Равновесная упаковка шаров в шар минимального радиуса

П.И. Стецюк[1], Т.Е. Романова[2]

[1] Институт кибернетики им. В.М. Глушкова, Киев
[2] Институт проблем машиностроения, Харьков

Международная конференция
"'Дискретная оптимизация и исследование операций"'
Новосибирск (Академгородок), 24 – 28 июня 2013

План доклада

1. Формулировка задачи и приложения
2. Математическая модель
3. Метод решения
4. Вычислительные результаты

Формулировка задачи I

Имеются n-мерные евклидовы шары S_i с радиусами r_i и весами w_i, $i = 1, \ldots, m$. Полагаем, что центр тяжести шара S_i находится в его центре.

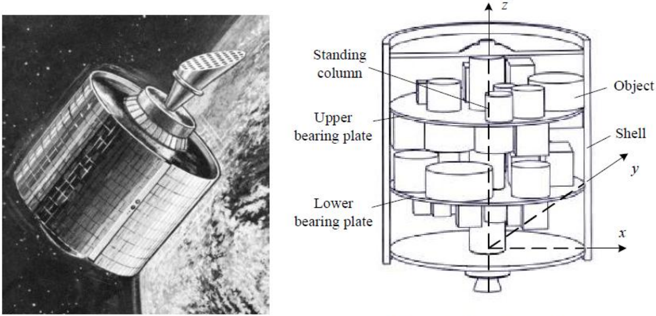
Равновесной упаковкой n-мерных шаров S_i, $i = 1, \ldots, m$ будем называть такую их упаковку в шар S, чтобы радиус шара S был минимальным и центр тяжести семейства шаров S_i, $i = 1, \ldots, m$, совпадал с центром шара S.

nd-задача

Найти равновесную упаковку n-мерных шаров S_i, $i = 1, \ldots, m$.

Приложения 2d-задач I

при моделировании спутников (космическая инженерия)



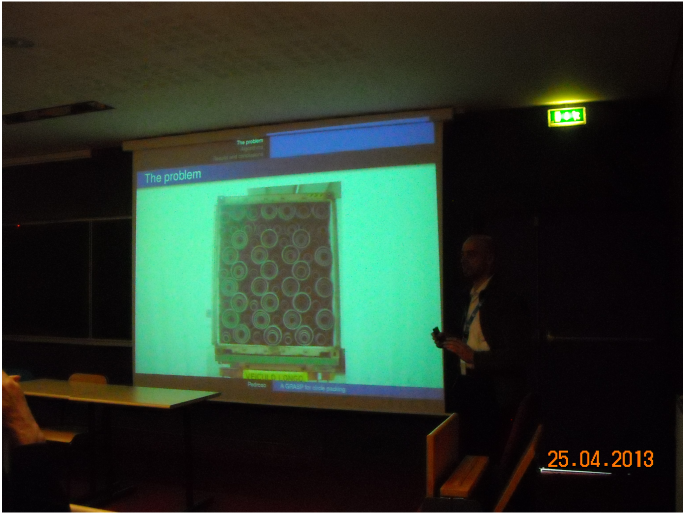
a. the international communication satellite

b. the simplified satellite module

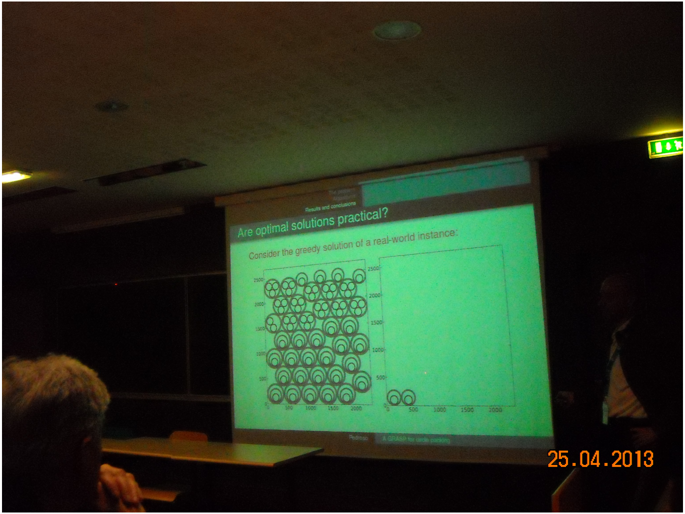
Fasano G., Pinter J.D., eds., Modeling and Optimization in Space Engineering. Springer Optimization and Its Applications. – Springer. – New York. – 2012. – 404 p.

Приложения 2d-задач II

при упаковке телескопических грузов (ESICUP-2013)



Приложения 2d-задач III



Математическая модель I

Многоэкстремальная задача нелинейного программирования:

$$r^* = \min_{x \in E^{m \times n}, r \geq R} r \tag{1}$$

при ограничениях

$$\sum_{k=1}^{n} x_{ik}^2 \leq (r - r_i)^2, \quad i = 1, \ldots, m, \tag{2}$$

$$\sum_{k=1}^{n} (x_{ik} - x_{jk})^2 \geq (r_i + r_j)^2, \quad 1 \leq i < j \leq m, \tag{3}$$

$$\sum_{i=1}^{m} \lambda_i x_{ik} = 0, \quad k = 1, \ldots, n. \tag{4}$$

Здесь неизвестные величины – $x_i = (x_{i1}, \ldots, x_{in})$ – центр шара S_i, r – радиус шара S,

известные величины – $\lambda_i = w_i / \sum_{i=1}^{m} w_i$, $i = 1, \ldots, m$, $R = \max_{i=1,\ldots,m} r_i$ – очевидная нижняя

граница на искомый радиус.

Математическая модель II

Ограничения в задаче (1)–(4) означают следующее:

$$\varphi_i(x,r) = \sum_{k=1}^{n} x_{ik}^2 - (r - r_i)^2 \leq 0, \quad i = 1, \ldots, m, \qquad (2')$$

т.е. шары S_i, $i = 1, \ldots, m$ содержатся внутри шара S,

$$\phi_{ij}(x) = -\sum_{k=1}^{n}(x_{ik} - x_{jk})^2 + (r_i + r_j)^2 \leq 0, \quad 1 \leq i < j \leq m, \quad (3')$$

т.е. никакие два шара из S_i, $i = 1, \ldots, m$ не пересекаются,

$$\Phi_k(x) = \sum_{i=1}^{m} \lambda_i x_{ik} = 0, \quad k = 1, \ldots, n. \qquad (4')$$

т.е. центр тяжести шаров S_i, $i = 1, \ldots, m$ находится в начале координат (центре шара S).

Метод решения I

Step 1. Строим новую функцию

$$\begin{aligned} f(x,r) &= r + P_1\left(\sum_{i=1}^{m}\max\{0,\varphi_i(x,r)\} + \sum_{i=1}^{m}\sum_{j>i}^{m}\max\{0,\phi_{ij}(x)\}\right) \\ &+ P_2\sum_{k=1}^{n}\max\{0,-\Phi_k(x)-\varepsilon_k,\Phi_k(x)-\varepsilon_k\} \\ &+ P_3\max\{0,-r+R\}, \end{aligned}$$

где P_1, P_2, P_3 – негладкие штрафы (неотрицательные)

Step 2. Заменяем задачу (1)–(4) следующей безусловной задачей негладкой оптимизации

$$\min_{x\in E^{m\times n},r} f(x,r). \tag{5}$$

Метод решения II

Для реализации модели (5)

1) генерируем стартовые точки (rand)
2) применяем r-алгоритм[1] для каждой стартовой точки

1

Shor N.Z. Nondifferentiable optimization and polynomial problems. – Boston; Dordrecht; London: Kluwer Academic Publishers, 1998. – 412 p.

Тестовый пример Романовой

2d-задача для пяти неравных кругов:

- $n = 2$, $m = 5$

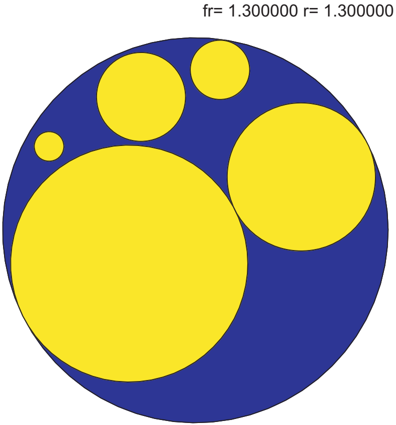
Круги имеют различные радиусы и массы:

- $r_1 = 0.1$, $r_2 = 0.2$, $r_3 = 0.3$, $r_4 = 0.5$, $r_5 = 0.8$
- $m_1 = 0.0785$, $m_2 = 0.314$, $m_3 = 0.7065$, $m_4 = 1.9625$, $m_5 = 5.024$

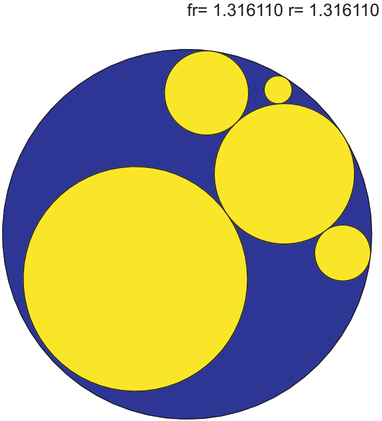
Точность для компонент центра тяжести

- $\varepsilon_1 = \varepsilon_2 = 0.0001$

Оптимальное размещение кругов



без учета центра тяжести



с учетом центра тяжести

Тестовый пример Пшеничного

2d-задача для семи кругов:

- $n = 2,\ m = 7$

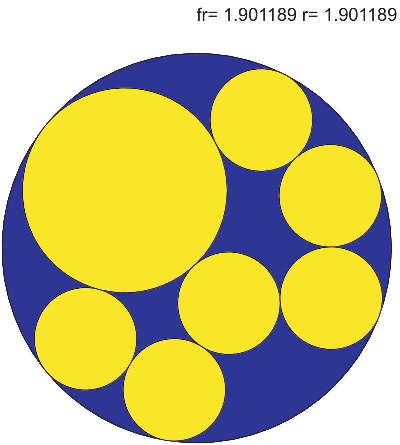
Круги имеют различные радиусы и массы:

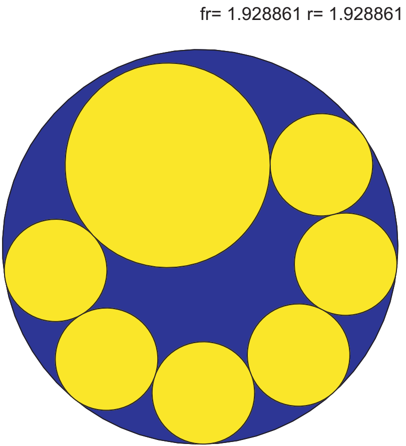
- $r_1 = 1,\ r_2 = r_3 = r_4 = r_5 = r_6 = r_7 = 0.5$
- $m_1 = 4,\ m_2 = m_3 = m_4 = m_5 = m_6 = m_7 = 1$

Точность для компонент центра тяжести

- $\varepsilon_1 = \varepsilon_2 = 0.0001$

Оптимальное размещение кругов





без учета центра тяжести

с учетом центра тяжести

Наши планы

адаптировать алгоритм для средних и больших задач, подкрепить его оценками снизу для r^*.

Thanks

Работа поддержана совместным грантом НТЦУ и НАНУ (проект № 5710).

Вопросы?

СПАСИБО ЗА ВНИМАНИЕ!

BACK UP SLIDES: Об octave-функции ralgb5

```
%     ralgb5 реализует $r(\alpha)$-алгоритм с адаптивным шагом,
%     использует подготовленную пользователем octave-функцию
%     function [f,g] = calcfg(x), которая вычисляет значения
%     функции $f=f(x)$ и её субградиента $g(x)$ в точке $x$.

%  Входные параметры:
%    calcfg -- имя функции calcfg(x) для вычисления f и g
%    x  -- начальная точка x0(1:n) (на выходе портится)
%    alpha -- коэффициент растяжения пространства
%    h0, nh, q1, q2 -- параметры адаптивной регулировки шага
%    epsx, epsg, maxitn -- параметры останова

%  Выходные параметры:
%    xr -- найденная точка минимума xr(n)
%    fr -- значение функции в точке минимума
%    itn -- число затраченных итераций
%    ncalls -- число вызовов функции calcfg
%    istop -- код останова (2=epsg,3=epsx,4=maxitn,5=error)
```

BACK UP SLIDES: Octave-функция ralgb5

```
function [xr,fr,itn,ncalls,istop]=ralgb5(calcfg,x,alpha,h0,q1,
                                         q2,nh,epsg,epsx,maxitn);
itn=0; hs=h0; B=eye(length(x)); xr=x;                       # row001
ncalls = 1; [fr,g0] = calcfg(xr);                           # row002
printf("itn %4d f %14.6e fr %14.6e ls %2d ncalls %4d\n",    # row003
       itn, fr, fr, 0, ncalls);
if(norm(g0) < epsg) istop = 2;  return; endif               # row004
for (itn = 1:maxitn)                                        # row005
   dx = B * (g1 = B' * g0)/norm(g1);                        # row006
   d = 1; ls = 0; ddx = 0;                                  # row007
   while (d > 0)                                            # row008
      x -= hs * dx; ddx += hs * norm(dx);                   # row009
      ncalls ++; [f, g1] = calcfg(x);                       # row010
      if (f < fr)  fr = f; xr = x; endif                    # row011
      if(norm(g1) < epsg) istop = 2;  return; endif         # row012
      ls ++; (mod(ls,nh)==0) && (hs *= q2);                 # row013
      if(ls > 500) istop = 5;  return; endif                # row014
      d = dx' * g1;                                         # row015
   endwhile                                                 # row016
   (ls == 1) && (hs *= q1);                                 # row017
   printf("itn %4d f %14.6e fr %14.6e ls %2d ncalls %4d\n", # row018
          itn, f, fr, ls, ncalls);
   if(ddx < epsx) istop = 3;  return; endif                 # row019
   xi = (dg = B' * (g1 - g0) )/norm(dg);                    # row020
   B += (1 / alpha - 1) * B * xi * xi';                     # row021
   g0 = g1;                                                 # row022
endfor                                                      # row023
istop = 4;                                                  # row024
endfunction
```

BACK UP SLIDES: Выбор параметров для ralgb5

При минимизации негладких функций рекомендуется:

$$\alpha = 2 \div 3, \ h_0 = 1.0, \ q_1 = 1.0, \ q_2 = 1.1 \div 1.2, \ n_h = 2 \div 3.$$

Если известна априорная оценка расстояния от начальной точки x_0 до точки минимума x^*, то начальный шаг h_0 целесообразно выбирать порядка $\|x_0 - x^*\|$.
При минимизации гладких функций рекомендуется

$$q_1 = 0.8 \div 0.95.$$

При таком выборе параметров, как правило, число спусков по направлению редко превосходит два, а за n шагов точность по функции улучшается в три-пять раз.