

An Efficient Proximal Point Algorithm for Unweighted Max-Min Dispersion Problem*

Siqi Tao

Department of Mathematics, College of Sciences, Shanghai University, Shanghai, China

Email: mybaby@i.shu.edu.cn

How to cite this paper: Tao, S.Q. (2018) An Efficient Proximal Point Algorithm for Unweighted Max-Min Dispersion Problem. *Advances in Pure Mathematics*, 8, 400-407. <https://doi.org/10.4236/apm.2018.84022>

Received: March 12, 2018

Accepted: April 15, 2018

Published: April 18, 2018

Copyright © 2018 by author and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

In this paper, we first reformulate the max-min dispersion problem as a saddle-point problem. Specifically, we introduce an auxiliary problem whose optimum value gives an upper bound on that of the original problem. Then we propose the saddle-point problem to be solved by an adaptive custom proximal point algorithm. Numerical results show that the proposed algorithm is efficient.

Keywords

Maximum Weighted Dispersion Problem, Adaptive Custom Proximal Point Algorithm, NP-Hard

1. Introduction

Consider the following weighted max-min dispersion problem:

$$\max_{x \in \mathcal{X}} \left\{ f(x) := \min_{i=1, \dots, m} \omega_i \|x - x^i\|^2 \right\}, \quad (1)$$

where $\mathcal{X} = \left\{ y \in \mathbb{R}^n \mid (y_1^2, \dots, y_n^2, 1)^T \in \mathcal{K} \right\}$, \mathcal{K} is a convex cone, $x^1, \dots, x^m \in \mathbb{R}^n$ are m given point, $\omega_i > 0$ for $i = 1, \dots, m$ and $\|\cdot\|$ denotes the Euclidean norm. Let $v(P_{\mathcal{X}})$ denote the optimal value of the problem (1). The problem aims to find a point x in a closed set \mathcal{X} that is furthest from a given set of points x^1, \dots, x^m in \mathbb{R}^n in a weighted max-min sense. It has wide applications in spatial management, facility location, and pattern recognition (see [1] [2] [3] [4] and references therein). In the equal weight case, *i.e.*, $\omega_1 = \dots = \omega_m$, (1) has the geometric interpretation of finding the largest Euclidean sphere with center in P_{box} and enclosing no given point.

*The research was supported by National Natural Science Foundation of China under the grant 11571221.

Without loss of generality, we assume that $\nu(P_\chi) > 0$. The weighted max-min dispersion problem is known to be NP-hard in general, even in the case of equal weights and $\chi = [-1, 1]^n$ [5] or $\chi = \{x \mid \|x\| \leq 1\}$ [6]. We denote the two special cases by P_{box} and P_{ball} , which correspond to setting $\mathcal{K} = \{y \in \mathbb{R}^{n+1} \mid y_j \leq y_{n+1}, j = 1, \dots, n\}$ and $\mathcal{K} = \{y \in \mathbb{R}^{n+1} \mid y_1 + \dots + y_n \leq y_{n+1}\}$ respectively.

In the low-dimensional cases of $n \leq 3$ and χ being a polyhedral set, this problem is solvable in polynomial time [4] [7]. For $n > 4$, heuristic approaches have been proposed [1] [4].

In paper [5], they use an optimal solution of convex relaxations from semidefinite programming (SDP) and second order cone programming (SOCP) to construct an approximate solution of (1), and prove an approximation bound of $\frac{1 - O(\sqrt{\ln(m)\gamma^*})}{2}$, where γ^* depends on χ . When $\chi = \{-1, 1\}^n$ or $\chi = [-1, 1]^n$, $\gamma^* = O\left(\frac{1}{n}\right)$. This is the first nontrivial approximation bound for a convex relaxation of (1). Wang and Xia [6] then focus on the study of P_{ball} and show the approximation bound of their algorithm is $\frac{1 - O(\sqrt{\ln(m)/n})}{2}$ based on a linear programming relaxation.

In this paper, we focus on the equal weight max-min dispersion problem, which is called by “max-min dispersion problem” for simplicity. Firstly, we model the max-min dispersion problem as a saddle point problem, and then we adopt an adaptive custom proximal point algorithm to obtain a ε -approximation scheme¹.

The remainder of the paper is organized as follows. In Section 2, we reformulate max-min dispersion problem as a saddle point problem. In Section 3, we propose a new adaptive custom proximal point algorithm to approximately solve the saddle point problem and establish the convergence analysis. Section 4 presents some numerical comparisons between our proximal point algorithm and SDP-based algorithm. Conclusions are made in Section 5.

2. Saddle Point Model

Without loss of generality, we drop the weight parameters ω_j from the objective function, since all the ω_j s are equal. In the following of this paper, we consider the problem:

$$\max_{x \in \mathcal{X}} \left\{ f(x) := \min_{i=1, \dots, m} \|x - x^i\|^2 \right\}. \quad (2)$$

Note that, it has been proved that this problem is NP-hard in general [5] [6]. Denote Δ_m by the unit simplex in \mathbb{R}^m , that is, $\Delta_m = \{x \in \mathbb{R}^m \mid x \geq 0, e^T x = 1\}$ with e being the all one vector, then (2) is equivalent to the following saddle point problem:

¹We call $g(x)$ is the ε -approximation of $g(x^*)$ if $g(x) \geq g(x^*) - \varepsilon$.

$$\max_{x \in \mathcal{X}} \min_{y \in \Delta_m} \sum_{i=1}^m y_i \|x - x^i\|^2. \tag{3}$$

$$\begin{aligned} \phi(x, y) &= \sum_{i=1}^m y_i \|x - x^i\|^2 = \sum_{i=1}^m \left(y_i \|x\|^2 - 2y_i (x^i)^T x + y_i \|x^i\|^2 \right) \\ &= -2(Ay)^T x - 2y^T b + \sum_{i=1}^m y_i \|x\|^2 \\ &= -2(Ay)^T x - 2y^T b + \gamma \|x\|^2, \end{aligned}$$

where $A = [A_1, \dots, A_m] \in \mathbb{R}^{n \times m}$, $A_i = x^i$ and $b = \left(-\frac{1}{2} \|x^1\|^2, \dots, -\frac{1}{2} \|x^m\|^2 \right)^T$,

$\gamma = \sum_{i=1}^m y_i = 1$. $\phi(x, y)$ is convex for x and concave for y separately, although the saddle point model is neither convex nor concave.

Define $g(x) = \min_{y \in \Delta_m} \phi(x, y)$, and let x^*, y^* be the optimal saddle point of objective (3). Note that x^* is also necessarily a minimizer of $g(x)$ and $g(x^*) = \max_{x \in \mathcal{X}} \min_{y \in \Delta_m} \sum_{i=1}^m y_i \|x - x^i\|^2$. Now it suffices for us to find a point x such that $g(x) \geq g(x^*) - \varepsilon$, because such an x is necessarily a ε -approximate solution to (3).

However, $\phi(x, y)$ is not strongly concave with respect to y . Furthermore, define the regularized saddle point problem

$$\max_{x \in \mathcal{X}} \min_{y \in \Delta_m} \left\{ \phi_\lambda(x, y) := -2y^T A^T x - 2y^T b + \gamma \|x\|^2 - \lambda \|y\| \right\}, \tag{4}$$

So $\phi_\lambda(x, y)$ is λ -strongly concave on y and γ -strongly convex on x .

Denote the optimal solution of (4) by (x°, y°) . The relation between the optimal value of (3) and that of (4) can be characterized in the following lemma.

Lemma 1. $g(x^*) - g(x^\circ) \leq \varepsilon/2$ if $\lambda \leq \frac{\varepsilon}{2}$.

Proof. Denoting $\tilde{y} = \arg \min_{y \in \Delta_m} \phi(x^\circ, y)$, we have

$$\begin{aligned} g(x^\circ) &= \phi(x^\circ, \tilde{y}) = \phi_\lambda(x^\circ, \tilde{y}) - \lambda \|\tilde{y}\| \geq \phi_\lambda(x^\circ, y^\circ) - \lambda \|\tilde{y}\| \\ &\geq \phi_\lambda(x^*, y^\circ) - \lambda \|\tilde{y}\| = \phi(x^*, y^\circ) + \lambda \|y^\circ\| - \lambda \|\tilde{y}\| \\ &\geq \phi(x^*, y^*) + \lambda \|y^\circ\| - \lambda \|\tilde{y}\| = g(x^*) + \lambda \|y^\circ\| - \lambda \|\tilde{y}\| \\ &\geq g(x^*) - \lambda \|y^\circ\|. \end{aligned}$$

Since when $y_1 = 1, y_2 = \dots = y_m = 0$, $\|y\|_{\max} = 1$, and we then have

$$g(x^*) - g(x^\circ) \leq \lambda \|y\| \leq \frac{\varepsilon}{2}.$$

3. Adaptive Custom Proximal Point Algorithm

In this section, we adopt an adaptive custom proximal point (ACPP) algorithm to solve (4), which is quadratic and then can be approximately solved in a short time. From the optimal conditions of the problem and the convexity of related functions, the (4) can be solved by the followed by the variational inequality: for

x°, y°

$$\|y\| - \|y^\circ\| + \begin{pmatrix} x - x^\circ \\ y - y^\circ \end{pmatrix}^T \begin{pmatrix} -2y^{\circ T}A + x^\circ \\ -2A^T x^\circ - 2b \end{pmatrix} \geq 0$$

And we denote

$$u = \begin{pmatrix} x \\ y \end{pmatrix}, u^\circ = \begin{pmatrix} x^\circ \\ y^\circ \end{pmatrix}, F(u^\circ) = \begin{pmatrix} -2y^{\circ T}A + x^\circ \\ -2A^T x^\circ - 2b \end{pmatrix}, \Omega = \mathbb{R}^n \times \mathbb{R}^m,$$

then the variational inequality can be reduction to: find the solution $u^\circ \in \Omega$, satisfy:

$$\|y\| - \|y^\circ\| + (u - u^\circ)^T F(u^\circ) \geq 0, \tag{5}$$

It's easy to verify that F is monotonous, so (5) is monotonous, and then the solution set is not empty.

We denote

$$M = \begin{pmatrix} tI_n & A^T \\ \theta A & sI_m \end{pmatrix} = \begin{pmatrix} tI_n + (1-\theta)\frac{1}{s}A^T A & A^T \\ \theta A & sI_m \end{pmatrix} \begin{pmatrix} I_n & 0 \\ (\theta-1)\frac{1}{s}A & I_m \end{pmatrix} \tag{6}$$

$$= H\tilde{M}, \theta \in [-1, 1]$$

We give the details of the (ACPP) method as in **Algorithm 1**.

Algorithm 1. A1: ACPP algorithm for the unweighted max-min dispersion model.

Step 0: Matrix $A \in \mathbb{R}^{n \times m}$, vector $b \in \mathbb{R}^m$, parameters $\lambda, \gamma > 0$, $\theta \in [-1, 1]$, $t > 0$, $s > 0$, $ts > \frac{1}{4}(1+\theta)^2 \lambda_{\max}(A^T A)$.

Step 1 Solve the variational inequality

$$\|y\| - \|y^\circ\| + (u - \tilde{u}^k)^T \{F(\tilde{u}^k) + M(\tilde{u}^k - u^k)\} \geq 0 \tag{7}$$

we can get a predicted point \tilde{u}^k .

correct u^{k+1} :

$$u^{k+1} = u^k - \gamma \alpha_k \tilde{M}(u^k - \tilde{u}^k) \tag{8}$$

Step 2

$$\tilde{y}^k = \arg \min_{y \in \mathbb{R}^m} \lambda \|y\| + \frac{t}{2} \left\| y - \left(y^k + \frac{1}{t} A^T y^k - 2b y^k \right) \right\|^2 \tag{9}$$

$$x^k = x^k - \frac{1}{s + \lambda} (A(1 + \theta) \tilde{y}^k - \theta x^k) + y^k \tag{10}$$

Step 3

$$\begin{pmatrix} y^{k+1} \\ x^{k+1} \end{pmatrix} = \begin{pmatrix} y^k \\ x^k \end{pmatrix} - \gamma \alpha_k \begin{pmatrix} I_n & \\ (\theta-1)\frac{1}{s}A & I_m \end{pmatrix} \begin{pmatrix} y^k - \tilde{y}^k \\ x^k - \tilde{x}^k \end{pmatrix} \tag{11}$$

where

$$\alpha_k = \frac{(u^k - \tilde{u}^k)^T M(u^k - \tilde{u}^k)}{\|\tilde{M}(u^k - \tilde{u}^k)\|_H^2} \tag{12}$$

Step 4 if $\frac{\|Ax^k - b\|_2}{\|b\|_2} \leq \varepsilon$ end; else for $k = k + 1$, do Step 2

In mathematics, the arguments of the minimum (abbreviated arg min or arg-min) are the points of the domain of some function at which the function values are minimized.

Convergence Analysis

We present a convergence theorem for A1 in this section. In order to proof our theorem, we now give some lemmas. The following lemmas 2-4 are standard results in [8] [9] [10].

Lemma 2. For H and M in (6), assume $s > 0, t > 0$, then the follow inequation is establish:

$$ts > \frac{1}{4}(1+\theta)^2 \lambda_{\max}(A^T A). \quad (13)$$

where H and $\frac{1}{2}(M + M^T)$ is positive definite matrix.

Lemma 3. \tilde{u}^k is the solution of (7), and M is defined in (6), then for $\forall u^\circ$, we have

$$(u^k - u^\circ)^T M (u^k - u^\circ) \geq (u^k - \tilde{u}^k)^T M (u^k - \tilde{u}^k)$$

Lemma 4. For \tilde{M} and H in (6), there exist a constant $c_0 > 0$, can make $\{u^k\}$ in (8) satisfy:

$$\|u^{k+1} - u^\circ\|_H^2 \leq \|u^k - u^\circ\|_H^2 - \gamma(2-\gamma)\alpha_k^\circ c_0 \|u^k - \tilde{u}^k\|^2$$

Now we can give the theorem of the ACPP algorithm.

Theorem 1. The ACPP algorithm is a shrinkage algorithm of the saddle point problem (4), and the sequence $\{u^k = (x^k, y^k)\}$ generated by the algorithm convergence to a solution of (4).

Proof. For $M \in R^{n \times n}$, there exist a constant c_0 , we have $d^T M d \geq c_0 \|d\|^2, \forall d \in R^n$, when the inequation is hold, the α_k° has a lower bound:

$$\alpha_k^\circ = \frac{c_0 \|u^k - \tilde{u}^k\|^2}{\|\tilde{M}(u^k - \tilde{u}^k)\|_H^2} \geq \frac{c_0}{\|\tilde{M}^T M\}}.$$

On the basis of lemma 4, we have

$$\|u^{k+1} - u^\circ\|_H^2 \leq \|u^k - u^\circ\|_H^2 - \gamma(2-\gamma) \frac{c_0^2}{\|\tilde{M}^T M\|} \|u^k - \tilde{u}^k\|^2$$

when $\gamma = 1$, ACPP algorithm is a H-norm shrinkage algorithm of the saddle point problem (4).

4. Numerical Results

In this section, we do some simple numerical comparisons. All the Numerical texts are implemented in Matlab R2014a and run on a laptop with 2.30 GHz processor and 4 GB RAM. We are now ready to apply **Algorithm 1** to our model

(4), which is shown in detail in **Algorithm 2**.

We present the numerical comparison between our ACPP algorithm and SDP-based algorithm proposed in [6] for solving P_{ball} ($\chi = \{x \mid \|x\| \leq 1\}$). We note that when the weighted $\omega_i = 1$ in [6], the two algorithm can comparable. We do numerical experiments on 24 random instances of dimension $n = 5$, where the number of input point m varies from 6 to 30. All the input points x^i ($i = 1, \dots, m$) with $m = 6, \dots, 30$ orderly form an $n \times 450$ matrix. We randomly generate this matrix using the following Matlab scripts:

$$\text{Rand}('state', 0); X = 2 * \text{rand}(n, 450) - 1;$$

where the $\text{rand}()$ is a random function that produces a random number between 0 and 1. We set $\varepsilon = 10^{-3}$, $\lambda = \varepsilon/2$, and report the numerical results in **Table 1**.

Algorithm 2. A2: ACPP algorithm for max-min dispersion problem.

Input: m point x^1, \dots, x^m and error constant $\varepsilon > 0$

Output: $x^{(T)}$ is an $1 - O(\varepsilon)$ approximation to model (4)

1: $A \leftarrow \left[(x^1)^T, \dots, (x^m)^T \right]$.

2: $\lambda = \frac{\varepsilon}{2}, \gamma = 1$, and $\frac{\|Ax^k + b\|_2}{\|b\|_2} < 10^{-5}, \theta = -0.25, s = 90, t = \frac{(1 + \theta)^2}{s\mu} \lambda_{\max}(A^T A), \mu \in (0, 4)$

3: $x^{(T)}, y^{(T)} \leftarrow \text{Al}(A, n, \lambda, \gamma, \theta, t, s)$

4: **return** $x^{(T)}$

Table 1. Numerical results for $n = 5, m = 6$ to 30.

m	cvx_opt	The SDP-based algorithm			2*our algorithm
		V_{\max}	V_{\min}	V_{ave}	
6	2.74	2.06	1.25	1.75	2.01
7	2.50	1.77	1.06	1.33	2.19
8	1.80	1.49	0.74	1.07	1.68
9	2.45	1.50	0.72	1.06	2.35
10	2.31	1.61	0.66	1.18	2.12
11	2.22	1.85	0.76	1.27	1.98
12	2.21	1.58	0.76	1.07	1.92
13	1.74	1.25	0.64	0.91	1.72
14	1.81	1.43	0.56	1.02	1.58
15	2.19	1.41	0.72	1.00	1.90
16	1.89	1.15	0.62	0.93	1.88
17	2.13	1.16	0.60	1.05	1.93
18	1.93	1.24	0.50	0.88	1.87
19	1.93	1.22	0.61	0.97	1.91
20	2.51	1.46	0.72	1.16	1.98
21	2.07	1.37	0.65	0.97	2.01

Continued

22	2.20	1.08	0.55	0.79	1.85
23	2.13	0.99	0.55	0.81	1.95
24	1.85	0.98	0.50	0.55	1.80
25	1.92	1.28	0.70	0.98	1.91
26	1.82	0.86	0.49	0.68	1.45
27	1.88	1.05	0.54	0.73	1.76
28	1.85	1.45	0.56	0.87	1.45
29	2.39	1.25	0.44	1.02	2.30
30	1.82	1.14	0.39	0.84	1.77

The columns cvx_opt present optimal objection function values of the 20 instance of P_{ball} [6]. The next two columns present the statistical results over the 10 runs of the algorithm proposed in [6] and our ACPP algorithm, respectively. The subcolumns v_{\max}, v_{\min} and v_{ave} give the best, the worst and the average objective function values found among 10 tests, respectively. The results show that compared with the SDP-based algorithm our algorithm is competitive in most cases.

From the table, we can see that the solution of our algorithm is very close to the exact solution of the second column, which is better than the SDP algorithm.

5. Conclusion

In this paper, we reformulate the max-min dispersion problem as a saddle point problem and then adopt an adaptive custom proximal point algorithm to obtain an approximation scheme. It can be proved that the proposed algorithm produces a ε -approximation solution to the max-min dispersion problem with equal weight. Numerical results show that the proposed algorithm is efficient.

References

- [1] Dasarthy, B. and White, L.J. (1980) A Maximin Location Problem. *Operations Research*, **28**, 1385-1401. <https://doi.org/10.1287/opre.28.6.1385>
- [2] Johbson, M.E., Moore, L.M. and Ylvisaker, D. (1990) Maximin Distance Designs. *Journal of Statistical Planning and Inference*, **26**, 131-148. [https://doi.org/10.1016/0378-3758\(90\)90122-B](https://doi.org/10.1016/0378-3758(90)90122-B)
- [3] Schaback, R. (1995) Multivariate Interpolation and Approximation by Translates of A Basis Function. World Scientific, Singapore, 491-514.
- [4] White, D.J. (1996) A Heuristic Approach to a Weighted Maxmin Disperation Problem. *IMA Journal of Management Mathematics*, **7**, 219-231. <https://doi.org/10.1093/imaman/7.3.219>
- [5] Haines, S., Loepky, J., Tseng, P. and Wang, X. (2013) Convex Relaxations of the Weighted Maxmin Dispersion Problem. *SIAM Journal on Optimization*, **23**, 2264-2294. <https://doi.org/10.1137/120888880>
- [6] Wang, S. and Xia, Y. (2016) On the Ball-Constrained Weighted Maximin Dispersion Problem. *SIAM Journal on Optimization*, **26**, 1565-1588.

-
- <https://doi.org/10.1137/15M1047167>
- [7] Ravi, S.S., Rosenkrantz, D.J. and Tayi, G.K. (1994) Heuristic and Special Case Algorithms for Dispersion Problems. *Operations Research*, **42**, 299-310.
<https://doi.org/10.1287/opre.42.2.299>
- [8] He, B. and Yuan, X. (2010) A Contraction Method with Implementable Proximal Regularization for Linearly Constrained Convex Programming. *Optimization On-line*, 1-14.
- [9] He, B., Fu, X. and Jiang, Z. (2009) A Proximal Point Algorithm Using a Linear Proximal Term. *Journal of Optimization Theory and Applications*, **141**, 209-239.
<https://doi.org/10.1007/s10957-008-9493-0>
- [10] Martinet, B. (1970) Regularisation d'equations variationnelles par approximations succesives. *Rev. Francaise d'Inform. Recherche Operation*, **4**, 154-159.