A Novel Separable Model and Decomposition Method for Sensor Locational Decision Problem

Linfeng Yang, Jie Li, Jin Ye, and Zhigang Zhao

1 School of Computer Electronics and Information, Guangxi University, Nanning, Guangxi 530004, China
2 Guangxi Experiment Center of Information Science, Guilin University of Electronic Technology, Guilin, Guangxi 541004, China
3 Department of Computer and Electronics Information Engineering, Guangxi Vocational&Technical College, Nanning, Guangxi 530226, China

Correspondence should be addressed to Jin Ye; yejin@gxu.edu.cn

Received 4 September 2013; Revised 17 January 2014; Accepted 10 February 2014; Published 23 March 2014

1. Introduction

Sensor locational decision problem covering a line (SLDPCL) is a problem to find a subset covering a given length segment at minimum cost with giving a set of sensors. And these sensors have variable radii with costs depending on their radii and fixed costs [1]. This problem was motivated by the following application. There is a river in a city, over which we need to track possible activities of objects or people. For this purpose, we need to locate a set of sensors so that every point on the river is under supervision by at least one sensor. It is assumed that (i) the river can be modeled as a tree network consisting of line segments and (ii) each sensor has a field of view defined by a radius. It is reasonable to view the river as piecewise strips of finite lengths. Then we may focus on the problem for a strip. The above problem for a strip can be viewed as an SLDPCL. Although the problem is easily stated, the actual locational decision is complicated due to some additional factors. Coverage depends not only on the river topology, sensor type, and power, but also on several parameters such as width of river and obstacles over it, potentially forbidden areas where sensors may not be located, and other characteristics associated with the physical environment, for example, the atmospheric and water conditions. Readers can further refer to [2] for further details on this scenario.

In this paper, a new separable mixed integer nonlinear programming (MINLP) formulation for the SLDPCL problem by decomposing the multivariate function into several univariate functions is presented. The outer approximation of a decomposed equivalent is tighter than the direct outer approximation of the original multivariate problem, so a tighter separable outer approximation methodology is proposed to improve the existing outer approximation method (OAM). Then, an improved OAM is proposed to solve the separable formulation for SLDPCL problem. This novel method consists of solving an alternating sequence of a nonlinear programming (NLP) problem and a mixed integer linear programming (MILP) problem. In order to solve the NLP problems efficiently, a new quickly interior point algorithm was presented based on the techniques of optimal centering parameter (OCP) and multiple centrality correction (MCC). At last, the novel OAM integrating improved interior point method (IPM) is presented to solve
this separable model for SLDPCL problem. Ten to 20000 sensors instances have been solved by the proposed method, and the results show that the method is scalable for large-scale SLDPCL problem. The main contributions of this paper are as follows.

(i) By simply introducing an auxiliary variable, we move the nonlinear function from objective function to constraints for the classic MINLP formulation of SLDPCL problem. After letting equivalent several univariate functions substitute for a multivariate function, we obtain a new separated formulation for SLDPCL problem.

(ii) An improved OAM is presented to solve the SLDPCL problem based on the aforementioned separated formulation. The proposed method can obtain tighter MILP relaxation than the traditional OAM does. Solving infeasible subproblem is not needed for the improved method, because all feasible solutions of the MILP relaxations are feasible for the corresponding SLDPCL problem as well.

(iii) A new quickly interior point algorithm was presented for solving NLP subproblems based on the techniques of OCP and MCC. Named as I-IPM, the proposed method involves integrating equilibrium distance-quality function (ED-QF) to establish a mathematical model for evaluating centering parameter. And the approximate expression of this model, which can be solved with fewer computations than the original one, was proposed using the linearization technique. After solving the approximate model with the line search technique, the optimal centering parameters can be obtained for the proposed method to own longer step length and less number of iterations than other IPMs. The aim of using MCC is twofold: (a) to increase the step length in both primal and dual spaces and (b) to offer better convergence performances.

The rest of this paper is organized as follows. Section 2 introduces some related works in coverage of sensor networks, especially in SLDPCL. Section 3 describes the classic MINLP formulation of SLDPCL, and we present a separable formulation for SLDPCL based on the classic one in this section as well. In Section 4, we present the improved OAM to solve the separable formulation, which consists of solving an alternating sequence of NLP and MILP. In Section 5, we present an improved IPM to solve those NLP subproblems effectively. Simulations are conducted and the results are shown in Section 6. Finally, we conclude this paper and point out the future work in Section 7.

2. Related Works

There are many applications of SLDPCL. In [6], with the assumptions that (i) a set of potential resources; (ii) a fixed activation cost for each resource; (iii) the congestion heavily influences the cost of a resource, a single-commodity network flow problem can be formulated and solved by using SLDPCL model. The effect of introducing additional stops in the existing railway network can be formulated as an SLDPCL model as well. This problem is comprised of covering a set of points in the plane by sensors. And the centers of these sensors have to lie on a set of line segments that represents the railway [7]. If only one period is considered, the unit commitment problem in power system also can be formulated as an SLDPCL model [8].

SLDPCL is a special case of coverage in sensor networks. Coverage, which reflects how well a sensor field is monitored, is one of the most important performance metrics to measure sensor networks. Simply said, sensor coverage models are abstract models trying to quantify how well sensors can sense physical phenomena at some location, or, in other words, how well sensors can cover such locations. In sensor networks, coverage problems can arise in all network stages and can be formulated in various ways with different scenarios, assumptions, and objectives [9]. There are two important problems relating to coverage. The first one is to formulate mathematical model. The other one is to solve the large-scale coverage problem models efficiently.

2.1. Modeling Coverage Problems. Sensor coverage models measure the sensing capability and quality by capturing the geometric relation between a space point and sensors. Modeling coverage problems fall into the field of geometric covering, in which some points or some areas are required to be covered by geometric objects [10]. For example, Chvátal [11] introduces the art gallery problem, in which cameras are to be placed to watch every wall of an art gallery room. The room is assumed to be a polygon with $n$ sides and $h$ holes, and the cameras are assumed to have a viewpoint of $360^\circ$ and rotate at an infinite speed. It is proved that at most $[(n + h)/3]$ cameras are sufficient. In [12], Huang and Tseng study the problem of deciding whether an area is sufficiently covered, in the sense that every point in the area is covered by at least $k$ sensors. They prove that as long as the perimeters of the sensors are sufficiently covered, the whole area is sufficiently covered. In [13], Wang et al. study the relationship between coverage and connectivity. They prove that if a region is $k$-covered then the sensor network is $k$-connected as long as the communication range is at least twice of the sensing range. In [1], the problem of coverage in a line segment is formulated as mixed integer nonlinear programming (MINLP) with quadratic objective function and linear constraints. Reference [14] shows that solving the aforementioned coverage problem with MINLP is NP-hard problem.

2.2. Solving Methods. The coverage problem often can be formulated as MINLP. As pointed in [14], this MINLP is NP-hard problem, which is rather difficult to solve efficiently for large-scale instances. For the problem in which a set of $m$ fixed locations for radars and a set of $n$ locations for the points are given, all located along a fixed line, [15] gives an $O((n + m)^3)$ dynamic programming approach. To lower the time complexity, [15] gives a linear-time 4-approximation algorithm as well. Reference [16] shows that the line segment coverage problem is polynomial time solvable by reducing it to an integer linear program with totally unimodular matrix.
All these methods are based on heuristics or approximations, which cannot guarantee obtaining the global optimal solutions. In [17], Li et al. consider the problem of covering a line segment with wireless sensors of adjustable ranges. They present a polynomial time exact algorithm for the discrete case. However, for the continuous case, where the cost of each sensor is given by a function \( f(r) = r^a \), the polynomial-time algorithm presented by them is an approximated one. Reference [18] presents an \( O(n^3) \) dynamic programming algorithm for the radar placement and power assignment problem, where \( n \) is the number of crucial points to be monitored. However, the proposed method often suffers from the curse of dimensionality just the same as classic dynamic programming does.

SLDPCL can be formulated as MINLP too, but it has some characteristics: (i) the need to cover a single line segment and (ii) a fixed activation cost for each sensor. According to these characteristics, searching more effective algorithm to solve SLDPCL is necessary. In [1], the authors provide an exact solution algorithm, based on a Lagrangian relaxation and a subgradient algorithm, to find a lower bound for SLDPCL. But the feasible solution only can be obtained by using a heuristic method. In [19], an \( \tilde{O}(n \log n) \) time 12-factor approximation algorithm is proposed for the problem of covering a set of line segments with minimum number of sensors. However, the algorithm needs that the line segments are axis-parallel. In [3], the bound of SLDPCL model is improved by using the convex envelope of the single block of the objective function. Then the MINLP model of SLDPCL is reformulated as mixed integer second-order cone programming (MISOCP), which can be solved by general-purpose mixed integer programming solvers. However, [20] points out that directly passing MISOCP to the solvers is less competitive than using piecewise linear approximations.

In the last 40 years, some algorithms have been proposed for solving convex MINLP to optimality [21]. In the early 70s, Geoffrion generalized Benders decomposition (BD) to make an exact algorithm for convex MINLP [21]. And in the 80s, Duran and Grossmann introduced the OAM [22]. Both BD method and OAM are decomposition strategies, which can solve MINLP efficiently. However, BD method and its variants use the dual information for outer approximation, while the OAM method and its variants are based on the use of the optimal primal information. So far, BD method has been used to solve the location problem successfully [23], which decomposes the location problem into a master problem that is an integer optimization problem and an NLP subproblem. However, to the authors’ knowledge, up to now, OAM method has not been used for solving location problem.

Solving the NLP subproblems efficiently plays a pivotal role in solving MINLP model for SLDPCL problem by using OAM method. IPMs [5] are currently considered the most powerful algorithms for large-scale NLPS. Some of the key ideas, such as primal-dual steps and center parameter, carry over directly from the LP case [4], but several important new challenges arise [24]. These include the strategy for updating the barrier parameter and the need to ensure progress toward the solution according to the structure of application problems.

3. Separable MINLP Formulation for SLDPCL

The traditional MINLP formulation of SLDPCL is reported in [1] by Agnetis et al. In this section, we give a separable MINLP for SLDPCL, which can be solved more efficiently by using decomposition strategies, especially by using OAM.

Suppose that there are \( q \) available discs (sensors), and they are represented by the set \( Q = \{1, 2, \ldots, q\} \). For all \( i \in Q \), at most one copy of disc \( i \) may be used for covering the line segment and the power level is allowed as \( 0 \leq r_i \leq \bar{r}_i \), where \( r_i \) denotes the disc coverage diameter and \( \bar{r}_i \) denotes the cover limit for sensor \( i \). The assumption may appear restrictive for real applications, but note that usage of multiple copies of the same disc type may be modeled by including in in \( Q \) a suitable number of items with the same characteristics.

For any selected disc \( i \in Q \), its contribution in the total cost function is

\[
c_{f,i}(r_i) = \begin{cases} 
0 & \text{if } r_i = 0 \\
\alpha_i + g_i(r_i) & \text{if } r_i > 0,
\end{cases}
\]

where the setup cost \( \alpha_i > 0 \) and operational cost \( g_i(r_i) = \beta_i r_i + \gamma_i r_i^2 \). \( \beta_i \geq 0, \gamma_i > 0 \).

The objective function of the SLDPCL problem, representing the total cost to be minimized, has the form

\[
\min \sum_{i \in Q} (\alpha_i u_i + \beta_i r_i + \gamma_i (r_i)^2),
\]

where \( u_i \) indicate selected sensors \( i \in Q \):

\[
u_i = \begin{cases} 
1 & \text{sensor } i \text{ has been selected} \\
0 & \text{sensor } i \text{ has not been selected}.
\end{cases}
\]

The constraints of the SLDPCL problem are as follows.

1. **Covering constraint**

\[
\sum_{i \in Q} r_i - r_{\text{all}} = 0,
\]

where \( r_{\text{all}} \) denotes the length of line segment needing to be covered.

2. **Sensor coverage distance limits**

\[
0 \leq r_i \leq u_i \bar{r}_i.
\]

Constraints (5) force \( r_i \) to be zero when the corresponding sensor \( i \) is not selected (and therefore the corresponding cost contribution is zero). Constraint (4) is the coverage constraint that assures that the whole line segment is covered.

Equations (4)-(5) are linear constraints. These constraints can be abbreviated as

\[
A_u u + A_r r \leq a_{ux}
\]

for the sake of convenience, where \( u = (u_i), r = (r_i), i = 1, \ldots, q, A_u, A_r \) are the coefficient matrices and \( a_{ux} \) is the constant vector.
Consequently, the SLDPCL problem is conveniently formulated as the following MINLP:

\[
\begin{align*}
\min & \sum_{i \in Q} (\alpha_i u_i + \beta_i r_i) + g(r) \\
\text{s.t.} & A_u u + A_r r \leq a_{ux} \\
& u \in \{0, 1\}^q,
\end{align*}
\]

where \( g(r) = \sum_{i=1}^q y_i(r_i)^2 \). By introducing an auxiliary variable \( \eta \), the SLDPCL problem (6) is equivalent to the form denoted as SMIP:

\[
\begin{align*}
\min & \sum_{i \in Q} (\alpha_i u_i + \beta_i r_i) + \eta \\
\text{s.t.} & g(r) \leq \eta \\
& A_u u + A_r r \leq a_{ux} \\
& u \in \{0, 1\}^q.
\end{align*}
\]

Problem (7) is special MINLP with separable constraint \( g(r) \leq \eta \). In the following sections, we will discuss how to solve this separable model for SLDPCL problem more efficiently. Model (7) is convex MINLP because of the following.

(a) \( g(r) \leq \eta \) is convex constraint due to \( \alpha_i > 0 \) for \( \forall i \in Q \).

(b) The other constraints and the object function all are linear.

4. OAM for Separable Mixed Integer Programming of SLDPCL

The OAM, a very successful approach to solving convex MINLP, was first developed by Duran and Grossmann [22] and then subsequently extended to a general mixed integer convex programming, nonconvex MINLP, and separable MINLP [25]. The OAM consists of solving an alternating sequence of a primal problem and MILP. A sequence of valid nondecreasing lower bounds and nonincreasing upper bounds is generated by the OAM which converges in a finite number of iterations.

In this section, we consider the solution of the SMIP formulation (7) for SLDPCL problem. The function \( g \) is convex and separable. By separable, we mean that the function \( g \) can be rewritten as a sum of some convex univariate functions [25]; that is, \( g(r) = \sum_{i \in Q} g_i(r_i) \), where \( g_i = y_i(r_i)^2 \) is convex univariate function. Now, we describe OAM algorithms [25] to solve the SLDPCL model SMIP.

The OAM solves SMIP formulation (7) by alternating finitely between an NLP subproblem and an MILP relaxed master problem. And this master problem is obtained by replacing the nonlinear constraints with their linear outer approximations taken in a set of points \( \Omega = \{r^1, \ldots, r^n\} \). The corresponding outer approximation master MILP of SMIP can be formed as

\[
\begin{align*}
\min & \sum_{i \in Q} (\alpha_i u_i + \beta_i r_i) + \eta \\
\text{MILP} (\Omega) & g(\bar{r}) + \nabla g(\bar{r})^T (r - \bar{r}) \leq \eta, \\
\text{s.t.} & A_u u + A_r r \leq a_{ux} \\
& u \in \{0, 1\}^q.
\end{align*}
\]

The integer vector \( u^{k+1} \) obtained by MILP(\( \Omega \)) is feasible for SMIP (7). The main theoretical justification of OAM is that it has the same optimal value as SMIP if MILP (\( \Omega \)) contains a suitable set of linearization points. Next, the NLP subproblem for fixed value of the integer decision vector \( u^{k+1} \) is defined as

\[
\begin{align*}
\min & \sum_{i \in Q} (\alpha_i u_i + \beta_i r_i) + \eta \\
\text{NLP} (u^{k+1}) & g(r) \leq \eta, \\
\text{s.t.} & A_u u^{k+1} + A_r r \leq a_{ux}.
\end{align*}
\]

We are now in a position to give the outer approximation algorithm in full detail [25].

Algorithm OAM

Initialization. Set the superbound \( z_{up} = 10^{10} \) and lower bound \( z_{low} = 10^{-10} \); set the tolerance \( \varepsilon \).

Step 1. Solve the continuous relaxation of (7) with IPM; the optimal solution is \( y^0 = (u^0, r^0, \eta^0) \) and \( k = 1 \).

Do while \( (z_{up} - z_{low})/(z_{up} + z_{low}) > \varepsilon \)

Step 2. Construct MILP (8), and solve it with branch-cut method. The optimal solution of (8) is \( (u^{k+1}, \bar{r}, \bar{\eta}) \), and optimal value is \( z_{low} \).

Step 3. Construct NLP (9), and solve it with IPM. The optimal solution of (9) is \( (r^{k+1}, \eta^{k+1}) \). If \( \sum_{i \in Q} (\alpha_i u_i^{k+1} + \beta_i r_i^{k+1}) + \eta^{k+1} < z_{up} \), let \( z_{up} = \sum_{i \in Q} (\alpha_i u_i^{k+1} + \beta_i r_i^{k+1}) + \eta^{k+1} \), and we get the current solution and object value are \( (r^{k+1}, u^{k+1}) \) and \( z_{up} \) separately.

Step 4. \( k = k + 1 \); update \( \Omega = \Omega \cup \{r^{k+1}\} \).

End do

Step 5. Output optimal object value \( z_{up} \) and the optimal solution \( (r^{k+1}, u^{k+1}) \).
The convex function $g$ can be outer approximated by subgradient inequalities through a single-step procedure that yields MILP ($\Omega$). Furthermore, it can be several-step approximated by constructing subgradient inequalities for some univariate functions $g_i$, $i = 1, \ldots, q$.

First, based on the analysis above, in (7), we introduce an auxiliary variable $v_i$ for each univariate function $g_i$, and the SLDPCL problem (7) can be reformulated as the separated model of the form

$$\min \sum_{i \in Q} (\alpha_i u_i + \beta_i r_i) + \eta$$

$$\sum_{i=1}^q v_i \leq \eta$$

$$g_i (r_i) = \gamma_i (r_i)^2 \leq v_i, \quad i = 1, \ldots, q$$

$$A_u u + A_r r \leq a_{nx}$$

$$u \in [0, 1]^q.$$  \hfill (10)

Clearly, if $(z, v, z) = (u, r, \eta), v = (v_i)$, is an optimal solution of (10), then $z$ is optimal for (7). According to [25], we have the following theorem.

**Theorem 1.** Give points $r^i, s = 1, \ldots, n_0$. Let $S_1 = \{\Gamma, r^i_1 | \Gamma \geq g(r^i) + \nabla g(r^i)(r^i - \hat{r}^i), s = 1, \ldots, n_0\}$ and $S_2 = \{\Gamma, r^i | \Gamma \geq \sum_{j=1}^q v_j, v_j \geq g_j(r^i) + \nabla g_j(r^i)(r^i - \hat{r}^i), s = 1, \ldots, n_0, j \in Q\}$. Then, $S_1 \supseteq S_2$.

It is easy to observe from Theorem 1 that the linear outer approximation MILP of (10) may be tighter than the one of (7). More precisely, Theorem 1 shows that, to obtain a linear relaxation of (7) of the same strength as a linear approximation of (10), one needs exponentially many linearization points. Thus, in general, we can expect the extended formulation to perform much better than the initial one in an outer approximation algorithm. Then, a tighter outer approximation master MILP for SMIP is given by

$$\min \sum_{i \in Q} (\alpha_i u_i + \beta_i r_i) + \eta$$

$$T\text{-MILP (}\Omega)$$

$$\sum_{i=1}^q v_i \leq \eta$$

$$g_i (r_i) + \frac{\partial g_i}{\partial r_i} (r^i - \hat{r}^i) \leq v_i, \quad i = 1, \ldots, q, \quad s = 1, \ldots, n_1$$

$$A_u u + A_r r \leq a_{nx}, \quad u \in [0, 1]^q.$$  \hfill (11)

We want to note that the T-MILP (11) is a lower linear approximation for MINLP model (10) of SLDPCL problem. We cannot guarantee that the solutions of T-MILP are feasible for MINLP (10). In [25], an inner approximation, which is MILP, is solved to obtain feasible solutions of (10). In our method, the T-MILP model has included all the constraints of SLDPCL problem. So the solutions of (11) are feasible for SLDPCL problem.

In addition, according to the description of Algorithm OAM in this section, we know that there are two NLPs, (9) in Step 3 and the relaxation of (7) in Step 1, which need to be solved efficiently. IPM [5] is a very appealing approach to large-scale NLP. We present a new quickly interior point algorithm for solving these two NLPs based on the techniques of OCP and MCC in the following section.

### 5. Improved IPM for NLP

#### 5.1. Primal-Dual IPM

IPMs became quite popular after 1984, when N. Karmarkar announced a fast polynomial-time interior method for linear programming [5]. In this subsection, we introduce classic primal-dual IPM and analyze the difficulty of setting centering parameter in IPM. Then, our improvement in setting centering parameter is proposed in the following subsections.

Without losing the generalization, we consider the NLP writing in a formal style as follows

$$\begin{align*}
\min & \quad f(x) \\
\text{s.t.} & \quad h(x) = 0 \\
& \quad g(x) \leq \bar{g}.
\end{align*}$$  \hfill (12)

By introducing slack variables in the inequality of constraints and introducing logarithmic barrier penalty terms in the objective function, we obtain the following logarithmic barrier problem for (12):

$$\begin{align*}
\min & \quad \varphi = f(x) - \mu \sum_{j=1}^r \ln l_j - \mu \sum_{j=1}^r \ln u_j \\
\text{s.t.} & \quad h(x) = 0 \\
& \quad g(x) - l - \bar{g} = 0 \\
& \quad g(x) + u - \bar{g} = 0.
\end{align*}$$

The Lagrangian function of (13) is

$$L(X, \mu) = f(x) - y^T h(x) - z^T \left(g(x) - l - \bar{g}\right) - \mu \sum_{j=1}^r \ln l_j - w^T \left(g(x) + u - \bar{g}\right) - \mu \sum_{j=1}^r \ln u_j.$$  \hfill (14)

According to the gradients of the Lagrangian function with respect to variables $x$, multipliers $y, z, w$, and slack variables $l, u$, we obtain the perturbing KKT equation of (12),

$$K(X, \mu) = \begin{bmatrix}
L_x(X); L_y(X); L_z(X) \\
L_w(X); L_l(X, \mu); L_u(X, \mu)
\end{bmatrix} = 0$$  \hfill (15)

$$L_x(X) = -\nabla f(x) + \nabla h(x) y + \nabla g(x)(z + w)$$  \hfill (16)

$$L_y(X) = h(x)$$
\[
L_z(X) = g(x) - 1 - \bar{g} \\
L_w(X) = g(x) + u - \bar{g} \\
L_{1}(X, \mu) = LZe - \mu_e \\
L_{u}(X, \mu) = UWe + \mu_e,
\]

where \(X = [x; y; z; w; l; u], 0 \leq z, l, u, 0 \geq w, Z = \text{Diag}(z), W = \text{Diag}(w), L = \text{Diag}(l), U = \text{Diag}(u), e = [1, 1, ..., 1]^T,\) and \(X\) is the interior point of (12) if \(z, l, u > 0\) and \(w < 0\).

Let \(g = \bar{T}z - u'w, \bar{g} = g/2r, \sigma \in [0, 1),\) and let \(\mu = \sigma \bar{g}\). Applying Newton's method to (15) gives the primal-dual system

\[
\begin{bmatrix}
H(X) & Vh(x) & Vg(x) & \nabla g(x) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \Delta x \\
Vh(x)^T & 0 & 0 & 0 & 0 & \Delta y \\
Vg(x)^T & 0 & 0 & 0 & -I & 0 & \Delta z \\
0 & 0 & L & 0 & 0 & 0 & 0 & 0 & 0 & \Delta l \\
0 & 0 & 0 & U & 0 & W & \Delta u
\end{bmatrix}
= -K(X, \sigma \bar{g}) .
\]

Denote (19) by

\[
M(\cdot) \Delta X = -K(X, \sigma \bar{g}) ,
\]

where \(H(X) = -\nabla^2 f(x) + \nabla^2 h(x)y + \nabla^2 g(x)(z + w)\).

After determining the primal-dual direction \(\Delta X_{pd}\) from (20), we compute primal and dual step lengths as follows:

\[
\begin{align*}
\tau_p^{pd} &= 0.9995 \min \left\{ 1, \frac{-l_i}{\Delta l_{pd}^i} + \frac{-u_i}{\Delta u_{pd}^i} \mid \Delta l_{pd}^i, \Delta u_{pd}^i > 0 \right\} , \\
\tau_d^{pd} &= 0.9995 \min \left\{ 1, \frac{-z_i}{\Delta z_{pd}^i} + \frac{-w_i}{\Delta w_{pd}^i} \mid \Delta z_{pd}^i, \Delta w_{pd}^i > 0 \right\} .
\end{align*}
\]

After enlarging step lengths \(\tau_p^{pd}\) and \(\tau_d^{pd}\) to be \(\bar{\tau}_p\) and \(\bar{\tau}_d\), respectively, we have

\[
\begin{bmatrix}
\bar{1} \\
\bar{u}
\end{bmatrix} = \begin{bmatrix} 1 \\ u \end{bmatrix} + \tau_p \begin{bmatrix} \Delta l_{pd} \\ \Delta u_{pd} \end{bmatrix}, \\
\begin{bmatrix}
\bar{z} \\
\bar{w}
\end{bmatrix} = \begin{bmatrix} z \\ w \end{bmatrix} + \tau_d \begin{bmatrix} \Delta z_{pd} \\ \Delta w_{pd} \end{bmatrix} .
\]

Let \(\bar{v} = \bar{L}z, H = [\sigma \bar{g} \theta, \sigma \bar{g} \theta], \theta \in (0, 1),\) and \(v = \pi (\bar{v} \mid H),\)

\[
\begin{bmatrix}
\bar{v}_i \\
\bar{v}_i \in H
\end{bmatrix} = \begin{bmatrix}
\sigma \bar{g} \theta \\
\sigma \bar{g} \theta
\end{bmatrix},
\]

\[
\bar{v} = \bar{U} \bar{w}, \quad v = -\pi (\bar{v} \mid H).\]

Let \(K_{mcc} = [0; 0; 0; 0; v - \bar{v}; v - \bar{v}];\) the centrality correction direction \(\Delta X_{mcc}\) can be obtained by solving the following equation system:

\[
M(\cdot) \Delta X_{mcc} = K_{mcc} .
\]

Then the primal-dual and centrality correction directions are combined to be the total direction

\[
\Delta X = \Delta X_{pd} + \Delta X_{mcc} .
\]

If we consider the total direction \(\Delta X\) as the new primal-dual direction \(\Delta X_{pd}\), a new centrality corrections step can be implemented again.

After computing primal and dual step lengths \(\tau_p\) and \(\tau_d\) for \(\Delta X\) according to (21), the iterating points can be updated as follows:

\[
\begin{bmatrix}
x^{k+1} \\
\eta^{k+1} \\
u^k + \tau_p \Delta x^k \\
\Delta x^k
\end{bmatrix} ,
\]

\[
\begin{bmatrix}
z^{k+1} \\
\xi^{k+1} \\
w^k + \tau_d \Delta z^k \\
\Delta z^k
\end{bmatrix} .
\]

The centering parameter \(\sigma\) is very important for IPM [4]. If \(\sigma = 1,\) the equation system (20) defines a centering direction, a Newton step toward the point at which all the pairwise products are identical to \(\bar{g}.\) At the other extreme, the value \(\sigma = 0\) gives the standard Newton step toward the point at \(g = 0.\) There are several strategies to set \(\sigma.\) The first one is fixed strategy, which is a method setting \(\sigma \in [0, 1)\) and keeping \(\sigma\) unchanged during the iterations of IPM [4]. The other one is Mehrotra's predictor-corrector (MPC) method which determines the value of \(\sigma\) by using a preliminary step computation [5]. Now we give the details of MPC method. First, let \(K_{af} = K(X, \sigma \bar{g}) \mid_{\nu=0}\), and we have \(\Delta X_{af} = K_{af}\) after solving \(M(\cdot) \Delta X_{af} = -K_{af}.\) We then compute affine step lengths \(\tau_p^f\) and \(\tau_d^f\) for \(\Delta X_{af}\) according to (21). Next, we compute \(g_{af} = (1 + \tau_p^f \Delta l_{af}^f)(z + \tau_d^f \Delta z_{af}^f) - (u + \tau_d^f \Delta u_{af}^f) / (w + \tau_d^f \Delta w_{af}^f)\); then \(\sigma = (g_{af}/g)^3.\) The third strategy setting \(\sigma\) is LOQO method [24]. It defines \(\sigma\) as \(\sigma = 0.1 \times \min (\rho(1 - \xi)/\xi, \rho)^3,\) where \(\xi = \min_i \{l_i z_i, -u_i w_i\}/\bar{g}, \rho = 0.05,\) and \(\bar{p} = 2.\)

5.2. Optimal Centering Parameter. The aforementioned methods of defining centering parameter are effective for solving linear programming, but all these methods are based on experience or experiment. In this subsection, we present a novel optimization model for \(\sigma,\) and the optimal \(\sigma\) can be obtained through minimizing the equilibrium distance-quality function (ED-QF) for the predicted iterating points.

**Definition 2.** ED-QF for strict interior point \(X\) can be denoted as \(d^p_{\text{exact}}(X) = \|K(X, \eta \bar{g})\|^2,\) where \(\eta \in (0, 1).\)

The following points are obvious.

(a) \(d^p_{\text{exact}}(X) \rightarrow \|K(X, \bar{g})\|^2\) for \(\eta \rightarrow 1,\) and ED-QF represents the distance between \(X\) and \(X_{\bar{g}}; X_{\bar{g}}\) satisfies \(K(X_{\bar{g}}, \bar{g}) = 0.\)
(b) \( d_{\text{exact}}^\eta(X) \rightarrow \|K(X, 0)\|^2 \) for \( \eta \rightarrow 0 \), and ED-QF represents the distance between \( X \) and the KKT point \( X_* \). \( X_* \) satisfies \( K(X_*, 0) = 0 \). It is important to note that, for \( \eta = 0, d_{\text{exact}}^\eta(X) \) is the equivalent of quality function in [26].

So ED-QF contains the centricity and optimization of \( X \) synchronously.

We are now in a position to give the model defining optimal \( \sigma \). For iteration point \( X \), the optimal \( \sigma \) is the one which minimizes the ED-QF of next iteration point \( \bar{X}(\sigma) \):

\[
\sigma = \arg\min_{\sigma} \{d_{\text{exact}}^\eta(\bar{X}(\sigma))\},
\]

where

\[
d_{\text{exact}}^\eta(\bar{X}(\sigma)) = L_z(\bar{X}(\sigma))^2 + L_y(\bar{X}(\sigma))^2 + L_w(\bar{X}(\sigma))^2 + L_1(\bar{X}(\sigma), \eta \bar{g}(\sigma))^2 + L_u(\bar{X}(\sigma), \eta \bar{g}(\sigma))^2,
\]

\( \bar{X}(\sigma) = \{\bar{x}(\sigma); \bar{y}(\sigma); \bar{z}(\sigma); \bar{w}(\sigma); \bar{\theta}(\sigma); \bar{u}(\sigma)\} \), \( \bar{x}(\sigma) = x + \bar{\tau}_p(\sigma) \Delta \bar{x}(\sigma), \bar{y}(\sigma) = y + \bar{\tau}_p(\sigma) \Delta \bar{y}(\sigma), \bar{z}(\sigma) = z + \bar{\tau}_d(\sigma) \Delta \bar{z}(\sigma), \bar{w}(\sigma) = w + \bar{\tau}_d(\sigma) \Delta \bar{w}(\sigma), \Delta \bar{X}(\sigma) = \Delta X_{\text{opt}}, \bar{\tau}_p(\sigma), \) and \( \bar{\tau}_d(\sigma) \) are the step lengths for \( \Delta \bar{X}(\sigma) \).

The evaluation of \( d_{\text{exact}}^\eta(\bar{X}(\sigma)) \) is, however, expensive since it requires the evaluation of the problem functions and derivatives for every value of \( \sigma \). We can avoid this expense by using a linear technique. Because \( f(\cdot), h(\cdot), \) and \( g(\cdot) \) are continuous and differentiable for our SLDPC model, we have

\[
f(\bar{x}(\sigma)) = f(x) + \nabla f(x)^T(\bar{x}(\sigma) - x),
\]

\[
h(\bar{x}(\sigma)) = h(x) + \nabla h(x)^T(\bar{x}(\sigma) - x)
\]

\[
g(\bar{x}(\sigma)) = g(x) + \nabla g(x)^T(\bar{x}(\sigma) - x),
\]

\[
\Delta \bar{X}(\sigma) = \Delta X_{\text{opt}}, \bar{\tau}_p(\sigma), \) and \( \bar{\tau}_d(\sigma) \) are the step lengths for \( \Delta \bar{X}(\sigma) \).

\[
L_z(\bar{X}(\sigma))^2 = \left\| g(\bar{x}(\sigma)) - \bar{I}(\sigma) - \bar{g}\right\|^2
\]

\[
= \left\| g(\bar{x}(\sigma)) - \bar{I}(\sigma) - \bar{g}\right\|^2
\]

\[
L_y(\bar{X}(\sigma))^2 = \left\| [\bar{x}(\sigma); \bar{y}(\sigma); \bar{z}(\sigma); \bar{w}(\sigma); \bar{\theta}(\sigma); \bar{u}(\sigma)]\right\|^2
\]

\[
L_w(\bar{X}(\sigma))^2 = \left\| [\bar{x}(\sigma); \bar{y}(\sigma); \bar{z}(\sigma); \bar{w}(\sigma); \bar{\theta}(\sigma); \bar{u}(\sigma)]\right\|^2
\]

\[
= \left\| \bar{u}(\sigma) + \bar{\tau}_d(\sigma) \bar{\Delta w}(\sigma) \right\|^2
\]

\[
= (1 - \bar{\tau}_d(\sigma)) \left\| L_x(\bar{X}(\sigma)) \right\|^2
\]

\[
= (1 - \bar{\tau}_d(\sigma)) \left\| L_x(\bar{X}(\sigma)) \right\|^2
\]

\[
= (1 - \bar{\tau}_d(\sigma)) \left\| L_x(\bar{X}(\sigma)) \right\|^2
\]

According to (29) and \( \bar{g}(\sigma) = \sigma \bar{g}, \) ED-QF for \( \bar{X}(\sigma) \) can be approximated as

\[
d_{\text{approx}}^\eta(\bar{X}(\sigma)) = (1 - \bar{\tau}_d(\sigma)) \left\| L_x(\bar{X}(\sigma)) \right\|^2 + (1 - \bar{\tau}_p(\sigma))
\]

\[
\times \left\| L_y(\bar{X}(\sigma)) \right\|^2 + \left\| L_z(\bar{X}(\sigma)) \right\|^2 + \left\| L_w(\bar{X}(\sigma)) \right\|^2
\]

\[
+ \left\| L(\sigma) \bar{Z}(\sigma) - \eta \sigma \bar{g}\right\|^2
\]

\[
+ \left\| L(\sigma) \bar{W}(\sigma) \right\|^2 - \eta \sigma \bar{g}\right\|^2
\]

\[
\approx d_{\text{exact}}^\eta(\bar{X}(\sigma)).
\]

Note that \( \Delta \bar{X}(\sigma) = \Delta \bar{X}(0) + \sigma (\Delta \bar{X}(1) - \Delta \bar{X}(0)) \). Therefore, after solving linear system (20) twice to obtain \( \Delta \bar{X}(0) \) and \( \Delta \bar{X}(1) \), we can get \( \Delta \bar{X}(\sigma) \) easily for any value of \( \sigma \). Then, the dominant computational overhead in the evaluation of \( d_{\text{approx}}^\eta(\bar{X}(\sigma)) \) lies in the computation of the maximal step lengths for \( \Delta \bar{X}(\sigma) \) and the last two terms in (30), which require a few simple vector operations. We implement a one-dimensional search scheme, based on a golden trisection procedure, for \( \sigma \in [\sigma_{\text{min}}, \sigma_{\text{max}}] \), to obtain an approximate minimizer of \( d_{\text{approx}}^\eta(\bar{X}(\sigma)) \).

Algorithm COP:

\textbf{Initialization}. \( \sigma_{\text{min}} = 0, \sigma_{\text{max}} = 100, k_\sigma = 1, \bar{\tau} = 0.618, \bar{\tau}_d = 0.382, k_\sigma = 10, \Omega = 0.1 \).
Figure 1: Illustration of 10-sensor case.

Figure 2: Comparison of cost per sensor with the number of sensors.

Figure 3: CPU time with the number of sensors.

Figure 4: Comparison of 4 algorithms in the average step length.

5.3. Improved IPM.

Algorithm Improved IPM (I-IPM)

Initialization. $k = 0$, $\epsilon = 10^{-5}$; $\eta, \delta, \gamma, \theta \in (0, 1)$, and in this paper $\eta = 0.05, \delta = 0.2, \gamma = 0.05, \theta = 0.2$; and $X^0$.

Step 1. If $\|K(X^k, 0)\|_2 \leq \epsilon$, where $K(X^k, 0)$ is defined according to (15), $X^k$ is optimal and stop.

Step 2. Compute $\sigma$ by using Algorithm OCP and let $\Delta X_{pd} = \Delta X(0) + \sigma(\Delta X(1) - \Delta X(0))$.

Step 3. Compute the step lengths $\tau_p^{pd}$ and $\tau_d^{pd}$ for $\Delta X_{pd}$ according to (21).
Table 1: Data for 10-sensor case.

| Sensor | $\alpha_i$ | $\beta_i$ | $\gamma_i$ | $\tau_i$ |
|--------|------------|-----------|------------|-----------|
| S1     | 11.8998    | 1.75579   | 0.08810    | 40        |
| S2     | 16.2612    | 1.64917   | 0.08374    | 35        |
| S3     | 22.3812    | 2.75158   | 0.07764    | 60        |
| S4     | 25.5095    | 0.85752   | 0.07449    | 20        |
| S5     | 34.0386    | 2.27160   | 0.06596    | 25        |
| S6     | 49.8364    | 2.26119   | 0.05016    | 15        |
| S7     | 58.5268    | 1.14134   | 0.04147    | 30        |
| S8     | 65.5098    | 1.70347   | 0.03449    | 80        |
| S9     | 75.1267    | 0.22756   | 0.02487    | 20        |
| S10    | 95.9744    | 0.16185   | 0.00403    | 35        |

Table 2: Comparison of 2 models in tightness.

| $q$   | $Va-(8)$  | $Va-(11)$ |
|-------|-----------|-----------|
| 10    | 490.3616  | 498.5428  |
| 20    | 992.1475  | 1003.2713 |
| 30    | 1488.2213 | 1431.4722 |
| 40    | 1985.7825 | 2018.3667 |
| 50    | 2481.5163 | 2522.4782 |
| 60    | 2976.4431 | 3029.1279 |
| 70    | 3472.5163 | 3529.9108 |
| 80    | 3968.5901 | 4036.1891 |
| 90    | 4464.6638 | 4539.6210 |
| 100   | 4911.0632 | 4996.1190 |

Step 4. MCC procedure.

Step 4.1. Set $k_{mcc}^{\text{max}} = 5$.

Step 4.2. $\bar{r}_p = \min((1 + \delta)r_p^{pd}, 1)$, $\bar{r}_d = \min((1 + \delta)r_d^{pd}, 1)$.

Step 4.3. Compute $\Delta X_{mcc}$ according to (24), compute $\Delta X$ according to (25) and compute step lengths $\tau_p$ and $\tau_d$ for $\Delta X$ according to (21).

Step 4.4. If the number of corrections less than $k_{mcc}^{\text{max}}$ and $\tau_p \geq (1 + \gamma \delta)r_p^{pd}$ or $\tau_d \geq (1 + \gamma \delta)r_d^{pd}$, then $\Delta X_{pd} = \Delta X$, $r_p^{pd} = \tau_p$, and $r_d^{pd} = \tau_d$, goto Step 4.2.

Step 5. According to (26), update the iteration point by using $\Delta X$, $\tau_p$, and $\tau_d$, $k = k + 1$, goto Step 1.

6. Computational Results and Complexity

6.1. Computational Results and Analysis. In this subsection we present some numerical results to test the impact of the separable formulation for the SLDPCL and the effectiveness of the proposed approaches. The machine on which we perform all of our computations is a Pentium (R) Dual Core 2.7 GHz Lenovo-PC computer with 2GB RAM, running MS-Windows 7, and Matlab 2010b. In the implementation of our algorithm, OAM described in Section 3, MILP master problem has been solved by calling MOSEK 6, while NLP problems have been solved by using Algorithm 1-IPM. Test cases with sensors ranging from 10 to 20000 are used for simulation. Data of 10-sensor case are all listed in Table 1 and $x_{all} = 150$. The data of the other cases are created by duplicating the ten-sensor base data, whereas the number of line to cover is adjusted in proportion to the case size. Set $\varepsilon = 10^{-2}$ for OAM and set accuracy of the MILP solver MOSEK allowed to stop to be 0.1%.

The Algorithm OAM is carried out for two different models (7) (nonseparated) and (10) (separated), which are denoted as OAM (primal OAM) and S-OAM (separated OAM), respectively. We first compare the tightness of the outer approximation master problems (8) and (11). The results are displayed in Table 2, which presents the lower bound at the first iteration for each strategy. The column “$q$” reports the total number of sensors, while columns “$Va-(8)$” and “$Va-(11)$” report the optimal values of the outer approximation master problems (8) and (11) at the first iteration, respectively. The results show that, as expected, the outer approximation of a separated equivalent is tighter than the direct outer approximation of the original multivariate problem.

The results of 10-sensor case are listed in Table 3. Note that once $x_i$ is given for all $i \in Q$ (we will have $x_i = 0$ for those sensors that are not selected), it is trivial to find the set of optimal locations: just align the discs so that they do not intersect and they cover the entire line. For this purpose we choose the diameters in such a way that their sum is equal to the length of the line, which in our case is equal to $x_{all} = 150$. An illustration of a feasible solution to 10-sensor case is given in Figure 1.

In addition, for the purpose of comparison, we implement OAM, S-OAM two different strategies on the test cases with sensors ranging from 10 to 100. The comparison of the total costs of the two approaches is presented in Table 4. "$F_{t}$" denotes the total cost. As it can be seen, the S-OAM can find higher-quality solutions in comparison with OAM.

We note that (6) is a mixed integer quadratic programming (MIQP), which can be solved by many efficient general-purpose mixed integer programming solvers such as MOSEK, CPLEX, and GUROBI. We solve (6) with MOSEK directly by setting the accuracy to 0.1 and 0.05, which are the relative optimality tolerances employed by the solver. The comparison of the results between MOSEK and our approach is presented in Table 5, where columns “$C_{time}$” and “$N_{it}$” denote the execution time and number of iterations, respectively. As it can be observed, S-OAM can find better solutions within similar order of time as MOSEK, during which no more than five iterations have been executed, while the accuracy in MOSEK is 0.1. When accuracy of MOSEK is set to 0.05, S-OAM gets the same high-quality solutions as MOSEK for all cases but with less execution time. To sum up, S-OAM is better than MOSEK in both accuracy cases.

Perspective function is a well-known tool in convex analysis. By using this tool, Frangioni reformulate SLDPCL as an MISOPC [3]. And it can be solved by using MOSEK too. To test the efficiency of the proposed method, we further compare it with MISOPC method [3]. We set the termination
tolerance to 0.05 for MOSEK. The comparison of the results between MISOCPl and our approach is presented in Table 6. And Table 6 shows that solving MISOCPl with MOSEK for SLDPCL takes 2-3 times longer running time or even more than our method when the quality of the solutions is of the same level.

Table 4: Comparison of total costs of two approaches.

| Sensor | $F_C$ ($\) |
|--------|-------------|
| S1     | 13.16       |
| S2     | 13.95       |
| S3     | 0           |
| S4     | 20.00       |
| S5     | 0           |
| S6     | 0           |
| S7     | 47.88       |
| S8     | 0           |
| S9     | 20.00       |
| S10    | 35.00       |

Figure 3 shows that the proposed method suits large-scaled SLDPCL problems.

There are two NLPs in Steps 1 and 3 of Algorithm OAM, which are solved with the Algorithm I-IPM presented in Section 4. We will show the efficiency of the proposed novel S-OAM in the following paragraphs.

After deleting Step 4 from Algorithm I-IPM, set $\sigma = 0.1$, set $\sigma$ according to LOQO method, set $\sigma$ according to MPC method, or set $\sigma$ to be optimal centering parameter; we obtain 4 different IPMs, denoted as PDIPM [4], LOQO [24], MPC [5], and ED-QF, respectively. When $\sigma = 0.1$, it is shown in Figure 4 that all the average step lengths (average of the primal and dual step lengths) are taken in every iteration by different IPMs for solving the relaxation of (7). It can be seen in Figure 4 that ED-QF generally takes a larger step length than the other IPMs, which leads to better convergence.

When different IPM was used to solve the relaxation of (7) for our proposed S-OAM algorithm, Table 7 gives the number of iterations to convergence and CPU times for the PDIPM [4], PCIPM [5], and I-IPM algorithms. It is obvious that I-IPM dominates the other IPMs both in number of iterations and CPU times.

6.2. Complexity Estimate. In full generality, SLDPCL is a challenging optimization problem, as SLDPCL combines the difficulty of optimizing over integer variables with the handling of nonlinear functions. As pointed in [14], it is NP-hard problem. In our method, we reduce this NP-hard problem to MILP and convex NLP. The convex NLP can be solved efficiently by using proposed I-IPM in polynomial times. MILP can be solved by using the off-the-shelf MILP solvers effectively, although it is an NP-hard problem as well. Then, the complexity of proposed method depends on three aspects. The first one is complexity of I-IPM. Let $\epsilon \in (0, 1)$ be given. Suppose that I-IPM for solving NLP generates a sequence of iterates that satisfies $\mu_{k+1} \leq (1 - 0.1/n^2)\mu_k$, $k = 0, 1, 2, \ldots$, for some positive constants $\rho$ and $\omega$. Suppose that the starting point satisfies $\mu_0 \leq 1/\epsilon^2$ for some positive constant $\kappa$. Then the I-IPM can obtain a solution satisfying $\mu_k \leq \epsilon$ with no more than $O(n^2/\log n)$ iterations. The similar proof can be found in [27]. The second aspect of complexity of S-OAM is the complexity of solving MILP, which is an NP-hard problem. The last one is the iterations number of S-OAM, which is a hard open problem for MINLP. So, it is very hard to estimate accurately the complexity of the proposed method.

7. Conclusions

SLDPCL problems come from many application problems in common with tremendous number of variables and constraints. In this paper, we combine OAM and IPM and give a novel S-OAM for SLDPCL. Our algorithm can get good CPU time and high-quality solutions. Furthermore, our S-OAM approach could be improved in at least two ways: (a) by using the preprocessing procedure in order to fix on some binary variables before the calling of our algorithm; (b) by embedding I-IPM in a branch-and-cut approach, like that of MOSEK, in order to improve the efficiency of solving subproblems.

Conflict of Interests

The authors declare that they do not have any commercial or associative interest that represents a conflict of interests in connection with the work submitted.
Table 5: Comparison of S-OAM and MOSEK.

| q   | Accuracy 0.1 | Accuracy 0.05 |
|-----|--------------|---------------|
|     | $F_C$ ($)   | $C_{time}$ (s) | $F_C$ ($)   | $C_{time}$ (s) | $F_C$ ($)   | $C_{time}$ (s) | $N_{it}$ |
| 10  | 564.800     | 6.4           | 550.974     | 11.3          | 550.974     | 7.3           | 3        |
| 50  | 2789.234    | 12.9          | 2754.876    | 30.7          | 2752.706    | 10.2          | 3        |
| 100 | 5557.353    | 21.3          | 5510.911    | 59.2          | 5505.370    | 19.7          | 4        |
| 200 | 11090.62    | 24.7          | 11019.972   | 89.1          | 11010.43    | 22.6          | 5        |
| 500 | 28076.57    | 31.5          | 27529.88    | 202.7         | 27526.08    | 20.0          | 3        |
| 1000| 55523.13    | 40.8          | 55069.23    | 290.6         | 55052.14    | 34.1          | 4        |
| 5000| 280352.2    | 631.9         | 275261.38   | 2031.9        | 273061.2    | 279.0         | 5        |
| 10000| 557846.9   | 2177.8        | —            | —             | 550521.9    | 627.6         | 3        |
| 20000| —         | —             | —            | —             | 1100965.9   | 1124.8        | 5        |

“—” denotes that MOSEK cannot reach the preset accuracy for this instance within preset 3600 seconds.

Table 6: Comparison of S-OAM and MISOCOP.

| q   | MISOCOP [3] | S-OAM       |
|-----|-------------|-------------|
|     | $F_C$ ($)   | $C_{time}$ (s) | $F_C$ ($)   | $C_{time}$ (s) |
| 10  | 550.974     | 5.2         | 550.974     | 7.3           |
| 50  | 2752.706    | 14.6        | 2752.706    | 10.2          |
| 100 | 5513.722    | 31.1        | 5505.370    | 19.7          |
| 200 | 11016.25    | 47.6        | 11010.43    | 22.6          |
| 500 | 27547.62    | 114.9       | 27526.08    | 20.0          |
| 1000| 55052.17    | 151.7       | 55052.14    | 34.1          |
| 5000| 275505.30   | 713.2       | 273061.2    | 279.0         |
| 10000| 550878.91  | 1821.0      | 550521.9    | 627.6         |
| 20000| —          | —           | —           | —             |

“—” denotes that MOSEK cannot reach the preset accuracy for this instance within preset 3600 seconds.

Table 7: Comparison of 3 IPMs in results.

| q   | PDIPM [4] | PCIPM [5] | I-IPM |
|-----|-----------|-----------|-------|
|     | $N_d/C_{time}$ (s) | $N_d/C_{time}$ (s) |
| 100 | 23/0.13 | 18/0.13 | 12/0.12 |
| 200 | 25/0.24 | 22/0.28 | 14/0.23 |
| 500 | 23/0.61 | 20/0.62 | 13/0.51 |
| 1000| 28/1.90 | 23/1.64 | 14/1.20 |
| 5000| 33/12.77| 24/9.81 | 14/6.63 |
| 10000| 36/27.12 | 22/18.04 | 15/13.84 |
| 20000| 35/51.04 | 22/39.67 | 13/26.19 |
| Sum | 203/93.81 | 151/70.19 | 95/48.72 |

Acknowledgments

Our works are supported by Guangxi Natural Science Foundation (2013GXNSFBA019246), Guangxi Experiment Center of Information Science, the Project Sponsored by the Scientific Research Foundation of Guangxi University (XJZ110593), and National Natural Science Foundation of China (71201049, 61163060).

References

[1] A. Agnetis, E. Grande, P. B. Mirchandani, and A. Pacifici, “Covering a line segment with variable radius discs,” Computers and Operations Research, vol. 36, no. 5, pp. 1423–1436, 2009.
[2] A. Pacifici, P. Mirchandani, M. Mecoli, and A. Agnetis, “Locational decisions for Barge-tracking radars,” Tech. Rep., The A.T.L.A.S. Research Center, Tucson, Ariz, USA, 2004.
[3] A. Frangioni, C. Gentile, E. Grande, and A. Pacifici, “Projected perspective reformulations with applications in design problems,” Operations Research, vol. 59, no. 5, pp. 1225–1232, 2011.
[4] S. Wright, Primal-Dual Interior-Point Methods, Society for Industrial & Applied Math, Philadelphia, 1997.
[5] S. Mehrotra, “On the implementation of a primal-dual interiort point method,” SIAM Journal on Optimization, vol. 2, no. 4, pp. 575–601, 1992.
[6] R. Boorstyn and H. Frank, “Large-scale network topological optimization,” IEEE Transactions on Communications, vol. 25, no. 1, pp. 29–47, 1977.
[7] H. Hamacher, A. Liebers, A. Schöbel, D. Wagner, and F. Wagner, “Locating new stops in a railway network,” Electronic Notes in Theoretical Computer Science, vol. 50, no. 1, pp. 13–23, 2001.
[8] J. M. Arroyo and A. J. Conejo, “A parallel repair genetic algorithm to solve the unit commitment problem,” IEEE Transactions on Power Systems, vol. 17, no. 4, pp. 1216–1224, 2002.
[9] B. Wang, “Coverage problems in sensor networks: a survey,” ACM Computing Surveys, vol. 43, no. 4, article 32, 2011.
[10] P. M. Pardalos and M. Resende, Handbook of Applied Optimization, Oxford University Press, Oxford, UK, 2002.
[11] V. Chvátal, “A combinatorial theorem in plane geometry,” Journal of Combinatorial Theory B, vol. 18, no. 1, pp. 39–41, 1975.
[15] N. Lev-Tov and D. Peleg, "Polynomial time approximation schemes for base station coverage with minimum total radii," Computer Networks, vol. 47, no. 4, pp. 489–501, 2005.

[16] V. Bilò, I. Caragiannis, C. Kaklamanis, and P. Kanellopoulos, "Geometric clustering to minimize the sum of cluster sizes," in Proceedings of the 13th Annual European Symposium on Algorithms (LNCS '05), vol. 3669, pp. 460–471, October 2005.

[17] M. Li, X. Sun, and Y. Zhao, "Minimum-cost linear coverage by sensors with adjustable ranges," in Lecture Notes in Computer Science, vol. 6843, pp. 25–35, 2011.

[18] Z. Zhang and D. Du, "Radar placement along banks of river," Journal of Global Optimization, vol. 52, no. 4, pp. 729–741, 2012.

[19] D. Dash, A. Bishnu, A. Gupta, and S. C. Nandy, "Approximation algorithms for deployment of sensors for line segment coverage in wireless sensor networks," Wireless Networks, vol. 19, no. 5, pp. 857–870, 2013.

[20] A. Frangioni and C. Gentile, "A computational comparison of reformulations of the perspective relaxation: SOCP vs. cutting planes," Operations Research Letters, vol. 37, no. 3, pp. 206–210, 2009.

[21] P. Bonami, L. T. Biegler, A. R. Conn et al., "An algorithmic framework for convex mixed integer nonlinear programs," Discrete Optimization, vol. 5, no. 2, pp. 186–204, 2008.

[22] M. Duran and I. Grossmann, "An outer-approximation algorithm for a class of mixed-integer nonlinear programs," Mathematical Programming, vol. 36, no. 3, pp. 307–339, 1986.

[23] I. Contreras, J. Cordeau, and G. Laporte, "Benders decomposition for large-scale uncapacitated hub location," Operations Research, vol. 59, no. 6, pp. 1477–1490, 2011.

[24] R. J. Vanderbei and D. F. Shanno, "An interior-point algorithm for nonconvex nonlinear programming," Computational Optimization and Applications, vol. 13, no. 1–3, pp. 231–252, 1999.

[25] H. Hijazi, P. Bonami, and A. Ouorou, "An outer-inner approximation for separable MINLPs," http://www.optimization-online.org.

[26] J. Nocedal, A. Wächter, and R. A. Waltz, "Adaptive barrier update strategies for nonlinear interior methods," SIAM Journal on Optimization, vol. 19, no. 4, pp. 1674–1693, 2008.

[27] Y. Nesterov, M. J. Todd, and Y. Ye, "Infeasible-start primal-dual methods and infeasibility detectors for nonlinear programming problems," Mathematical Programming B, vol. 84, no. 2, pp. 227–267, 1999.