**حل مسئله Bin Packingبا الگوریتم علف هرز IWO در متلب**

**در مسئله** Bin Packing **اشیاء دارای حجم‌های مختلف باید در تعداد متناهی از جعبه از حجم V به شکلی که تعداد جعبه‌های استفاده‌شده کمینه شود قرار داده شود.**

**صورت مسئله:**

30 تا شی با حجم مشخص به این صورت داریم:

v = [6 3 4 6 8 7 4 7 7 5 5 6 7 7 6 4 8 7 8 8 2 3 4 5 6 5 5 7 7 12]

می خواهیم این اشیا را به گونه ای بسته بندی کنیم که هر بسته به اندازه حجم 30 بیشتر جا نگیرد، همه اشیا باید بسته بندی شوند و هیچ شی نباید در دو بسته قرار بگیرد.

**شرح کد:**

این سورس کد شامل 3 فایل می باشد که عبارتند از:

**:CreateModel.m** برای ایجاد مدل(همان اشیا) و مشخص کردن تعداد آن ها و حداکثر حجم هر بسته و مقدار دهی اولیه پارامتر های مدل مسیله از آن استفاده می شود.

function model = CreateModel()

model.v = [6 3 4 6 8 7 4 7 7 5 5 6 7 7 6 4 8 7 8 8 2 3 4 5 6 5 5 7 7 12];

ایجاد اشیا و مشخص کردن حجم آن ها و دادن آن به مدل

model.n = numel(model.v);

مشخص کردن تعداد اشیا و دادن آن به مدل

model.Vmax = 30;

مشخص کردن حداکثر حجم بسته و دادن آن به مدل

end

**BinPackingFit.m:**این تابع برای مشخص کردن میزان برازش راه حل می باشد.

function [z, sol] = BinPackingFit(x, model)

n = model.n; تعداد اشیا

v = model.v;حجم اشیا

Vmax = model.Vmax;ماکزیمم حجم هر بسته

[~, q]=sort(x,'descend');مرتب سازی نزولی جمعیت

Sep = find(q>n);پیدا کردن شی بزرگ تر از بزرگ ترین شی

From = [0 Sep] + 1;آرایه ای از 0 تا بزرگترین شی

To = [Sep 2\*n] - 1; آرایه ای از بزرگترین شی تا یکی کمتر از 2 برابر تعداد اشیا

B = {};

for i=1:n

Bi = q(From(i):To(i));برای همه اشیا اگر حجمشان بزرگتر از 0 است قبول است

if numel(Bi)>0

B = [B; Bi]; %#قبوله

end

end

nBin = numel(B);

nBin به تعداد B

Viol = zeros(nBin,1);

تخلف، یک آرایه به طول nBin

for i=1:nBin

حجم هر بسته با توجه به اشیا درونش

Vi = sum(v(B{i}));

ماکزیمم Vi تقسیم بر ماکزیمم حجم گنجایش هر بسته و 0 به ما میزان تخلف را می دهد

Viol(i) = max(Vi/Vmax-1, 0);

end

تخطی، میانگین Viol

MeanViol = mean(Viol);

یک ضریبی از تخطی می خواهیم که n تعداد اشیا است

alpha = 2\*n;

تعداد بسته ها + الفا ضربدر تخطی هزینه می شود.

K = nBin + alpha\*MeanViol;

برازندگی عکس هزینه با یک ضریب 1 در مخرج جمع می شود.

Z=1/(1+k);

این مقادیر را به راه حل می دهیم.

sol.nBin = nBin;

sol.B = B;

sol.Viol = Viol;

sol.MeanViol = MeanViol;

end

**iwo.m:** که فایل اصلی برنامه است که الگوریتم علف هرز در آن نوشته شده و پارامترهایش تعریف و مقداردهی شده است و دو تابع قبلی برای تعریف مسیله و محاسبه برازندگی در آن فراخوانی شده است.

clc;

clear;

close all;

تعریف مسیله

model = CreateModel(); Bin Packing ایجاد مدل

تابع محاسبه فیتنس

fitFunction = @(x) BinPackingFit(x, model);

nVar = 2\*model.n-1; تعداد متغیرهای تصمیم(تعداد خانه های آرایه که اشیا را درآن قرار می دهیم)

VarSize = [1 nVar]; اندازه ماتریس متغیرهای تصمیم(اندازه آرایه حاوی اشیا)

VarMin = 0; حد پایین مقادیر اشیا

VarMax = 1; حد بالا مقادیر اشیا

پارامترهای iwo

MaxIt = 500; ماکزیمم تعداد تکرار الگوریتم

nPop0 = 20; اندازه جمعیت اولیه

nPop = 100; حداکثر اندازه جمعیت

Smin = 0; حداقل تعداد دانه

Smax = 5; حداکثر تعداد دانه

Exponent = 2; نماینده کاهش واریانس

sigma\_initial = 1; مقدار اولیه انحراف معیار

sigma\_final = 0.001; مقدار نهایی انحراف معیار

%% Initialization

ساختار گیاه خالی

empty\_plant.Position = [];

empty\_plant.fit = [];

empty\_plant.Sol = [];

pop = repmat(empty\_plant, nPop0, 1); اولیه آرایه جمعیت

for i = 1:numel(pop)

موقعیت را مقدماتی کنید

pop(i).Position = unifrnd(VarMin, VarMax, VarSize);

ارزیابی

[pop(i).fit, pop(i).Sol] = fitFunction(pop(i).Position);

end

بهترین تاریخچه فیتنس را شروع کنید

Bestfits = zeros(MaxIt, 1);

IWO حلقه اصلی

for it = 1:MaxIt

آپدیت پراکندگی

sigma = ((MaxIt - it)/(MaxIt - 1))^Exponent \* (sigma\_initial - sigma\_final) + sigma\_final;

بهترین و بدترین مقدار فیتنس را بده

fits = [pop.fit];

Bestfit = max(fits);

Worstfit = min(fits);

جمعیت فرزندان را شروع کنید

newpop = [];

تولید مثل

for i = 1:numel(pop)

ratio = (pop(i).fit - Worstfit)/(Bestfit - Worstfit);

S = floor(Smin + (Smax - Smin)\*ratio);

for j = 1:S

مقداردهی اولیه فرزندان

newsol = empty\_plant;

ایجاد مکان تصادفی

newsol.Position = pop(i).Position + sigma \* randn(VarSize);

مرزهای پایین / بالا را اعمال کنید

newsol.Position = max(newsol.Position, VarMin);

newsol.Position = min(newsol.Position, VarMax);

فرزندان را ارزیابی کنید

[newsol.fit, newsol.Sol] = fitFunction(newsol.Position);

فرزندان را به جمعیت اضافه کنید

newpop = [newpop

newsol]; قبوله

end

end

ادغام جمعیت ها

pop = [pop

newpop];

مرتب سازی بر جمعیت

[~, SortOrder]=sort([pop.fit],'descend');

pop = pop(SortOrder);

حذف رقابتی (حذف اعضای اضافی)

if numel(pop)>nPop

pop = pop(1:nPop);

end

بهترین راه حل تاکنون پیدا شده را ذخیره کنید

BestSol = pop(1);

بهترین تاریخچه فیتنس را ذخیره کنید

Bestfits(it) = BestSol.fit;

نمایش اطلاعات هر تکرار

disp(['Iteration ' num2str(it) ': Best fit = ' num2str(Bestfits(it))]);

end

نتایج

figure;

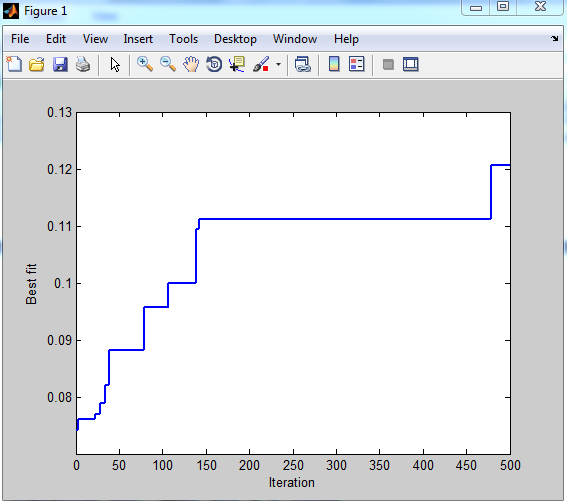
plot(Bestfits,'LineWidth',2);

xlabel('Iteration');

ylabel('Best fit');

بعد از اجرای کد iwo.m نتایج به شرح زیر است:

**نتایج:**

****

**Iteration 1: Best fit = 0.074074**

**Iteration 2: Best fit = 0.074074**

**Iteration 3: Best fit = 0.076087**

**Iteration 4: Best fit = 0.076087**

**Iteration 5: Best fit = 0.076087**

**Iteration 6: Best fit = 0.076087**

**Iteration 7: Best fit = 0.076087**

**Iteration 8: Best fit = 0.076087**

**Iteration 9: Best fit = 0.076087**

**Iteration 10: Best fit = 0.076087**

**Iteration 11: Best fit = 0.076087**

**Iteration 12: Best fit = 0.076087**

**Iteration 13: Best fit = 0.076087**

**Iteration 14: Best fit = 0.076087**

**Iteration 15: Best fit = 0.076087**

**Iteration 16: Best fit = 0.076087**

**Iteration 17: Best fit = 0.076087**

**Iteration 18: Best fit = 0.076087**

**Iteration 19: Best fit = 0.076087**

**Iteration 20: Best fit = 0.076087**

**Iteration 21: Best fit = 0.076087**

**Iteration 22: Best fit = 0.076923**

**Iteration 23: Best fit = 0.076923**

**Iteration 24: Best fit = 0.076923**

**Iteration 25: Best fit = 0.076923**

**Iteration 26: Best fit = 0.076923**

**Iteration 27: Best fit = 0.076923**

**Iteration 28: Best fit = 0.078947**

**Iteration 29: Best fit = 0.078947**

**Iteration 30: Best fit = 0.078947**

**Iteration 31: Best fit = 0.078947**

**Iteration 32: Best fit = 0.078947**

**Iteration 33: Best fit = 0.078947**

**Iteration 34: Best fit = 0.081967**

**Iteration 35: Best fit = 0.081967**

**Iteration 36: Best fit = 0.081967**

**Iteration 37: Best fit = 0.081967**

**Iteration 38: Best fit = 0.081967**

**Iteration 39: Best fit = 0.088235**

**Iteration 40: Best fit = 0.088235**

**Iteration 41: Best fit = 0.088235**

**Iteration 42: Best fit = 0.088235**

**Iteration 43: Best fit = 0.088235**

**Iteration 44: Best fit = 0.088235**

**Iteration 45: Best fit = 0.088235**

**Iteration 46: Best fit = 0.088235**

**Iteration 47: Best fit = 0.088235**

**Iteration 48: Best fit = 0.088235**

**Iteration 49: Best fit = 0.088235**

**Iteration 50: Best fit = 0.088235**

**Iteration 51: Best fit = 0.088235**

**Iteration 52: Best fit = 0.088235**

**Iteration 53: Best fit = 0.088235**

**Iteration 54: Best fit = 0.088235**

**Iteration 55: Best fit = 0.088235**

**Iteration 56: Best fit = 0.088235**

**Iteration 57: Best fit = 0.088235**

**Iteration 58: Best fit = 0.088235**

**Iteration 59: Best fit = 0.088235**

**Iteration 60: Best fit = 0.088235**

**Iteration 61: Best fit = 0.088235**

**Iteration 62: Best fit = 0.088235**

**Iteration 63: Best fit = 0.088235**

**Iteration 64: Best fit = 0.088235**

**Iteration 65: Best fit = 0.088235**

**Iteration 66: Best fit = 0.088235**

**Iteration 67: Best fit = 0.088235**

**Iteration 68: Best fit = 0.088235**

**Iteration 69: Best fit = 0.088235**

**Iteration 70: Best fit = 0.088235**

**Iteration 71: Best fit = 0.088235**

**Iteration 72: Best fit = 0.088235**

**Iteration 73: Best fit = 0.088235**

**Iteration 74: Best fit = 0.088235**

**Iteration 75: Best fit = 0.088235**

**Iteration 76: Best fit = 0.088235**

**Iteration 77: Best fit = 0.088235**

**Iteration 78: Best fit = 0.088235**

**Iteration 79: Best fit = 0.095745**

**Iteration 80: Best fit = 0.095745**

**Iteration 81: Best fit = 0.095745**

**Iteration 82: Best fit = 0.095745**

**Iteration 83: Best fit = 0.095745**

**Iteration 84: Best fit = 0.095745**

**Iteration 85: Best fit = 0.095745**

**Iteration 86: Best fit = 0.095745**

**Iteration 87: Best fit = 0.095745**

**Iteration 88: Best fit = 0.095745**

**Iteration 89: Best fit = 0.095745**

**Iteration 90: Best fit = 0.095745**

**Iteration 91: Best fit = 0.095745**

**Iteration 92: Best fit = 0.095745**

**Iteration 93: Best fit = 0.095745**

**Iteration 94: Best fit = 0.095745**

**Iteration 95: Best fit = 0.095745**

**Iteration 96: Best fit = 0.095745**

**Iteration 97: Best fit = 0.095745**

**Iteration 98: Best fit = 0.095745**

**Iteration 99: Best fit = 0.095745**

**Iteration 100: Best fit = 0.095745**

**Iteration 101: Best fit = 0.095745**

**Iteration 102: Best fit = 0.095745**

**Iteration 103: Best fit = 0.095745**

**Iteration 104: Best fit = 0.095745**

**Iteration 105: Best fit = 0.095745**

**Iteration 106: Best fit = 0.095745**

**Iteration 107: Best fit = 0.1**

**Iteration 108: Best fit = 0.1**

**Iteration 109: Best fit = 0.1**

**Iteration 110: Best fit = 0.1**

**Iteration 111: Best fit = 0.1**

**Iteration 112: Best fit = 0.1**

**Iteration 113: Best fit = 0.1**

**Iteration 114: Best fit = 0.1**

**Iteration 115: Best fit = 0.1**

**Iteration 116: Best fit = 0.1**

**Iteration 117: Best fit = 0.1**

**Iteration 118: Best fit = 0.1**

**Iteration 119: Best fit = 0.1**

**Iteration 120: Best fit = 0.1**

**Iteration 121: Best fit = 0.1**

**Iteration 122: Best fit = 0.1**

**Iteration 123: Best fit = 0.1**

**Iteration 124: Best fit = 0.1**

**Iteration 125: Best fit = 0.1**

**Iteration 126: Best fit = 0.1**

**Iteration 127: Best fit = 0.1**

**Iteration 128: Best fit = 0.1**

**Iteration 129: Best fit = 0.1**

**Iteration 130: Best fit = 0.1**

**Iteration 131: Best fit = 0.1**

**Iteration 132: Best fit = 0.1**

**Iteration 133: Best fit = 0.1**

**Iteration 134: Best fit = 0.1**

**Iteration 135: Best fit = 0.1**

**Iteration 136: Best fit = 0.1**

**Iteration 137: Best fit = 0.1**

**Iteration 138: Best fit = 0.1**

**Iteration 139: Best fit = 0.10937**

**Iteration 140: Best fit = 0.10937**

**Iteration 141: Best fit = 0.10937**

**Iteration 142: Best fit = 0.11111**

**Iteration 143: Best fit = 0.11111**

**Iteration 144: Best fit = 0.11111**

**Iteration 145: Best fit = 0.11111**

**Iteration 146: Best fit = 0.11111**

**Iteration 147: Best fit = 0.11111**

**Iteration 148: Best fit = 0.11111**

**Iteration 149: Best fit = 0.11111**

**Iteration 150: Best fit = 0.11111**

**Iteration 151: Best fit = 0.11111**

**Iteration 152: Best fit = 0.11111**

**Iteration 153: Best fit = 0.11111**

**Iteration 154: Best fit = 0.11111**

**Iteration 155: Best fit = 0.11111**

**Iteration 156: Best fit = 0.11111**

**Iteration 157: Best fit = 0.11111**

**Iteration 158: Best fit = 0.11111**

**Iteration 159: Best fit = 0.11111**

**Iteration 160: Best fit = 0.11111**

**Iteration 161: Best fit = 0.11111**

**Iteration 162: Best fit = 0.11111**

**Iteration 163: Best fit = 0.11111**

**Iteration 164: Best fit = 0.11111**

**Iteration 165: Best fit = 0.11111**

**Iteration 166: Best fit = 0.11111**

**Iteration 167: Best fit = 0.11111**

**Iteration 168: Best fit = 0.11111**

**Iteration 169: Best fit = 0.11111**

**Iteration 170: Best fit = 0.11111**

**Iteration 171: Best fit = 0.11111**

**Iteration 172: Best fit = 0.11111**

**Iteration 173: Best fit = 0.11111**

**Iteration 174: Best fit = 0.11111**

**Iteration 175: Best fit = 0.11111**

**Iteration 176: Best fit = 0.11111**

**Iteration 177: Best fit = 0.11111**

**Iteration 178: Best fit = 0.11111**

**Iteration 179: Best fit = 0.11111**

**Iteration 180: Best fit = 0.11111**

**Iteration 181: Best fit = 0.11111**

**Iteration 182: Best fit = 0.11111**

**Iteration 183: Best fit = 0.11111**

**Iteration 184: Best fit = 0.11111**

**Iteration 185: Best fit = 0.11111**

**Iteration 186: Best fit = 0.11111**

**Iteration 187: Best fit = 0.11111**

**Iteration 188: Best fit = 0.11111**

**Iteration 189: Best fit = 0.11111**

**Iteration 190: Best fit = 0.11111**

**Iteration 191: Best fit = 0.11111**

**Iteration 192: Best fit = 0.11111**

**Iteration 193: Best fit = 0.11111**

**Iteration 194: Best fit = 0.11111**

**Iteration 195: Best fit = 0.11111**

**Iteration 196: Best fit = 0.11111**

**Iteration 197: Best fit = 0.11111**

**Iteration 198: Best fit = 0.11111**

**Iteration 199: Best fit = 0.11111**

**Iteration 200: Best fit = 0.11111**

**Iteration 201: Best fit = 0.11111**

**Iteration 202: Best fit = 0.11111**

**Iteration 203: Best fit = 0.11111**

**Iteration 204: Best fit = 0.11111**

**Iteration 205: Best fit = 0.11111**

**Iteration 206: Best fit = 0.11111**

**Iteration 207: Best fit = 0.11111**

**Iteration 208: Best fit = 0.11111**

**Iteration 209: Best fit = 0.11111**

**Iteration 210: Best fit = 0.11111**

**Iteration 211: Best fit = 0.11111**

**Iteration 212: Best fit = 0.11111**

**Iteration 213: Best fit = 0.11111**

**Iteration 214: Best fit = 0.11111**

**Iteration 215: Best fit = 0.11111**

**Iteration 216: Best fit = 0.11111**

**Iteration 217: Best fit = 0.11111**

**Iteration 218: Best fit = 0.11111**

**Iteration 219: Best fit = 0.11111**

**Iteration 220: Best fit = 0.11111**

**Iteration 221: Best fit = 0.11111**

**Iteration 222: Best fit = 0.11111**

**Iteration 223: Best fit = 0.11111**

**Iteration 224: Best fit = 0.11111**

**Iteration 225: Best fit = 0.11111**

**Iteration 226: Best fit = 0.11111**

**Iteration 227: Best fit = 0.11111**

**Iteration 228: Best fit = 0.11111**

**Iteration 229: Best fit = 0.11111**

**Iteration 230: Best fit = 0.11111**

**Iteration 231: Best fit = 0.11111**

**Iteration 232: Best fit = 0.11111**

**Iteration 233: Best fit = 0.11111**

**Iteration 234: Best fit = 0.11111**

**Iteration 235: Best fit = 0.11111**

**Iteration 236: Best fit = 0.11111**

**Iteration 237: Best fit = 0.11111**

**Iteration 238: Best fit = 0.11111**

**Iteration 239: Best fit = 0.11111**

**Iteration 240: Best fit = 0.11111**

**Iteration 241: Best fit = 0.11111**

**Iteration 242: Best fit = 0.11111**

**Iteration 243: Best fit = 0.11111**

**Iteration 244: Best fit = 0.11111**

**Iteration 245: Best fit = 0.11111**

**Iteration 246: Best fit = 0.11111**

**Iteration 247: Best fit = 0.11111**

**Iteration 248: Best fit = 0.11111**

**Iteration 249: Best fit = 0.11111**

**Iteration 250: Best fit = 0.11111**

**Iteration 251: Best fit = 0.11111**

**Iteration 252: Best fit = 0.11111**

**Iteration 253: Best fit = 0.11111**

**Iteration 254: Best fit = 0.11111**

**Iteration 255: Best fit = 0.11111**

**Iteration 256: Best fit = 0.11111**

**Iteration 257: Best fit = 0.11111**

**Iteration 258: Best fit = 0.11111**

**Iteration 259: Best fit = 0.11111**

**Iteration 260: Best fit = 0.11111**

**Iteration 261: Best fit = 0.11111**

**Iteration 262: Best fit = 0.11111**

**Iteration 263: Best fit = 0.11111**

**Iteration 264: Best fit = 0.11111**

**Iteration 265: Best fit = 0.11111**

**Iteration 266: Best fit = 0.11111**

**Iteration 267: Best fit = 0.11111**

**Iteration 268: Best fit = 0.11111**

**Iteration 269: Best fit = 0.11111**

**Iteration 270: Best fit = 0.11111**

**Iteration 271: Best fit = 0.11111**

**Iteration 272: Best fit = 0.11111**

**Iteration 273: Best fit = 0.11111**

**Iteration 274: Best fit = 0.11111**

**Iteration 275: Best fit = 0.11111**

**Iteration 276: Best fit = 0.11111**

**Iteration 277: Best fit = 0.11111**

**Iteration 278: Best fit = 0.11111**

**Iteration 279: Best fit = 0.11111**

**Iteration 280: Best fit = 0.11111**

**Iteration 281: Best fit = 0.11111**

**Iteration 282: Best fit = 0.11111**

**Iteration 283: Best fit = 0.11111**

**Iteration 284: Best fit = 0.11111**

**Iteration 285: Best fit = 0.11111**

**Iteration 286: Best fit = 0.11111**

**Iteration 287: Best fit = 0.11111**

**Iteration 288: Best fit = 0.11111**

**Iteration 289: Best fit = 0.11111**

**Iteration 290: Best fit = 0.11111**

**Iteration 291: Best fit = 0.11111**

**Iteration 292: Best fit = 0.11111**

**Iteration 293: Best fit = 0.11111**

**Iteration 294: Best fit = 0.11111**

**Iteration 295: Best fit = 0.11111**

**Iteration 296: Best fit = 0.11111**

**Iteration 297: Best fit = 0.11111**

**Iteration 298: Best fit = 0.11111**

**Iteration 299: Best fit = 0.11111**

**Iteration 300: Best fit = 0.11111**

**Iteration 301: Best fit = 0.11111**

**Iteration 302: Best fit = 0.11111**

**Iteration 303: Best fit = 0.11111**

**Iteration 304: Best fit = 0.11111**

**Iteration 305: Best fit = 0.11111**

**Iteration 306: Best fit = 0.11111**

**Iteration 307: Best fit = 0.11111**

**Iteration 308: Best fit = 0.11111**

**Iteration 309: Best fit = 0.11111**

**Iteration 310: Best fit = 0.11111**

**Iteration 311: Best fit = 0.11111**

**Iteration 312: Best fit = 0.11111**

**Iteration 313: Best fit = 0.11111**

**Iteration 314: Best fit = 0.11111**

**Iteration 315: Best fit = 0.11111**

**Iteration 316: Best fit = 0.11111**

**Iteration 317: Best fit = 0.11111**

**Iteration 318: Best fit = 0.11111**

**Iteration 319: Best fit = 0.11111**

**Iteration 320: Best fit = 0.11111**

**Iteration 321: Best fit = 0.11111**

**Iteration 322: Best fit = 0.11111**

**Iteration 323: Best fit = 0.11111**

**Iteration 324: Best fit = 0.11111**

**Iteration 325: Best fit = 0.11111**

**Iteration 326: Best fit = 0.11111**

**Iteration 327: Best fit = 0.11111**

**Iteration 328: Best fit = 0.11111**

**Iteration 329: Best fit = 0.11111**

**Iteration 330: Best fit = 0.11111**

**Iteration 331: Best fit = 0.11111**

**Iteration 332: Best fit = 0.11111**

**Iteration 333: Best fit = 0.11111**

**Iteration 334: Best fit = 0.11111**

**Iteration 335: Best fit = 0.11111**

**Iteration 336: Best fit = 0.11111**

**Iteration 337: Best fit = 0.11111**

**Iteration 338: Best fit = 0.11111**

**Iteration 339: Best fit = 0.11111**

**Iteration 340: Best fit = 0.11111**

**Iteration 341: Best fit = 0.11111**

**Iteration 342: Best fit = 0.11111**

**Iteration 343: Best fit = 0.11111**

**Iteration 344: Best fit = 0.11111**

**Iteration 345: Best fit = 0.11111**

**Iteration 346: Best fit = 0.11111**

**Iteration 347: Best fit = 0.11111**

**Iteration 348: Best fit = 0.11111**

**Iteration 349: Best fit = 0.11111**

**Iteration 350: Best fit = 0.11111**

**Iteration 351: Best fit = 0.11111**

**Iteration 352: Best fit = 0.11111**

**Iteration 353: Best fit = 0.11111**

**Iteration 354: Best fit = 0.11111**

**Iteration 355: Best fit = 0.11111**

**Iteration 356: Best fit = 0.11111**

**Iteration 357: Best fit = 0.11111**

**Iteration 358: Best fit = 0.11111**

**Iteration 359: Best fit = 0.11111**

**Iteration 360: Best fit = 0.11111**

**Iteration 361: Best fit = 0.11111**

**Iteration 362: Best fit = 0.11111**

**Iteration 363: Best fit = 0.11111**

**Iteration 364: Best fit = 0.11111**

**Iteration 365: Best fit = 0.11111**

**Iteration 366: Best fit = 0.11111**

**Iteration 367: Best fit = 0.11111**

**Iteration 368: Best fit = 0.11111**

**Iteration 369: Best fit = 0.11111**

**Iteration 370: Best fit = 0.11111**

**Iteration 371: Best fit = 0.11111**

**Iteration 372: Best fit = 0.11111**

**Iteration 373: Best fit = 0.11111**

**Iteration 374: Best fit = 0.11111**

**Iteration 375: Best fit = 0.11111**

**Iteration 376: Best fit = 0.11111**

**Iteration 377: Best fit = 0.11111**

**Iteration 378: Best fit = 0.11111**

**Iteration 379: Best fit = 0.11111**

**Iteration 380: Best fit = 0.11111**

**Iteration 381: Best fit = 0.11111**

**Iteration 382: Best fit = 0.11111**

**Iteration 383: Best fit = 0.11111**

**Iteration 384: Best fit = 0.11111**

**Iteration 385: Best fit = 0.11111**

**Iteration 386: Best fit = 0.11111**

**Iteration 387: Best fit = 0.11111**

**Iteration 388: Best fit = 0.11111**

**Iteration 389: Best fit = 0.11111**

**Iteration 390: Best fit = 0.11111**

**Iteration 391: Best fit = 0.11111**

**Iteration 392: Best fit = 0.11111**

**Iteration 393: Best fit = 0.11111**

**Iteration 394: Best fit = 0.11111**

**Iteration 395: Best fit = 0.11111**

**Iteration 396: Best fit = 0.11111**

**Iteration 397: Best fit = 0.11111**

**Iteration 398: Best fit = 0.11111**

**Iteration 399: Best fit = 0.11111**

**Iteration 400: Best fit = 0.11111**

**Iteration 401: Best fit = 0.11111**

**Iteration 402: Best fit = 0.11111**

**Iteration 403: Best fit = 0.11111**

**Iteration 404: Best fit = 0.11111**

**Iteration 405: Best fit = 0.11111**

**Iteration 406: Best fit = 0.11111**

**Iteration 407: Best fit = 0.11111**

**Iteration 408: Best fit = 0.11111**

**Iteration 409: Best fit = 0.11111**

**Iteration 410: Best fit = 0.11111**

**Iteration 411: Best fit = 0.11111**

**Iteration 412: Best fit = 0.11111**

**Iteration 413: Best fit = 0.11111**

**Iteration 414: Best fit = 0.11111**

**Iteration 415: Best fit = 0.11111**

**Iteration 416: Best fit = 0.11111**

**Iteration 417: Best fit = 0.11111**

**Iteration 418: Best fit = 0.11111**

**Iteration 419: Best fit = 0.11111**

**Iteration 420: Best fit = 0.11111**

**Iteration 421: Best fit = 0.11111**

**Iteration 422: Best fit = 0.11111**

**Iteration 423: Best fit = 0.11111**

**Iteration 424: Best fit = 0.11111**

**Iteration 425: Best fit = 0.11111**

**Iteration 426: Best fit = 0.11111**

**Iteration 427: Best fit = 0.11111**

**Iteration 428: Best fit = 0.11111**

**Iteration 429: Best fit = 0.11111**

**Iteration 430: Best fit = 0.11111**

**Iteration 431: Best fit = 0.11111**

**Iteration 432: Best fit = 0.11111**

**Iteration 433: Best fit = 0.11111**

**Iteration 434: Best fit = 0.11111**

**Iteration 435: Best fit = 0.11111**

**Iteration 436: Best fit = 0.11111**

**Iteration 437: Best fit = 0.11111**

**Iteration 438: Best fit = 0.11111**

**Iteration 439: Best fit = 0.11111**

**Iteration 440: Best fit = 0.11111**

**Iteration 441: Best fit = 0.11111**

**Iteration 442: Best fit = 0.11111**

**Iteration 443: Best fit = 0.11111**

**Iteration 444: Best fit = 0.11111**

**Iteration 445: Best fit = 0.11111**

**Iteration 446: Best fit = 0.11111**

**Iteration 447: Best fit = 0.11111**

**Iteration 448: Best fit = 0.11111**

**Iteration 449: Best fit = 0.11111**

**Iteration 450: Best fit = 0.11111**

**Iteration 451: Best fit = 0.11111**

**Iteration 452: Best fit = 0.11111**

**Iteration 453: Best fit = 0.11111**

**Iteration 454: Best fit = 0.11111**

**Iteration 455: Best fit = 0.11111**

**Iteration 456: Best fit = 0.11111**

**Iteration 457: Best fit = 0.11111**

**Iteration 458: Best fit = 0.11111**

**Iteration 459: Best fit = 0.11111**

**Iteration 460: Best fit = 0.11111**

**Iteration 461: Best fit = 0.11111**

**Iteration 462: Best fit = 0.11111**

**Iteration 463: Best fit = 0.11111**

**Iteration 464: Best fit = 0.11111**

**Iteration 465: Best fit = 0.11111**

**Iteration 466: Best fit = 0.11111**

**Iteration 467: Best fit = 0.11111**

**Iteration 468: Best fit = 0.11111**

**Iteration 469: Best fit = 0.11111**

**Iteration 470: Best fit = 0.11111**

**Iteration 471: Best fit = 0.11111**

**Iteration 472: Best fit = 0.11111**

**Iteration 473: Best fit = 0.11111**

**Iteration 474: Best fit = 0.11111**

**Iteration 475: Best fit = 0.11111**

**Iteration 476: Best fit = 0.11111**

**Iteration 477: Best fit = 0.11111**

**Iteration 478: Best fit = 0.11111**

**Iteration 479: Best fit = 0.12069**

**Iteration 480: Best fit = 0.12069**

**Iteration 481: Best fit = 0.12069**

**Iteration 482: Best fit = 0.12069**

**Iteration 483: Best fit = 0.12069**

**Iteration 484: Best fit = 0.12069**

**Iteration 485: Best fit = 0.12069**

**Iteration 486: Best fit = 0.12069**

**Iteration 487: Best fit = 0.12069**

**Iteration 488: Best fit = 0.12069**

**Iteration 489: Best fit = 0.12069**

**Iteration 490: Best fit = 0.12069**

**Iteration 491: Best fit = 0.12069**

**Iteration 492: Best fit = 0.12069**

**Iteration 493: Best fit = 0.12069**

**Iteration 494: Best fit = 0.12069**

**Iteration 495: Best fit = 0.12069**

**Iteration 496: Best fit = 0.12069**

**Iteration 497: Best fit = 0.12069**

**Iteration 498: Best fit = 0.12069**

**Iteration 499: Best fit = 0.12069**

**Iteration 500: Best fit = 0.12069**

**>>**

**>> BestSol.Sol**

**ans =**

**nBin: 7**

**B: {7x1 cell}**

**Viol: [7x1 double]**

**MeanViol: 0.0048**

**>> BestSol.Sol.B**

**ans =**

**[1x6 double]**

**[1x5 double]**

**[1x5 double]**

**[1x3 double]**

**[1x3 double]**

**[1x3 double]**

**[1x5 double]**

**>> BestSol.Position**

**ans =**

**Columns 1 through 11**

**0.7348 1.0000 0 1.0000 0.5265 0.0016 0.8441 0.8995 1.0000 0.2056 0.3547**

**Columns 12 through 22**

**0.7888 0.3335 0.0041 0.0015 0.4592 0.7496 0.7482 1.0000 0.4133 0.7251 1.0000**

**Columns 23 through 33**

**1.0000 0.3074 0.8923 0.0011 0.9644 0.8626 0.0956 0.1176 0.5587 0.3680 0.8240**

**Columns 34 through 44**

**0.3812 0.9871 0.0065 0.0286 1.0000 0.9976 0.0618 0.6577 0.0095 0.9983 0.0200**

**Columns 45 through 55**

**0.9944 0.6313 0.0108 0.9812 0.2745 0.0118 0.9913 0.9953 0.0147 0.9983 0.9864**

**Columns 56 through 59**

**0.9723 1.0000 0.0105 0.9853**

**>>**