Optimal Sensor Placement and Maintenance Strategies for Mass-Flow Networks

Chao-An Lai and Chuei-Tin Chang*

Department of Chemical Engineering, National Cheng Kung University, Tainan, Taiwan 70101, Republic of China

Chin-Leng Ko and Cheng-Liang Chen

Department of Chemical Engineering, National Taiwan University, Taipei, Taiwan 10617, Republic of China

A mathematical programming model has been developed in this study to determine the best measurement locations in a given process network and also the optimal numbers of redundant and spare sensors used in a corrective maintenance program. The model solution yields the maximum system availability under a set of user supplied limitations on life-cycle cost and/or estimator's precision. Genetic algorithms were used to identify the optimum in an evolutionary process. The usefulness of the proposed approach is demonstrated with extensive case studies.

Introduction

Accurate estimators of key process variables are considered to be essential for assessing the performance of any chemical plant. To achieve a high degree of precision and/or reliability in data reconciliation, one of the main emphases in the past has been concerned with the optimal placement of sensors. Vaclavek and Loucka[1] first proposed a design technique to ensure variable observability on the basis of graph theory. Kretsovalis and Mah[2] developed a combinatorial search algorithm to enhance the precision of reconciled results. Capital cost was later adopted as the objective function of the sensor-network optimization problem.[3] Ali and Narasimhan[4] suggested that system reliability should also be considered as an important design criterion in identifying the sensor locations. They developed a computational procedure to determine the *estimation reliability* of each process variable and chose the smallest among them as the system reliability. This approach was then extended to redundant networks and bilinear networks.[5,6] The above sensor location goals were incorporated into various MINLP formulations by Bagajewicz and co-workers.[7−10] These optimization problems can be solved easily with implicit tree-type enumeration techniques,[7] as well as genetic algorithms.[11] Raghuraj et al.[12] added yet another constraint, i.e., the fault diagnostic observability, in the design of sensor networks. Bhushan and Rengaswamy[13] developed a corresponding digraph-based sensor placement procedure. Finally, a comprehensive review of the mathematical models used for the design of sensor networks can be found in Bagajewicz.[14]

In addition to the problem of sensor placement, it is apparent that issues in stipulating an appropriate maintenance policy should also be addressed to improve the availability and/or precision of the measurement systems. In the literature, only Sanchez and Bagajewicz[15] tried to analyze the impact of integrating a simple corrective maintenance program in the design of flow sensor networks. Although they obtained reasonable results in that study, it should be noted that discussions of other maintenance practices have not been included at all. Notice that, in many industrial plants, important operation goals such as safety and quality can often be closely linked to the availability and precision of monitoring systems. It is thus the intention of our study to incorporate the possibilities of redundant and spare hardware in a *corrective* maintenance model and also to develop a mathematical program for generating the best sensor locations and their maintenance strategy simultaneously in a *mass-flow* network.

The rest of this paper is organized as follows. To facilitate later discussions, the structure of our problem formulation is first presented explicitly. The computational approach used to evaluate system availability is then briefly outlined. Next, a comprehensive corrective maintenance program, including options for repair and replacement, is described in detail. On the basis of the corresponding state-space diagram, a general Markov model can be derived to describe the sensor states associated with each process stream. The expected costs of repair and replacement and the life-cycle cost of the sensor network can also be determined accordingly. The usefulness of the proposed mathematical program is demonstrated with an application example at the end of this paper.

Problem Formulation

In the proposed mathematical programming model, the system availability is always used as the objective function to be maximized. Notice that system reliability is not considered here because availability is a more appropriate performance measure in the development of maintenance policies. Two types of inequalities are allowed in this model. In particular, the life-cycle cost of the sensor network and the estimator's precision of a designated set of process streams can be constrained. The general structure of the problem formulation can

* To whom correspondence should be addressed. Tel.: 886-6-275-7575 ext. 62663. Fax: 886-6-234-4496. E-mail: ctchang@mail.ncku.edu.tw.

10.1021/ie020567j CCC: $25.00
Published on Web 08/15/2003

be written as

$$\max A^{S} \tag{1}$$

subject to

$$\mathrm{LCC} \leq C_{T} \tag{2}$$

$$\sigma_p \leq \sigma_p^* \qquad \forall p \in \tilde{\mathbf{P}} \tag{3}$$

where A^S is the system availability; LCC and C_T represent the life-cycle cost and its highest allowable limit, respectively; and σ_p and σ_p^* denote the standard deviation of the flow-rate estimate of stream p and its upper bound, respectively. Notice that the parameters C_T and σ_p^* and also the streams in set $\tilde{\mathbf{P}}$ should be supplied by the designer in advance. Here, $\tilde{\mathbf{P}} \subset \mathbf{P}$, and $\mathbf{P}$ is the set of all process streams.

The design variables of this problem are integers. They can be classified into three vectors

$$\mathbf{l} = [l_1, l_2, ..., l_N]^T$$

$$\mathbf{m} = [m_1, m_2, ..., m_N]^T$$

$$\mathbf{n} = [n_1, n_2, ..., n_N]^T$$

where N is the total number of the process streams. The variables l_p, m_p, and n_p denote the sensor type and the numbers of purchased and installed sensors, respectively, for stream p. Note that all of these variables should be greater than or equal to zero and that $m_p \geq n_p\ \forall p \in \mathbf{P}$. Furthermore, if stream p is not equipped with on-line sensors, then it is assumed in this study that $l_p = m_p = n_p = 0$.

Estimation Availability

Without considering the options to repair and/or replace sensors, Ali and Narasimhan[4] suggested that the smallest value among the estimation *reliabilities* of all process streams can be treated as a performance measure of the sensor network. The same approach is taken in this study to determine system availability for the development of optimal maintenance policy. In other words

$$A^S = \min_{p \in \mathbf{P}} A_p \tag{4}$$

where A_p is the estimation availability of stream p, and it can be computed according to the following equation

$$A_p = \begin{cases} A_p^I & \text{if stream } p \text{ is unmeasured} \\ 1 - (1 - A_p^I)(1 - A_p^D) & \text{if stream } p \text{ is measured} \end{cases} \tag{5}$$

where A_p^D and A_p^I denote the direct and indirect estimation availabilities, respectively. The direct estimation availability A_p^D is dependent on the failure rate(s) and also the maintenance policy of the sensor(s) installed on stream p. Its computation procedure is described in detail in the following sections. On the other hand, the indirect availability A_p^I is a function of the direct estimation availabilities of the streams in cut sets containing stream p. This is due to a theorem stating that all of the different ways of indirectly estimating the mass flow of stream p are given by the cut sets that contain stream p in which the mass flow of every other stream is measured.[5] Let $\boldsymbol{S_1}$, $\boldsymbol{S_2}$, ..., $\boldsymbol{S_M}$ be the stream sets obtained by removing stream p from such cut sets. Then the availability of valid data for estimating stream p indirectly can be determined on the basis of

$$A_p^I = \Pr\{\boldsymbol{S_1} \cup \boldsymbol{S_2} \cup \cdots \cup \boldsymbol{S_M}\} \tag{6}$$

Because an evaluation algorithm of this probability has been well documented by Ali and Narasimhan,[5] the detailed computation steps are not illustrated in this paper for the sake of brevity.

Maintenance Policy

As mentioned previously, a change in the sensor maintenance policy can have a significant impact on the direct estimation availability. Assuming a constant failure rate λ, it can be shown that A_p^D for a *nonrepairable* sensor is exponentially distributed over time,[16] i.e.

$$A_p^D(t) = e^{-\lambda t} \tag{7}$$

If the simplest corrective maintenance policy is adopted, i.e., the sensor is repaired only after it fails, then the sensor availability can be treated as the direct estimation availability of stream p. Specifically

$$A_p^D(t) = \frac{\mu}{\lambda + \mu} + \frac{\lambda}{\lambda + \mu} e^{-(\lambda+\mu)t} \tag{8}$$

Here, the repair rate μ is also assumed to be constant. The derivation of this equation can be found in a standard textbook, e.g., Henley and Kumamoto.[16] If $\mu \gg \lambda$ and $t \gg 1/\mu$, the time-variant availability can be approximated with a constant steady-state value,[16] i.e.

$$A_p^D(t) \approx \frac{\mu}{\lambda + \mu} \tag{9}$$

In this study, a more comprehensive corrective maintenance program is adopted. In particular, both on-line redundant and off-line spare sensors are allowed to improve the system availability and also the estimator's precision. This maintenance strategy can be summarized as follows:

1. A total of m_p sensors are purchased for measuring the flow rate of stream p. Among them, n_p ($n_p \leq m_p$) redundant sensors are installed on-line. The remaining $m_p - n_p$ sensors are stored *off-line* and treated as spares. It is assumed that a normal spare sensor can never fail.
2. If an on-line sensor fails and at least one off-line sensor is functional, then the former is replaced with a spare. The failed sensor is taken off-line and then placed in a queue for repair. This practice is due to the assumption that replacement is much faster than repair.
3. The repair process for the failed *off-line* sensors is in effect only when all on-line sensors are normal. It is assumed that these failed sensors can only be repaired one-by-one in sequence.
4. The repair process for the failed *on-line* sensors is in effect if none of the off-line sensors are functional. It

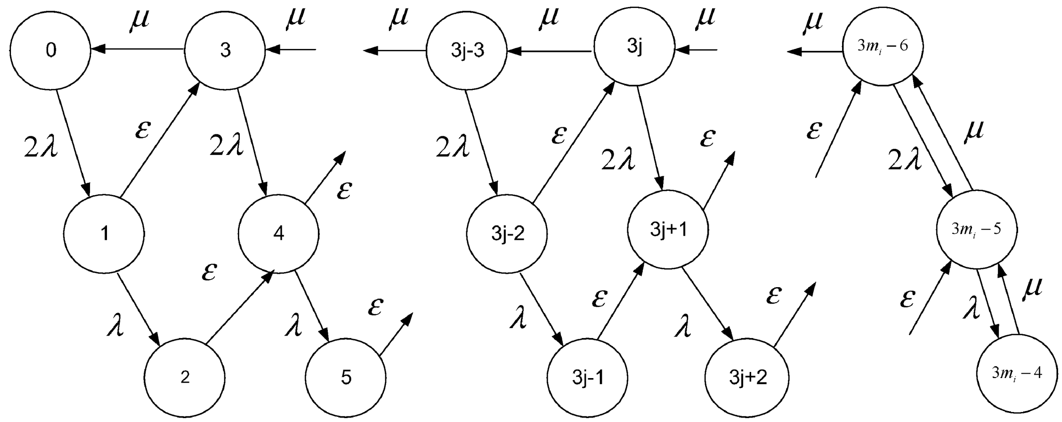

Figure 1. Markov diagram of a general corrective maintenance program ($m_p \geq n_p = 2$).

is again assumed that these failed sensors can only be repaired one-by-one in sequence.

General Markov Model

The state-space method is adopted in this work for availability evaluation. All sensors used for the measurement of a particular process stream can be characterized by their states and by the possible transitions among these states. A state-space diagram or Markov diagram can be used for the representation of these transition processes. For illustration convenience, let us consider the case when $m_p \geq n_p = 2$. The corresponding Markov diagram can be found in Figure 1. Notice that each node in this diagram represents a distinct system state and each state can be characterized by the numbers of failed (or functional) sensors on-line and also off-line. Specifically, the definitions of the states in this model are given below:

State $3j$ ($j = 0, 1, 2, ..., m_p - n_p$): All on-line sensors are normal, i.e., the number of failed on-line sensors is zero. Among the $m_p - n_p$ off-line sensors, j of them are out of order, but the rest are functional.

State $3j + 1$ ($j = 0, 1, 2, ..., m_p - n_p$): One of the on-line sensors is not working. The conditions of the off-line sensors are the same as those in state $3j$, i.e., the number of failed off-line sensors is j.

State $3j + 2$ ($j = 0, 1, 2, ..., m_p - n_p$): All on-line sensors are broken, i.e., the number of failed on-line sensors is 2. The number of failed off-line sensors is again j.

Notice that the transition rates are marked next to the arcs connecting the states. In particular, λ, μ, and ϵ denote the failure rate, repair rate, and replacement rate, respectively, of a single sensor. The following observations can be made from this Markov diagram:

1. Sensor failure can occur when the system is in states $3j$ and $3j + 1$ for $j = 0, 1, 2, ..., m_p - n_p$.

2. Sensor replacement can be done under the condition that the system state is $3j + 1$ or $3j + 2$ for $j = 0, 1, 2, ..., m_p - n_p - 1$.

3. Sensor repair can be carried out only if the system state is $3(m_p - n_p) + 1$, $3(m_p - n_p) + 2$, or $3j$ for $j = 0, 1, 2, ..., m_p - n_p$.

Notice that the second maintenance policy described in the previous section is reflected in item 2 above, and the third and fourth policies are consistent with item 3. Item 1 is trivial as a failure can only occur when at least one on-line sensor is functional. Notice also that the failure rate associated with the transition from state $3j$ to state $3j + 1$ is 2λ. This is because there are two normal sensors that might fail.

Let us assume that the entire operation period is long enough that the steady-state probabilities of the system states can be reached within a relatively short time interval. These probabilities can be related to a set of state equations derived from the Markov diagram in Figure 1. Specifically, the models associated with the system states 0, 1, and 2 can be written as

$$-2\lambda P_0 + \mu P_3 = 0 \tag{10}$$

$$2\lambda P_0 - (\lambda + \epsilon)P_1 = 0 \tag{11}$$

$$\lambda P_1 - \epsilon P_2 = 0 \tag{12}$$

where P_0, P_1, P_2, and P_3 denote the long-term probabilities of the system in states 0, 1, 2, and 3, respectively. For the intermediate states $3j$, $3j + 1$, and $3j + 2$ ($j = 1, 2, ..., m_p - 3$), the following state equations are valid

$$\epsilon P_{3j-2} + \mu P_{3j+3} - (\mu + 2\lambda)P_{3j} = 0 \tag{13}$$

$$2\lambda P_{3j} + \epsilon P_{3j-1} - (\lambda + \epsilon)P_{3j+1} = 0 \tag{14}$$

$$\lambda P_{3j+1} - \epsilon P_{3j+2} = 0 \tag{15}$$

Here again, the symbol P_k is used to denote probability of intermediate system state k. For states $3m_p - 6$, $3m_p - 5$, and $3m_p - 4$, i.e., states in which all off-line sensors are not working, the maintenance practices are not the same as those for the intermediate states. Consequently, a slightly different set of state equations can be obtained

$$\epsilon P_{3m_p-8} + \mu P_{3m_p-5} - (\mu + 2\lambda)P_{3m_p-6} = 0 \tag{16}$$

$$2\lambda P_{3m_p-6} + \epsilon P_{3m_p-7} + \mu P_{3m_p-4} - (\lambda + \mu)P_{3m_p-5} = 0 \tag{17}$$

$$\lambda P_{3m_p-5} - \mu P_{3m_p-4} = 0 \tag{18}$$

Finally, it is obvious that the sum of the probabilities of all possible states should be 1, i.e.

$$\sum_{k=0}^{3m_p-4} P_k = 1 \tag{19}$$

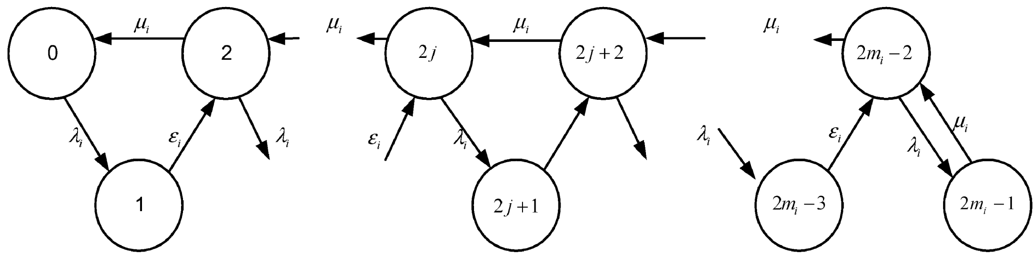

Figure 2. Markov diagram of a corrective maintenance program with only spare sensors ($m_p \geq n_p = 1$).

Notice that eqs 10−17 and 19 can be solved simultaneously. The solution procedure is outlined below. First, one can obtain from eqs 10−12 that

$$P_1 = \frac{2\lambda}{\lambda + \epsilon} P_0 \tag{20}$$

$$P_2 = \frac{2\lambda^2}{\epsilon(\lambda + \epsilon)} P_0 \tag{21}$$

$$P_3 = \frac{2\lambda}{\mu} P_0 \tag{22}$$

Next, from eqs 13−15, the following relations can be derived

$$P_{3j+1} = \frac{\epsilon}{\lambda + \epsilon} P_{3j-1} + \frac{2\lambda}{\lambda + \epsilon} P_{3j} \tag{23}$$

$$P_{3j+2} = \frac{\lambda}{\epsilon} P_{3j+1} \tag{24}$$

$$P_{3j+3} = \frac{2\lambda + \mu}{\mu} P_{3j} - \frac{\epsilon}{\mu} P_{3j-2} \tag{25}$$

From eqs 20−25, it is clear that all intermediate probabilities can be expressed in terms of P_0. In addition, eqs 16 and 17 can be rearranged as

$$P_{3m_p-5} = \frac{2\lambda}{\lambda + \mu} P_{3m_p-6} - \frac{\mu}{\lambda + \mu} P_{3m_p-4} + \frac{\epsilon}{\lambda + \mu} P_{3m_p-7} \tag{26}$$

$$P_{3m_p-4} = \frac{\lambda}{\mu} P_{3m_p-5} \tag{27}$$

Thus, the last two probabilities in the Markov model can also be written as explicit functions of P_0. Consequently, all of the above results can be substituted into eq 19 to solve for P_0. Once the analytical form of P_0 is obtained, the other probabilities can also be determined accordingly.

In this work, the flow rate of stream p is assumed to be measurable as long as at least one on-line sensor is normal. Thus, its direct estimation availability can be expressed as

$$A_p^{\mathrm{D}} = \sum_{j=0}^{m_p-2} (P_{3j} + P_{3j+1}) \tag{28}$$

Finally, notice that there are $n_p + 1$ rows and $m_p - n_p + 1$ columns in the Markov diagram presented in Figure 1. Various maintenance models can be derived by selecting appropriate values for n_p and m_p. Three special cases of the general Markov model are presented below:

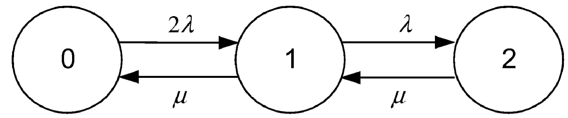

Figure 3. Markov diagram of a corrective maintenance program with only redundant sensors ($m_p = n_p = 2$).

Corrective Maintenance with Only Spare Sensors. In this case, there is only one on-line sensor, and thus $m_p > n_p = 1$. The Markov diagram is reduced to the one presented in Figure 2. By following the same derivation procedure, the following results can be obtained

$$P_{2j} = \frac{\left(\frac{\lambda}{\mu}\right)^j}{\left(1 + \frac{\lambda}{\epsilon}\right)\sum_{k=0}^{m_p-2}\left(\frac{\lambda}{\mu}\right)^k + \left(1 + \frac{\lambda}{\mu}\right)\left(\frac{\lambda}{\mu}\right)^{m_p-1}} \quad j = 0, 1, ..., m_p - 1 \tag{29}$$

$$P_{2j+1} = \frac{\left(\frac{\lambda}{\mu}\right)^j \frac{\lambda}{\epsilon}}{\left(1 + \frac{\lambda}{\epsilon}\right)\sum_{k=0}^{m_p-2}\left(\frac{\lambda}{\mu}\right)^k + \left(1 + \frac{\lambda}{\mu}\right)\left(\frac{\lambda}{\mu}\right)^{m_p-1}} \quad j = 0, 1, ..., m_p - 2 \tag{30}$$

$$P_{2m_p-1} = \frac{\left(\frac{\lambda}{\mu}\right)^{m_p}}{\left(1 + \frac{\lambda}{\epsilon}\right)\sum_{k=0}^{m_p-2}\left(\frac{\lambda}{\mu}\right)^k + \left(1 + \frac{\lambda}{\mu}\right)\left(\frac{\lambda}{\mu}\right)^{m_p-1}} \tag{31}$$

The direct estimation availability of stream p can thus be computed according to

$$A_p^{\mathrm{D}} = \sum_{j=0}^{m_p-1} P_{2j} \tag{32}$$

Corrective Maintenance with Only Redundant Sensors. In this case, there are *no* off-line sensors, and therefore $m_p = n_p \geq 2$. Let us consider the special example when $n_p = 2$. The Markov diagram in Figure 1 now becomes the simplified version shown in Figure 3. The same derivation procedure can be used to produce an explicit formula for the direct estimation availability

$$A_p^{\mathrm{D}} = P_0 + P_1 = \frac{\mu(2\lambda + \mu)}{(\mu + \lambda)^2 + \lambda^2} \tag{33}$$

Corrective Maintenance without Redundant and Spare Sensors. Here, $m_p = n_p = 1$, and the result in eq 9 can be derived in a straightforward fashion.

Life-Cycle Cost

The life-cycle cost of the sensor network is, of course, the sum of the costs of all sensors, i.e.

$$\mathrm{LCC} = \sum_{p \in \boldsymbol{P}} \mathrm{LCC_p} \tag{34}$$

where $\mathrm{LCC_p}$ is the life-cycle cost of the sensors for the measurement of stream p. As suggested by Sanchez and Bagajewicz,[15] this cost can be considered to be dominated by the procurement cost (PC_p) and maintenance cost (MC_p), i.e.

$$\mathrm{LCC_p} \approx \mathrm{PC}_p + \mathrm{MC}_p \tag{35}$$

In this study, the procurement cost is determined according to

$$\mathrm{PC}_p = m_p C_p^{\mathrm{C}} \tag{36}$$

where C_p^{C} denotes the capital cost of a single sensor for measuring the flow rate of stream p. On the other hand, the maintenance cost is the sum of the expected costs of repair and replacement, $\mathrm{EC}_p^{\mathrm{repr}}$ and $\mathrm{EC}_p^{\mathrm{repl}}$, i.e.

$$\mathrm{MC}_p = \mathrm{EC}_p^{\mathrm{repr}} + \mathrm{EC}_p^{\mathrm{repl}} \tag{37}$$

These expected costs can be estimated on the basis of the maintenance model. Specifically

$$\mathrm{EC}_p^{\mathrm{repr}} = C_p^{\mathrm{repr}} \sum_{k=1}^{T} \frac{\mathrm{EN}_p^{\mathrm{repr}}(k-1, k)}{(1+r)^k} \tag{38}$$

and

$$\mathrm{EC}_p^{\mathrm{repl}} = C_p^{\mathrm{repl}} \sum_{k=1}^{T} \frac{\mathrm{EN}_p^{\mathrm{repl}}(k-1, k)}{(1+r)^k} \tag{39}$$

where T represents the operating life of the sensor network in years; r is the interest rate; C_p^{repr} and C_p^{repl} denote the operating costs per repair and per replacement, respectively; and $\mathrm{EN}_p^{\mathrm{repr}}(k-1, k)$ and $\mathrm{EN}_p^{\mathrm{repl}}(k-1, k)$ represent the expected numbers of repairs and replacements, respectively, in time interval $[(k-1), k]$. By definition,[16] the expected number of repairs between two distinct times, t_1 and t_2, can be determined by integrating the unconditional repair intensity $\nu(t)$, i.e., $\mathrm{EN}_p^{\mathrm{repr}}(t_1, t_2) = \int_{t_1}^{t_2} \nu(t)\,\mathrm{d}t$, and $\nu(t)$ is the probability that a sensor will be repaired per unit time at time t. Because sensor repairs are allowed only at states $3j$ ($j = 1, 2, ..., m_p - 2$), $3m_p - 5$, and $3m_p - 4$ in the general Markov model (see Figure 1), the expected number of repairs in a year can thus be approximated by the following equation[16]

$$\begin{aligned} \mathrm{EN}_p^{\mathrm{repr}}(k-1, k) \approx \\ \int_{k-1}^{k} \mu\left(P_{3m_p-4} + P_{3m_p-5} + \sum_{j=1}^{m_p-2} P_{3j}\right) \mathrm{d}t \\ = \mu\left(P_{3m_p-4} + P_{3m_p-5} + \sum_{j=1}^{m_p-2} P_{3j}\right) \end{aligned} \tag{40}$$

The approximation formula for expected number of replacements can be derived in a similar fashion, i.e.

$$\mathrm{EN}_p^{\mathrm{repl}}(k-1, k) = \epsilon \sum_{j=0}^{m_p-3} (P_{3j+1} + P_{3j+2}) \tag{41}$$

For the sake of completeness, the formulas for evaluating the expected numbers of repairs and replacements in three special cases of the general Markov model are also listed below:

Corrective Maintenance with Only Spare Sensors.

$$\mathrm{EN}_p^{\mathrm{repr}}(k-1, k) = \mu\left(P_{2m_p-1} + \sum_{j=1}^{m_p-1} P_{2j}\right) \tag{42}$$

$$\mathrm{EN}_p^{\mathrm{repl}}(k-1, k) = \epsilon \sum_{j=0}^{m_p-2} P_{2j+1} \tag{43}$$

Corrective Maintenance with Only Redundant Sensors.

$$\mathrm{EN}_p^{\mathrm{repr}}(k-1, k) = \mu(P_1 + P_2) = \frac{2\lambda\mu(\lambda+\mu)}{(\lambda+\mu)^2 + \lambda^2} \tag{44}$$

$$\mathrm{EN}_p^{\mathrm{repl}}(k-1, k) = 0 \tag{45}$$

Corrective Maintenance without Redundant and Spare Sensors.

$$\mathrm{EN}_p^{\mathrm{repr}}(k-1, k) = \mu[1 - A_p(t)] = \frac{\lambda\mu}{\lambda+\mu} \tag{46}$$

$$\mathrm{EN}_p^{\mathrm{repl}}(k-1, k) = 0 \tag{47}$$

Precision of Flow Estimator

To satisfy the constraint imposed by eq 3, it is necessary to evaluate the standard deviations of the estimated flow rates. The conventional approach[7] is followed in our work to perform this task. Because the formulas for computing the variances of reconciled data are well documented,[17] they are not repeated in this paper. However, it should be noted that the conventional formulation was established on the basis of the assumption that at most one sensor can be installed on a process stream. If redundant sensors are allowed in sensor network design, the reduced incidence matrix must be modified before the existing formulas can be used. Specifically, if n_p ($n_p \geq 1$) sensors are installed on a process stream (say stream p), then the corresponding arc in the process graph should be treated as n_p fictitious arcs connected in series, and the $n_p - 1$ connecting nodes of these arcs should be viewed as fictitious units

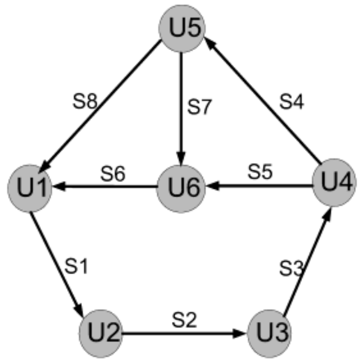

Figure 4. Process graph of the simplified ammonia process.

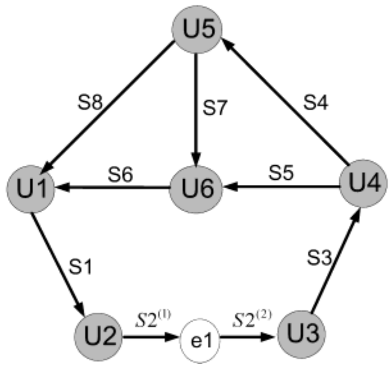

Figure 5. Modified process graph of the simplified ammonia process ($n_2 = 2$).

in the process. Consequently, every fictitious arc in the modified process graph is associated with only one on-line sensor. The resulting mass balance equations can then be used in the conventional data reconciliation calculations.

For illustration purposes, let us consider the process graph shown in Figure 4 representing a simplified ammonia process.[18] This graph contains six nodes and eight arcs, with node E denoting the environment and nodes U1−U5 the major units. If hardware redundancy is not considered in sensor network design, then the reduced incidence matrix can be written as

$$\begin{bmatrix} -1 & 0 & 0 & 0 & 0 & +1 & 0 & +1 \\ +1 & -1^* & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & +1^* & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & +1 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & +1 & 0 & 0 & -1 & -1 \end{bmatrix} \tag{48}$$

Notice that the rows and columns correspond to the nodes and arcs, respectively, and that the structure of the process graph is represented with the nonzero entries in the matrix. For example, the positions of −1 and +1 in the second column can be used to reflect the fact that stream S2 flows from unit U2 to unit U3. If two redundant sensors are installed on stream S2, i.e., $n_2 = 2$, then the modified process graph can be found in Figure 5, and the corresponding incidence matrix is

$$\begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & +1 & 0 & +1 \\ +1 & -1^* & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & +1^* & -1^* & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & +1^* & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & +1 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & +1 & 0 & 0 & -1 & -1 \end{bmatrix} \tag{49}$$

Table 1. Sensor Parameters

parameter	sensor type 1	sensor type 2	sensor type 3
purchase cost (C^{C})	350.0	250.0	200.0
repair cost (C^{repr})	70.0	50.0	40.0
replacement cost (C^{repl})	7.0	5.0	4.0
failure rate (λ)	0.3	0.6	0.7
repair rate (μ)	1.0	1.0	1.0
replacement rate (ϵ)	50.0	50 0.0	50.0
sensor precision (σ/m)	1.5%	2.5%	3.0%

As mentioned before, stream S2 is now treated as two fictitious arcs connected in series, i.e., $S2^{(1)}$ and $S2^{(2)}$, and they are represented with the nonzero entries (marked by asterisks) in the second and third columns in the above matrix. Notice also that the connecting node of these two fictitious arcs, i.e., e1, is reflected by an additional row, i.e., the third row. If there are three or more sensors on stream S2 or redundant sensors on the other process streams, the corresponding reduced incidence matrix can be constructed in the same way. As long as the incidence matrix is available, the standard deviation of any stream can be evaluated easily according to the conventional method.

Solution Algorithms

As mentioned previously, the design variables of the present optimization problem, i.e., l_p, m_p, and n_p, are restricted to zeros or natural numbers. Because gradient-based methods are not suitable for such problems, genetic algorithms (GAs) were used to solve the mathematical program defined in eqs 1−3.

Generally speaking, GAs can be considered as a stochastic evolution strategy imitating the natural selection process of biological species. The basic analogy is established between a design variable in the optimization problem and a *gene* in a *chromosome* (or individual). Thus, one chromosome represents a possible solution. In our studies, the conventional encoding approach is used for representing l_p, m_p, and n_p with standard genetic codes.[19,20] The evolution process in GA begins with an initial population. The genetic code of each chromosome in this population is produced with a random number generator. Once the initial population is available, three commonly used evolution steps, i.e., reproduction, crossover, and mutation, are executed repeatedly until the total number of generations reaches an assigned value. In this study, the reproduction step is performed according to the popular roulette-wheel selection scheme. A two-point crossover procedure is then implemented to swap genes between chromosomes. Finally, the uniform mutation technique is adopted to alter the genetic codes of several randomly chosen chromosomes. After reproduction, crossover, and mutation are performed in each generation, an elitism strategy is practiced to preserve the best chromosome identified so far. In particular, the largest fitness value achieved in the current generation is compared with that of a chromosome temporarily stored in a buffer. If the former is larger, then this chromosome should be stored instead, and the current population remains unchanged. Otherwise, the worst chromosome in the current population should be replaced by the one originally stored in the buffer.

When this fitness measure is used in an optimization problem, it usually serves as the objective function. The constraint equations can often be handled by introduc-

Table 2. Optimal Solutions of Mathematical Programming Model ($m_p = n_p = 1$ or $m_p = n_p = 0$)

	case no.		
	1	2	3
cost limit (C_T)	2000.0	3000.0	4000.0
precision requirement set	A	A	A
system availability (A^S)	0.8318	0.9542	0.9598
life-cycle cost (LCC)	1983.0	2984.0	3410.0
estimation precision of stream 2 (σ_2/m_2)	0.5898%	0.5804%	0.5414%
estimation precision of stream 5 (σ_5/m_5)	1.3720%	1.2319%	1.2011%

Table 3. Optimal Sensor Network Designs ($m_p = n_p = 1$ or $m_p = n_p = 0$)

	stream no. (p)							
	1	2	3	4	5	6	7	8
				case no. 1				
sensor type (l_p)	0	0	1	0	1	3	1	1
est. avail. (A_p)	0.8736	0.8736	0.9169	0.8333	0.8742	0.8318	0.8742	0.9169
				case no. 2				
sensor type (l_p)	1	0	1	1	1	1	1	1
est. avail. (A_p)	0.9924	0.9911	0.9924	0.9800	0.9615	0.9615	0.9542	0.9800
				case no. 3				
sensor type (l_p)	1	1	1	1	1	1	1	1
est. avail. (A_p)	0.9982	0.9980	0.9982	0.9830	0.9688	0.9688	0.9598	0.9830

ing additional penalties. To facilitate implementation of the roulette-wheel selection scheme in reproduction, the fitness measure must always be represented by a positive number. Thus, the following fitness function is used in our study

$$\text{fit} = \text{obj} + \sum_i (w_i)(\text{pen}_i) \tag{50}$$

where obj denotes the objective function of optimization problem, pen$_i$ represents the penalty function associated with the ith inequality constraint, and w_i is the corresponding weight. The objective function used in this study has already been described previously in eq 1. The penalty functions are expressed as

$$\text{pen}_i = \frac{1}{z_i + 1} \tag{51}$$

where

$$z_i = \begin{cases} 0 & \text{if the corresponding constraint is satisfied} \\ (C_T - \text{LCC})^2 & \text{if the corresponding cost constraint is violated} \\ (\sigma_p^* - \sigma_p)^2 & \text{if the corresponding precision constraint is violated} \end{cases} \tag{52}$$

Application Example

The ammonia process presented in Figure 4 is again considered here to demonstrate the usefulness of the proposed mathematical programming model. In particular, the optimization problem defined in eqs 1−3 has been solved repeatedly under different constraint levels to study the impacts of exercising various maintenance options on the system availability and also on the estimator's precision. The sensor parameters given in Table 1 were used in all of our computations. It should be noted that the precision of each sensor is defined as the ratio between the standard deviation (σ) and unbiased mean (m) of its measurement data under normal operating conditions. It is further assumed that, in this example, $\tilde{\mathbf{P}} = \{2, 5\}$, and the precision requirements specified in eq 3 are imposed according to two sets of parameters, i.e.

set A

$$\frac{\sigma_2^*}{m_2} = 1.0\% \qquad \frac{\sigma_5^*}{m_5} = 1.5\%$$

set B

$$\frac{\sigma_2^*}{m_2} = 0.5\% \qquad \frac{\sigma_5^*}{m_5} = 1.0\%$$

The expected values of the normal flow measurements of streams 1−8, i.e., $m_1 - m_8$, used in this example are 100, 100, 100, 60, 40, 70, 30, and 30 kg/s, respectively. Notice from eq 5 that it might not be necessary to obtain direct measurements of a process stream to raise its estimation availability to a satisfactory level. It is thus assumed that no sensors should be purchased for the streams on which on-line sensors are not installed, i.e., $m_p = 0$ if $n_p = 0$. Finally, a life cycle of 5 years and an interest rate of 3.0% are used in all of the case studies presented below.

Let us first consider the results of applying the simplest corrective maintenance policy to the streams on which a single sensor is installed, i.e., $m_p = n_p = 1$ or $m_p = n_p = 0$ ($\forall p \in \boldsymbol{P}$). The optimal solutions of the corresponding mathematical programs can be found in Table 2. Three cases are presented here. They were obtained with the same precision requirement, i.e., set A, but different cost limits, i.e., 2000, 3000, and 4000. It can be observed from Table 2 that, by relaxing the cost constraint, the optimal system availability can be gradually improved. The optimal sensor network designs in these three cases are presented in Table 3. Notice that, if the design variable for sensor type (l_p) assumes a value of zero, then the corresponding flow should be unmeasured. Notice also that the best sensor, i.e., type 1, is installed on every stream in case 3. Thus, the corresponding system availability, i.e., 0.9598 (see Table 2), is the maximum achievable value if spare and redundant sensors are not allowed in the corrective maintenance program.

Table 4. Optimal Solutions of Corrective Maintenance Model ($m_p \geq n_p = 1$ or $m_p = n_p = 0$)

	case no.		
	4	5	6
cost limit (C_T)	3000.0	4000.0	4500.0
precision requirement set	A	A	A
system availability (A^S)	0.9542	0.9843	0.9916
life-cycle cost (LCC)	2984.0	3957.0	4483.0
estimation precision of stream 2 (σ_2/m_2)	0.5804%	0.5930%	0.6774%
estimation precision of stream 5 (σ_5/m_5)	1.2319%	1.2421%	1.2512%

Table 5. Optimal Sensor Network Designs ($m_p \geq n_p = 1$ or $m_p = n_p = 0$)

	stream no. (p)							
	1	2	3	4	5	6	7	8
case no. 4								
total quan. (m_p)	1	0	1	1	1	1	1	1
sensor type (l_p)	1	0	1	1	1	1	1	1
est. avail. (A_p)	0.9924	0.9911	0.9924	0.9800	0.9615	0.9615	0.9542	0.9800
case no. 5								
total quan. (m_p)	1	1	1	2	1	1	1	2
sensor type (l_p)	1	2	2	1	1	1	1	1
est. avail. (A_p)	0.9977	0.9974	0.9977	0.9955	0.9843	0.9859	0.9879	0.9954
case no. 6								
total quan. (m_p)	1	0	1	2	2	2	2	2
sensor type (l_p)	1	0	1	2	1	1	1	2
est. avail. (A_p)	0.9973	0.9968	0.9973	0.9962	0.9916	0.9916	0.9916	0.9962

Table 6. Optimal Solutions of Corrective Maintenance Model ($m_p \geq n_p \geq 1$ or $m_p = n_p = 0$)′

	case no.			
	7	8	9	10
cost limit (C_T)	4500.0	5000.0	6000.0	6500.0
precision requirement set	B	B	B	B
system availability (A^S)	0.9793	0.9902	0.9954	0.9967
life-cycle cost (LCC)	4476.0	4982.0	5913.0	6456.0
estimation precision of stream 2 (σ_2/m_2)	0.4848%	0.4738%	0.4185%	0.4185%
estimation precision of stream 5 (σ_5/m_5)	0.9342%	0.9084%	0.8886%	0.8886%

Table 7. Optimal Sensor Network Designs ($m_p \geq n_p \geq 1$ or $m_p = n_p = 0$)

	stream no. (p)							
	1	2	3	4	5	6	7	8
case no. 7								
total quan. (m_p)	1	0	1	2	2	2	1	2
redun. deg. (n_p)	1	0	1	2	2	2	1	2
sensor type (l_p)	2	0	2	1	1	1	1	1
est. avail. (A_p)	0.9957	0.9948	0.9957	0.9933	0.9814	0.9814	0.9793	0.9933
case no. 8								
total quan. (m_p)	2	0	1	2	2	2	2	1
redun. deg. (n_p)	2	0	1	2	2	2	2	1
sensor type (l_p)	1	0	2	1	1	1	1	1
est. avail. (A_p)	0.9984	0.9978	0.9980	0.9975	0.9907	0.9907	0.9902	0.9952
case no. 9								
total quan. (m_p)	2	0	2	2	2	2	2	2
redun. deg. (n_p)	2	0	2	2	2	2	2	2
sensor type (l_p)	1	0	1	1	1	1	1	1
est. avail. (A_p)	0.9997	0.9996	0.9997	0.9986	0.9963	0.9963	0.9954	0.9986
case no. 10								
total quan. (m_p)	2	0	2	2	2	2	3	2
redun. deg. (n_p)	2	0	2	2	2	2	2	2
sensor type (l_p)	1	0	1	1	1	1	1	1
est. avail. (A_p)	0.9997	0.9996	0.9997	0.9991	0.9967	0.9967	0.9976	0.9991

Next, let us introduce an additional practice, i.e., spares, into the above maintenance strategy to enhance the system performance. Specifically, if a sensor is installed on stream p, then spares are also allowed, i.e., $m_p \geq n_p = 1$ or $m_p = n_p = 0$ ($\forall p \in \boldsymbol{P}$). The solutions of the corresponding mathematical program and the optimal sensor network designs are presented in Tables 4 and 5. From the results in Tables 2 and 4, it can be clearly seen that it is indeed possible to improve the system availability to a level higher than the upper limit achieved with the simple corrective maintenance strategy mentioned previously. Also, from the values of σ_2/m_2 and σ_5/m_5, it can be concluded that the estimator's precision of stream 2 or 5 cannot really be enhanced with spares. Notice also that the network designs in cases 2 and 4 are the same under the same cost constraint, i.e., $C_T = 3000$. This is because the system availability is defined as the lowest value among all estimation availabilities, i.e., eqs 4 and 5, and the spares can only improve the *direct* estimation availabilities of the corresponding streams, i.e., eq 32. Because the cost limit in case 4 is relatively low, incorporating the spares must cause an increase in the number of unmeasured streams and thus might yield a lower system availability. Consequently, the highest availability in this case is achieved by placing sensors on as many process streams as possible and by excluding spares in the network design. On the other hand, from cases 5 and 6 in Table 5, it is obvious that spares can be used to increase the system availability effectively if the cost constraint allows direct measurement of almost every stream in the process.

As mentioned before, the estimator's precision for a process stream is not affected by the use of spares. Thus, the benefits of introducing hardware redundancy are studied in a series of additional case studies. In particular, the options of spare and redundant sensors are both incorporated in the mathematical programming model under more stringent precision requirements, i.e., set B. If a stream (say stream p) is directly measured, then the general corrective maintenance strategy is applicable to the corresponding sensors, i.e., $m_p \geq n_p \geq 1$. If a stream is not required to be measured directly, then no sensors should be purchased, i.e., $m_p = n_p = 0$. The solutions of the mathematical program and the optimal sensor network designs are presented in Tables 6 and 7. From Table 6, it is clear that the system availability and estimator's precision can both be improved to the desired levels if the cost constraint can be sufficiently relaxed. Notice from Table 7 that the degree of redundancy (n_p) on each stream in the network design of case 9 is the same as that of case 10. As a result, the precision levels of streams 2 and 5 in these two cases are also the same (see Table 6). The extra cost of case 10 is incurred from the addition of a spare sensor for stream 7. Its impact is mainly an increase in the system availability.

Conclusions

A comprehensive mathematical programming model has been developed in this study for identifying the best measurement locations in a given process network and also the optimal numbers of redundant and spare sensors used in the general corrective maintenance program. From the results we have obtained in extensive case studies, the following conclusions can be drawn:

1. Spare sensors can be used to significantly raise the system availability.
2. Although adopting redundant hardware is a less effective means for improving availability, it can be applied to reduce the variability in flow estimation.
3. If the life-cycle cost of a sensor network is not a limiting condition, the maximum precision and availability achieved with the conventional design and maintenance strategies can both be surpassed with the use of a proper combination of redundant and spare sensors.

Acknowledgment

This work was supported by the National Science Council of the ROC government under Grant NSC90-2214-E006-004.

Nomenclature

A_p = estimation availability of stream p
A_p^{D} = direct estimation availability of stream p
A_p^{I} = indirect estimation availability of stream p
A^{S} = system availability
C_{T} = upper limit of life-cycle cost, LCC
C_p^{C} = capital cost of a single sensor for the measurement of stream p
C_p^{repl} = operating cost per replacement
C_p^{repr} = operating cost per repair
$\mathrm{E}C_p^{\mathrm{repl}}$ = expected cost of sensor replacement for stream p
$\mathrm{E}C_p^{\mathrm{repr}}$ = expected cost of sensor repair for stream p
$\mathrm{EN}_p^{\mathrm{repl}}(t_1, t_2)$ = expected number of replacements in the time interval $[t_1, t_2]$
$\mathrm{EN}_p^{\mathrm{repr}}(t_1, t_2)$ = expected number of repairs in the time interval $[t_1, t_2]$
fit = fitness function in the genetic algorithm
LCC = life-cycle cost of the sensor network
LCC_p = life-cycle cost of the sensors for the measurement of stream p
l_p = design variable denoting the sensor type on stream p
$\mathbf{l}$ = vector $[l_1, l_2, ..., l_N]^{\mathrm{T}}$
m_p = design variable denoting the total number of sensors purchased for stream p
$\mathbf{m}$ = vector $[m_1, m_2, ..., m_N]^{\mathrm{T}}$
MC_p = maintenance cost of stream p
n_p = design variable denoting the number of redundant sensors installed on stream p
$\mathbf{n}$ = vector $[n_1, n_2, ..., n_N]^{\mathrm{T}}$
obj = objective function
P_k = long-term probability of the system at state k
$\mathbf{P}$ = set of all process streams
$\tilde{\mathbf{P}}$ = set of process streams on which precision requirements are imposed
PC_p = procurement cost of stream p
pen_i = penalty function associated with the ith inequality constraint
$\mathbf{S}_j$ = stream set obtained by removing the stream under consideration from the jth cut set containing such stream
r = interest rate
T = operating life of the sensor network
w_i = weight of the ith penalty in the fitness function

Greek Letters

λ = failure rate
μ = repair rate
ϵ = replacement rate
σ_p = standard deviation of the flow-rate estimate of stream p
σ_p^* = upper bound of σ_p

Literature Cited

(1) Vaclavek, V.; Loucka, M. Selection of Measurement Necessary to Achieve Multicomponent Mass Balances in Chemical Plants. *Chem. Eng. Sci.* **1976**, *31*, 1199.
(2) Kretsovalis, A.; Mah, R. S. H. Effect of Redundancy on Estimation Accuracy in Process Data Reconciliation. *Chem. Eng. Sci.* **1987**, *42*, 2115.
(3) Madron, F.; Veverka, V. Optimal Selection of Measuring Points in Complex Plants by Linear Models. *AIChE J.* **1992**, *38*, 227.
(4) Ali, Y.; Narasimhan, S. Sensor Network Design for Maximizing Reliability of Linear Processes. *AIChE J.* **1993**, *39*, 820.
(5) Ali, Y.; Narasimhan, S. Redundant Sensor Network Design for Linear Processes. *AIChE J.* **1995**, *41*, 2237.
(6) Ali, Y.; Narasimhan, S. Sensor Network Design for Maximizing Reliability of Bilinear Processes. *AIChE J.* **1996**, *42*, 2563.
(7) Bagajewicz, M. J. Design and Retrofit of Sensor Networks in Process Plants. *AIChE J.* **1997**, *43*, 2300.
(8) Bagajewicz, M. J.; Sanchez, M. C. Duality of Sensor Network Design Models for Parameter Estimation. *AIChE J.* **1999**, *45*, 661.
(9) Bagajewicz, M. J.; Sanchez, M. C. Design and Upgrade of Nonredundant and Redundant Linear Sensor Networks. *AIChE J.* **1999**, *45*, 1927.

(10) Bagajewicz, M. J.; Sanchez, M. C. Cost-Optimal Design of Reliable Sensor Networks. *Comput. Chem. Eng.* **2000**, *23*, 1757.

(11) Sen, S.; Narasimhan, S.; Deb, K. Sensor Network Design of Linear Processes Using Genetic Algorithms. *Comput. Chem. Eng.* **1998**, *22*, 385.

(12) Raghuraj, R.; Bhushan, M.; Rengaswamy, R. Locating Sensors in Complex Chemical Plants Based on Fault Diagnostic Observability Criteria. *AIChE J.* **1999**, *45*, 310.

(13) Bhushan, M.; Rengaswamy, R. Design of Sensor Network Based on the Signed Directed Graph of the Process for Efficient Fault Diagnosis. *Ind. Eng. Chem. Res.* **2000**, *39*, 999.

(14) Bagajewicz, M. J. *Process Plant Instrumentation: Design and Upgrade*; Technomic Publishing Co.: Lancaster, PA, 2001.

(15) Sanchez, M. C.; Bagajewicz, M. J. On the Impact of Corrective Maintenance in the Design of Sensor Networks. *Ind. Eng. Chem. Res.* **2000**, *39*, 977.

(16) Henley, E. J.; Kumamoto, H. *Reliability Engineering and Risk Assessment*; Prentice Hall: Englewood Cliffs, NJ, 1981.

(17) Narasimhan, S.; Jordache, C. *Data Reconciliation & Gross Error Detection: An Intelligent Use of Process Data*; Gulf Publishing Company: Houston, TX, 2000.

(18) Kretsovalis, A.; Mah, R. S. H. Observability and Redundancy Classification in Generalized Process Networks: I. Theorems. *Comput. Chem. Eng.* **1988**, *12*, 671.

(19) Holland, J. H. *Adaptation in Natural and Artificial Systems*; University of Michigan Press: Ann Arbor, MI, 1975.

(20) Michalewicz, Z. *Genetic Algorithm + Data Structure = Evolution Programs*, 3rd ed.; Springer-Verlag: Berlin, 1992.

Received for review July 29, 2002
Revised manuscript received June 20, 2003
Accepted June 30, 2003

IE020567J