17th European Symposium on Computer Aided Process Engineering – ESCAPE17
V. Plesu and P.S. Agachi (Editors)

Constraint Programming based multi-objective sensor network design for fault diagnosis

Prakash R. Kotecha, Mani Bhushan, Ravindra D. Gudi.

Department of Chemical Engineering, IIT Bombay, Powai, Mumbai – 400 076, India. [prakash_kotecha,mbhushan,ravigudi]@iitb.ac.in

Abstract

Optimal placement of sensors based on different criteria viz., precision, reliability, cost, and fault unobservability has been an important area of research in the last few years. Most of the sensor location problems proposed in literature have been solved either using graph theoretic approaches or conventional mathematical optimization techniques. However, these techniques have not been able to satisfactorily address the issue of tradeoffs between multiple objectives, the determination of which is important from a designer's perspective in terms of providing design flexibility. In this article, we address this challenge by proposing the use of Constraint Programming (CP) as a potential alternative to conventional solution techniques to determine the pareto optimal solutions. CP is an intelligent enumeration based optimization technique that uses domain reduction as its inference engine and has recently emerged as a powerful tool for solving combinatorial optimization problem in operations research. We also present efficient reformulation of some existing problems using the superior modelling power of CP.

Keywords Constraint Programming(CP),sensor network design, multiobjective optimization.

1. Introduction

The importance of optimal placement of sensors has been discussed in [1-5]. Most of these methods use graph theoretic approaches [1,2] and either have time issues [2] or do not guarantee optimality [1]. Mathematical programming techniques [3,4] have also been reported to have computational issues for larger flowsheet [3]. Moreover, most of the existing research has not satisfactorily addressed [4,5] the design of sensor networks in a multi-objective framework. In this article, we show the use of CP in addressing this deficiency for the design of combinatorial sensor network design problems. CP is an intelligent enumeration based optimization technique that uses constraints to reduce the domain of the decision variables. Its strong domain reduction inference engine has made it more suitable for discrete optimization problems when compared to the traditional mathematical programming techniques [6]. The important merits of CP stem from the fact that it does not distinguish between linear and non-linear programming. Moreover, its superior modeling power and the ability to easily determine all the multiple global optima make it highly suitable for discrete optimization problems. Additional literature on CP can be found in [6]. In this article, we specifically show the superior modeling power of CP along with its use in the evaluation of trade-offs between various conflicting objectives by generating the pareto-optimal front.

2. Sensor Network Design for fault diagnosis

Fault detection and diagnosis (FDD) plays an important role in the operation of a chemical plant. Efficient FDD requires strategic placement of sensors. Bhushan and Rengaswamy [4] have designed sensor networks for the minimization of the maximum unobservability of all the faults. In their approach, every fault *i* has an occurrence probability (f_i) and every sensor *j* has a failure probability (s_j). A fault can remain undetected if the fault occurs and the associated sensors fail at the same time. This event has been termed as the unobservability of fault *i* which can be calculated as [4]

$$U_i = f_i \prod_{j=1}^{n} \left(s_j\right)^{\left(b_{ij} x_j\right)} \tag{1}$$

In the above expression, *n* denotes the number of variables, x_j denotes the number of sensors on the j^{th} variable and can be greater than one (incase of hardware redundancy), and b_{ij} is the i,j^{th} entry of the cause-effect bipartite matrix *B*. The faults form the rows of this matrix and the variables form the columns. If the i^{th} fault affects j^{th} variable, then the b_{ij} entry is one and is zero otherwise.

1. Superior Modeling in CP

The traditional mathematical programming techniques require the constraints to be in the form of inequalities and this sometime forces the inclusion of additional variables and constraints thereby potentially increasing the computational burden. In this section, we utilize the high expressive modeling power of CP to present an efficient reformulation of the MILP formulation (Eq.

(2)) available in the literature [4,5] for the design of sensor networks with minimum unobservability. Additionally, the objectives of minimizing the sensor network cost and the maximization of the network distribution were considered in decreasing order of precedence.

$$\begin{aligned}
&Min \quad \left[\alpha_1 U + \alpha_2 \sum_{j=1}^{n} c_j x_j - \sum_{j=1}^{n} n_j\right] \\
&s.t \quad \sum_{j=1}^{n} c_j x_j \le C^*; \quad U \ge \log\left(U_i\right), \quad i = 1,2..m; \quad n_j \le x_j, \quad j = 1,.....,n; \\
&\quad n_j \in \{0,1\}; \quad U \in \mathbb{R}^-; \quad x_j \in \mathbb{Z}^+
\end{aligned} \tag{2}$$

where α_1 and α_2 are the lexicographic constants, c_j denotes the cost of the sensor measuring the j^{th} variable, C^* denotes the maximum available cost for the sensor network and m denotes the number of faults. The term n_j takes a value of one if the j^{th} variable is measured and zero otherwise and hence the term $\sum_{j=1}^{n} n_j$ denotes the network distribution. We now present an efficient CP based formulation in Eq. (3) which is much smaller in size than the above MILP formulation without compromising on the rigor of representation.

$$\begin{aligned}
&Min \quad \left[\alpha_1 \max_{\forall i \in M}\left(\log U_i\right) + \alpha_2 \sum_{j=1}^{n} c_j x_j - \sum_{i=1}^{n} \min\left(x_j, 1\right)\right] \\
&s.t \quad \sum_{j=1}^{n} c_j x_j \le C^*; \qquad x_j \in \mathbb{Z}^+
\end{aligned} \tag{3}$$

It can be seen that the term *min* $(x_j,1)$ is equivalent to the term n_j for it takes a value of one if the j^{th} variable is measured and zero otherwise. Thus the n binary variables n_j along with their constraints can be eliminated. Similarly, the term U in Eq. (2) corresponds to the maximum unobservability of the m faults and this can be easily represented by the term $\max_{\forall i \in M}\left(\log U_i\right)$ thereby additionally eliminating the m unobservability constraints. Table 1 compares the dimensionality of the MILP and CP based formulations. It can be seen that the CP model is much smaller than the MILP based model. This reduction in dimensionality can translate to potential savings in the computational burden. It has to be noted that unlike in the MILP formulation, the number of binary variables and the number of constraints are independent of the number of process variables and the number of faults in the CP based formulation.

Table 1. Comparison of dimensionality of MILP and CP Formulations

Formulations	Binary Variables	Integer Variables	Continuous variables	Constraints
MILP Formulation	*Network*	*n*	*1*	*m+1*
CP Formulation	0	*n*	*0*	*1*

2. Multi-objective Optimization: Pareto Front

The lexicographic optimization approach for multi-objective optimization [4,5] suffers from the drawback that it requires apriori knowledge of the precedence level in the various objective functions. In reality, this precedence level may not be explicitly known to the designer and hence the designer may be interested in evaluating tradeoffs between various conflicting objectives without specifying any precedence levels. Such trade-offs are characterized as pareto-optimal front and are the set of non-dominated solutions [4,5]. In the following discussion, we exploit the ability of CP to solve feasibility problems to determine such pareto-front. Also, we assume that the minimization of the unobservability is the primary objective and the designer needs to study the tradeoffs between the network distribution and the cost of the sensor network. The pareto-front is determined using the following two steps.

Step 1: This step involves the solution of an optimization problem to determine the minimum unobservability, $U^{optimal}$

$$\begin{aligned} &Min \quad \max_{\forall i \in M} \left(\log U_i \right) \\ &s.t \quad \sum_{j=1}^{n} c_j x_j \le C^*; \quad x_j \in \mathbb{Z}^+ \end{aligned} \tag{4}$$

Step 2: This step involves the solution of a feasibility problem to determine all the solutions that have the unobservability equal to $U^{optimal}$

$$\begin{aligned} &Solve \quad \sum_{j=1}^{n} c_j x_j \le C^* \\ &\max_{\forall i \in M} \left(\log U_i \right) = U^{optimal}; \quad x_j \in \mathbb{Z}^+ \end{aligned} \tag{5}$$

The set of solutions to Step 2 inherently contain all the pareto-optimal solution and can be obtained by a simple, straight forward post-optimality analysis. The cost of such solutions and network distribution can be easily generated from the sensor network configuration. Thus, we will be able to generate all the trade-off solutions between the network distribution and the cost of sensor network. An important point to be noted is that the set of solutions to Step 2 also contain all the realizations (solutions with identical set of objective function values but with different sensor network configuration) for each of the pareto-point. Further, all these tradeoff solutions have minimum unobservability, $U^{optimal}$. It can be easily seen that this procedure can be applied to optimization problems for the determination of multiple global optimal solution as well. We now demonstrate these ideas on the TE case study.

2.1. Case Study: Tennessee Eastman (TE) Process

We demonstrate the suitability of CP to solve the above formulations on the benchmark TE problem. This problem has 50 variables and 15 faults and has been taken from literature [4,5]. The list of variables and faults can be found in [5]. The costs of the sensors along with the fault occurrence and sensor failure probabilities have been taken from [5] and are not reproduced here. The results presented in this section are based on the assumption of single fault resolution case after the removal of the redundant constraints [4,5].

2.2. Results

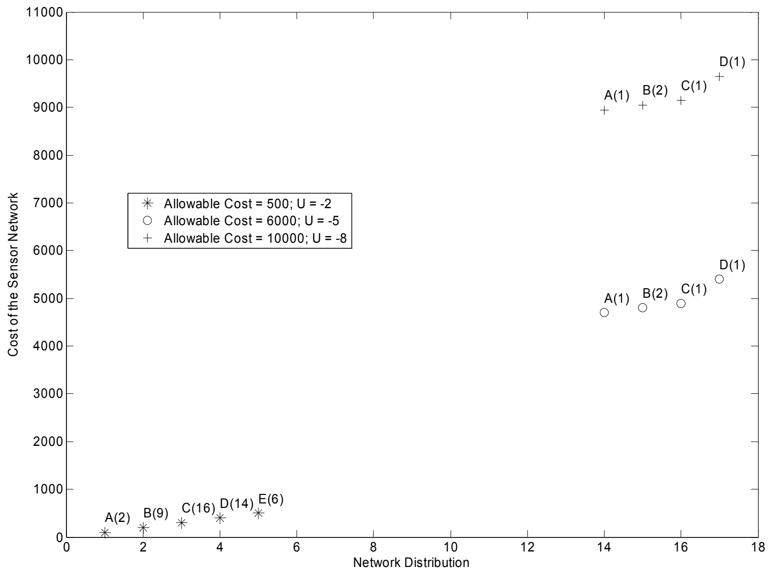
1. Dimensionality of the problem

Based on Table 1, it can be seen that the MILP formulation will have 50 additional binary variables and 65 additional constraints compared to the CP formulation. While for this problem, no significant computational benefit was observed, in general, as the number of variables increase, this reduction in dimensionality can translate to reduction in the computational burden.

2. Pareto-optimal fronts

Figure 1 shows the pareto-fronts between the network distribution and the cost of the sensor network for three different available costs: C^*= 500, 6000 and 10000. The number of realizations at each pareto solution is also shown. For example, solution A for C^*= 500 has two sensor network configurations that have a network distribution=1 and a cost=100 units. Thus, the designer can choose a sensor network configuration based on the different tradeoffs.

Figure 1: Pareto-optimal fronts between network distribution and cost of the sensor network



3. Conclusions

In this article, we have shown the suitability of CP to solve the combinatorial sensor network design problems. We have shown the superiority of CP to efficiently model the sensor network design problem and its applicability to determine the pareto-front for various conflicting objectives. Thus, it can be seen that the use of CP enables efficient modeling and also gives a wider choice of solutions along with the tradeoffs for the multi-objective optimization problems.

References

1. Y. Ali, S. Narasimhan, AIChE J., 39 (1993) 820-828.
2. M. Bagajewicz, AIChE J. 43 (1997) 2300-2306.
3. M. Bagajewicz, E. Cabrera, AIChE J. 48 (2002) 2271-2282.
4. M. Bhushan and R. Rengsswamy, IFAC-DYCOPS, Boston, 2004.
5. M. Bhushan, S. Narasimhan and R. Rengswamy, Robust sensor network design for fault diagnosis, Submitted to Comput. Chem. Engng., Sept. 2006.
6. C.T. Maravelias and I.E. Grossmann, Comput. Chem. Eng. 28, (2004) 1921