PERGAMON

Computers and Chemical Engineering 23 (1998) 11–25

Computers & Chemical Engineering

A dual temperature simulated annealing approach for solving bilevel programming problems

Kemal H. Sahin, Amy R. Ciric*

Department of Chemical Engineering, University of Cincinnati, Cincinnati, OH 45221-0171, USA

Received 25 August 1997; revised 22 July 1998

Abstract

This paper presents a dual temperature simulated annealing approach to bilevel programming problems. Bilevel programming problems arise when one optimization problem, the inner problem, is a constraint of a second optimization problem, the outer problem. In this paper, the inner problem is stochastically relaxed with a parameter that can be used as a temperature scale in simulated annealing. Solving the outer problem with simulated annealing as well leads to the dual temperature approach. The technique is demonstrated with several linear, nonlinear, and mixed integer nonlinear bilevel programming problems, including a safe plant layout problem that simultaneously minimizes cost and the damage caused during a worst case scenario accident.

1. Introduction

Bilevel programming arises when one optimization problem constrains another. Applications of BLPPs are encountered in many areas that involve decision making based on multiple criteria, with applications as diverse as nuclear warfare strategy (Bracken et al., 1977) and the design of transportation networks (Yang and Yagar, 1994). In chemical engineering, Clark (1990) has applied BLPPs to optimize chemical processes by minimizing cost, while minimizing the Gibbs free energy in reactors and separators. Brengel and Seider (1992) use bilevel programming to simultaneously design a chemical process and its model predictive controller.

A general formulation of bilevel programming problems (BLPP) is shown below:

$$\begin{aligned} \min_{x} \quad & F(x, y) \\ \text{s.t.} \quad & G(x, y) \leqslant 0, \\ & H(x, y) = 0, \end{aligned} \tag{P}$$

$$\begin{aligned} \min_{y} \quad & f(x, y) \\ \text{s.t.} \quad & g(x, y) \leqslant 0, \\ & h(x, y) = 0. \end{aligned}$$

The outer objective function, $F(x, y)$, is constrained by equality constraints $H(x, y)$ and inequality constraints $G(x, y)$. In addition, the outer problem is constrained by an inner problem that minimizes $f(x, y)$ with respect to y, inequality constraints $g(x, y)$, and equality constraints $h(x, y)$. This inner problem is always active.

Many published approaches for solving BLPPs, which are NP hard (Bard, 1991) and may have Pareto optimal solutions (Clark and Westerberg, 1983), replace the inner optimization problem with its Karush–Kuhn–Tucker (KKT) conditions. This approach transforms the BLPP into a single-level optimization problem but it also introduces nonconvex terms in the constraint set. For example, inequality constraints in the inner problem lead to the nonconvex complementarity condition

$$\mu_i g_i = 0.$$

where μ_i are the Lagrange multipliers of the respective constraints g_i. In linear bilevel systems, this term is bilinear and is the only nonconvexity.

*Corresponding author.

S0098-1354/98/$ – see front matter
PII: S0098-1354(98)00267-1

Bard and co-workers have developed a branch and bound method for tackling the convexity problem caused by the complementarity conditions (Bard and Moore, 1990). The optimization is performed by removing the complementarity condition, solving this relaxed problem, and then branching by setting $\mu_i = 0$ or $g_i = 0$. This procedure is repeated until either all complementarity conditions are satisfied or the solution at the lowest branch yields infeasible points. This method has been applied to linear inner problems, inner problems with nonlinear objective functions (Edmunds and Bard, 1991) and nonlinear BLPPs with integer variables in the outer problem (Edmunds and Bard, 1992).

Clark and Westerberg (1990) have developed two methods for handling the nonconvexities caused by the complementarity condition. The first method is an active set strategy that splits the feasible region into subregions where some specific constraints are active. With this, the nonconvexity of the complementarity conditions in a subregion for inactive constraints can be eliminated by setting $\mu_i = 0$. Their second method relaxes the complementary condition and slowly tightens the relaxation.

Al-Khayyal et al. (1992) have developed an algorithm based on KKT conditions, using

$$0 = \sum_i |u_i^2 - (\max(0, u_i + g_i(x, y)))^2| + \sum_j |v_j^2 - (\max(0, v_j + h_j(x, y)))^2| \quad \forall j$$

to replace the nonconvex complementary conditions. The reformulated problem is solved with a branch and bound approach.

Visweswaran et al. (1996) noted that in linear problems, the complementarity condition has nonconvex bilinear terms, and these bilinear terms are the only nonconvexities in the transformed problem. As a result, linear BLPPs and nonlinear BLPPs with linear, bilinear and quadratic forms in the outer problem and inner objective, and linear inner constraints could be solved to global optimality with their global optimization technique (Floudas and Visweswaran, 1990, 1993).

These solution techniques are limited by the common strategy they use to transform the BLPP to a single level problem. KKT conditions cannot be applied to discontinuous functions. Further, if the inner problem is nonconvex, methods that use the KKT transformation cannot find the global optimal solution of the BLPP, even if global optimizers are used, because the KKT conditions describe local optima.

Some researchers have solved BLPPs without using the KKT formulation. Linear BLPPs can be solved with vertex enumeration methods, as optimal solutions of linear programming problems have to lie on a vertex of the solution space. Wen and Hsu (1991) give an overview of these methods in a review of linear BLPPs.

Bard (1983) has developed a grid search algorithm that is applicable to nonlinear objective functions. The algorithm determines upper and lower bounds for the objective function and performs a grid search to obtain the optimal solution that lies within these bounds.

Bard and co-workers have also addressed general mixed integer linear problems (MILP) (Moore and Bard, 1990) and MILP problems with binary variables (Bard and Moore, 1992) with an approach that uses the branch and bound method developed for integer programming problems. The method relaxes the outer problem, followed by relaxing the inner optimization and repeats this step until the complete branch and bound tree is obtained.

The above-stated approaches do not use KKT conditions. Although this is an initial advantage since these methods can be applied to discontinuous problems, they have other limitations. Clark and Westerberg (1990) have pointed out that Bard's grid search algorithm is limited to problems where the outer constraints are functions of x only. Additionally, the solution techniques for MILP and discrete BLPPs are limited to linear formulations only, as are the methods described by Wen and Hsu (1991). Since chemical engineering problems are often nonlinear, this limitation reduces the applicability of those methods.

This paper presents an alternative approach to bilevel programming that avoids the problem associated with the deterministic methods. The key idea is to stochastically relax or fuzzify the constraint on the outer problem that is formed by the inner problem. This idea lends to a dual temperature simulated annealing approach that may find near global solutions of BLPP's with discrete or nonconvex inner or outer optimization problems. Hence, the proposed approach will be used for solving various types of BLPPs, including formulations where integer variables are present in both inner and outer optimization problems.

2. A brief review of simulated annealing

Simulated annealing (Kirkpatrick et al., 1983) is a stochastic optimization method that is based on an analogy with physical annealing. Physical annealing involves heating a material to a very high temperature, then slowly cooling it. This procedure is used to bring the atoms that make up the material to their lowest energy configuration.

In simulated annealing, the objective of an optimization problem is analogous to the energy of a physical system, the variables are analogous to the position of the atoms, and the feasible region is analogous to the phase space. Introducing a temperature scale to an optimization problem makes the analogy complete.

With this analogy, one can apply Monte Carlo simulation methods, originally developed to simulate physical systems, to optimization problems. The key idea is to create a Markov chain of points in the space of optimization variables. Candidate points are generated by making a random change to the optimization variables; if the objective function decreases, the candidate point is added to the chain. If the objective function increases, the new point is accepted with probability $\Delta E/T$. As the chain grows, the distribution of energy levels increasingly resembles the Boltzmann distribution, and the Markov chain comes to resemble a series of atomic configurations sampled from a system at equilibrium. In optimization problems, this simulation strategy allows "uphill" moves, where the objective function increases. These uphill moves may enable the simulation to escape locally optimal solutions.

In simulated annealing, the system is allowed to approach equilibrium at some high temperature. After a series of candidate points are generated to form a Markov chain, the temperature is dropped. This procedure is repeated until the variation in the final solution variables of the Markov chain is negligible.

This strategy slowly squeezes out costly solutions, until only the optimum remains. It has been shown that this approach asymptotically approaches the global optimum, as the number of points in each Markov chain increases, and as the temperature is dropped more slowly.

Examples of chemical engineering applications of simulated annealing include utility systems synthesis (Maia et al., 1995), separation sequence design (Floquet et al., 1994), and heat exchanger and pipe network designs (Dolan et al., 1989). Cardoso and co-workers (1996) have solved cross-extraction and crude pipeline distribution problems using SIMPSA, a modified simulated annealing technique.

Simulated annealing is a technique that can be applied to solve NP hard and nonconvex problems. As described in the previous section, BLPPs show both characteristics. The following section illustrates how this stochastic method can be used for solving BLPPs.

3. Mathematical development

This section describes the development of the dual temperature simulated annealing (DTSA) algorithm. Some basic sets are defined, and a fuzzified form of the bilevel programming problem (P) is introduced. This fuzzy problem is used to develop a DTSA method for bilevel programming.

3.1. Sets

Before developing the algorithm it is convenient to define some sets. $\theta(x)$ is the feasible region of the inner problem of (P):

$$\theta(x) = \{y:\ y \in Y,\ g(x, y) \leqslant 0,\ h(x, y) = 0\}.$$

$\Phi(x)$ is the solution set of the inner problem of (P):

$$\Phi(x) = \{y:\ y \in \theta(x),\ f(x, y) \leqslant f(x, y^*) \forall\, y^* \in \theta(x)\}.$$

Γ is the solution set of the outer constraints:

$$\Gamma = \{(x, y):\ G(x, y) \leqslant 0,\ H(x, y) = 0\}.$$

κ is the combined feasible region of the inner and outer problems:

$$\kappa = \Gamma \cap \theta.$$

Lastly, Ω is the set of feasible solutions of problem (P):

$$\Omega = \Gamma \cap \Phi(x).$$

As shown in Fig. 1, Ω consists of the points in $\Phi(x)$ that also satisfy the outer constraint set. With these definitions, Problem (P) can be written as

$$\min_x \quad F(x, y)$$

$$\text{s.t.} \quad (x, y) \in \Gamma.$$

$$y \in \Phi(x).$$

3.2. Fuzzification of $\Phi(x)$

This work constructs a series of approximations to $\Phi(x)$. These approximations are loose when one is far from the optimum, but tighten as the optimum of P is approached.

The key idea is to stochastically relax, or "fuzzify" $\Phi(x)$, creating a fuzzy set $\Phi'(x, T_{in})$. Here, T_{in} is a measure of the looseness of the approximation. When T_{in} is large, $\Phi'(x, T_{in})$ is also quite large; when T_{in} is small, $\Phi'(x, T_{in})$ is small. Note that the fuzzy $\Phi'(x, T_{in})$ leads to a fuzzy feasible region of the overall BLPP:

$$\Omega'(T_{in}) = \Gamma \cap \Phi'(x, T_{in}).$$

With this fuzzy set, the BLPP problem can be approximated with

$$\min_x \quad F(x, y)$$

$$\text{s.t.} \quad (x, y) \in \Gamma \qquad (P'(T_{in}))$$

$$y \in \Phi'(x, T_{in}).$$

The difference between $\Phi(x)$ and $\Phi'(x, T_{in})$ is illustrated in Fig. 2. This shows $\Phi'(x, T_{in})$ for problem P', for three different values of T_{in}. Since Φ' is a fuzzy set, the feasible region is also fuzzy. The shaded areas in Fig. 2 mark regions with a 95% probability that $y \in \Phi'(x, T_{in})$.

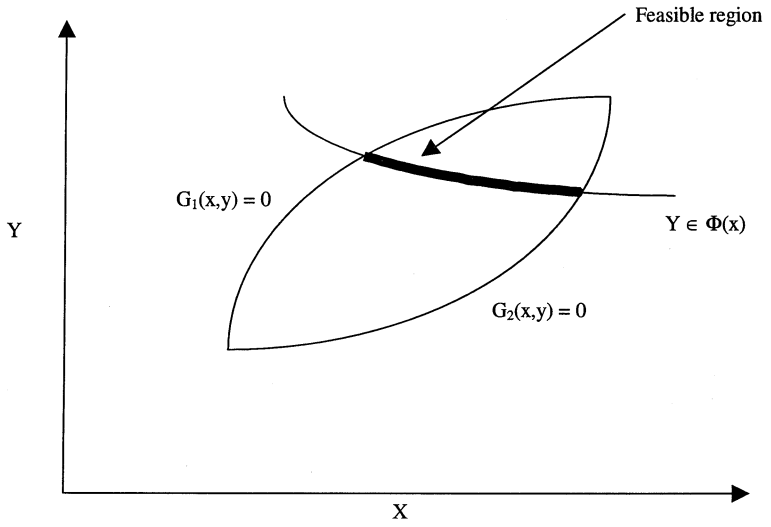

Fig. 1. Feasible region of BLPP.

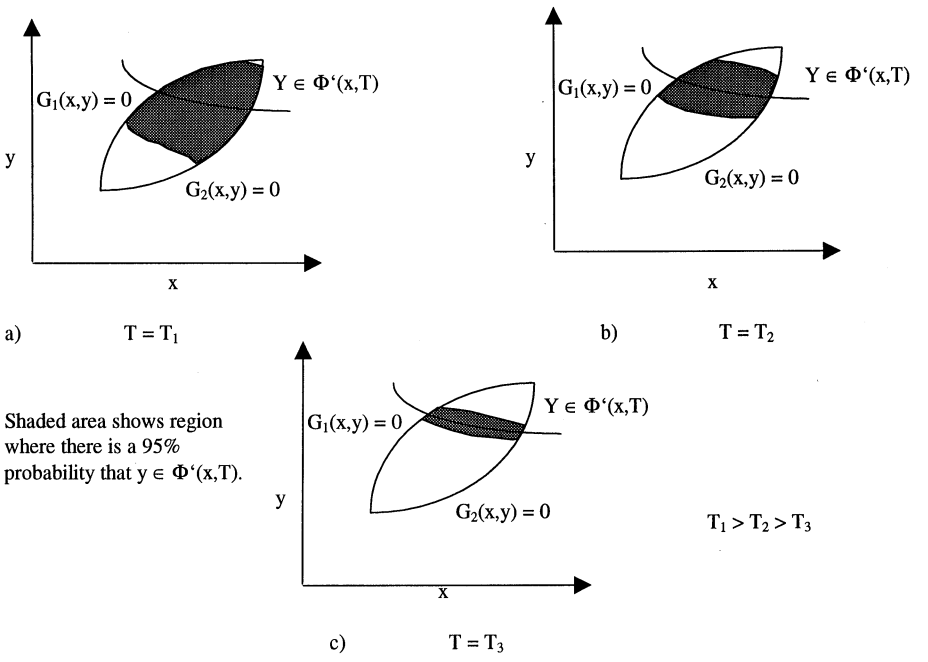

Fig. 2. Fuzzification of inner set $\phi(x)$.

Another temperature, T_{out}, is introduced to determine the optimal solution of the outer problem. This temperature is used as a simulated annealing parameter for the outer problem. Since inner searches around the fuzzy region $\Phi'(x, T_{in})$ are performed between outer searches, the method will converge to an optimal solution that lies in the narrow band $\Phi'(x, T_{in})$ for the final, low T_{in}.

The addition of a second annealing temperature for the outer problem results in the dual temperature simulated annealing method (DTSA). In the following section, the procedures used for determining the optimal solution of the BLPP using DTSA are described.

3.3. Algorithmic structure

The DTSA algorithm is a stochastic optimization method for solving bilevel programming problems. After generating an initial point, the method continues with initializing T_{out} and T_{in}. This step is followed by the optimization procedure, which searches sequentially over

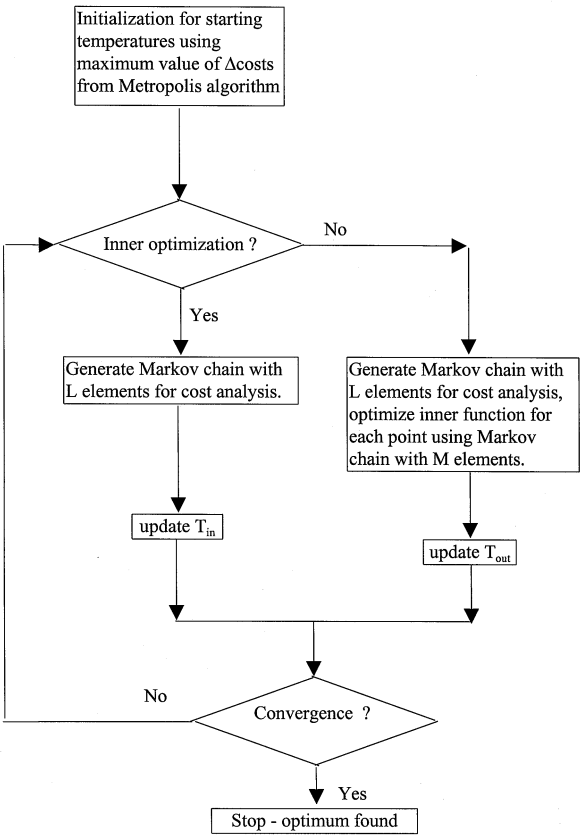

Fig. 3. Algorithm overview.

$\Phi'(x, T_{in})$ or $\Omega'(T_{in})$. After each search using Markov chains of a given length, the temperatures are updated and searches are continued until convergence criteria are met. A detailed description of steps used in the algorithm are given below, while an outline of the algorithm is given in Fig. 3.

3.3.1. Generating an initial starting point in κ.

The starting point for the DTSA algorithm must satisfy the equality and inequality constraints of both the inner and outer problems. A starting point can be found by solving the following optimization problem:

$$\min \quad \sum(r_i + s_j + t_j + u_k + v_m + w_m)$$

$$\text{s.t.} \quad G_i(x, y) \leqslant r_i,$$

$$H_j(x, y) + s_j - t_j = 0,$$

$$g_k(x, y) \leqslant u_k,$$

$$h_m(x, y) + v_m - w_m = 0,$$

$$r_i, s_j, t_j, u_k, v_m, w_m \geqslant 0.$$

If the solution of this problem is not zero, the constraints cannot be solved simultaneously and the BLPP has no solution.

3.3.2. Generating a point in θ or Γ.

New trial points that satisfy the equality and inequality constraints of either the inner or the outer problem are generated in the following way. First, consider the inner problem. The equality constraints are solved by partitioning y into subsets of dependent and independent variables, y_I and y_D. Trial values of the independent variables, y'_I, are generated by randomly perturbing the independent variables; the dependent variables y_D solve the equality constraints:

$$y'_I = y^o_I + ar,$$

$$h(x, y'_I, y'_D) = 0.$$

Here, y^o_I is the current value of the independent variables, r is a randomly generated vector with elements between -1 and 1, and a is a scalar reflecting the magnitude of the perturbation, initially set to 1. In the case of integer variables, the variable is rounded off to the nearest integer value. For discrete variables, each variable is assigned depending on the random variable r, with equal probabilities for all discrete variables.

If the point $y' = \{y'_I, y'_D\}$ satisfies the constraints $g(x, y') \leqslant 0$, then y' is in $\theta(x)$. Otherwise, a is reduced by half until the inequality constraints are satisfied. If the new value of a is 3% of the original value, following 5 reductions, a new vector r is generated.

This procedure can also be used to generate trial points x' in Γ.

3.3.3. Generating points in $\Phi'(x, T_{in})$.

Points in $\Phi'(x, T_{in})$ are generated using points y' in $\theta(x)$. These y' are added to a Markov chain if

$$f(x, y') \leqslant f(x, y)$$

or, if

$$f(x, y') \geqslant f(x, y) \quad \text{and} \quad [f(x, y') - f(x, y)]/T_{in} \leqslant \xi,$$

where ξ is a random number between zero and one. This procedure is repeated for a number of trial points and the last accepted trial point is retained.

3.3.4. Generating points in $\Omega'(T_{in})$.

Points in $\Omega'(T_{in})$ are obtained from points x' in Γ. For each point x', a point y' in $\Phi'(x', T_{in})$ is found with the procedure described above using M trial points, and the best point y^* is retained. If (x', y^*) is in κ, and

$$F(x', y^*) \leqslant F(x, y)$$

or, if

$$F(x', y^*) \geqslant F(x, y)$$

and

$$[F(x', y^*) - F(x, y)]/T_{out} \leqslant \xi,$$

(x', y^*) is accepted. If (x', y^*) lies out of κ, the value of a that was used for determining x' is doubled and a new

stochastic search is performed to find y^*. If the value of a is increased 5 times, a new vector r is generated.

3.3.5. Initial values of T_{out} and T_{in}

The initial temperatures are found by generating a sufficiently large Markov chain of trial points. These points are elements of θ or $\Phi'(x, T_{in}) \cap \Gamma$ for the inner or outer starting temperature, respectively. The maximum change in objective function values, Δ_{max} gives the starting temperatures with an acceptance ratio close to 95% as:

$$T = \ln(0.95) * \Delta_{max}.$$

If the outer initial temperature is of several orders of magnitudes larger than the inner temperature, the outer search is performed several times consecutively before inner search is performed. This avoids prematurely quenching the inner problem around a local optimum at the beginning of the optimization. One inner search is followed by $K-1$ outer searches, with

$$K = \max(2, \log(T_{out}/T_{in}))$$

being calculated following initialization. This K value is kept constant throughout the optimization. Note that when T_{out}/T_{in} is small, K is 2 and the outer search is performed once for every inner search.

3.3.6. Annealing schedule.

The inner and outer temperatures are annealed with the Aarts-Van Laarhoven (1985) cooling schedule:

$$T^{new} = T^{old}\left(1 + \frac{\ln(1+\delta)T^{old}}{3\sigma}\right)^{-1}.$$

Here, δ is a user-defined cooling parameter and σ is the standard deviation of trial points in the last Markov chain.

3.3.7. Stopping criteria.

The algorithm stops when both of the following criteria are met:

- The Euclidean norm between the point obtained after an outer search and the point obtained from the previous outer search is less than ε, and
- The following inner optimization does not change the inner objective function value or the the inner optimization variables stay within a range ε.

The DTSA algorithm is given in Appendix A.

4. Examples

The method was tested using several example functions covering different properties of BLPPs. All examples are solved on Sun Ultra 2 Workstations. Values of parameters used during optimization are given in Table 1 for four of the examples. In Table 2 the total number of Markov chains generated for inner and outer optimization is given as steps, in addition to the CPU times, and the initial and final temperatures are obtained for these examples. The sensitivity of the optimization with respect to the parameters — number of trial points in a Markov chain, cooling rate, and radius of convergence — is analyzed for Example 2, and these data are provided in Table 3. Finally, Table 4 represents the effect of different starting points for Example 4.

4.1. Example 1

This example is from Clark and Westerberg (1990).

$$\begin{aligned}
&\max_{x} \quad x + 3y_1 - 2y_2 \\
&\text{s.t.} \quad 0 \leqslant x \leqslant 8, \\
&\qquad \max_{y_1, y_2} \quad y_1 \\
&\qquad \text{s.t.} \quad 2x - y_1 - 4y_2 \geqslant -16, \\
&\qquad\qquad -8x - 3y_1 + 2y_2 \geqslant -48, \\
&\qquad\qquad 2x - y_1 + 3y_2 \geqslant 12, \\
&\qquad\qquad 0 \leqslant y_1 \leqslant 4.
\end{aligned}$$

The problem consists of linear objective functions, with one variable, x, in the outer problem and two variables, y_1 and y_2, in the inner problem. The inner

Table 1
Optimization parameters for examples

Example	δ_{out}	δ_{in}	M	L	err
Ex. 1	0.5	0.5	500	100	0.1
Ex. 3	0.5	0.5	1000	100	0.01
Ex. 4	0.1	0.1	50	5	0.01
Ex. 5	0.01	0.1	Outer: 500, inner: 40	15	0.1

Table 2
Computational performance for examples

Example	$T_{initial}$	T_{final}	K	steps	CPU time (s)
Ex. 1	$T_{out} = 115$, $T_{in} = 19$	$T_{out} = 0.867$, $T_{in} = 0.344$	2	87	44.58
Ex. 3	$T_{out} = 312$, $T_{in} = 848$	$T_{out} = 0.178$, $T_{in} = 3.597 \times 10^{-5}$	2	447	199.38
Ex. 4	$T_{out} = 31098$, $T_{in} = 10177$	$T_{out} = 18.9$, $T_{in} = 13.3$	2	165	0.28
Ex. 5	$T_{out} = 5493$, $T_{in} = 10364$	$T_{out} = 5.30$, $T_{in} = 3.14 \times 10^{-5}$	2	55899	6990.53

Table 3
Effects of optimization parameters for Example 2

M	δ	ε	CPU time (s)	Steps	Global opt
100	0.5	0.1	0.67	61	2 of 5
100	0.5	0.01	18.69	2944	0 of 5
100	0.1	0.01	15.22	1319	1 of 5
500	0.5	0.1	8.09	45	2 of 5
500	0.5	0.01	230.26	756	4 of 5
500	0.1	0.01	389.02	1255	4 of 5

optimization is linearly constrained. While the outer optimization is defined over the interval $x \in [0, 8]$, y_1 can take values between 0 and 4.

Starting at [1.000 0.000 4.000] the method converged to [4.959 3.901 2.173]. This solution is close to the global optimum [5 4 2] reported by Clark and Westerberg (1990). Tables 1 and 2 provide detailed information on the parameters used for this example and the computational performance.

4.2. Example 2

The second example, also from Clark and Westerberg (1990), involves quadratic objective functions. The inner problem has linear inequality constraints, as shown below:

$$\max_x \quad -\{(x-3)^2 + (y-2)^2\}$$

$$\text{s.t.} \quad 0 \leqslant x \leqslant 8,$$

$$\max_y \quad -(y-5)^2,$$

$$\text{s.t.} \quad 2x - y \geqslant -1,$$

$$-x + 2y \geqslant 2,$$

$$-x - 2y \geqslant -14.$$

The feasible region of the inner optimization is the triangular area in Fig. 4. The heavy lines are $\Phi(x)$, while the thin dashed lines are the boundaries of $\Gamma(x)$.

The BLPP has locally optimal solutions at $(x, y) = [1\ 3]$, $(x, y) = [3\ 5]$, and $(x, y) = [4.4\ 4.8]$. The global optimum of the BLPP lies at $(x, y) = [1\ 3]$. At this point, the inner objective function is -4, and the outer objective function is 5. The initial point was taken as the locally optimal point $(x, y) = [3\ 5]$ to test whether the approach will converge to the global solution.

The effect of the optimization parameters was analyzed with sets of 5 optimizations using different Markov chain lengths, cooling speeds, and sensitivity ranges for determining convergence. When the number of trial points for outer optimization is set to $M = 100$, $L = 10$ trial points are generated to determine a near optimal y^* value. For 500 outer trial points, L increases to 22. For both cases, runs were performed using $\delta = 0.5$ and $\varepsilon = 0.1$, $\delta = 0.5$ and $\varepsilon = 0.01$, and $\delta = 0.1$ and $\varepsilon = 0.01$. The results of these optimizations are given in Table 3. The procedure reached the global optimum 80% of the time when $M = 500$ and $\varepsilon = 0.01$. When $M = 100$, the CPU times are an order of magnitude less, but only one-fifth of the runs reached the global optimum. In this particular example, decreasing the cooling parameter from 0.1 to 0.5 had no significant effect on the convergence. Similarly, increasing ε lowers computation times, but can lead to preliminary termination at a sub-optimal solution.

An example solution for $M = 500$, $\delta = 0.1$, and $\varepsilon = 0.01$ is [0.953 2.905]. This point was determined after 189.19 seconds CPU time and 677 steps. The initial temperatures for this run were $T_{out} = 161.06$ and $T_{in} = 76.82$, while the final values were obtained as $T_{out} = 0.344$ and $T_{in} = 0.104$. As for all other runs, the value for K was 2.

4.3. Example 3

This example illustrates the effect of constraints in the outer problem.

$$\max_x \quad -\{(x-3)^2 + (y-2)^2\}$$

$$\text{s.t.} \quad 2x - y \geqslant -1,$$

$$-x + 2y \geqslant 2,$$

$$-x - 2y \geqslant -14,$$

$$0 \leqslant x \leqslant 8,$$

$$\max_y \quad -(y-5)^2.$$

Table 4
Performance results for Example 4

Starting point	Convergence to global opt (%)	Avg. CPU time (s)	Avg. distance to global opt
[1.0, 1.0, 1, 1]	80	0.75	0.14
[5.0, 5.0, 0, 1]	80	0.75	0.04
[9.0, 9.0, 1, 0]	80	0.89	0.09

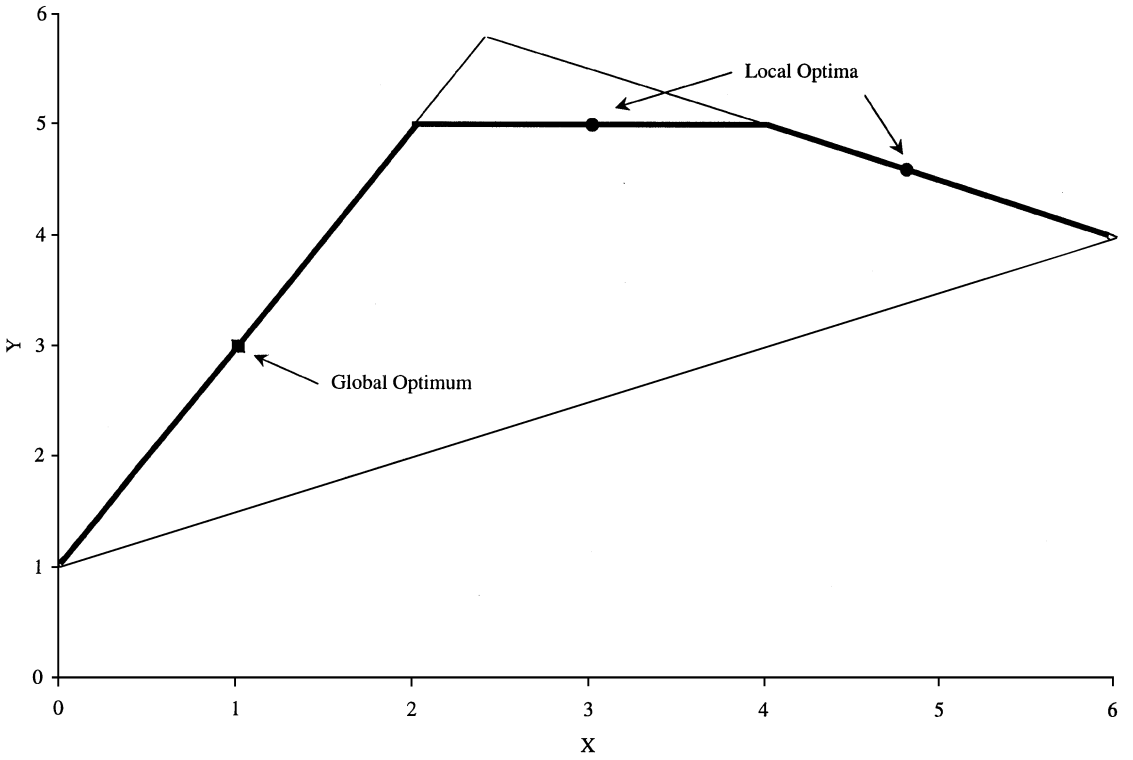

Fig. 4. Example 2.

This optimization has the same objective functions as given in Example 2, but the constraints of the inner optimization have been moved to the outer optimization.

Fig. 5 shows the feasible region for this problem. $\Phi(x)$ is the heavy line $y = 5$. Γ, the feasible solution set to the outer constraints, is indicated by the thin lines. Note that κ is a null set for $x \in [0, 2)$ and $x \in (4, 6]$, and the BLPP has no solution for those regions.

For the starting point [3.000 1.000] the algorithm converged to [3.132 5.003]. Optimization parameters and computational performance are given in Tables 1 and 2, respectively.

4.4. Example 4

This example involves integer variables that appear nonlinearly in both the inner and outer problems. Since the integer variables of the inner optimization are present in both the inner and outer optimization problems, current deterministic BLPP solution techniques cannot be applied to solve this type of problem.

$$\max_{x_1, x_2} \quad (-\tfrac{2}{5}x_1^2x_2 + 4x_2^2)(y_1)(y_2) + (-x_2^3 + 3x_1^2x_2) \times (1 - y_1)y_2 + (2x_2^2 - x_1)(1 - y_2)$$

$$\text{s.t.} \quad 0 \leqslant x_1 \leqslant 10,$$

$$0 \leqslant x_2 \leqslant 10,$$

$$\max_{y_1, y_2} \quad (x_1^2x_2^2 + 8x_2^3 - 14x_1^2 - 5x_1)y_1y_2 + (-x_1x_2^2 + 5x_1x_2 + 4x_2)(1 - y_1)y_2 + 8x_1 \cdot y1 \cdot (1 - y_2)$$

$$\text{s.t.} \quad y_1 + y_2 \geqslant 1$$

$$y_1, y_2 \in \{0, 1\}.$$

The feasible region of the problem is shown in Fig. 6. The dark lines show the borders for the solutions of the inner optimization problem. The shading is used to express the objective function value for the BLPP, where light shading is used for high numerical values. The feasible region of the BLPP is discontinuous and nonconvex due to the different values of the integer variables found from the inner optimization.

The global optimum of this problem is $[x_1\ x_2\ y_1\ y_2] =$ [6.038 2.957 0 1]. Performance of the DTSA algorithm is compared in Table 4 for starting points $[x_1\ x_2\ y_1\ y_2] =$ [1.0 1.0 1 1], [5.0 5.0 1 0], and [9.0 9.0 1 0]. The optimization is performed five times for each starting point, using the optimization parameters shown in Table 1. On average, the method converged to the global optimum 80% of the time. The average convergence time and average distance values are given only for the results where DTSA found the global solution. In Table 2, the computationally most efficient performance is reported.

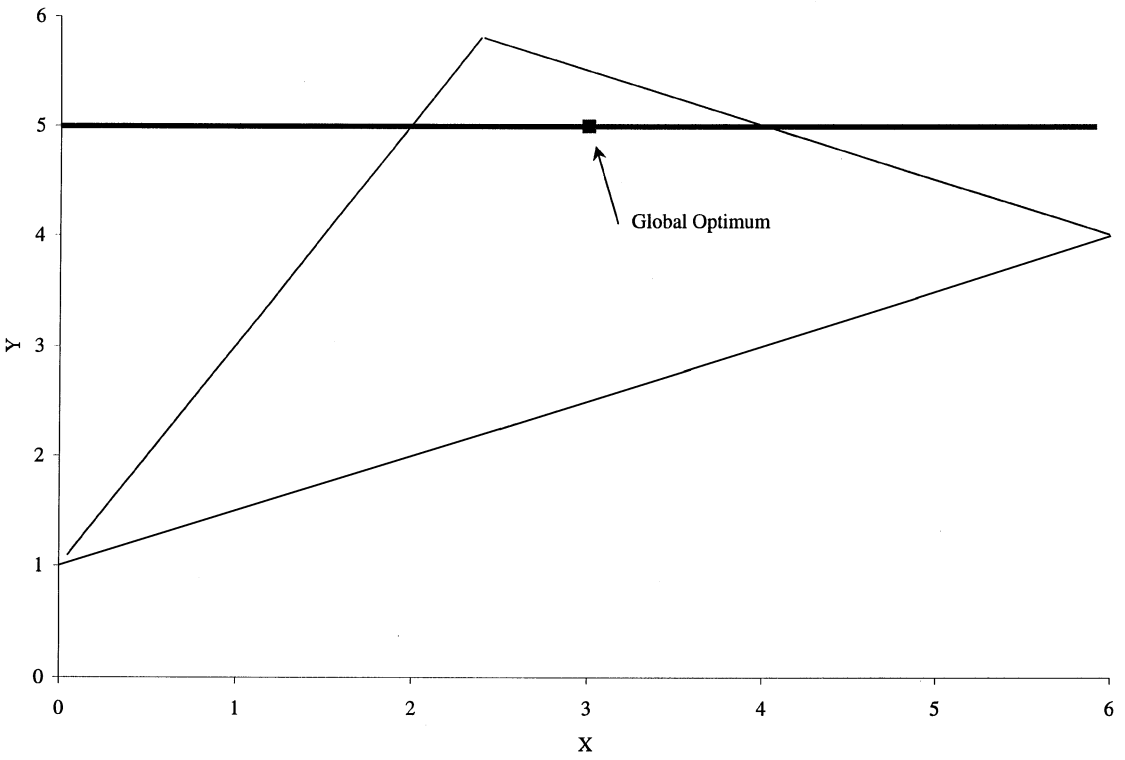

Fig. 5. Example 3.

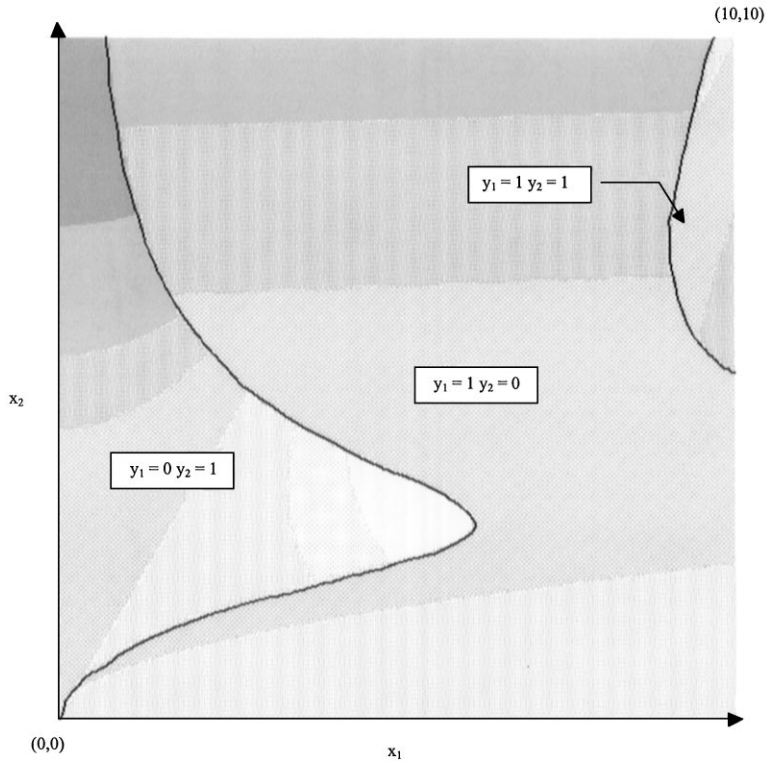

Fig. 6. Example 4.

This fastest performance converged to $[x_1\ x_2\ y_1 y_2] = [5.985\ 2.722\ 0\ 1]$ starting at [9.0 9.0 1 0]. The best performance converged to the point $[x_1\ x_2\ y_1\ y_2] = [6.012\ 2.921\ 0\ 1]$, in 1.15 s CPU time, from the starting point [5.0 5.0 0 1].

4.5. Example 5

Fig. 7 shows a 7 unit ethylene oxide plant (Penteado and Ciric, 1996). Internal disturbances may cause an explosion in one of five different units within this plant. The blast wave and shrapnel of this explosion may trigger secondary explosions in nearby units, and these secondary explosions may trigger tertiary explosions in more distant units. The likelihood of such a catastrophe and the damage it causes depends upon the process and unit operation design, the safety devices attached to each unit operation, operating practices, and the process layout.

This example uses bilevel programming to find a safe process layout for this plant. An inner optimization problem finds the worst case scenario and calculates the damage it causes, while an outer optimization problem minimizes this and other costs associated with the layout.

The BLPP formulation is:

$$\min_x \quad pc + lc + wc + \sum_i c_i\omega_i$$

$$\text{s.t.} \quad pc = \sum_i^N \sum_{j=i+1}^N d_{ij}\gamma_{ij} upc, \tag{o1}$$

$$lc = (2R_{wall})^2 ulc, \tag{o2}$$

$$wc = 2\pi R_{wall}\, uwc, \tag{o3}$$

$$d_{ij} = \sqrt{(x_{x,i} - x_{x,j})^2 + (x_{y,i} - x_{y,j})^2}, \tag{o4}$$

$$d_{ij} \geqslant d_{ij}^{min}, \tag{o5}$$

$$x_{wall} = \sum_i^N x_{x,i}/N, \tag{o6}$$

$$y_{wall} = \sum_i^N x_{y,i}/N, \tag{o7}$$

$$d_{i,wall} = \sqrt{(x_{x,i} - x_{wal})^2 + (x_{y,i} - y_{wall})^2}, \tag{o8}$$

$$R_{wall} \geqslant d_{i,wall} + fp_i \quad \forall i, \tag{o9}$$

$$0 \leqslant x_{x,i} \leqslant 200, \qquad 0 \leqslant x_{y,i} \leqslant 200, \tag{o10, o11}$$

$$\max_x \quad \sum_i c_i\omega_i,$$

$$\text{s.t.} \quad \sum_i z_i = 1. \tag{i1}$$

$$d_{ij} - r_i \leqslant U(2 - y_{ij} - \omega_j), \tag{i2}$$

$$d_{ij} - r_i \geqslant -U(1 + y_{ij} - \omega_j), \tag{i3}$$

$$y_{ij} \leqslant \omega_j, \tag{i4}$$

$$\omega_i \geqslant z_i, \tag{i5}$$

$$z_i, \omega_i, y_{ij} \in [0, 1] \tag{i6}$$

The outer optimization problem minimizes the cost of piping (pc), land (lc), a surrounding circular wall (wc), and the cost associated with a worst-case scenario accident ($\sum_i c_i\omega_i$). The position of unit i is given by two-dimensional Cartesian coordinates ($x_{x,i}$, $x_{y,i}$), which are the decision variables of the outer optimization. The distance between units i and j, d_{ij}, must be greater than some minimum d_{ij}^{min}. The land area occupied by the process is the square area surrounding the circular wall. Each unit occupies a circular footprint of radius fp.

The inner optimization problem finds the worst-case scenario. The inner optimization problem has three sets of integer variables. The decision variable of the inner

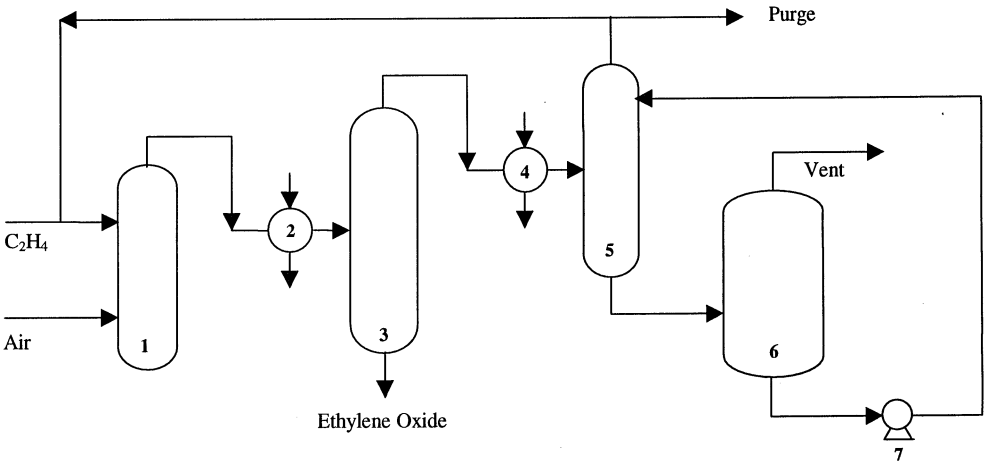

Fig. 7. Example 5, flowsheet.

optimization, z_i, is one if an internal disturbance creates an explosion at unit i, and y_{ij} is one if an explosion at unit i creates an explosion at unit j. Lastly, ω_i is one if unit i explodes, either from an internal disturbance or from a nearby explosion.

Constraint (i1) in the inner problem assures that there is only one primary explosion; constraints (i2) and (i3) set $y_{ij} = 1$ if unit j is in the blast radius r_i of unit i, and constraint (i5) sets ω_i to one if z_i is one.

If unit i explodes, then surrounding units j are damaged if the distance d_{ij} is less than the radius of propagation, r_i. Table 5 gives the process unit costs, the radii of propagation, r, and the radii of footprints, fp. The values for the parameter γ_{ij}, which is 1 if units i and j are connected by a pipe, are provided in Table 6. Costs for land, pipes and the wall are given in Table 7.

Since the worst-case scenario is a unimodal function, the approximation of the inner optimal solution is performed using $L = 40$ trials. Similarly, a faster cooling approach can be applied as shown in Table 1 to enforce the selection of the worst-case scenario and to limit the acceptance possibilities of sub-optimal propagations. The starting point for the optimization was the same initial layout used by Penteado and Ciric (1996).

Table 5
Process unit data for Example 5

Unit #	Unit	Cost of unit (k$)	r (m)	fp (m)
1	Reactor	130	9.283	2.61
2	Heat exchanger	22	6.488	5.71
3	EO absorber	304	30.434	3.84
4	Heat exchanger	8	7.120	4.24
5	CO_2 absorber	162.6	27.747	3.84
6	Flash	10	0.000	1.30
7	Pump	3	0.000	1.20

Table 8 summarizes the results obtained from DTSA and the final layout is shown in Fig. 8, where the grey

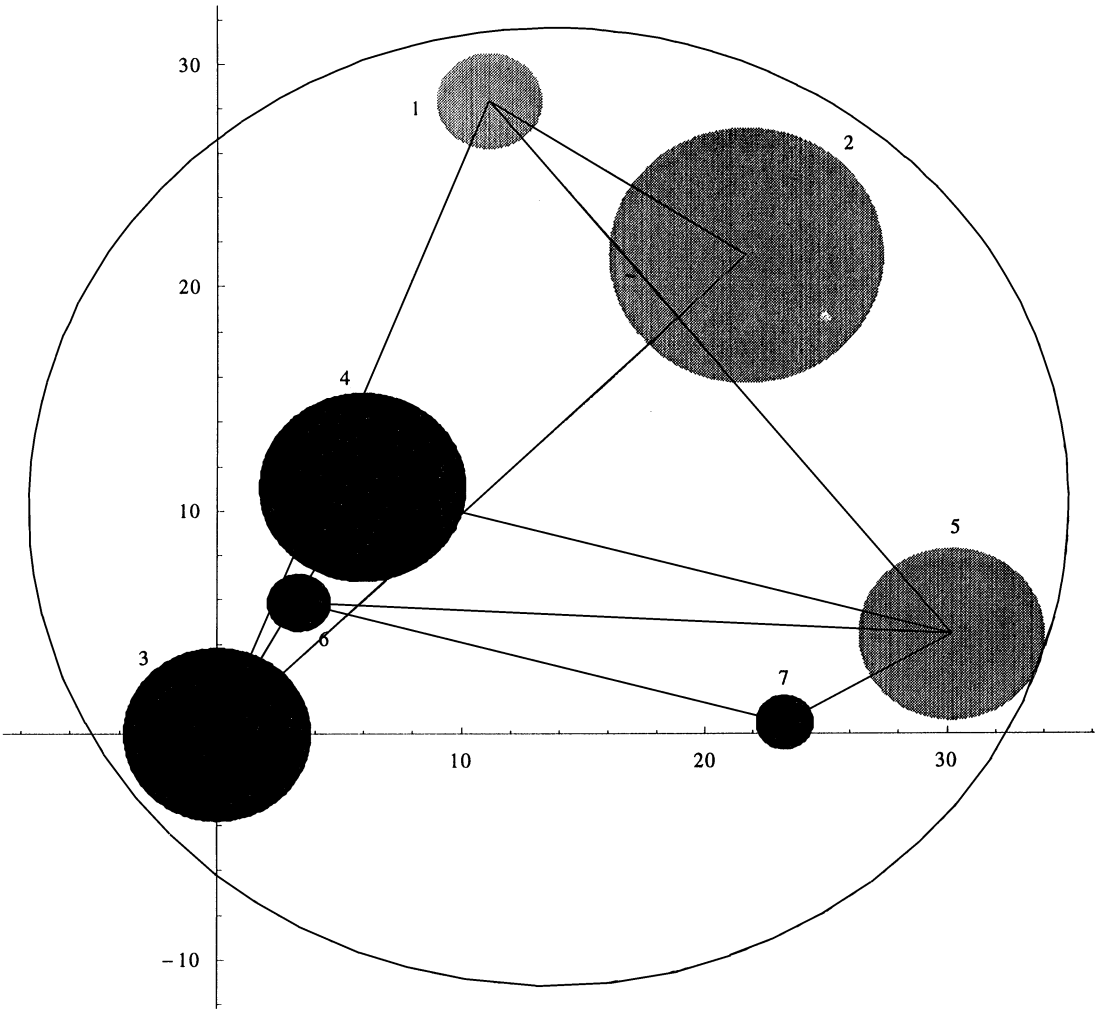

Fig. 8. Example 5, layout.

Table 6
Connectivity data for Example 5

Unit #	1	2	3	4	5	6	7
1	1	1	0	0	1	0	0
2	1	1	1	0	0	0	0
3	0	1	1	1	0	0	0
4	0	0	1	1	1	0	0
5	1	0	0	1	1	1	1
6	0	0	0	0	1	1	1
7	0	0	0	0	1	1	1

Table 7
Cost data for Example 5

Type	Cost ($)
Pipe cost per m	196.8
Land cost per sqm	67
Wall cost per m	518

Table 8
Results for Example 5

Unit #	*x* coordinates (m)	*y* coordinates (m)	Explode
1	11.13	28.33	No
2	21.71	21.41	No
3	0.00	0.00	Yes
4	5.94	11.05	Yes
5	30.20	4.44	No
6	3.33	5.86	Yes
7	23.20	8.51	Yes

units are out of the area of propagation for the accident. In the worst case scenario, unit 3 explodes, causing secondary explosions in units 4, 6, and 7. Unit 6 is positioned close to unit 3 in order to decrease the radius of the blast wall. Replacing the equipment would cost $225,000. The total cost associated with the layout is $456,000.

5. Conclusions

This paper presented a new method for solving bilevel programming problems. The inner optimization problem is stochastically relaxed and the tightness of this relaxation is controlled by a parameter that can be used as a simulated annealing temperature. The outer optimization is solved by simulated annealing, creating a dual temperature simulated annealing approach. This stochastic approach to bilevel programming problems can solve problems that cannot be solved using conventional optimization techniques.

The method was demonstrated on a number of linear, nonlinear and mixed integer nonlinear programming problems, as well as a safe plant layout design problem.

As seen in Example 2 the optimization parameters effect both the success of the algorithm as well as the CPU times. It is possible that the performance of the method is further improved by determining parameter ranges for rapid convergence.

Insights gained from this algorithm are currently being used to develop efficient deterministic techniques for identifying economically optimal and stable chemical plants.

Acknowledgements

The authors gratefully acknowledge financial support from the NSF under grant CTS-9408694. Additional funding was provided by the University Research Council of the University of Cincinnati.

Appendix A. Algorithm

A flowsheet of this algorithm is given in Figs 3, 9–11.

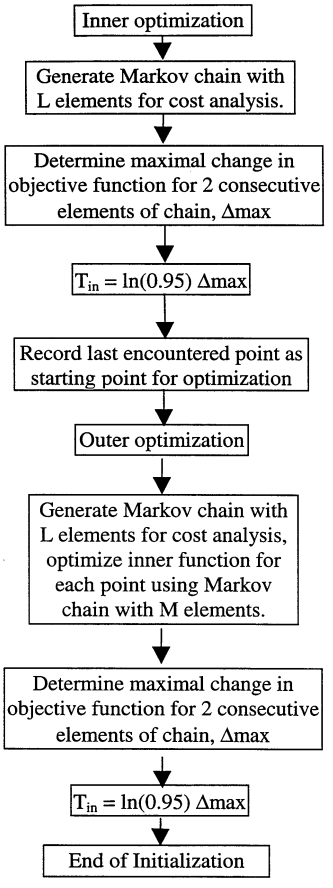

Fig. 9. Algorithm — initialization.

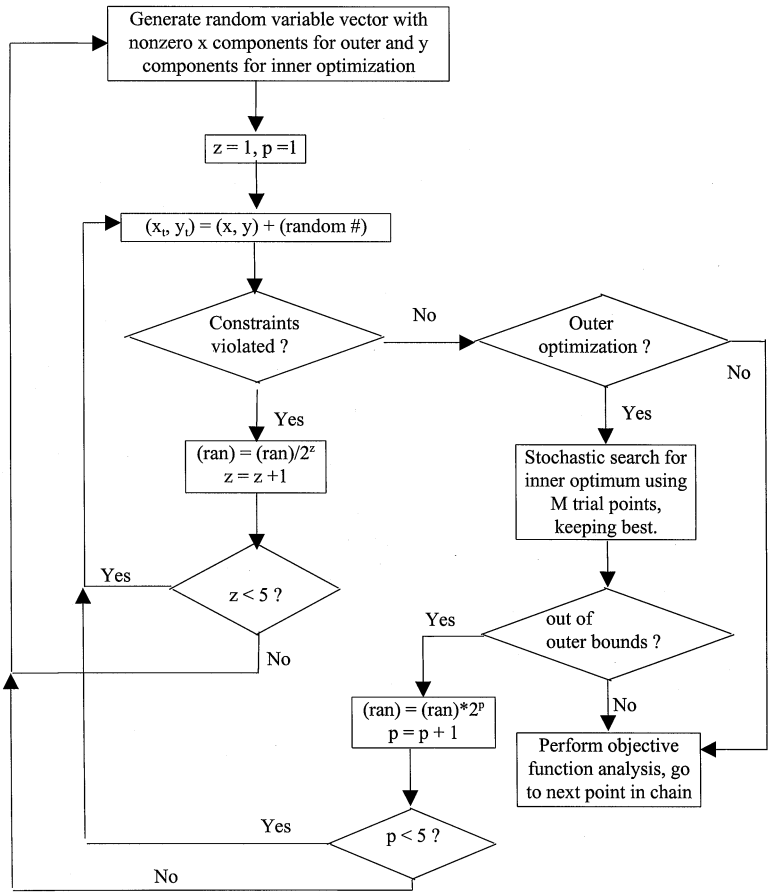

Fig. 10. Algorithm — optimization.

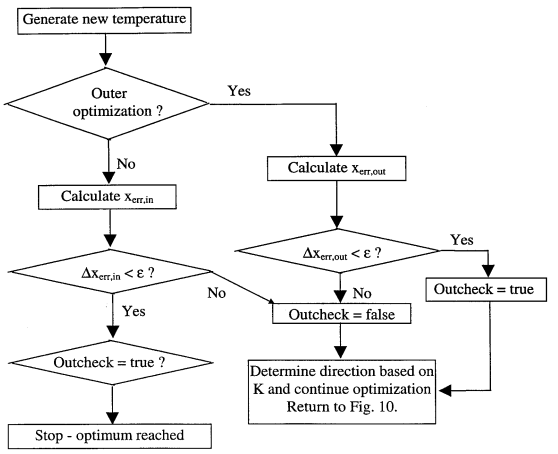

Fig. 11. Algorithm — convergence criteria.

1. Starting Point
   Identify a feasible starting point (x, y), either by inspection or by solving the optimization problem

$$\min \quad \sum (r_i + s_j + t_j + u_k + v_m + w_m)$$

$$\text{s.t.} \quad G_i(x, y) \leqslant r_i,$$

$$H_j(x, y) + s_j - t_j = 0,$$

$$g_k(x, y) \leqslant u_k,$$

$$h_m(x, y) + v_m - w_m = 0,$$

$$r_i, s_j, t_j, u_k, v_m, w_m \geqslant 0.$$

   If this problem is infeasible, STOP — the BLPP is also infeasible. Otherwise, go to step 2.
2. Initialization
   (a) Define ε, δ, set count = 1
   (b) For $i = 1..L$, do the following:
      i. Set $delmax = -\infty$
      ii. Obtain trial point (x_t, y_t) as explained in Section 3.3.2 for inner optimization.
      iii. $f_i = f(x_t, y_t)$
      iv. $delmax = \max(delmax, (f_i - f(x, y)))$
   (c) $T_{in} = \ln(0.95) \cdot delmax$
   (d) Store (x, y) as starting point for Step 3. Record (x, y) as $(x_{out,old}, y_{out,old})$ and $(x_{in,old}, y_{in,old})$
   (e) For $i = 1..L$, do the following:
      i. Set $delmax = -\infty$
      ii. Obtain trial point (x_t, y_t) as explained in Section 3.3.4 for outer optimization.

iii. $F_i = F(x_t, y_t)$
iv. $delmax = \max(delmax, (F_i - F(x, y)))$

(f) $T_{out} = \ln(0.95) \cdot delmax$
(g) Let $K = \max(2, \log(T_{out}/T_{in}))$

3. Optimization at temperature T
(a) If (count mod K) $\neq 0$ then go to 3(b), else go to 3(c)
(b) *Outer optimization*
i. For $i = 1..L$, do the following:
A. Obtain trial point (x_t, y_t) as explained in Section 3.3.4 for outer optimization.
B. If $F_i \leqslant F(x, y)$ or ran# $> (F_i - F(x, y))/T$, let $(x, y) = (x_t, y_{t,in,opt})$. Record $F_i = F(x_t, y_{t,in,opt})$. Let $F_i = 0$ if the point was not accepted.
ii. Calculate the variance, σ^2_{out}, of nonzero terms in F_i.
iii. Calculate $T^{new}_{out} = T^{old}_{out}\left(1 + \frac{\ln(1+\delta)T^{old}}{3\sigma_{out}T^{old}}\right)^{-1}$.
iv. count = count + 1
(c) *Inner optimization*
i. For $i = 1..L$, do the following:
A. Define trial point (x_t, y_t) as explained in Section 3.3.2 for inner optimization.
B. If $f_i \leqslant f(x, y)$ or ran# $> (f_i - f(x, y))/T$, let $(x, y) = (x_t, y_t)$. Record $f_i = f(x_t, y_t)$ Let $f_i = 0$ if the point was not accepted.
ii. Calculate the variance, σ^2_{in}, of nonzero terms in f_i.
iii. Calculate $T^{new}_{in} = T^{old}_{in}\left(1 + \frac{\ln(1+\delta)T^{old}}{3\sigma_{in}T^{old}}\right)^{-1}$
iv. count = count + 1

4. Convergence criteria
(a) If ((count − 1) mod $K \neq 0$ then go to 4(b) else go to 4(c)
(b) Outer optimization
i. Let $x_{err,out} = [[x\,y] - [x\,y]_{out,old}] \cdot [[x\,y] - [x\,y]_{out,old}]^T$
ii. Set $(x, y)_{out,old} = (x, y)$
iii. If $x_{err,out} \leqslant \varepsilon$, set *outcheck* = true, incheck = false, $minin = f(x, y)$
(c) Inner optimization
i. Let $x_{err,in} = [[y] - [y]_{in,old}] \cdot [[y] - [y]_{in,old}]^T$
ii. Set $(x, y)_{in,old} = (x, y)$
iii. If $x_{err,in} \leqslant \varepsilon$ or $f(x, y) = minin$, set *incheck* = true
(d) If *incheck* = true and *outcheck* = true then STOP — optimum found. Else set *optcheck* = false, go to Step 3.

Nomenclature

a	scaling factor
$F(x, y)$	outer objective function
$f(x, y)$	inner objective function
$G(x, y)$	outer equality constraints
$g(x, y)$	inner equality constraints
$H(x, y)$	outer inequality constraints
$h(x, y)$	inner inequality constraints
K	optimization order, number of consecutive outer searches prior inner searches
L	number of trial points at each temperature
M	number of trial points for inner stochastic search during outer search
r	Random disturbance vector
T_{in}	inner simulated annealing temperature and fuzzification parameter
T_{out}	outer simulated annealing tempreature
x	outer decision variable
y	inner decision variable
y_D	dependent inner variables
y_I	independent inner variables
x', y'	trial points
y^*	optimal y value for given x
r_i	slack variables for $G(x, y)$
s_j	slack variables for $H(x, y)$
t_j	slack variables for $H(x, y)$
u_k	slack variables for $g(x, y)$
v_m	slack variables for $h(x, y)$
w_m	slack variables for $h(x, y)$

Greek letters

δ	cooling parameter
ε	convergence limit
μ	Lagrange multiplier for KKT conditions
κ	combined feasible region of the inner and outer problems
Γ	solution set of the outer constraints
$\theta(x)$	feasible region of the inner problem
$\Phi(x)$	solution set of the inner problem
$\Phi'(x, T_{in})$	fuzzified solution of inner problem
Ω	feasible solutions of BLPP
$\Omega'(T_{in})$	fuzzified solutions of BLPP
σ	standard deviation of Markov chain obtained from last search

Notation for Example 5

c_i	cost of unit i
d_{ij}	distance between units
d^{min}_{ij}	minimum distance between units i and j
$d_{i,wall}$	distance between unit i and wall
r_i	blast radius for unit i
fp_i	Footprint of unit i
lc	land cost
N	number of units
pc	piping cost
R_{wall}	radius of blast wall
ulc	land cost per unit area

upc	pipe cost per unit length
uwc	wall cost per unit length
wc	wall cost
$x_{x,i}$	x coordinate for unit i
$x_{y,i}$	y coordinate for unit i
x_{wall}	x coordinate for center of wall
y_{wall}	y coordinate for center of wall
y_{ij}	integer parameter identifying connection between units i and j
z_i	integer variable associated with unit initiating explosions

Subscripts

i, j	index on units

Greek letters

γ_{ij}	integer parameter for connection between units i and j
ω_i	integer variable associated with exploding units

References

Aarts, E. H., & van Laarhoven, P. J. M. (1985). Statistical cooling: a general approach to combinatorial optimization problems. *Philips J. Res.*, *40*, 193–226.

Al-Khayyal, F. A., Horst, R., & Pardalos, P. M. (1992). Global optimization of concave functions subject to quadratic constraints: an application in nonlinear bilevel programming. *Ann. Oper. Res.*, *34*, 125–147.

Bard, J. F. (1983). An algorithm for solving the general bilevel programming problem. *Math. Oper. Res.*, *8*, 260–272.

Bard, J. F. (1991). some properties of the bilevel programming problem. *J. Optim. Theory Appl.*, *2*, 371–378.

Bard, J. F., & Moore, J. T. (1990). A branch and bound algorithm for the bilevel programming problem. *SIAM J. Sci. Stat. Comput. 11*, 281–292.

Bard, J. F., & Moore, J. T. (1992). An algorithm for the discrete bilevel programming problem. *Naval Res. Logistics*, *39*, 419–435.

Bracken, J., Falk, J. E., & Miercort, F. A. (1977). A strategic weapons exchange allocation model. *Oper. Res. 25*, 968–976.

Brengel, D. D., & Seider, W. D. (1992). Coordinated design and control optimization of nonlinear processes. *Comput. Chem. Engng*, *16*, 861–886.

Cardoso, M. F., Salcedo, R. L., & de Azevedo, S. F. (1996). Simplex-simulated annealing approach to continuous non-linear optimization. *Comput. Chem. Engng*, *20*, 1065–1080.

Clark, P. A. (1990). Bilevel programming for chemical process design — II. Performance study for nondegenerate problems. *Comput. Chem. Engng*, *14*, 99–109.

Clark, P. A., & Westerberg, A. (1983). Optimization for design problems having more than one objective. *Comput. Chem. Engng*, *7*, 259–278.

Clark, P. A., & Westerberg, A. (1990). Bilevel programming for chemical process design — I. Fundamentals and algorithms. *Comput. Chem. Engng 14*, 87–97.

Dolan, W. B., Cummings, P. T., & LeVan, M. (1989). Process optimization via simulated annealing: application to network design. *A.I.Ch.E. J. 35*, 725–736.

Edmunds, T. A., & Bard, J. F. (1991). Algorithms for nonlinear bilevel mathematical programs. *IEEE Trans. Systems, Man, Cybernet.*, *21*, 83–89.

Edmunds, T. A., & Bard, J. F. (1992). An algorithm for the mixed-integer nonlinear bilevel programming problem. *Ann. Oper. Res.*, *34*, 149–162.

Floquet, P., Pibouleau, L., & Domenech, S. (1994). Separation sequence synthesis: how to use simulated annealing procedure?. *Comput. Chem. Engng*, *18*, 1141–1148.

Floudas, C. A., & Visweswaran, V. (1990). A global optimization algorithm GOP for certain classes of nonconvex NLPs. I — Theory, *Comput. Chem. Engng*, *14*, 1397–1417.

Floudas, C. A., & Visweswaran, V. (1993). Primal-relaxed dual global optimization approach. *JOTA*, *78*, 187–225.

Kirkpatrick, S., Gelatt, J., & Vecci, M. P. (1983). Optimization by simulated annealing. *Science*, *220*, 671–680.

Maia, L. O. A., de Carvalho, L. A. V., & Oassim, R. Y. (1995). Optimization for design problems having more than one objective. *Comput. Chem. Engng*, *19*, 481–488.

Moore, J. T., & Bard, J. F. (1990). The mixed integer linear bilevel programming problem. *Oper. Res.*, *38*, 911–921.

Penteado, F. D. d. B., & Ciric, A. R. (1996). An MINLP approach for safe process plant layout. *Ind. Engng Chem. Res.*, *35*, 1354–1361.

Visweswaran, V., Floudas, C. Ierapetritou, M. & Pistikopoulos, E. (1996). A decomposition based global optimization approach for solving bilevel linear and nonlinear quadratic programs. In: Floudas, C. & Pardalos, P. (Eds). *State of the art in global optimization: computational methods and applications*, Vol. 7 of *Book series on nonconvex optimization and its applications.* (139–162). Dordrecht: Kluwer Academic Publishers.

Wen, U-P., & Hsu, S.-H. (1991). Linear bi-level programming problems — A review. *J. Opt. Res. Soc.*, *42*, 125–133.

Yang, H., & Yagar, S. (1994). Traffic assignment and traffic control in general freeway-arterial corridor systems. *Transp. Res. B*, *28B*, 463–486.