Operations Research and Fuzziology 运筹与模糊学, 2014, 4, 7-14
http://dx.doi.org/10.12677/orf.2014.41002 Published Online February 2014 (http://www.hanspub.org/journal/orf.html)

An Efficient Lagrangian Smoothing Heuristic for the p-Dispersion-Sum Problem

Yujun Gong, Huimin Zou

School of Mathematics and System Sciences, Beihang University, Beijing
Email: gyjgongzuo@163.com, 245278793@qq.com

Received: Apr. 24th, 2013; revised: May 5th, 2013; accepted: May 7th, 2013

Abstract: In this article, we propose a Lagrangian smoothing algorithm for the p-dispersion-sum problem (PDSP), a problem to locate p facilities at some of n predefined locations by maximizing the distance sum between the p established facilities, where the continuation subproblems are solved by the truncated Frank-Wolfe algorithm. We make the iteration from Lagrangian function to penalty function by controlling a parameter. We establish practical stopping criteria and prove that our algorithm finitely terminates at a KKT point. Compared to the discrete algorithm, the smoothing algorithm is free from the constraints of the number of nodes and more extensive. Numerical results indicate that our approach outperforms good and rapid for solving randomly generated problems in dimensional n ≥ 100.

Keywords: Quadratic Program; Lagrangian Smoothing; Frank-Wolfe Algorithm; Heuristic

离散 p-扩散问题的连续化算法

龚玉君，邹慧敏

北京航空航天大学，数学与系统科学学院，北京
Email: gyjgongzuo@163.com, 245278793@qq.com

收稿日期：2013 年 4 月 24 日；修回日期：2013 年 5 月 5 日；录用日期：2013 年 5 月 7 日

摘　要：p-扩散问题主要研究在事先定义的 n 个位置中如何选取 p 个位置，使得这 p 个位置之间的距离之和最大，具有很实际的应用价值。本文提出了解决 p-扩散问题的一种连续化方法，采用连续的算法解决离散问题策略，通过控制一个参数使得从拉格朗日函数向罚函数过渡迭代求解。这种算法的子问题采用了截断的 Frank-Wolfe 算法来避免收敛过慢，相比较离散算法，不受结点数量的制约，更具有广泛性。本文建立了有效的迭代终止准则，并且证明了这种算法最终会收敛在一个 KKT 点。最后，针对这种连续化算法，我们做了大量数值实验，验证了算法的可行性与有效性。

关键词：二次规划问题；拉格朗日光滑算法；启发式；连续算法

1. 引言

p-扩散问题主要是研究在预先定义的 n 个工厂地址中选取 p 个位置，使得选取的 p 个工厂彼此之间的距离之和最大化的问题。也称为 p-dispersion-sum problem(PDSP)定位问题，是典型的二次规划问题。在现代社会生活中，二次规划在工程设计、经济学、统计学及设施布局等诸多领域有着广泛应用[1]，许多非线性规划问题可

以转化为此类问题的求解，如整数线性规划、整数二次规划等，因此研究这类问题既具有理论意义，又具有实用价值[2]。我们研究的 p-扩散问题可以广泛的应用到电信、仓库库位、军事防御等方面的选址问题。有十分广泛的应用前景，也称为最重 K 子图问题[3]。

2. 正文

2.1. p-扩散问题简介

PDSP 问题是整数规划问题，有很多人进行了不同方向的研究，对于这个问题的主要研究有如下，Krarup[4]研究了简单的 0-1 模型，即各个工厂间距离为 1，Kortsarz andPeleg[5]、Asahiro[6]、Hassin[7]、Srivastav and Wolf[8]也进行了相同的研究，Kincaid[9]提出了基于模拟退火处理的启发式算法，但是目前来看，对于这个问题并没有得到较好的结果，Pisinger[10]一文中，则提出了几个关于当 $n = 60$ 时的好的解决办法。Billionnet and Soutif[11]中，则讨论了一些当 $n \geq 100$ 时的解决办法。

p-扩散问题是典型的非线性规划问题。目前，对求解非线性规划问题已经有很多的算法，通常可分为间接算法和直接算法。间接算法通过处理从原问题中得到的若干个线性规划来求解非线性问题，而直接算法则直接处理原问题即采用离散方法解决。

2.2. p-扩散问题基本数学模型

在本文中我们使用 x_j 表示一个工厂是否被选中，于是 PDSP 问题可以转化成，求一个二次整数方程的最大值问题

$$\text{PDSP} := \max \sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij} x_i x_j \tag{1}$$

$$\text{s.t.} \ \sum_{i=1}^{n} x_i = p \tag{2}$$

$$x_i \in \{0,1\}, i = 1,2,\cdots,n. \tag{3}$$

我们设定矩阵 $D = (d_{ij}), i, j = 1,2,\cdots,n.$ 表示 n 个工厂中 i 工厂与 j 工厂之间的距离。D 矩阵是对称矩阵，D 具有这样的性质：$x^{\mathrm{T}} D x = x^{\mathrm{T}} \frac{D + D^{\mathrm{T}}}{2} x$，其实这在实际问题中也很好理解，在大多数工厂选址问题中，我们一般认为 $d_{ii} = 0$，$i = 1,2,\cdots,n$ 在本文中我们也按做同样处理。

则原问题(1.1)就可以改写成

$$\begin{aligned} &\text{PDSP} := \max x^{\mathrm{T}} D x \\ &\text{s.t.} \ \sum_{i=1}^{n} x_i = p \\ &x_i \in \{0,1\}, i = 1,2,\cdots,n. \end{aligned} \tag{4}$$

也可以写成：

$$\begin{aligned} &\text{PDSP} := \max x^{\mathrm{T}} D x \\ &\text{s.t.} \ e^{\mathrm{T}} x = p \\ &x_i^2 = x_i, i = 1,2,\cdots,n. \end{aligned} \tag{5}$$

PDSP 问题是整数规划问题，现在解决该类问题比较典型的解法是广泛应用于整数规划问题的分支定界法，分支定界方法是通过对非整变量附加线性不等式约束(整数)使得原问题转化成两个子问题，继续求解定界，重复下去，直到得出最优解为止的算法[12]。

在本文中，我们使用一种较新的启发式拉格朗日算法，在 Frank-Wolfe 连续算法的基础上，做一些改进，本文会使用一个简单的例子通过计算，最终证明 Frank-Wolfe 连续算法会终止在一个 KKT 点并且得到的结果不错。

事先说明：对于 n 维向量 a 和 b， $a \le b$ 当且仅当 $a_i \le b_i, i = 1, 2, \cdots, n$。

二维范数定义为：$\|a\|_2 = \sqrt{a^{\mathrm{T}} a}$， $\|D\|_\infty = \max 1 \le i \le n \sum_{j=1}^{n} |A_{ij}|$。

标记函数 $sign(a_i) = \begin{cases} 1 & a_i \ge 0 \\ -1 & a_i < 0 \end{cases}$。

2.3. 离散 p-扩散问题的拉格朗日光滑化算法

在 Xia[13]中，介绍了一种最大割(max-cut)问题的连续的拉格朗日光滑算法连续算法，在 Xia[13]论文的启发下，本论文采用类似方法。

对于 PDSP 问题(5)等价于

$$\begin{aligned} &\mathrm{PDSP} := \max x^{\mathrm{T}} D x \\ &\text{s.t. } \ x^{\mathrm{T}} x = p \\ &x_i^2 = x_i, i = 1, 2, \cdots, n. \end{aligned} \tag{6}$$

对约束(6)引进拉格朗日乘子 μ， 我们得到拉格朗日函数为

$$L(x;\mu) = x^{\mathrm{T}} (D + \mu I) x - \mu p \tag{7}$$

其中乘子 μ 使得 $D + \mu I \le 0$ 且 $\mu \le 0$ 的参数。

对于 PDSP 问题(5)的柯朗惩罚函数为：

$$x^{\mathrm{T}} D x + \frac{1}{1-t}\left(x^{\mathrm{T}} x - e^{\mathrm{T}} x\right) = x^{\mathrm{T}} D x + \frac{1}{1-t} x^{\mathrm{T}} x - \frac{p}{1-t}$$

去掉常数项，我们记

$$P(x;t) = x^{\mathrm{T}} D x + \frac{1}{1-t} x^{\mathrm{T}} x \tag{8}$$

则它们的联合问题可以写成：

$$H(x;\mu;t) = tP(x;t) + (1-t)L(x;\mu) = -(1-t)\mu p + x^{\mathrm{T}}\left(D + t(1-t)^{-1} I + (1-t)\mu I\right)x \tag{9}$$

下面我们求解如下参数问题 $Q(t;\mu)$：

$$\begin{aligned} &\max_{x \in [0,e]} H(x;\mu;t) \\ &\text{s.t. } \ e^{\mathrm{T}} x = p \end{aligned} \tag{10}$$

使用标准的 Frank-Wolfe 算法[10]来解决对于含给定的参数 t 的最优化问题(10)，对于任意给定的收敛点 x_k 用泰勒展开近似代替目标方程，去除常数项后产生一个线性规划问题。

我们首先证明在 $t \in (t^*, 1)$ 的时候，存在参数 $t^* \in (0,1)$ 使得问题(5)与问题(10)等价，为此我们需要一些简单的引理。

引理 1 多面体 $\left\{x \in [0,1]^n ; e^{\mathrm{T}} x = p\right\}$ 的顶点集为 $V = \left\{x \in \{0,1\}^n ; e^{\mathrm{T}} x = p\right\}$。

证明：假设存在点 $x \in V$ 不是顶点，则存在顶点 $y \in V$， $z \in V$ 使得 $x = \alpha y + (1-\alpha) z$。

对于 $\forall i$，如果 $x_i = 0$，则推出 $\alpha y_i + (1-\alpha) z_i = 0$，由于 $y, z \in [0,1]^n$ 所以 $y_i \ge 0$， $z_i \ge 0$ 所以可以推出 $y_i = z_i = 0$。

如果 $x_i = 1$ 则推出 $\alpha y_i + (1-\alpha) z_i = 1$，由于 $y_i \le 1$， $z_i \le 1$，所以可以推出 $y_i = z_i = 1$ 所以 $x = y = z$，假设不成立，即点集 V 都是顶点。

假设存在点 $x \notin V$ 是顶点，有 $e^{\mathrm{T}}x = p$，因为 p 是整数，所以至少存在两个指标 i, j，使得 $0 < x_i, x_j < 1$

构造这样的顶点 $y \in V$，$z \in V$ 使得

$$\begin{cases} y_k = x_k, k \ne i, j \\ y_i = x_i - \varepsilon \\ y_j = x_j + \varepsilon \end{cases}, \begin{cases} z_k = x_k, k \ne i, j \\ z_i = x_i + \varepsilon \\ z_j = x_j - \varepsilon \end{cases}$$

则有 $x = \dfrac{y+z}{2}$，即 x 不是顶点，假设不成立，证明完成。

引理 2 存在 $t^* \in (0,1)$，对任意 $t \in (t^*,1)$，PDSP 问题(5)与问题(6)等价，

证明：由引理 1，只需要证明问题(5)是凹规划问题，注意到问题(10)的黑塞矩阵为

$$\nabla^2 H(x;\mu;t) = D + \frac{t}{1-t} I + (1-t)\mu I$$

显然存在一个 $t^* \in (0,1)$ 当 $t \ge t^*$ 时都有 $\nabla^2 H(x;\mu;t)$ 是正定矩阵，最优解在顶点取的，因此问题(5)与问题(10)的最优解只相差有限的常数项，证明完成。

对于任意给定的收敛点 x_k 用泰勒展开近似代替目标方程，去除常数项后产生一个线性规划问题。

$$\begin{aligned} &\max \nabla_x H(x_k;\mu;t)^{\mathrm{T}} x \\ &\text{s.t. } 0 \le x \le e, e^{\mathrm{T}} x = p \end{aligned} \tag{11}$$

下面我们说明由于其特殊结构，线性规划子问题(11)可以快速求解。

首先注意到，该问题等价于寻找前 p 个最大的分量 $\left(\nabla_x H(x^*;\mu;t)\right)_i$，依次挨个找出最大的分量，这样计算的复杂度为 $O(np)$，另外我们还可以依次挨个删除最小的分量，这样计算的复杂度为 $O(n(n-p))$。当 p 靠近 $n/2$ 时，复杂度接近 $O(n^2)$，此时我们还有更快的做法：

对向量 $\nabla_x H(x^*;\mu;t)$ 的 n 个分量进行从大到小排序，选取前 p 个 $\left(\nabla_x H(x^*;\mu;t)\right)_i$，令相应的 $(x_i^*)_i = 1$，即：

$$(x_k^*)_i = \begin{cases} 1 & \text{如果}\left(\nabla_x H(x_k;\mu;t)\right)_i \text{排在}p\text{以内}, \\ 0, & \text{否则}. \end{cases} \quad (i = 1, \cdots, n) \tag{12}$$

因此求解线性规划子问题(3.6)复杂度为 $\min\left\{O(n\ln n), O(np), O(n(n-p))\right\}$。

这样选择出来的 x_k^* 用于构造下一个搜索方向 $d_x = x_k^* - x_k$，在这里我们使用同样的线性搜索方法：

$$\alpha^* = \arg\max_{\alpha\in[0,1]} H(x_k + \alpha d_k;\mu;t) \tag{13}$$

产生下一个迭代点

$$x_{k+1} = x_k + \alpha^* d_k \tag{14}$$

很容易证明由上述 Frank-Wolfe 算法产生的点列 $\{x_k\}$ 最终收敛于点 x^*，是问题(1.4)的一个 KKT 收敛点，但是由于 Frank-Wolfe 是线性收敛，速度很慢获得 x^* 十分耗时间，而求解的参数 t 子问题并不一定是我们最终的问题，因此我们同样采用截断的 Frank-Wolfe 算法来获得近似的 x^*，

为了介绍一个实际的停止迭代点，我们需要一些辅助定理。

引理 1 $x^* \in \{0,1\}^n$ 是问题 $Q(t;\mu)$ (10)的 KKT 点，当且仅当

$$\max_{i\in I^*}\left(\nabla_x H(x^*;\mu;t)_i\right) \le \min_{i\in J^*}\left(\nabla_x H(x^*;\mu;t)\right)_i \tag{15}$$

其中 $I^* \in \{i : x_i^* = 0\}$，$J^* \in \{i : x_i^* = 1\}$。

证明：问题 $Q(t;\mu)$ (10)的 KKT 的条件如下

$$\begin{cases} -\left(\nabla_x H\left(x^*;\mu;t\right)\right)_i - \lambda + \alpha_i^* - \beta_i^* = 0 & i = 1,\cdots,n \\ \alpha_i^* \ge 0 & i = 1,\cdots,n \\ \beta_i^* \ge 0 & i = 1,\cdots,n \\ 0 \le x_i^* \le 1 & i = 1,\cdots,n \\ \alpha_i^*(1 - x_i^*) = 0 & i = 1,\cdots,n \\ \beta_i^* x_i^* = 0 & i = 1,\cdots,n \\ e^T x = p \end{cases} \tag{16}$$

简化上述 KKT 条件

$$\begin{cases} \alpha_i^* = \nabla_x H\left(x^*;\mu;t\right)_i x_i^* + \lambda x_i^* \ge 0, & i = 1,\cdots,n, \\ \beta_i^* = -\nabla_x H\left(x^*;\mu;t\right)_i \left(1 - x_i^*\right) - \lambda\left(1 - x_i^*\right) \ge 0, & i = 1,\cdots,n. \end{cases} \tag{17}$$

由于 $x^* \in \{0,1\}^n$，上式又等价于

$$\begin{cases} \text{当} x_i = 1 \text{时}, -\nabla_x H\left(x^*;\mu;t\right)_i - \lambda \ge 0, & i = 1,\cdots,n, \\ \text{当} x_i = 1 \text{时}, -\nabla_x H\left(x^*;\mu;t\right)_i - \lambda \le 0, & i = 1,\cdots,n. \end{cases} \tag{18}$$

即在以下区间内有值存在：

$$\max_{i\in I^*}\left(\nabla_x H\left(x^*;\mu;t\right)_i\right) \le \min_{i\in J^*}\left(\nabla_x H\left(x^*;\mu;t\right)\right)_i$$

另一方面，如果(15)成立时，在(17)中定义的 α^*，β^* 以及区间 $\left[\max_{i\in I^*}\left(\nabla_x H\left(x^*;\mu;t\right)_i\right), \min_{i\in J^*}\left(\nabla_x H\left(x^*;\mu;t\right)\right)_i\right]$ 中的取任一值作为乘子 λ (3.11)也满足，证明结束。

引理 2 如果 $x^* \in \{0,1\}^n$ 是 $Q(t;\mu)$ (10)的 KKT 点，则对于所有的 $t' > t$，x^* 也是问题 $Q(t';\mu)$ 的 KKT 点。

证明：根据(9)，

$$\nabla_x H\left(x^*;\mu;t\right)_i = \nabla_x H\left(x^*;\mu;t'\right)_i + 2\left(\left(t(1-t)^{-1} - t'(1-t')^{-1}\right)I + (t'-t)\mu I\right)x_i^*$$

由于 $t' > t$ 且 $\mu \le 0$ 我们有

$$\nabla_x H\left(x^*;\mu;t'\right)_i = \nabla_x H\left(x^*;\mu;t\right)_i, \quad \forall i \in I^*$$

注意 $t' > t$ 蕴含 $t(1-t)^{-1} < t'(1-t')^{-1}$

$$\nabla_x H\left(x^*;\mu;t'\right)_j \ge \nabla_x H\left(x^*;\mu;t\right)_j, \quad \forall j \in J^*$$

$$\max_{i\in I^*}\left(\nabla_x H\left(x^*;\mu;t'\right)_i\right) = \max_{i\in I^*}\left(\nabla_x H\left(x^*;\mu;t\right)_i\right) \le \min_{i\in J^*}\left(\nabla_x H\left(x^*;\mu;t\right)\right)_i \le \min_{i\in J^*}\left(\nabla_x H\left(x^*;\mu;t'\right)\right)_i$$

证明结束。

根据引理 2 我们注意到，在 $\{-1,1\}^n$ 中获得了一个 KKT 点以后其实没有必要更新 t，因此使用条件 $x^* \in \{-1,1\}^n$ 和(15)作为终止准则。

相应的解决 PDSP 问题的算法[LS-TFW]如下

步骤 1：初始化 $k := 0$，$t_0 := 0$，$x_0 \in [0,e]$。设置

$$m\text{，}\ M\text{，}\ \mu := \min\left(0, -\lambda_{\max}\left(D\right)\right)\text{。}$$

步骤 2：从 $j := 0$ 到 $j := m-1$，按照(12)和(13)分别计算 x_k^*，α^*
并更新产生下一个迭代点

$$x_{j+1} = x_k + \alpha^*\left(x_j^* - x_j\right) \tag{19}$$

如果 $x^* \in \{0,1\}^n$ 并且有

$$\max_{i\in I^*}\left(\nabla_x H\left(x^*;\mu;t\right)_i\right) \le \min_{i\in J^*}\left(\nabla_x H\left(x^*;\mu;t\right)\right)_i$$

成立其中 $I^* \in \{i : x_i^* = 0\}$，$J^* \in \{i : x_i^* = 1\}$，则停止计算并返回 $x_k^* = x_{j+1}$，否则进入步骤 3。

步骤 3：如果 $k = M$，则停止计算并返回 $x^* = sign\left(x_m\right)$；否则更新 $k := k+1$，$t_k := \dfrac{k}{M+1}$，$x_0 := x_m$ 返回到步骤 2 继续计算。

3. 实例计算

在这一部分，我们会使用本文介绍的连续算法，在典型分支对称的框架下进行实验并与分支定界算法计算的最优解进行对比[14]，说明连续算法具有结果较优且不受结点数量限制的优越性。

初始设定 $M = 20$，$m = 0$，$x_0 = \left(\dfrac{p}{n}, \dfrac{p}{n}, \cdots, \dfrac{p}{n}\right)$，在 matlab 中进行算法 LS-TFW 的编程计算，实验以下两种典型的随机实例。

GEO：这个几何问题反映了在欧式空间中典型的选址问题，也可以参考 Erkut[15]，即这个 n 个工厂随机分布在一个 100 × 100 的矩阵中，d_{ij} 代表工厂 i 与工厂 j 之间的欧氏距离。

WGEO：加权的几何问题是为了说明当工厂间距离不同时的情况，与之前一样 n 个工厂随机分布在一个 100 × 100 的矩阵中，每个工厂都安排一个介于[5, ⋯ ,10]之间的权重 ω_i，距离 d_{ij} 选取的是工厂 i 与工厂 j 之间 $\omega_i\omega_j$ 倍的欧氏距离。

在以上所有的实例中，我们都设定 $d_{ii} = 0\ \left(i = 1, \cdots, n\right)$，其中在每个实例中，$p$ 都随机产生于[2, ⋯ ,n – 2]中。在每个类型中我们选取 10 个例子在一个 2.1 GHz 的电脑的 MATLAB7.8 软件上运行，前 5 个例子进行连续算法计算的结果和最优解比较。后 5 个例子，由于我们的最优解是由分支定界法计算受结点 n 数量的限制十分耗时间，我们就只列出连续算法的计算结果。

1) 在 GEO 类型中随机产生 10 组数据计算结果如下：

实验编号	大小 (n, p)	LS-TFW 算法结果	LS-TFW 运算时间(秒)	最优解
1	(10,4)	966.10	0.01	966.10
2	(20,8)	3844.64	0.02	3758.72
3	(30,20)	22822.04	0.02	22822.04
4	(40,19)	23682.68	0.02	23689.86
5	(50,30)	55074.87	0.02	55262.00
6	(100,74)	313807.41	0.0418	-
7	(200,101)	632164.07	0.0818	-
8	(300,208)	2495007.87	0.4069	-
9	(400,268)	4215851.28	0.7826	-
10	(500,236)	3570651.94	1.1543	-

2) 在 WGEO 类型中随机产生 10 组数据计算结果如下：

实验编号	大小 (n, p)	LS-TFW 算法结果	LS-TFW 运算时间(秒)	最优解
1	(10,6)	167387.81	3.43	167765.68
2	(20,15)	699125.36	0.10	699125.36
3	(30,23)	1744664.02	0.02	1744664.02
4	(40,19)	4384835.68	0.02	4384835.68
5	(50,37)	5232448.64	0.02	5232448.64
6	(100,74)	21831260.19	0.0307	-
7	(200,184)	105162118.77	0.0806	-
8	(300,183)	124969852.00	0.3783	-
9	(400,6)	219276.53	0.7642	-
10	(500,98)	53349707.46	1.0366	-

由上述实例可以看出本文介绍的连续算法可以得到很好的结果，当 n 比较小的时候几乎等于最优解的，而当 $n \geq 50$ 较大时，分支定界法计算就很耗时间，而连续算法不受结点数量的限制，计算迅速方便，表现良好，值得我们进一步深入继续研究。

4. 总结

在本文中，我们介绍了一种启发式的连续算法，并把连续算法应用到我们研究的离散的 p-扩散问题。一般解决这种问题采用的是离散的分支定界方法，由于计算机计算技术的限制，离散算法具有一定的局限性，在本文中，我们着重介绍了一种新的启发式拉格朗日算法，在 Frank-Wolfe 连续算法的基础上，通过控制一个参数使得原问题从拉格朗日函数向罚函数过渡迭代求解，建立了有效的迭代终止准则，并且证明了这种算法最终会收敛在一个 KKT 点，最后进行了大量的数值计算验证算法具有良好的表现，下一步，我们会继续研究这种连续算法的与其他连续算法的相比，算法结果的优越程度，以及进一步研究连续算法相比于离散算法较优的其他性质。

参考文献 (References)

[1] Horst, R. and Pardalos, P. M. (1975) Handbook of globa1 optimization. Kluwer Academic Publishers, Boston.

[2] Pardalos, P.M. and Rosen, J.B. (1987) Constrained global optimization: Algorithm and applications. Springer-Verlag, Berlin.

[3] Adams, W.P. and Sherali, H.D. (1986) A tight linearization and an algorithm for zero-one quadratic programming problems. *Management Science*, **10**, 1274-1290

[4] Krarup, J., Pisinger, D. and Plastria, F. (2002) Discrete location problems with push-pull objectives. *Discrete Applied Mathematics*, **123**, 363-378.

[5] Kortsarz, G. and Peleg, D. (1993) On choosing a dense subgraph. Proceedings of the 34th Annual IEEE Symposium on Foundations of Computer Science, Palo Alto, 3-5 November 1993, 692-701.

[6] Asahiro, Y., Iwama, K., Tamaki, H. and Tokuyama, T. (1996) Greedily finding a dense subgraph. Proceedings of the 5th Scandinavian Workshop on Algorithm Theory. *Lectures notes in Computer Science*, **1097**, 136-148.

[7] Hassin, R., Rubinstein, S. and Tamir, A. (1997) Approximation algorithms for maximum dispersion. *Operations Research Letters*, **21**, 133-137.

[8] Srivastav, A. and Wolf, K. (1998) Finding dense subgraph with semidefiinite programming. In: K. Jansen and J. Rolim, Eds., *Approximation Algorithms for Combinatorial Optimization*, Springer, Berlin, 181-191.

[9] Kincaid, R.K. (1992) Good solutions to discrete noxious location problems via metaheuristics. *Annals of Operations Research*, **40**, 265-281.

[10] Krarup, J., Pisinger, D. and Plastria, F. (2002) Discrete location problems with push-pull objectives. *Discrete Applied Mathematics*, **123**, 363-378.

[11] Billionnet, A. and Soutif, E. (2004) An exact method based on lagrangian decomposition for the 0-1 quadratic knapsack problem. *European Journal of Operational Research*, **157**, 565-575.

[12] Xia Yong and Guo, Z.F. (2011) Quadratic-type efficient upper bounds for the p-dispersion-sum problem.

[13] Xia Yong and Guo, Z.F. (2010) An efficient lagrangian smoothing heuristic for MAX-CUT. *Indian Journal of Pure & Applied Mathematics*, **41**, 683-700

[14] Pisinger, D. (2006) Upper bounds and exact algorithms for p-dispersion problems. *Computers & Operations Research*, **33**, 1380-1398.

[15] Erkut, E. (1990) The discrete p-dispersion problem. *European Journal of Operational Research*, **46**, 48-60.