Optimization of Desalination Location Problem Using MILP

A. M. Emhamed and B. Czuczai
Dept. of Chemical Engineering, Budapest University of Technology and Economics,
H-1111, Budapest, Muegyetem rkp. 3, Hungary

L. Horvath
Research Laboratory of Material and Environmental Sciences,
Chemical Research Center of HAS, H-1525 Budapest, Hungary

E. Rev and Z. Lelkes
Dept. of Chemical Engineering, Budapest University of Technology and Economics,
H-1111, Budapest, Muegyetem rkp. 3, Hungary

DOI 10.1002/aic.11255
Published online July 31, 2007 in Wiley InterScience (www.interscience.wiley.com).

A new mixed-integer linear programming model for location problem is developed in this work in order to find the optimal co-ordinates of the desalination plants. The model takes into account the given locations and capacities of the water incomes, the demands, and the costs of plants and pipelining. Feasible and infeasible plant regions are distinguished for locating the plants. The model has been developed in two consecutive phases. First, a basic model is developed that provides a solution within short time but does not take into account the possibility of pipeline branching. Application of this model gives rise to redundant pipelines to some connections, involving extra costs. Pipeline branching is dealt with an improved model developed in the second phase. This improved model provides realistic solution but with much longer computation time. The results of applying the different models on motivated examples of different sizes are detailed. © 2007 American Institute of Chemical Engineers *AIChE J,* 53: 2367–2383, 2007
Keywords: location, distance, pipeline, feasible region, MILP

Introduction

A shortage of potable water in many populated areas around the world where 2 billion people go without fresh water on a daily basis has already reached the point at which supply of seawater is the only solution.

Seawater, however, must be desalinated before use. Desalination is a process that removes minerals (not limited to salt) from seawater, brackish water, or treated waste water. The best known desalination processes are multi-stage flash, multi-effect distillation, vapor compression, solar distillation (SD), reverse osmosis (RO), and electrodialysis. RO and SD are considered in this work. In RO, feed water is pumped at high pressure through permeable membranes, separating salts from the water. The quality of the produced water depends on the applied pressure, the concentration of salts in the feed water, and the salt permeation constant of the membranes.[1] SD is a clean and environmentally safe technology for fresh water production. The simplest and most practical solar still is of a single-basin type, which consists of a box with a tilted

Correspondence concerning this article should be addressed to Z. Lelkes at lelkes@mail.bme.hu.

or curved transparent glazing that serves also as a condensation surface, and trough(s) to collect the distillate.[2]

Finding the optimal location of the desalination plants is an important problem on the North-African coasts.[3]

Facility location problems are frequently studied in the continuous and discrete optimization literature. The essential elements of such a problem are facilities to be located on a planar ground, and customers with specified demands and locations on the same ground. The goal is to find the optimal position of the facilities such that the objective function, generally a cost function depending on the sum of the distances between the facilities and the customers, is minimized. The location problems are different from layout problems where the facilities have considerable size when compared to the plane. In the location problems, the size of the facilities can be neglected when compared to the plane. A useful survey related to location problems is available.[4]

The area on which the facilities will be sited can be restricted. The (in)feasible region can be convex or nonconvex.[5,6]

The facility location problems are usually solved with heuristics. The problem is NP-hard, hence exact algorithms can solve problems of moderate size only, and they rarely occur in the literature.[7,8]

The problem presented in this work, that is, finding the optimal location of desalination plants and assigning the connection from water intakes to plants and from plants to customers, has not been addressed in the literature so far.

As it is detailed later in the section "Problem statement," the seawater is transported by pipelines from water intakes to plants and, the desalinated water is also transported by pipelines from plants to customers. In our problem, the minimization of the investment cost arising from the pipelines is the main issue; that constitutes the main difficulty. The pipeline is needed only if it transports water. If there is no water transportation between two particular objects, then there is no need for the pipeline between them to exist. In other words, our problem is a location–allocation problem, where logic (or binary) variables are needed to represent the existence of connections between objects. If the pipeline exists, then the investment cost—calculated from its length, which is a variable—will be considered in the objective function. If the pipeline does not exist then its investment cost is 0.

No similar problems possessing all of these features are found in the literature. Some location problems existing in the literature[9] (Capacitated Facility Location Problem or CFLP) do not capture the substance of our problem, the fix connections between the facilities and the customers. They are only trade-off problems between the variable and fixed cost of the production, and they do not consider the variable distances.

Most of the previous works[6,9] deal with problems to locate facilities in the plane so that the sum of the distances measured between the facilities and the customers is at minimum. The main difference between these and our problem is that we consider the pipelining cost in that case only when the connection between the facility and customer really exists. The objective function of the literature problems is the sum of all the distances. It follows that no binary variables to represent the existence of distinct pipelines are needed in their models. They can be solved with linear programming.

Some other location problems[7,9] are more similar to ours. These models already consider the distinct connections, but the possible locations of the facilities are discrete and known, so the distances between the facilities and the customers are also known in advance as parameters.

The problem most reminiscent to ours is the theoretical multi-Weber problem where the task is to partition the full set of fixed points into subsets and finding several minimizing points (one per set) simultaneously. This problem is mainly solved by heuristics. However, Rosing[10] showed that the objective function of the problem is very steep near the optimum. Therefore, it is essential to apply an exact solution method; otherwise, the results would be very far from the optimum.

The only exact method is worked out by Rosing,[11] but his problem and model are still very different from ours and the technique applied by him is not suitable to solve our problem because of the aforementioned differences. Additionally, his solution method is based on the possibility of encircling the (customers') locations by disjunctive convex hulls each of which includes the (plant's) location serving the points in the actual hull only. Since the plant capacities are finite, and there are forbidden zones, the optimal solution may, and in practice will, contain pipeline connections from plants to customers in such a way that the above convex hull assumption is not satisfied.

Since no similar problems and methods applicable to our problem were found in the literature, we had to work out a new model to solve the problem. This new model and the obtained results are detailed in the present work.

Beside mixed-integer linear programming (MILP), it is also possible to apply other techniques like constraint logic programming (CLP) and generalized disjunctive programming (GDP).[12] In the present work, however, we investigate the applicability and performance of the MILP only. Although applying MILP requires longer computation time than using merely heuristics, it is an exact method and guarantees optimal solution. In addition, the process of model building helps us in understanding thoroughly the characteristics and the main difficulties of the problem.

Problem Statement and Test Problems

The desalination plant location problem presented here involves a given number of cities each of which contains at least one storage tank to receive the imported potable water. The potable water will be produced in different kinds of desalination plants, and from seawater, which is delivered to the plants by pipelines from seawater intakes located at the sea coast. The locations of the seawater intakes and the potable water storage tanks are known. The demands for water at each storage location (at each city) are also known. The storage tanks, the desalination plants, and the sea water intakes are considered punctually, i.e. their extension on the plane is not considered. The problem also includes forbidden zones such as cities, sea surface, and some other places at which desalination plants are not allowed to exist. An example of the location problem plane is depicted in Figure 1. The total cost of the problem consists of the fixed costs of the plants and pipelines, and the variable costs of the production and transportation. Our task is to find the optimal number and

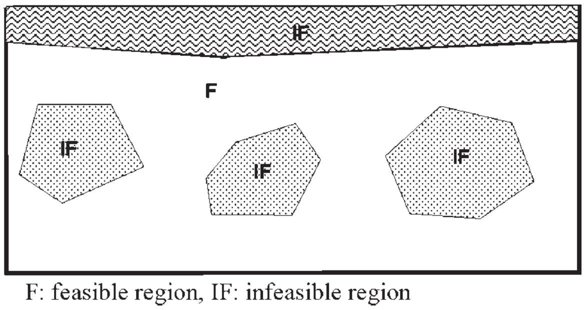

Figure 1. The location problem plane.

coordinates of the plants, such that the total cost is at minimum.

Two example problems have been worked out for testing the capability of the developed models.

Problem I

A 20 km × 10 km rectangle contains five storage tanks, and a part of the sea. The complementary area can be subdivided into at least 12 triangles. The locations of two seawater intakes and five fresh water storage tanks are given. At most eight SD plants with capacity of 10 m^3/day, three RO plants with capacity of 250 m^3/day, and at most three RO plants with capacity of 500 m^3/day may be used. Detailed data of the Problem I are shown in Tables A1–A4 in Appendix.

Problem II

A 50 km × 30 km rectangle contains six cities and a part of the sea. The complementary area can be subdivided into at least 23 triangles. The locations of two seawater intakes and six fresh water storage tanks are given. At most eight SD plants with capacity of 10 m^3/day, three RO plants with capacity of 250 m^3/day, and at most three RO plants with capacity of 500 m^3/day may be used. Detailed data of the Problem II are shown in Tables A16–A19 in Appendix.

Model Versions and Development Strategy

MILP models have been developed to determine the optimal location of the desalination plants, the amount of seawater transported from the water intakes to the plants, and the amount of potable water transported from the plants to the storage tanks.

The model is developed in three consecutive steps:

1. Superstructure development
2. GDP model development
3. Mathematical programming (MP) model development

A superstructure is a graph or network that contains all the considered structures in a combined form. The actual structures can be derived by deleting some elements of the superstructure. The superstructure is developed in order to visualize and concretize the frame within which solution structures are looked for.

Developing a MP model directly based on the superstructure is a rather demanding task. The model developer should take into account several viewpoints including feasibility of the structures and the models, and relations between the imagined structures and their algebraic models. Existence of structure elements are modeled with binary variables in an algebraic model. Validity of some continuous relations (e.g. material balances) depends on such existence. The relations referring to the existence and coexistence of the structure elements are complicated expressions of mixed binary and continuous variables.

GDP models[12] are much more similar to the human approach than an MP model because GDP applies logic variables for describing existence and coexistence of conditional entities in the superstructure. Therefore, instead of directly developing an MP model from the superstructure, it is much easier to develop a GDP model first. The logic relations of the GDP model are then transformed to binary and mixed binary relations with well known techniques like Big-M or Convex Hull.[13] In this way, a basic MP model is obtained that can be improved in subsequent development steps.

The structure of this work is the following: first the superstructure is presented, and then the MILP model is detailed. For the sake of intelligibility, the logic expressions and relations are explained first in their GDP form, and then their transformations to algebraic form are detailed.

Where it was possible, we applied convex hull, since it results in better relaxation than the Big-M technique. In case of some expressions, however, the Big-M technique was the only solution.

Finally, the additional constraints, which are used merely for improving the performance and accelerating the solution, are presented in their algebraic form only.

The Basic Model

Superstructure

The superstructure includes three kinds (p) of desalination plants (u_p), seawater intakes (w), and fresh water storage tanks (s). The superstructure is based on R-graph representation,[14] and is depicted in Figure 2. The units and the pipelines are all conditional elements of the superstructure.

Model formulation

The model contains material balances, logic expressions on the existence of the plants and the pipelines, and constraints restricting the plant locations to the feasible region. To decrease the computation time, additional constraints are also introduced, see subsection "model variants."

Material Balance Constraints. The material balances between the water intakes and the plants, and also between

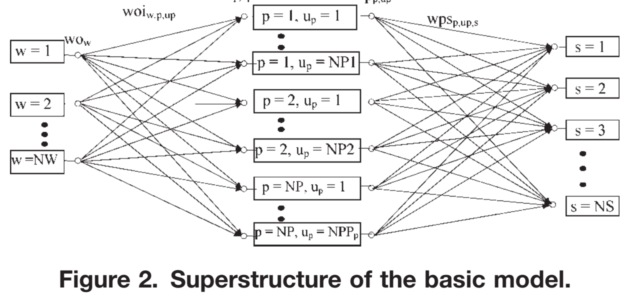

Figure 2. Superstructure of the basic model.

the plants and the storage tanks, must be satisfied. The following indices are used in this part of the model: w for seawater intakes, s for potable water storage tanks, p for the type of desalination plants (three types are used in our case), u_p for the units of type p of desalination plant, and k for the x–y coordinates. The material balance constraints describe the requirement of satisfying the conservation of material between the objects; between water intakes and plants (Eqs. 1 and 2), between plants and storage tanks (Eqs. 4 and 5), and the equation describing the efficiency of the plants (Eq. 3). The equations are given below, as follows.

$$\mathrm{wo_w} = \sum_{\mathrm{p}} \sum_{\mathrm{u_p}} \mathrm{woi_{w,p,u_p}} \qquad w \in W \tag{1}$$

$$\mathrm{wi_{p,u_p}} = \sum_{\mathrm{w}} \mathrm{woi_{w,p,u_p}} \qquad p \in \mathrm{PT}, u_\mathrm{p} \in U_\mathrm{p} \tag{2}$$

$$\mathrm{wp_{p,u_p}} = E_\mathrm{p} \cdot \mathrm{wi_{p,u_p}} \qquad p \in \mathrm{PT}, u_\mathrm{p} \in U_\mathrm{p} \tag{3}$$

$$\mathrm{wp_{p,u_p}} = \sum_{\mathrm{s}} \mathrm{wps_{p,u_p,s}} \qquad p \in \mathrm{PT}, u_\mathrm{p} \in U_\mathrm{p} \tag{4}$$

$$\mathrm{ws_s} = \sum_{\mathrm{p}} \sum_{\mathrm{u_p}} \mathrm{wps_{p,u_p,s}} \qquad s \in S \tag{5}$$

Distances. The transportation costs are proportional to the length of each pipeline; these in turn depend on the distances between sea water intakes and plants, and between plants and storage tanks. Distances are defined according to Manhattan metric.[15] Manhattan distance is the expression $D_{12}^{\mathrm{m}} = |x_2 - x_1| + |y_2 - y_1|$ used for approximating the nonlinear Euclidian distance $D_{12}^{\mathrm{eu}} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ of two points (x_1, y_1) and (x_2, y_2). Although the absolute-value function is not linear, it can be modeled using two linear non-equalities $\mathrm{Dx}_{12}^{\mathrm{m}} \geq x_2 - x_1$ and $\mathrm{Dx}_{12}^{\mathrm{m}} \geq x_1 - x_2$ if $\mathrm{Dx}_{12}^{\mathrm{m}}$ is minimized in the objective function.

Equations computing the distance between seawater intake and any plant are:

$$\mathrm{Dw_{w,p,u_p,k}} \geq (\mathrm{pw_{w,k}} - \mathrm{pp_{p,u_p,k}}) \qquad p \in \mathrm{PT}, u_\mathrm{p} \in U_\mathrm{p}, w \in W, k \in K \tag{6}$$

$$\mathrm{Dw_{w,p,u_p,k}} \geq (\mathrm{pp_{p,u_p,k}} - \mathrm{pw_{w,k}}) \qquad p \in \mathrm{PT}, u_\mathrm{p} \in U_\mathrm{p}, w \in W, k \in K \tag{7}$$

Equations computing the distance between the plants and storage tanks are:

$$\mathrm{Ds_{p,u_p,s,k}} \geq (\mathrm{pp_{p,u_p,k}} - \mathrm{ps_{s,k}}) \quad p \in \mathrm{PT}, u_\mathrm{p} \in U_\mathrm{p}, s \in S, k \in K \tag{8}$$

$$\mathrm{Ds_{p,u_p,s,k}} \geq (\mathrm{ps_{s,k}} - \mathrm{pp_{p,u_p,k}}) \quad p \in \mathrm{PT}, u_\mathrm{p} \in U_\mathrm{p}, s \in S, k \in K \tag{9}$$

Capacity constraints The amount of produced potable water is constrained by the capacity of the plant:

$$\mathrm{wp_{p,u_p}} \leq \mathrm{CAP_p} \qquad p \in \mathrm{PT}, u_\mathrm{p} \in U_\mathrm{p} \tag{10}$$

Logic Expressions. *Logic expression on the existence of the connection between seawater intake w and plant u_p of type p.* The amount of water transported through a non-existing pipeline must be 0. The fixed cost of an existing pipeline is calculated from its length:

$$\begin{bmatrix} \mathrm{Zwp_{w,p,u_p}} \\ 0 \leq \mathrm{woi_{w,p,u_p}} \leq \mathrm{CAPWP_{w,p}} \\ \mathrm{fcostwp_{w,p,u_p,k}} = \mathrm{fcp} \cdot \mathrm{Dw_{w,p,u_p,k}} \end{bmatrix} \vee \begin{bmatrix} \neg\mathrm{Zwp_{w,p,u_p}} \\ \mathrm{woi_{w,p,u_p}} = 0 \\ \mathrm{fcostwp_{w,p,u_p,k}} = 0 \end{bmatrix}$$
$$w \in W, p \in \mathrm{PT}, u_\mathrm{p} \in U_\mathrm{p}, k \in K \tag{11}$$

Logic expression on the existence of plants u_p of type p. The amount of water produced by a non-existing plant must be 0. The fixed cost of an existing plant is equal to its fixed cost parameter:

$$\begin{bmatrix} \mathrm{Zp_{p,u_p}} \\ 0 \leq \mathrm{wp_{p,u_p}} \\ \mathrm{fcostp_{p,u_p}} = \mathrm{fcup_p} \end{bmatrix} \vee \begin{bmatrix} \neg\mathrm{Zp_{p,u_p}} \\ \mathrm{wp_{p,u_p}} = 0 \\ \mathrm{fcostp_{p,u_p}} = 0 \end{bmatrix}$$
$$p \in \mathrm{PT}, u_\mathrm{p} \in U_\mathrm{p} \tag{12}$$

Logic expression on the existence of the connection between plant u_p of type p and storage tank s. The amount of water transported by a non-existing pipeline must be 0. The fixed cost of an existing pipeline is calculated from its length:

$$\begin{bmatrix} \mathrm{Zps_{p,u_p,s}} \\ 0 \leq \mathrm{wps_{p,u_p,s}} \leq \mathrm{CAPPS_{p,s}} \\ \mathrm{fcostps_{p,u_p,s,k}} = \mathrm{fcp} \cdot \mathrm{Ds_{p,u_p,s,k}} \end{bmatrix} \vee \begin{bmatrix} \neg\mathrm{Zps_{p,u_p,s}} \\ \mathrm{wps_{p,u_p,s}} = 0 \\ \mathrm{fcostps_{p,u_p,s,k}} = 0 \end{bmatrix}$$
$$p \in \mathrm{PT}, u_\mathrm{p} \in U_\mathrm{p}, s \in S, k \in K \tag{13}$$

Transformation of Logic Expressions. The above logic expressions are transformed to algebraic expressions using the Big-M technique. For example, Eq. 13 is reformulated as a system of equations below:

$$\mathrm{wps_{p,u_p,s}} \leq \mathrm{CAPPS_{p,s}} \cdot \mathrm{yps_{p,u_p,s}} \qquad p \in \mathrm{PT}, u_\mathrm{p} \in U_\mathrm{p}, s \in S \tag{14}$$

$$\mathrm{fcostps_{p,u_p,s,k}} - \mathrm{fcp} \cdot \mathrm{Ds_{p,u_p,s,k}} \leq \mathrm{UDS_{s,k}} \cdot (1 - \mathrm{yps_{p,u_p,s}}) \qquad p \in \mathrm{PT}, u_\mathrm{p} \in U_\mathrm{p}, s \in S, k \in K \tag{15}$$

$$-\mathrm{UDS_{s,k}} \cdot (1 - \mathrm{yps_{p,u_p,s}}) \leq \mathrm{fcostps_{p,u_p,s,k}} - \mathrm{fcp} \cdot \mathrm{Ds_{p,u_p,s,k}} \qquad p \in \mathrm{PT}, u_\mathrm{p} \in U_\mathrm{p}, s \in S, k \in K \tag{16}$$

Reformulation of the other logic expressions is performed with the same technique.

Assignment of Feasible and Infeasible Regions. The plane of the location problem is divided into two regions. Feasible region (F) is the area where the desalination plants may be located, and infeasible region (IF) or forbidden zone is the area where plants should not be located (e.g. sea and cities, see the problem statement), as depicted in Figure 1. The plants must be located in the feasible region. Two dis-

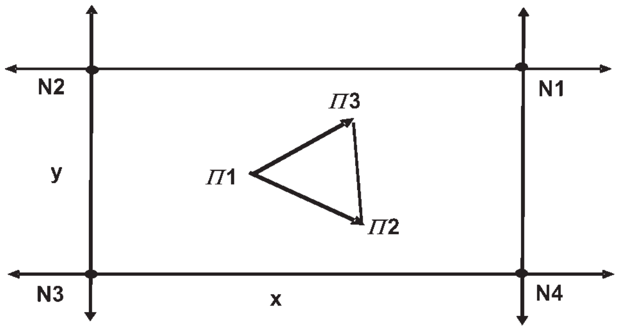

Figure 3. The triangle.

tinct methods are separately used to assign the feasible region.

Use of triangles. The whole feasible region is subdivided into disjunctive triangles denoted by index *j*. This subdivision is made in advance to any optimization. The number of triangles should be as small as possible because the number of binary variables used for modeling the problem strongly depends on it. How to find a subdivision with a minimum or at least a small number of triangles is not discussed here. The subdivision actually applied is considered as a constituting part of the problem statement.

Such a triangle is shown in Figure 3. The $\overline{\Pi 1 \Pi 2}$ and $\overline{\Pi 2 \Pi 3}$ directed edges of the triangle in Figure 3 are used as the basis vectors of an oblique frame of axes. In general, any point *P* inside the rectangular shown in Figure 3 can be represented by the following expression:

$$P_{\mathrm{k}} = \Pi 1_{\mathrm{k}} + \Lambda 21 \cdot (\Pi 2_{\mathrm{k}} - \Pi 1_{\mathrm{k}}) + \Lambda 31 \times (\Pi 3_{\mathrm{k}} - \Pi 1_{\mathrm{k}}) \quad (k \in K) \quad (17)$$

where Λ21 and Λ31 are coordinates of the oblique system. If Λ21 and Λ31 are non-negative and their sum is less than or equal to 1 then *P* is situated in the triangle constituted by Π1, Π2, and Π3. Why it is so is shortly explained in the Appendix.

We introduce a new logic variable $\mathrm{Zpj}_{\mathrm{p,u_{p,j}}}$ which represents the information whether a plant u_{p} of type *p* is located inside triangle *j* (i.e. $\mathrm{Zpj}_{\mathrm{p,u_{p,j}}}$ is true) or outside it (i.e. $\mathrm{Zpj}_{\mathrm{p,u_{p,j}}}$ is false). The following mixed logic expression is applied in the model for obtaining correct Λ21 and Λ31 values. If plant u_{p} of type *p* is located inside triangle *j* then Λ21 and Λ31 belonging to them are non-negative and their sum is less than or equal to 1:

$$\begin{bmatrix} \mathrm{Zpj}_{\mathrm{p,u_p,j}} \\ 0 \le \lambda 21_{\mathrm{p,u_p,j}} \\ 0 \le \lambda 31_{\mathrm{p,u_p,j}} \\ \lambda 21_{\mathrm{p,u_p,j}} + \lambda 31_{\mathrm{p,u_p,j}} \le 1 \end{bmatrix} \vee \begin{bmatrix} \neg \mathrm{Zpj}_{\mathrm{p,u_p,j}} \\ L21_{\mathrm{j}} \le \lambda 21_{\mathrm{p,u_p,j}} \le U21_{\mathrm{j}} \\ L31_{\mathrm{j}} \le \lambda 31_{\mathrm{p,u_p,j}} \le U31_{\mathrm{j}} \end{bmatrix} \quad p \in \mathrm{PT}, u_{\mathrm{p}} \in U_{\mathrm{p}}, j \in J \quad (18)$$

Equation 18 is reformulated in the following way to represent the plant locations with binary variables instead of logic ones:

$$\mathrm{pp}_{\mathrm{p,u_p,k}} = P1_{\mathrm{j,k}} + \lambda 21_{\mathrm{p,u_p,j}} \cdot (P2_{\mathrm{j,k}} - P1_{\mathrm{j,k}}) + \lambda 31_{\mathrm{p,u_p,j}} \times (P3_{\mathrm{j,k}} - P1_{\mathrm{j,k}}) \quad p \in \mathrm{PT}, u_{\mathrm{p}} \in U_{\mathrm{p}}, j \in J, k \in K \quad (19)$$

$$L21_{\mathrm{j}} \cdot (1 - \mathrm{ypj}_{\mathrm{p,u_p,j}}) \le \lambda 21_{p,u_p,j} \quad p \in \mathrm{PT}, u_{\mathrm{p}} \in U_{\mathrm{p}}, j \in J \quad (20)$$

$$L31_{\mathrm{j}} \cdot (1 - \mathrm{ypj}_{\mathrm{p,u_p,j}}) \le \lambda 31_{p,u_p,j} \quad p \in \mathrm{PT}, u_{\mathrm{p}} \in U_{\mathrm{p}}, j \in J \quad (21)$$

$$\lambda 21_{\mathrm{p,u_p,j}} + \lambda 31_{\mathrm{p,u_p,j}} - 1 \le (U21_{\mathrm{j}} + U31_{\mathrm{j}}) \cdot (1 - \mathrm{ypj}_{\mathrm{p,u_p,j}}) \quad p \in \mathrm{PT}, u_{\mathrm{p}} \in U_{\mathrm{p}}, j \in J \quad (22)$$

Because of the above equations, the lambdas are non-negative (Eqs. 20 and 21), and their sum is less than or equal to 1 (Eq. 22) if the binary variable ypj of the existence of the plant u_{p} of type *p* inside the triangle *j* equals to 1. Equation 23 expresses the requirement that plant u_{p} of type *p* must be contained by exactly one of the triangles.

$$\sum_{\mathrm{j}} \mathrm{ypj}_{\mathrm{p,u_p,j}} = 1 \quad p \in \mathrm{PT}, u_{\mathrm{p}} \in U_{\mathrm{p}} \quad (23)$$

Use of polygons. Any convex polygon can be drawn as an intersection of triangles. A particular point is contained by the polygon if and only if it is contained by each triangle enclosing the polygon. A sample of such a polygon is shown in Figure 4.

In this version, polygons (indexed by *i*) are used to assign the feasible region. Each nontriangle polygon can be assigned by two or more triangles. They must share the whole area of the polygon. The set of triangles (indexed by *j*) is sorted to subsets (each indexed by j_{i}) covering polygon *i*. The triangles belonging to different subsets may overlap. The binary variable $\mathrm{ypi}_{\mathrm{p,u_p}}$,i expresses the existence of plant u_{p} of type *p* within polygon *i*.

Equation 19 remains unchanged in this model. Equations 20–23 are substituted with Eqs. 24–27, as follows:

$$L21_{\mathrm{j_i}} \cdot (1 - \mathrm{ypi}_{\mathrm{p,u_p,i}}) \le \lambda 21_{\mathrm{p,u_p,j_i}} \quad i \in I, p \in \mathrm{PT}, u_{\mathrm{p}} \in U_{\mathrm{p}}, j_{\mathrm{i}} \in J_{\mathrm{i}} \quad (24)$$

$$L31_{\mathrm{j_i}} \cdot (1 - \mathrm{ypi}_{\mathrm{p,u_p,i}}) \le \lambda 31_{\mathrm{p,u_p,j_i}} \quad i \in I, p \in \mathrm{PT}, u_{\mathrm{p}} \in U_{\mathrm{p}}, j_{\mathrm{i}} \in J_{\mathrm{i}} \quad (25)$$

$$\lambda 21_{\mathrm{p,u_p,j_i}} + \lambda 31_{\mathrm{p,u_p,j_i}} - 1 \le (U21_{\mathrm{j_i}} + U31_{\mathrm{j_i}}) \cdot (1 - \mathrm{ypi}_{\mathrm{p,u_p,i}}) \quad i \in I, p \in \mathrm{PT}, u_{\mathrm{p}} \in U_{\mathrm{p}}, j_i \in J_i \quad (26)$$

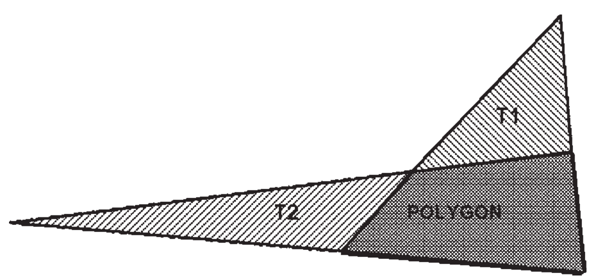

Figure 4. Polygon assignment.

$$\sum_{i} \mathrm{ypi}_{p,u_p,i} = 1 \qquad p \in \mathrm{PT}, u_p \in U_p \tag{27}$$

Objective Function. The objective function of the desalination location model is the total cost. The total cost is the sum of the capital costs of the existing plants, the variable production costs of the plants, the variable transportation costs to and from these plants, and the capital costs of the pipelines:

$$\begin{aligned}\mathrm{obj} = &\sum_{p}\sum_{u_p} \mathrm{fcostp}_{p,u_p} + \sum_{p}\sum_{u_p}(\mathrm{vcpup}_p \cdot \mathrm{wp}_{p,u_p}) \\ &+ \sum_{w}\sum_{p}\sum_{u_p}(\mathrm{vcp} \cdot \mathrm{woi}_{w,p,u_p}) + \sum_{s}\sum_{p}\sum_{u_p}(\mathrm{vcp} \cdot \mathrm{wps}_{p,u_p,s}) \\ &+ \sum_{w}\sum_{p}\sum_{u_p}\sum_{k} \mathrm{fcostwp}_{w,p,u_p,k} + \sum_{s}\sum_{p}\sum_{u_p}\sum_{k} \mathrm{fcostps}_{p,u_p,s,k}\end{aligned} \tag{28}$$

Additional Equations. These equations are used merely for accelerating the solution procedure; they are not part of the Basic MP but the results of the model-improving phase; therefore, we present them in their algebraic form only. In order to decrease the solution time, relaxation of the Big-M equations must be as tight as possible. For example, an upper bound to the distance parameter $\mathrm{UDS}_{s,k}$ between a particular storage tank and any plant can be calculated as the maximum of the distances between the storage tank and the borders of the plane.

To enhance the lower bound of the total cost (and thereby to make faster the optimization process) another constraint, using the minimum distance between the storage and the nearest water intake ($\mathrm{Dmins}_{s,k}$), is employed for calculating the minimum of the pipeline costs. If a particular connection between a storage tank and a plant exists then the sum of the pipelining costs calculated from the length of the pipelines between the storage tank and the plant, and between the plant and all the water intakes, must be larger than the pipelining cost calculated from the minimum distance between the storage tank and the nearest water intake:

$$\mathrm{fcp} \cdot \mathrm{Dmins}_{s,k} \cdot \mathrm{yps}_{p,u_p,s} \le \mathrm{fcostps}_{p,u_p,s,k} + \sum_{w} \mathrm{fcostwp}_{w,p,u_p,k} \qquad p \in \mathrm{PT}, u_p \in U_p, s \in S, k \in K \tag{29}$$

Calculation of the minimum cost for water intakes is performed with the same method.

Additional equations with binary variables for describing co-existence, or exclusion of co-existence, of some structure elements are also applied. With these equations, infeasible solutions can be excluded without examining the node in the Branch and Bound algorithm.

If a particular plant exists then at least one pipeline should be directed to this plant from one of the water intakes, and at least one pipeline should connect this plant to one of the storage tanks. These constraints are described by Eqs. 30 and 31.

$$\mathrm{yp}_{p,u_p} \le \sum_{w} \mathrm{ywp}_{w,p,u_p} \qquad p \in \mathrm{PT}, u_p \in U_p \tag{30}$$

$$\mathrm{yp}_{p,u_p} \le \sum_{s} \mathrm{yps}_{p,u_p,s} \qquad p \in \mathrm{PT}, u_p \in U_p \tag{31}$$

If a particular pipeline from a particular water intake to a particular plant exists then that plant must also exist (Eq. 32). Similarly, if a particular pipeline from a particular plant to a particular storage tank exists then that plant must also exist (Eq. 33).

$$\mathrm{ywp}_{w,p,u_p} \le \mathrm{yp}_{p,u_p} \qquad w \in W, p \in \mathrm{PT}, u_p \in U_p \tag{32}$$

$$\mathrm{yps}_{p,u_p,s} \le \mathrm{yp}_{p,u_p} \qquad p \in \mathrm{PT}, u_p \in U_p,\ s \in S \tag{33}$$

Model Variants. The general model hitherto explained is formulated in such a way that it may have multiple solutions with the same objective function value. Therefore, additional constraints are inserted to the model in order to decrease the number of equivalent solutions, without restricting the generality of the model. In order to decrease the computation time, six different variants are created.

Variant 0: No additional constraint is used; this is the basic variant.

Variant 1: A particular plant has a higher x coordinate than its predecessor in the increasing order of its index:

$$\mathrm{pp}_{p,u_p-1,x} \le \mathrm{pp}_{p,u_p,x} \qquad p \in \mathrm{PT}, u_p \in U_p \tag{34}$$

Variant 2: The plants are ordered along the y axis:

$$\mathrm{pp}_{p,u_p-1,y} \le \mathrm{pp}_{p,u_p,y} \qquad p \in \mathrm{PT}, u_p \in U_p \tag{35}$$

Variant 3: A particular plant cannot exist if its predecessor does not exist:

$$\mathrm{yp}_{p,u_p-1} \ge \mathrm{yp}_{p,u_p} \qquad p \in \mathrm{PT}, u_p \in U_p \tag{36}$$

Variant 4: Combination of the first and the third variants.

Variant 5: Combination of the second and the third variants.

Test runs of the basic model

GAMS 20.0. modeling system was used for all of the problems described in this work. CPLEX version 7.0.0 was applied as the MILP solver. The runs were performed on a PC with Intel 2.39 GHz and 752 MB RAM.

Results of Applying the Basic Model to Problem I. Results with triangles. The complementary area is subdivided into 12 triangles. The model statistics are detailed in Table 1.

The variants of additional constraints are applied on the above example problem. The optimum was found $1,036,557/day with all the six variants, but the computation time was different, as shown in Table 2.

The results in the Table 2 show that the shortest computation time is found with Variant 5.

The optimal plant allocation and the structure of water distribution found with Variant 5 are shown in Figures 5 and 6, respectively. One plant is situated between S1 and S4 sto-

Table 1. Model Statistics with Variant 0 of the Basic Model for Problem I, Using Triangles

Number of constraints	2076
Number of continuous variables	1320
Number of binary variables	280

Table 2. Computation Time Required for Different Variants of Problem I, Using Triangles

Variant	CPU Time (s)
0	774.77
1	221.02
2	512.77
3	278.08
4	103.39
5	98.42

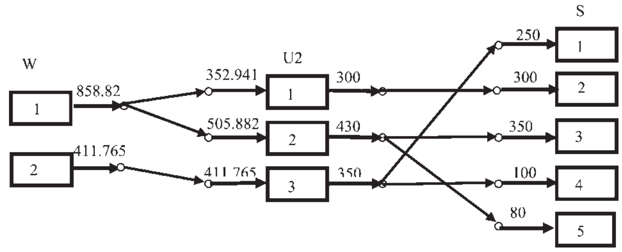

Figure 6. Optimal water distribution found with Variant 5, using triangles.

rages, one is mapped to S2 storage only, and one is located in the same point as S5 storage. There are no U1 or U3 plants in use because their capacities are smaller than of the U2 plants. It is not economical to use plants with small capacities if there are no consumers with small demands, or if the consumers are close to each other.

Results with the use of polygons. To reduce the computation time, the feasible region is assigned with polygons instead of triangles. Use of 10 polygons is sufficient to cover the feasible region. Variant 5 is selected to solve because it has been the best one with triangles. The model statistics and results are detailed in Table 3. The same optimal solution is found ($1,036,557/day); the number of the binary variables is smaller by 10%; the computational time is decreased by 30%.

The optimal plant allocation and the structure of the distribution found with Variant 5 are shown in Figures 7 and 8, respectively. Not surprisingly, the structure found is similar to that found with triangles.

Results of Applying the Basic Model to Problem II. Variant 5 of the basic model is applied for 17 polygons instead of 23 triangles. The model statistics and results are detailed in Table 4.

The optimum is found ($3,019,669/day) in 886.41 s. The optimal plant allocation and the structure of the distribution are shown in Figures 9 and 10, respectively. The U1(2), U2(1), and U2(2) plants have all distinct pipelines. Obviously, it would be more economical if there was a splitting point in the center of the area enclosed by S1-S2-S3-S4 storage tanks. In this way only one pipeline would be needed between the W(2) water intake and the central splitting point. This problem will be addressed subsequently.

Evaluation of the results of the basic model. The basic model is able to find an optimal solution for the defined problems in reasonable time. The effect of using additional equations to cut off the redundant solutions considerably accelerates the solutions process. Variant 0 without any additional equation decreasing the redundancy was the slowest one. The best result can be reached with Variant 5 where the plants are ordered along the *y*-axis, and a particular plant does not exist if its predecessor does not exist. It is therefore worthwhile to use this variant henceforward, although the difference of computational time between the Variants 4 and 5 is minimal.

The use of the polygons instead of triangles provides us with the same optimal solution. The saving in time was more than 30% in the case of Problem I; therefore we used that in the case of Problem II. When solving Problem II, the number of the binary variables increased by 44%, and the solution time increased to its 1300%. This is a common phenomenon because in general the computational time changes exponentially with the number of binary variables.

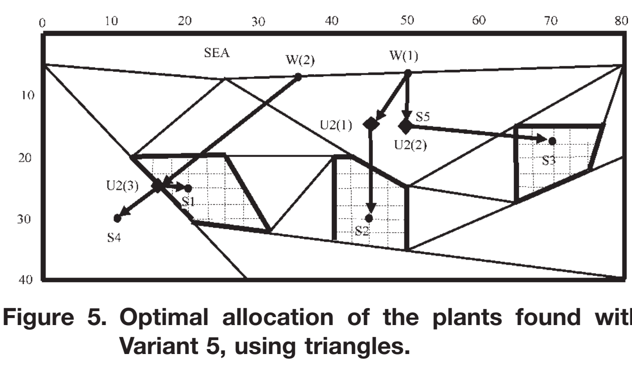

Figure 5. Optimal allocation of the plants found with Variant 5, using triangles.

The Improved Model

As it can be seen in Figure 5, the pipelines between the first water intake and the second and third u_2 plants partially coincide. In our basic model, distinct pipelines are designed to and from different plants even if their trace lines coincide. This gives rise to redundant costs compared to the real life solution of designing common pipeline to the common section, and then branching it.

One of the consequences of neglecting the branching option is that plant types with low capacity (i.e. the SD plants) will almost never be found economical. If the supplement of a storage tank needs the usage of more SD plants then each plant to storage tank connection requires a particular pipeline even if they are located at the same point of the plane. This is due to the inability of the pipeline to branch.

In order to deal with branching, so-called transportation units are introduced in the superstructure. The transportation units cannot produce potable water from seawater; the water flows unchanged through them. These new type units can have several inlet and outlet pipelines; this option provides us with the possibility of pipeline branching and/or merging.

Table 3. Model Statistics and Results with Variant 5 of the Basic Model for Problem I, Using Polygons

Number of constraints	2566
Number of continuous variables	1488
Number of binary variables	252
Solution [$/day]	1,036,557.463
CPU time (s)	65

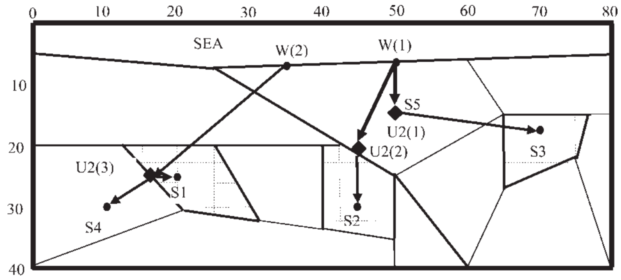

Figure 7. Optimal allocation of the plants found with Variant 5, using polygons.

In order to avoid mixing the potable water with seawater, two different types of transportation units are introduced. One is used for branching and merging seawater, the other one for branching and merging potable water. The transportation units are located either between the water intakes and the plants (denoted with ut1), or between the plants and the storage tanks (denoted with ut2).

Superstructure

The superstructure includes three kinds (p) of desalination plants (u_p), seawater intakes (w), fresh water storage tanks (s), first transportation units (ut1), and second transportation units (ut2). The superstructure is depicted in Figure 11.

Model formulation

Material Balance Constraints. The material balance constraints describe the requirement of conserving the water between the water intakes and first transport units (Eqs. 37 and 38), between the first transport units and plants (Eq. 39 and 40), between the plants and second transport units (Eqs. 42 and 43), and between the second transport units and storage tanks (Eqs. 44 and 45). Equation 41 describes the efficiency of the plants. The equations are given below, as follow.

$$\mathrm{wo}_{\mathrm{w}} = \sum_{\mathrm{ut1}} \mathrm{wwut1}_{\mathrm{w,ut1}} \qquad w \in W \tag{37}$$

$$\mathrm{wut1}_{\mathrm{ut1}} = \sum_{\mathrm{w}} \mathrm{wwut1}_{\mathrm{w,ut1}} \qquad \mathrm{ut1} \in \mathrm{UT1} \tag{38}$$

$$\mathrm{wut1}_{\mathrm{ut1}} = \sum_{\mathrm{p}} \sum_{\mathrm{u_p}} \mathrm{wut1}p_{\mathrm{ut1,p,u_p}} \qquad \mathrm{ut1} \in \mathrm{UT1} \tag{39}$$

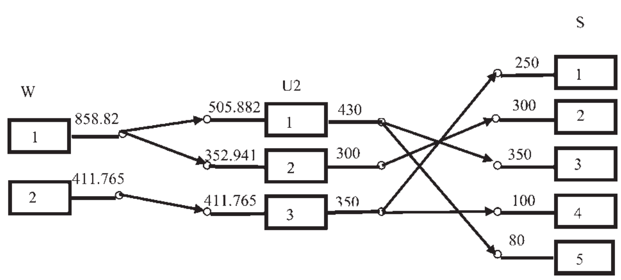

Figure 8. Optimal water distribution found with Variant 5, using polygons.

Table 4. Model Statistics and Results with Variant 5 of the Basic Model for Problem II, Using Polygons

Number of constraints	3659
Number of cont. variables	1992
Number of bin. variables	364
Solution [$/day]	3,019,669.4613
CPU time (s)	886.41

$$\mathrm{wi}_{\mathrm{p,u_p}} = \sum_{\mathrm{ut1}} \mathrm{wut1}p_{\mathrm{ut1,p,u_p}} \qquad p \in \mathrm{PT}, u_{\mathrm{p}} \in U_{\mathrm{p}} \tag{40}$$

$$\mathrm{wp}_{\mathrm{p,u_p}} = E_{\mathrm{p}} \cdot \mathrm{wi}_{\mathrm{p,u_p}} \qquad p \in \mathrm{PT}, u_{\mathrm{p}} \in U_{\mathrm{p}} \tag{41}$$

$$\mathrm{wp}_{\mathrm{p,u_p}} = \sum_{\mathrm{ut2}} \mathrm{wput2}_{\mathrm{p,u_p,ut2}} \qquad p \in \mathrm{PT}, u_{\mathrm{p}} \in U_{\mathrm{p}} \tag{42}$$

$$\mathrm{wut2}_{\mathrm{ut2}} = \sum_{\mathrm{p}} \sum_{\mathrm{u_p}} \mathrm{wput2}_{\mathrm{p,u_p,ut2}} \qquad \mathrm{ut2} \in \mathrm{UT2} \tag{43}$$

$$\mathrm{wut2}_{\mathrm{ut2}} = \sum_{\mathrm{s}} \mathrm{wut2}s_{\mathrm{ut2,s}} \qquad \mathrm{ut2} \in \mathrm{UT2} \tag{44}$$

$$\mathrm{ws}_{\mathrm{s}} = \sum_{\mathrm{ut2}} \mathrm{wut2}s_{\mathrm{ut2,s}} \qquad s \in S \tag{45}$$

Distances. Distances between seawater intake and first transportation units, between first transportation units and plants, between plants and second transportation units, and between second transportation units and storage tanks have to be calculated in the model in order to determine the transportation cost. Distances are defined according to Manhattan metric.[15]

Equations computing the distance between seawater intake and any first transportation unit are:

$$\mathrm{Dwut1}_{\mathrm{w,ut1,k}} \geq (\mathrm{pw}_{\mathrm{w,k}} - \mathrm{put1}_{\mathrm{ut1,k}}) \quad \mathrm{ut1} \in \mathrm{UT1}, w \in W, k \in K \tag{46}$$

$$\mathrm{Dwut1}_{\mathrm{w,ut1,k}} \geq (\mathrm{put1}_{\mathrm{ut1,k}} - \mathrm{pw}_{\mathrm{w,k}}) \quad \mathrm{ut1} \in \mathrm{UT1}, w \in W, k \in K \tag{47}$$

Equations computing the distance between the first transportation unit and any plant are:

$$\mathrm{Dut1}p_{\mathrm{ut1,p,u_p,k}} \geq (\mathrm{put1}_{\mathrm{ut1,k}} - \mathrm{pp}_{\mathrm{p,u_p,k}}) \quad p \in \mathrm{PT}, u_{\mathrm{p}} \in U_{\mathrm{p}}, \mathrm{ut1} \in \mathrm{UT1}, k \in K \tag{48}$$

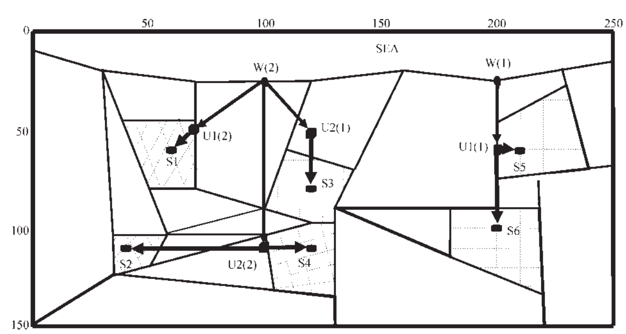

Figure 9. Optimal allocation of the plants, using polygons.

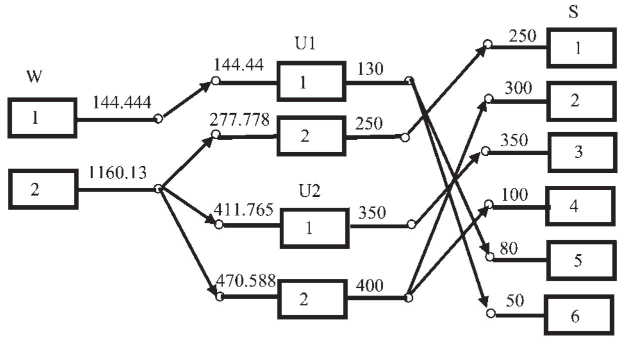

Figure 10. Optimal water distribution, using polygons.

$$\mathrm{Dut1}p_{\mathrm{ut1,p,u_p,k}} \geq (\mathrm{pp}_{\mathrm{p,u_p,k}} - \mathrm{put1}_{\mathrm{ut1,k}}) \quad p \in \mathrm{PT}, u_\mathrm{p} \in U_\mathrm{p}, \mathrm{ut1} \in \mathrm{UT1}, k \in K \tag{49}$$

Equations computing the distance between plant and any second transportation unit are:

$$\mathrm{Dput2}_{\mathrm{p,u_p,ut2,k}} \geq (\mathrm{put2}_{\mathrm{ut2,k}} - \mathrm{pp}_{\mathrm{p,u_p,k}}) \quad p \in \mathrm{PT}, u_\mathrm{p} \in U_\mathrm{p}, \mathrm{ut2} \in \mathrm{UT2}, k \in K \tag{50}$$

$$\mathrm{Dput2}_{\mathrm{p,u_p,ut2,k}} \geq (\mathrm{pp}_{\mathrm{p,u_p,k}} - \mathrm{put2}_{\mathrm{ut2,k}}) \quad p \in \mathrm{PT}, u_\mathrm{p} \in U_\mathrm{p}, \mathrm{ut2} \in \mathrm{UT2}, k \in K \tag{51}$$

Equations computing the distance between the second transportation unit and storage tanks are:

$$\mathrm{Dut2}s_{\mathrm{ut2,s,k}} \geq (\mathrm{put2}_{\mathrm{ut2,k}} - \mathrm{ps}_{\mathrm{s,k}}) \quad \mathrm{ut2} \in \mathrm{UT2}, s \in S, k \in K \tag{52}$$

$$\mathrm{Dut2}s_{\mathrm{ut2,s,k}} \geq (\mathrm{ps}_{\mathrm{s,k}} - \mathrm{put2}_{\mathrm{ut2,k}}) \quad \mathrm{ut2} \in \mathrm{UT2}, s \in S, k \in K \tag{53}$$

Logic Expressions. *Logic expression on the existence of the connection between sea water intake w and first transportation unit ut1*. The amount of water transported through a nonexisting pipeline must be zero. The fixed cost of an existing pipeline is calculated from its length:

$$\begin{bmatrix} \mathrm{Zwut1}_{\mathrm{w,ut1}} \\ 0 \leq \mathrm{wwut1}_{\mathrm{w,ut1}} \leq \mathrm{CAPWUT1}_{\mathrm{w,ut1}} \\ \mathrm{fcostwut1}_{\mathrm{w,ut1,k}} = \mathrm{fcp} \cdot \mathrm{Dwut1}_{\mathrm{w,ut1,k}} \end{bmatrix} \vee \begin{bmatrix} \neg\mathrm{Zwut1}_{\mathrm{w,ut1}} \\ \mathrm{wwut1}_{\mathrm{w,ut1}} = 0 \\ \mathrm{fcostwut1}_{\mathrm{w,ut1,k}} = 0 \end{bmatrix} \quad w \in W, \mathrm{ut1} \in \mathrm{UT1}, k \in K \tag{54}$$

Logic expression on the existence of the connection between first transportation unit ut1 and plant u_p of type p. The amount of water transported through a non-existing pipeline must be zero. The fixed cost of an existing pipeline is calculated from its length:

$$\begin{bmatrix} \mathrm{Zut1}p_{\mathrm{ut1,p,u_p}} \\ 0 \leq \mathrm{wut1}p_{\mathrm{ut1,p,u_p}} \leq \mathrm{CAPUTP}_{\mathrm{ut1,p}} \\ \mathrm{fcostut1}p_{\mathrm{ut1,p,u_p,k}} = \mathrm{fcp} \cdot \mathrm{Dut1}p_{\mathrm{ut1,p,u_p,k}} \end{bmatrix} \vee \begin{bmatrix} \neg\mathrm{Zut1}p_{\mathrm{ut1,p,u_p}} \\ \mathrm{wut1}p_{\mathrm{ut1,p,u_p}} = 0 \\ \mathrm{fcostut1}p_{\mathrm{ut1,p,u_p,k}} = 0 \end{bmatrix} \quad \mathrm{ut1} \in \mathrm{UT1}, p \in \mathrm{PT}, u_\mathrm{p} \in U_\mathrm{p}, k \in K \tag{55}$$

Logic expression on the existence of plants u_p of type p. This is same as used in the basic model.

Logic expression on the existence of the connection between plant u_p of type p and second transportation unit ut2. The amount of water transported by a non-existing pipeline must be zero. The fixed cost of an existing pipeline is calculated from its length:

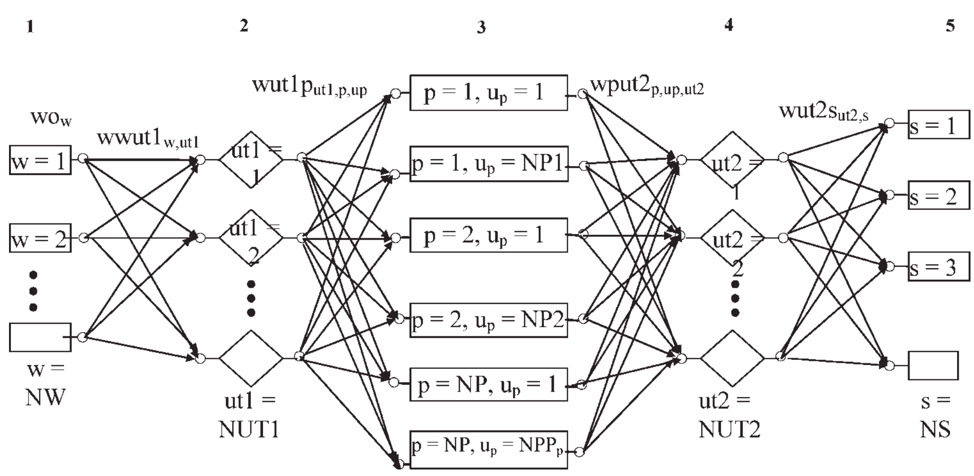

Figure 11. The improved superstructure of location problem.

$$\begin{bmatrix} \text{Zput2}_{p,u_p,ut2} \\ 0 \le \text{wput2}_{p,u_p,ut2} \le \text{CAPPUT}_{p,ut2} \\ \text{fcostput2}_{p,u_p,ut2,k} = \text{fcp} \cdot \text{Dput2}_{p,u_p,ut2,k} \end{bmatrix} \vee \begin{bmatrix} \neg\text{Zput2}_{p,u_p,ut2} \\ \text{wput2}_{p,u_p,ut2} = 0 \\ \text{fcostput2}_{p,u_p,ut2,k} = 0 \end{bmatrix}$$
$$p \in \text{PT}, u_p \in U_p, \text{ut2} \in \text{UT2}, k \in K \quad (56)$$

Logic expression on the existence of the connection between second transportation unit ut2 and storage tanks s. The amount of water transported by a non-existing pipeline must be zero. The fixed cost of an existing pipeline is calculated from its length:

$$\begin{bmatrix} \text{Zut2}s_{\text{ut2,s}} \\ 0 \le \text{wut2}s_{\text{ut2,s}} \le \text{CAPUTS}_{\text{ut2,s}} \\ \text{fcostut2}s_{\text{ut2,s,k}} = \text{fcp} \cdot \text{Dut2}s_{\text{ut2,s,k}} \end{bmatrix} \vee \begin{bmatrix} \neg\text{Zut2}s_{\text{ut2,s}} \\ \text{wut2}s_{\text{ut2,s}} = 0 \\ \text{fcostut2}s_{\text{ut2,s,k}} = 0 \end{bmatrix}$$
$$\text{ut2} \in \text{UT2}, s \in S, k \in K \quad (57)$$

Transformation to algebraic equations. Equation 57 is reformulated to algebraic equations using the same method used for transforming Eqs. 11–13:

$$\text{wut2}s_{\text{ut2,s}} \le \text{CAPUTS}_{\text{ut2,s}} \cdot \text{yut2}s_{\text{ut2,s}} \quad \text{ut2} \in \text{UT2}, s \in S \quad (58)$$

$$\text{fcostut2}s_{\text{ut2,s,k}} - \text{fcp} \cdot \text{Dut2}s_{\text{ut2,s,k}} \le \text{UDS}_{\text{s,k}} \cdot (1 - \text{yut2}s_{\text{ut2,s}}) \quad \text{ut2} \in \text{UT2}, s \in S, k \in K \quad (59)$$

$$-\text{UDS}_{\text{s,k}} \cdot (1 - \text{yut2}s_{\text{ut2,s}}) \le \text{fcostut2}s_{\text{ut2,s,k}} - \text{fcp} \cdot \text{Dut2}s_{\text{ut2,s,k}} \quad \text{ut2} \in \text{UT2}, s \in S, k \in K \quad (60)$$

Reformulation of the other logic expressions is performed with the same technique.

Assignment of the Feasible Region. The equations of the assignment of feasible region are the same as in the basic model.

Objective Function. The objective function is the total cost. The total cost is the sum of the capital costs of the existing plants, the variable production costs of the plants, the variable transportation costs to and from these plants, and the capital costs of the pipelines:

$$\begin{aligned} \text{obj} = &\sum_{p}\sum_{u_p} \text{fcostp}_{p,u_p} + \sum_{p}\sum_{u_p} (\text{vcpup}_p \cdot \text{wp}_{p,u_p}) \\ &+ \sum_{w}\sum_{\text{ut1}} (\text{vcp} \cdot \text{wwut1}_{\text{w,ut1}}) + \sum_{\text{ut1}}\sum_{p}\sum_{u_p} (\text{vcp} \cdot \text{wut1}p_{\text{ut1},p,u_p}) \\ &+ \sum_{\text{ut2}}\sum_{p}\sum_{u_p} (\text{vcp} \cdot \text{wput2}_{p,u_p,\text{ut2}}) + \sum_{s}\sum_{\text{ut2}} (\text{vcp} \cdot \text{wut2}s_{\text{ut2,s}}) \\ &+ \sum_{w}\sum_{\text{ut1}}\sum_{k} \text{fcostwut1}_{\text{w,ut1,k}} + \sum_{\text{ut1}}\sum_{p}\sum_{u_p}\sum_{k} \text{fcostut1}p_{\text{ut1},p,u_p,k} \\ &+ \sum_{\text{ut2}}\sum_{p}\sum_{u_p}\sum_{k} \text{fcostput2}_{p,u_p,\text{ut2},k} + \sum_{s}\sum_{\text{ut2}}\sum_{k} \text{fcostut2}s_{\text{ut2,s,k}} \end{aligned} \quad (61)$$

Additional Equations. A relation similar to Eq. 29 is used in the improved model, but not the same one because it cannot enhance the solution process here. Namely, if a particular connection from a storage tank to a plant exists then the pipeline cost of this connection and the sum of the costs of the connections to the same plant from the water intakes must be at least as high in the basic model as the cost of a pipeline to the storage tank from the nearest water intake.

This constraint is changed in the improved model because of the presence of the transportation units. If a particular connection to a storage tank from a transportation unit ut2 exist then the pipeline cost of this connection and the sum of the costs of the connections to this transportation unit ut2 from any plant, plus the sum of the costs of the connections to any plant from any transportation unit ut1, plus the sum of the costs of the connections to any transportation unit ut1 from any water intake, must be at least as high as the cost of a pipeline to the storage tank from the nearest water intake. This constraint is expressed by Eq. 62.

$$\begin{aligned} \text{fcp} \cdot \text{Dmins}_{\text{s,k}} \cdot \text{yut2}s_{\text{ut2,s}} \le &\ \text{fcostut2}s_{\text{ut2,s,k}} \\ &+ \sum_{p}\sum_{u_p} \text{fcostput2}_{p,u_p,\text{ut2},k} + \sum_{\text{ut1}}\sum_{p}\sum_{u_p} \text{fcostut1}p_{\text{ut1},p,u_p,k} \\ &+ \sum_{w}\sum_{\text{ut1}} \text{fcostwut1}_{\text{w,ut1,k}} \quad \text{ut2} \in \text{UT2}, s \in S, k, \in K \end{aligned} \quad (62)$$

The following relations are analogous to Eqs. 30–33:

$$\text{yp}_{p,u_p} \le \sum_{\text{ut1}} \text{yut1}p_{\text{ut1},p,u_p} \quad p \in \text{PT}, u_p \in U_p \quad (63)$$

$$\text{yp}_{p,u_p} \le \sum_{\text{ut2}} \text{yput2}_{p,u_p,\text{ut2}} \quad p \in \text{PT}, u_p \in U_p \quad (64)$$

$$\text{yut1}p_{\text{ut1},p,u_p} \le \text{yp}_{p,u_p} \quad \text{ut1} \in \text{UT1}, p \in \text{PT}, u_p \in U_p \quad (65)$$

$$\text{yput2}_{p,u_p,\text{ut2}} \le \text{yp}_{p,u_p} \quad p \in \text{PT}, u_p \in U_p, \text{ut2} \in \text{UT2} \quad (66)$$

Tightening the Relaxation. The above model in its hitherto presented form cannot be used within reasonable time. The relaxation made with the Big-M equations is therefore tightened by applying reasonable estimation for the maximum distance Big-M parameters. Such estimations can be made by using the solution obtained with the basic model.

In the original form, the upper bound of the distance is equal to the extent of the plane in a direction. This is decreased by creating a particular feasible region for each transport unit and plant. The plane is divided to three approximately equal size sub-regions along the *x*-axis. One third of the plants and transportation units are located in the left hand side of the plane, another third in the middle, and the last third in the right hand side of the plane.

Table 5. Model Statistics and Results with Improved Model for Problem I, Using Polygons

Number of constraints	3826
Number of cont. variables	2373
Number of bin. variables	272
Solution [$/day]	926,338.8335
CPU time (s)	3917.63

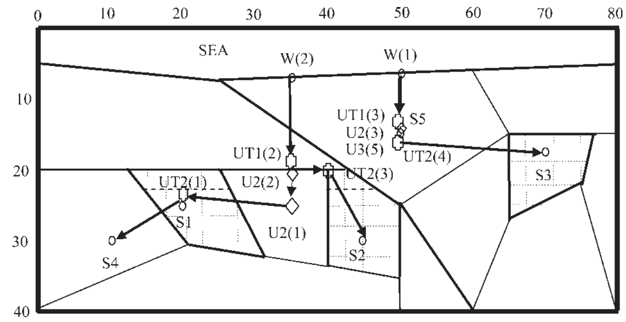

Figure 12. Optimal allocation of the plants to Problem I using polygons with transportation units.

The transportation units located in the left hand side region are not connected to the storage tanks located in the right hand side region of the plane, according to the estimation used for bounding. It is rather improbable to have an optimum solution with such a connection because of the high transportation costs due to the long pipelines. Therefore, the connections between units being far from each other are forbidden. The Big-M equations, i.e. Eqs. 59 and 60, are applied to the allowed connections only:

$$\text{fcostut2}s_{\text{ut2,s,k}} - \text{fcp} \cdot \text{Dut2}s_{\text{ut2,s,k}} \leq \text{fcp} \cdot \text{UDUT2}S_{\text{ut2,s,k}} \times (1 - \text{yut2}s_{\text{ut2,s}}) \qquad (\text{ut2}, s) \in \{(\text{ut2}, s) | \text{ut2} \in \text{UT2}, s \in S, \text{yut2}s.\text{up}(\text{ut2}, s) = 1\}, k \in K \quad (67)$$

$$- \text{fcp} \cdot \text{UDUT2}S_{\text{ut2,s,k}} \cdot (1 - \text{yut2}s_{\text{ut2,s}}) \leq \text{fcostut2}s_{\text{ut2,s,k}} - \text{fcp} \cdot \text{Dut2}s_{\text{ut2,s,k}} \qquad (\text{ut2}, s) \in \{(\text{ut2}, s) | \text{ut2} \in \text{UT2}, s \in S, \text{yut2}s.\text{up}(\text{ut2}, s) = 1\}, k \in K \quad (68)$$

Consequently, the Big-M parameters will not be the maximum distances ((i.e. the width or height in the plane), but the maximum *allowed* distances only.

This approach basically is a heuristic. To avoid cutting off the optimum, it has to be used in an iterative form. First, the Big-M parameters of the distances in Eqs. 67 and 68 can be estimated using the results obtained with the basic model. Then after increasing the maximum allowed distances and the number of the allowed connections, the resulted change in the best value of the objective function can be checked. The algorithm is terminated when there is no further improvement.

Application of restrictions (33)–(35) for the locations and connections of the plants as in the basic model can cut off the optimum or even all the feasible solutions in case of the improved model. Therefore, these constraints are not applied in the improved model.

The actual feasible regions and forbidden connections used in the test examples are detailed in Tables A5–A15 and Tables A20–A30 in the Appendix.

Table 6. Model Statistics and Results with Improved Model for Problem II, Using Polygons

Number of constraints	4384
Number of cont. variables	2815
Number of bin. variables	368
Solution [$/day]	2,546,144.5489
CPU time (s)	7051.75

Test runs of the improved model

The problems were solved in the same computing environment used for the basic model.

Results of Applying the Improved Model to Problem I. The model statistics and results are detailed in Table 5. It can be seen that the computational time is increased by more than an order of magnitude, although the number of binary variables increased with 8% only. This is due to the deteriorating relaxation.

The optimal solution found is $926,338/day, 10.63% less than the solution obtained with the basic model; the computation time was 3617.63 s. The optimal plant allocation and the structure of the distribution are shown in Figures 12 and 13, respectively. Here the S2 storage tank obtains its water from W(2) water intake. In point (35; 20) there is a splitting centre. There is only one pipeline between W(2) and this splitting point. This could not be possible with the basic model. Similarly, there is another splitting centre in (20; 25) where the water comes to S4 storage tank from S1. This could not be possible with the basic model, either. The effect of these two splitting centers provides the decreasing in the total cost. New phenomena is that one U3 SD plant is used as well. This, again, was not possible with the basic model because this plant would need a distinct pipeline. Here, the plant is situated in the splitting point. In this way, only one pipeline is needed between the W(1) water intake and the splitting point; and only one is needed between splitting point and S3 storage tank.

Results of Applying the Improved Model to Problem II. The model statistics and the results are detailed in Table 6. It can be seen that the computational time increased by its

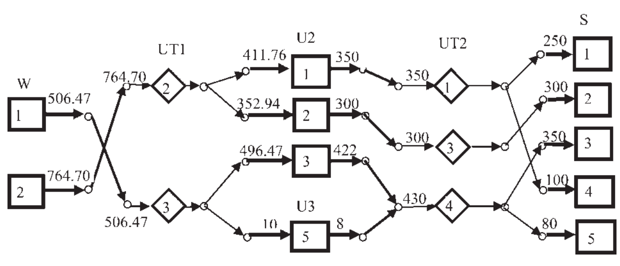

Figure 13. Optimal water distribution to Problem I using polygons with transportation units.

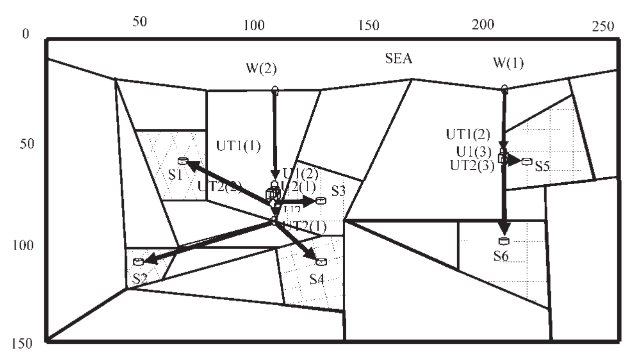

Figure 14. Optimal allocation of the plants to Problem II using polygons with transportation units.

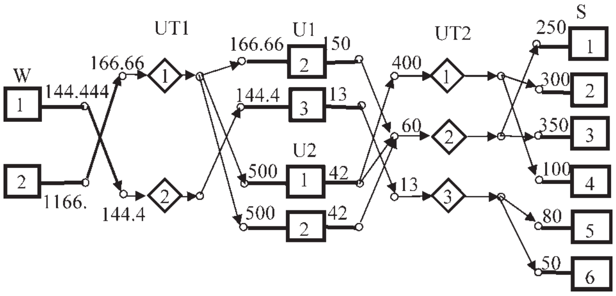

Figure 15. Optimal water distribution to Problem II using polygons with transportation units.

700% although the number of the binary variables increased with 1% only. This is due to the deteriorating relaxation, again.

The optimal solution found is \$2,546,144/day, which is 13.23% less than the previous solution; the computation time is 7051.75 s.

The plant allocation and the structure of the distribution are shown in Figures 14 and 15, respectively. There is a huge water distribution hub in point (100; 75). Between W(2) and this splitting point, there is only one pipeline. This could not be possible with the basic model.

Conclusion with the improved model

Using well-estimated Big-M values, the improved model provides a solution which takes into account the possibility of pipeline branching. Consequently, the solution obtained using the improved model designs network for the studied examples with 10–15% less total cost than the basic model.

Conclusions

Successfully applicable MILP models have been developed for determining optimal desalination plant positions in a given feasible subset of the plane.

The basic model is able to find a near optimal solution with the deficiency of allocating redundant multiple pipelines. Equivalent solutions can be excluded by applying additional constraints. All the six model variants obtained in this way result in identical solution. Variant 5 is the fastest one; this variant applies a special ordering of locations (Eq. 35) and plant existence (Eq. 36).

The feasible region can be allocated with using triangles or polygons; use of polygons decreases the computation time.

The deficiency of the basic model is removed by introducing transportation units in the superstructure. The improved model is capable of finding optimal solution with branching pipelines.

Acknowledgments

This research was supported by OTKA K 62099, F 46282.

Notation

Sets

I = set of polygons
J = set of triangles
$J_i \subseteq J$ = set of triangles belonging to polygon i
K = set of coordinates
PT = set of plant types
S = set of storage tanks
U_p = set of plants of type p
UT1 = set of first transportation units
UT2 = set of second transportation units
W = set of water intakes

Indices

i = index for polygons
j = index for triangles
j_i = index for triangles belonging to polygon i
k = index for coordinates
p = index for plant types
s = index for storage tanks
u_p = index for plants of type p
ut1 = index for first transportation units
ut2 = index for second transportation units
w = index for water intakes

Parameters

CAP_p = capacity of plants of type p (m^3/day)
$\mathrm{CAPPS}_{p,s}$ = capacity of the pipeline between plant u_p of type p and storage tank s (m^3/day)
$\mathrm{CAPPUT}_{p,ut2}$ = capacity of the pipeline between plant u_p of type p and transport unit ut2 (m^3/day)
$\mathrm{CAPUTP}_{ut1,p}$ = capacity of the pipeline between transport unit ut1 and plant u_p of type p (m^3/day)
$\mathrm{CAPUTS}_{ut2,s}$ = capacity of the pipeline between transport unit ut2 and storage tank s (m^3/day)
$\mathrm{CAPWP}_{w,p}$ = capacity of the pipeline between water intake w and plant u_p of type p (m^3/day)
$\mathrm{CAPWUT}_{w,ut1}$ = capacity of the pipeline between water intake w and transportation unit ut1 (m^3/day)
$\mathrm{Dmins}_{s,k}$ = minimal distance measured from storage tank s to the nearest water intake
E_p = efficiency of plants of type p
fcp = fixed cost factor of the pipeline (\$/day)
fcup_p = fixed cost factor of plants of type p (\$/day)
$L21_j$ = lower bound of lambda of edge $\overline{P1P2}$ of triangle j
$L31_j$ = lower bound of lambda of edge $\overline{P1P3}$ of triangle j
NW = number of water intakes
NP = number of plant types
NP1 = number of plants of type 1
NP2 = number of plants of type 2
NPP_p = number of plants of type NP
NUT1 = number of first transport units
NUT2 = number of second transport units
NS = number of storage tanks
$P1_{j,k}$ = coordinate k of point 1 of triangle j
$P2_{j,k}$ = coordinate k of point 2 of triangle j
$P3_{j,k}$ = coordinate k of point 3 of triangle j
$\mathrm{ps}_{s,k}$ = coordinate k of storage tank s
$\mathrm{pw}_{w,k}$ = coordinate k of water intake w
$\Pi1_k$ = coordinate k of point $\Pi1$ used in the explanation of the idea of triangles
$\Pi2_k$ = coordinate k of point $\Pi2$ used in the explanation of the idea of triangles
$\Pi3_k$ = coordinate k of point $\Pi3$ used in the explanation of the idea of triangles
$U21_j$ = upper bound of lambda of edge $\overline{P1P2}$ of triangle j
$U31_j$ = upper bound of lambda of edge $\overline{P1P3}$ of triangle j
$\mathrm{UDS}_{s,k}$ = maximal distance measured from storage tank s
$\mathrm{UDUT1}P_{ut1,p,u_{p,k}}$ = maximal allowed distance measured from transport unit ut1 to plant u_p of type p
$\mathrm{UDPUT2}_{p,u_{p,ut2,k}}$ = maximal allowed distance measured from plant u_p of type p to transport unit ut2
$\mathrm{UDUT2}S_{ut2,s,k}$ = maximal allowed distance measured from transport unit ut2 to storage tank s
$\mathrm{UDWUT1}_{w,ut1,k}$ = maximal allowed distance measured from water intake w to transport unit ut1
vcp = variable cost factor of water transported by pipeline (\$/m^3)

$vcpup_p$ = variable cost factor of plants of type p ($/$m^3$)
ws_s = potable water transported to storage tank s (m^3/day)
$xplo_{p,u_p}$ = lower bound for the x-coordinate of plant u_p of type p
$xpup_{p,u_p}$ = upper bound for the x-coordinate of plant u_p of type p
$xut1lo_{ut1}$ = lower bound for the x-coordinate of transport unit ut1
$xut1up_{ut1}$ = upper bound for the x-coordinate of transport unit ut1
$xut2lo_{ut2}$ = lower bound for the x-coordinate of transport unit ut2
$xut2up_{ut2}$ = upper bound for the x-coordinate of transport unit ut2
$yplo_{p,u_p}$ = lower bound for the y-coordinate of plant u_p of type p
$ypup_{p,u_p}$ = upper bound for the y-coordinate of plant u_p of type p
$yut1lo_{ut1}$ = lower bound for the y-coordinate of transport unit ut1
$yut1up_{ut1}$ = upper bound for the y-coordinate of transport unit ut1
$yut2lo_{ut2}$ = lower bound for the y-coordinate of transport unit ut2
$yut2up_{ut2}$ = upper bound for the y-coordinate of transport unit ut2

Variables

$Dput2_{p,u_{p,ut2,k}}$ = distance in coordinate k between plant u_p of type p and transportation unit ut2
$Ds_{p,u_{p,s,k}}$ = distance in coordinate k between plant u_p of type p and storage tank s
$Dut1p_{ut1,p,u_{p,k}}$ = distance in coordinate k between transportation unit ut1 and plant u_p of type p
$Dut2s_{ut2,s,k}$ = distance in coordinate k between transportation unit ut2 and storage tank s
$Dw_{w,p,u_{p,k}}$ = distance in coordinate k between seawater intake w and plant u_p of type p
$Dwut1_{w,ut1,k}$ = distance in coordinate k between seawater intake w and transportation unit ut1
$fcostp_{p,u_p}$ = fixed cost of plant u_p of type p ($/day)
$fcostps_{p,u_{p,s,k}}$ = fixed cost of pipeline between plant u_p of type p and storage tank s ($/day)
$fcostput2_{p,u_{p,ut2,k}}$ = fixed cost of pipeline between plant u_p of type p and transport unit ut2 ($/day)
$fcostut1p_{ut1,p,u_{p,k}}$ = fixed cost of pipeline between transport unit ut1 and plant u_p of type p ($/day)
$fcostut2s_{ut2,s,k}$ = fixed cost of pipeline between transport unit ut2 and storage tank s ($/day)
$fcostwp_{w,p,u_{p,k}}$ = fixed cost of pipeline between water intake w and plant u_p of type p ($/day)
$fcostwut1_{w,ut1,k}$ = fixed cost of pipeline between water intake w and transport unit ut1 ($/day)
$\lambda 21_{p,u_{p,j}}$ = lambda of plant u_p of type p on edge $\overline{P1P2}$ belonging to triangle j
$\lambda 31_{p,u_{p,j}}$ = lambda of plant u_p of type p on edge $\overline{P1P3}$ belonging to triangle j
$\Lambda 21$ = lambda of P on edge $\overline{\Pi 1 \Pi 2}$ used in the explanation of the idea of triangles
$\Lambda 31$ = lambda of P on edge $\overline{\Pi 1 \Pi 3}$ used in the explanation of the idea of triangles
obj = objective function ($/day)
P_k = coordinate k of a general point used in the explanation of the idea of triangles
$pp_{p,u_{p,k}}$ = coordinate k of plant u_p of type p
$put1_{ut1,k}$ = coordinate k of transport unit ut1
$put2_{ut2,k}$ = coordinate k of transport unit ut2
wo_w = seawater produced by water intake w (m^3/day)
wi_{p,u_p} = seawater transported to plant u_p of type p (m^3/day)
woi_{w,p,u_p} = seawater transported from seawater intake w to plant u_p of type p (m^3/day)
wp_{p,u_p} = potable water produced by plant u_p of type p (m^3/day)
$wps_{p,u_{p,s}}$ = potable water transported from plant u_p of type p to storage tank s (m^3/day)
$wput2_{p,u_{p,ut2}}$ = potable water transported from plant u_p of type p to transportation unit ut2 (m^3/day)
$wut1_{ut1}$ = seawater transported by transportation unit ut1 (m^3/day)
$wut1p_{ut1,p,u_p}$ = seawater transported from transportation unit ut1 to plant u_p of type p (m^3/day)
$wut2_{ut2}$ = potable water transported by transportation unit ut2 (m^3/day)
$wut2s_{ut2,s}$ = potable water transported from transportation unit ut2 to storage tank s (m^3/day)
$wwut1_{w,ut1}$ = seawater transported from seawater intake w to transportation unit ut1 (m^3/day)

Logic variables

Zwp_{w,p,u_p} = true, if there is a pipeline between water intake w and plant u_p of type p
Zp_{p,u_p} = true, if plant u_p of type p exists
$Zpj_{p,u_{p,j}}$ = true, if plant u_p of type p exists within triangle j
$Zps_{p,u_{p,s}}$ = true, if there is a pipeline between plant u_p of type p and storage tank s
$Zput2_{p,u_{p,ut2}}$ = true, if there is a pipeline between plant u_p of type p and transportation unit ut2
$Zut1p_{ut1,p,u_p}$ = true, if there is a pipeline between transportation unit ut1 and plant u_p of type p
$Zut2s_{ut2,s}$ = true, if there is a pipeline between transportation unit ut2 and storage tank s
$Zwut1_{w,ut1}$ = true, if there is a pipeline between water intake w and transportation unit ut1

Binary variables

yp_{p,u_p} = 1, if plant u_p of type p exists; 0 otherwise
$ypi_{p,u_{p,i}}$ = 1, if plant u_p of type p exists within polygon i; 0 otherwise
$ypj_{p,u_{p,j}}$ = 1, if plant u_p of type p exists within triangle j; 0 otherwise
$yps_{p,u_{p,s}}$ = 1, if there is a pipeline between plant u_p of type p and storage tank s; 0 otherwise
$yput2_{p,u_{p,ut2}}$ = 1, if there is a pipeline between plant u_p of type p and transportation unit ut2; 0 otherwise
$yut1p_{ut1,p,u_p}$ = 1, if there is a pipeline between transportation unit ut1 and plant u_p of type p; 0 otherwise
$yut2s_{ut2,s}$ = 1, if there is a pipeline between transportation unit ut2 and storage tank s; 0 otherwise
ywp_{w,p,u_p} = 1, if there is a pipeline between water intake w and plant u_p of type p; 0 otherwise
$ywut1_{w,ut1}$ = 1, if there is a pipeline between water intake w and transportation unit ut1; 0 otherwise

Literature Cited

1. http://www.coastal.ca.gov/desalrpt/dchap1.html.
2. Mink G, Horváth L, Evseev EG, Kudish AI. Design parameters, performance testing and analysis of a double-glazed, air-blown solar still with thermal energy recycle. *Sol Energy*. 1998;64:265–277.
3. Andrinne J, Alardin F. Desalination site selection on North-African coasts. *Desalination*. 2004;165:231–239.
4. ReVelle CS, Eiselt HA. Location analysis: a synthesis and survey. *Eur J Oper Res*. 2005;165:1–19.
5. Nickel S. Restricted center problems under polyhedral gauges. *Eur J Oper Res*. 1998;104:343–357.
6. Özyurt DB, Realff MB. Geographic and process information for chemical plant layout problem. *AIChE J*. 1999;45:2161–2174.
7. Boland N, Domínguez-Marín P, Nickel S, Puerto J. Exact procedures for solving the discrete ordered median problem. *Comput Oper Res*. 2006;33:3270–3300.
8. Wu LY, Zhang XS, Zhang JL. Capacitated facility location problem with general setup cost. *Comput Oper Res*. 2006;33:1226–1241.
9. Klose A, Drexl A. Facility location models for distribution system design. *Eur J Oper Res*. 2005;162:4–29.
10. Rosing KE. Towards the solution of the (generalized) multi-Weber problem. *Environ Plann Ser B*. 1991;18:347–360.
11. Rosing KE. An optimal method for solving the (generalized) multi-Weber problem. *Eur J Oper Res*. 1992;58:414–426.
12. Yeomans H, Grossmann IE. A systematic modeling framework of superstructure optimization in process synthesis. *Comput Chem Eng*. 1999;23:709–731.

13. Vecchietti A, Lee S, Grossmann IE. Modeling of discrete/continuous optimization problems: characterization and formulation of disjunctions and their relaxations. *Comput Chem Eng*. 2003;27:433–448.
14. Farkas T, Rev E, Lelkes Z. Process flowsheet superstructures: structural multiplicity and redundancy, Part 1: Basic GDP and MINLP representations. *Comput Chem Eng*. 2005;29:2180–2197.
15. Peer SK, Sharma DK, Ravindranath K, Naidu MM. Computation of arc length in the presence of barriers in networks. *Appl Math Comput*. 2006;175:541–556.

Appendix

Data of the Problem I (Tables A1–A4)

Table A1. Number, Capacity, and Efficiencies of Plants of Different Types

p	NPP_p	CAP_p	E_p
1	3	250	0.9
2	3	500	0.85
3	8	10	0.8

Table A2. Coordinates of the Triangles

	$P1$		$P2$		$P3$	
Δ	x	y	x	Y	x	y
1	0	40	0	5	31.27389	40
2	13.40305	20	0	5	25	8
3	13.40305	20	25	8	42.647	20
4	33	34	25	20	40	20
5	33	34	40	20	40	34.8936
6	31.27389	40	25	32.9787	80	40
7	50	25	25	8	80	5
8	50	25	65	15	65	27
9	50	36.1702	50	25	65	27
10	50	36.1702	80	17.8298	80	40
11	65	15	80	5	76.85268	15
12	75	20.88652	80	5	80	17.8298

Table A3. Coordinates of the Water Intakes

Sea Water Intake	x	y
1	50	6.636364
2	35	7.545455

Table A4. The Demands and Coordinates of Storage Tanks

Sea Water Intake	Demand (m^3/day)	x	y
1	250	10	25
2	300	45	30
3	350	70	18
4	100	10	30
5	80	50	15

Actual Feasible regions and forbidden connections for Problem I (Tables A5–A15).

Table A5. Lower and Upper Bounds for the *x*- and *y*-Coordinates of Transport Unit ut1

ut1	xut1lo	xut1up	yut1lo	yut1up
1	10	27	10	20
2	27	45	10	20
3	40	55	10	20
4	50	70	10	20

Table A6. Lower and Upper Bounds for the *x*- and *y*-Coordinates of Plant u_p of Type p

p	u_p	xplo	xpup	yplo	ypup
1	1	10	65	10	30
	2	10	65	10	30
	3	10	65	10	30
2	1	10	65	10	30
	2	10	65	10	30
	3	10	65	10	30
3	1	10	65	10	30
	2	10	65	10	30
	3	10	65	10	30
	4	10	65	10	30
	5	10	65	10	30
	6	10	65	10	30
	7	10	65	10	30
	8	10	65	10	30

Table A7. Lower Bounds for the *x*-Coordinates of Transport Unit ut2

ut2	xut2lo	xut2up	yut2lo	yut2up
1	10	27	10	30
2	27	45	10	30
3	40	55	10	30
4	50	70	10	30

Table A8. Forbidden Connections from Water Intake *w* to Transport Unit ut1

w	ut1
1	1
1	2
2	3
2	4

The upper bound of the binary variable $ywut1_{w,ut1}$ is equal to 0.

Table A9. Forbidden Connections from Transport Unit ut1 to Plant u_p of Type p

ut1	p	u_p
1	1	3
2	1	3
3	1	1
4	1	1
1	2	3
2	2	3
3	2	1
4	2	1

The upper bound of the binary variable $yut1p_{ut1,p,up}$ is equal to 0.

Table A10. Forbidden Connections from Plant u_p of Type p to Transport Unit ut2

p	u_p	ut2
1	1	3
1	1	4
1	3	1
1	3	2
2	1	3
2	1	4
2	3	1
2	3	2

The upper bound of the binary variable $yput2_{p,up,ut2}$ is equal to 0.

Table A11. Forbidden Connections from Transport Unit ut2 to Storage Tank *s*

ut2	*s*
3	1
4	1
1	2
2	2
1	3
2	3
3	4
4	4
1	5
2	5

The upper bound of the binary variable $yut2s_{ut2,s}$ is equal to 0.

Table A12. Upper Bounds for the Distance Between Water Intakes and Transportation Units

Upper Bound	*k*	Value
$UDWUT1_{w,ut1,k}$	1	20
$UDWUT1_{w,ut1,k}$	2	12

Table A13. Upper Bounds for the Distance Between Transportation Units and Plants

Upper Bound	*k*	Value
$UDUT1P_{ut1,p,up,k}$	1	25
$UDUT1P_{ut1,p,up,k}$	2	25

Table A14. Upper Bounds for the Distance Between Plants and Transportation Units

Upper Bound	*k*	Value
$UDPUT2_{p,up,ut2,k}$	1	25
$UDPUT2_{p,up,ut2,k}$	2	25

Table A15. Upper Bounds for the Distance Between Transportation Units and Storage Tanks

Upper Bound	*k*	Value
$UDUT2S_{ut2,s,k}$	1	30
$UDUT2S_{ut2,s,k}$	2	20

Data of Problem II (Tables A16–A19)

Table A16. Number, Capacity, and Efficiencies of Plants of Different Types

p	NPP_p	CAP_p	E_p
1	3	250	0.9
2	3	500	0.85
3	8	10	0.8

Table A17. Triangle Coordinates

	P1		P2		P3	
Δ	*x*	*y*	*x*	*y*	*x*	*y*
1	0	150	0	10	35	125
2	35	125	0	10	30	20
3	33.952	103	30	20	57	103
4	38.132	45	70	25	30	20
5	38.132	45	70	25	70	45
6	57	103	49.51	80	70	80
7	0	150	35	125	250	150

Table A17. Triangle Coordinates (continued)

	P1		P2		P3	
Δ	*x*	*y*	*x*	*y*	*x*	*y*
8	35	125	100	104.35	103.8	133
9	46.658	121.296	70	80	120	98
10	70	80	120	25	100	90.8
11	70	80	70	25	120	25
12	110	57.9	120	25	138.57	70
13	138.571	70	120	25	160	20
14	130	90	160	20	200	25
15	130	90	200	25	250	15
16	130	90	200	46.25	200	90
17	130	90	180	90	180	115
18	130	136.04	130	90	250	150
19	220	135	250	115	250	150
20	220	135	220	91	250	115
21	200	90	200	75	220	91
22	200	75	240	70	250	115
23	230.487	27.195	250	15	250	115

Table A18. Coordinates of the Water Intakes

Sea Water Intake	*x*	*y*
1	200	25
2	100	25

Table A19. The Demands and Coordinates of Storage Tanks

Sea Water Intake	Demand (m^3/day)	*x*	*y*
1	250	60	60
2	300	40	110
3	350	120	80
4	100	120	110
5	80	210	60
6	50	200	100

Actual Feasible Regions and Forbidden Connections for Problem I (Tables A20–A30).

Table A20. Lower and Upper Bounds for the *x*- and *y*-Coordinates of Transport Unit ut1

ut1	xut1lo	xut1up	yut1lo	yut1up
1	50	120	30	100
2	130	200	30	100

Table A21. Lower and Upper Bounds for the *x*- and *y*-Coordinates of Plant u_p of Type *p*

p	u_p	xplo	xpup	yplo	ypup
1	1	40	210	20	100
	2	40	210	20	100
	3	40	210	20	100
2	1	40	210	20	100
	2	40	210	20	100
	3	40	210	20	100
3	1	40	210	20	100
	2	40	210	20	100
	3	40	210	20	100
	4	40	210	20	100
	5	40	210	20	100
	6	40	210	20	100
	7	40	210	20	100
	8	40	210	20	100

Table A22. Lower Bounds for the *x*-Coordinates of Transport Unit ut2

ut2	xut2lo	xut2up	yut2lo	yut2up
1	50	110	30	90
2	50	110	30	90
3	130	200	30	90
4	130	200	30	90

Table A23. Forbidden Connections from Water Intake *w* to Transport Unit ut1

w	ut1
1	1
2	2

Table A24. Forbidden Connections from Transport Unit ut1 to Plant u_p of Type *p*

ut1	*p*	u_p
1	1	3
1	2	3
2	1	1
2	2	1

Table A25. Forbidden Connections from Plant u_p of Type *p* to Transport Unit ut2

p	u_p	ut2
1	1	3
1	1	4
1	3	1
1	3	2
2	1	3
2	1	4
2	3	1
2	3	2

Table A26. Forbidden Connections from Transport Unit ut2 to Storage Tank *s*

ut2	*s*
3	1
4	1
3	2
4	2
1	5
2	5
1	6
2	6

Table A27. Upper Bounds for the Distance Between Water Intakes and Transportation Units

Upper Bound	*k*	Value
$UDWUT1_{w,ut1,k}$	1	70
$UDWUT1_{w,ut1,k}$	2	70

Table A28. Upper Bounds for the Distance Between Transportation Units and Plants

Upper Bound	*p*	u_p	*k*	Value
$UDUT1P_{ut1,p,up,k}$	1,2	1	1	30
$UDUT1P_{ut1,p,up,k}$	1,2	1	2	30
$UDUT1P_{ut1,p,up,k}$	1,2	2	1	100
$UDUT1P_{ut1,p,up,k}$	1,2	2	2	50
$UDUT1P_{ut1,p,up,k}$	1,2	3	1	30
$UDUT1P_{ut1,p,up,k}$	1,2	3	2	30

Table A29. Upper Bounds for the Distance Between Plants and Transportation Units

Upper Bound	*p*	u_p	*k*	Value
$UDPUT2_{p,up,ut2,k}$	1,2	1	1	30
$UDPUT2_{p,up,ut2,k}$	1,2	1	2	30
$UDPUT2_{p,up,ut2,k}$	1,2	2	1	100
$UDPUT2_{p,up,ut2,k}$	1,2	2	2	50
$UDPUT2_{p,up,ut2,k}$	1,2	3	1	30
$UDPUT2_{p,up,ut2,k}$	1,2	3	2	30

Table A30. Upper Bounds for the Distance Between Transportation Units and Storage Tanks

Upper Bound	*k*	Value
$UDUT2S_{ut2,s,k}$	1	80
$UDUT2S_{ut2,s,k}$	2	50

Visualized explanation of the use of triangles

It is well known from linear algebra that in Figure A1 the following relations hold:

1. In point Π2, $\Lambda21 = 1$ and $\Lambda31 = 0$.
   In point Π3, $\Lambda21 = 0$ and $\Lambda31 = 1$.
   In point Π4, $\Lambda21 = 0$ and $0 \leq \Lambda31 \leq 1$.
   In point Π5, $0 \leq \Lambda21 \leq 1$ and $\Lambda31 = 0$.
2. Any point Π6 in the plane can be obtained by adding an appropriate $\Lambda21 \cdot \overline{\Pi1\Pi2}$ vector to an appropriate point Π4 lying somewhere in the $\overline{\Pi1\Pi3}$ edge. The straight line in which the $\overline{\Pi1\Pi3}$ edge lies splits the plane into two parts. If Λ21 is nonnegative then point Π6 lies on that side of plane in which point Π2 lies, see Figure A2
3. Similarly, point Π6 in the plane can be obtained by adding an appropriate $\Lambda31 \cdot \overline{\Pi1\Pi3}$ vector to an appropriate point Π5 lying somewhere in the $\overline{\Pi1\Pi2}$ edge. The straight line in which the $\overline{\Pi1\Pi2}$ edge lies splits the plane into two parts. If Λ31 is nonnegative then point Π6 lies on that side of plane in which point Π3 lies, see Figure A3.

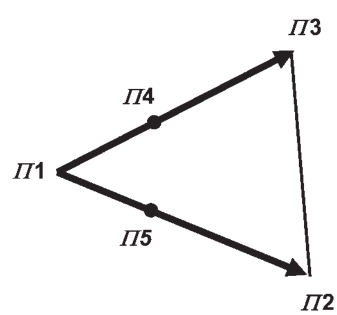

Figure A1. Coordinates of specific points.

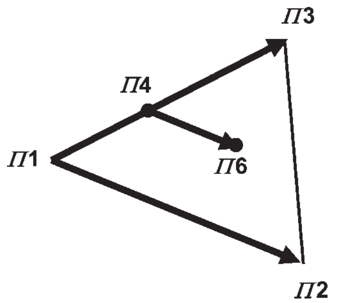

Figure A2. Representation of point Π6 inside the triangle with coordinate Λ12.

4. The point lies accurately on the $\overline{\Pi2\Pi3}$ edge if the sum of the lambda-s is equal to 1. This can be shown as follows.

Let Λ21 be equal to 1 − Λ31. Byequivalent transformations, one can obtain the following series of expressions:

$$
\begin{aligned}
P_k &= \Pi1_k + \Lambda21 \cdot (\Pi2_k - \Pi1_k) + \Lambda31 \cdot (\Pi3_k - \Pi1_k) \\
&= \Pi1_k + (1 - \Lambda31) \cdot (\Pi2_k - \Pi1_k) + \Lambda31 \cdot (\Pi3_k - \Pi1_k) \\
&= \Pi1_k + \Pi2_k - \Pi1_k - \Lambda31 \cdot \Pi2_k + \Lambda31 \cdot \Pi1_k + \Lambda31 \cdot \Pi3_k \\
&\quad - \Lambda31 \cdot \Pi1_k = \Pi2_k - \Lambda31 \cdot \Pi2_k + \Lambda31 \cdot \Pi3_k \\
&= \Pi2_k + \Lambda31 \cdot (\Pi3_k - \Pi2_k)
\end{aligned}
$$

As a result, $P_k = \Pi2_k + \Lambda31 \cdot (\Pi3_k - \Pi2_k)$, ie. P_k lies on the $\overline{\Pi2\Pi3}$ edge.

5. Any point Π6 in the plane can be obtained by adding an appropriate $L21 \cdot \overline{\Pi1\Pi3}$ vector to an appropriate point Π7 lying somewhere in the $\overline{\Pi2\Pi3}$ edge. The straight line in which the $\overline{\Pi2\Pi3}$ edge lies splits the plane into two parts. If $L21$ is nonnegative then point Π6 lies on that side of plane in which point Π1 lies, see Figure A4.

Any point Π6 in the plane can be obtained from an appropriate point Π7 point (lying somewhere on the $\overline{\Pi2\Pi3}$ edge) by an appropriate $L21 \cdot \overline{\Pi1\Pi3}$ vector. The $\overline{\Pi2\Pi3}$ edge splits the plane into two parts. If the sum of the lambdas belonging to Π6 point is less than 1, then $L21$ must be negative. If $L21$ is negative, then we are approaching the Π1 point, i.e. the Π6 point is lying on that side of plane in which the Π1 point lies.

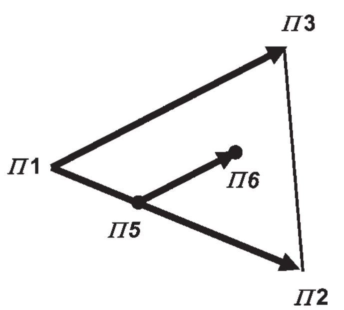

Figure A3. Representation of point Π6 inside the triangle with coordinate Λ13.

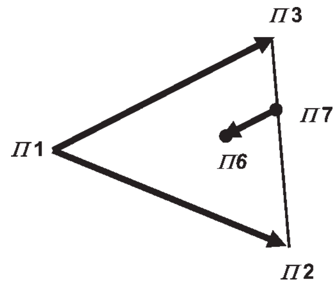

Figure A4. Points inside if sum of Λ12 and Λ13 is less than 1.

A summary is shown in Figure A5. If Λ21 is nonnegative then the point lies in the horizontal hatched area. If Λ31 is nonnegative then the point lies in the vertical hatched area. If their sum is less than 1, then the point lies in the bricked-hatched area.

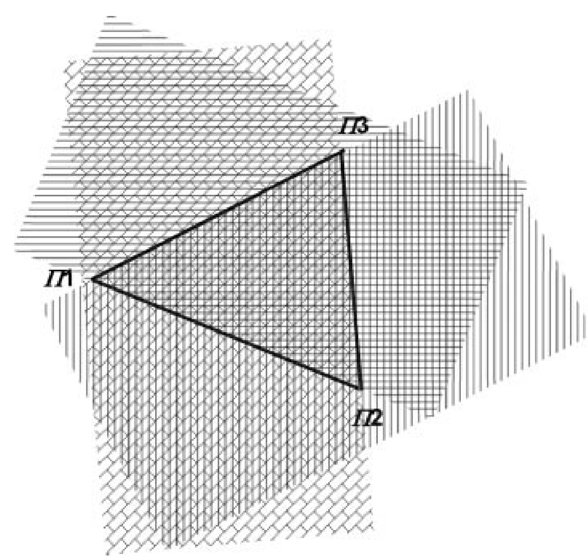

Figure A5. Visualization of overlapping the different regions belonging to the three conditions.

Manuscript received Oct. 13, 2006, and revision received Jun. 6, 2007.