کند
    W = 500;

                                 اینجا هم اطلاعات مسیله را ذخیره کردیم.

    model.n = n;
    model.v = v;
    model.w = w;
    model.W = W;

end
```

تابع MyFit(x, model) برای محاسبه فیتنس بکار می رود

```
function [z, sol] = MyFit(x, model)


      ارزش و حجم اشیایی که برداشتیم و ظرفیت حجم کوله پشتی را نیاز
داریم

    v=model.v;
    w=model.w;
    W=model.W;

                مجموع اشیایی که انتخاب شده اند ضرب در ارزش شان

    V1 = sum(v.*x);
    W1 = sum(w.*x);
                مجموع اشیایی که انتخاب نشده اند ضرب در ارزش شان

    V0 = sum(v.*(1-x));
    W0 = sum(w.*(1-x));

          میزان تخلف(باید حجم کتر از ظرفیت حجم کوله پشتی باشه)
Violation = max(W1/W-1, 0);
```

```
هزینه تخلف
    z = V0*(1+100*Violation);

    sol.V1 = V1;
    sol.W1 = W1;
    sol.V0 = V0;
    sol.W0 = W0;
    sol.Violation = Violation;
    sol.z = 1/(1+z);
    sol.IsFeasible = (Violation == 0); اگر تخلف نداشته باشیم
جواب قابل قبول است.

end
```

RouletteWheelSelection(P): چرخ رولت

```
function j = RouletteWheelSelection(P)

    r = rand;

    C = cumsum(P);

    j = find(r <= C, 1, 'first');

end
```

aco.m: کد الگوریتم مورچه به همراه تعریف مسیله و تابع بالا

```
clc;
clear;
close all;

تعریف مسیله

model = CreateModel();      ایجاد مدل

FitFunction = @(x) MyFit(x, model);    تابع برازندگی

nVar = model.n; تعداد متغیرها
```

```
پارامترهای الگوریتم مورچه

MaxIt = 300;       تعداد ماکزیمم تکرار

nAnt = 40;         تعداد مورچه هااندازه جمعیت
Q = 1;

tau0 = 0.1;         فورون اولیه

alpha = 1;         وزن نمایی فورون
beta = 0.02;       وزن نمایی هیوریستسک

rho = 0.1;         میزان تبخیر


مقداردهی اولیه

N = [0 1];

eta = [model.w./model.v
     model.v./model.w];            اطلاعات هیوریستسک

tau = tau0*ones(2, nVar);       ماتریس فورون

BestFit = zeros(MaxIt, 1);      آرایه نگه داری ارزش بهترین
 برازندگی

مورچه خالی
empty_ant.Tour = [];
empty_ant.x = [];
empty_ant.Fit = [];
empty_ant.Sol = [];

ماتریس کلونی مورچه

ant = repmat(empty_ant, nAnt, 1);

بهترین مورچه
BestSol.Fit = 0;

حلقه اصلی مورچه

for it = 1:MaxIt
```

```
حرکت مورچه
for k = 1:nAnt

    ant(k).Tour = [];

    for l = 1:nVar

        P = tau(:, l).^alpha.*eta(:, l).^beta;

        P = P/sum(P);

        j = RouletteWheelSelection(P);

        ant(k).Tour = [ant(k).Tour j];

    end

    ant(k).x = N(ant(k).Tour);

    [ant(k).Fit, ant(k).Sol] = FitFunction(ant(k).x);

    if ant(k).Fit>BestSol.Fit
        BestSol = ant(k);
    end

end

بروزرسانی فورون ها
for k = 1:nAnt

    tour = ant(k).Tour;

    for l = 1:nVar

        tau(tour(l), l) = tau(tour(l), l)+Q/ant(k).Fit;

    end

end

     تبخیر
tau = (1-rho)*tau;

ذخیره بهترین برازندگی
BestFit(it) = BestSol.Fit;
```

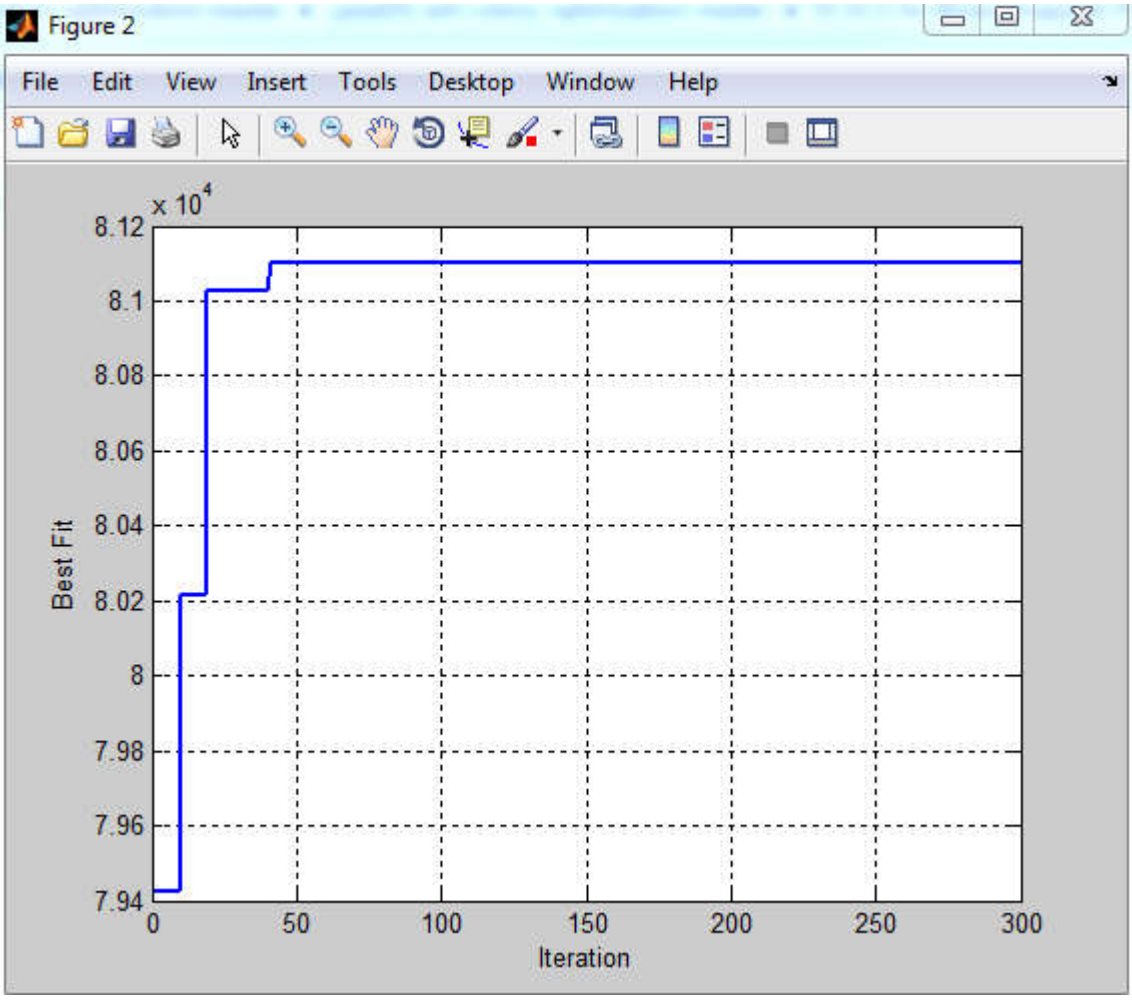
```
    اطلاعات تکرار را نمایش بده
    if BestSol.Sol.IsFeasible
        FeasiblityFlag = '*';
    else
        FeasiblityFlag = '';
    end
    disp(['Iteration ' num2str(it) ': Best Fit = '
num2str(BestFit(it)) ' ' FeasiblityFlag]);

end

نتایج

figure;
plot(BestFit, 'LineWidth', 2);
xlabel('Iteration');
ylabel('Best Fit');
grid on;
```

نتایج:




```
Iteration 1: Best Fit = 79424
Iteration 2: Best Fit = 79424
Iteration 3: Best Fit = 79424
Iteration 4: Best Fit = 79424
Iteration 5: Best Fit = 79424
Iteration 6: Best Fit = 79424
Iteration 7: Best Fit = 79424
Iteration 8: Best Fit = 79424
Iteration 9: Best Fit = 79424
Iteration 10: Best Fit = 80215.8
Iteration 11: Best Fit = 80215.8
Iteration 12: Best Fit = 80215.8
Iteration 13: Best Fit = 80215.8
Iteration 14: Best Fit = 80215.8
Iteration 15: Best Fit = 80215.8
Iteration 16: Best Fit = 80215.8
Iteration 17: Best Fit = 80215.8
```

```
Iteration 18: Best Fit = 80215.8
Iteration 19: Best Fit = 81028.8
Iteration 20: Best Fit = 81028.8
Iteration 21: Best Fit = 81028.8
Iteration 22: Best Fit = 81028.8
Iteration 23: Best Fit = 81028.8
Iteration 24: Best Fit = 81028.8
Iteration 25: Best Fit = 81028.8
Iteration 26: Best Fit = 81028.8
Iteration 27: Best Fit = 81028.8
Iteration 28: Best Fit = 81028.8
Iteration 29: Best Fit = 81028.8
Iteration 30: Best Fit = 81028.8
Iteration 31: Best Fit = 81028.8
Iteration 32: Best Fit = 81028.8
Iteration 33: Best Fit = 81028.8
Iteration 34: Best Fit = 81028.8
Iteration 35: Best Fit = 81028.8
Iteration 36: Best Fit = 81028.8
Iteration 37: Best Fit = 81028.8
Iteration 38: Best Fit = 81028.8
Iteration 39: Best Fit = 81028.8
Iteration 40: Best Fit = 81028.8
Iteration 41: Best Fit = 81100.8
Iteration 42: Best Fit = 81100.8
Iteration 43: Best Fit = 81100.8
Iteration 44: Best Fit = 81100.8
Iteration 45: Best Fit = 81100.8
Iteration 46: Best Fit = 81100.8
Iteration 47: Best Fit = 81100.8
Iteration 48: Best Fit = 81100.8
Iteration 49: Best Fit = 81100.8
Iteration 50: Best Fit = 81100.8
Iteration 51: Best Fit = 81100.8
Iteration 52: Best Fit = 81100.8
Iteration 53: Best Fit = 81100.8
Iteration 54: Best Fit = 81100.8
Iteration 55: Best Fit = 81100.8
Iteration 56: Best Fit = 81100.8
Iteration 57: Best Fit = 81100.8
Iteration 58: Best Fit = 81100.8
Iteration 59: Best Fit = 81100.8
Iteration 60: Best Fit = 81100.8
Iteration 61: Best Fit = 81100.8
```

```
Iteration 62: Best Fit = 81100.8
Iteration 63: Best Fit = 81100.8
Iteration 64: Best Fit = 81100.8
Iteration 65: Best Fit = 81100.8
Iteration 66: Best Fit = 81100.8
Iteration 67: Best Fit = 81100.8
Iteration 68: Best Fit = 81100.8
Iteration 69: Best Fit = 81100.8
Iteration 70: Best Fit = 81100.8
Iteration 71: Best Fit = 81100.8
Iteration 72: Best Fit = 81100.8
Iteration 73: Best Fit = 81100.8
Iteration 74: Best Fit = 81100.8
Iteration 75: Best Fit = 81100.8
Iteration 76: Best Fit = 81100.8
Iteration 77: Best Fit = 81100.8
Iteration 78: Best Fit = 81100.8
Iteration 79: Best Fit = 81100.8
Iteration 80: Best Fit = 81100.8
Iteration 81: Best Fit = 81100.8
Iteration 82: Best Fit = 81100.8
Iteration 83: Best Fit = 81100.8
Iteration 84: Best Fit = 81100.8
Iteration 85: Best Fit = 81100.8
Iteration 86: Best Fit = 81100.8
Iteration 87: Best Fit = 81100.8
Iteration 88: Best Fit = 81100.8
Iteration 89: Best Fit = 81100.8
Iteration 90: Best Fit = 81100.8
Iteration 91: Best Fit = 81100.8
Iteration 92: Best Fit = 81100.8
Iteration 93: Best Fit = 81100.8
Iteration 94: Best Fit = 81100.8
Iteration 95: Best Fit = 81100.8
Iteration 96: Best Fit = 81100.8
Iteration 97: Best Fit = 81100.8
Iteration 98: Best Fit = 81100.8
Iteration 99: Best Fit = 81100.8
Iteration 100: Best Fit = 81100.8
Iteration 101: Best Fit = 81100.8
Iteration 102: Best Fit = 81100.8
Iteration 103: Best Fit = 81100.8
Iteration 104: Best Fit = 81100.8
Iteration 105: Best Fit = 81100.8
```

```
Iteration 106: Best Fit = 81100.8
Iteration 107: Best Fit = 81100.8
Iteration 108: Best Fit = 81100.8
Iteration 109: Best Fit = 81100.8
Iteration 110: Best Fit = 81100.8
Iteration 111: Best Fit = 81100.8
Iteration 112: Best Fit = 81100.8
Iteration 113: Best Fit = 81100.8
Iteration 114: Best Fit = 81100.8
Iteration 115: Best Fit = 81100.8
Iteration 116: Best Fit = 81100.8
Iteration 117: Best Fit = 81100.8
Iteration 118: Best Fit = 81100.8
Iteration 119: Best Fit = 81100.8
Iteration 120: Best Fit = 81100.8
Iteration 121: Best Fit = 81100.8
Iteration 122: Best Fit = 81100.8
Iteration 123: Best Fit = 81100.8
Iteration 124: Best Fit = 81100.8
Iteration 125: Best Fit = 81100.8
Iteration 126: Best Fit = 81100.8
Iteration 127: Best Fit = 81100.8
Iteration 128: Best Fit = 81100.8
Iteration 129: Best Fit = 81100.8
Iteration 130: Best Fit = 81100.8
Iteration 131: Best Fit = 81100.8
Iteration 132: Best Fit = 81100.8
Iteration 133: Best Fit = 81100.8
Iteration 134: Best Fit = 81100.8
Iteration 135: Best Fit = 81100.8
Iteration 136: Best Fit = 81100.8
Iteration 137: Best Fit = 81100.8
Iteration 138: Best Fit = 81100.8
Iteration 139: Best Fit = 81100.8
Iteration 140: Best Fit = 81100.8
Iteration 141: Best Fit = 81100.8
Iteration 142: Best Fit = 81100.8
Iteration 143: Best Fit = 81100.8
Iteration 144: Best Fit = 81100.8
Iteration 145: Best Fit = 81100.8
Iteration 146: Best Fit = 81100.8
Iteration 147: Best Fit = 81100.8
Iteration 148: Best Fit = 81100.8
Iteration 149: Best Fit = 81100.8
```

```
Iteration 150: Best Fit = 81100.8
Iteration 151: Best Fit = 81100.8
Iteration 152: Best Fit = 81100.8
Iteration 153: Best Fit = 81100.8
Iteration 154: Best Fit = 81100.8
Iteration 155: Best Fit = 81100.8
Iteration 156: Best Fit = 81100.8
Iteration 157: Best Fit = 81100.8
Iteration 158: Best Fit = 81100.8
Iteration 159: Best Fit = 81100.8
Iteration 160: Best Fit = 81100.8
Iteration 161: Best Fit = 81100.8
Iteration 162: Best Fit = 81100.8
Iteration 163: Best Fit = 81100.8
Iteration 164: Best Fit = 81100.8
Iteration 165: Best Fit = 81100.8
Iteration 166: Best Fit = 81100.8
Iteration 167: Best Fit = 81100.8
Iteration 168: Best Fit = 81100.8
Iteration 169: Best Fit = 81100.8
Iteration 170: Best Fit = 81100.8
Iteration 171: Best Fit = 81100.8
Iteration 172: Best Fit = 81100.8
Iteration 173: Best Fit = 81100.8
Iteration 174: Best Fit = 81100.8
Iteration 175: Best Fit = 81100.8
Iteration 176: Best Fit = 81100.8
Iteration 177: Best Fit = 81100.8
Iteration 178: Best Fit = 81100.8
Iteration 179: Best Fit = 81100.8
Iteration 180: Best Fit = 81100.8
Iteration 181: Best Fit = 81100.8
Iteration 182: Best Fit = 81100.8
Iteration 183: Best Fit = 81100.8
Iteration 184: Best Fit = 81100.8
Iteration 185: Best Fit = 81100.8
Iteration 186: Best Fit = 81100.8
Iteration 187: Best Fit = 81100.8
Iteration 188: Best Fit = 81100.8
Iteration 189: Best Fit = 81100.8
Iteration 190: Best Fit = 81100.8
Iteration 191: Best Fit = 81100.8
Iteration 192: Best Fit = 81100.8
Iteration 193: Best Fit = 81100.8
```

```
Iteration 194: Best Fit = 81100.8
Iteration 195: Best Fit = 81100.8
Iteration 196: Best Fit = 81100.8
Iteration 197: Best Fit = 81100.8
Iteration 198: Best Fit = 81100.8
Iteration 199: Best Fit = 81100.8
Iteration 200: Best Fit = 81100.8
Iteration 201: Best Fit = 81100.8
Iteration 202: Best Fit = 81100.8
Iteration 203: Best Fit = 81100.8
Iteration 204: Best Fit = 81100.8
Iteration 205: Best Fit = 81100.8
Iteration 206: Best Fit = 81100.8
Iteration 207: Best Fit = 81100.8
Iteration 208: Best Fit = 81100.8
Iteration 209: Best Fit = 81100.8
Iteration 210: Best Fit = 81100.8
Iteration 211: Best Fit = 81100.8
Iteration 212: Best Fit = 81100.8
Iteration 213: Best Fit = 81100.8
Iteration 214: Best Fit = 81100.8
Iteration 215: Best Fit = 81100.8
Iteration 216: Best Fit = 81100.8
Iteration 217: Best Fit = 81100.8
Iteration 218: Best Fit = 81100.8
Iteration 219: Best Fit = 81100.8
Iteration 220: Best Fit = 81100.8
Iteration 221: Best Fit = 81100.8
Iteration 222: Best Fit = 81100.8
Iteration 223: Best Fit = 81100.8
Iteration 224: Best Fit = 81100.8
Iteration 225: Best Fit = 81100.8
Iteration 226: Best Fit = 81100.8
Iteration 227: Best Fit = 81100.8
Iteration 228: Best Fit = 81100.8
Iteration 229: Best Fit = 81100.8
Iteration 230: Best Fit = 81100.8
Iteration 231: Best Fit = 81100.8
Iteration 232: Best Fit = 81100.8
Iteration 233: Best Fit = 81100.8
Iteration 234: Best Fit = 81100.8
Iteration 235: Best Fit = 81100.8
Iteration 236: Best Fit = 81100.8
Iteration 237: Best Fit = 81100.8
```

```
Iteration 238: Best Fit = 81100.8
Iteration 239: Best Fit = 81100.8
Iteration 240: Best Fit = 81100.8
Iteration 241: Best Fit = 81100.8
Iteration 242: Best Fit = 81100.8
Iteration 243: Best Fit = 81100.8
Iteration 244: Best Fit = 81100.8
Iteration 245: Best Fit = 81100.8
Iteration 246: Best Fit = 81100.8
Iteration 247: Best Fit = 81100.8
Iteration 248: Best Fit = 81100.8
Iteration 249: Best Fit = 81100.8
Iteration 250: Best Fit = 81100.8
Iteration 251: Best Fit = 81100.8
Iteration 252: Best Fit = 81100.8
Iteration 253: Best Fit = 81100.8
Iteration 254: Best Fit = 81100.8
Iteration 255: Best Fit = 81100.8
Iteration 256: Best Fit = 81100.8
Iteration 257: Best Fit = 81100.8
Iteration 258: Best Fit = 81100.8
Iteration 259: Best Fit = 81100.8
Iteration 260: Best Fit = 81100.8
Iteration 261: Best Fit = 81100.8
Iteration 262: Best Fit = 81100.8
Iteration 263: Best Fit = 81100.8
Iteration 264: Best Fit = 81100.8
Iteration 265: Best Fit = 81100.8
Iteration 266: Best Fit = 81100.8
Iteration 267: Best Fit = 81100.8
Iteration 268: Best Fit = 81100.8
Iteration 269: Best Fit = 81100.8
Iteration 270: Best Fit = 81100.8
Iteration 271: Best Fit = 81100.8
Iteration 272: Best Fit = 81100.8
Iteration 273: Best Fit = 81100.8
Iteration 274: Best Fit = 81100.8
Iteration 275: Best Fit = 81100.8
Iteration 276: Best Fit = 81100.8
Iteration 277: Best Fit = 81100.8
Iteration 278: Best Fit = 81100.8
Iteration 279: Best Fit = 81100.8
Iteration 280: Best Fit = 81100.8
Iteration 281: Best Fit = 81100.8
```

```
Iteration 282: Best Fit = 81100.8
Iteration 283: Best Fit = 81100.8
Iteration 284: Best Fit = 81100.8
Iteration 285: Best Fit = 81100.8
Iteration 286: Best Fit = 81100.8
Iteration 287: Best Fit = 81100.8
Iteration 288: Best Fit = 81100.8
Iteration 289: Best Fit = 81100.8
Iteration 290: Best Fit = 81100.8
Iteration 291: Best Fit = 81100.8
Iteration 292: Best Fit = 81100.8
Iteration 293: Best Fit = 81100.8
Iteration 294: Best Fit = 81100.8
Iteration 295: Best Fit = 81100.8
Iteration 296: Best Fit = 81100.8
Iteration 297: Best Fit = 81100.8
Iteration 298: Best Fit = 81100.8
Iteration 299: Best Fit = 81100.8
Iteration 300: Best Fit = 81100.8
>> BestSol.Sol

ans =

            V1: 4363
            W1: 671
            V0: 2304
            W0: 218
     Violation: 0.3420
             z: 1.2330e-05
    IsFeasible: 0

>> BestSol

BestSol =

    Tour: [2 1 2 2 2 1 2 2 2 2 2 1 2 2 1 2 2 2 1 1]
       x: [1 0 1 1 1 0 1 1 1 1 1 0 1 1 0 1 1 1 0 0]
     Fit: 8.1101e+04
     Sol: [1x1 struct]

>>
```