European Journal of Operational Research 222 (2012) 17–30

Contents lists available at SciVerse ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Discrete Optimization

Integer linear programming models for grid-based light post location problem

Md. Noor-E-Alam [a], Andrew Mah [b], John Doucette [a,c,*]

[a] Dept. of Mechanical Engineering, University of Alberta, 4–9 Mechanical Engineering Bldg., Edmonton, AB, Canada T6G 2G8
[b] Ekoik Labs, Toronto, ON, Canada
[c] TRLabs, Edmonton, AB, Canada

ARTICLE INFO

Article history:
Received 11 January 2012
Accepted 24 April 2012
Available online 1 May 2012

Keywords:
Linear programming
Location analysis
Grid-based location problems
Integer programming

ABSTRACT

Selecting optimal location is a key decision problem in business and engineering. This research focuses to develop mathematical models for a special type of location problems called grid-based location problems. It uses a real-world problem of placing lights in a park to minimize the amount of darkness and excess supply. The non-linear nature of the supply function (arising from the light physics) and heterogeneous demand distribution make this decision problem truly intractable to solve. We develop ILP models that are designed to provide the optimal solution for the light post problem: the total number of light posts, the location of each light post, and their capacities (i.e., brightness). Finally, the ILP models are implemented within a standard modeling language and solved with the CPLEX solver. Results show that the ILP models are quite efficient in solving moderately sized problems with a very small optimality gap.

1. Introduction

Many real-world facility location problems can be approximated by a grid-based system of small-sized cells. These cells can then be used to model a heterogeneous demand distribution. We can also express the amount of supply in each cell from its supply distribution relationships with the various potential facility locations. Based on these demand distributions and supply relationships, we can then determine the optimal capacities and locations to place our facilities while fulfilling certain objectives. In this paper, these types of location problems are referred to as *grid-based location problems* (GBLPs). In the GBLPs, we will seek the optimum number, location(s), and size(s) of facilities to place. The applications of GBLPs are wide ranging, and include problems in business, every discipline of engineering, defence, resource exploitation, and even the medical sciences. To make such complex decisions, we need to develop mathematical models, and procedures to solve them.

Determining optimal location is a common, and often complex, problem in business and engineering. Over the last several decades and especially in recent years, several methods have been developed in the area of location theory resulting in a number of notable solving methods. These methods are problem specific and particularly designed for the various types of the location problem. One of the most significant facility location problems was first proposed by Cooper (1963), now well-known as the multisource Weber problem. The Weber problem has a known number of facilities and all the fixed costs for the facilities are equal. Since 1963, a lot of research has been done on the Weber problem. Wesolowsky (1972) proposed a model for the solution of the Weber problem using rectilinear distances. Sherali and Nordai (1988) focused on a capacitated multi-facility Weber problem (CMFWP) and demonstrated that the CMFWP is NP-Hard. Manzour-al-Ajdad et al. (2012) proposed an algorithm for solving a single-source CMFWP. Katz and Cooper (1974) first proposed a probabilistic multi-facility Weber problem which was later revisited by Altinel et al. (2009). A two-dimensional facility model is discussed by Francis (1964) to locate multiple new facilities with respect to existing facilities.

The complexity of the location problem depends on the nature of the problem and the criterion to be considered to make the decision. These criterions are selected by the decision maker from the problem description (Teixeira and Antunes, 2008). Marín (2011) described a new discrete location problem where the number of customers allocated to every plant has to be balanced. Ingolfsson et al. (2008) described an ambulance location optimization model that minimizes the number of ambulances needed to provide a specified service level. The model measures service level as the fraction of calls reached within a given time standard and considers response time to be composed of a random delay (prior to travel to the scene) plus a random travel time. Drezner and Wesolowsky (1997) proposed a method of placing signal detectors to cover a certain area such that the probability that an event is not detected is minimized.

Another special type of location problem is a location problem with the objective of coverage, which is first introduced by Church

* Corresponding author at: Dept. of Mechanical Engineering, University of Alberta, 4–9 Mechanical Engineering Bldg., Edmonton, AB, Canada T6G 2G8. Tel.: +1 780 492 2651.
E-mail address: john.doucette@ualberta.ca (J. Doucette).

0377-2217/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.ejor.2012.04.032

and ReVelle (1974). It ensures a set of facilities for each customer. The key applications of this model are to find optimum location of emergency services, retail facilities, cell-phone towers and sensor networks. The well-known *uncapacitated facility location* (UFL) problem is similar to these problems, except for the consideration of variable transportation costs (Wolsey, 1998). Moreover, the UFL becomes a *capacitated facility location* (CFL) problem when there is an upper limit for the amount of supply (Ghiani et al., 2002; Chen et al., 2010). For more information on the coverage location models, readers are referred to Berman et al. (2010). Of all the models developed, it is important to recognize that models to solve the location problems can be classified into two distinct groups: discrete location analysis and continuous location analysis. Discrete location analysis, the most common form of modeling a location problem, typically refers to the use of a node-and-network (transportation) approach where facilities and supply points are modeled as the vertices and nodes (Domschke and Krispin, 1997). Continuous location analysis involves the modeling of the location problem on a continuous plane. With the continuous location–allocation problem, all customer demands are coordinate points and furthermore, the feasible solution for the optimal placement of the facilities can be any coordinate point in the plane. Daskin (1995) points out that modeling the location as a grid can be NP-Complete, and as such, a transportation network (discrete location analysis) is typically employed.

To make location decisions, we need to develop mathematical models, more specifically, *integer linear programming* (ILP) problems. To model a real-world problem, we generally need to consider a large number of discrete variables, a heterogeneous demand distribution, non-linear supply distributions, and fixed costs associated with facility placement. Furthermore, to get an optimum decision, the ILP models need to be designed in such a way that they will simultaneously determine the locations, sizes and number of facilities to achieve certain objectives. Combinations of these considerations make the problem a large scale ILP problem, which are generally not scalable and often become intractable even with small problems. Therefore different types of heuristics are used to find the near optimal solution. Genetic search algorithm has been used to find solutions for location problems (Abdinnour-Helm and Venkataramanan, 1998). A genetic search algorithm is also used by Taniguchi et al. (1999) to obtain a near optimal solution for a logistics terminal location problem that also factors in traffic conditions by using queuing theory and nonlinear programming to trade-off between both transportation and facility costs at terminals to minimize total logistics costs. In Aytug and Saydam (2002), a *genetic algorithm* (GA) is used to solve large-scale maximum expected covering location problems. Methods such as the simultaneous perturbation stochastic approximation (SPSA), finite difference gradient (FDG), and very fast simulated annealing (VFSA) algorithms have also been used. Bangerth et al. (2006) compared and analyzed the efficiency, effectiveness, and reliability of these optimization algorithms for solving location problems. They found that none of these algorithms guarantees the optimal solution, but demonstrated that both SPSA and VFSA are very efficient in finding nearly optimal solutions with a high probability. Other methods for solving location problems proposed in the past few years have included methods such as a gravity model (Kubis and Hartmann, 2007), ILP-based formulations (Chen et al., 2005), the use of a Tabu search (Gendron et al., 2003), and the usage of a Greedy Algorithm (Zhang, 2006). Canbolat and Wesolowsky (2010) proposed an alternate local search heuristic approach to solving the Weber problem with a probabilistic line barrier method.

In this paper, we propose new formulations for a multisource location problem with the goal of determining the optimal combination for a facility distribution problem: the number of facilities, the location of each facility, and their capacities. Furthermore, in real-world situations, demand is not a singular point, but rather, many individual points located adjacent to each other forming a heterogeneous distribution that is extremely complex in nature. Such facility location problems can be approximated by a GBLP, where the entire area of this location problem is divided into small cells. These cells are then used to locate the heterogeneous demand distribution. On the other hand, we can express the amount of supply in each cell associated with each individual facility located in a specific cell from its supply distribution relationship. From this demand distribution and supply relationship, we have to place sufficiently-sized facilities in such a way that we can fulfill certain objectives. Our research herein focuses on the development of ILP models for GBLP, using the grid-based light post location problem to make optimal decisions in installation of lights in a city park.

The remainder of this paper is organized as follows. Section 2 provides the description of the problem with demand and supply calculation process. In Section 3, we discuss the basic model with simplified supply distribution. In Section 4, we propose two enhanced models with enhanced supply distribution. Section 5 describes the result analysis. Section 6 ends with conclusions and future research opportunities.

2. Problem description

Suppose we consider a city park, described as a 2-dimensional grid of known dimensions. Light posts must be installed throughout the park to provide adequate lighting conditions. We must determine the location and light intensity of each light post such that dark areas are lit and excess (waste) lighting is minimized. The brighter the light source, the more expensive this will be due to installation and electricity costs. As such, the objective is to satisfy the demand as much as possible while minimizing excess supply. Factors affecting the number of lights, their size, and their placement are many and varied. In a city park, there are different areas used for various purposes. Trees in the park and its topography create demand variation throughout the park. Furthermore, installing lights in boundary regions would not be feasible due to various physical restrictions such as roadways and underground power cables for utility service. This city park can be represented by a grid-based area, where the heterogeneous demand distribution can be represented by each cell in the grid. The idea is that the light sources should be placed in such a way that the areas they illuminate don't overlap too much, but not so far apart that there are unlit cells.

2.1. Demand distribution

A key characteristic of this problem is that the demand distribution is not composed of discrete points but rather a gradient of interrelated demand points. While the exact relationship between the various points is either not easily defined or may not be known, the existence of a relationship can be assumed. For example, light level measurements can be taken at coarse discrete intervals in order to determine light requirements (demands). While the data is pertinent for discrete locations, the data does not imply that the area where no data has been taken has a value of zero. As such, in order to infill the data grid to estimate the light value, we can use the known data measurements to estimate the values for which no measurements were taken in the two-dimensional plane. To model the interrelated demand data (i.e., generate the demand distribution), we turn to *finite element method* (FEM), a technique commonly used in many fields of engineering (Chapra and Canale, 2002). We applied Liebmann's method (also known as the Gauss–Seidel method) to the demand grids. In order to apply the

finite difference method of FEM to a problem, there must be some correlation between each recorded point. In this problem, it can be assumed that a cell adjacent to a location that was measured would have similar lighting requirements as the measured cell. Thus, it can be stated that the lighting required between two linear points would be a function of the magnitudes at each point as well as the distance from each point and therefore, the finite difference method can be applied. The following Eq. (1) is applied at each grid point:

$$D_{i,j} = \frac{1}{4}\left(D_{i-1,j} + D_{i+1,j} + D_{i,j-1} + D_{i,j+1}\right) \quad \forall i,j \tag{1}$$

Here, $D_{i,j}$ is the demand for the grid point whose coordinates are at (i,j), and were determined using Liebmann's method, an iterative convergence approach. Note that although we happened to use the specific method described to enumerate the individual cell-by-cell demands in our various grids, the source of demand data in and of itself is of little importance to the contribution herein. In reality, demands are simply be inputs to the problem, and would be determined by whatever method was appropriate for the user. Using the finite element method, one would take sample the actual demands at a small number of cells in the grid, perhaps through use of a light meter, and then apply the finite element method to calculate demands at the unmeasured cells. But of course, one could simply acquire the full demand distribution directly by measuring the actual demands in all cells individually, and therefore the finite element method or some other estimation technique would not be needed.

2.2. Supply calculations

Supply for a grid cell associated with each source can be calculated on the basis of the distribution of light brightness throughout the grid. It is well known that an inverse square relationship exists between the brightness of light and the distance from the light source (Simons and Bean, 2001). According to this relationship, brightness at distance r can be calculated with Eq. (2), where S is the supply at distance r and P is the luminosity of the point source.

$$S = \frac{P}{r^2} \tag{2}$$

Fig. 1 shows the apparent brightness of a source with luminosity P at distances 1–5 distance units with the above relationship. The surface brightness decreases as the distance increases because the light must spread out over a larger surface. However, this relationship is not entirely accurate; various locations on a horizontal plane will have slightly different degrees of brightness since they will all be somewhat different distances from the point source of light. A better representation of the geometry can be seen in Fig. 2, where the brightness can be calculated with Eqs. (3)–(5) (Simons and Bean, 2001). In this model, the vertical distance between the point source, o, and point a on the horizontal plane is r, as we used it in the previous simplified model, above. The angle between vertical line oa and line ob is α_b, and the angle between vertical line oa and line oc is α_c. In this relationship, as we move along the horizontal plane away from point a, the distance between the point source and the various locations on the plane will increase. As a result, the amount of supply will decrease according to Eqs. (3)–(5), where S_a,S_b and S_c represent the total amount of available supply at location a, b and c.

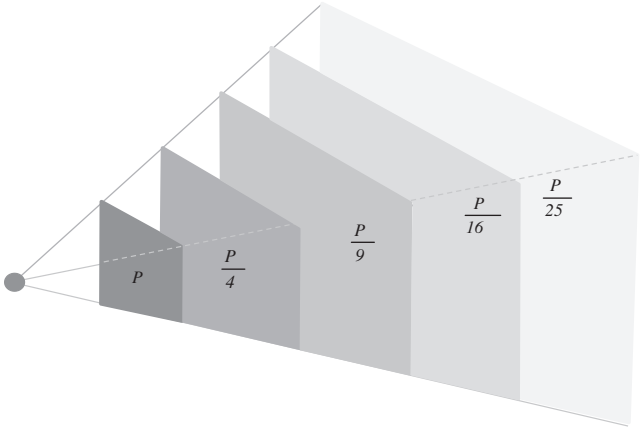

Fig. 1. Basic relationship between light brightness and distance, adapted from NASA (2006).

$$S_a = \frac{P}{r^2} \tag{3}$$

$$S_b = \frac{P}{r^2}\cos(\alpha_b) \tag{4}$$

$$S_c = \frac{P}{r^2}\cos(\alpha_c) \tag{5}$$

3. Basic model

As mentioned earlier, our goal is to develop one or more optimization models that will help us to determine the location, size, and number of light posts to place in order that we can achieve an optimal distribution of light over a grid with varying demands for light intensity. As a first step, we develop a basic model, which we will expand on in later, more accurate models.

3.1. Simplified supply calculation

Before presenting the model itself, we first need to simplify the scale of the problem by assuming that each cell within the grid is uniform throughout the entire cell. In other words, the amount of light intensity at one point in the cell is the same as in all other points in the cell. Furthermore, we will assume that certain neighboring cells will have identical light intensities, as shown in Fig. 3.

Here, the individual cell directly below the light source will have the light intensity calculated by Eq. (3). Then if we expand out to a 3×3 grid surrounding that cell, all of those new cells will have the light intensity calculated by Eq. (4). Expanding again to a 5×5 grid centered on the light source, then those cells will have the light intensity calculated by Eq. (5), and so on. We can also note that angles α_b and α_c can be calculated by Eqs. (6) and (7), where ab

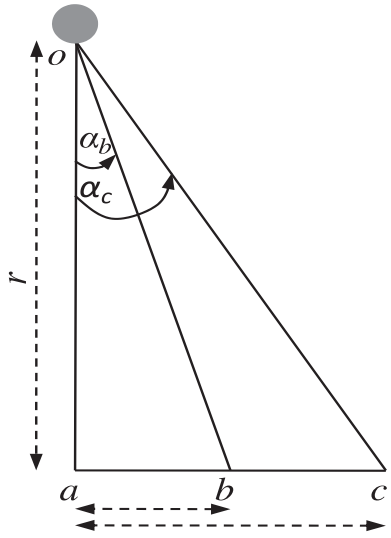

Fig. 2. Geometry of a point source of light above a horizontal plane.

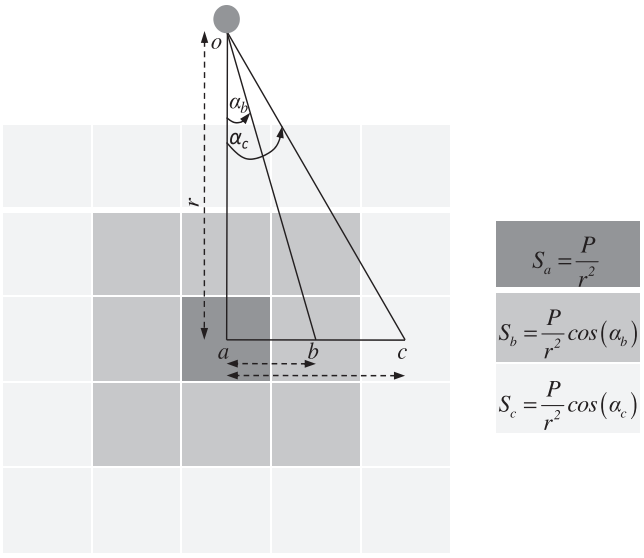

Fig. 3. Distribution of light supply for the basic model.

is the distance between points a and b, and ac is the distance between points a and c. This will become particularly important as we further develop our models.

$$\alpha_b = tan^{-1}\left(\frac{ab}{r}\right) \tag{6}$$

$$\alpha_c = tan^{-1}\left(\frac{ac}{r}\right) \tag{7}$$

3.2. Basic optimization model

In order to formulate our basic optimization model, we first need to define the notation we will use, as follows:

Q is the set of all x-coordinates in the grid, indexed by i
R is the set of all y-coordinates in the grid, indexed by j
i_{max} is the maximum value of i
i_{min} is the minimum value of i
j_{max} is the maximum value of j
j_{min} is the minimum value of j
β is the system boundary constant
N is the set of all light sources, indexed by n
n_{opt} is the optimum number of light sources
S_{ij} is the total supply at location (i, j)
S_{ijn} is the supply at location (i, j) from the nth light post
S_{in} is the supply at location (i) in x direction from the light source n
S_{jn} is the supply at location (j) in y direction from the light source n
x_n is the x coordinate of the optimal location of the nth light post
y_n is the y coordinate of the optimal location of the nth light post
P_n is the optimum size of the nth light source (required to be integer)
UB is the upper bound on decision variable P_n

It is obvious from the nature of our light demand distribution that the optimal placement of the light post would satisfy as much demand as possible while also minimizing the light source surplus. Therefore, our objective function will be to minimize both the unmet demand and extra supply of light. To fulfill this goal, Eq. (8) will be used as the objective function for the basic model:

$$Minimize \sum_{i\in Q}\sum_{j\in R}|D_{ij} - S_{ij}| \tag{8}$$

In order to minimize this objective function, we now define the feasible region with the following constraint equations, being careful to appropriately represent the model illustrated in Fig. 3. To do so, we first need to recognize that the amount of light, S_{ijn}, available at coordinates (i, j) is the minimum of the light available if we calculate along the x-axis only, or along the y-axis only. In order to arrive at the amount of available light for some arbitrary cell, we can calculate the available light the cell would have if we considered only the x-axis direction and then only the y-axis direction, and take the smaller of the two values. For instance, according to the discussion relating to Fig. 3, the cell second from the top and furthest to the left has available light as defined by Eq. (5). If we calculate the available light at that cell using only the x-axis direction, we would find the distance from the light source would be ac, and therefore we would use Eq. (5). However, if we calculate the available light using only the y-axis direction, we would find the distance from the light source would be ab, and so we would need to use Eq. (4) instead. The former provides the smaller value for available light at that cell, and so that is the equation we must use.

From the above illustration of light source distribution, Eqs. (9) and (10) can be used to calculate the supply of the nth light source in the x-axis and y-axis directions, with a single source located at (x_n, y_n). These equations follow from Eqs. (3)–(5), above.

$$S_{in} = \frac{P_n}{r^2} cos\left(tan^{-1}\left(\frac{|i - x_n|}{r}\right)\right) \quad \forall i, n \tag{9}$$

$$S_{jn} = \frac{P_n}{r^2} cos\left(tan^{-1}\left(\frac{|j - y_n|}{r}\right)\right) \quad \forall i, n \tag{10}$$

Finally, the supply of light at location (i, j) is calculated with Eq. (11) for nth light source. Total supply for all light sources is calculated with Eq. (12).

$$S_{ijn} = min(S_{in}, S_{jn}) \quad \forall i, j, n \tag{11}$$

$$S_{ij} = \sum_{n\in N} S_{ijn} \quad \forall i, j \tag{12}$$

The following two bounding constraints (13) and (14) are incorporated so that our optimization problem will not consider the feasible region very near to the boundary area. Constraint Eq. (15) is used to put an upper bound on decision variable P_n.

$$i_{\min} + \beta \leqslant x_n \leqslant i_{\max} - \beta \quad \forall n \tag{13}$$

$$j_{\min} + \beta \leqslant y_n \leqslant j_{\max} - \beta \quad \forall n \tag{14}$$

$$0 \leqslant P_n \leqslant UB \quad \forall n \tag{15}$$

The system boundary constant, β, can be determined empirically on the basis of problem description, and the upper bound, UB, is determined depending on the maximum magnitude of the demand. Eqs. (8)–(15) constitute the basic optimization model.

3.3. Equivalent basic ILP model

In the above optimization model, the objective function and some of the constraint equations are not linear. Our objective of this research is to develop an equivalent *integer linear programming* (ILP) model to get the optimum number, locations, and sizes of light posts.

3.3.1. Linearization of objective function

In our basic model, the objective function has an absolute operator, which makes this a non-linear function. However, we can develop a set of equivalent linear equations to handle this nonlinearity in the objective function. We introduce the following new notation in addition to the notation we have already used:

ED_{ij} is the excess demand at location (i, j)
ES_{ij} is the excess supply at location (i, j)

They can now linearize the objective function as follows in Eqs. (16)–(20):

$$Minimize \sum_{i\in Q}\sum_{j\in R}(ED_{ij} + ES_{ij}) \quad (16)$$

$$ED_{ij} \geqslant D_{ij} - S_{ij} \quad \forall i,j \quad (17)$$

$$ED_{ij} \geqslant 0 \quad \forall i,j \quad (18)$$

$$ES_{ij} \geqslant S_{ij} - D_{ij} \quad \forall i,j \quad (19)$$

$$ES_{ij} \geqslant 0 \quad \forall i,j \quad (20)$$

3.3.2. Linearization of constraint equations

From the nature of light distribution, we know that the further a cell is located from the source cell, the less the light supply there will be. For simplification, we ignore as negligible any supply more than 2 units distant from the source, as illustrated in Fig. 3. To develop an equivalent linear equation for constraint Eqs. (9) and (10), sets of piecewise if-then constraints in (21) and (22) are developed. From this it follows that we may have seven possible cases in the x-axis direction and seven possible cases in the y-axis direction.

$$S_{in} = \begin{cases} \frac{p_n}{r^2} & \text{if } i = x_n \\ \frac{p_n}{r^2} cos(\tan^{-1}(\frac{1}{r})) & \text{if } i = x_n + 1 \\ \frac{p_n}{r^2} cos(\tan^{-1}(\frac{2}{r})) & \text{if } i = x_n + 2 \\ \frac{p_n}{r^2} cos(\tan^{-1}(\frac{1}{r})) & \text{if } i = x_n - 1 \\ \frac{p_n}{r^2} cos(\tan^{-1}(\frac{2}{r})) & \text{if } i = x_n - 2 \\ 0 & \text{if } i > x_n + 2 \\ 0 & \text{if } i < x_n - 2 \end{cases} \quad \forall i,n \quad (21)$$

$$S_{jn} = \begin{cases} \frac{p_n}{r^2} & \text{if } j = y_n \\ \frac{p_n}{r^2} cos(\tan^{-1}(\frac{1}{r})) & \text{if } j = y_n + 1 \\ \frac{p_n}{r^2} cos(\tan^{-1}(\frac{2}{r})) & \text{if } j = y_n + 2 \\ \frac{p_n}{r^2} cos(tan^{-1}(\frac{1}{r})) & \text{if } j = y_n - 1 \\ \frac{p_n}{r^2} cos(\tan^{-1}(\frac{2}{r})) & \text{if } j = y_n - 2 \\ 0 & \text{if } j > y_n + 2 \\ 0 & \text{if } j < y_n - 2 \end{cases} \quad \forall j,n \quad (22)$$

However, if–then constraints are themselves non-linear. Their equivalent linear equations can be developed using the method from Winston and Venkataramanan (2003). If an if–then constraint can be expressed in a form where: if some function $K(x_1, x_2, \ldots, x_n) > 0$, then some other function $L(x_1, x_2, \ldots, x_n) \geqslant 0$, then we can replace that if–then pair of equations with the following two linear equations where $y \in \{0,1\}$ and M is some large positive number:

$$-L(x_1, x_2, \ldots, x_n) \leqslant My \quad (23)$$

$$K(x_1, x_2, \ldots, x_n) \leqslant M(1 - y) \quad (24)$$

To use this technique, we can express the constraints in the first part of Eq. (21) with Eqs. (25)–(28), where z_1 and z_2 are binary variables:

$$z_1 = 1 \quad if\ i + 1 > x_n \quad \forall n \quad (25)$$

$$z_2 = 1 \quad if\ x_n + 1 > i \quad \forall n \quad (26)$$

$$S_{in} = \frac{p_n}{r^2} \quad if\ z_1 + z_2 > 1 \quad \forall n \quad (27)$$

$$S_{in} = 0 \quad if\ z_1 + z_2 < 2 \quad \forall n \quad (28)$$

We can do the same for the remaining constraints in Eq. (21) and those in Eq. (22) as well, where each of the 14 separate if–then constraints is expressed as an equivalent set of four new if–then constraint equations. Each of those is further converted into an equivalent pair of linear constraints as Eqs. (23) and (24), for a total of 112 sets of constraints ("sets" because we have one of each of those constraints for each light source, n). And this is only for linearization of constraints (21) and (22). Eq. (11) must also be linearized in a similar manner.

3.4. Preliminary results

We solve our instance of the above problem on an 8 processor ACPI multiprocessor X64-based PC with Intel Xeon® CPU X5460 running at 3.16 GHz with 32 GB memory. We have implemented our models in AMPL (Fourer et al., 2002), and used CPLEX 11.2 solver (ILOG, 2007) to solve them.

To obtain preliminary results, we chose a small problem with a 7×7 grid, shown in Fig. 4. In this solution, other parameters are assumed as follows: $\beta = 2$, $UB = 10$, and $r = 2$. In addition, we solve five instances of the problem, with 1–5 light sources. We plot the objective function values of the five instances of the problem in Fig. 5. While we show data for all five instance, only those with 1 and 2 light sources solve to optimality in a five-day runtime window (that's five days for each instance). For the instances with 3, 4, and 5 light sources, after a five-day runtime for each, the solver was only able to obtain sub-optimal solutions with optimality gaps of 8%, 64%, and 79%, respectively. This suggests that this model is practically intractable, likely a result of the great many binary variables and associated constraints, which tend to make optimization problems computationally heavy. This drives our efforts for an enhanced model in the next section.

4. Enhanced models

To reduce the above computational burden in the basic ILP model and to represent a more precise light distribution model, we can now propose enhanced models. The light distribution model used above does not accurately represent the real light distribution model. Instead, various locations on a horizontal plane will have different degrees of brightness since they will all be

2.10	2.40	2.40	2.30	2.20	1.60	0.50
2.40	2.50	2.60	2.40	2.30	1.60	0.50
2.40	2.60	2.50	2.40	2.40	1.70	0.60
2.40	2.40	2.50	2.50	2.50	1.80	0.60
2.30	2.40	2.40	2.60	2.50	1.90	0.60
1.70	1.70	1.70	1.80	1.90	1.20	0.50
0.60	0.60	0.60	0.60	0.60	0.50	0.00

Fig. 4. Demand distribution for 7×7 grid.

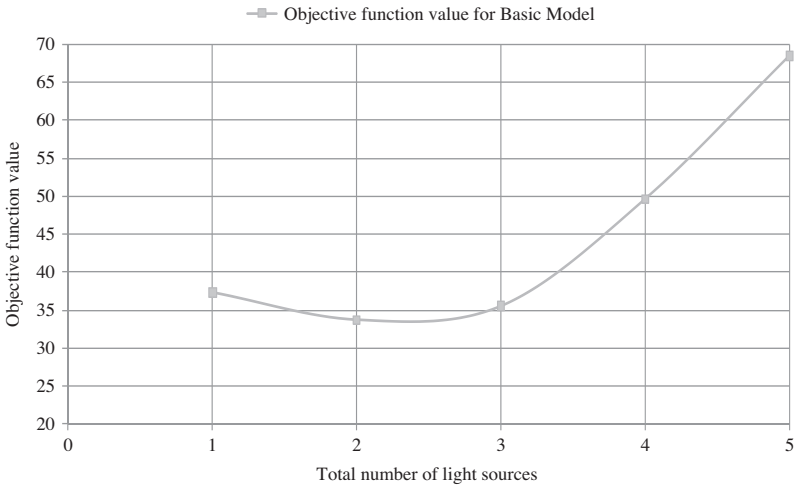

Fig. 5. Objective function values for 7×7 grid test cases solved with the basic ILP model.

somewhat different distances from the point source of light. Because the rectilinear distances from the point source to the different cells are not the same, neighbouring cells do not necessarily have identical light intensities. A better representation of the supply distribution can be seen in Fig. 6, where the brightness can be calculated with the relationships shown. These relationships are determined according to the brightness calculation process described in Fig. 2 with Eqs. (3)–(5), except that in the present figure, distances are calculated as the 3-dimensional Euclidean distances between the light source and the centres of the various cells. In those equations, the vertical distance between the point source, *o*, and centre point of a grid, *a*, is *r*, as we used it in the previous model, above. The angle between vertical line *oa* and line *ob* or *ob*′ is α_b, the angle between vertical line *oa* and line *oc* is α_c, the angle between vertical line *oa* and line *od* or *od*′ is α_d, the angle between vertical line *oa* and line *oe* or *oe*′ is α_e, and the angle between vertical line *oa* and line *of* is α_f. In the following two enhanced ILP models we will consider this exact calculation of light supply when optimizing light post locations.

4.1. Model 2

In the basic model we have a large number of constraint equations and 0/1 decision variables, even for a small problem (say, the 7×7 grid solved earlier) and a simplified demand distribution. As a result, that ILP model is simply not scalable as described. To overcome this difficulty we have developed a second ILP model, in which we use the same objective function, but a highly simplified approach for defining the feasible region. In fact, we can replace the original sets of constraints with a single set by pre-processing supply data.

We see in the basic model that it is required to consider $112n$ linear constraint equations just for Eqs. (9) and (10). As a result we need to consider a huge number of linear constraints and binary variables in order to calculate the simplified supply distribution. To overcome this difficulty, we use the following constraint Eq. (29), where the *x* and *y* coordinates of the location are considered as parameters, instead of variables. We introduce the following new notation in addition to the notation we have already used:

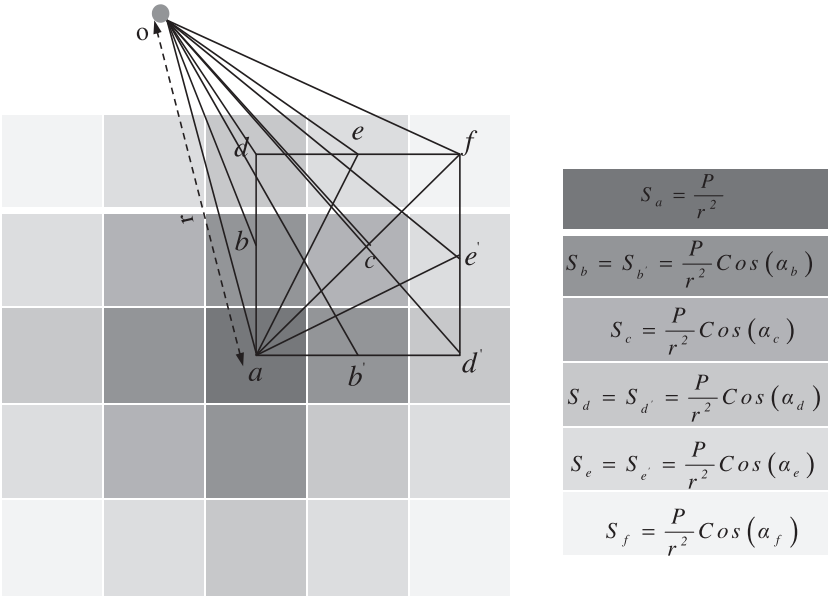

Fig. 6. Exact distribution of light supply.

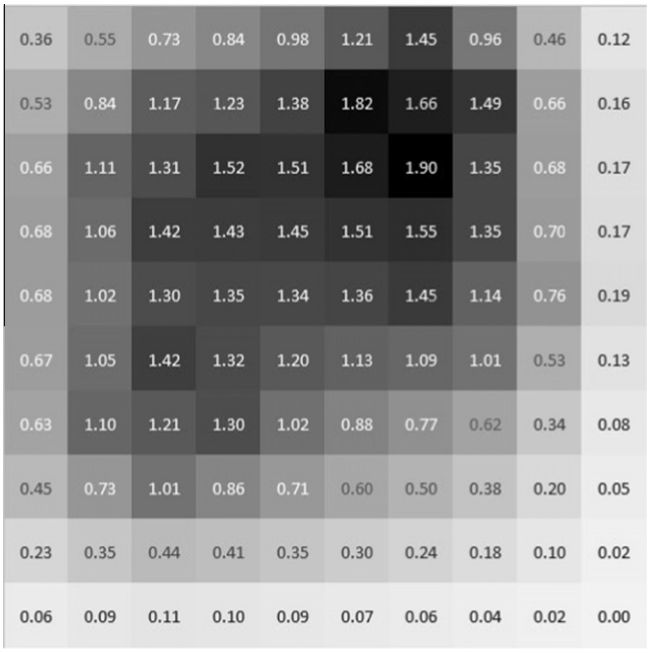

Fig. 7. Demand distribution for 10×10 grid.

$\boldsymbol{X}$ is the set of all x-coordinates of the light source, indexed by x, $i_{min} + \beta \leqslant x \leqslant i_{max} - \beta$
Y is the set of all y-coordinates of the light source, indexed by y, $j_{min} + \beta \leqslant y \leqslant j_{max} - \beta$
P_{xy} is the size of the light source at location (x, y) and it is required to be integer
UB_{xy} is the upper bound on decision variable P_{xy}

From the illustration of light source distribution (in Fig. 6), Eq. (29) can be used to calculate the supply in each cell (i, j). This equation follows from the relationships provided in Fig. 6, above. Suppose the location of a light post is at point a, and we want to calculate the supply at point e. In this case we need to find the value of angle α_e, to calculate the supply at point e. This angle can be calculated by the ratio of ae and r. The value of ae is the distance between the point, a and the point, e. On the other hand, the total supply in a particular cell (i, j) is the summation of all the individual supplies coming from all light sources. Considering these facts, the following equation can be used to calculate the supply in each cell:

$$S_{ij} = \sum_{x \in X} \sum_{y \in Y} \frac{P_{xy}}{r^2} cos\left(\tan^{-1}\left(\frac{\sqrt{(i-x)^2 + (j-y)^2}}{r}\right)\right) \quad \forall i,j \tag{29}$$

Furthermore, bounding constraint (30) is used to put an upper bound on decision variable P_{xy}.

$$0 \leqslant P_{xy} \leqslant UB_{xy} \quad \forall x, y \tag{30}$$

The objective function, (16), along with the constraint equations, (17)–(20), (29) and (30), constitute the whole of our first enhanced ILP model, which we will refer to as Model 2. To better handle this model, the *cos*() component in the Eq. (29) is calculated as a part of data pre-processing and fed into the model. We can note that in Model 2, we do not actually have any capability to control the number of light sources, and in fact, it is conceivable that a light source could be placed at each cell. While this can be dealt with by including a cost (in the objective function) for each light source we place, we can also more directly control that as we did in the basic model. We do that in our next enhanced model, which we will call Model 3.

4.2. Model 3

To control the number of sources in Model 2, we can incorporate the following sets of constraint equations, (31)–(33), where T_{xy} is a binary variable and n_a is the total number of allowable light sources:

$$T_{xy} = 1 \quad \text{if } P_{xy} > 0 \quad \forall x, y \tag{31}$$

$$T_{xy} = 0 \quad \text{if } P_{xy} = 0 \quad \forall x, y \tag{32}$$

$$\sum_{x \in X} \sum_{y \in Y} T_{xy} = n_a \tag{33}$$

Finally, the objective function, (16), along with the constraint equations, (17)–(20) and (29)–(33) constitute our second enhanced ILP model, which is referred to as Model 3. By controlling the number of light sources, we can optimize our problem for various instances of n_a in order to observe how this will impact the objective function, which provides additional insights we might not have otherwise.

As with many of the equations we have seen so far, Eqs. (31) and (32) are non-linear, and so they too will need to be replaced with equivalent linear equations, as we did earlier. We can use the technique outlined in Eqs. (23) and (24) to develop equivalent linear equations for these two non-linear equations.

5. Result analysis

We use the same experimental set up described earlier to run test case solutions for the enhanced models, Models 2 and 3. However, because these enhanced models are much more scalable, we use the larger 10×10, 10×12, 12×12, 10×15, 10×17, 10×20 and 15×15 test-case grids with demands shown in Figs. 7–13. We used a CPLEX *mipgap* setting of 0.001, which means all test cases solved to full termination are provably within 0.1% of optimality.

Figs. 14–20 show the respective solution data for these seven test-case grids. In each of those figures, the square data points represent the *optimum objective function values* (OOFVs) of the optimally solved test case with the indicated number of light sources (i.e., input parameter, n_a) using Model 3. The triangular data points represent the OOFV of the Model 2 solutions, and we note that for these data points, the number of light sources indicated along the x-axis is not an input, but rather, is obtained from the solution itself along with the OOFV. The diamond data points represent the CPU time required to solve the test cases with the indicated number of light sources. Note that in all seven of these figures, the OOFV data points are to be read against the left-hand y-axes, while CPU time data points are to be read against the right-hand y-axes.

We can observe that in general, Model 3 is much more capable than Model 1 of solving larger test cases in a reasonable period of time, at least those of intermediate size (e.g., 10×17). For instance, all solutions for the 10×10, 10×12, 12×12, 10×15 and 10×17 test cases were solved to optimality (well, within the 0.1% optimality gap specified above). The highly irregular nature of CPU times for those test cases was unexpected, but we think the reason is because peculiarities within the problem, although minor in the grand scheme of things, can add enough additional complexity to individual test cases that the underlying complexity of the problem is overwhelmed and CPU time can triple, say from approximately 12 seconds or so on the 10×17 test case with 20 light sources to a little under 40 seconds for the test case with 21 light sources. In other words, due to the heterogeneity of demand distribution, some instances of the problem might create much tighter LP

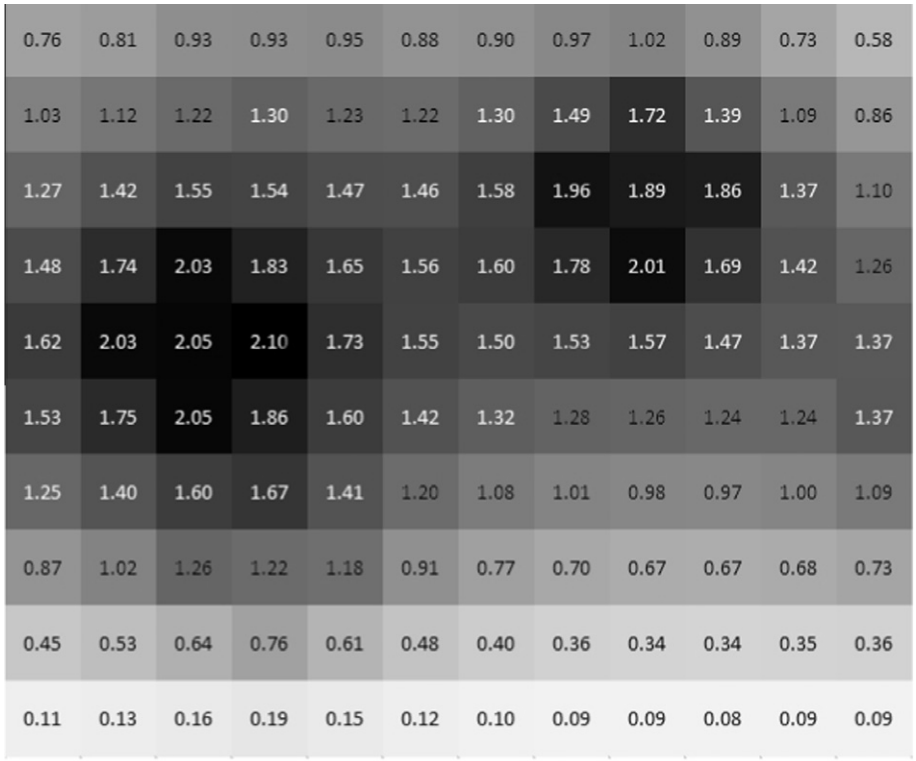

Fig. 8. Demand distribution for 10×12 grid.

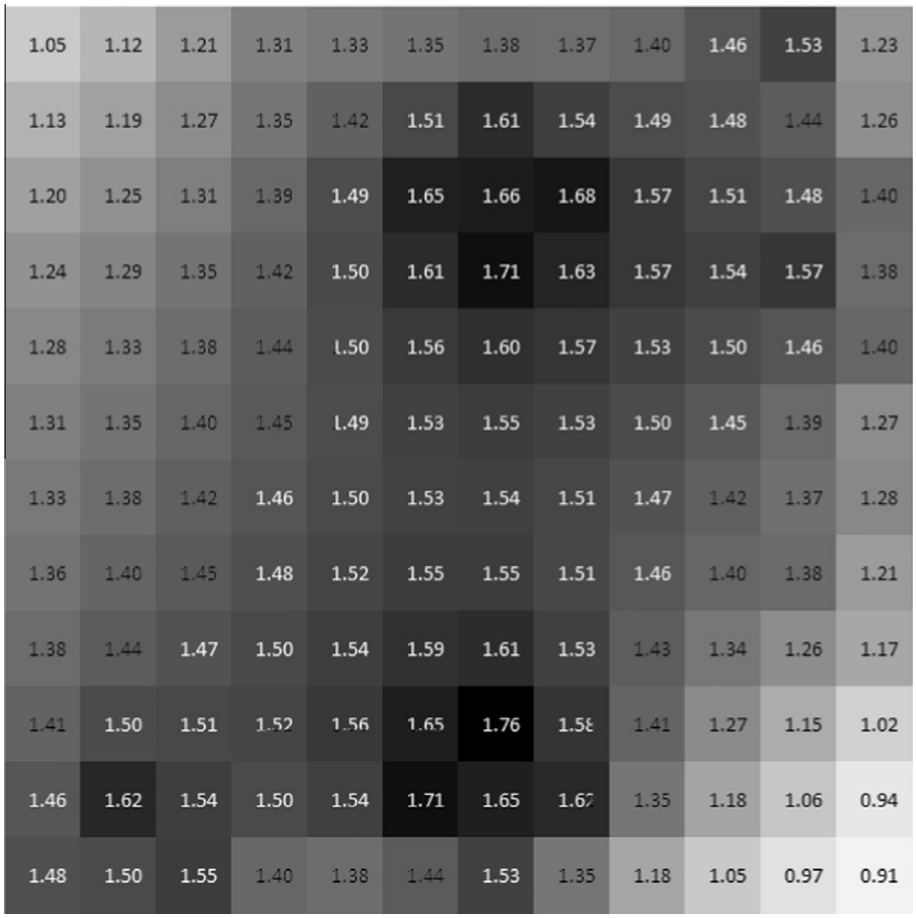

Fig. 9. Demand distribution for 12×12 grid.

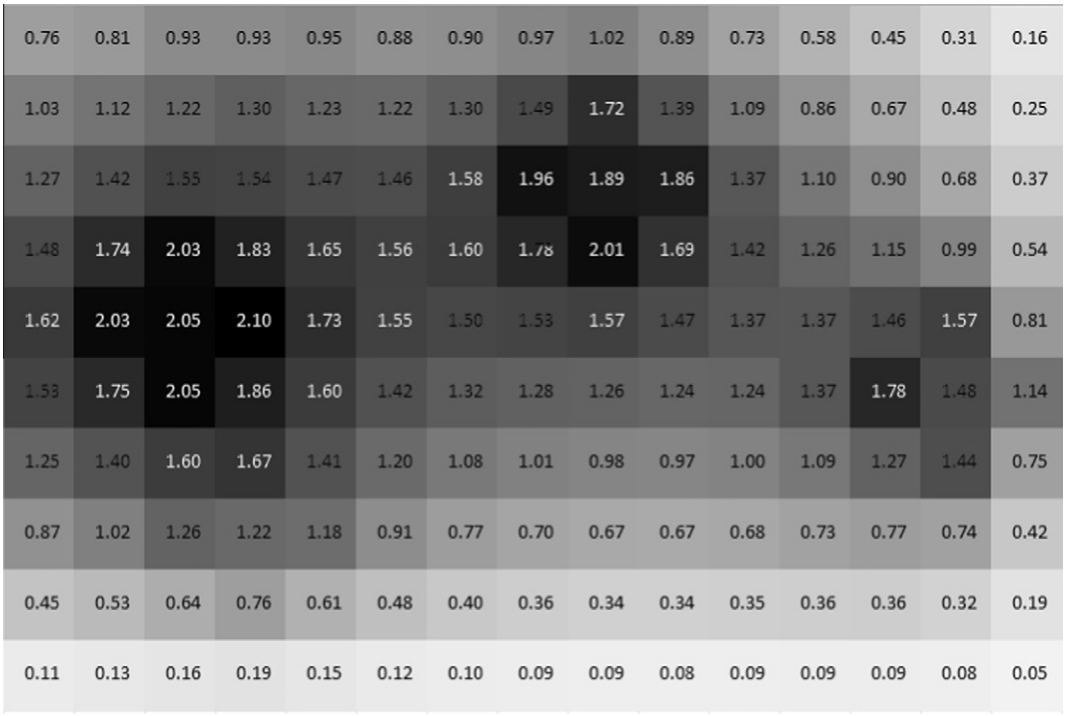

Fig. 10. Demand distribution for 10×15 grid.

relaxations than other instances when we add or take away a light source, and/or algorithms used by CPLEX's internal branch-and-bound procedures might be better suited to some of those specific cases. However, in the larger test cases on the 10×20 grid and 15×15 grid, the underlying complexity becomes larger and stable enough that those minor variations and peculiarities in the problem are not enough to significantly impact the CPU time, so we can observe a more well-defined increase in solution times as the

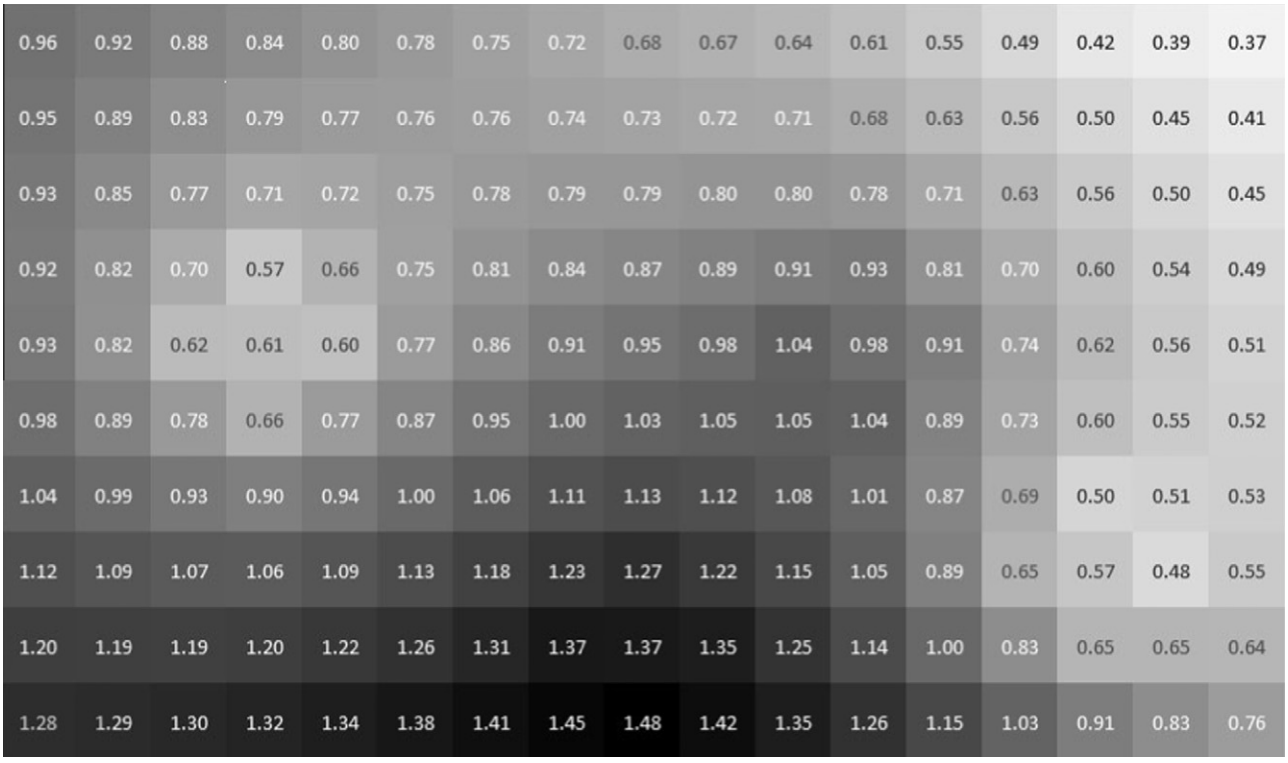

Fig. 11. Demand distribution for 10×17 grid.

0.35	0.61	0.71	0.82	0.76	0.81	0.93	0.93	0.95	0.88	0.90	0.97	1.02	0.89	0.73	0.58	0.45	0.31	0.16	0.04
0.44	0.69	0.94	0.98	1.03	1.12	1.22	1.30	1.23	1.22	1.30	1.49	1.72	1.39	1.09	0.86	0.67	0.48	0.25	0.06
0.50	0.76	0.99	1.13	1.27	1.42	1.55	1.54	1.47	1.46	1.58	1.96	1.89	1.86	1.37	1.10	0.90	0.68	0.37	0.09
0.56	0.86	1.13	1.30	1.48	1.74	2.03	1.83	1.65	1.56	1.60	1.78	2.01	1.69	1.42	1.26	1.15	0.99	0.54	0.14
0.61	0.98	1.36	1.45	1.62	2.03	2.05	2.10	1.73	1.55	1.50	1.53	1.57	1.47	1.37	1.37	1.46	1.57	0.81	0.20
0.62	1.08	1.28	1.51	1.53	1.75	2.05	1.86	1.60	1.42	1.32	1.28	1.26	1.24	1.24	1.37	1.78	1.48	1.14	0.29
0.50	0.81	1.15	1.17	1.25	1.40	1.60	1.67	1.41	1.20	1.08	1.01	0.98	0.97	1.00	1.09	1.27	1.44	0.75	0.19
0.34	0.53	0.70	0.78	0.87	1.02	1.26	1.22	1.18	0.91	0.77	0.70	0.67	0.67	0.68	0.73	0.77	0.74	0.42	0.10
0.17	0.26	0.34	0.39	0.45	0.53	0.64	0.76	0.61	0.48	0.40	0.36	0.34	0.34	0.35	0.36	0.36	0.32	0.19	0.05
0.04	0.06	0.08	0.10	0.11	0.13	0.16	0.19	0.15	0.12	0.10	0.09	0.09	0.08	0.09	0.09	0.09	0.08	0.05	0.00

Fig. 12. Demand distribution for 10×20 grid.

0.51	0.51	0.54	0.62	0.74	0.88	0.78	0.70	0.65	0.64	0.66	0.69	0.69	0.58	0.43
0.72	0.72	0.77	0.88	1.10	1.12	1.16	1.00	0.94	0.94	0.98	1.07	1.17	0.91	0.64
0.88	0.89	0.93	1.02	1.17	1.33	1.25	1.19	1.17	1.18	1.25	1.45	1.35	1.24	0.83
1.01	1.01	1.05	1.12	1.21	1.31	1.33	1.35	1.38	1.37	1.40	1.46	1.53	1.23	0.90
1.09	1.10	1.13	1.19	1.27	1.35	1.42	1.51	1.61	1.54	1.49	1.48	1.44	1.26	0.96
1.15	1.16	1.20	1.25	1.31	1.39	1.49	1.65	1.66	1.68	1.57	1.51	1.48	1.40	1.05
1.19	1.21	1.24	1.29	1.35	1.42	1.50	1.61	1.71	1.63	1.57	1.54	1.57	1.38	1.16
1.22	1.24	1.28	1.33	1.38	1.44	1.50	1.56	1.60	1.57	1.53	1.50	1.46	1.40	1.07
1.24	1.27	1.31	1.35	1.40	1.45	1.49	1.53	1.55	1.53	1.50	1.45	1.39	1.27	1.00
1.25	1.29	1.33	1.38	1.42	1.46	1.50	1.53	1.54	1.51	1.47	1.42	1.37	1.28	0.99
1.26	1.31	1.36	1.40	1.45	1.48	1.52	1.55	1.55	1.51	1.46	1.40	1.38	1.21	1.01
1.26	1.32	1.38	1.44	1.47	1.50	1.54	1.59	1.61	1.53	1.43	1.34	1.26	1.17	0.90
1.25	1.32	1.41	1.50	1.51	1.52	1.56	1.65	1.76	1.58	1.41	1.27	1.15	1.02	0.82
1.21	1.31	1.46	1.62	1.54	1.50	1.54	1.71	1.65	1.62	1.35	1.18	1.06	0.94	0.79
1.14	1.25	1.48	1.50	1.55	1.40	1.38	1.44	1.53	1.35	1.18	1.05	0.97	0.91	0.84

Fig. 13. Demand distribution for 15×15 grid.

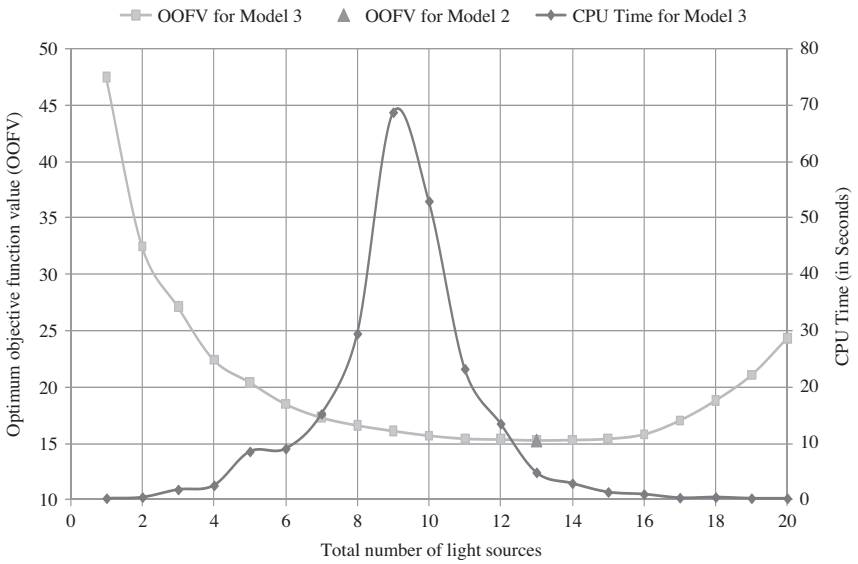

Fig. 14. Variation of objective function value and CPU time for 10×10 grid.

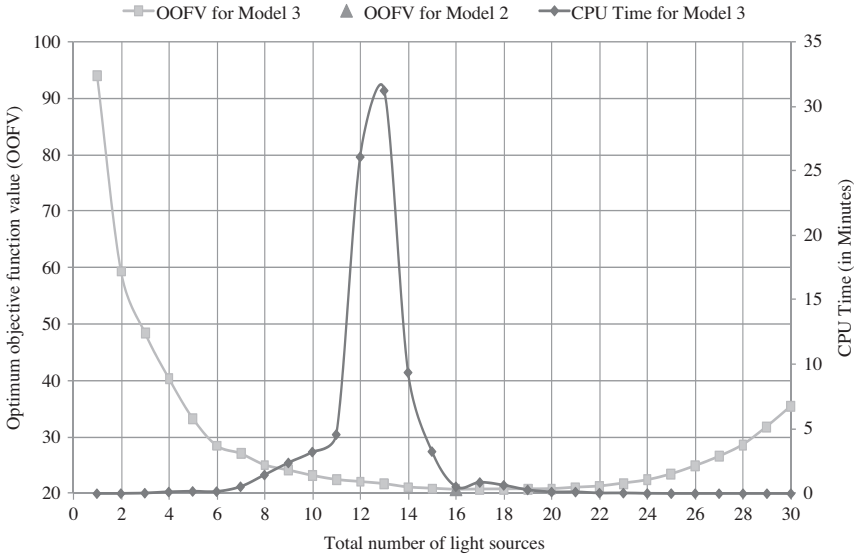

Fig. 15. Variation of objective function value and CPU time for 10×12 grid.

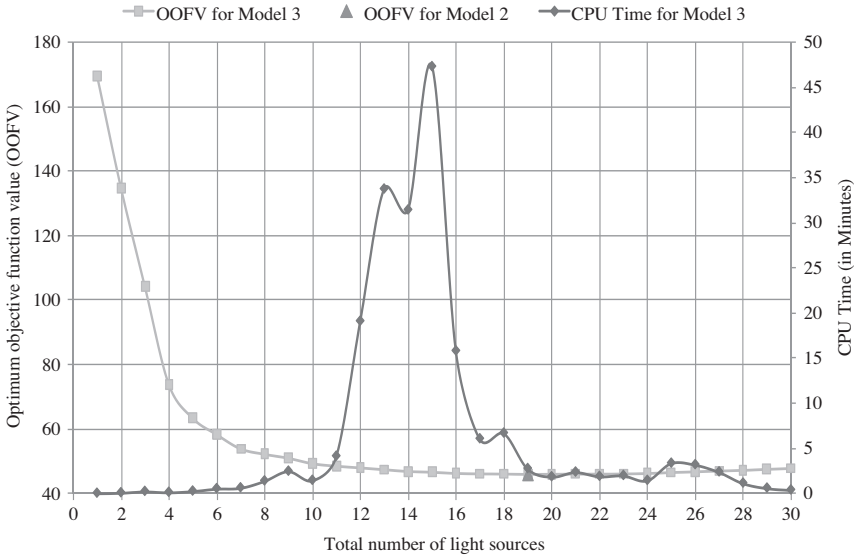

Fig. 16. Variation of objective function value and CPU time for 12×12 grid.

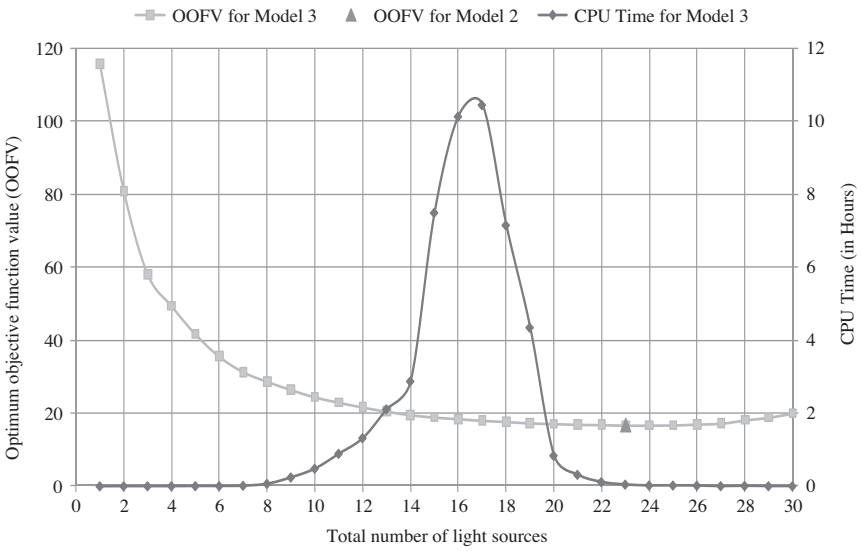

Fig. 17. Variation of objective function value and CPU time for 10×15 grid.

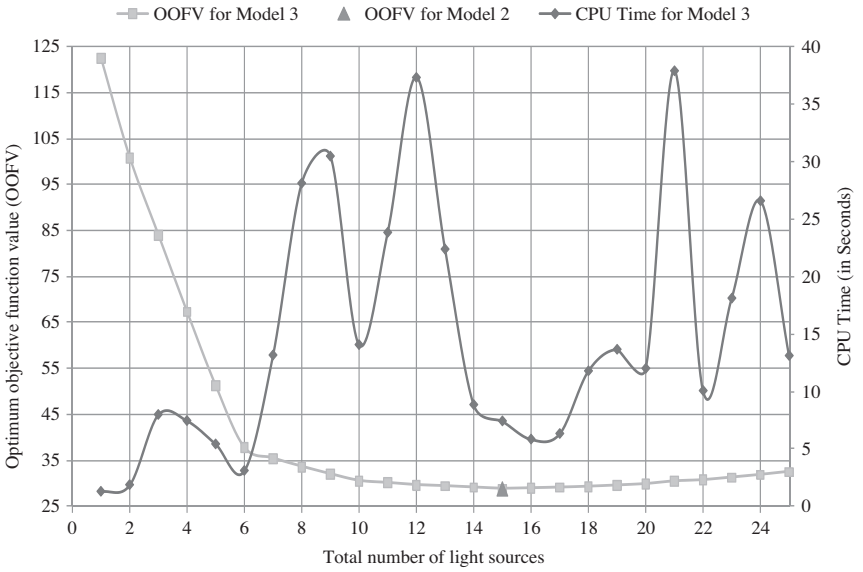

Fig. 18. Variation of objective function value and CPU time for 10×17 grid.

problem becomes more complex (i.e., we add more light sources, increasing n_a, and therefore increasing the number of constraints in the problem).

In Model 3, we can observe that the optimum number of light sources (n_{opt}) is 13 for the 10×10 grid, 16 for the 10×12 grid, 19 for the 12×12 grid, 23 for the 10×15 grid and 15 for the 10×17 grid, and the OOFV of these solutions correspond to the optimal solutions using Model 2 on those same grids. We can also observe that, at least for the 10×10, 10×12, 12×12, 10×15 and 10×17 grids, objective function values initially decrease as we increase the number of light sources. This continues until we reach the Model 2 optimal solution (i.e., an optimal number of light sources), after which objective function values increase with increasing values of n_a.

While the Model 3 solutions were generally obtainable for the 10×20 grid and 15×15 grid with smaller values of n_a (the number of light sources), we could not obtain solutions for Model 2 on the 10×20 and 15×15 grids with any reasonable optimality gap after over many days of solution time. Similarly, as we increase the number of light sources, Model 3 problems become increasingly difficult to solve in these two grids. With $n_a \leqslant 10$, solution runtimes are exceedingly fast, just seconds or minutes. However, when we set $n_a = 11$ for 10×20 grid and $n_a = 14$ for 15×15 grid, runtime increases to approximately one day, growing in an exponential-like fashion thereafter with increasing values of n_a, with $n_a = 13$ taking approximately 2 weeks to solve the 10×20 grid and with $n_a = 16$ taking 19 days to solve 15×15 grid. Solutions with $n_a \geqslant 14$ for 10×20 grid and $n_a \geqslant 16$ for 15×15 grid were not obtainable in reasonable time, even very sub-optimal solutions. In fact, we do not even reach an overall optimal number of light sources, as objective function values are still decreasing with increasing values of n_a.

Table 1 summarizes the runtime statistics for our various test case grids using Model 3 (showing only the instances with n_a equivalent to the respective optimal number of light sources). As discussed above, CPLEX was able to solve most test cases within a

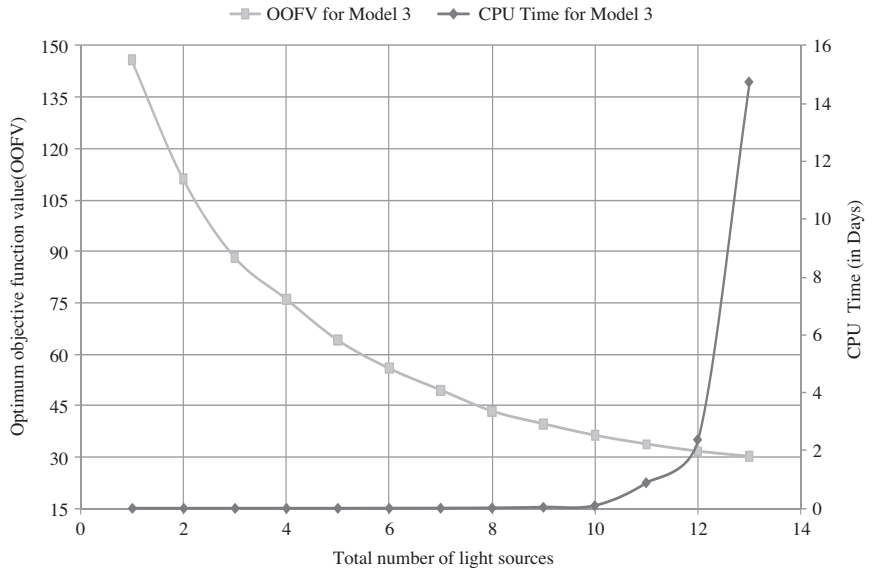

Fig. 19. Variation of objective function value and CPU time for 10×20 grid.

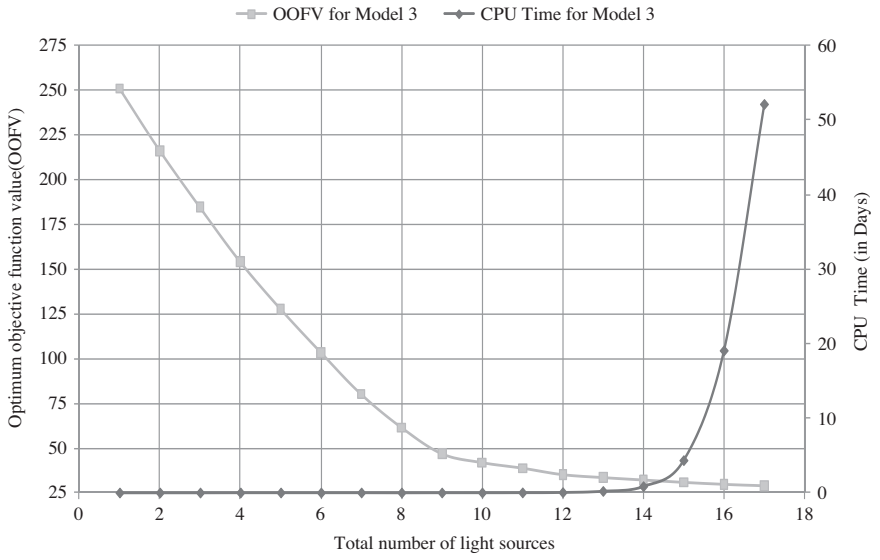

Fig. 20. Variation of objective function value and CPU time for 15×15 grid.

Table 1
Runtime statistics for selected Model 3 test cases.

Grid size	OOFV	# of lights	Simplex iterations	Branch and bound nodes	CPU time (seconds)	MIP gap	LP gap
10×10	15.28	13	87,171	5941	2.71875	0.000616	0.0903
10×12	20.63	16	1,792,730	92,705	47.3438	0.000989	0.0785
10×15	16.64	23	4,182,388	216,631	166.594	0.000996	0.1532
10×17	28.81	15	93,506	4965	4.0625	0.000810	0.0538
10×20	30.13	13[a]	2,147,483,648	72,440,018	1,033,440	0.000999	0.4191
12×12	45.99	19	3,893,188	185,741	107.922	0.000999	0.0239
15×15	28.94	17[a]	2,147,483,648	1,369,697,304	4,491,820	0.001	0.2698

[a] We were unable to solve for larger numbers of lights in these grids.

reasonable amount of time (several minutes or less) and with a reasonable number of Simplex iterations and branch-and-bound nodes. In addition, our proposed model solves moderately sized instances with optimality gaps ranging from 0.06% to 0.1%. However, the model becomes computationally difficult for test case instances with $n_a \geqslant 11$ in the 10×20 grid and $n_a \geqslant 14$ for the 15×15 grid, due in part to the greater LP gaps. Note that the LP gaps given in the table represent the difference between the optimal (or best found) integer solution and the fully relaxed version of the problem (i.e., the root of the branch-and-bound tree). To solve for

these larger instances of those grids, CPLEX had to explore quite a significant number of branch-and-bound nodes, involving a very large number of Simplex iterations, requiring weeks to reach optimality. In general for such instances, we find that the CPLEX solver's branch-and-bound procedure makes considerable progress early on, with rapid improvements in the objective function values of the best-to-date branch-and-bound nodes. In most of the problems tested, optimality gaps are reduced to 0.1% or less in just a few seconds or minutes. However, reductions in optimality gap are slow for large problems, and even after many days of runtime, higher optimality gaps remain or the solver runs out of memory. In future extensions of this work, our target is to develop techniques to solve even these large test case instances efficiently. For now, however, we were able to obtain optimal solutions for $n_a = 13$ in the 10×20 grid and $n_a = 17$ in the 15×15 grid, and so we provide the data corresponding to those instances of the problem in the table.

6. Conclusion

This paper proposes three GBLP ILP models to optimally place light posts in a park. Our ILP models are designed to optimize the number of light posts, their locations, and their sizes. While this particular problem represents just one specific GBLP, in reality, many problems can be modelled as GBLPs, and thus, can be solved using these methods. For example, we could use these methods to determine where to optimally place retail outlets and/or warehouses; a retail outlet will supply an area similar and meet the surrounding cell's demand. Retail demands can be established through market research or surveys in the surrounding neighbourhoods and the finite difference method can be applied to create an approximate demand distribution like those we used above. Other topics where our models can potentially be applied include health/biological sciences (e.g., optimal application of radiation), communications (e.g., transmitter locations), real estate, emergency service dispatching, physics, and resource exploration/exploitation.

Our results demonstrate that our ILP models can be used to solve GBLPs, and that they are scalable at least up to intermediate sized problems. However, for larger problem, it takes days and even weeks to solve to optimality. In the future, we plan to extend this work to develop advanced optimization techniques to solve large-scale problems using relaxation-based decomposition along with the addition of logical restrictions. Furthermore, we have also developed an extension to the models where we include a fixed-charge component to the objective function (and the associated constraint equations) to include light source installation costs and other such features to our models.

Acknowledgements

The authors thank the Killam Trusts and NSERC for their financial support in this research. The authors are indebted to the editor and anonymous reviewers for their constructive comments to improve the overall quality of this paper.

References

Abdinnour-Helm, S., Venkataramanan, M.A., 1998. Solution approaches to hub location problems. Annals of Operations Research 78, 31–50.

Altinel, I.K., Durmaz, E., Aras, N., Özkisacik, K.C., 2009. A location–allocation heuristic for the capacitated multi-facility Weber problem with probabilistic customer locations. European Journal of Operational Research 198, 790–799.

Aytug, H., Saydam, C., 2002. Solving large-scale maximum expected covering location problems by genetic algorithms: a comparative study. European Journal of Operational Research 141, 480–494.

Bangerth, W., Klie, H., Wheeler, M.F., Stoffa, P.L., Sen, M.K., 2006. On optimization algorithms for the reservoir oil well placement problem. Computational Geosciences 10, 303–319.

Berman, O., Drezner, Z., Krass, D., 2010. Generalized coverage: new developments in covering location models. Computers & Operations Research 37, 1675–1687.

Canbolat, M.S., Wesolowsky, G.O., 2010. The rectilinear distance Weber problem in the presence of a probabilistic line barrier. European Journal of Operational Research 202, 114–121.

Chapra, S.C., Canale, R.P., 2002. Numerical Methods for Engineers. McGraw-Hill, New York.

Chen, D.-S., Baton, R.G., Dang, Y., 2010. Applied integer programming. A John Willey & Sons, Inc., Publication (Chapter 2).

Chen, G., Ozturk, O., Kandemir, M., 2005. An ILP-based approach to locality optimization. Lecture Notes in Computer Science 3602, 149–163.

Church, R.L., ReVelle, C.S., 1974. The maximal covering location problem. Papers of the Regional Science Association 32, 101–118.

Cooper, L., 1963. Location–allocation problems. Operations Research 11, 331–343.

Daskin, M.S., 1995. Network and Discrete Location: Models, Algorithms, and Applications. John Wiley, New York.

Domschke, W., Krispin, G., 1997. Location and layout planning – a survey. OR Spektrum 19, 181–194.

Drezner, Z., Wesolowsky, G.O., 1997. On the best location of signal detectors. IIE Transactions 29, 1007–1015.

Fourer, R., Gay, D.M., Kernighan, B.W., 2002. AMPL: A Modeling Language for Mathematical Programming. Duxbury Press, Cole Publishing Co.

Francis, R.L., 1964. On the location of multiple new facilities with respect to existing facilities. Journal of Industrial Engineering 15, 106–107.

Gendron, B., Potvin, J.-Y., Soriano, P., 2003. A Tabu Search with slope scaling for the multi commodity capacitated location problem with balancing requirements. Annals of Operations Research 122, 193–217.

Ghiani, G., Grandinetti, L., Guerriero, F., Musmanno, R., 2002. A Lagrangean heuristic for the plant location problem with multiple facilities in the same site. Optimization Methods and Software 17, 1059–1076.

ILOG, 2007. ILOG CPLEX 11.0 User's Manual, ILOG, Inc.

Ingolfsson, A., Budge, S., Erkut, E., 2008. Optimal ambulance location with random delays and travel times. Health Care Management Science 11, 262–274.

Katz, I.N., Cooper, L., 1974. An always convergent numerical scheme for a random locational equilibrium problem. SIAM Journal of Numerical Analysis 11, 683–692.

Kubis, A., Hartmann, M., 2007. Analysis of location of large-area shopping centers – a probabilistic gravity model for the Halle–Leipzig area. Jahrbuch für Regionalwissenschaft 27, 43–57.

Manzour-al-Ajdad, S.M.H., Torabi, S.A., Eshghi, K., 2012. Single-source capacitated multi-facility weber problem – an iterative two phase heuristic algorithm. Computers & Operations Research 39, 1465–1476.

Marín, A., 2011. The discrete facility location problem with balanced allocation of customers. European Journal of Operational Research 210, 27–38.

NASA, 2006. More on Brightness as a Function of Distance. <http://imagine.gsfc.nasa.gov/YBA/M31-velocity/1overR2-more.html> (05.02.10).

Sherali, H.D., Nordai, F.L., 1988. NP-hard, capacitated balanced p-median problems on a chain graph with a continuum of link demands. Mathematics of Operations Research 13, 32–49.

Simons, R.H., Bean, A.R., 2001. Lighting Engineering: Applied Calculations. Architectural Press.

Taniguchi, E., Noritake, M., Yamada, T., Izumitani, T., 1999. Optimal size and location planning of public logistics terminals. Transportation Research Part E: Logistics and Transportation Review 35, 207–222.

Teixeira, J.C., Antunes, A.P., 2008. A hierarchical location model for public facility planning. European Journal of Operational Research 185, 92–104.

Wesolowsky, G.O., 1972. Rectangular distance location under the minimax optimality criterion. Transportation Science 6, 103–113.

Winston, W.L., Venkataramanan, M., 2003. Introduction to Mathematical Programming, fourth ed. Thomson (Chapter 9).

Wolsey, L.A., 1998. Integer Programming. Series in Discrete Mathematics and Optimization. Wiley-Interscience, Toronto (Chapter 1).

Zhang, J., 2006. Approximating the two-level facility location problem via a quasi-greedy approach. Mathematical Programming, Series A 108, 159–176.