STRUCTURAL CONTROL ARCHITECTURE OPTIMIZATION FOR 3-D SYSTEMS USING ADVANCED MULTI-OBJECTIVE GENETIC ALGORITHMS

A Dissertation

by

YOUNG JIN CHA

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

December 2008

Major Subject: Civil Engineering

STRUCTURAL CONTROL ARCHITECTURE OPTIMIZATION FOR 3-D SYSTEMS USING ADVANCED MULTI-OBJECTIVE GENETIC ALGORITHMS

A Dissertation

by

YOUNG JIN CHA

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Chair of Committee,	Luciana R. Barroso
Committee Members,	Anne M. Raich
	Norris Stubbs
	Sergiy Butenko
Head of Department,	David Rosowsky

October 2008

Major Subject: Civil Engineering

ABSTRACT

Structural Control Architecture Optimization for 3-D Systems Using Advanced Multi-Objective Genetic Algorithms. (December 2008)

Young Jin Cha, B.S., Kumoh National Institute of Technology; M.S., Yonsei University

Chair of Advisory Committee: Dr. Luciana R. Barroso

The architectures of the control devices in active control algorithm are an important fact in civil structural buildings. Traditional research has limitations in finding the optimal architecture of control devices such as using predefined numbers or locations of sensors and dampers within the 2-and 3-dimensional (3-D) model of the structure. Previous research using single-objective optimization only provides limited data for defining the architecture of sensors and control devices. The Linear Quadratic Gaussian (LQG) control algorithm is used as the active control strategy. The American Society of Civil Engineers (ASCE) control benchmark building definition is used to develop the building system model. The proposed gene manipulation genetic algorithm (GMGA) determines the near-optimal Pareto fronts which consist of varying numbers and locations of sensors and control devices for controlling the ASCE benchmark building by considering multi-objectives such as interstory drift and minimizing the number of the control devices.

The proposed GMGA reduced the central processing unit (CPU) run time and produced more optimal Pareto fronts for the 2-D and 3-D 20-story building models.

Using the GMGA provided several benefits: (1) the possibility to apply any pre-suggested multi-objective optimization mechanism; (2) the availability to perform a objective optimization problem; (3) the adoptability of the diverse encoding provided by the GA; (4) the possibility of including the engineering judgment in generating the next generation population by using a gene creation mechanisms; and (5) the flexibility of the gene creation mechanism in applying and changing the mechanism dependent on optimization problem.

The near-optimal Pareto fronts obtained offer the structural engineer a diverse choice in designing control system and installing the control devices. The locations and numbers of the dampers and sensors in each story are highly dependent on the sensor locations. By providing near-Pareto fronts of possible solutions to the engineer that also consider diverse earthquakes, the engineer can get normalized patterns of architectures of control devices and sensors about random earthquakes.

TABLE OF CONTENTS

Page

Page

Page

LIST OF FIGURES

Page

Page

Page

Page

Page

Page

Page

Page

Page

LIST OF TABLES

Page

1 INTRODUCTION TO RESEARCH ON MULTI-OBJECTIVE OPTIMAL CONTROL STRATEGIES FOR 3-D SYSTEMS

1.1 Introduction

During the last twenty years, the implementation of control strategies in structural design has aided significantly towards the reduction of the damage to civil structural systems and in the enhancement of the safety of human lives and properties. Specifically in the area of structural control, structural engineers have worked to reduce the damage of building from strong winds and earthquakes and suggested many alternatives to control the structure with that have improved performance and reduced structural damage. The most basic method of controlling the response of a structure is to use passive devices, which are energy dissipation devices that use the properties of the material's deformation to dissipate energy. More advanced approaches use active and semi-active devices. Active structural control is a highly adaptive and powerful mechanism that can be used to protect civil structural systems. This control mechanism uses an external electric power source to operate active devices and uses feedback concerning the measured responses of the structural system to determine the control force that the devices must apply in real-time.

This dissertation follows the style of *Journal of Structural Engineering*.

These forces are determined using computational algorithms designed to control specific response parameters, such as interstory drift or accelerations. Recently interest in optimizing the installation of the control devices and sensors for structural control civil structural system which includes the numbers and locations of these devices has increased, in addition to research concerning the development of advanced control strategies. Optimizing the installation architecture that defines the control device type and location and the sensor locations is required in order to optimize system performances as indicated by interstory drifts and connection rotation, and to reduce the damage structures receive in extreme loading events.

1.2 Motivation and Significance

Many researchers are interested in active control rather than passive control since active control provides strong seismic protection and performance since it can be designed for multi-level hazards. A well-known active control algorithm is the Linear Quadratic Gaussian (LQG) controller. The LQG controller is known for its efficiency and applicability to real civil building systems (Ohtori et al. 2004). LQG controller performance, however, is heavily dependent on the accelerometer location and the number and location of active devices.

There has been some research performed concerning what is the most efficient placement of active devices and sensors to civil structures. However, these studies were performed based on the assumptions made concerning several critical facts. For example, some researchers used predefined numbers and locations of dampers (Rao et al. 1991),

and in some cases, collocated a sensor/actuator was used to optimize the numbers of damper (Dhingra and Lee 1995). The location and the number of dampers used in very important in reducing the structural response to the seismic loads, and the locations of the sensors also affects the control response in the LQG controller. Some research was performed with the Linear Quadratic Regulator (LQR) that assumes full state feedback control or nearly same type of controller (Rao et al. 1991; Dhingra and Lee 1995; Li et al. 2000; Abdullah et al. 2001). The assumption of full state feedback is highly idealized and the results do not correspond well to locations of acceleration feedback controllers. In addition, most of the previous research efforts were carried out on 2 dimensional (2-D) models. The American Society of Civil Engineering (ASCE) Committee on Structural control has suggested the use of 3-, 9-, 20-story steel buildings as benchmark structures designed for the SAC (Joint venture of three non-profit organizations: The Structural Engineers Association of California (SEAOC), the Applied Technology Council (ATC) and California Universities for Research in Earthquake Engineering (CUREE)) project (Spencer et al. 1999). Using 3-D structural model is required to reflect the real condition of the benchmark building and suggest practical architectures of installation of active devices and sensors.

The significance of this research is the development of practical methods and procedures to find near-optimal architectures of any kind of control devices and sensors with any kind of control algorithms with development of 3-D structure modeling method for practical civil structures. The proposed optimization methods and procedures, therefore, are applicable to other problems related to finding near-architectures of

control devices and sensor with its own control algorithm, and then the resulting data can be offered to structural or control engineers. So that they con choose the architecture of control devices and sensor which is well matched to the structure.

1.3 Objectives and Scopes

The primary objective of this investigation is to use practical and realistic ASCE control benchmark modeling, high-performance active controller, and advanced optimization methods to determine useful and diverse alternative architectures for active control devices and sensors. This study focuses on optimizing strategies to control the seismic response of steel moment resisting frames (SMRF). The scope of the research covers four areas:

1. Development of 2-D and 3-D structural models for the ASCE control benchmark building.
2. Application of active hydraulic devices to the LQG controller
3. Implementation of the multi-objective optimization based on traditional genetic algorithm (GA) for optimizing the actuator/sensor installation.
4. Development of advanced GA operators for the multi-objective GA to obtain optimal architectures of active control devices and sensors.

The 2-D and 3-D structural modeling methods are suggested for the ASCE control benchmark building systems. The finite element method (FEM) is used for linear

modeling of the SMRF. A LQG controller is applied to the modelled 2-D and 3-D structural system in order to reduce the structural damage simulated for well-known earthquakes including El Centro, Hachinohe, Northridge, and Kobe. The performance of implicit redundant representation (IRR) GA and SGA on optimizing actuator/sensor layouts for the control optimization problems are compared. A multi-objective optimization methodology based on non-dominated sorting genetic algorithm (NSGA) and IRR GA is implemented and used to obtain the Pareto front for a multi-objective problem with two objectives. The two objectives are the total number of active dampers and sensors used in the whole story and the maximum root mean square (RMS) interstory drift of the whole story. Advanced multi-objective optimization methodologies based on the NSGA-II and SPEA2 are developed and implemented to improve the optimization performance in finding the Pareto front related to the two objectives. Comparisons of the results of the multi-objective optimization methodologies using traditional and advanced GA are performed. Based on these comparisons and case studies, the new high-performance multi-objective genetic algorithm (MOGA) is developed.

1.4 Overview and Organization

A brief discussion of the motivation and objectives and scope of the research effort is presented in Section 1. In Section 2, the research literature is reviewed concerning relevant work in controlling strategies and optimizing of the placement of

active dampers and sensors. For the optimization, the generic genetic algorithms and multi-objective genetic algorithm methodologies are also reviewed and compared.

Section 3 defines the methods of 2- and 3-dimentional (2-D and 3-D) modeling of the American Society of Civil Engineering (ASCE) control benchmark buildings. The finite element method (FEM) is applied and static condensation is used to eradicate the zero massed degrees of freedoms. For 3-D modeling, a two-way unsymmetric plan model (Chopra 2000) is used. The mode shapes and frequencies are calculated and compared with other researcher's results to check the model's accuracy.

In Section 4, LQG control law is explained and adopted to the system models defined in Section 3. A detailed explanation of the LQG control design is provided. To explain LQG control law, the LQR regulator and Kalman filter is also reviewed. The LQG weighting matrix is determined using the controllability/observability properties and closed-loop stability requirements. Additionally the structural responses of displacements and accelerations are found and the LQG controller's performance is verified by comparison with the uncontrolled responses.

The performance comparisons of the SGA and IRR GA are carried out with the single objective optimization problem in the Section 5, and the basic multi-objective optimization method using NSGA and IRR GA is defined. The Pareto fronts are also defined for the 2-D 3-, and 20-story building and the 3-D 20-story building models.

The development of advanced multi-objective optimization methods are described using NSGA-II (Deb et al. 2000) and SPEA2 (Zitzler et al. 2001, 2002a,

2002b). The proposed methods are used to performed the optimization for the defined control optimization problem.

In Section 7, the gene manipulation genetic algorithm (GMGA) is proposed. With the creation rule, new individuals are created based on engineering judgment using the non-dominated individuals in multi-objective problem and superior individuals in the single objective problem.

In Section 8, the conclusions of this research effort are stated, including a discussion of the limitations of the proposed methods identified by this research, and future recommendations.

2 LITERATURE REVIEW

In order to protect the property of human beings or reduce structural damage that occurs from natural hazards or a strong earthquake, researchers have tried to develop retrofit or hazard protection strategies from the natural hazards. Specially, after the strong Northridge earthquakes in California, Kobe and Hanshin, structural engineers focused on developing structural control concepts. Although the control knowledge was already studied in the electrical and mechanical engineering area, new control design criteria is needed because of the diverse and various characteristics of each structural system. Therefore, structural control systems were developed and applied to improve the performance of real buildings and bridges in extreme events. However, the optimal placement of control devices has not been studied and subsequently there are not any design guidelines or criteria to follow. Structural control problems are also very complex and highly non continuous. Applying advanced optimization knowledge is required to provide the necessary knowledge on this topic.

To perform this dissertation research, a review of the current status of research in two areas is required. One is concerning optimization methods and the other is concerning, control strategies and devices. In this literature review, the basic control concepts such as passive, active, semi-active and hybrid control systems are summarized and reviewed. In addition, basic optimization concepts, advanced genetic algorithms and multi-objective genetic algorithms are also summarized and discussed.

2.1 Overview of Structural Control

The basic concept of control is to try to modify the behavior of an object to suit our needs. Although many control schemes have been developed in other engineering disciplines, the differences of controlling a large structural system that is fixed on the ground and subjected to environmental disturbances such as strong winds, waves, and earthquakes, which are uncertain in their occurrence and intensity to the traditional control system, make the application of pre-existing control methods difficult. With the motivation of preventing the loss of human lives from the natural hazards and reducing the damage to civil structures, there has been a big advancement in civil structural control area during last ten years.

To design a control system for civil structures, we need to define the control schemes, control devices and controllers that can be used in general. Currently there are four main types of control systems: passive, active, semi-active and hybrid that can be applied to the control of real buildings or studied by structural engineers.

2.1.1 Passive Control Scheme

The passive control scheme operates without using an external energy supply. Generally it is inexpensive and uses the material or material properties such as inelastic behavior, friction, a high-polymer or highly viscous liquid, mass itself or hysteretic behavior. The base isolation system is also a kind of passive control schemes because it

does not used external power. Instead it uses the material's hysteretic energy. The main advantages of passive control systems are (Olariu 1995):

- Reduce the structural element size in the cross-sectional area because the passive control system decreases the peak acceleration and the maximum seismic shearing forces.
- Offer safety of structures under emergency unexpected conditions.
- Prevent damage and sliding or rolling of equipment.
- Base-isolation offers irregular structures a benefit of installation.
- To not overload the existing foundation, the passive system limits lateral forces in the retrofitted buildings.

The disadvantages of passive control systems are (Olariu 1995):

- Various uncertainties in the design of passive response control structures, such as variability of wind or earthquake loads and variability of structural properties.
- Not many choices for the base-isolation system.
- Overturning moments can be problem because decreasing available resisting moments.
- Only experts in control have an adequate knowledge of response control.

2.1.2 Active Control Scheme

Active control schemes need external power to operate the force actuators. The components of the control system are the actuator, sensor, and controller. Active control schemes can be divided in two types depending on how they calculate the control force. The first one is a closed-loop control, which uses information about structural response only, which is measured with physical sensors. The other type is an open-loop control in which the control input is applied without the knowledge of the structural response (Barroso 1999). The advantages of active control schemes are (Soong and Spencer 2000):

- The effectiveness of the control response is enhanced and only limited by the capacity of the actuator used in the control systems.
- Active control systems are insensitive to site conditions and ground motion more than passive control system.
- Active control systems can be applied to multi-hazard situations such as controlling responses to strong wind and earthquakes.
- Multiple control objectives such as human comfort, structural safety, can be selected.

The disadvantages of active control systems are (Soong and Spencer 2000):

- Control actuator and control cost is very expensive
- Need large external power source to operate the control system
- Need maintenance for the installed actuators and controllers

- If the power is lost, during the earthquake or strong wind, the control system does not work.

2.1.3 Semi-Active Control Scheme

Semi-active control system means that it uses passive supplemental energy dissipation devices, and also the concept of feedback control algorithm to modify the behavior of these devices. For example, viscous fluid, orificing of fluid, and sliding friction can be modified by the structural response or external excitation. So the stiffness control devices, electrorheological dampers, magnetorheological dampers, friction devices, fluid viscous dampers, tuned mass dampers and tuned liquid dampers can be used as semi-active control devices (Symans and Constantinou 1999). The main advantages of semi-active control systems are (Soong and Spencer 2002);

- Use small amount of external power, such that battery power is enough to operate in some cases.
- Do not have the potential to destabilize the structural system because not enough force is generated.
- The performance is significantly better than passive control system.
- Have the potential to achieve the majority of the performance that is achieved using fully active systems.

2.1.4 Hybrid Control Schemes

Hybrid control systems are composed of two control systems, such as active and passive control systems, or semi-active and passive control systems. Most well-known hybrid dampers are a combination of tuned mass damper (TMD) and an actuator. Although the reduction of the structural response is mainly dependent on the TMD, the mass of the TMD is moved by the actuator forces. This hybrid control system has nearly the same advantages as semi-active system.

2.2 Overview of Genetic Algorithm

The natural environment is complex and its hazardous winds, waves, and earthquakes can not be defined specifically in strength, direction and the time when they will occur. To adapt to these random natural environmental conditions, human beings have used their brain to survive in dangerous environments. From this concept, many scientists try to give some intelligence to machines for the better efficiency of their performance of their objectives and tasks. This research area is artificial intelligence (AI). AI means a study to adapt to environment and takes actions to perform its objectives. One of the well expressed AI's characteristics is the area of computational intelligence (CI). As a branch of AI, CI is a comprehensive expression of fuzzy systems, neural networks, and evolutionary computation (EC). CI's main characters are applying learning, adaptation and evolution to solve a problem. One of the subset of EC is a population-based evolutionary algorithm (EA) which is classified as a metaheuristic optimization algorithm. Mechanisms of EA came from the biological evolution such as

selection, recombination, mutation, and so on. Genetic algorithms (GA) are the one of the well known EA theories that are based on the Darwin's evolutionary theory (www.wikipedia.org).

There are two types of mathematical search methods. The first is the direct method which uses local fitness information obtained from the random search or grid search to find local optima and the second is the indirect method, which uses the gradient of the objective function and constraints to get local optima. However, most real-world problems are very complex and sometimes they are discontinuous and unpredictable. In these problems, the local gradient can not give enough information to find the global optimal value.

Genetic algorithms (GA) were first suggested by Holland (1975) and Goldberg (1989) developed it more systemically. Goldberg also identified the differences of GAs and with traditional mathematical search methods by defines four characteristics of GA. The first is that GAs work with a encoding of the parameters set rather than the parameters themselves. The second is that GA searches for more optimal solution from a population of points not just a single point. The third is that GAs do not use derivatives or other auxiliary knowledge, but use objective function information. The last is that GAs do not use deterministic rules, but use probabilistic transition rules.

2.2.1 Simple Genetic Algorithm (SGA)

The flowchart for the SGA is shown in Figure 2.1. SGA uses selection, crossover, and mutation for the search and reproduction operator. In the first step of a SGA, an

initial population of individuals is randomly generated with the binary bits ‘0’ and ‘1’. The bit values of these encoded individuals of the initial population are mapped to define variable and decoded as real value. Each decoded individual can be evaluated by the fitness function. The fitness function evaluated determines how well the individuals compute in the problem environment. The successive process is the selection. Based on the fitness value, the better adapted individual will be chosen with diverse selection methods. The two most popular selection methods are roulette wheel selection and tournament selection. The method used for selection is dependent on the problem characters and GA user’s engineering judgment.

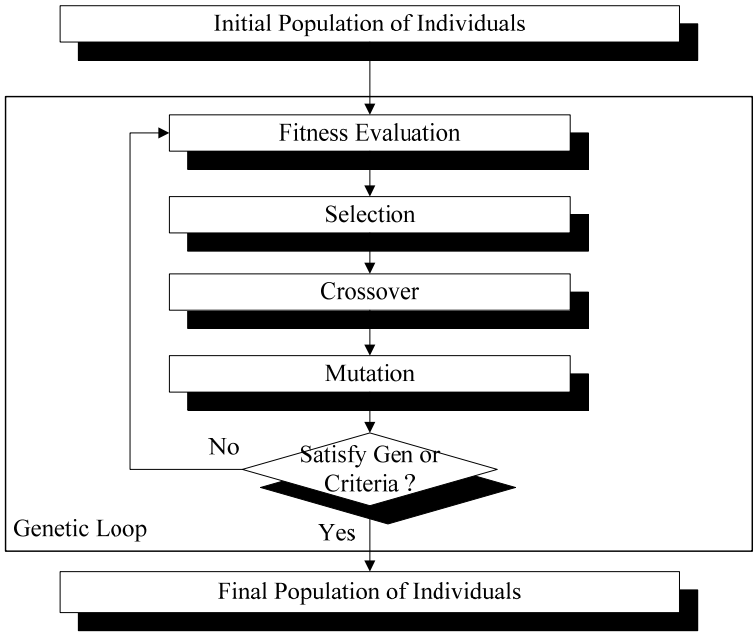

Figure 2.1. SGA flowchart

The selected individuals undergo reproduction using crossover and mutation. Then the genetic loop, which is the iterative process, composed of fitness evaluation, selection, crossover, and mutation are satisfied, such as convergence of the population will continue until the iterations or after set problem criteria. Individual of the final population represent the optimal or near-optimal solution of the optimization problem.

2.2.2 Selection Operator

Environments have the ability to control the number of creatures by giving more chances to survive to better adapted lives. To apply this mechanism to genetic algorithms, the selection method was suggested. The most popular selection method is tournament selection (Goldberg 1989). Tournament selection is shown in Figure 2.2.

The process of the tournament selection is to define the individuals competing against each other in the tournament by randomly selecting from the current population. The defined tournament selection size is usually 3 for the single objective problem and 2 for the multi-objective problem used in the research. Each of the selected individuals is compared to each other in fitness, and then the best fittest individual is selected. This selection process continues until the selected number of individuals is same as population size.

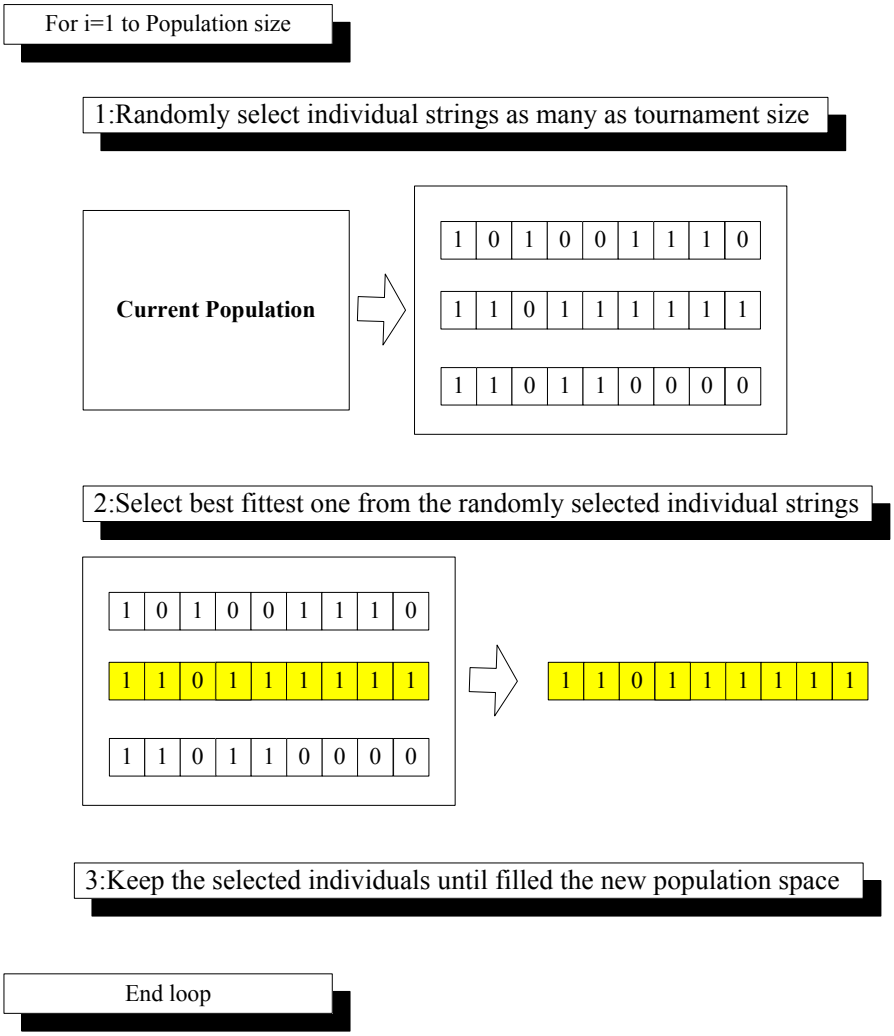

Figure 2.2. Tournament selection operator

2.2.3 Crossover

Crossover operator imitates the sexual mating of the lives of nature. From the sexual mating the each life reproduces their offspring, and during this process each parent's genes will be mixed wise each other. This gene mixing concept is adopted by the crossover operator in the GA. The algorithm of crossover is simple, but its effect and

performance in getting near-optimal results is very powerful. The detailed process of generic one-point crossover procedure is shown in Figure 2.3.

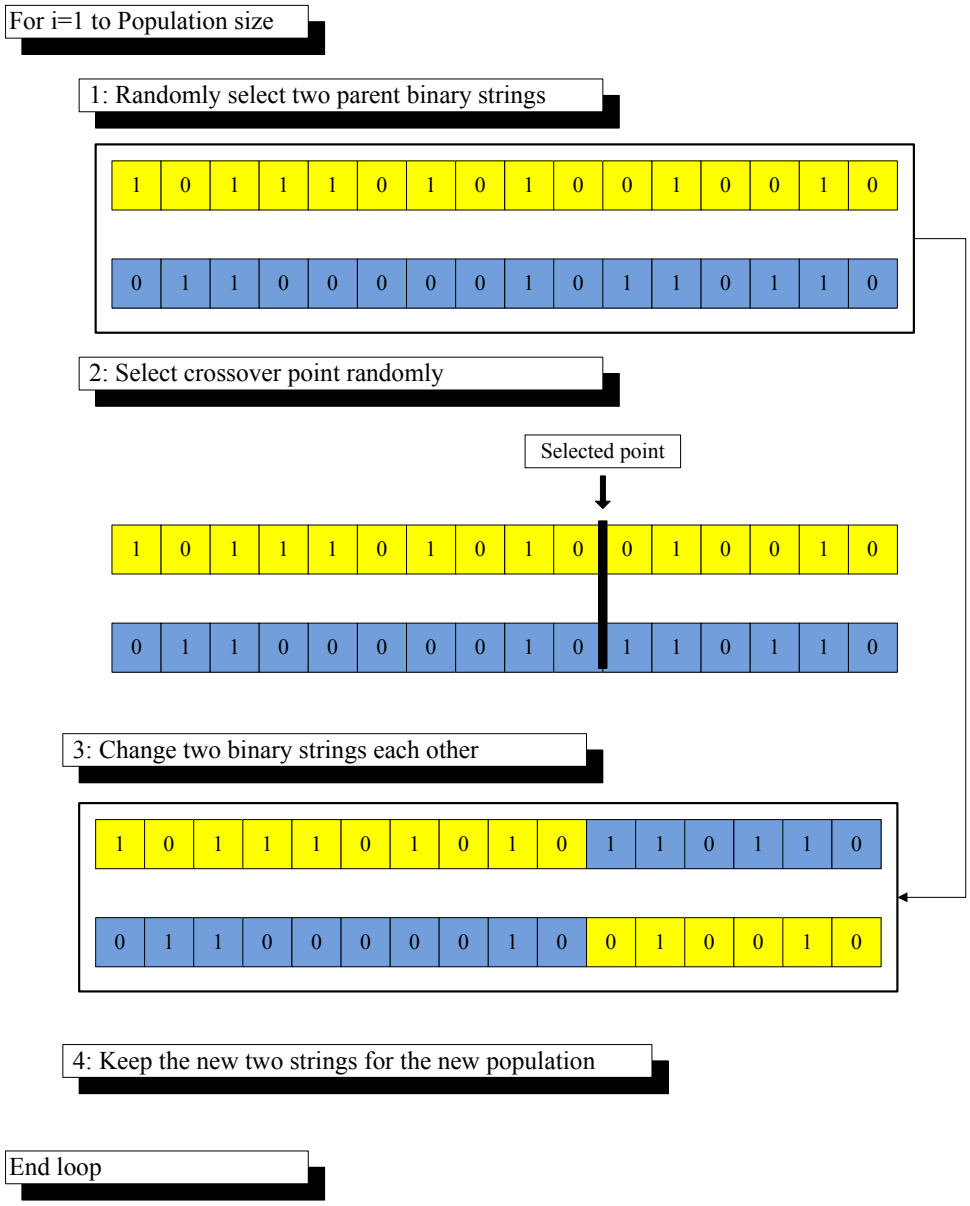

Figure 2.3. Crossover operator

At first, two parent binary strings are randomly selected from the population, and the crossover point is also randomly selected. With the crossover point defined, the two parent binary strings will exchange string segments with each other as shown in Figure 2.3. Each time two new individuals are generated by this operator. The crossover process is repeated until the new population is filled. Often one-point crossover is used by GA, but by the conditions of optimization problem characteristics and environment, multi-point crossover (Eshelman et al. 1989) which uses multiple crossover points, uniform crossover (Syswerda 1989), and cycle crossover (Oliver et al. 1987) can be used.

2.2.4 Mutation

Crossover mechanism makes the population robust enough to adapt to the problem environment during continuous generations. However, it makes each individual string resemble each other over time. The problem that results is that the genetic solution can be pre-converged to local optima. The crossover mechanism can not get out of this condition because the crossover operator by itself does not create new information in the case of a single genetic algorithm.

Thus the GA needs a mutation operator to help prevent premature convergence to local points and to reintroduce new previously lost information to the population. The mutation mechanism is shown in Figure 2.4. The first step is selection of the population bit in order of precedence, and then if the randomly generated mutation probability is bigger than the predefined mutation probability, the selected bit is turned over for example if the zero is selected bit, it will be one as shown in Figure 2.4. This simple but

powerful operation helps to prevent the lose of gene information of the current population and helps to sustain the diversity of gene information throughout the population.

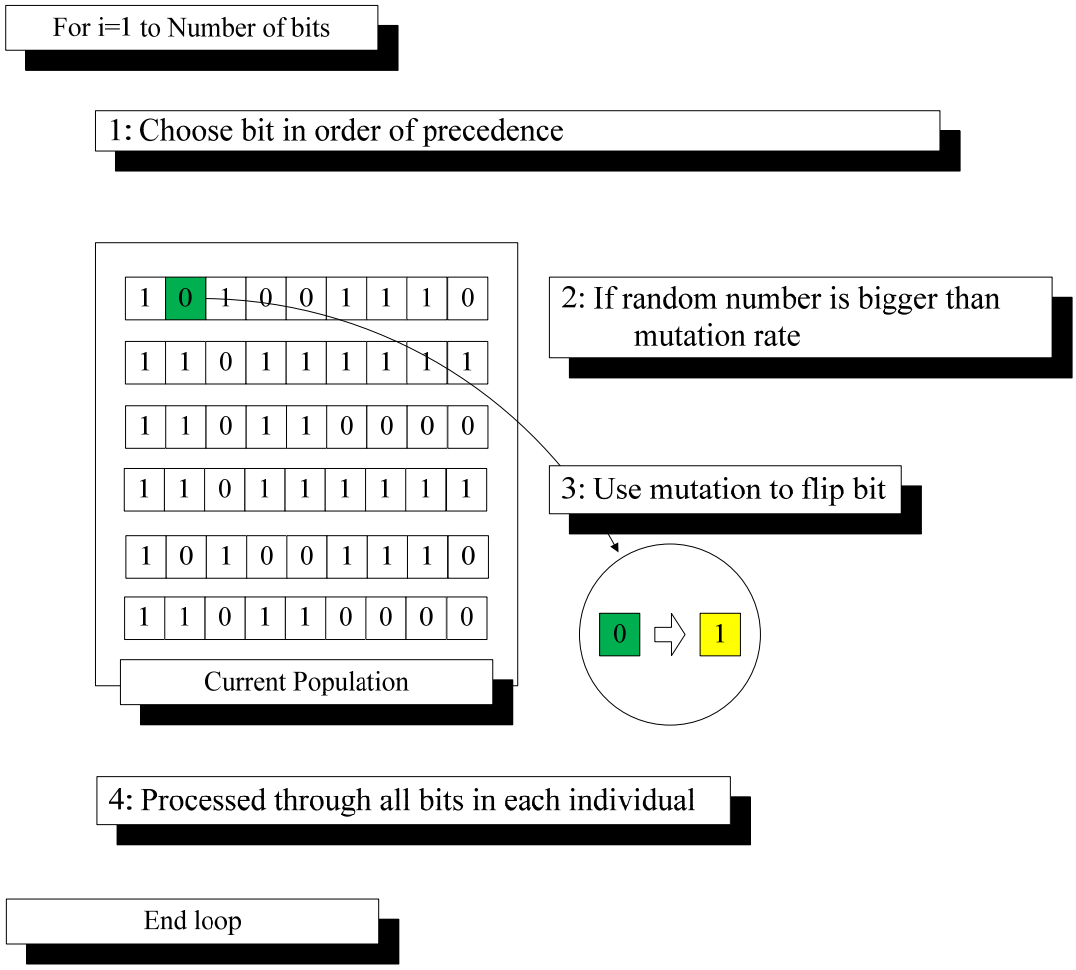

Figure 2.4. Mutation operator

2.2.5 Elitism

During genetic iterations and its genetic operators, the best individual of the each generation can be demolished by its genetic operators or probabilistic character of selection mechanism. To transfer the best individual to the next generation without losing its superiority, GA uses the elitism. Elite strategy keeps the best individual of the current generation. In general, this elite strategy enhances the performance of GA. However, using elitism makes the GA solution converge to local solutions because it can reduce the diversity of the population information. Therefore, how many individuals will be chosen as elite strings is cautiously considered with the characteristics of the optimization problem itself.

2.3 Overview of Implicit Redundant Representation Genetic Algorithm (IRR GA)

Simple genetic algorithm is popularly used in diverse optimization problems such as complex and highly non-continuous problem domain, but this traditional optimization method is limited to working with a structured parameter number. SGA is not flexible in expression of the diverse variable numbers that are required to search an unstructured problem domain. Raich and Ghaboussi (1998) developed the IRR GA to enhance GA performance by providing self-organization of the unstructured problem representation and by maintaining a diverse population. The IRR GA provides adaptive behavior with background of floating non-coding segments to provide an understanding of the role that introns play in biological systems (Wu and Lindsay 1996).

2.3.1 Description of Implicit Redundant Representation GA

Implicit redundant representation concept in GA is developed by the Raich (1999) to optimize frame systems defined with an unstructured domain in parameter values. The essential and redundant sections of a string in IRR are allowed to interact dynamically by using a string length that is longer than the required length to encode the parameter values. IRR uses gene instances to express the design parameters and gene locator (GL) to notify the starting point of the gene instance and redundant material that will be used in the future generation by the genetic operators as shown in Figure 2.5. The gene instance has encoded parameter values and these values are decoded in a similar way to other SGAs.

Due to the free drifting character of the gene instance in the string and redundant material, the number of parameters expressed in the string is not fixed, and this means that IRR GA do not have constraints to define and limit the search range. The ability of the IRR GA to generate a varying number of design parameters in each individual enables diverse parameter alternatives to be created and optimized in a non-continuous and discrete design domain. These allow the GA process to create and destroy design variables and provide a flexible representation for unstructured designs. The GL is defined by the user. Generally GL has simple pattern of 111 for the binary encoding and the user has to consider the probability of an occurrence of the GL pattern within the expressed string length, because the probability of occurrence of the GL pattern means that the probability of the gene instance that has the design parameters (Raich 1999).

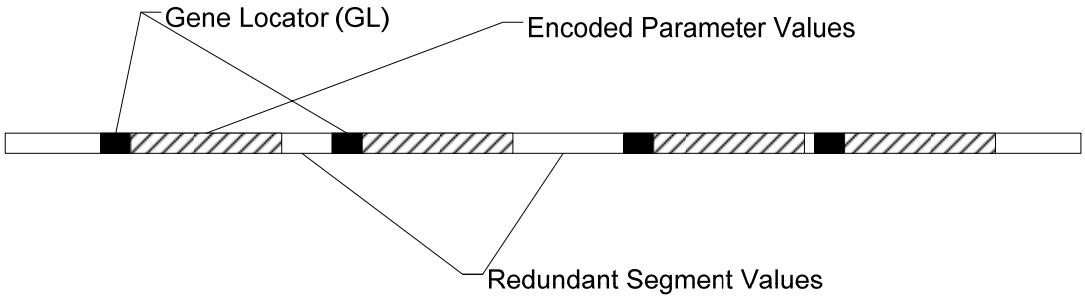

Figure 2.5. Representation format (Raich 1999)

2.3.2 Comparison and Performance of Implicit Redundant Representation GA with Structured GA

The dynamic characters of IRR GA in unstructured and non-continuous optimization domain problem are fundamentally different from the structured GA (Dasgupta 1994). Raich (1999) summarized the difference between two methods in their operations of redundancy:

1. The number of redundant blocks is constrained to a fixed number by maintaining a population of strings having the same number of redundant blocks specified in structured GA.
2. Each of the IRR GA strings has the randomly initialized location and length of the redundant segments.
3. The movement of the active or redundant segments within the strings is not allowed in the structured GA.

The enhanced performance of the IRR GA compared to the GA, SGA (Goldberg 1989) and the structured GA are also summarized by Raich (1999):

1. Although convergence to the maximum fitness is reached, redundant segments in the string lead to higher diversity in the population.
2. The gene instances which contain the binary or integer value and redundant segments are affected by the mutation operator.
3. Due to the possibility of crossover occurring in the redundant segments, disruption of building blocks caused by crossover will be reduced.
4. IRR GA does not constrain the number of parameters represented and it allows to varying within the strings in the population.

with diverse advantages of the IRR GA, the non-continuous and complex architectures of the 2-D and 3-D control devices and sensors optimization problem will be easily found.

2.4 Overview of Previous Research on Multi-objective Genetic Algorithm

In natural environments, there are many cases that can not be expressed with only one objective of the optimization problem, but need more than one objective to consider and express that sort of complex optimization problem. When the design criterion that should be met is more than one, it is called multicriteria, multi-attribute or multi-objective problem. Multi-objective problems can be expressed and solved by using composite objective weighting, non-Pareto population-based, and Pareto near-optimal criteria approaches. Thus in this section, these three methods are reviewed in detail.

2.4.1 Composite Objective Weighting Approaches

Composite objective weighting approach uses only one fitness function that is composed of two or more objectives and their weighting values. This can be an advantage of the composite objective weighting approach because it expresses a multi-objective problem as a simple one function optimization problem in treating the problem with GA. C in Equation (2.1) is a composite fitness function. All the objectives are weighted and summed and represented by one equation as Equation. (2.1) (Srinivas and Deb 1994):

$$C = \sum_{i=1}^{N} w_i f_i(x), \text{where } x \in X, \qquad \begin{cases} X: \text{ the feasilbe area} \\ \sum_{i=1}^{N} w_i = 1 \end{cases} \tag{2.1}$$

where N is number of objectives, w_i is the i-th weighting value of i-th objective, f_i.

Each objective can be weighted by the problem environment or researcher's priority of satisfying the objectives and engineering judgment. The advantage of this composite fitness function is that the emphasis of one objective over the other can be controlled and the obtained solution is a usually a Pareto-optimum solution. However, in general it is hard to decide beforehand how much weight to assign to each objective.

The other traditional method to represent the multi-objective problem is the scalarization. The single objective function derived from multiple objectives by using a demand level vector $\bar{y}$ is as follows (Srinivas and Deb 1994):

$$Z = \left[\sum_{i=1}^{N} \left| f(x) - \bar{y}_i \right|^r \right]^{1/r}, \qquad \begin{cases} 1 \le r < \infty, \\ X: \text{ the feasible area} \\ \bar{y}: \text{individual optima of objectives} \end{cases} \tag{2.2}$$

where N is number of objectives. The solution obtained by solving the above equation depends on the chosen demand-level vector. Naturally arbitrary selection of a demand level may be highly undesirable because a wrong demand level will lead to a non Pareto-optimal solution.

The min-max method attempts to minimize the relative deviations of the single objective functions from individual optimum. This means that it minimizes the conflict among objectives (Srinivas and Deb 1994);

$$\text{minimize F(x)} = \max\left[Z_j(x) \right], \quad \begin{cases} j = 1, 2, ..., N. \\ x \in X : \textit{feasible area} \\ Z_j(x) = \dfrac{f_j - \overline{f}_j}{\overline{f}_j} \end{cases} \tag{2.3}$$

where $\overline{f}_j > 0$. When the objectives have equal priorities and required to be optimized, this method yields best the possible optimal solution, but by the dimensionless weightings, the equality of each objective is broken (Srinivas and Deb 1994).

One of the composite fitness function methods is Hajela and Lin's (1992) evolutionary algorithm (HLEA) which employees a weighted and summed fitness assignment to the structural optimization problem. Although this approach provides a single optimal value from the multi optimal problem, this single optimal solution is highly dependent on the weights and coefficient of constraints. Syswerda and Palmucci (1991), and Jones et al. (1993) also applied the concept of the weighted sum approach. However, Richardson et al. (1989) found that small variations in the weights can result large changes in GA results.

2.4.2 Non-Pareto Population-based Approaches

Schaffer (1985) suggested vector evaluated genetic algorithm (VEGA) for the solving a problem with multicriteria functions. VEGA creates equal sized subpopulation for each criteria fitness, and individuals of the population are ranked by each criterion. Mating and crossover are performed each other freely as shown in Figure 2.6 (Schaffer 1985). The Figure 2.6 uses the N criteria. It can find some optimal tradeoff curves.

VEGA has two characters. One is a linear combination of the objectives and the other is a sub-population system. The linear weighted objective approaches in VEGA explains why the population tends to split into the speciation. However, the sub-population unit weighted concept provides much better result rather than just linear weighted objective approaches (Fonseca and Fleming 1995).

Although VEGA has better performance than blind linear weighted composite approaches, the weak point of VEGA is that the optimal values converge to the specific and extreme points related to each objective rather than to a suggest non-dominated Pareto optimal tradeoff curve, so VEGA can generate only moderately good values for all objectives not best value for any objective (Coello et al. 2001).

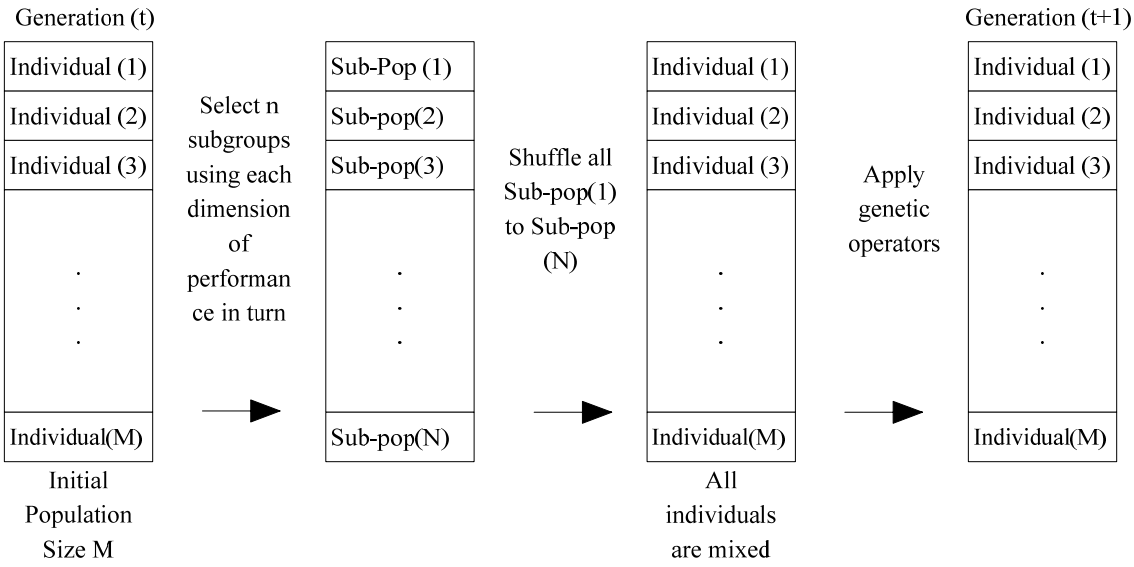

Figure 2.6. Schematic of VEGA selection mechanism

2.4.3 Pareto Optimal Criteria Approaches

The composite fitness function approach has some disadvantages in defining how much weight on each objective. In addition, Fonseca and Fleming (1995) also found that the composite fitness function approach becomes more difficult to perform when the number of objectives increases due to the subjectiveness of the weights and coefficients related to the constraints. Therefore, the weak point of composite fitness function approach is that it is hard to define the each weighting value from the problem statement without biased weighting to among objectives. By these reasons, Pareto optimal criteria approaches are suggested and studied by Goldberg (1989).

2.4.3.1 Concept of Pareto Optimal Curve and Non-dominated Solution

In most cases the objectives of the multi-objective problem will conflict with other state objectives. Therefore, all the feasible solutions do not allow simultaneous optimal solutions for all objectives. Namely, individual optimal solutions of each objective are usually different (Hans 1988). For example, when we want to find the optimal velocity and cost of fuel of vehicle in the limited driving distance, if the driver runs the vehicle with high speed, the car will consume more gas than when the driver drives slowly. Therefore, more than one point with satisfy each objective, but each point will not be dominated by the other points. These kinds of curves are called Pareto tradeoff optimal curve shown in Figure 2.7.

From Figure 2.7, points 1 to 5 are non-dominated because there are no points better than these points on all criteria, but the other '○' points are dominated by the non-dominated points: 1, 2, 3, 4, and 5. If there is a point in Area 1 in Figure 2.7, point 1 is dominated by the point of Area 1 and point 3 also can be applied to same rule. To express the concept of the Pareto optimality with a mathematical equation, ($x <_p$ y) means that x is partially less than y, when the following conditions hold (Goldberg 1989):

$$(x <_p \text{y}) \Leftrightarrow (\forall_i)(x_i \le y_i) \wedge (\exists i)(x_i < y_i) \tag{2.4}$$

from the Equation (2.4), we can say that x dominates y.

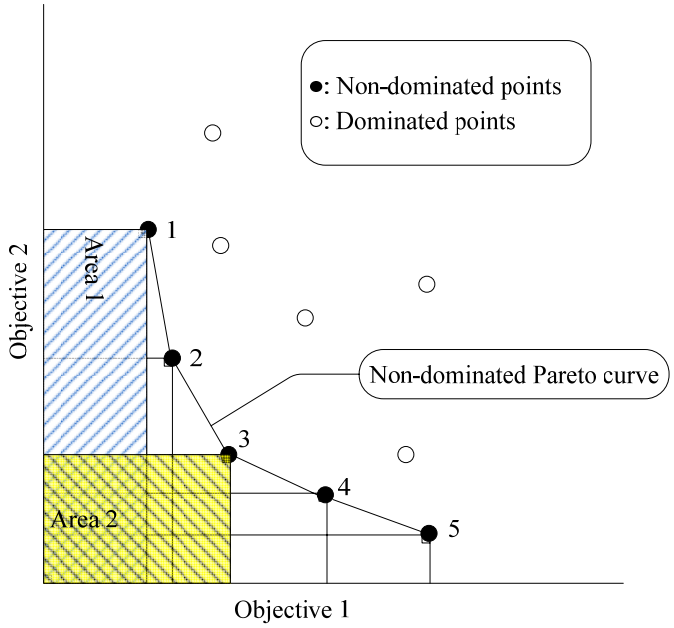

Figure 2.7. Non-dominated Pareto curve

To overcome the weak point of VEGA, Goldberg (1989) suggested the use of non-dominated Pareto ranking and selection to move a population towards the Pareto front in a multi-objective problem. To find the non-dominated Pareto curve, each one individual's objective function value and compare to all the population individuals, which determines the first non-dominated front. Without this first non-dominated front set, this same procedure is repeated until all the individuals in the population are assigned in a front with a rank. These rank assigned individuals can be used to select individuals as mating pool for the next population. The first non-dominated set has a higher probability to become chosen by the selection operator. However, from this non-dominated Pareto ranking mechanism, the optimal solution can be easily converged to local optima, so the sharing function (Goldberg 1989) is adopted to scatter the

individuals to feasible area more evenly. The concepts of sharing function and crowding operator to scatter the individuals to feasible area are discussed next.

2.4.3.2 Concept of Crowding Operator and Sharing Function

The premature and convergence to a single solution in evolutionary algorithms when they use a limited population set is a well-known phenomenon, even though the final goal is to find multiple global optima. This convergence phenomenon is called genetic drift (De Jong 1975). Holland (1975) suggested the usage of environmental niche and crowding operator to keep genetic drift from the genetic algorithm analysis. The role of the crowding operator is suggested to identify the situations that how many individuals dominate an environmental niche. In that case, the competition for the next generation in selection step increases rapidly. The individuals have lower possibility to survive in next generation. The percentage of the population that is allowed to reproduce is called generation gap, and the number of individuals initially selected as candidates to be replaced by a particular offspring is the crowding factor (Shrinivas and Deb 1994; Coello et al. 2001).

Sharing which is achieved by performing selection relies on the fundamental concept of the sharing function suggested by the Goldberg and Richardson (1987). Sharing function defines the degraded fitness values obtained by dividing the original fitness value of an individual by a quantity proportional to the number of individuals around it (Shrinivas and Deb 1994). Goldberg and Richardson (1987) defined a sharing function $sh(d_{ij})$, and the sharing function can be expressed as different function by

using the power factor α which is generally 1 as shown in Figure 2.8, but it will be dependent on the optimization problem characters. The general format of the sharing function is shown as follows (Goldberg and Richardson 1987):

$$sh(d_{ij}) = \begin{cases} 1 - \left(\dfrac{d_{ij}}{\sigma_{share}} \right)^{\alpha}, & \text{if } d_{ij} < \sigma_{share} \\ 0, & \text{otherwise} \end{cases} \tag{2.5}$$

where d_{ij} is the metric distance between individual string i and j, and σ_{share} is the sharing parameter or radius to control the range of the sharing.

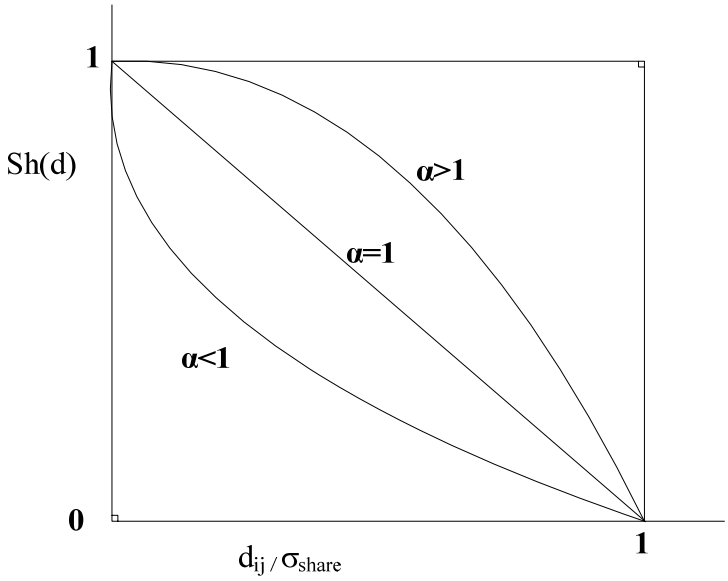

Figure 2.8. Power law sharing functions $sh = sh(d)$ (Goldberg and Richardson 1987)

From the sharing function, the modified fitness is achieved as (Goldberg and Richardson 1987):

$$f_{s_i} = \frac{f_i}{\sum_{j=1}^{M} sh(d_{ij})} \tag{2.6}$$

where the M is the number of individuals located in vicinity of the i-th individual. The p refers to the number of variables encoded in the EA, and d_{ij} is the p-dimensional Euclidean distance (Coello et al. 2001):

$$d_{ij} = \sqrt{\sum_{k=1}^{p} (x_{k,i} - x_{k,j})^2} \tag{2.7}$$

Although a σ_{share} value generally between 1 to 2 is used, or by performing several trials the value is defined, Deb and Goldberg (1989) suggested an equation to determine the value of sharing parameter (Coello et al. 2001):

$$\sigma_{share} = \frac{r}{\sqrt[p]{q}} = \frac{\sqrt{\sum_{k=1}^{p} (x_{k,\max} - x_{k,\min})^2}}{\sqrt[p]{2q}} \tag{2.8}$$

where r is the volume of a p-dimensional hypersphere of radius σ_{share} and q is the number of Pareto-solutions that the GA would like to find.

2.4.3.2 Multi-Objective Genetic Algorithms (MOGA)

Fonseca and Fleming (1993) suggested a modification to the general SGA at the selection level. The basic concepts of the MOGA are the ranking based on the Pareto dominance and sharing function. The pseudocode for determining rank is shown in Figure 2.9 (Fonseca and Fleming 1993). The Pareto dominance based rank is same as one plus the number which certain individual dominates as show in Figure 2.10. Thus a non-dominated individual's rank should be 1 and the other dominated individuals are

penalized by the degree of the population density. The main selection mechanism is that at first all current individuals are sorted according to the rank and assigned a fitness to individuals by interpolating from the best to worst. The best will have the largest value and according to specific function the other individuals also will be assigned fitness value. The average fitness values of same rank individuals are then calculated. The average fitness value is assigned to the same rank individuals. Therefore all individuals in same rank have the same probability to be selected for the next generation.

```
Initialize Population
Evaluate current Population
Assign individual ranks based on Pareto dominance
Calculate niche count
Assign sharing fitness in the way proposed by Goldberg (1989)
Assign Shared Fitness based on the average fitness of same rank individual

while iteration <= Generation
  Selection by means of stochastic universal sampling
  One point crossover
  Mutation
  Evaluate objective values
  Assign individual ranks based on Pareto dominance
  Calculate niche count
  Assign sharing fitness in the way proposed by Goldberg (1989)
  Assign Shared Fitness based on the average fitness of same rank individual
   iteration =iteration +1
end
```

Figure 2.9. Fonseca and Fleming's MOGA pseudocode

The main criticism of MOGA is that the block type of fitness assignment for individuals of the same rank is exposed to large selection pressure and it results in the premature of convergence of the population. It implies two different vectors with same objective function values and then performs sharing function. However it can not exist simultaneously in the population under this scheme. And the weak point is that the performance is dependent on the value of the sharing factor (Coello et al. 2001; Srinivas and Deb 1994).

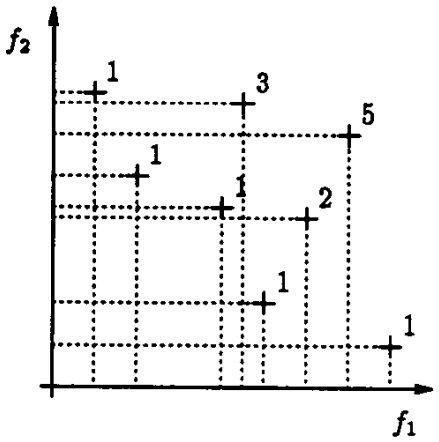

Figure 2.10. Multi-objective ranking based on the Pareto dominance (Fonseca and Fleming 1993)

2.4.3.4 Niched Pareto Genetic Algorithm (NPGA)

Horn and Nafpliotis (1993) proposed the niched Pareto genetic algorithm (NPGA) which uses the concept of tournament selection and a non-dominated Pareto set. A lot of research has considered multi-criteria optimization problem using the GAs to find all possible optimal tradeoffs among the multiple, conflicting objectives. Horn et al. (1994) suggested non-dominated Pareto set, in that there are no other solutions superior

in all attributes. In attribute space, the set of non-dominated solutions lie on a surface known as the Pareto optimal frontier. The goal of a Pareto GA is to find a representative sampling of solutions all along the Pareto front.

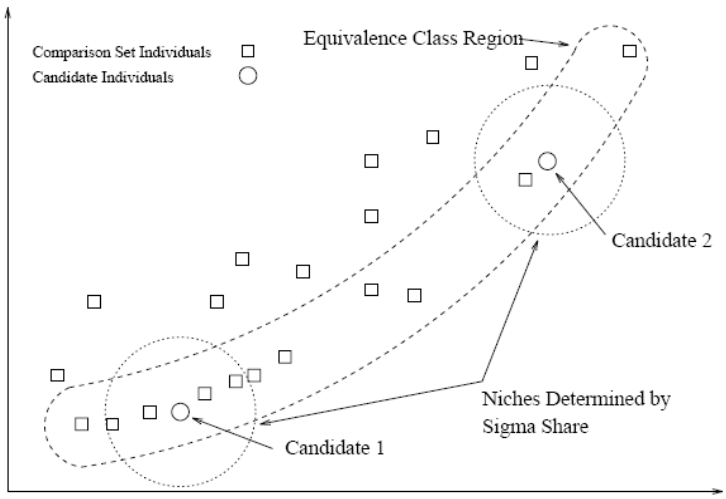

Figure 2.11. Equivalence class sharing (Horn et al. 1994)

From the Figure 2.11, fitness sharing only occurs in cases where both candidates are selected from the binary tournament selection are dominated or non-dominated. Fitness sharing was first suggested by the Goldberg and Richardson (1987) in order to prevent genetic drift and premature convergence to local optimal values and maintain the current population's genetic diversity. The most common sharing function is the triangular sharing function that is shown in Figure 2.7 as $\alpha = 1$. In this step, the researcher has to define how to degrade the fitness value and how to estimate the niche count means that how much crowded in predefined sharing radius or length. From the Figure 2.12, candidate 1 and 2 is on the same non-dominated Pareto front which means

that neither one has priority in selection. However, candidate 2 will be selected as next population because its niche count is smaller than the niche count of candidate 2. The sparser candidate will be selected for the next generation to imagine the genetic diversity and to prevent the convergence to local optimal points.

```
Initialize Population
while iteration <= Generation
  for j=1 to Population size
     NPGA's binary tournament selection
       if (candidate 1 is dominated)
          then (Select candidate 2)
       else if (candidate 2 is dominated)
         then (Select candidate 1)
       else if (both candidate dominated or not dominated)
         then (Perform predefined fitness sharing)
            then (Lower niche count candidate is returned)
  end loop j
Crossover
Mutation
Evaluate individuals
iteration =iteration +1
end
```

Figure 2.12. Horn and Nafpliotis's NPGA pseudocode

To define σ_{share}, Horn et al (1994) suggested the basis shown as Equation (2.9). The number of objectives is n since $Area_{pareto}$ which is the area of the Pareto frontier can be $n-1$ dimension.

$$Area_{niche}[\sigma_{share}]^{n-1} \approx \frac{Area_{pareto}}{N(Population\ size)} \tag{2.9}$$

where $Area_{pareto}$, is the size of Pareto front is also need to define with the minimum and maximum guideline. If the attributes (i.e., objectives or criteria) are two, the minimum Pareto front is the hypotenuse of the two attribute axes (Horn et al. 1994):

$$\min(Area_{pareto}) = \sqrt{\left|Obj_1^{best} - Obj_1^{worst}\right|^2 + \left|Obj_2^{best} - Obj_2^{worst}\right|^2} \tag{2.10}$$

The upper bound on the area of the Pareto surface is generally two-dimensional Euclidean distance which is the sum of the attribute ranges (Horn et al. 1994):

$$\max(Area_{pareto}) < \left(\left|Obj_1^{best} - Obj_1^{worst}\right| + \left|Obj_2^{best} - Obj_2^{worst}\right|\right) \tag{2.11}$$

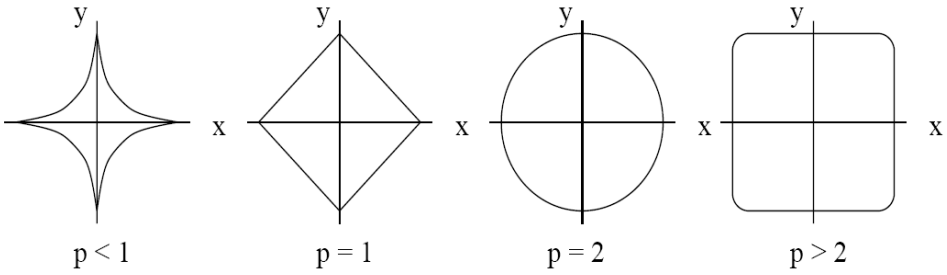

Figure 2.13. Niche shapes in two dimensions, as the degree p of the Holder metric varies (Horn et al. 1994)

The distances of the search area points are varying with the degree of the Holder metric as shown in Equation (2.12) and Figure 2.13 (Horn et al. 1994):

$$\text{Distance}(x, y) = \left(\sum_{i=1}^{n} \left|x_i - y_i\right|^p\right)^{1/p} \tag{2.12}$$

2.4.3.5 Non-dominate Sorting Genetic Algorithm (NSGA)

Srinivas and Deb (1994) developed the non-dominated sorting genetic algorithm (NSGA). NSGA offers an unbiased Pareto optimal set. NSGA only differs from the SGA in the selection operator used. The population is ranked on the basis of its non-domination characteristic. To prevent premature convergence of specific individuals and in order to maintain diversity and multiple optimal points, the sharing methods that were was discussed by Goldberg and Richardson (1987) are used.

The detailed procedure is that first, the non-dominated individuals are found and then each is given an equal reproductive potential value. Then the sharing method is applied by assigning a degraded fitness value that are obtained by dividing the equal reproductive potential value by a quantity proportional to the number of individuals around it with the Equation (2.13) (Goldberg and Richardson 1987). As explained in Figure 2.14, these classifying and sharing processes are performed on the entire population. Naturally the second new dummy set fitness value should be kept smaller than the minimum shared dummy fitness set.

$$Sh(d_{ij}) = \begin{Bmatrix} 1 - \left(\dfrac{d_{ij}}{\sigma_{share}} \right)^2, & \text{if } d_{ij} < \sigma_{share} \\ 0, & \text{otherwise} \end{Bmatrix} \tag{2.13}$$

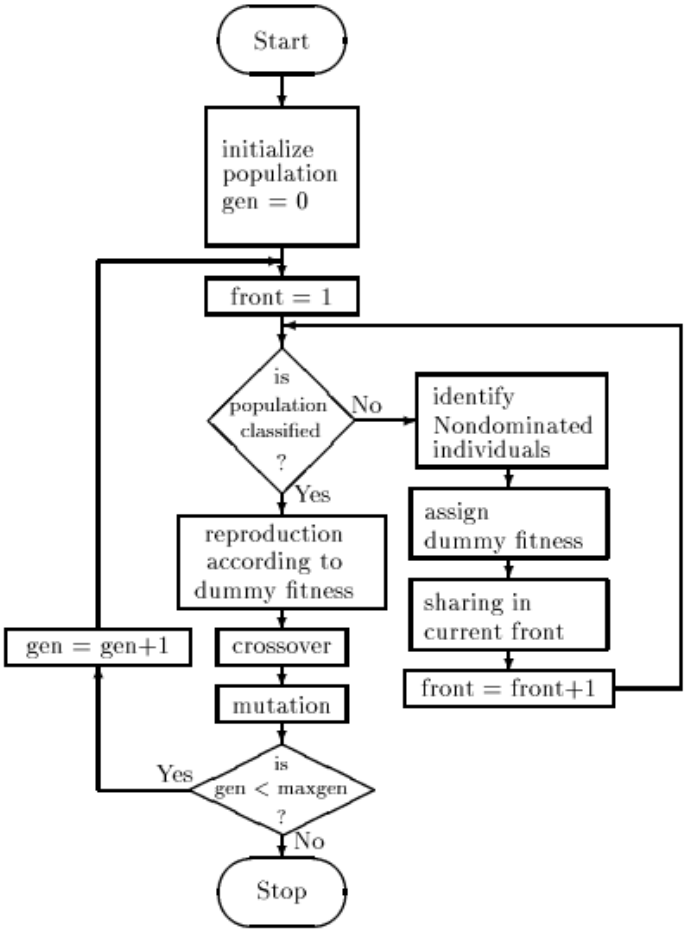

Figure 2.14. Flow chart of NSGA (Srinivas and Deb 1994).

Cheng and Li (1997) suggested the combined Pareto genetic algorithm. The niching mechanism is sharing the individual method to find non-dominated frontier. Cunha et al (1997) use the clustering concepts that reduced the number of non-dominated solutions stored without destroying the characteristics of the tradeoff front.

2.4.3.6 Strength Pareto Evolutionary Algorithm (SPEA)

Zitzler and Thiele (1999) proposed the strength Pareto evolutionary algorithm (SPEA) with combination of several features of previous multi-objective EA's in a unique manner. SPEA have some similarities in its process to other EA's (Zitzler and Thiele 1999):

- Stores the non-dominated solutions found so far in an external population
- Uses the concept of Pareto dominance in order to assign scalar fitness values to individuals
- Performs clustering to reduce the number of non-dominated solutions stored without destroying the characteristics of the tradeoff front

The originalities of the SPEA method are (Zitzler and Thiele 1999):

- Combines the above three techniques into a single algorithm
- Whether members of the population dominate each other is irrelevant. The individual fitness is calculated only from the solutions of external set.
- The individuals of the external set will participate in the selection process.
- The Pareto based new niching method which does not rely on any sharing or niche radius is developed to sustain diversity in the population.

```
Initialize Population (P)
Initialize empty external set (E)
Evaluate initial Population (P)

while iteration <= Generation
  for j=1 to Population size
      Non-dorminated sorting of (P)
      Copy non-dominated individuals of (P) to (E)
      Remove individuals in (E) which are dominated
      by any other member of (E)
      Do the clustering method to prune the exceeded number of size of (E)
      Calculate the fitness of individual in (P) and (E)
      Select individuals from (P)+(E) until the mating pool is filled
      with binary tournament selection
  end loop j

Problem-specific crossover
Mutation
iteration = iteration +1
end
```

Figure 2.15. Zitzler and Thiele' SPEA pseudocode

The SPEA's basic flow chart is shown in Figure 2.15. The non-dominated sorting and assigning fitness is a little bit different with previously reviewed GAs. The individuals in the archive set are ranked and then the individuals in current population are evaluated. The fitness of the archive set is defined as (Zitzler and Thiele 1999):

$$s_i = \frac{n}{N+1} \tag{2.14}$$

where n is the number of individuals in P that are covered by i, and N is the size of P.

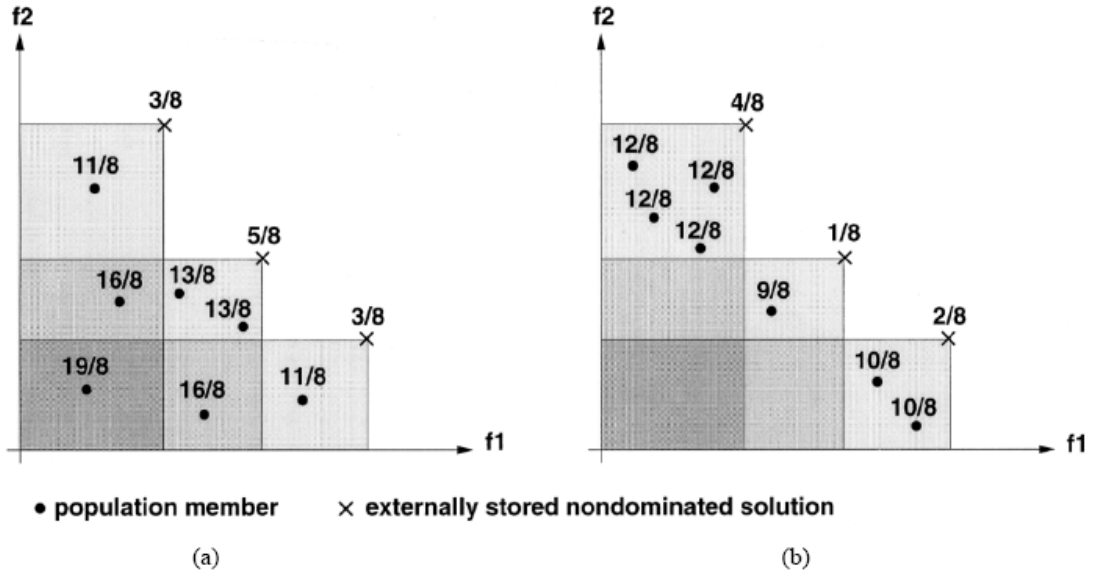

Figure 2.16. Fitness and strength calculation method of SPEA (Zitzler and Thiele 1999)

The fitness of the current population is calculated by the summing the strengths of all external non-dominated solutions i that cover j. The equation is expressed as (Zitzler and Thiele 1999):

$$f_i = 1 + \sum_{i,i \succeq j} s_i \tag{2.15}$$

where $f_j \in [1, N]$. The visual example is shown in Figure 2.16 and the N is 7.

When the size of archive set is over than the defined size, the clustering analysis is carried out and then prunes the inferior individuals. The *average linkage method* (Morse 1980) is suggested as clustering analysis. The first step is initialing cluster set C. And calculate the distance d of two cluster c_1 and c_2. The equation of d is expressed d (Zitzler and Thiele 1999):

$$d = \frac{1}{|c_1| \cdot |c_2|} \cdot \sum_{i_1 \in c_1, i_2 \in c_2} \|i_1 - i_2\| \tag{2.16}$$

Where the metric $\|\cdot\|$ is the distance between two individuals i_1 and i_2. And then determine two clusters c_1 and c_2 with minimal distance d. This chosen clusters is added to the larger cluster. Finally compute the reduced non-dominated set by selecting a representative individual per cluster (Zitzler and Thiele 1999).

2.4.3.7 Strength Pareto Evolutionary Algorithm2 (SPEA2)

A modified version of the SPEA is suggested by Zitzler et al. (2001, 2002a, 2002b) and gives the name as SPEA2. The main improvements are (Zitzler et al. 2001, 2002a, 2002b):

- A fine-grained fitness assignment strategy
- Density estimation technique
- Enhanced archive truncation method

The main loop of SPEA2 is as shown in Figure 2.17 (Zitzler et al. 2001, 2002a, 2002b). In the fitness assignment, both dominating and dominated solutions are considered simultaneously to avoid the situation that individuals are dominated by the same archived members as shown in Figure 2.18. As a first step to calculate the row fitness, the strength values of the current population and archive set are calculated. The strength value of an individual is the number of individuals it dominates in the current and archived population. With these defined strength values, the row fitness value is

determined by adding the strength values of its dominators in archive and current population (Zitzler et al 2001).

Initialize population (P)
Initialize empty external set (E)

while iteration <= Generation
  if Generation = 1
    Evaluate initial population (P) and assign fitness value
    Non-dominate sorting of (P)
    Copy of non-dominated individuals to external set (E)
    Copy next non-dominated individuals until fill external set (E)
    until fill the external set
    Perform binary tournament selection to fill mating pool (P_{t+1})
  else
    Evaluate population (P) and assign fitness value
    Non-dorminated sorting of (P) and (E)
    Copy non-dominated individuals of (P) and (E) to (E)
    Perform truncation operator to sustain the size of external set (E)
    Perform binary tournament selection to fill mating pool (P_{t+1})
  end

  Problem-specific crossover
  Mutation
  iteration = iteration +1
  end

Figure 2.17. Zitzler et al.'s SPEA2 pseudocode

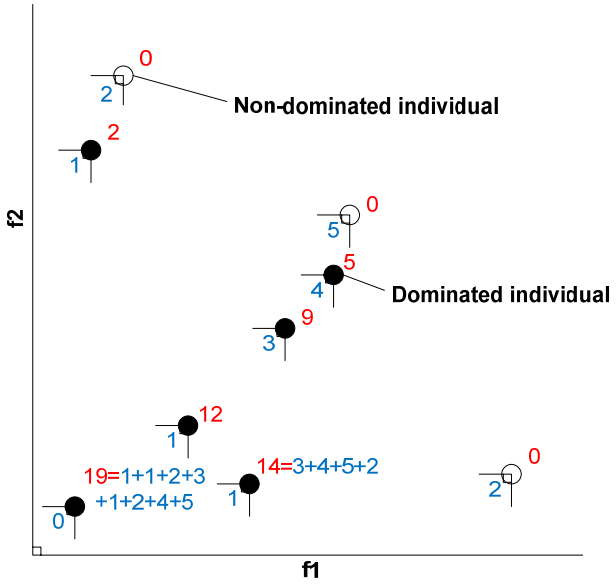

Figure 2.18. Fitness and strength calculation method of SPEA2

One difference in the SPEA2 mechanism is truncation operator. The truncation operator is performed to sustain the archive set size. In case when the archive size is smaller than the defined size, the number of shortage individual is coped from the current population to archive. In case when the archive size is bigger than the defined size, the individual that has the minimum distance to another individual is chosen for removal, but if there are several individuals with the same minimum distance, the tie is broken by considering the second smallest distances.

2.4.3.8 Non-dominated Sorting Genetic Algorithm2 (NSGA2)

Deb at al. (2000) suggested an enhanced version of NSGA to remove the disadvantages of the NSGA and improve its performance. The main criticism of the NSGA is as follows (Deb at al. 2000):

- High computational complexity of non-dominated sorting: In case of large population size, the population needs to be sorted every generation.
- Lack of elitism
- Need to specify the sharing parameter σ_{share}

With the non-dominated sorted current population P, to fill the population for the next generation E, the non-dominated fronts will be added to E until the size exceeds defined population size. The individuals in E are assigned the *crowding distance*. To estimate the density of individuals surrounding a particular point in the phenotype non-dominated Pareto front graph, the average distance of the two points on either side of this point along each of the objectives is used as *crowding distance.* This crowding distance is used for estimating of the size of the largest cuboid enclosing the point i without including any other point in the population. As shown in Figure 2.19, and Equation (2.17) , the crowding distance, $I[i]_{\text{distance}}$ is (Deb et al. 2000):

$$I[i]_{\text{distance}} = I[i]_{\text{distance}} + (I[i+1].m - I[i-1].m) \tag{2.17}$$

where

m : the number of objectives

$I[i]_{\text{distance}}$: the m-th objective function value of the i-th individual in the set I

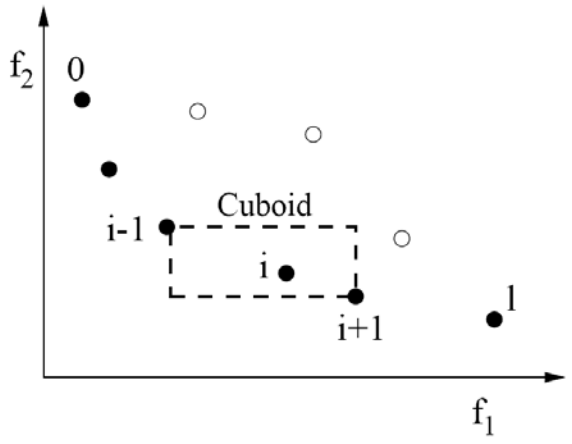

Figure 2.19. The crowding distance calculation (Deb et al. 2000)

The E is also sorted according to the crowded comparison operator. Thereafter, only the defined number of individuals will be selected as E. The crowded comparison operator is shown as follows in Equation (2.18) (Deb et al. 2000):

$$i \geq_n j \text{ if } (i_{rank} < j_{rank}) \text{ or } ((i_{rank} = j_{rank}) \text{ and } (i_{\text{distance}} > j_{\text{distance}})) \quad (2.18)$$

where

i_{rank} : Non-dominated rank
i_{distance} : Local crowding distance

From Equation (2.18), the individual with lower rank is preferable with differing non-domination ranks. If two points are in the same front, the individual which has the larger local crowding distance is selected. The pseudocode of The NSGA-II is shown in Figure 2.20.

```
Initialize population (P)
Evaluate fitness and objective values
Assign rank based on Pareto dominance
Binary tournament selection
Crossover
Mutation

while iteration <= Generation
    Evaluate population (P) and (E)
    Non-dorminated sorting of T(=P+E)
    Assign non-dominated fronts level
    Calculate crowding distance between points on each front
    Binary tournament selection
    Recombiantion and Mutation
    iteration = iteration +1
end
```

Figure 2.20. Deb et al.'s NSGA-II pseudocode

2.5 Overview of Previous Research on Optimization of Displacement of Active Damper and Sensor Using GA

The placement of control devices such as passive, active, and semi-active was not much considered in developing control algorithms or control system design. However, the locations and numbers of control devices and sensors in systems is very important in improving the performance of control system. In most cases, with predefined locations and numbers of damper and sensor, the control strategies were studied for the best efficiency and controllability. To determine actuator or sensor

configuration controlled system, several approaches were tried. The first one is based on system eigenvalues or controllability index and the second one is simulated annealing algorithm and the last one is a genetic algorithm based method.

2.5.1 System Eigenvalues and Controllability Approaches

Arbel (1981) studied the controllability measures and actuator placements in oscillatory systems to minimize the control energy, which is dependent on the concept of controllability and the optimization problem. DeLorenzo (1990) studied the specifying a sensor and actuator configuration for regulation of large-scale, linear, stochastic systems. The control algorithm is LQG controller and an efficient weight-selection technique is based on successive approximation. However these approaches have some limitations, including using a predefined number of dampers.

2.5.2 Simulated Annealing Based Approaches

The basic concept of simulated annealing comes from metallurgy. By heating and cooling metals, a hardened and stable material can be obtained. From this procedure, the current position of balls needs to be disturbed to go to lower position as shown in Figure 2.21. This mechanism of disturbing the balls to move to lower places and then getting more stable position is the main concept of simulated annealing algorithm.

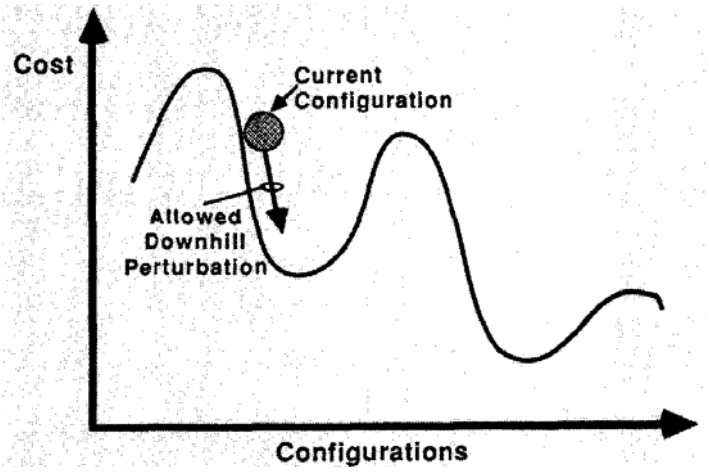

Figure 2.21. Configuration space: balls and hills (Rutenbar, R. A. 1989)

Chen et al (1991) tried to find the optimal placement of active/passive members in truss structures by applying simulated annealing. Simulated annealing is generic probabilistic algorithm to find global optimal values by using the repetition of simulating then annealing proven in metallurgy. The simulated annealing technique offers mechanism to allow climbing out of local optima. Although to create the next candidate solution, the heuristic technique use the worst-out-best-in, this simulated annealing algorithm create next candidate for the solution by randomly displacing a point by a small amount.

Moita et al (2006) also used the simulated annealing algorithm for optimal design in vibration control of adaptive structures. The control algorithm was an active controller and the piezoelectric sensors and actuators were applied to laminated reinforced composite structures. To minimize the vibration amplitude and maximize the first

natural frequency, control is initialized. To maximize the piezoelectric actuators efficiency, the positions of patches were also optimized.

The actuator placement optimization problem for statistical static distortion correction of truss structures was solved by GA and improved simulated annealing (ISA) algorithm (Onoda and Hanawa 1992). From the comparison of the results, the GA provided better results than the ISA or simulated annealing method. Therefore genetic algorithm based approaches will be discussed next.

2.5.3 Genetic Algorithm Based Approaches

Rao et al. (1991) first attempted to find the optimal locations of a pre-defined number of actuators in a two-bay truss using genetic algorithms. The linear quadratic regulator (LQR) was used as the active controller for the dissipation energy of active controller, and the dissipation energy is used as the objective function for maximization.

Dhingra and Lee (1995) studied the optimization of the feedback gains of LQR regulator and the optimum locations of collocated actuators/sensors, which trying tried to optimize the weight of simple truss structures using hybrid optimization method simultaneously. The GA and gradient-based search procedures were used as a hybrid method to reduce the analysis time of the original GA approach. The fundamental optimization process of hybrid method is (Dhingra and Lee, 1995);

1. Calculate the mass (M), stiffness (K), and damping (C) and determine frequencies and mode shapes of the structure.

2. Determine the plant matrix (A), input matrix (B), and the output matrix (S).
3. Check the observability and stability of the system.
4. Solve the LQR equation and determine the optimal gain matrix (G).
5. Evaluate objective functions and constraint function values.
6. Update all information and repeat step 1 to 6 until optimization criteria is satisfied.
7. And then the gradient-based search procedure is applied to find optimal design values.

Furuya and Haftka (1995) used the integer genetic algorithms to find the optimal placements of actuators in large space structures. To suggest some guidelines for determining the population size, crossover method such as single-point crossover, and two point crossover and selection method, several cases of parameters were treated.

Li et al. (2000) developed the multi-level genetic algorithm (MLGA) to find the optimal locations, number of dampers and optimal feedback gains simultaneously. The controller is LQR and it calculates control forces based on the system acceleration not on the displacement because the main goal of installing control systems under wind loads is to reduce the discomfort of the occupants and damage to sensitive equipment and nonstructural components in building (Li et al. 2000). The main procedure of MLGA is shown in Figure 2.19 and its detailed explanation is (Li et al. 2000):

1. Determine the number of actuators from the first level GA1, and send it to GA2, which determines the number of variables in GA2. Thus, the length of a chromosome in GA2 is determined according to the number of actuators.
2. The sizes of population and parameters of genetic operators in GA2 vary with the number of actuators because the length of individual changes with the number of actuators.
3. The second-level modules as shown in Figure 2.22 offers the necessary information to the first module with include the maximum acceleration and terminal and convergent parameters in the lower-level modules.

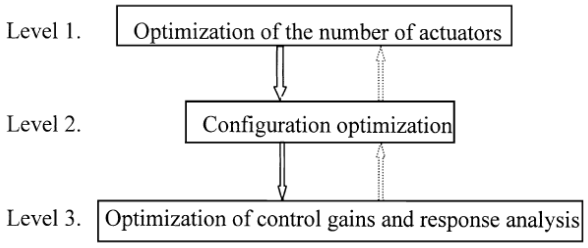

Figure 2.22. Block diagram of multi-level optimization design (Li et al. 2000)

Li et al (2004) developed a Two-level genetic algorithm (TLGA) to reduce the complexity of the MLGA to solve the same optimization problem as performed with the MLGA case (Li et al. 2000). Only one module in the lower level GA_{11} is required because to solve the sub-problem of optimal numbers, module GA_{11} is used to solve the sub-problem of optimal actuator numbers, and because to solve the optimal placement of

actuators, the module GA_{21} is used as shown in Figure 2.23. The sub-problem of optimal control force in the first level (Figure 2.22) is solved using the optimal control algorithm and is included in the configuration optimization sub-problem.

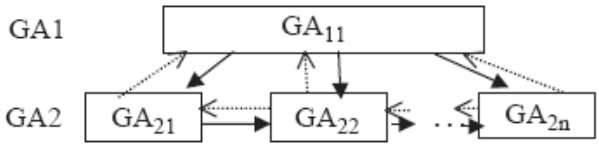

Figure 2.23. Structure of the TLGA (Li et al. 2004)

Abdullah et al. (2001) adopted the genetic algorithm and gradient-based optimization to optimally place the collocated sensor/actuator in structural system. As the controller, the Direct Velocity Feedback Controller (DVFC) developed by Chung et al. (1992) is used. The DVFC use the velocity of the system as sensor response at the position of the actuator, thus it need not to define the sensor location. This DVFC is nearly same with LQR controller without using feedback velocity as sensor. The optimization methods were genetic algorithms and gradient-based optimization techniques. To reduce the computational time of genetic algorithms, a generic gain was used to evaluate and rank the individual strings. This researcher's controller gain was suggested by Yang and Samali (1983) who studied the structure with identical properties and the gain was calculated by finding a single gain for the number of controllers in each first story. The main optimization procedure is shown in Figure 2.24. Richardson and Abdullah (2002) used real-coded genetic algorithm to solve the same collocated

sensor/actuator problem and to compare with this hybrid method combining binary genetic algorithm and gradient-based optimization techniques on a building structure.

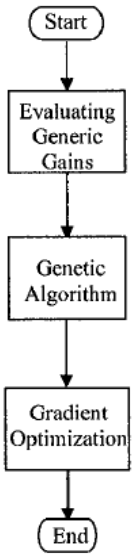

Figure 2.24. Flow chart of entire design procedure (Abdullah et al. 2001)

Ahlawat and Ramaswamy (2002) adopted a fuzzy logic controller (FLC) to drive an active mass driver (AMD) with tuned mass damper (TMD) in 2-D system. To design the hybrid mass damper (HMD) system, the multi-objective genetic algorithm was used. The objectives were to minimize the interstory drift normalized by the uncontrolled interstory drift and to maximize maximum of acceleration response of system normalized by the uncontrolled acceleration of system.

For the selection method, the two-branch tournament approach (Crossley et al. 1999) was applied. This approach does not use the non-dominated sorting to get Pareto optimal values; instead it uses a constant coefficient with a linear external function

strategy to give penalty with combining an objective function with the constraint functions.

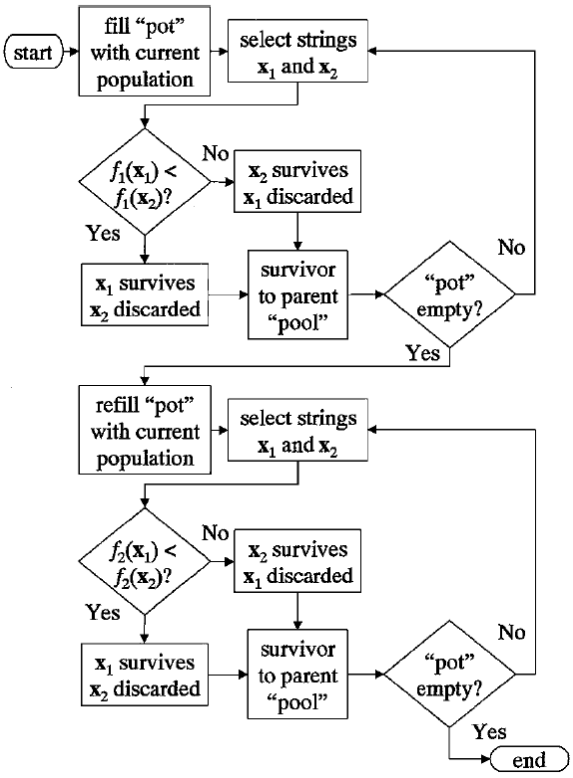

Figure 2.25. Two-branch tournament flowchart (Crossley et al. 1999)

The detailed procedure is shown in Figure 2.25. At first the entire current population is placed in a pool and then two individual are randomly chosen without replacement. The two chosen strings will be compared using the first objective and the better one will be selected. The process will continue until the pool is empty. The pool is refilled, and this process is repeated for the second objective. Finally with the selected population, the binary tournament selection is implemented.

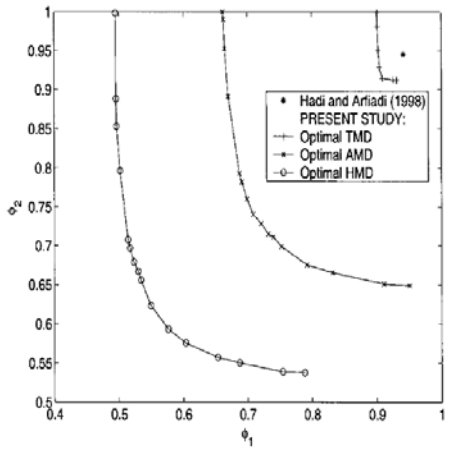

Figure 2.26. Variation of the objectives for the Pareto optimal designs (Ahlawat and Ramaswamy 2002).

The design variables were the mass of AMD, and stiffness, damping and mass of the TMD. The design variables of the FLC controller were the input and output functions. This paper also suggested the non-dominated Pareto curve for control design as shown in Figure 2.26. However this multi-objective genetic algorithm is pretty same with VEGA, and then it has the disadvantage of VEGA that the optimal values converge to the specific and extreme points related to each objective rather than to suggest a non-dominated Pareto optimal tradeoff curve. Furthermore FLC also used the predefined location of sensors.

3 DESCRIPTION OF MODELING OF STRUCTURES

A lot of investigations on approaches to protect human being properties and lives and enhance the performance of retrofits to the natural hazards have been suggested and performed utilizing their own civil structures such as bridges and buildings in the last twenty years. However, the efficacy of performances of the developed control algorithms or devices could not be compared because of differences of the structural systems, which limited future research and development. The American Society of Civil Engineers (ASCE) Committee on Structural Control has recognized the necessity of structural control benchmark problems to address this issue. In order for a broader impact, the ASCE committee on structural control chose to use buildings designed for the SAC Project (Joint venture of three non-profit organizations: The Structural Engineers Association of California (SEAOC), the Applied Technology Council (ATC) and California Universities for Research in Earthquake Engineering (CUREE)) project (Spencer et al. 1999). The 3-story building model was suggested and the first results were reported at the ASCE Structures Congress, held in Portland, Oregon by the Balas (1998), Lu and Skelton (1998), Spencer et al (1997), and Wu et al (1997). And for more practical comparison of results of developed control systems, the committee suggested the 20-story steel building as the benchmark structure.

To verify and suggest more advanced optimal configurations of active dampers and sensors, this study uses the ASCE Structural Control 3- and 20-story benchmark

building and its suggested model (Ohtori 2004), and modified the model and then developed 3-dimentional (3-D) 20-story benchmark building model.

3.1 Description and Modeling of 2-D 3-Story Benchmark Control System

The 3-, 9-, and 20-story building were designed by Brandow & Johnston Associates (1996) for the SAC Phase II Steel Project. These three building structures were suggested as ASCE structural control benchmark system. To perform this study, 3-story building was modeled using finite element method (FEM) to determine the stiffness and mass matrix of the equation of motion. Although ASCE structural control benchmark program offered a Matlab R2007b code from the website: http://sstl.cee.uiuc.edu/, it was necessary to modify that model and then generate a 2-D 3 degree-of-freedom (DOF) system with per floor level one DOF.

This 3-story building is 120 ft by 180 ft in plan, and 39 ft in elevation. Each bay which is one of the structural components, is 30 ft with four bays in the north-south (N-S) direction and six bays in the east-west (E-W) direction. The exterior beams of the 3-story building are steel moment resisting frames (SMRF) and the interior beams are simple framing with composite floors. The columns of the structure are 50 ksi steel with wide flange as shown in Figures 3.1 and 3.2. The floor system is composed of wide flange beams with the floor slab. Moreover floor system provides diaphragm action which is assumed to be rigid in the horizontal plane (Barroso 1999; Ohtori 2004).

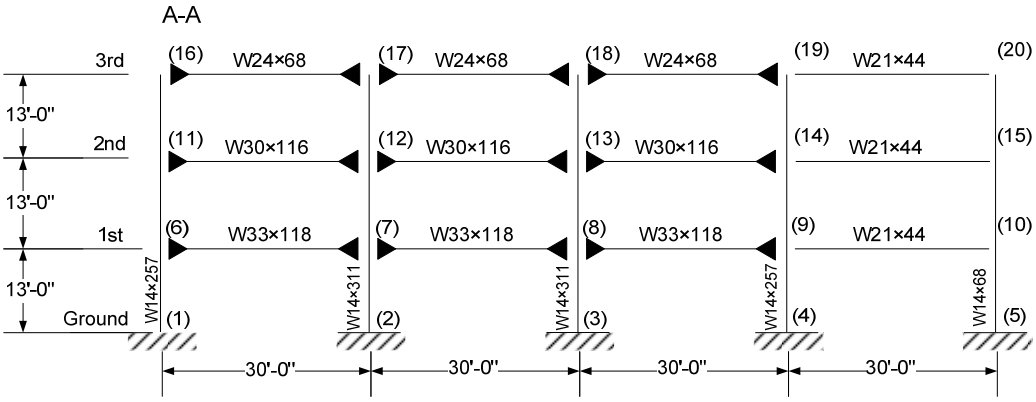

Figure 3.1. 3-story benchmark building N-S steel moment-resisting frame

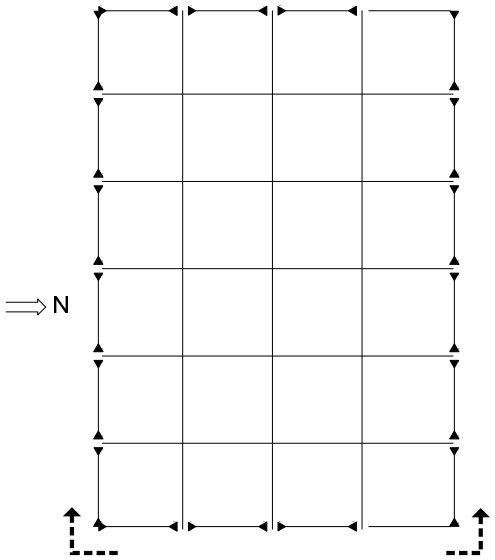

Figure 3.2. 3-story building plan

The weak N-S direction is selected for the control analysis for the 2-D 3-story problem. Each moment resisting frame (MRF) resists only one-half of the seismic mass related to the entire structure. To establish the equation of motion, the mass, stiffness,

and damping matrix are determined. To find these matrices, the structure unit (Figure. 3.1) is used. The mass matrix is defined by shifting the slave element's mass to master mass. The 8, 13 and 18 nodes are the each story's master node. And finally the 3×3 diagonal mass matrix $\mathbf{M}_{\mathbf{2D3s}}$ is determined.

To determine the global stiffness matrix, the columns and beams of the systems are modeled as plane-frame elements. The nodes are assigned at beam-to-column joints and each node has three degree of freedoms (DOF), the horizontal, vertical and rotational components are expressed as shown in Figure 3.3.

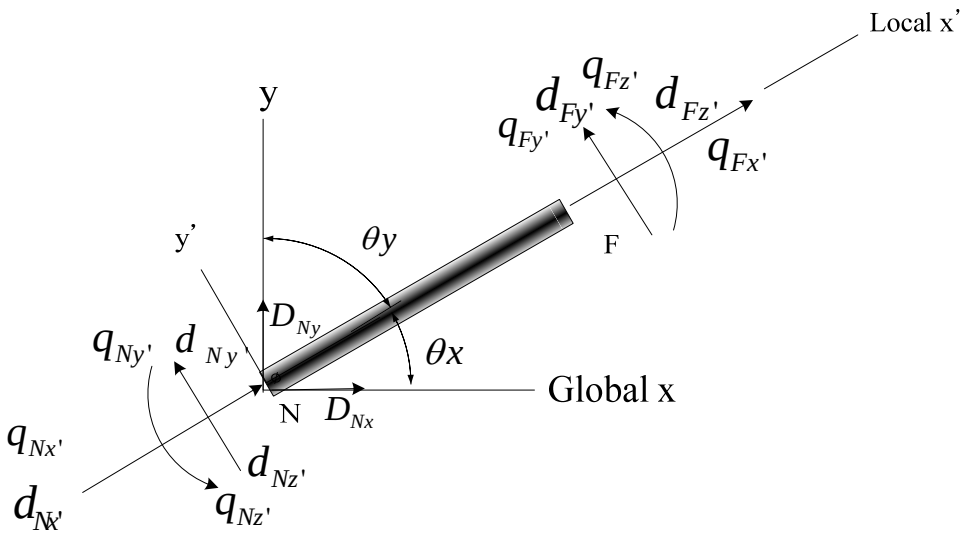

Figure 3.3. Element force and displacement coordinate systems

The global stiffness matrices are assembled from elements of each story and static condensation. The rotational and vertical stiffness and the mass of each element is ignored. To find the stiffness matrix, $\mathbf{K}_{\mathbf{2D3s}}$ of 3-story building system, the frame-

member stiffness method is used. Each element, such as columns, beams, and bracings is expressed as shown in Figure.3.3 and its governing equation is $\mathbf{q} = \mathbf{kd}$. Where $\mathbf{q}$ is the internal member load matrix, $\mathbf{k}$ is the symmetric stiffness matrix, and $\mathbf{d}$ is the deformation matrix in the local coordinate. When the element's boundary condition is fixed end, the exact expression of each beam force and displacement relationship is expressed with Equation (3.1), and if the boundary condition of the element is pinned connection, the exact stiffness local k matrix is Equation (3.2) (Hibbeler 1999).

$$\begin{bmatrix} q_{Nx'} \\ q_{Ny'} \\ q_{Nz'} \\ q_{Fx'} \\ q_{Fy'} \\ q_{Fz'} \end{bmatrix} = \begin{bmatrix} \frac{AE}{L} & 0 & 0 & -\frac{AE}{L} & 0 & 0 \\ 0 & \frac{12AE}{L^3} & \frac{6EI}{L^2} & 0 & -\frac{12AE}{L^3} & \frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{4AE}{L} & 0 & -\frac{6EI}{L^2} & \frac{2AE}{L} \\ -\frac{AE}{L} & 0 & 0 & \frac{AE}{L} & 0 & 0 \\ 0 & -\frac{12AE}{L^3} & -\frac{6EI}{L^2} & 0 & \frac{12AE}{L^3} & -\frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{2AE}{L} & 0 & -\frac{6EI}{L^2} & \frac{4AE}{L} \end{bmatrix} \begin{bmatrix} d_{Nx'} \\ d_{Ny'} \\ d_{Nz'} \\ d_{Fx'} \\ d_{Fy'} \\ d_{Fz'} \end{bmatrix} \tag{3.1}$$

$$\begin{bmatrix} q_{Nx'} \\ q_{Ny'} \\ q_{Nz'} \\ q_{Fx'} \\ q_{Fy'} \\ q_{Fz'} \end{bmatrix} = \begin{bmatrix} \frac{AE}{L} & 0 & 0 & -\frac{AE}{L} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{AE}{L} & 0 & 0 & \frac{AE}{L} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} d_{Nx'} \\ d_{Ny'} \\ d_{Nz'} \\ d_{Fx'} \\ d_{Fy'} \\ d_{Fz'} \end{bmatrix} \tag{3.2}$$

where A is the section area, E is the modulus of elasticity, I is the second moment of inertia, and L is the length of each element.

Stiffness matrix, **k** is for the local coordinate, so we need to be able to transform the internal member loads **q** and deformations **d** from local coordinates to global coordinates and then the transformation matrices are required. From the Figure 3.3, the local and global displacement relationship is found (Hibbeler 1999):

$$\begin{aligned} d_{Nx'} &= D_{Nx}\cos\theta_x, \quad d_{Ny'} = -D_{Nx}\cos\theta_y \\ d_{Nx'} &= D_{Ny}\cos\theta_y, \quad d_{Ny'} = -D_{Ny}\cos\theta_x, \quad d_{Nz'} = D_{Nz} \end{aligned} \tag{3.3}$$

$$\begin{aligned} d_{Fx'} &= D_{Fx}\cos\theta_x, \quad d_{Fy'} = -D_{Fx}\cos\theta_y \\ d_{Fx'} &= D_{Fy}\cos\theta_y, \quad d_{Fy'} = -D_{Fy}\cos\theta_x, \quad d_{Fz'} = D_{Fz} \end{aligned} \tag{3.4}$$

with the defining new variables for convenience $\cos\theta_x = \lambda_x$ and $\cos\theta_y = \lambda_y$, finally the local displacement, d is transformed by the global transformation relationship, T . T is expressed as a matrix with form as in Equation (3.5) (Hibbeler 1999).

$$\mathbf{d = TD}$$

$$\begin{bmatrix} d_{Nx'} \\ d_{Ny'} \\ d_{Nz'} \\ d_{Fx'} \\ d_{Fy'} \\ d_{Fz'} \end{bmatrix} = \begin{bmatrix} \lambda_x & \lambda_y & 0 & 0 & 0 & 0 \\ -\lambda_y & \lambda_x & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda_x & \lambda_y & 0 \\ 0 & 0 & 0 & -\lambda_y & \lambda_x & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} D_{Nx'} \\ D_{Ny'} \\ D_{Nz'} \\ D_{Fx'} \\ D_{Fy'} \\ D_{Fz'} \end{bmatrix} \tag{3.5}$$

where **D** is the global displacement matrix.

The local load components also need to be transformed to global coordinates to determine the global stiffness matrix, **K** . The local stiffness matrix values of each node are mapped to the global matrix, and then the stiffness values of the slave nodes and elements of each story are shifted to the master elements. The matrix size can be 9×9 matrix because the vertical and rotational stiffness values are ignored in the 2-D problem.

The detail relationship of global and local element forces and displacements is expressed in Figure 3.4.

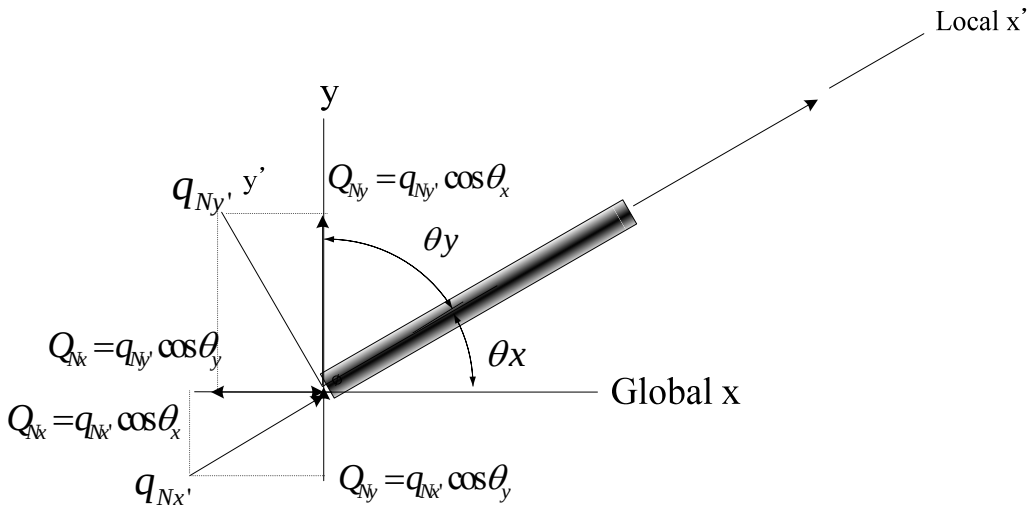

Figure 3.4. Relationship of global and local element force and displacement in coordinate systems

with the mathematical expression (Hibbeler 1999):

$$\begin{aligned} &Q_{Nx} = q_{Nx'} \cos\theta_x, \quad Q_{Ny} = q_{Nx'} \cos\theta_y \\ &Q_{Nx} = -q_{Ny} \cos\theta_y, \quad Q_{Ny'} = -q_{Ny} \cos\theta_x, \quad Q_{Nz} = q_{Nz'} \end{aligned} \tag{3.6}$$

$$\begin{aligned} &Q_{Fx} = q_{Fx'} \cos\theta_x, \quad Q_{Fy} = -q_{Fx'} \cos\theta_y \\ &Q_{Fx} = -q_{Fy'} \cos\theta_y, \quad Q_{Fy} = -q_{Fy'} \cos\theta_x, \quad Q_{Fz} = q_{Fz'} \end{aligned} \tag{3.7}$$

Therefore, the transformation matrix is expressed as Equation (3.8) (Hibbeler 1999):

$$\begin{bmatrix} Q_{Nx} \\ Q_{Ny} \\ Q_{Nz} \\ Q_{Fx} \\ Q_{Fy} \\ Q_{Fz} \end{bmatrix} = \begin{bmatrix} \lambda_x & -\lambda_y & 0 & 0 & 0 & 0 \\ \lambda_y & \lambda_x & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda_x & -\lambda_y & 0 \\ 0 & 0 & 0 & \lambda_y & \lambda_x & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} q_{Nx'} \\ q_{Ny'} \\ q_{Nz'} \\ q_{Fx'} \\ q_{Fy'} \\ q_{Fz'} \end{bmatrix} \tag{3.8}$$

where $\mathbf{Q} = \mathbf{T}^{\mathrm{T}}\mathbf{q}$, and $\mathbf{Q}$ is the member's global force component matrix.

Equations (3.5) and (3.8) need to be combined to express the global force, stiffness and displacement relationship (Hibbeler 1999):

$$\mathbf{q} = \mathbf{kTD} \tag{3.9}$$

$$\mathbf{Q} = \mathbf{T}^{\mathrm{T}}\mathbf{q} \tag{3.10}$$

$$\mathbf{Q} = \mathbf{T}^{\mathrm{T}}\mathbf{kTD} \tag{3.11}$$

If the global stiffness matrix is defined by the above equations, the vertical coordinate values can be neglected. Slave node stiffness values of each story will be added to stiffness value of master node, and then the slave nodes and vertical and rotational nodes will be eliminated by using the static condensation. Finally the master horizontal and rotational components remain. The final 3 DOFs will be obtained by (Chopra 2000):

$$\begin{bmatrix} \mathbf{m}_{tt} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{u}}_t \\ \ddot{\mathbf{u}}_0 \end{Bmatrix} + \begin{bmatrix} \mathbf{k}_{tt} & \mathbf{k}_{t0} \\ \mathbf{k}_{0t} & \mathbf{k}_{00} \end{bmatrix} \begin{Bmatrix} \mathbf{u}_t \\ \mathbf{u}_0 \end{Bmatrix} = \begin{Bmatrix} \mathbf{p}_t(\mathbf{t}) \\ \mathbf{0} \end{Bmatrix} \tag{3.12}$$

$$\mathbf{m}_{tt}\ddot{\mathbf{u}}_t + \mathbf{k}_{tt}\mathbf{u}_t + \mathbf{k}_{t0}\mathbf{u}_0 = \mathbf{p}_t(\mathbf{t}) \tag{3.13}$$

$$\mathbf{k}_{0t}\mathbf{u}_t + \mathbf{k}_{00}\mathbf{u}_0 = \mathbf{0} \tag{3.14}$$

$$\mathbf{u}_0 = -\mathbf{k}_{00}^{-1}\mathbf{k}_{0t}\mathbf{u}_t \tag{3.15}$$

where $\mathbf{m}_{tt}$, $\mathbf{k}_{tt}$, and $\mathbf{p}_t$ are each the mass, stiffness, and external load matrices of the horizontal direction. Each $\ddot{\mathbf{u}}_t$, $\ddot{\mathbf{u}}_0$, $\mathbf{u}_t$, and $\mathbf{u}_0$ denotes the accelerations of the mass matrix and zero mass matrix and displacement of mass matrix and zero matrix. $\mathbf{k}_{t0} = \mathbf{k}_{0t}^{\mathrm{T}}$ is known as dynamic DOFs. The Equation (3.12) is partitioned to Equation

(3.13), and (3.14). From the Equation (3.14), $\mathbf{u_0}$ is expressed as Equation (3.15). By substituting Equation (3.15) in Equation(3.12):

$$\mathbf{m_{tt}\ddot{u}_t} + \hat{\mathbf{k}}_{\mathbf{tt}}\mathbf{u_t} = \mathbf{p_t(t)} \tag{3.16}$$

$$\hat{\mathbf{k}}_{\mathbf{tt}} = \mathbf{k_{tt}} - \mathbf{k_{0t}}^{\mathbf{T}}\mathbf{k_{00}k_{0t}} \tag{3.17}$$

The final equation of motion is expressed with the 3×3 matrices as shown in Equation (3.16). The condensed stiffness matrix is expressed as $\hat{\mathbf{k}}_{\mathbf{tt}}$.

The seismic mass for the N-S direction is $\mathbf{M_{2D3s}}(kg)$, $\mathbf{K_{2D3s}}(kN/m)$, and mode shapes are:

$$\mathbf{M_{2D3s}} = \begin{bmatrix} 478350 & 0 & 0 \\ 0 & 478350 & 0 \\ 0 & 0 & 517790 \end{bmatrix} \begin{bmatrix} 1st \\ 2nd \\ 3rd \end{bmatrix} \tag{3.18}$$

$$\mathbf{K_{2D3s}} = 10^8 \times \begin{bmatrix} 4.3750 & -2.3762 & 0.4121 \\ -2.3762 & 3.1513 & -1.2966 \\ 0.4121 & -1.2966 & 0.9438 \end{bmatrix} \begin{bmatrix} 1st \\ 2nd \\ 3rd \end{bmatrix} \tag{3.19}$$

$$\mathbf{\Phi} = \begin{bmatrix} 0.2240 & 0.6216 & -0.7590 \\ 0.5358 & 0.5892 & 0.6239 \\ 0.8141 & -0.5163 & -0.1865 \end{bmatrix} \tag{3.20}$$

The mass matrix, $\mathbf{M_{2D3s}}$ is also determined by the same process with out the calculation of the each stiffness of element in local coordinate. With the one-half of the seismic mass to the entire structure, the mass matrix is defined by shifting the slave element's mass to master mass.

The damping matrix is defined based on the *Rayleigh Damping* with reduced system and the assumption of modal damping. Damping in the first mode of vibration 2% with a maximum of 10% critical damping in any one mode and then the final damping matrix is given by (Spencer et al. 1999):

$$\mathbf{C}_{\mathbf{2D3s}} = \mathbf{M}_{\mathbf{2D3s}}\mathbf{\Phi}\begin{bmatrix} 2\zeta_1\omega_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & 2\zeta_n\omega_n \end{bmatrix}\mathbf{\Phi}^{-1} \tag{3.21}$$

Thus the final value of 2-D 3-story building damping matrix with unit $kN-s/m$ is:

$$\mathbf{C}_{\mathbf{2D3s}} = 10^5 \times \begin{bmatrix} 7.7510 & -3.7180 & 0.6449 \\ -3.7180 & 5.8363 & -2.0288 \\ 0.6449 & -2.0288 & 2.4570 \end{bmatrix}\begin{bmatrix} 1st \\ 2nd \\ 3rd \end{bmatrix} \tag{3.22}$$

thus the final equation of motion for the 2-D 3-story building is expressed in Equation (3.23):

$$\mathbf{M}_{\mathbf{2D3s}}\ddot{\mathbf{x}} + \mathbf{C}_{\mathbf{2D3s}}\dot{\mathbf{x}} + \mathbf{K}_{\mathbf{2D3s}}\mathbf{x} = -\mathbf{M}_{\mathbf{2D3s}}\mathbf{G}_{\mathbf{2D3s}}\ddot{x}_g + \mathbf{P}_{\mathbf{2D3s}}\mathbf{f}_{\mathbf{2D3s}} \tag{3.23}$$

where $\mathbf{G}_{\mathbf{2D3s}}$ is vector defining the loading of ground acceleration onto the horizontal degrees of freedom, $\mathbf{P}_{\mathbf{2D3s}}$ is vector defining the loading of control forces onto the structure, and $\mathbf{f}_{\mathbf{2D3s}}$ is vector of forces produced by the control devices.

The natural frequencies of the 3-story structural system model are compared with others who have modeled this structure & indicate good agreement (Barroso 1999; Ohtori 2004) as shown in Table 3.1.

Table 3.1. Comparison of natural frequencies

	1st (Hz)	2nd (Hz)	3rd (Hz)
Current model	0.9982	3.0705	5.8349
Barroso (1999)	0.9804	3.0303	5.8824
Ohtori (2004)	0.99	3.06	5.83

In a linear system, when the input is applied to system, one can see that the output of the system to a steady-state load consists of a superposition of harmonics. This initial response contains a *transient response component* and the overall response will settle down after long time elapses after the input was applied. This stable and settled response is the *steady-state response*. If the transient response decays to zero after some time and steady-state response remains, the system is called *stable.* To check the characteristic of a linear system, applying a harmonic input to the system is termed a *frequency response analysis*. Generally a complex quantity is expressed in terms of its *magnitude* and the *phase* in stead of real and imaginary parts in control system. A complex quantity is represented in complex space with *phasor.* The magnitude is the length of *phasor* and phase is the direction of *phasor*. The plot of magnitude and phase in the frequency domain is very useful, because the range of frequencies required to study a linear system is very large (Tewari 2002). The magnitude is usually converted to gain in decibels (dB) by using following equation (Tewari 2002):

$$Gain = 20\log_{10}\left|G(i\omega)\right| \tag{3.24}$$

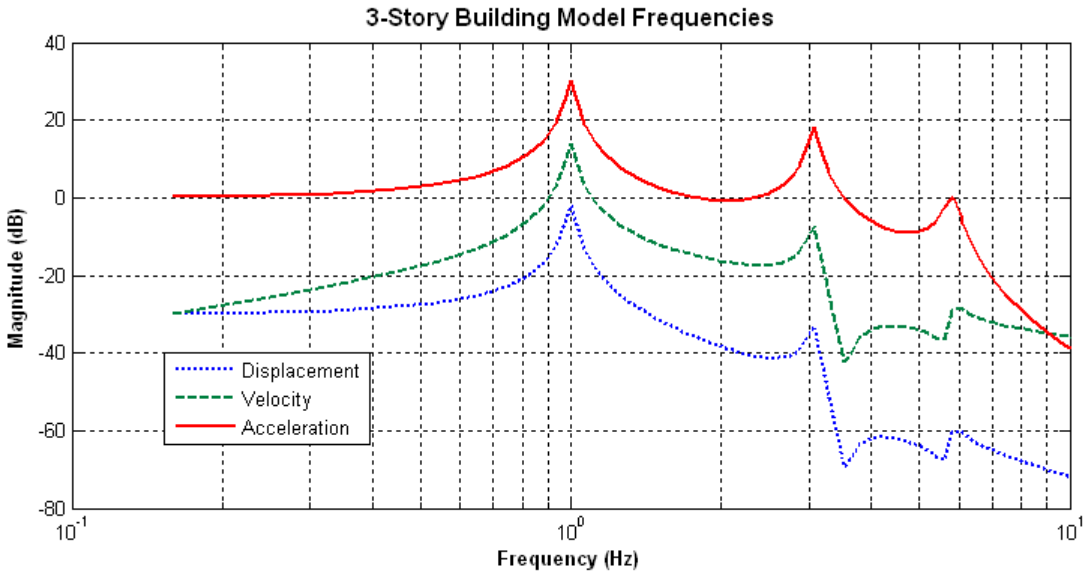

Figure 3.5. 3-story building model transfer functions from ground acceleration to absolute roof acceleration, velocity, and displacement

From Table 3.1 and Figure 3.5, the 3-story benchmark model for this research is compared to other models of this structure published in the literature & the suggested data of the structural model. The current model shows good agreement to the other researcher's results (Barroso 1999; Ohtori 2004). The resulting mode shapes for this structural system are presented in Figure. 3.6.

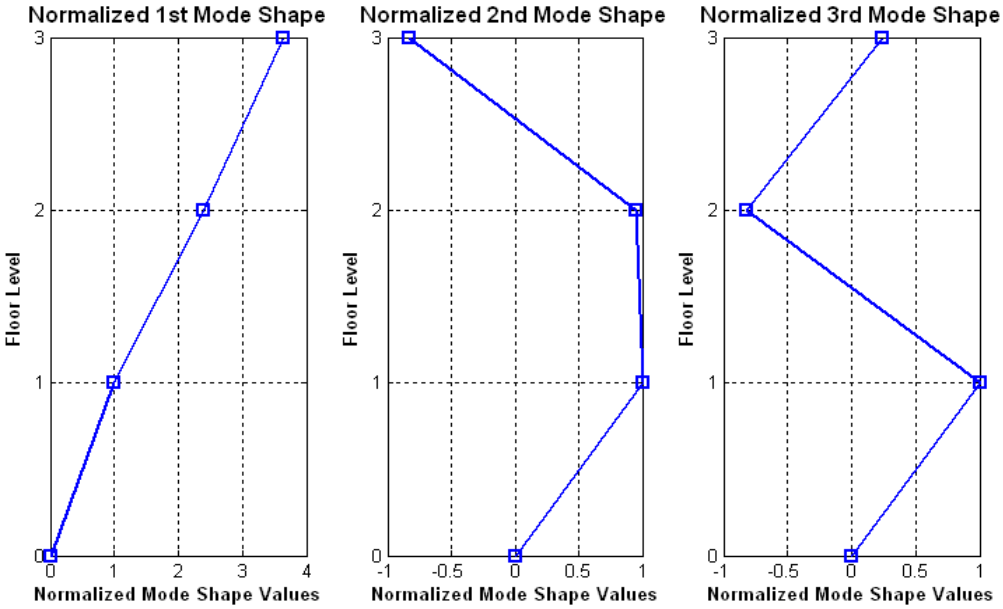

Figure 3.6. 3-story building model first three mode shapes

3.2 Description of 2-D 20-Story Benchmark Control System

For this research, the 20-story building was also modeled using FEM. Although this 2-D 20-story ASCE structural control benchmark building was modeled with the commercial program Matlab coding, the coding is modified to fit the needs of this research and then simplified t a 2-D 20 DOF system. This 20-story building is 100 ft by 120 ft in plan, and 265 ft in elevation. Each bay is 20 ft with five bays in the north-south (N-S) direction and six bays in the east-west (E-W) direction. The exterior beams of building are steel moment resisting frames and the interior beams are simple framing with composite floors. The columns of structure are 50 ksi steel with wide flange as shown in Figures. 3.7 and 3.8.

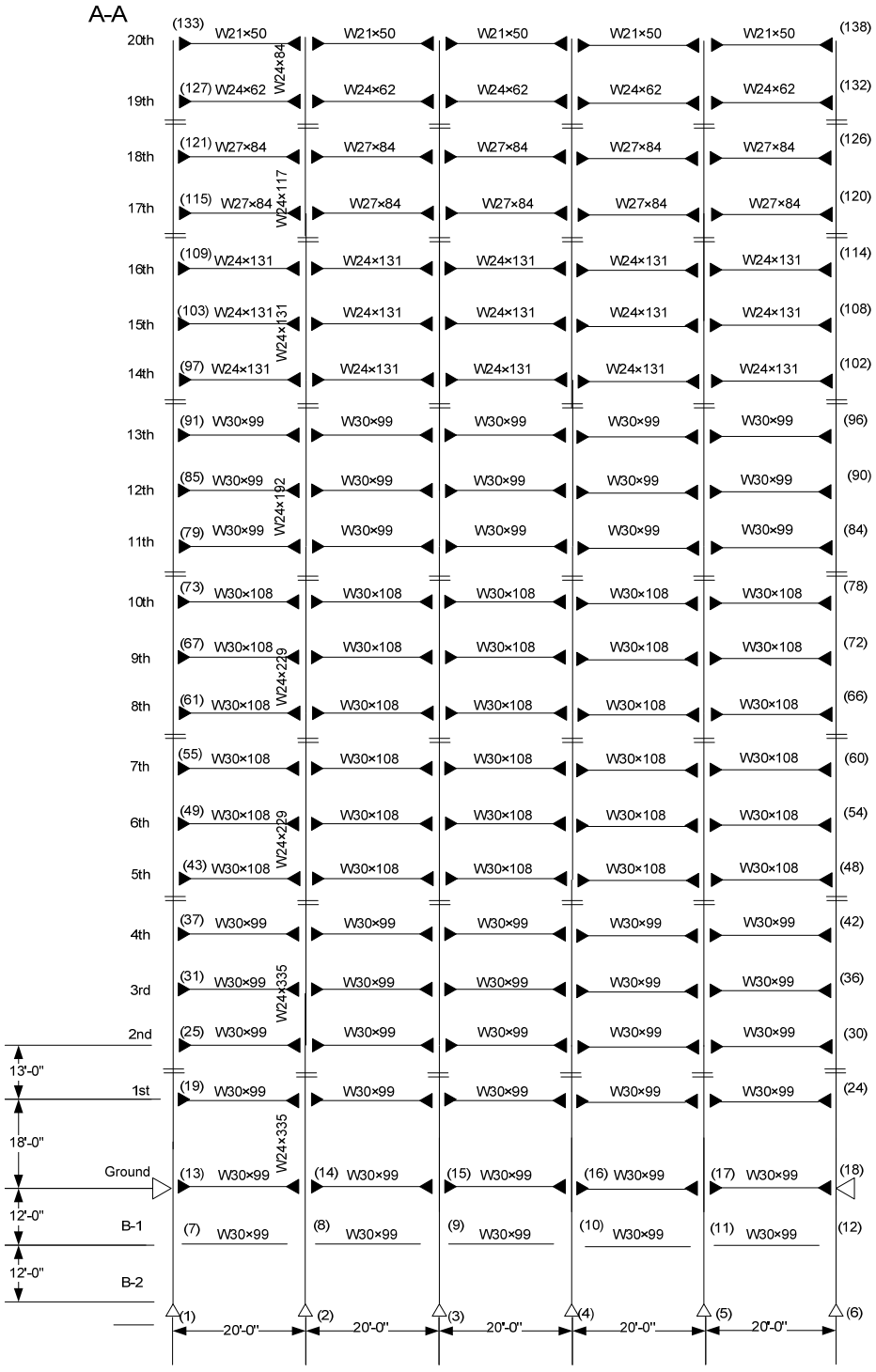

Figure 3.7. 20-story building system

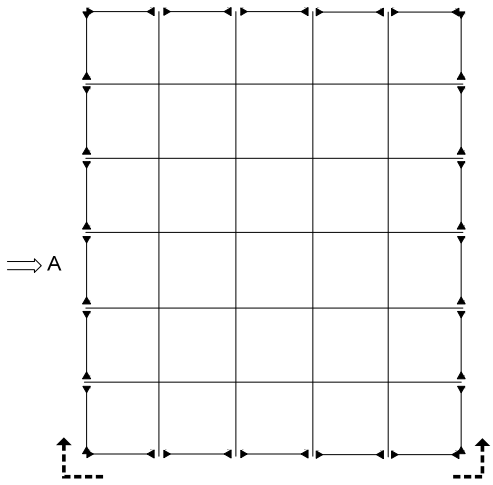

Figure 3.8. 20-story building plan

The pinned connections are used at the base of the structure, and the splices are used to connect the columns at every third story, indicated by the parallel horizontal lines across the columns. The floor system is composed of wide flange beams with the floor slab. Moreover floor system provides diaphragm action which is assumed to be rigid in the horizontal plane (Barroso 1999).The fundamental procedure to model this 20-story building system is nearly same with 3-story building, but the 2 basement stories of the 20-story building are neglected. Although the basements are neglected in modeling 20-story structural system, the overall seismic behavior of structural model is not significantly different when the basement is considered because it is constrained in horizontal direction (Barroso 1999).

The damping matrix is defined based on the *Rayleigh Damping* with reduced system and the assumption of modal damping. Damping in the first mode of vibration 2% with a maximum of 10% critical damping in any one mode and then the final damping matrix is given by (Spencer et al. 1999):

$$\mathbf{C}_{\mathbf{2D20s}} = \mathbf{M}\boldsymbol{\Phi}\begin{bmatrix} 2\zeta_1\omega_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & 2\zeta_n\omega_n \end{bmatrix}\boldsymbol{\Phi}^{-1} \tag{3.25}$$

Finally the equation of motion for the 2-D 20-story building is expressed in Equation (3.26):

$$\mathbf{M}_{\mathbf{2D20s}}\ddot{\mathbf{x}} + \mathbf{C}_{\mathbf{2D20s}}\dot{\mathbf{x}} + \mathbf{K}_{\mathbf{2D20s}}\mathbf{x} = -\mathbf{M}_{\mathbf{2D20s}}\mathbf{G}_{\mathbf{2D20s}}\ddot{x}_g + \mathbf{P}_{\mathbf{2D20s}}\mathbf{f}_{\mathbf{2D20s}} \tag{3.26}$$

where $\mathbf{G}_{\mathbf{2D20s}}$ is the vector defining the loading of ground acceleration onto the evaluation model, $\mathbf{P}_{\mathbf{2D20s}}$ is the vector defining the loading of control forces onto the structure, and $\mathbf{f}_{\mathbf{2D20s}}$ is the vector of forces produced by the control devices. The natural frequencies of the 20-story structural system model are compared with others who have modeled this structure (Ohtori 2004):

Table 3.2. Comparison of first 10 frequencies

Unit(Hz)	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th
Current Model	0.2911	0.8129	1.3414	1.8747	2.4494	3.0642	3.6955	4.4077	5.1384	5.8848
Barroso (1999)	0.261	0.753	1.30	1.83	2.40	2.44	2.92	3.01	3.63	3.68

Three basic analytical models were developed by Gupta and Krawinkler (1999) with different modeling assumptions, but the first frequencies ranges are between 0.2899 and 0.2512. From the Table 3.2 and Figures 3.9 and 3.10, current suggested 20-story benchmark model is compared with other suggested data of the same structural model.

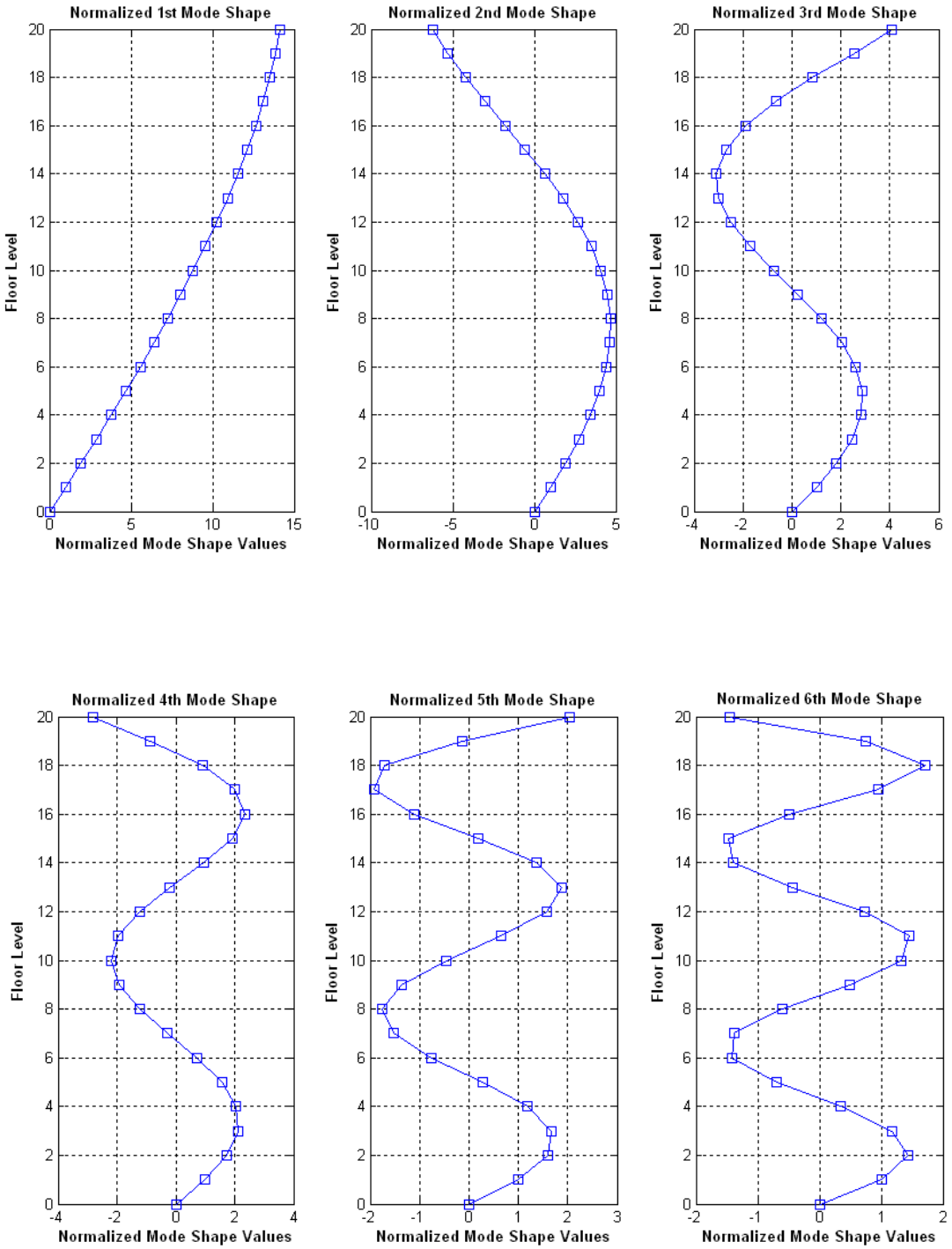

Figure 3.9. 20-story building plan mode shapes

Although the frequencies after 6-th are a little bit different, the general frequencies and the transfer functions from the ground acceleration to absolute roof acceleration, velocity and displacement and the suggested mode shapes shown in Figure 3.9 are nearly same with other researcher' results (Ohtori 2004). Thus this model is appropriate for use in comparative studies with other researchers investigating the benchmark cases.

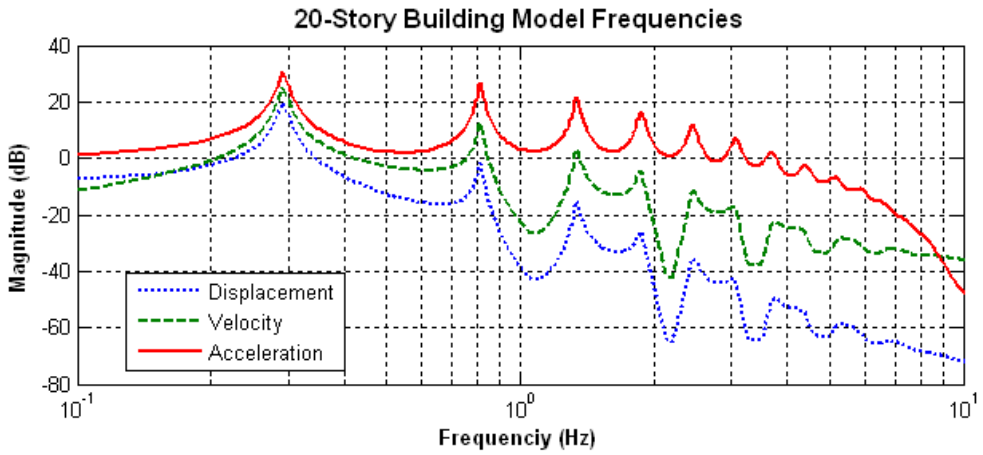

Figure 3.10. 20-story building model transfer functions from ground acceleration to absolute roof acceleration, velocity, and displacement

3.3 Description of 3-D 20-Story Benchmark Control System

For better evaluation of the structural and environmental real conditions of the 20-story benchmark system, a 3-D model is developed for the active control approach. The 3-D system with 60 DOFs system for a 20-story building is developed by using the Ritz method and static condensation to reduce the zero mass DOFs in vertical direction. The whole procedure is same as with a 2-D system, but in this case the transversal and

rotational inertia are not ignored. The first procedures are the determination of mass, stiffness, and damping matrix by using the FEM. The mass and stiffness matrices are defined with half seismic mass, and then static reduction method is used to eradicate the zero mass DOFs. Finally the damping matrix is defined by one of the proportional Rayleigh damping. By adopting the two-way unsymmetric plan (Figure 3.11) of Chopra (2000), the stiffness matrix $\mathbf{k}_{\mathbf{story}}$ of each story is defined for the 3-D system as (Chopra, 2000):

$$\mathbf{k}_{\mathbf{story}} = \begin{bmatrix} k_{xB}+k_{xC} & 0 & (d/2)(k_{xC}-k_{xB}) \\ 0 & k_y & ek_y \\ (d/2)(k_{xC}-k_{xB}) & ek_y & e^2k_y+(d^2/4)(k_{xB}+k_{xC}) \end{bmatrix} \tag{3.27}$$

where k_{xB}, k_y, and k_{xC} are each the lateral stiffnesses of the frame B, A, and C, and d, and e are each the distances of the each frames from the center of axis as shown in Figure 3.11.

Finally, the $\mathbf{M}_{\mathbf{3D20s}}$ and $\mathbf{K}_{\mathbf{3D20s}}$ matrices are composed of 60 DOFs and the damping matrix $\mathbf{C}_{\mathbf{3D20s}}$ (60×60) also can be defined by using the *Rayleigh damping*:

$$\mathbf{M}_{\mathbf{3D20s}}\ddot{\mathbf{x}}+\mathbf{C}_{\mathbf{3D20s}}\dot{\mathbf{x}}+\mathbf{K}_{\mathbf{3D20s}}\mathbf{x}=-\mathbf{M}_{\mathbf{3D20s}}\mathbf{G}_{\mathbf{3D20s}}\ddot{x}_g+\mathbf{P}_{\mathbf{3D20s}}\mathbf{f}_{\mathbf{3D20s}} \tag{28}$$

where $\mathbf{G}_{\mathbf{3D20s}}$ is ground motion matrix, $\mathbf{P}_{\mathbf{3D20s}}$ is a vector defining how the force(s) produced by the control device(s) enter the structure, and $\mathbf{f}_{\mathbf{3D20s}}$ is the control force input.

The typical transfer functions of the ground excitation to roof acceleration, velocity, and displacement of the 3-D 20-story building of the longitudinal, transverse and rotational directions are shown in Figures from 3.12 to 3.14. These values are

compared to upper 2-D 20-story building system model and the mode shapes shown in Table 3.3.

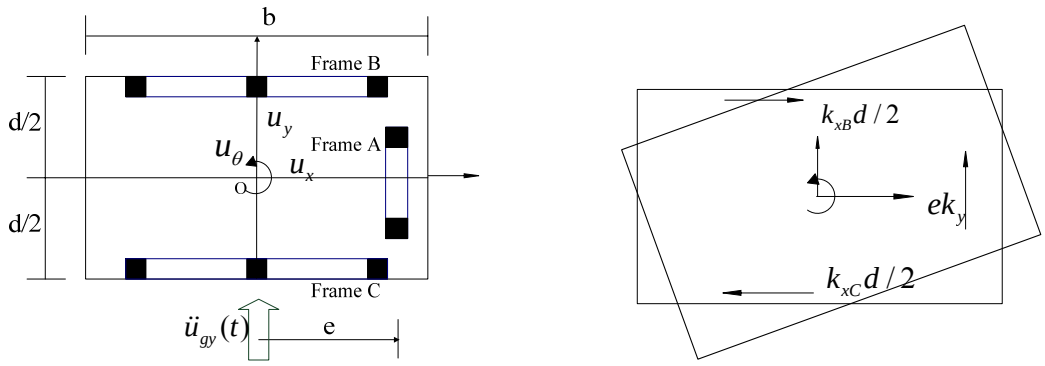

Figure 3.11. Stiffness of the each story of two-way unsymmetric system (Chopra 2000)

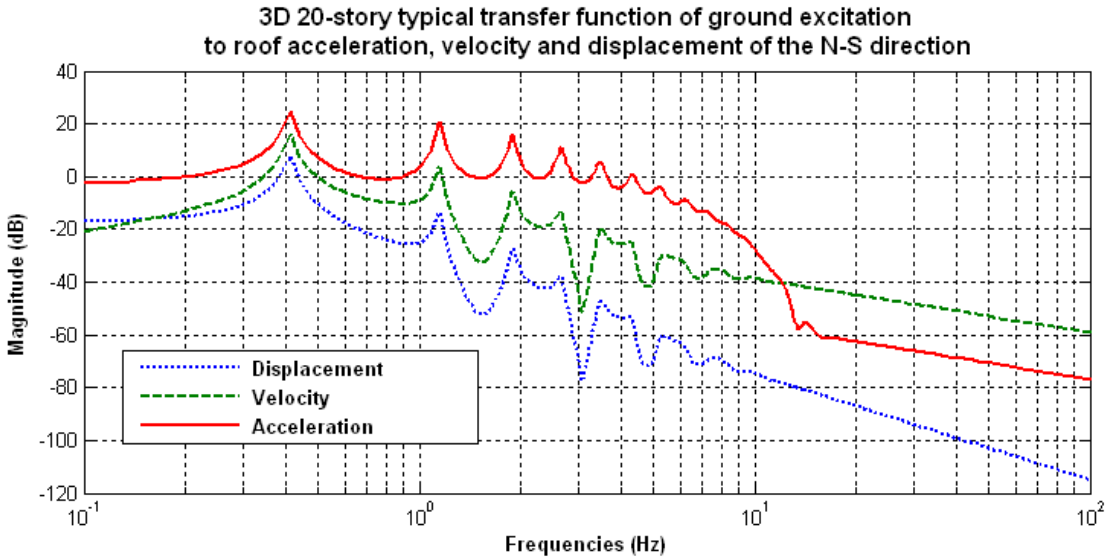

Figure 3.12. 3-D 20-story building model typical transfer functions from ground acceleration to absolute roof acceleration, velocity, and displacement of the N-S direction.

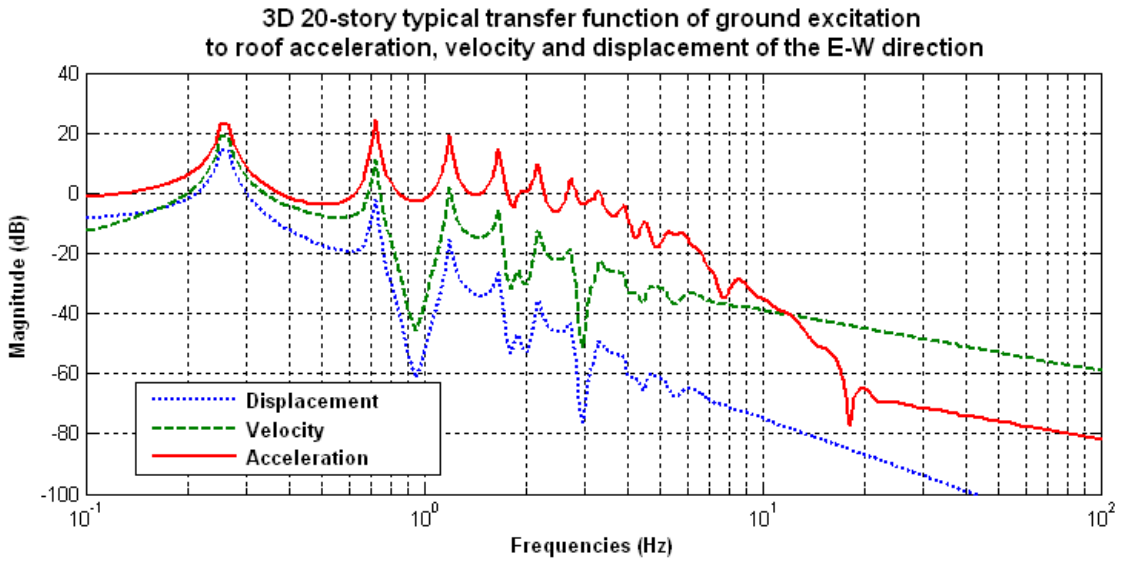

Figure 3.13. 3-D 20-story building model typical transfer functions from ground acceleration to absolute roof acceleration, velocity, and displacement of the E-W direction.

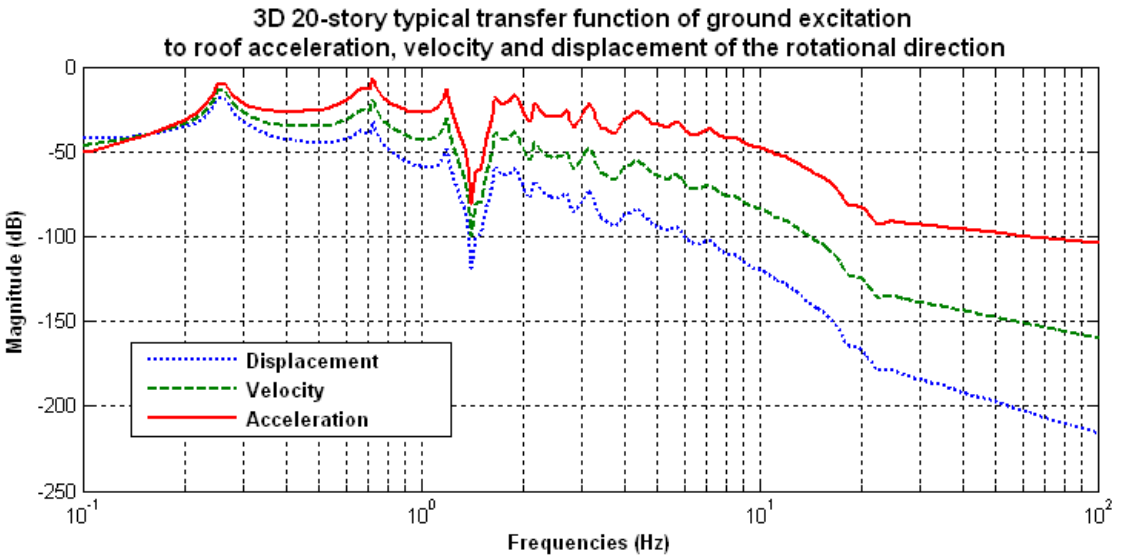

Figure 3.14. 3-D 20-story building model typical transfer functions from ground acceleration to absolute roof acceleration, velocity, and displacement of the rotational direction.

The 3-D 20-story model has reasonable responses to the ground excitation in the transfer function and mode shapes that are comparable to the 2-D 20-story model. Thus these 2-D 3-, 20-story building and 3-D 20-story building models are used to the active control system model.

Table 3.3. First 3 mode shapes of 3-D 20-story benchmark building model

	First Mode Shape			Second Mode Shape			Third Mode Shape		
	N-S	E-W	Rot. (rad)	N-S	E-W	Rot. (rad)	N-S	E-W	Rot. (rad)
1st	0	-0.0234	0.0005	-0.0235	0	0	0	-0.0216	-0.0053
2nd	0	-0.0452	0.0010	-0.0452	0	0	0	-0.0422	-0.0102
3rd	0	-0.0670	0.0015	-0.0671	0	0	0	-0.0630	-0.0151
4th	0	-0.0885	0.0020	-0.0886	0	0	0	-0.0835	-0.0200
5th	0	-0.1097	0.0025	-0.1098	0	0	0	-0.1038	-0.0247
6th	0	-0.1308	0.0030	-0.1309	0	0	0	-0.1241	-0.0295
7th	0	-0.1510	0.0035	-0.1511	0	0	0	-0.1437	-0.0340
8th	0	-0.1703	0.0039	-0.1704	0	0	0	-0.1625	-0.0384
9th	0	-0.1887	0.0043	-0.1888	0	0	0	-0.1805	-0.0425
10th	0	-0.2061	0.0047	-0.2062	0	0	0	-0.1978	-0.0464
11th	0	-0.2234	0.0051	-0.2235	0	0	0	-0.2150	-0.0503
12th	0	-0.2403	0.0055	-0.2404	0	0	0	-0.2321	-0.0540
13th	0	-0.2558	0.0059	-0.2558	0	0	0	-0.2479	-0.0575
14th	0	-0.2705	0.0062	-0.2706	0	0	0	-0.2632	-0.0608
15th	0	-0.2845	0.0065	-0.2846	0	0	0	-0.2777	-0.0639
16th	0	-0.2965	0.0068	-0.2965	0	0	0	-0.2903	-0.0666
17th	0	-0.3074	0.0071	-0.3074	0	0	0	-0.3019	-0.0690
18th	0	-0.3168	0.0073	-0.3169	0	0	0	-0.3121	-0.0711
19th	0	-0.3251	0.0075	-0.3251	0	0	0	-0.3210	-0.0729
20th	0	-0.3314	0.0076	-0.3314	0	0	0	-0.3279	-0.0744

4 DESCRIPTION OF MODELING OF LQG CONTROLLER AND CONTROL DEVICE

Yao (1972) first attempted to employ the concept of active control systems in civil structures to reduce the response to dynamic loads. After that, many active control concept investigations were conducted into protecting structures and human properties. However, most of the previous research utilized the assumption of full state feedback, which means all the states of structure are directly measurable. However, this full state feedback situation is possible only in special cases or very simple structural systems that are not typical of civil structures. Most of these researches into active controllers were based on measurement the displacement or velocity, but they are generally hard to measure and are very costly in order to get the precise information into structural responses. Considering practical limitations of the acquisition of the structural response, the acceleration can be good alternative measurement of structural response, and accelerometers are inexpensive and provide accurate and reliable measurements, moreover acceleration measurement are not dependent on the inertial reference for the structural system. So for application to realistic structural systems and to calculate the active control force, the acceleration response of structural system is best choice for output responses calculations.

Linear Quadratic Gaussian (LQG) compensator utilizes the acceleration response of structural system. As such, LQG controller is an output feedback controller as it is based on partial measurement of the system states or the linear combinations of the

states. In this section, the LQG control concept is applied as an active structural controller. Thus, the basic concepts of active control system, active hydraulic actuator and LQG compensator will be reviewed and treated in this section.

4.1 Basic Control Concepts

The basic concept of the control is that in order to produce the desired response of a plant or structural system, active devices regulate the plant by adding force. There are two basic control approaches: *Open-loop* and *closed-loop control systems*. An open-loop control system determines the control inputs that are applied to the system or structure without any information about the structure outputs. For example, when one drives a vehicle, one push the acceleration pedal to increase the velocity of the vehicle, but if the actual increase in velocity is not known, the driver does role as the open-loop controller for the open-loop control system. In this open-loop control system, full information, or an exact prediction about the transfer function of the system which is the relationship between control input and output, is required to achieve the desired response of the plant. On the other hand, if the driver can see the speedometer of his car, then the driver can adjust the acceleration pedal (control input) according to the actual measured speed of car (control output), the driver does role as the feedback controller and the control system is a closed-loop system (Tewari 2002).

The relationship between the control outputs in a linear system and control inputs are defined as the *transfer function* for the system. As shown in Figure 4.1, the closed-loop controller constantly receives the measured output data, so it can achieve the

desired output even in the presence of noise or uncertainties such as sensor noise, environmental noise (wind), car model uncertainty, speedometer uncertainty and its noise. This is a feedback controller where the controller can get the actual information of system outputs (Tewari 2002).

The difference between the desired output and the actual output is *error*, and its absolute value is *overshoot*. In the case, where the desired output is a constant value, the controller is called a *regulator*. Sometimes, the error can be the desired output to be controlled, and controllers try to make this error zero in general. On the other hand, when the desired output response varies with time, the control system is called a *tracking system.* Thus, if the closed-loop system controller can minimize the error as soon as possible, the controller is a good controller (Tewari 2002).

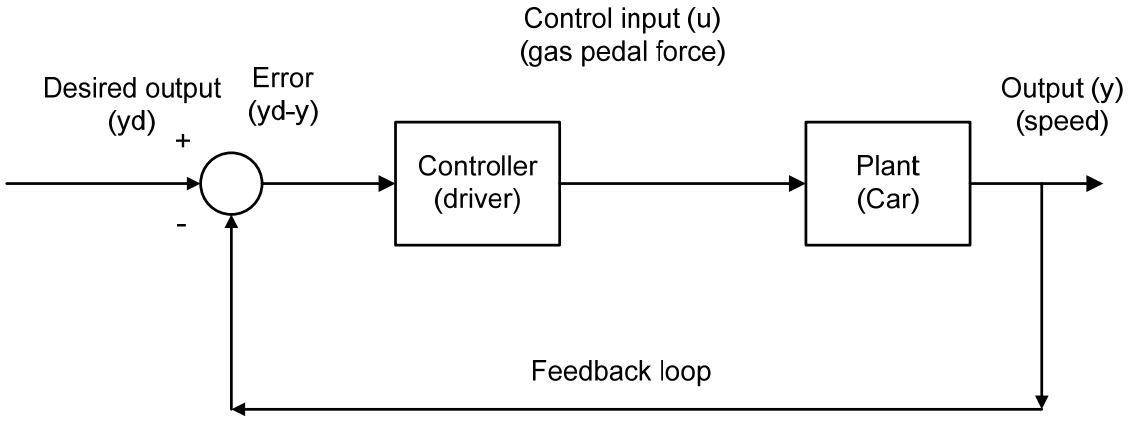

Figure 4.1. Feedback control System (Tewari 2002)

There is another way to classify the control system by using the *state* concept. The meaning of the state is any set of physical quantities specified at a given time in

order to determine the behavior of the system. The state can be the plant's displacement, and velocity. Based on knowledge of the state, a system can be characterized as *deterministic* or *stochastic (probabilistic).* A deterministic system is one where if the states of the system and inputs are specified at some time, then the states of the system can be exactly predicted at a later time. In contrast, although one fully knows the state of a system at specific time, such as its initial condition, if one can not determine the state of the system at a later time, the system is stochastic system. So the stochastic system's state can be estimated by the stochastic governing laws and initial conditions (Tewari 2002).

To design a control system and controller, there are several things that control designers have to consider and check during modeling the control system. For example, stability, robustness, observability, and controllability are important concepts when one designs the closed-loop feedback control system. Moreover, to find the optimal placement and number of actuator and sensor using genetic algorithms (GAs), inevitably GAs need to try to analyze response control using random positions of dampers and sensors, not an engineering perspective of the position of actuators and sensors. In this process, many cases related to the stability or observability or controllability problem need to be checked and verified. Thus, a clear understanding of the exact concept of these theories is required. For the better understanding of the LQG control system, these basic concepts are reviewed in this section.

4.1.1 State-space Model

A single input and single output problem can be easily analyzed by utilizing the transfer functions of the system and/or controller to evaluate its performance. In contrast, a multivariable system with multivariable inputs is not easily analyzed in this fashion. Thus, the control designer needs a better mathematical procedure for handling arbitrary input responses of a multivariable system. The main idea of the state space model comes from the describing a dynamic system with first-order differential equations in the vector-valued state of the system. This state-space form allows the control designer to get the calculation of the responses of multivariable system with an arbitrary input. The basic form of the state-space model is:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \tag{4.1}$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \tag{4.2}$$

where $\dot{\mathbf{x}}$ is state vector, $\mathbf{y}$ is output vector, and $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, and $\mathbf{D}$ with controller canonical forms are system, input, output and matrix regulating matrices respectively.

The advantages of the state-space method are (Tewari 2002; Franklin et al. 2002):

- Time responses of a multivariable system are obtained directly by solving a set of first-order differential equations.
- Develop more general method by studying the equation itself.
- Connect internal and external descriptions. For example, internal descriptions such as displacement, velocity or acceleration are computed

from the state variables, and these variables can be connected to the sensored output values.

4.1.2 Stability

Stability is the ability of a system to approach one of its equilibrium points once displaced from it (Tewari 2002). As shown in Figure 4.1, let's assume the transfer function of a plant and a controller are $G(s)$ and $H(s)$ respectively, and the feedback control system is shown in Figure 4.2. The gain of this system is:

$$Transfer\ function = \frac{G(s)H(s)}{1+G(s)H(s)} \tag{4.3}$$

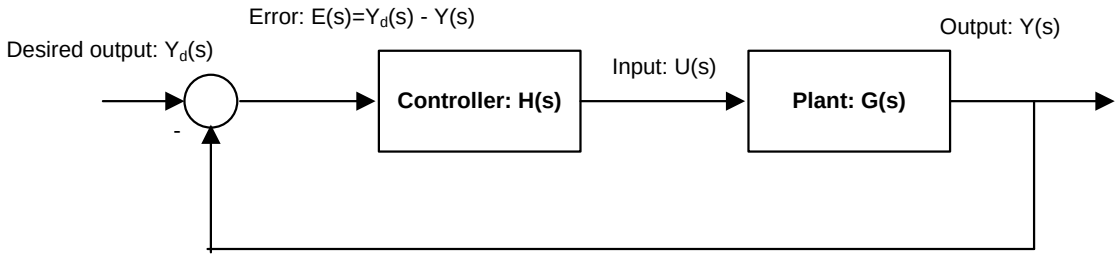

Figure 4.2. Basic feedback control system (Tewari 2002)

To check the stability, the poles of the denominator in gain in Equation (4.3) are determined. If the real parts of all the roots of the denominator of the transfer function are negative values, the initial response to initial conditions are inclined to a steady-state value in the limit time. Such linear control systems are called *asymptotically stable.* In case of any root of denominator of transfer function has one or more positive real value,

and then the initial response will be infinite in magnitude in the limited time, the system is called *unstable* system. And in case of all the roots of the denominator of a system have real parts less than or equal to zero, and then if the initial response of the control system will sustain oscillating motion with a finite amplitude in the limited time, the control system is called *stable but not asymptotically stable.* However, if the system has multiple roots of denominator with same imaginary part or any one repeated roots of denominator has zero real part, the system is called *unstable* (Tewari 2002).

4.1.3 Robustness

The final goal of the control system is to design a system that can adapt to the real environment and perform its own duties. However, there are many kinds of uncertainties, assumptions and disturbances when one develops a control system. A main source of uncertainty arises from the model. Uncertainties arise from assumptions regarding the natural environment and system modeling. Natural environment assumptions include characteristics such as temperature, wind velocity, humidity, etc.

The disturbances, having an adverse effect on the system response, are the part of the system that can not be mathematically modeled when discussing a control system. For example, the road roughness, tire condition, and wind velocity, etc (Tewari 2002). The robustness is that although there are many uncertainties, assumptions, and disturbances in control system, if the control results are reasonable and get the ultimate goal of the control system, the control system can be called as robust control system.

There are three different kinds of robustness: *robust stability*, *robust performance*, and *robust control*. For example, if the roots of the denominator of a system are tolerant to the change of operational and environmental factors from the nominal case; namely have some ranges of stability, and then the system can sustain the stability, the system is called as robust stability system. And if the performance of system satisfies the requirement of the control designer's standard with the range of poles (roots of the denominator), the system is call as robust performance system. And then if the designed controller satisfies robust stability and performance requirements, the control system is called robust control (Franklin et al. 2002; Stefani et al. 2002).

4.1.4 Controllability

As we already reviewed about the state-space model, and stability, the control system is related to complex differential equations because control problems sometimes need to solve the state-equations with time-varying systems. Thus, Laplace Transform of the transfer function is used to express the transfer function of controller and plant. The controllability is related to the controller transfer function and plant system's stability. For example, let's assume the transfer function for the plant, $T(s)$, and the controller transfer function, $R(s)$ (Tewari 2002):

$$T(s) = (s + 2.5)/[(s-3)(s+1)(s+3)] \tag{4.4}$$

$$R(s) = K(s-3)/s \tag{4.5}$$

where K is control design parameter. This plant system is unstable because it has the positive pole, $s = 3$, but by applying a controller the resulting system transfer function can be stable. The closed-loop control transfer function is same as Equation (4.3) (Tewari 2002):

$$\frac{T(s)R(s)}{1+T(s)R(s)} = \frac{K(s+2.5)}{[s(s+1)(s+3)+K(s+2.5)]} \tag{4.6}$$

From the Equation (4.6), this closed-loop control system's stability is definitely dependent on the controller design parameter value, K. If we choose positive values for the K, the control system is stable and it is controllable system, but when we choose negative values for the K, then the system can be unstable. Many cases are possible to stabilize by adding adequate controller transfer function, $R(s)$ to system, but there are also many cases impossible to stabilize with a controller transfer function. At this time, the designer has to determine what system is possible or impossible to stabilize. Equation (4.7) express the criteria equation used to determine what system is possible to stabilize and therefore controllable. When the C matrix has rank n, then the rank is same as the order of the system, the system is controllable system. However, when a zero of the controller cancels a pole of the plant, it may also lead to an uncontrollable closed-loop system, even though the plant itself may be controllable (Tewari 2002):

$$C = \begin{bmatrix} \mathbf{B} \\ \mathbf{AB} \\ (\mathbf{A})^2\mathbf{B} \\ \vdots \\ (\mathbf{A})^{n-1}\mathbf{B} \end{bmatrix} \tag{4.7}$$

where $\mathbf{A}$, and $\mathbf{B}$ are the state-space system and input matrices respectively.

4.1.5 Observability

The state-space approach using full-state feedback provides sufficient number of controller design parameters to move all the closed-loop poles independently of each other. This full state-feedback system assumes that all the state variables of the plant are able to measurable using sensors. Thus adequate controller gains and controller transfer function can move system's poles to make a controllable system. However, the states of the real plant are not always measurable. Some state variables are hard to measure because there are no accurate sensors or ummeasurable due to its physical uncertainties. Thus, if it is impossible to measure the state variable, one needs to estimate the state variables. To estimate the required state variables to control the system, we need to observe the measurable output state variable and then reestablish the state variable information. This state estimator is called an observer. If the observer estimates all state variables, the observer is full-order observer. However the directly measurable states need not to be estimated, so if only unmeasurable states are estimated by the observer, the observer is called reduced order observer. The controllers that generate the control inputs to the plant based on the estimated state-vector are called compensators (Franklin et al. 2002; Tewari 2002).

When defines the observability of system, the most important thing to consider is whether all the state variables contribute to the output of the system. The relationship between the output and state variables also affect the system observability. Thus, the matrix defining the observability consists of the system matrix and output matrix. When

the order of system and the rank of the Equation (4.8) are same, the system is observable (Stefani et al. 2002; Tewari 2002):

$$O = \begin{bmatrix} \mathbf{C}^T \\ \mathbf{A}^T\mathbf{C}^T \\ (\mathbf{A}^T)^2\mathbf{C}^T \\ \vdots \\ (\mathbf{A}^T)^{n-1}\mathbf{C}^T \end{bmatrix} \tag{4.8}$$

where $\mathbf{A}$, and $\mathbf{C}$ are the state-space system and output matrices respectively.

4.2 Hydraulic Actuator Model

The governing equation for the i-th hydraulic actuator with position feedback is (Spencer et al. 1999):

$$\dot{f}^i = \frac{2\beta}{V}(Ak_q\gamma(u_i - y^a{}_i) - k_c f^i - A^2\dot{y}_i^a) \tag{4.9}$$

$$\dot{f}^i = a_1 u_i - a_1 y^a{}_i - a_2\dot{y}_i^a - a_3 f^i \tag{4.10}$$

where $a_1 = 5.8128784\times10^6$ kN/m-sec, $a_2 = 5.4641931\times10^4$ kN/m, and $a_3 = 1.6210740\times10^3$ sec^{-1}. A is cross-sectional area of the actuator, β is the bulk modulus of the fluid, V is the characteristic hydraulic fluid volume of the actuator, u_i is the command signal to the ith actuator, f^i is the force generated by the ith actuator, y_i^a is the displacement of the ith actuator, k_c and k_q are the system constants, and γ is the proportional feedback gain stabilizing the actuator.

The hydraulic actuators have a 1000 KN capacity and ± 8.9cm stroke. The displacement of each control device is the same as the interstory drift of the each level and the actuators are installed horizontally with the floor slabs. The governing equation for the displacements of the control devices and the relative displacements of the floors are (Spencer et al. 1999):

$$y_i^{dev}(t) = d_i(t) = x_{\eta i}(t) - x_{\eta i-1}(t) \tag{4.11}$$

where

$$\mathbf{y}^{\mathbf{dev}} = \mathbf{\Delta x}_{\boldsymbol{\eta}} \tag{4.12}$$

$$\mathbf{\Delta} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ -1 & 1 & 0 & \dots & 0 \\ 0 & & \dots & & \vdots \\ 0 & \dots & -1 & 1 & 0 \\ 0 & \dots & 0 & -1 & 1 \end{bmatrix} \tag{4.13}$$

$\mathbf{\Delta}$ is the transformation matrix from the floor displacements to the control device displacements. Equations (4.14) and (4.15) are the state space equations for the dynamics of the hydraulic actuators for 20-story building (Spencer et al. 1999).

$$\dot{\mathbf{x}}^{\mathbf{dev}} = \mathbf{A}_{\mathbf{dev}}\mathbf{x}^{\mathbf{dev}} + \mathbf{B}_{\mathbf{dev}}\begin{bmatrix} \mathbf{u} \\ \mathbf{y}_{\mathbf{c}} \end{bmatrix} \tag{4.14}$$

$$\mathbf{f} = \mathbf{C}_{\mathbf{dev}}\mathbf{x}^{\mathbf{dev}} + \mathbf{D}_{\mathbf{dev}}\begin{bmatrix} \mathbf{u} \\ \mathbf{y}_{\mathbf{c}} \end{bmatrix} \tag{4.15}$$

$$\mathbf{A}_{\mathbf{dev}} = \left[-a_3\mathbf{I}\right], \mathbf{B}_{\mathbf{dev}} = \left[a_1\mathbf{I} \quad -a_1\mathbf{\Delta} \quad -a_2\mathbf{\Delta}\right] \tag{4.16}$$

In Equation (4.15), $\mathbf{C}_{\mathbf{dev}}$ is a diagonal $\left[ndof \times ndof\right]$ matrix, and $\mathbf{D}_{\mathbf{dev}}$ is a zero $\left[ndof \times 3ndof\right]$ matrix. Where $ndof$ =number of degrees of freedom.

4.3 Linear Quadratic Regulator (LQR) Model

To date, many civil engineering researchers adopted active control approaches to civil structures to reduce structural damage from strong earthquakes or winds. As a traditional control concept, the designing control systems by pole-placement need a lot of trial to find the optimal control design values. One of the well-known active control algorithms is Linear Quadratic Gaussian (LQG) compensator. This optimal control algorithm is characterized by working to minimize the cost associated with generating control inputs and to maximize the control effect.

This LQG optimal compensator combines optimal regulator and optimal observer (Kalman filer). From the combined system of LQR and Kalman filter (optimal estimator), LQG overcomes the LQR assumption that all the feedback variables are able to be measured. As previously discussed, active control is a very powerful approach control the behavior of civil structures under strong winds or earthquakes. With these efficiencies, benefits, and proven performance (Dyke 1996) in civil structures, the LQG compensator is selected for implementation in this research. In this section, the actuator is modeled, and then the LQR regulator and optimal Kalman filter are modeled to establish the LQG controller and these steps are approximately followed by the suggested ASCE control benchmark building model case (Ohtori 2004). Although ASCE structural control benchmark program offered a Matlab R2007b code from the website: http://sstl.cee.uiuc.edu/, it was necessary to modify that LQG model and then generate a 3-D 80 degree-of-freedom (DOF) system with per floor level one DOF.

The optimal compensator design process is as follows. The first step is to design an optimal regulator for a linear plant assuming full-state feedback and a quadratic objective function. The regulator is designed to generate a control input **u(t)**, based on the measured state-vector, **x(t)**. The next step is to design a Kalman filter for the plant by assuming a known control input, **u(t)**, a measured output, **y(t)**, and white noise, **v(t)** with known power spectral densities. The Kalman filter is designed to provide an optimal estimate of the state-vector. The final step is to combine the separately-designed optimal regulator and Kalman filter into an optimal compensator. This generates the input vector based on the estimated state-vector, rather than the actual state-vector and measured output vector.

As presented in Section 3, the equations of motion (EoM) for the 2-D 3- and 20-story buildings and 3-D 20-story building were developed using finite element formations. In turn, the EoM was converted to a state-space formulation. The resulting stat-space expressions for the closed-loop system are (Spencer et al. 1999):

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{f} + \mathbf{E}\ddot{x}_g \tag{4.17}$$

$$\mathbf{y_m} = \mathbf{C_m}\mathbf{x} + \mathbf{D_m}\mathbf{f} + \mathbf{F_m}\ddot{x}_g + \mathbf{v} \tag{4.18}$$

$$\mathbf{y_r} = \mathbf{C_r}\mathbf{x} + \mathbf{D_r}\mathbf{u} + \mathbf{F_r}\ddot{x}_g \tag{4.19}$$

$$\mathbf{y_c} = \mathbf{C_c}\mathbf{x} + \mathbf{D_c}\mathbf{f} + \mathbf{F_c}\ddot{x}_g \tag{4.20}$$

where $\mathbf{x}$ is the state vector, $\mathbf{y_m}$ is the vector corresponding to the measured output, $\mathbf{y_r}$ is the vector of regulated responses, $\mathbf{y_c}$ is the vector of output responses that

are used as inputs to control device models, $\mathbf{f}$ is control force input, and $\mathbf{v}$ is a measurement noise vector where:

$$\mathbf{A}=\begin{bmatrix}\mathbf{0}_{ndof\times ndof} & \mathbf{I}_{ndof\times ndof}\\ -\mathbf{M}^{-1}\mathbf{K}_{ndof\times ndof} & -\mathbf{M}^{-1}\mathbf{C}_{ndof\times ndof}\end{bmatrix},\ \mathbf{B}=\begin{bmatrix}\mathbf{0}_{ndof\times ndof}\\ \mathbf{M}^{-1}\mathbf{f}_{\mathbf{a}}\boldsymbol{\Delta}_{ndof\times ndof}\end{bmatrix},\ \mathbf{E}=\begin{bmatrix}\mathbf{0}_{ndof\times 1}\\ -\mathbf{G}_{ndof\times 1}\end{bmatrix},$$

$$\mathbf{C_m}=\begin{bmatrix}\mathbf{I}_{ndof\times ndof} & \mathbf{0}_{ndof\times ndof}\\ \mathbf{0}_{3\times 3} & \mathbf{I}_{ndof\times ndof}\\ -\mathbf{M}^{-1}\mathbf{K}_{ndof\times ndof} & -\mathbf{M}^{-1}\mathbf{C}_{ndof\times ndof}\end{bmatrix},\ \mathbf{D_m}=\begin{bmatrix}\mathbf{0}_{ndof\times ndof}\\ \mathbf{0}_{ndof\times ndof}\\ \mathbf{f}_{\mathbf{a}}(\mathbf{M}^{-1}\boldsymbol{\Delta}_{ndof\times ndof})\end{bmatrix},\ \mathbf{F_m}=\begin{bmatrix}\mathbf{0}_{ndof\times 1}\\ \mathbf{0}_{ndof\times 1}\\ -\mathbf{ones}_{ndof\times 1}\end{bmatrix}.$$

where the *ndof* is 3 for the 2-D 3-story, the 20 for the 2-D 20-story, and 60 for the 3-D 20-story case, $\mathbf{G}$ is vector defining the loading of ground acceleration onto the evaluation model, and $\mathbf{f_a}$ is the control device force $1000\,KN$.

To combine the actuator and structural model, after several matrix calculations, the governing equations for the evaluation model are redefined as shown in below Equations (4.21) and (4.22) :

$$\dot{\mathbf{x}}=\mathbf{A_s}\mathbf{x}+\mathbf{B_s}\begin{bmatrix}\ddot{\mathbf{x}}_{\mathbf{g}}\\ \mathbf{f}\end{bmatrix} \tag{4.21}$$

$$\mathbf{y}=\mathbf{C_s}\mathbf{x}+\mathbf{D_s}\begin{bmatrix}\ddot{\mathbf{x}}_{\mathbf{g}}\\ \mathbf{f}\end{bmatrix}+\mathbf{v} \tag{4.22}$$

where

$$\mathbf{A_s}=\left[\begin{array}{c|c}\mathbf{0}_{ndof\times ndof} & \mathbf{I}_{ndof\times ndof}\\ \hline -\mathbf{M}^{-1}\mathbf{K}_{ndof\times ndof} & -\mathbf{M}^{-1}\mathbf{C}_{ndof\times ndof}\end{array}\right],\ \mathbf{B_s}=\left[\begin{array}{c|c}\mathbf{0}_{ndof\times 1} & \mathbf{0}_{ndof\times ndof}\\ \hline -\mathbf{G}_{ndof\times 1} & \mathbf{M}^{-1}\mathbf{f}_{\mathbf{a}}\boldsymbol{\Delta}_{ndof\times ndof}\end{array}\right],$$

$$\mathbf{C_s}=\left[\begin{array}{c|c}\mathbf{I}_{ndof\times ndof} & \mathbf{0}_{ndof\times ndof}\\ \hline \mathbf{0}_{ndof\times ndof} & \mathbf{I}_{ndof\times ndof}\\ \hline -\mathbf{M}^{-1}\mathbf{K}_{ndof\times ndof} & -\mathbf{M}^{-1}\mathbf{C}_{ndof\times ndof}\end{array}\right],\ \mathbf{D_s}=\left[\begin{array}{c|c}\mathbf{0}_{ndof\times 1} & \mathbf{0}_{ndof\times ndof}\\ \hline \mathbf{0}_{ndof\times 1} & \mathbf{0}_{ndof\times ndof}\\ \hline -\mathbf{ones}_{ndof\times 1} & \mathbf{f}_{\mathbf{a}}(\mathbf{M}^{-1}\boldsymbol{\Delta}_{ndof\times ndof})\end{array}\right].$$

The design model for the LQG controller is formed by stacking the states of the reduced structural model and the actuator model as shown in follows:

$$\dot{\mathbf{x}}^{\mathbf{d}} = \mathbf{A}_{\mathbf{d}}\mathbf{x}^{\mathbf{d}} + \mathbf{B}_{\mathbf{d}}\begin{bmatrix} \ddot{\mathbf{x}}_{\mathbf{g}} \\ \mathbf{u} \end{bmatrix} \tag{4.23}$$

$$\mathbf{y}_{\mathbf{md}} = \mathbf{C}_{\mathbf{md}}\mathbf{x}^{\mathbf{d}} + \mathbf{D}_{\mathbf{md}}\begin{bmatrix} \ddot{\mathbf{x}}_{\mathbf{g}} \\ \mathbf{u} \end{bmatrix} + \mathbf{v} \tag{4.24}$$

$$\mathbf{y}_{\mathbf{rd}} = \mathbf{C}_{\mathbf{rd}}\mathbf{x}^{\mathbf{d}} + \mathbf{D}_{\mathbf{rd}}\begin{bmatrix} \ddot{\mathbf{x}}_{\mathbf{g}} \\ \mathbf{u} \end{bmatrix} + \mathbf{v} \tag{4.25}$$

where $\mathbf{x}^{\mathbf{d}}$ is the state vector consisted with displacement and acceleration for the design model, $\mathbf{y}_{\mathbf{md}}$ is the vector corresponding to the measured output for the design model, $\mathbf{y}_{\mathbf{rd}}$ is the vector of regulated responses for the design model, and the associated matrices are:

$$\mathbf{A}_{\mathbf{d}} = \begin{bmatrix} \mathbf{A}_{\mathbf{s}} & \mathbf{B}_{\mathbf{s}}(I,II)\times\mathbf{C}_{dev} \\ -a_1(\boldsymbol{\Delta}\times\mathbf{C}_{\mathbf{s}}(I,I) - a_2\times(\boldsymbol{\Delta}\times\mathbf{C}_{\mathbf{s}}(I,II)) & -a_1(\boldsymbol{\Delta}\times\mathbf{D}_{\mathbf{s}}(I,II)) - a_2\times(\boldsymbol{\Delta}\times\mathbf{D}_{\mathbf{s}}(II,II)) + \mathbf{A}_{\mathbf{dev}} \end{bmatrix} -$$

$$\mathbf{B}_{\mathbf{d}} = \begin{bmatrix} \mathbf{B}_{\mathbf{s}}(:,I) & -a_1(\boldsymbol{\Delta}\times\mathbf{D}_{\mathbf{s}}(I,I)) - a_2\times(\boldsymbol{\Delta}\times\mathbf{D}_{\mathbf{s}}(I,II)) & 0_{2ndof\times 1} & a_1(\mathbf{I}_{ndof\times ndof}) \end{bmatrix}_{3ndof\times(ndof+1)}$$

$$\mathbf{C}_{\mathbf{md}} = \begin{bmatrix} \mathbf{C}_{\mathbf{s}} & \mathbf{D}_{\mathbf{s}}(:,II)\times\mathbf{C}_{\mathbf{dev}} \end{bmatrix}_{3ndof\times 3ndof},\ \mathbf{D}_{\mathbf{md}} = \begin{bmatrix} \mathbf{D}_{\mathbf{s}}(:,I) & \mathbf{0}_{3ndof\times ndof} \end{bmatrix}_{3ndof\times(ndof+1)}$$

$$\mathbf{C}_{\mathbf{rd}} = \begin{bmatrix} \mathbf{C}_{\mathbf{s}} & \mathbf{D}_{\mathbf{s}}(:,II)\times\mathbf{C}_{\mathbf{dev}} \end{bmatrix}_{3ndof\times 3ndof},\ \mathbf{D}_{\mathbf{rd}} = \begin{bmatrix} \mathbf{D}_{\mathbf{s}}(:,I) & \mathbf{0}_{3ndof\times ndof} \end{bmatrix}_{3ndof\times(ndof+1)}$$

where the $\mathbf{C}_{\mathbf{md}}$, $\mathbf{D}_{\mathbf{md}}$, $\mathbf{C}_{\mathbf{rd}}$, and $\mathbf{D}_{\mathbf{rd}}$ will be used dependent on the sensor locations and the number of actuator and locations.

The objective of a LQR regulator is to obtain initial condition errors that trend to zeros. With output weighted, the objective is achieved by selecting the control input **u(t)** to minimize a quadratic performance criterion or cost function, J, where:

$$J = \lim_{T\to\infty} \frac{1}{T} \mathrm{E}\left\{ \int_0^T \left[\mathbf{y}_{\mathbf{rd}}^{\mathbf{T}} \mathbf{Q}(\tau) \mathbf{y}_{\mathbf{rd}} + \mathbf{R}(\tau) \mathbf{u}^{\mathbf{2}}(\tau) \right] d\tau \right\} \tag{4.26}$$

where the $\mathbf{Q}$ and $\mathbf{R}$ should be positive semi-definite state weighting matrices and $\mathbf{u} = [\ddot{\mathbf{x}}_{\mathbf{g}} \quad \mathbf{u}]^{\mathbf{T}}$. The relationship of $\mathbf{Q}$ and $\mathbf{R}$ is that when $\mathbf{R}$ is larger than $\mathbf{Q}$, the control force is considered to be more important. Based on the separation principle, the full state feedback gain matrix $\mathbf{K}$ is found, where:

$$\mathbf{u} = -\mathbf{K}\hat{\mathbf{x}}^{\mathbf{d}} \tag{4.27}$$

where $\hat{\mathbf{x}}^{\mathbf{d}}$ is the Kalman Filter estimate of the state vector based on the our model, and the $\mathbf{K}$ is given by:

$$\mathbf{K} = \tilde{\mathbf{R}}^{-1}(\mathbf{B}_{\mathbf{d}}^{\mathbf{T}}\mathbf{P} + \tilde{\mathbf{N}}) \tag{4.28}$$

where $\mathbf{P}$ results in the following algebraic Riccati equation (Tewari 2002):

$$\left[\tilde{\mathbf{A}}\right]^{\mathbf{T}}[\mathbf{P}] + [\mathbf{P}]\left[\tilde{\mathbf{A}}\right] - [\mathbf{P}][\mathbf{B}_{\mathbf{d}}]\left[\tilde{\mathbf{R}}\right]^{-1}[\mathbf{B}_{\mathbf{d}}]^{\mathbf{T}}[\mathbf{P}] + \left[\tilde{\mathbf{Q}}\right] = \mathbf{0} \tag{4.29}$$

where

$$\tilde{\mathbf{S}} = \mathbf{C}_{\mathbf{rd}}^{\mathbf{t}}\mathbf{Q}\mathbf{D}_{\mathbf{rd}},\ \tilde{\mathbf{N}} = \mathbf{C}_{\mathbf{rd}}^{\mathbf{t}}\mathbf{Q}\mathbf{C}_{\mathbf{rd}},\ \tilde{\mathbf{R}} = \mathbf{R} + \mathbf{D}_{\mathbf{rd}}^{\mathbf{T}}\mathbf{Q}\mathbf{D}_{\mathbf{rd}},\ \tilde{\mathbf{A}} = \mathbf{A}_{\mathbf{d}} - \mathbf{B}_{\mathbf{d}}\tilde{\mathbf{R}}^{-1}\tilde{\mathbf{S}}^{\mathbf{T}}, \text{ and}$$

$$\tilde{\mathbf{Q}} = \mathbf{C}_{\mathbf{rd}}^{\mathbf{T}}\mathbf{Q}\mathbf{C}_{\mathbf{rd}} - \tilde{\mathbf{S}}\tilde{\mathbf{R}}^{-1}\tilde{\mathbf{S}}^{\mathbf{T}}$$

The solution of the algebraic Riccati equation is found with MATLAB lqry.m routine (MATLAB 2007).

4.4 Design Kalman Filter

To design optimal Kalman filter, the procedures are followed by the suggested ASCE control benchmark problem (Spencer 1999). The governing equation of Kalman filter optimal estimator is given by (Spencer 1999);

$$\dot{\hat{\mathbf{x}}}^{\mathbf{d}} = \mathbf{A}_{\mathbf{d}}\hat{\mathbf{x}}^{\mathbf{d}} + \mathbf{B}_{\mathbf{d}}\mathbf{u} + \mathbf{L}(\mathbf{y}_{\mathbf{m}} - \mathbf{C}_{\mathbf{md}}\hat{\mathbf{x}}^{\mathbf{d}} - \mathbf{D}_{\mathbf{md}}\mathbf{u}) \tag{4.30}$$

It can be shown that the matrix $\mathbf{L}$ that minimizes J_∞ is given by (Spencer 1999)

$$\mathbf{L} = \left[\underset{\sim}{\mathbf{R}}^{-1}(\gamma\mathbf{D}_{\mathbf{mb}}^{\mathbf{T}} + \mathbf{C}_{\mathbf{MD}}\mathbf{P}_e)\right]^{\mathbf{T}} \tag{4.31}$$

where $[\mathbf{P}_e]$ is the positive definite solution of Ricatti equation as given by (Spencer 1999);

$$[\mathbf{P}_e][\underset{\sim}{\mathbf{A}}] + [\underset{\sim}{\mathbf{A}}]^{\mathbf{T}}[\mathbf{P}_e] - [\mathbf{P}_e][\underset{\sim}{\mathbf{G}}][\mathbf{P}_e] + [\mathbf{Q}_e] = \mathbf{0} \tag{4.32}$$

where γ is the independent Gaussian white noise coefficient and where (Spencer 1999);

$$\underset{\sim}{\mathbf{A}} = \mathbf{A}_{\mathbf{d}}^{\mathbf{T}} - \mathbf{C}_{\mathbf{md}}^{\mathbf{T}}\underset{\sim}{\mathbf{R}}^{-1}(\gamma\mathbf{D}_{\mathbf{md}}^{\mathbf{T}}),\ \underset{\sim}{\mathbf{G}} = \mathbf{C}_{\mathbf{md}}^{\mathbf{T}}\underset{\sim}{\mathbf{R}}^{-1}\mathbf{C}_{\mathbf{md}},$$

$$\mathbf{Q}_{\mathbf{e}} = \gamma(\mathbf{B}_{\mathbf{d}}(:,I))^T - \gamma^2\mathbf{B}_{\mathbf{d}}(:,I)\mathbf{D}_{\mathbf{md}}(:,I)\underset{\sim}{\mathbf{R}}^{-1}\mathbf{D}_{\mathbf{md}}(:,I)\mathbf{B}_{\mathbf{d}}(:,I)\text{, and}$$

$$\underset{\sim}{\mathbf{R}} = \mathbf{I} + \gamma\mathbf{D}_{\mathbf{md}}(:,I)\mathbf{D}_{\mathbf{md}}(:,I)^T$$

The solution of the algebraic Riccati equation is found with MATLAB lqe2.m routine (MATLAB 2007). Finally, the controller state-space matrixes are founded.

$$\mathbf{A}_{\mathbf{c}} = \left[\mathbf{A}_{\mathbf{d}} - \mathbf{B}_{\mathbf{d}}(:,II)\mathbf{K} - \mathbf{L}\mathbf{C}_{\mathbf{md}} + \mathbf{L}\mathbf{D}_{\mathbf{md}}(:,II)\mathbf{K}\right]_{3ndof\times 3ndof} \tag{4.33}$$

$$\mathbf{B}_{\mathbf{c}} = [\mathbf{L}]_{3ndof\times ndof} \tag{4.34}$$

$$\mathbf{C}_{\mathbf{c}} = -[\mathbf{K}]_{ndof\times 3ndof} \tag{4.35}$$

$$\mathbf{D}_c = -[\mathbf{0}]_{ndof \times ndof} \tag{4.36}$$

4.5 LQG Controller Model

Simulink which is a Graphical User's Interface (GUI) software is one of the important sub-tools of the MATLAB. This GUI software works directly with the block-diagram of a control system rather than with equations themselves such as differential equations, or transfer functions to produce a simulation of the system's response to arbitrary inputs and initial conditions (Simulink 2000). For the LQG control analysis, the Simulink is utilized to model whole control system design.

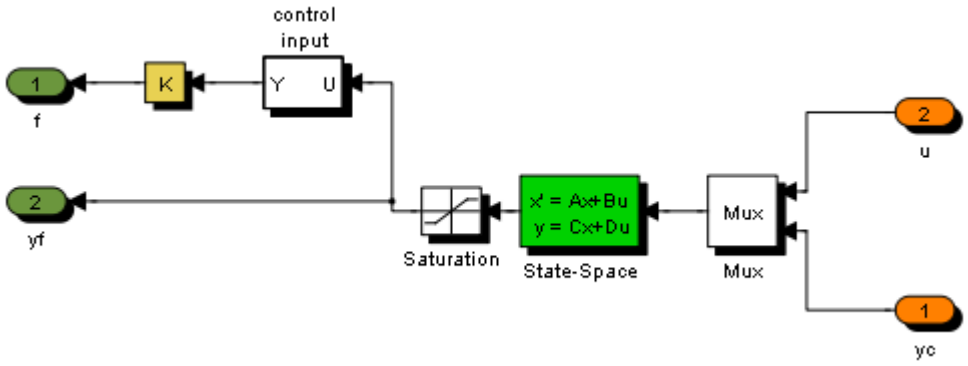

Figure 4.3. Simulink blocks for the hydraulic actuators in LQG control design

The Simulink model for the hydraulic actuators is suggested as shown in Figure 4.3. The matrices for the state-space equation were already determined by Equations (4.14) and (4.15). The LQG controller matrices also determined by the state-space Equations (4.33) ~(4.36) and the Simulink block is shown in Figure 4.4. So as to reduce

the computational time, quantization is used, and digital control analysis also used for the efficiency of the control analysis.

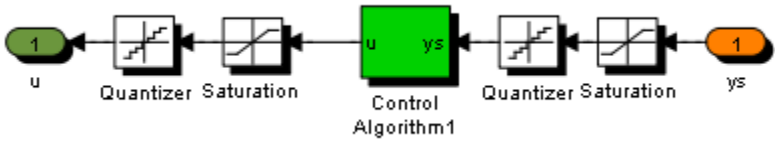

Figure 4.4. Simulink blocks for the LQG controller in LQG control design

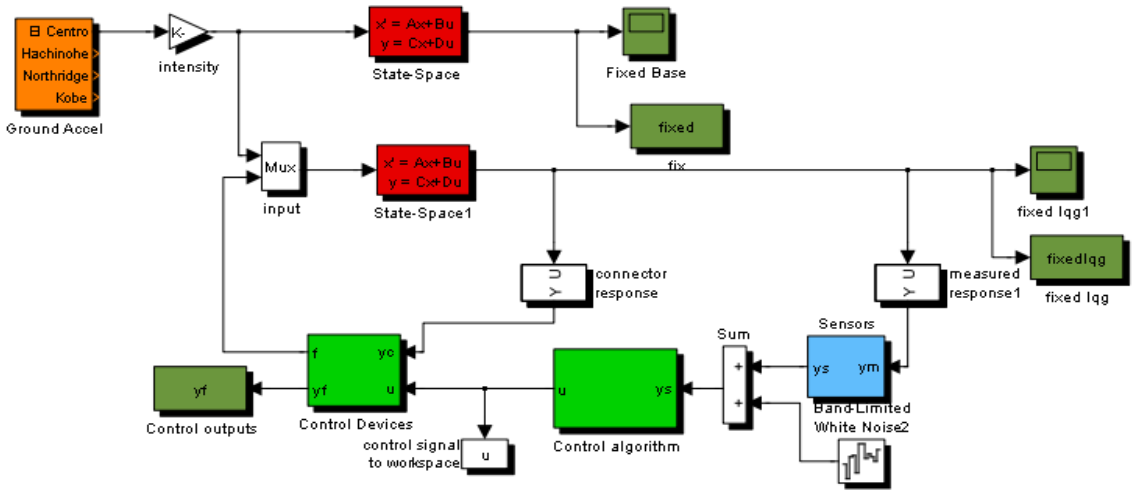

Figure 4.5. Simulink blocks LQG control design

From the combination of LQR and Kalman Filter, the LQG compensator is established, and this LQG compensator is expressed with MATLAB Simulink as shown in Figure 4.5. The Figures from 4.6 to 4.9 develop the 3-strory building system displacement, acceleration, control force and signal outputs and Figures from 4.10 to 4.13 are the 20-strory building system displacement, acceleration, control force and

signal outputs. The displacement, control fore and control signal output of the 1st story of the 3-D 20-story building system by using LQG are shown in the Figures 4.14 and 4.15. Under the El Centro earthquake, the controlled and uncontrolled analyses are carried out. From the comparison with other researcher's results (Ohtori 2004), the displacements and accelerations are nearly identical. Thus, the LQG controller design is successfully performed, and then these three LQG control problems are ready to be used as control analysis with diverse positions and numbers of active dampers and sensors. The optimization methods for the optimal architectures of the active devices and sensors will be suggested in next sections.

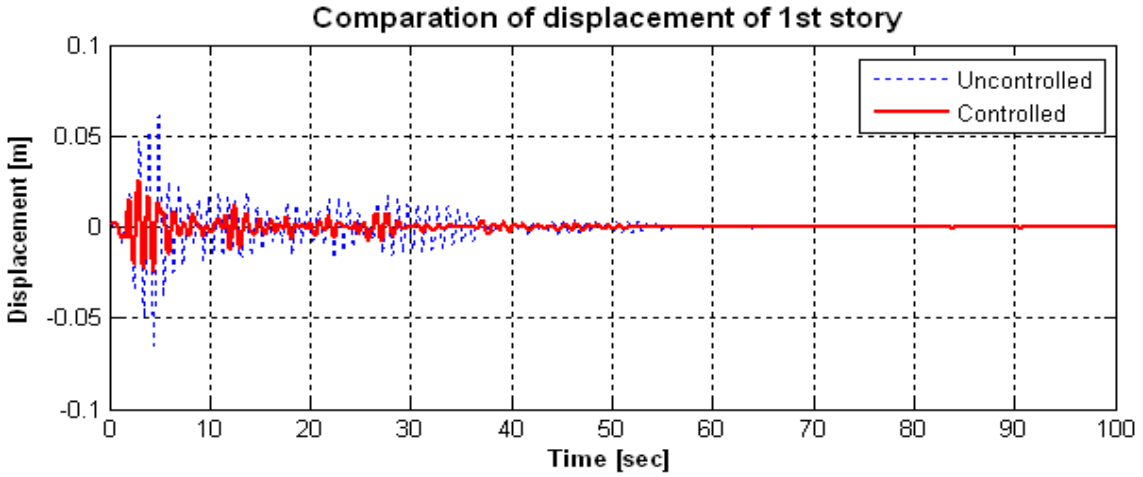

Figure 4.6. 3-strory building system controlled and uncontrolled displacement responses

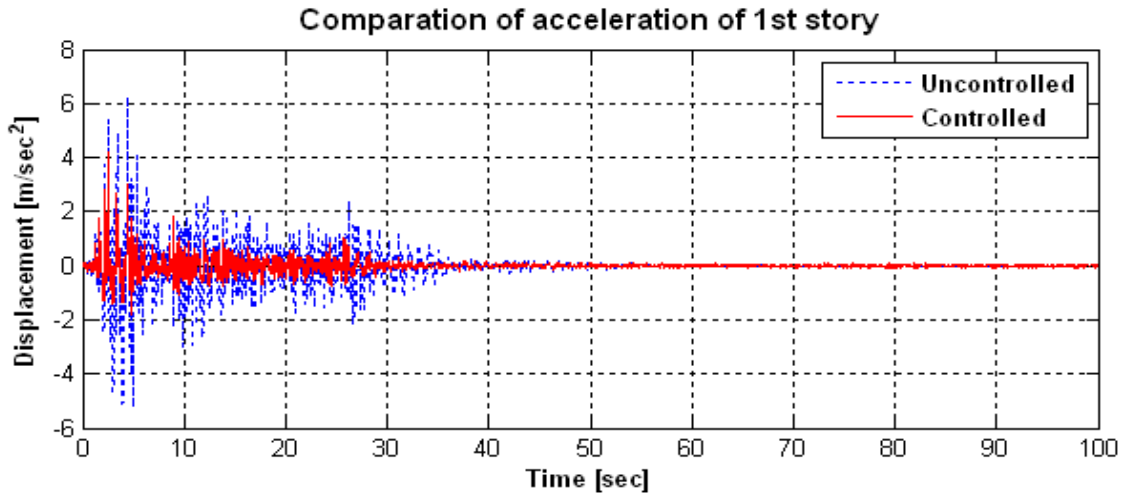

Figure 4.7. 3-strory building system controlled and uncontrolled acceleration responses

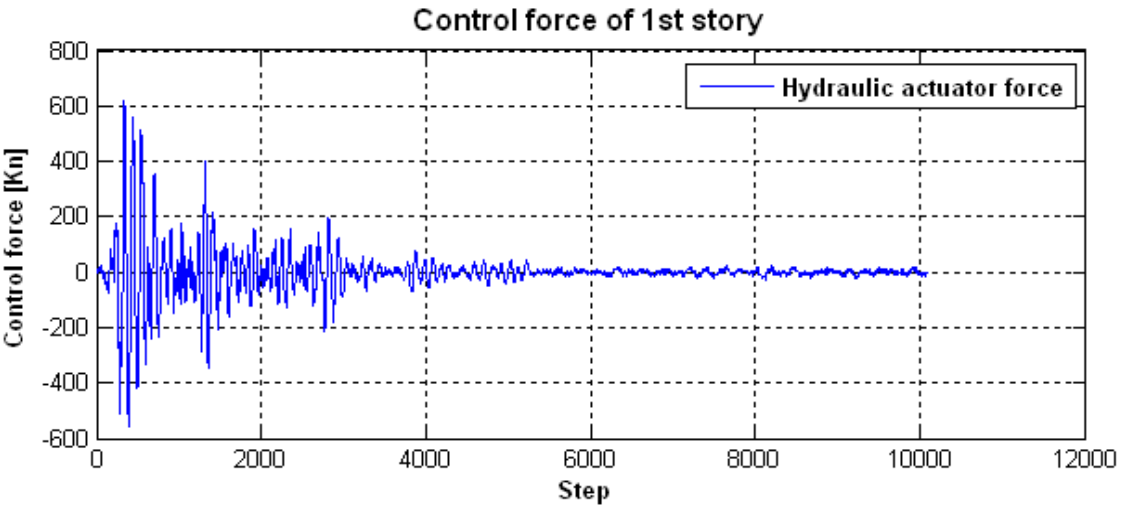

Figure 4.8. 3-strory building system control force outputs

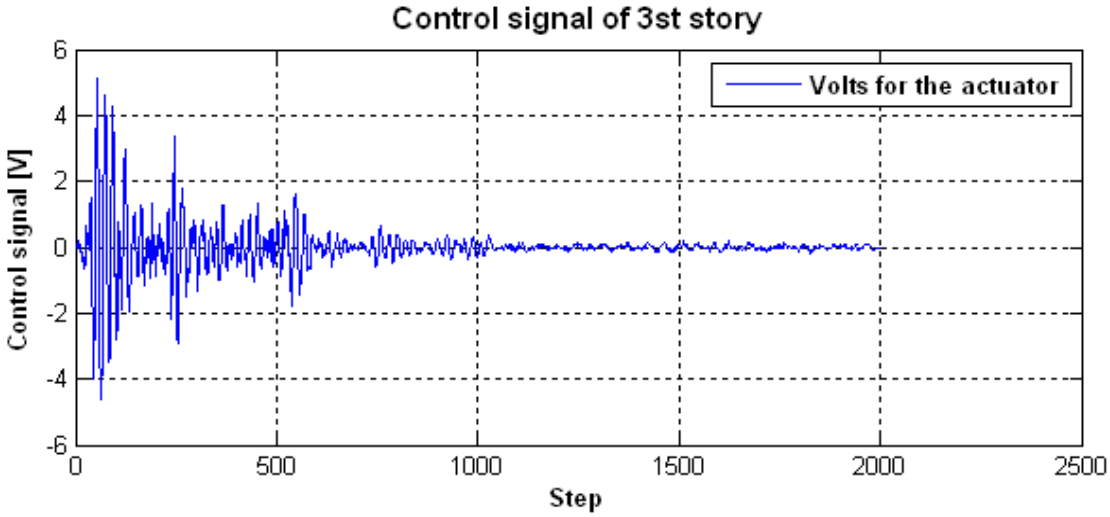

Figure 4.9. 3-strory building system control signal outputs

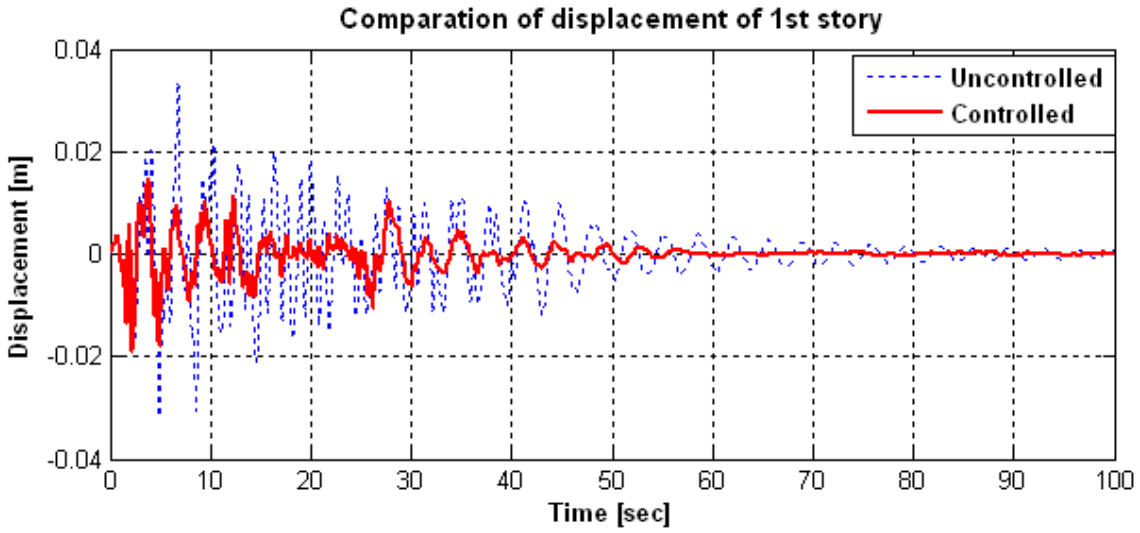

Figure 4.10. 20-strory building system controlled and uncontrolled displacement responses

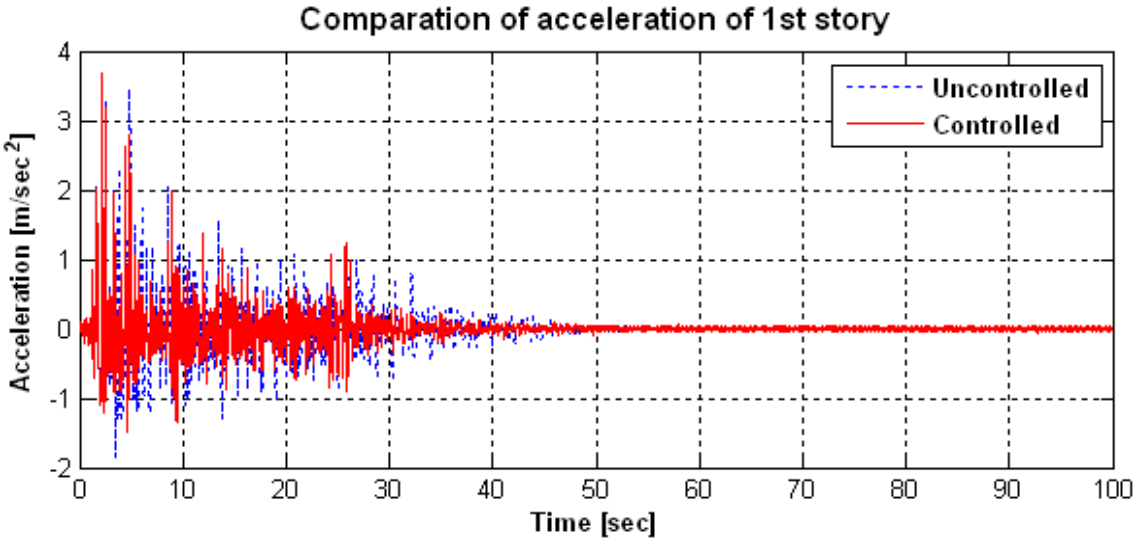

Figure 4.11. 20-strory building system controlled and uncontrolled acceleration responses

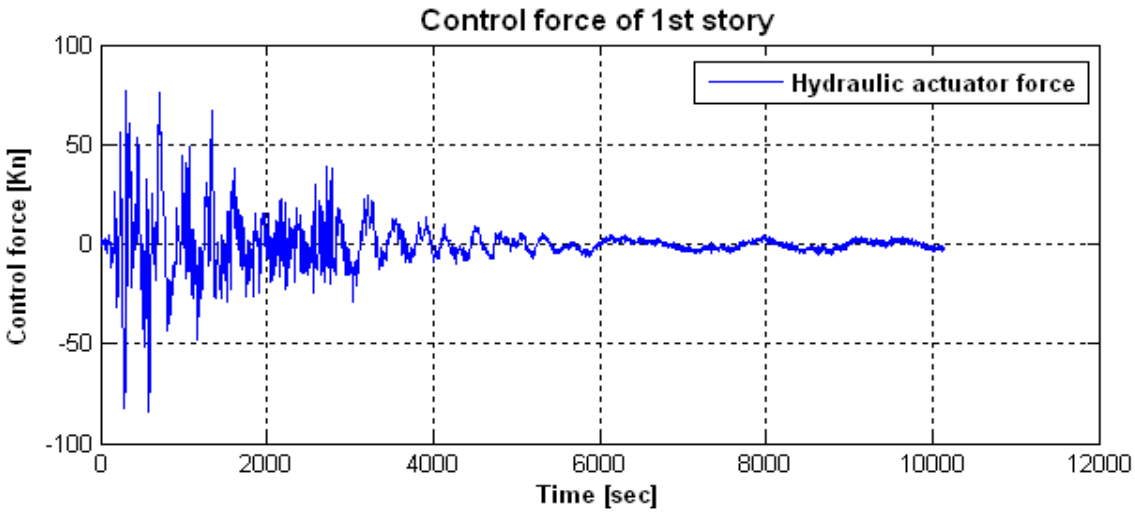

Figure 4.12. 20-strory building system control force outputs

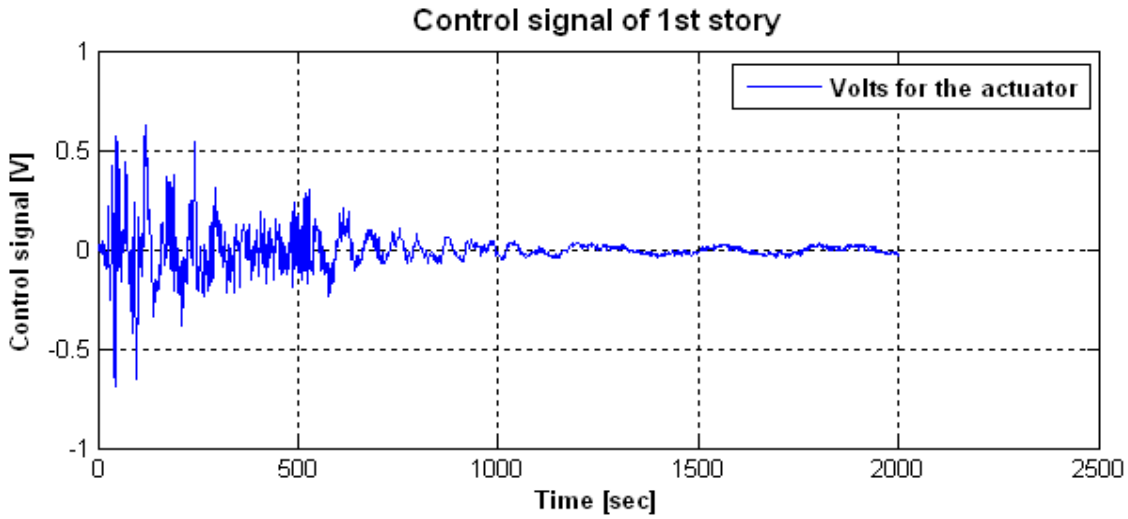

Figure 4.13. 20-strory building system control signal outputs

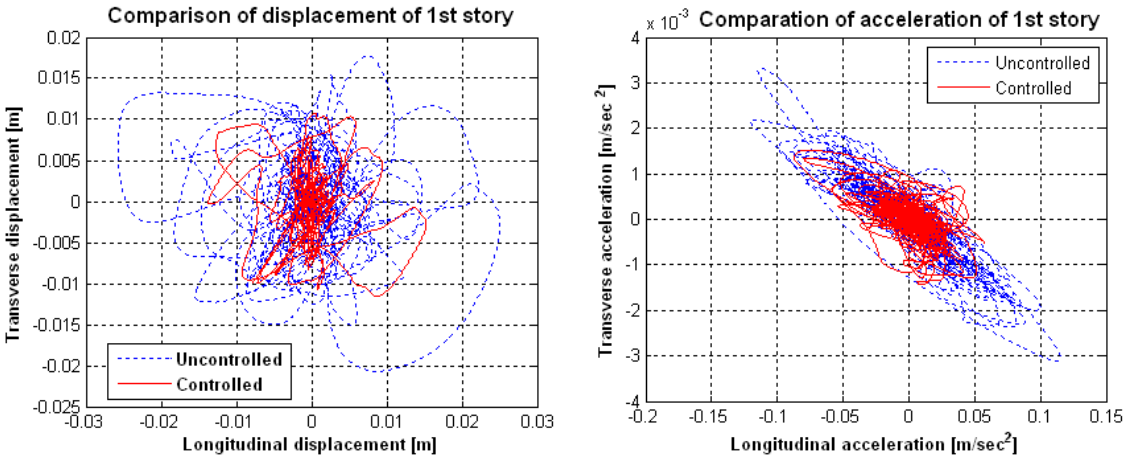

Figure 4.14. 3-D 20-strory building system controlled and uncontrolled displacement and acceleration responses

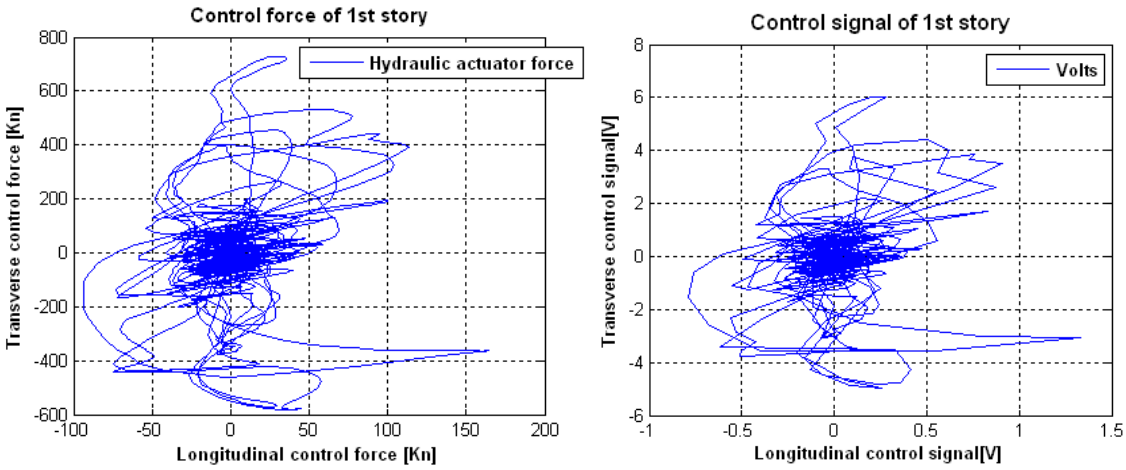

Figure 4.15. 3-D 20-strory building system controlled and uncontrolled displacement and acceleration responses

5 PERFORMANCE COMPARISONS OF SGA AND IRR GA AND INTRODUCTION OF MULTI-OBJECTIVE OPTIMIZATION METHOD

In order to find the near-optimal set of the numbers and locations of active dampers and sensor, all the control design variables such as locations and numbers of dampers and sensors in each story are expressed in a binary string. The performance of genetic algorithms (GA) is very dependent on the encoding policy which is how to encode the design variables to a binary string. To consider the high discrete and non-continuous characters of the feasible number and location of the damper and sensor in active control of civil structures, flexible and unstructured form of encoding policy is required rather than structured form like simple genetic algorithm (SGA). With the redundant segments in a binary string, the drifts of gene locators (GL) can search very random, diverse and wide range of feasible design variables. Thus, in this Section, the two encoding policies are studied. The one is SGA and the other is implicit redundant representation (IRR) GA to compare the search ability of each encoding policies of the GAs about the active damper and sensor placement problem, even though Raich (1999) already verified the supremacy of IRR GA than SGA in the truss structure optimization unstructured problem.

Installations of the active dampers and sensors in civil structures are not always possible in any location where is theoretically and practically ideal because of infeasible

structural type or form or impossible data acquisition of the structural responses. Thus in order to offer diverse design alternatives to control designers, the diverse architectures of the dampers and sensors design option should be searched. The multi-objective optimization methodology can offer a near optimal set to designers. If two or more objectives are required, the two objectives can not be satisfied in same time with best optimal value. With the concept of Pareto curve, the multi-objectives can be considered evenly without emphasizing one of the objectives. Thus, the proto type of the multi-objective optimization methodology to search the near optimal Pareto front for the architectures of the placement of active dampers and sensors is suggested based on the IRR GA and well-known non-dominated sorting genetic algorithm (NSGA) developed by Srinivas and Deb (1994).

5.1 Comparison of the Performance between SGA and IRR GA

The performances of SGA and IRR GA are compared to develop multi-objective optimization methodology. Even though Raich (1999) already tried to compare the search ability in truss structure optimization in unstructured problem domains, this optimal placements of active dampers and sensors are a structured problem, and the problem characters are also very different with the truss structure design problem. Then the performance between SGA and IRR GA needs to be compared before research of the optimization of the architectures of the active dampers and sensors.

The basic format of the SGA binary string is shown in Figure 5.1. To express the number of the installed active dampers in each story of the 3-story building, 3 bits are

used, and each sensor's presence/absence is determined by an 'on/off' bit (1/0 bit) in the GA individuals. Thus each binary string length is 12 bits; the number of dampers in each story is expressed in 3 bits and sensor's presence/absence is expressed in 1 bit.

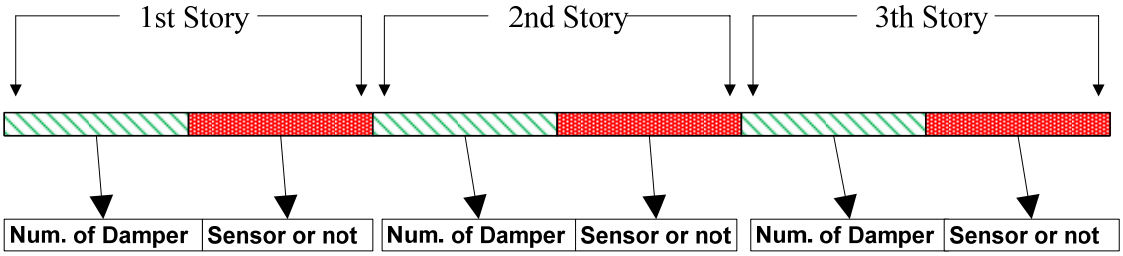

Figure 5.1. Encoding policy of the SGA

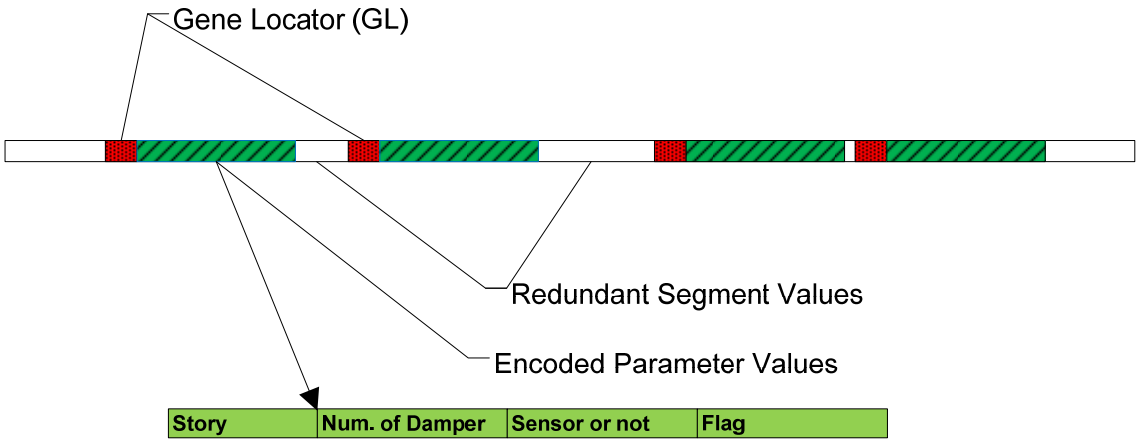

Figure 5.2. Encoding policy of the IRR GA

With redundant segment and GL, the gene instance which has design variables in IRR GA can drift in a binary string randomly and freely. As discussed in Section 2, the IRR GA has strong diverse and wide search ability. The basic format of an IRR GA

string is shown in Figure 5.2. Unlike the SGA pattern, IRR GA can express its design variables with partially overlapped way.

For example, if the story section bits express the 2nd story in first and second gene instances and 1st story in third and fourth gene instances, to get the 1st and 2nd story design variable information, the flag variable information should be compared each other and chosen the biggest flag value one in each two gene instances (1st and 2nd, and 3rd and 4th) in Figure 5.2. In this case there is no information for the 3rd story. So the third story's number of damper and sensor should be zeros.

In the IRR GA string, the each string length is dependent on the user's decision. The user needs to consider the probability of a single occurrence of a specific GL pattern. The probability of a single occurrence of a specific GL pattern ([1 1 1]) is calculated as following Equation (5.1) (Raich 1999):

$$P = \beta^n \left[\frac{1-\beta}{1-\beta^n} \right] \sum_{j=0}^{n-1} \beta^j \tag{5.1}$$

where

β = the probability of a single occurrence of the specified bit value,
n = the number of bits specified in the GA pattern.

The required number of gene instances for the 2-D 3-story cases is 3 because it is 3-story building, so at least 3 gene instances are required. The number of bits in an encoded gene instance is 9 with 2 bits for story, 3 bits for number of a damper and 1 bit for a sensor, and the number of bits specified in the gene locator (GL) pattern is 3. So we can calculate the minimum required length of string by the Equation (5.2) (Raich 1999):

$$N_0 = \frac{(l_s - l_g + 1)P}{(1 + l_g P)}$$
$$l_s = \frac{N_0(1 + l_g P) + P(l_g - 1)}{P} \tag{5.2}$$

where

l_s = the number of bits in the string minus (n-1) bits to account for the end of the string,
l_g = the number of bits in an encoded gene instance,
P = the probability of a single occurance of a specific GL pattern,
N_0 = the total number of instances of GL pattern and encoded genes.

With the defined variable encoding policy, in order to apply GAs to carry out the numbers and locations of the actuator and sensor optimization problem, the most important thing is establishing a flowchart of the whole control analysis and GA process. The control analysis is embedded to the step of the evaluation of binaries. The objective function needs to be defined and several kinds of checking list such as controllability and stability also treated in the middle of the GAs' iteration. The flowchart for the performance comparison of the SGA and IRR GA is shown in Figure 5.3. The fundamental procedures has steps: generating initial population, evaluating the binaries, selecting good binaries to make new binaries, doing crossover, and doing mutation to make new population for the next generation.

When evaluate each binary, the 4 things should be checked; the controllability, observability, stability and the number of sensors which are installed in whole structure. Although the control system is controllable and the responses are also observable and the whole control system is stable, the binary has the number of sensors installed in structure

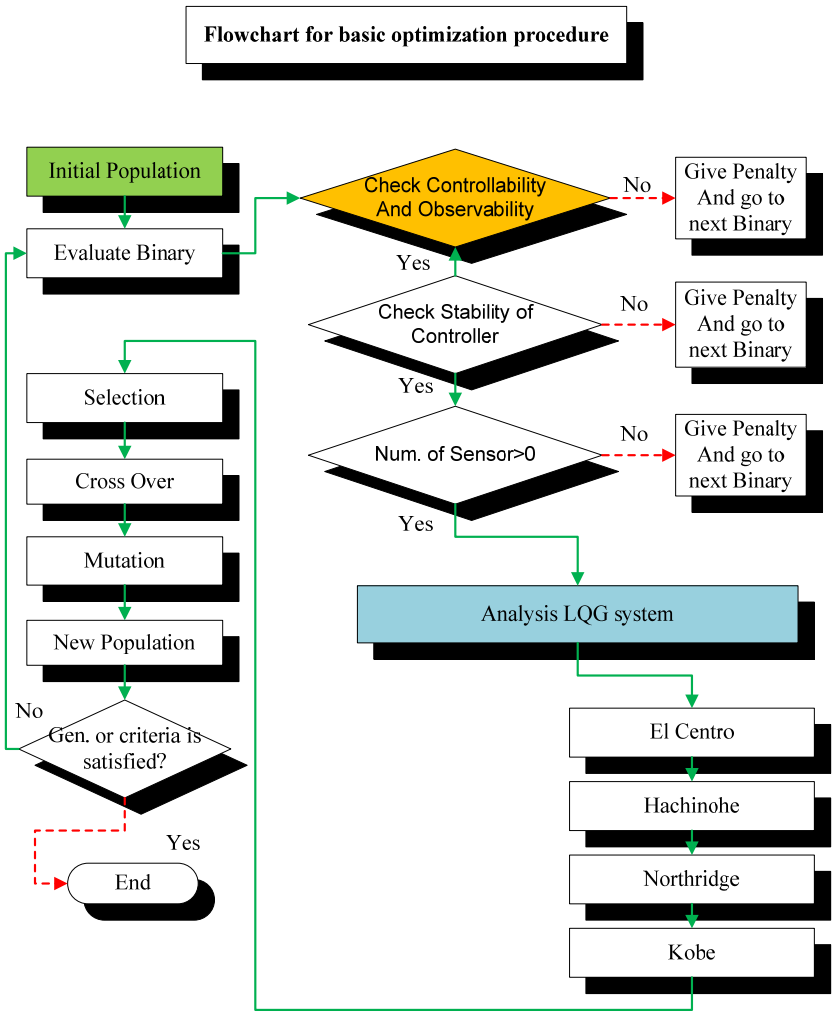

Figure 5.3. Basic flowchart of the optimization procedure

is zero. The control system is finally impossible to control the earthquake response because Linear Quadratic Gaussian (LQG) is optimal feedback controller. When the feedback controller does not have information about the observed responses of the system, the estimator of feedback controller do not estimated the other unobservable responses of the system. Therefore, the control system needs to be checked the number of sensors in the binary. When the number of a sensor is zero, the binary should be

penalized with defined value. Thus in the selection step, the penalized binaries are hard to survive because of their low fitness. The binaries already passed the checking section can be the inputs of the LQG control analysis about 4 well-known earthquakes: El Centro, Hachinohe, Northridge, and Kobe.

A maximum drift is the objective. In the selection step, the tournament selection method was used with the tournament selection size, 3. The crossover is carried out with single random point method. This crossover operator is recombination operator. This step does not make any new genetic information but just do the recomposition and reposition of genetic information which is already existed in the current population. The last step is the mutation which helps to prevent the current population from pre-convergence to local optima and losing diversity. It introduces or reintroduces information into the population. These steps can be one genetic loop and this loop will be continued until satisfying defined criteria or generation.

The GA properties for the 3-story building are expressed in Table 5.1. From the Table 5.1, with small iterations of the GA' loops compared to SGA, the IRR GA found the better minimal maximum drift with same population size, mutation and crossover probabilities. The maximum drifts which are non-dimensionalized and normalized with respect to the associated floor height are very important damage index (Barroso 1999):

$$J = \max\left\{\frac{\max\limits_{t,i}\dfrac{\left|d_i^c(t)\right|}{h_i}}{\max\limits_{t,i}\dfrac{\left|d_i^u(t)\right|}{h_i}}\right\} \tag{5.3}$$

where $d_i^c(t)$ and $d_i^u(t)$ are each controlled and uncontrolled interstory drift of the above ground floors over the time history of earthquakes: El Centro, Hachinohe, Northridge and Kobe. h_i is the height of each of the associated stories, and the interstory drifts are calculated by (Spencer 1999):

$$d_i(t) = x_{\eta i}(t) - x_{\eta i-1}(t) \tag{5.4}$$

where

η = the set of states corresponding to the horizontal displacement of above ground floors
i = the floors of the benchmark buildings (1~3, 1~20, 1~40:3D 20 story building)
$x_{\eta 0}(t)$ is assumed to be zero

The objective function is described as:

$$F = 1 - J \tag{5.5}$$

generally the maximum drifts do not bigger than 1 because this 3-story building system is fully controllable and observable. Thus, this optimization problem is the finding maximum value of the objective function.

Table 5.1. Comparison of the genetic properties of SGA and IRR GA

	SGA	*IRR GA*
String length	12	54
Population size	40	40
Mutation probability	0.01	0.01
Crossover probability	0.8	0.8
Number of Damper	5 : 5 : 5	5 : 5 : 3
Sensor locations	1st and 2nd story	1nd story
Full convergence	Generation: 40	Generation: 8

From the comparison and detail observation of the Figures from 5.4 and to 5.9, the locations of the active actuators and sensors are very important than just installation of many dampers and sensors as many as they are possible to install. It is contrast to the general engineering judgment; the more power the better performance. This fact gives us why we have to research about the optimal placements of dampers and sensors in civil structure control problem. The acquired optimal placements of the actuators and sensors in each story is as shown in Table 5.1: [5, 5, 3] from the 1st to 3th story of building and the sensor location also 1nd story are the best place to reduce the maximum drifts. Furthermore the search performance of the IRR GA is also better than the SGA with small generation. From this 3-story benchmark building case, the IRR GA has advantage in saving central processing unit (CPU) analysis time with better optimal results. Thus the IRR GA will be used for the GA search engine in this research.

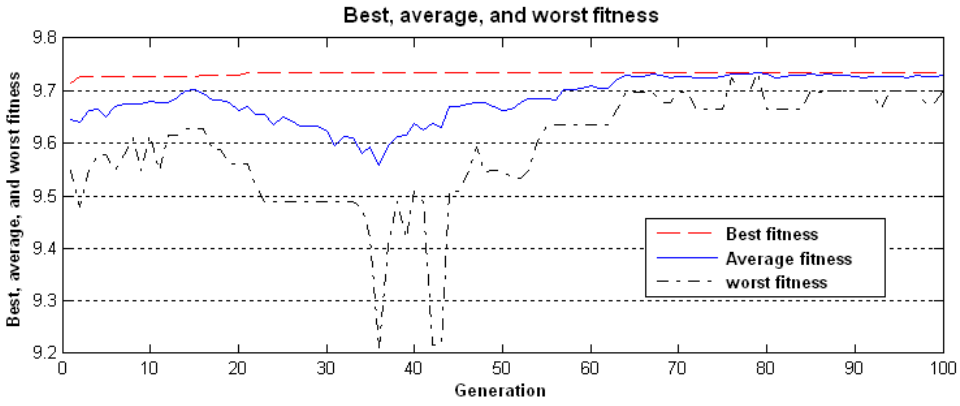

Figure 5.4. Best, average, and worst fitness during generations using SGA

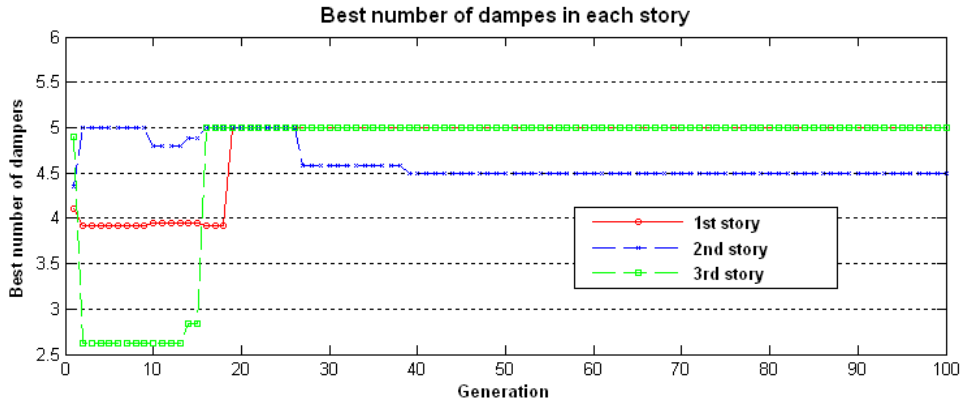

Figure 5.5. Best number of dampers in each story using IRR GA

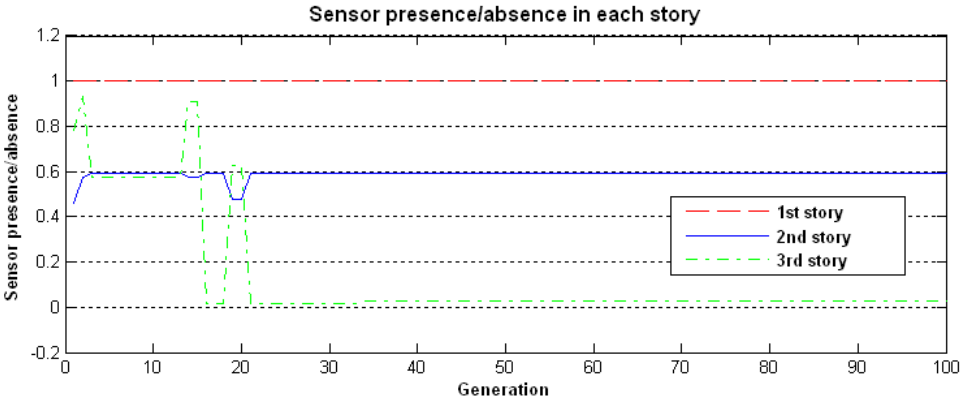

Figure 5.6. Presence/absence of sensor in each story using SGA

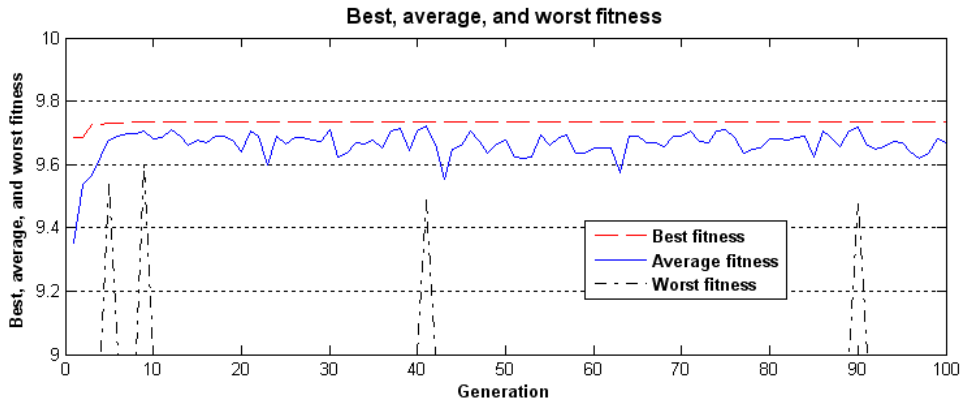

Figure 5.7. Best, average, and worst fitness during generations using IRR GA

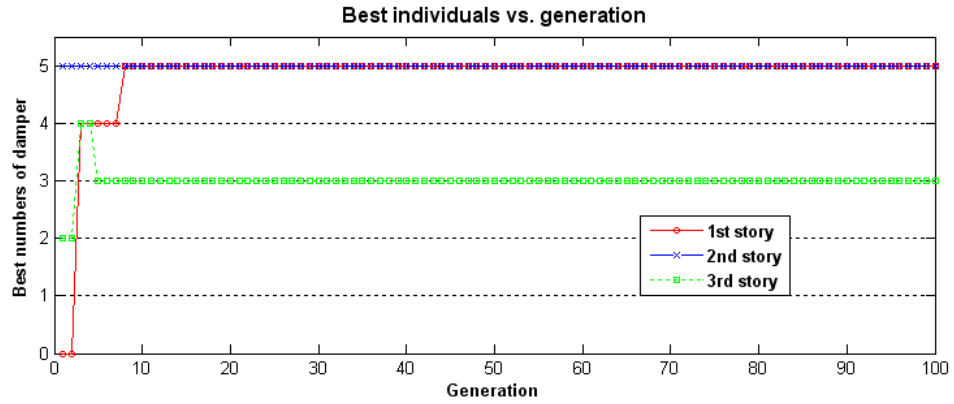

Figure 5.8. Best number of dampers in each story using IRR GA

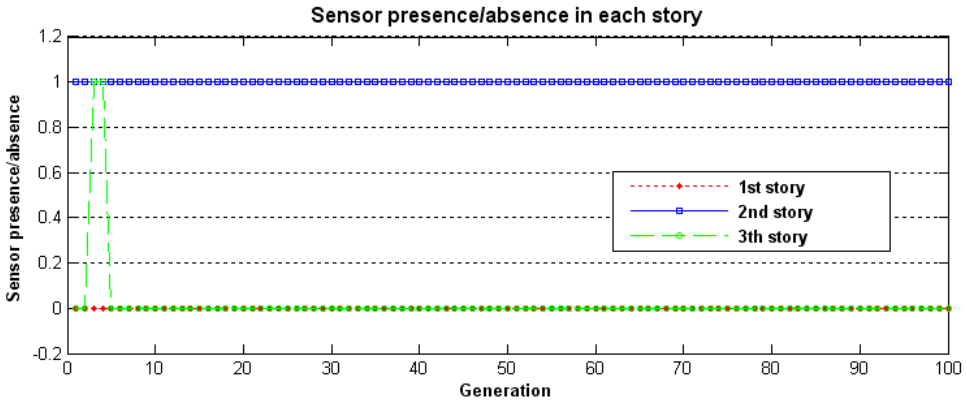

Figure 5.9. Presence/absence of sensor in each story using IRR GA

5.2 Multi-objective Optimization Using NSGA and IRR GA

In this section, with the basic flowchart of optimization procedure as shown in Figure 5.3, and the IRR GA and the non-dominated sorting genetic algorithm (NSGA) suggested by Srinivas and Deb (1994) is adopted as multi-objective selection method. This combined IRR GA and NSGA is defined as NS-IRR GA. The flowchart of the NS-

IRR GA is shown in Figure 5.10. As we already reviewed, NSGA concept is only different in selection step compared to SGA. Generally, NSGA offers unbiased Pareto-optimal set by using the sharing method. As shown in Figure 5.10, the detail procedure is expressed in the dashed line box.

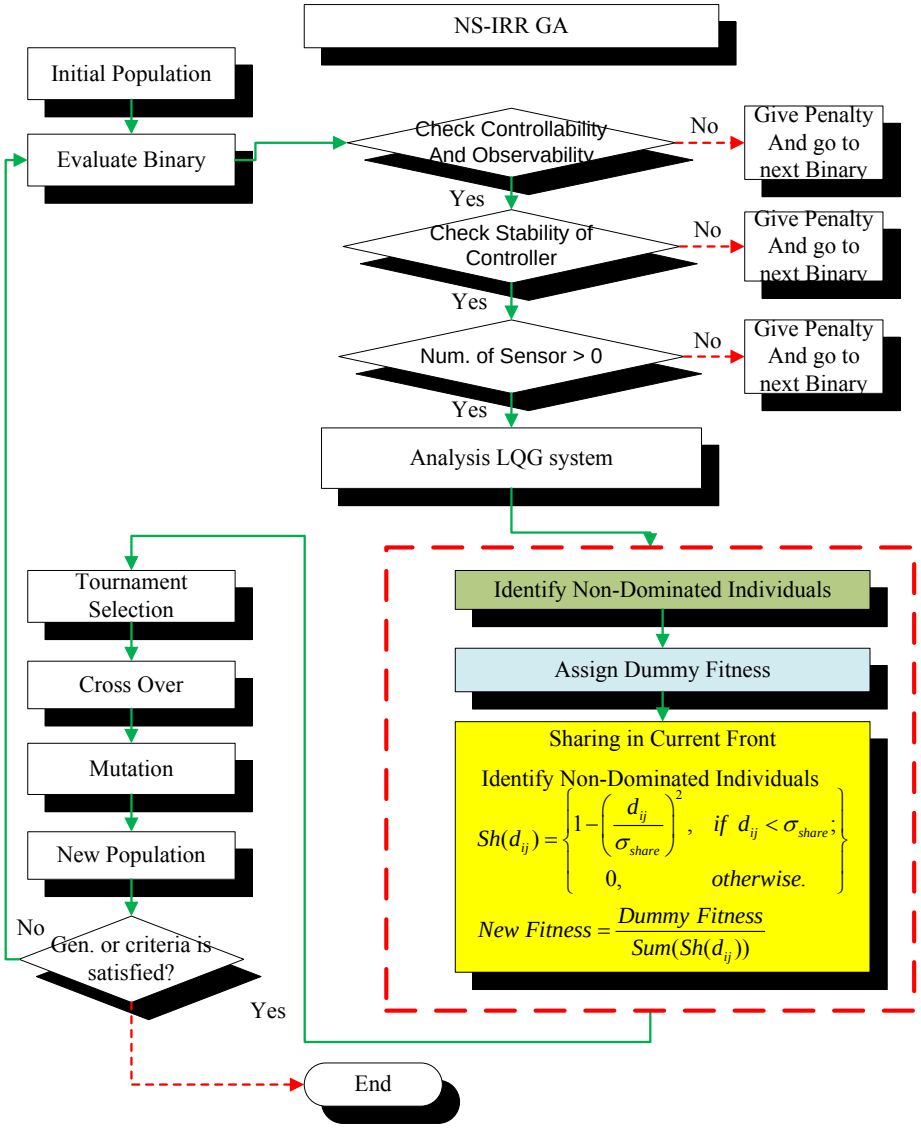

Figure 5.10. Flowchart of NS-IRR GA

For the first step of the NS-IRR GA, the non-dominated individuals should be found, and then eradicated from the current population set. The non-dominated strings are assigned rank 1. And then find non-dominated individuals from the remnant individuals. These second non-dominated strings' rank is 2, and eradicated from the current population set. These procedures are repeated until all the individuals in current population are ranked. Basically the same rank individuals have a same reproductive potential by assigning dummy fitness values. The sharing function is applied to assign degraded fitness values which are calculated by the $Sh(d_{ij})$ value to the same rank individuals to give a difference in selection probability for the next generation. To get $Sh(d_{ij})$ values of the individuals in Figure 5.10, the relative distance, d_{ij} , which is the distance between individuals i and j should be calculated in the same ranked individuals. If the σ_{share} defined by the user is bigger than calculated distance like case B, d_{ij}, the $Sh(d_{ij})$ is calculated by the equation as showed in Figure 5.11. In case, the d_{ij} is bigger than σ_{share} like case A, the zero value is assigned to $Sh(d_{ij})$. The final fitness value for the each individual is determined by dividing the summation of the $Sh(d_{ij})$ to dummy fitness in a same rank as shown in Equation (5.6).

$$New\ fitness(i) = \frac{dummy\ fitness(i)}{Sum(Sh(d_{ij}))} \tag{5.6}$$

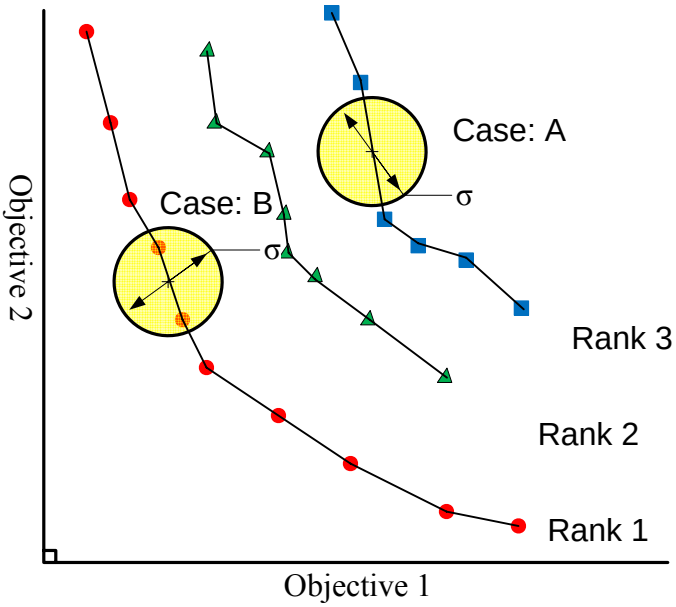

Figure 5.11. Sharing schema of NSGA

These procedures continue until all the individuals in all ranks to get the final fitness values. The minimum fitness value of the first rank individual should be bigger than the maximum fitness value of the second rank individual.

5.2.1 The Pareto Optimal Curves Using the NS-IRR GA

The 3-story benchmark building is analyzed by the NS-IRR GA. The analysis parameters for this optimization problem are expressed in Table 5.2. The parameters of the NS-IRR GA are defined from the previous IRR GA analysis, and the string length is also defined based on the redundancy, 0.5. The two crossover rates are tried to experiment the effects of the crossover rate. The crossover rate, 0.9, is better than 0.8 in finding near near-optimal Pareto curve as shown in Figure 5.12. With the small number

of iterations, the good results were achieved. Some of the individuals with crossover probability, 0.8 are better than the probability, 0.9. The number of individuals of the non-dominated front is bigger than the case of 0.8, and more diverse individuals are found. The 0.9 crossover probability will be used in this research for the 20-story building problem.

Table 5.2. NS-IRR GA parameters for the 3-story benchmark building

	3-story benchmark building
String length	54
Max. Generation	200
Mutation probability	0.01
Crossover probability	0.9
Population size	30

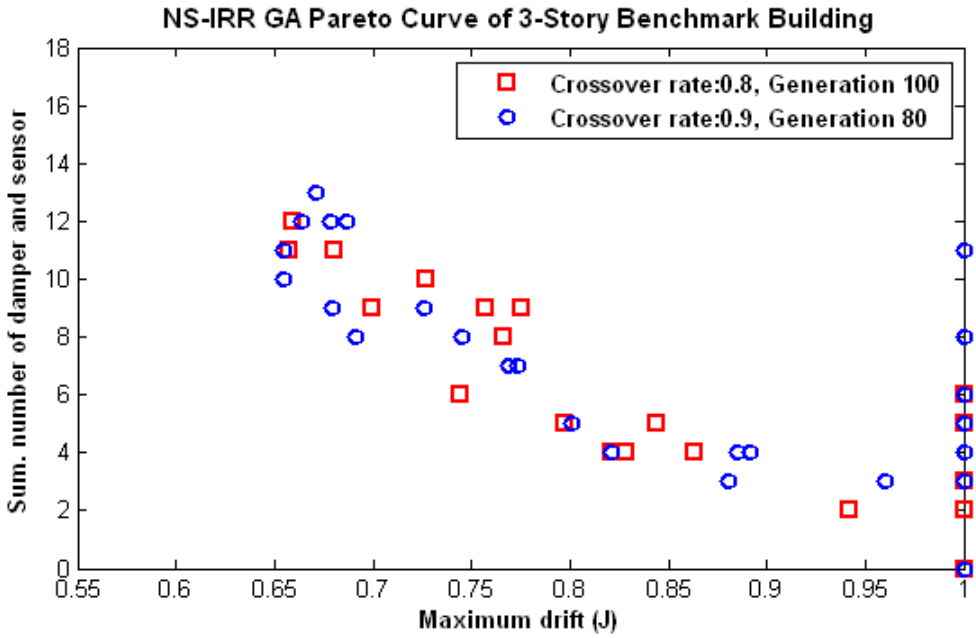

Figure 5.12. Comparison of the crossover probabilities in NS-IRR GA

From the detail survey of each generation result of the NS-IRR GA, NSGA method as the multi-objective selection method has some deficiencies in searching and keeping non-dominated individuals. As shown in Figure 5.13, the non-dominated individuals in the previous generation (80) did not survive in the next generation (90). Although this NS-IRR GA wants to find and keep them for the diverse and best near-optimal result, the non-dominated individuals found in the previous generation are lost and find again because of the week keeping ability during iterations. These losing and finding processes are repeated through the generations. What the problem is that even though the optimization procedure is finished, the best non-dominated individuals did not saved. Thus, to keep the previous non-dominated individuals, better optimization method is needed to be applied. In the next Section, more advanced multi-objective optimization methodologies will be applied to IRR GA.

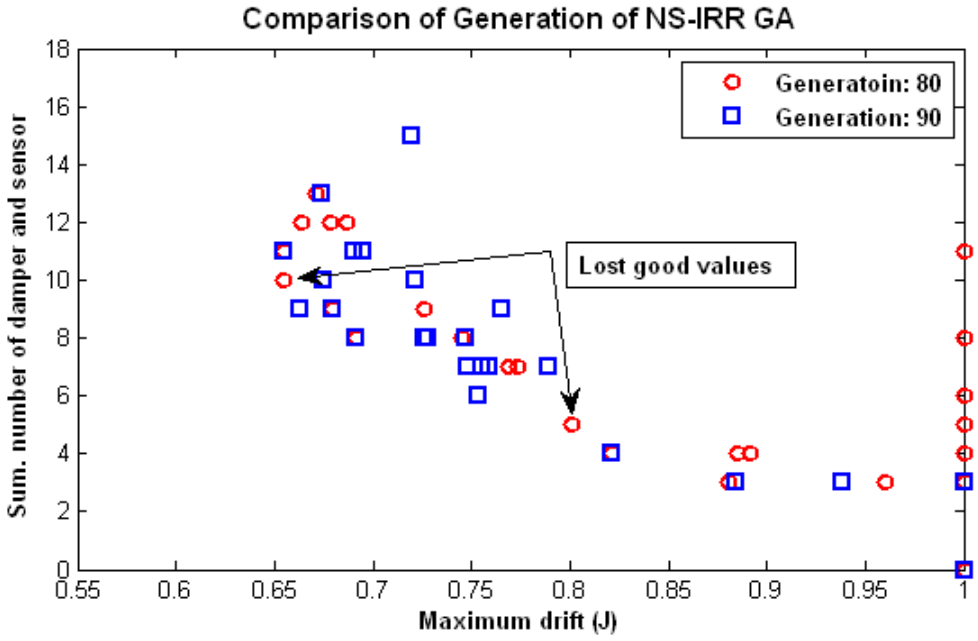

Figure 5.13. Deficiency of NSGA

6 MULTI-OBJECTIVE OPTIMAL PLACEMENTS OF CONTROL DEVICES AND SENSORS USING ADVANCED ALGORITHMS

In the previous Section, the basic multi-objective optimization concept; NS-IRR GA was defined and applied to find the optimal Pareto front in the American Society of Civil Engineers (ASCE) control benchmark problem. However, the NS-IRR GA has some disadvantages in finding and keeping the non-dominated Pareto front individuals. Thus more advanced method is needed to get rid of that kind of problems. Although many multi-objective genetic algorithms (MOGA) were suggested such as niched Pareto genetic algorithm (NPGA)(Horn and Nafpliotis 1993), NPGA2 (Erickson et al 2001), multi-objective genetic algorithm (MOGA) (Fonseca and Fleming 1993), multiobjective messy genetic algorithm (MOMGA) (Van Veldhuizen and Lamont 2000), MOMGA-II (Zydallis et al 2001), Pareto archived evolution strategy (PAES) (Knowles and Corne 2000), Pareto envelope-based selection algorithm (PESA) (Corne et al 2000), strength Pareto evolutionary algorithm2 (SPEA2) (Zitzler et al. 2001, 2002a, 2002b), and non-dominated sorting genetic algorithm-II (NSGA-II) (Deb at al. 2000), the previous comparative studies (Zitzler et al. 2002a) showed that the SPEA2 and NSGA-II performed better than other Gas and their predecessors SPEA and NSGA in all problems; sphear model (SPH-m) function (Schaffer 1985), Zitzler, Deb, and Thiele's sixth (ZDT6) test function (Zitzler et al 1999), Quagliarella and Vicini (QV) function (Quagliarella and Vicini 1997), Kursawe's (KUR) function (Kursawe 1991), and

knapsack problem (KP-750-m) (Zitzler and Thiele 1999). These problems are combinatorial and continuous from the researches. The PESA has fastest convergence, due to its higher elitism intensity, but has difficulties on some problems because it does not always keep the boundary solutions. SPEA2 and NSGA-II showed the best performance overall, and in higher dimensional objective spaces, SPEA2 seems to have advantages over PESA and NSGA-II in long term generation, but slow in convergence. So in this Section, the SPEA2 and NSGA-II methods will be applied to the IRR GA and then SP2-IRR GA and NSII-IRR GA are defined.

6.1 The Description of NSII-IRR GA

Last for a decade years, lots of the multi-objective optimization methods using GA were suggested. The first attempt of the multi-objective optimization is the NSGA (Srinivas and Deb 1994) based on the concept of Goldberg's approach (Goldberg 1989). However, we already certified its disadvantages from the previous Section 5. The main disadvantages are (Deb et al. 2000):

- NSGA can not keep all the non-dominated individuals which are already found from the previous iterations.
- User have to specify the sharing parameter σ_{share}.
- Non-dominated sorting ways used in NSGA is needed to high computational complexity.

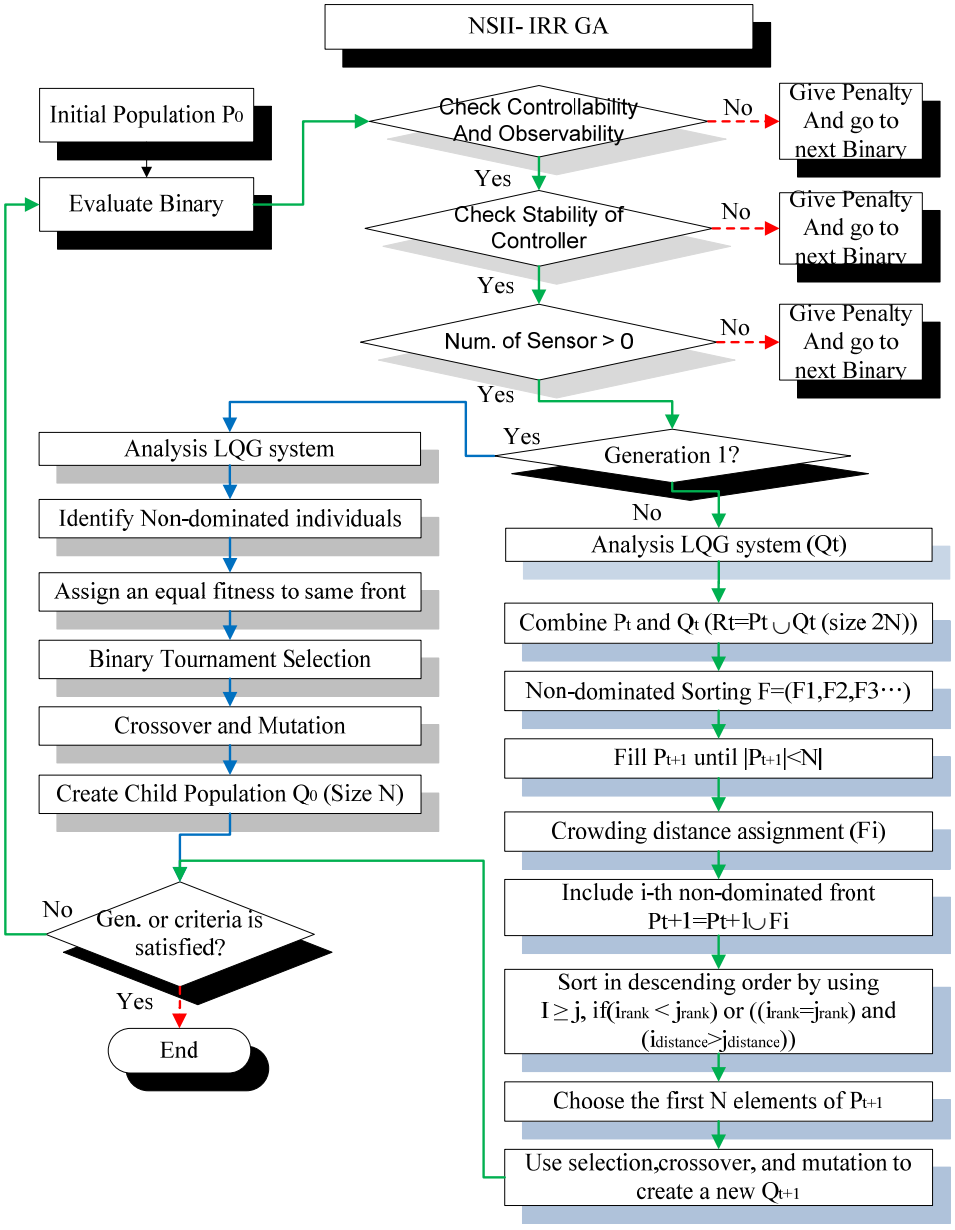

Figure 6.1. Flowchart of the optimization of the control devices architectures based on NSII-IRR GA

Deb et al (2000) suggested much improved version of the NSGA which is called NSGA-II to address all the disadvantageous issues. In this Section, IRR GA and NSGA-II are

compounded and then defined NSII-IRR GA to carry out the control device placement optimization problem as shown in Figure 6.1.

The basic procedure of the NSII-IRR GA is same as the NS-IRR GA, but the fitness assignment and selection steps are different. In the NSGA-II, the fast-non-dominated–sorting method was suggested to reduce the running time, but this research is related to the control analysis with 4 well-known earthquakes. The each running time about one individual about 5 minutes in 2-D 20-story and 7 or above minutes in 3-D 20-story building models, but the GA parts took just 1~2 second with non-dominated-sorting without the control analysis step. Thus in this NSII-IRR GA, the traditional non-dominated sorting method is used because the fast nondominated-sorting did not save much time than the normal sorting method. The running time in commercial language software programs such as C++ or Matlab is highly dependent on how to code the algorithms. Furthermore, it can be good to compare the performance of SPEA2 performance in selection step.

The detail procedure of the selection part for the NSII-IRR GA is followed to the NSGA-II. For the first step, a random population, P_t , is generated. And the non-dominated sorting is performed. The same rank will be assigned same dummy fitness value. The binary tournament selection is carried out and then the genetic operators also performed to create child population, Q_t with predefined size N . With the created P_t and Q_t, a combined population R_t is formed with size $2N$. This combined population is sorted according to non-domination. With the non-dominated sorted population, R_t, to

fill the P_{t+1}, the non-dominated fronts not just a individual will be added to the P_{t+1} until the size exceeds N. The individuals in P_{t+1} are assigned the *crowding distance*. To estimate the density of individuals surrounding a particular point in the phenotype non-dominated Pareto front graph, the average distance of the two points on either side of this point along each of the objectives is used as *crowding distance.* This crowding distance is used for estimating of the size of the largest cuboid enclosing the point *i* without including any other point in the population. As shown in Figure 6.2, and Equation (2.17) , the crowding distance, $I[i]_{\text{distance}}$ is (Deb et al. 2000):

$$I[i]_{\text{distance}} = I[i]_{\text{distance}} + (I[i+1].m - I[i-1].m) \tag{6.1}$$

where

m : the number of objectives
$I[i]_{\text{distance}}$: the m-th objective function value of the i-th individual in the set I

The P_{t+1} is also sorted according to the crowded comparison operator. Thereafter, only N number of individuals will be selected as P_{t+1}. The crowded comparison operator is showed as followed Equation (2.18) (Deb et al. 2000):

$$i \geq_n j \text{ if } (i_{rank} < j_{rank}) \text{ or } ((i_{rank} = j_{rank}) \text{ and } (i_{\text{distance}} > j_{\text{distance}})) \tag{6.2}$$

where

i_{rank} : Non-dominated rank
i_{distance} : Local crowding distance

From the Equation (2.18), the individuals of lower rank is preferable and if the rank is same, the individual which has larger the crowding distance is selected. The selection,

crossover, and mutation operator are carried out to create new child population, Q_{t+1}, in the P_{t+1}. In the selection step, the binary tournament selection is used together with the crowded comparison operator. These whole steps will be continued until satisfying defined criteria or generation.

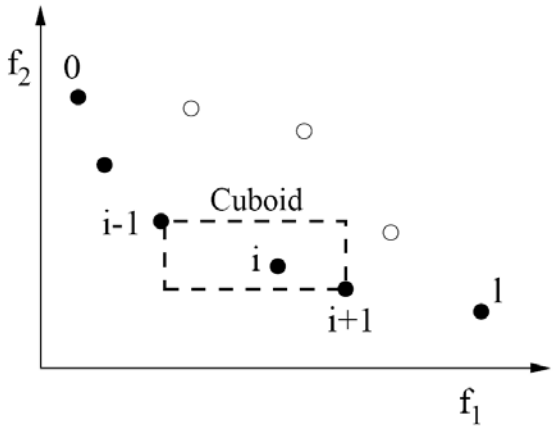

Figure 6.2. The crowding distance calculation (Deb et al. 2000)

6.2 The Description of SP2-IRR GA

Although SPEA was one of the first techniques that were extensively compared to several existing evolution-based methods, NSGA-II and PESA have been shown to outperform SPEA on certain test problems. The SPEA2 version was suggested by the Zitzler et al. (2001, 2002a, 2002b) to overcome its weak points**.** With the excellent ability to keep the non-dominated individuals with archive, the SPEA2 has strong ability to search near-optimal non-dominated front. This improved SPEA2 is applied to active

control actuator and sensor placement problem, and then the SP2-IRR GA is defined. The main flowchart for SP2-IRR GA is shown in Figure 6.2.

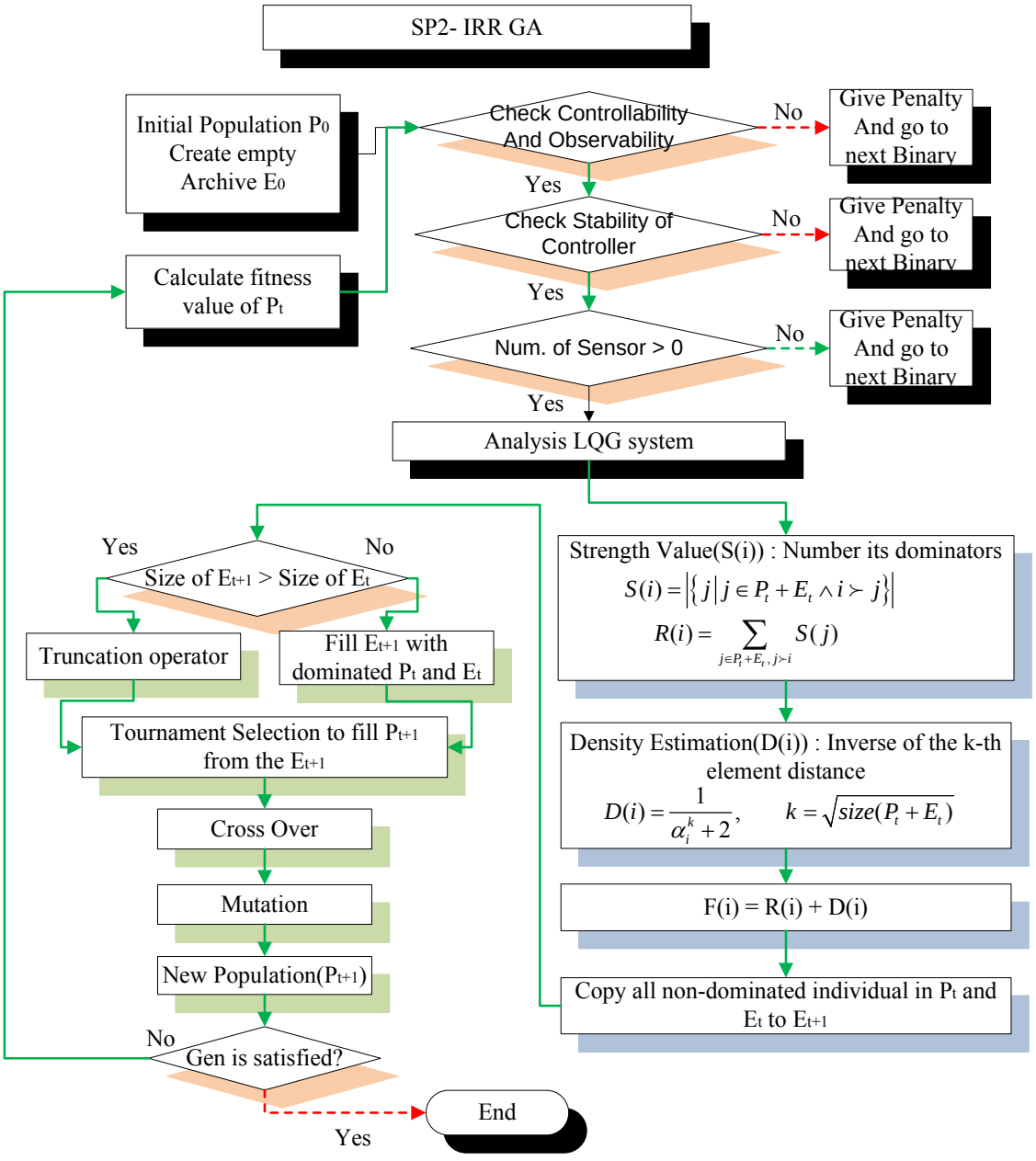

Figure 6.3. Flowchart of the SP2-IRR GA to optimize the control device architectures based on SP2-IRR GA

Unlike the SPEA, SPEA2 uses a fixed archive set size to keep the non-dominated front individuals, a fine-grained fitness assignment strategy, and the members of the archive participate in the mating selection process (Zitzler et al 2001, 2002a, 2002b). The main disadvantages of the SPEA are (Zitzler et al 2001, 2002a, 2002b):

- The archive of SPEA does not consider when the archive contains only a single individual, and all population members have the same rank independent of whether they dominate each other or not.
- In case of many individuals do not dominate each other, density information has to be used in order to guide the search more effectively, and the clustering makes use of this information, but only with regard to the archive and not to the population.
- Although the clustering technique performed well in non-dominated set without destroying its characteristics, but it has some possibility to lose outer solutions.

To complement of the weak points of SPEA, the SPEA2 is applied to perform multi-objective optimization, for the first step, the initial mating population P_0 is randomly generated and empty archive also defined with predefined size. The LQG control is carried out with current population, and then the fitness values are assigned to individuals. The strengths of each individual in the archive and current population are calculated by the following strength value $S(i)$ (Zitzler et al 2002a):

$$S(i) = \left| \left\{ j \,\middle|\, j \in P_t + \overline{P}_t \wedge i \succ j \right\} \right| \tag{6.3}$$

where

$|\cdot|$: the cardinality of a set
$+$: multiset union
$\succ$: Pareto dominance relation

From the basis of the S values, the row fitness $R(i)$ is determined (Zitzler et al 2002a):

$$R(i) = \sum_{j \in P_t + \bar{P}_t, j \succ i} S(j) \tag{6.4}$$

This $R(i)$ means that it adds the strength values of each individual's dominator. The row fitness of the current population considers the number of its dominator and the number that it dominates to consider the situation that there are many same fitness individuals in archive. As already mentioned the SPEA's disadvantages, if most individuals do not dominate each other, this only sort of niching mechanism based on the concept of Pareto dominance will not perform well in selection. Thereafter, additional density information is incorporated to discriminate between individuals having same identical raw fitness values. The density estimate can be the inverse of the k -th nearest neighbor. The common k is equal to (Silverman 1986):

$$k = \sqrt{N + \bar{N}} \tag{6.5}$$

where N is the number of current population size and $\bar{N}$ is the number of the archive size, and the density estimate value is determined by the following (Zitzler et al 2002a):

$$D(i) = \frac{1}{\sigma_i^k + 2} \tag{6.6}$$

where σ_i^k is the k -th nearest distance. Then the final fitness is calculated by (Zitzler et al 2002a):

$$F(i) = R(i) + D(i) \tag{6.7}$$

Thus, in the SPEA2, there is nothing to define using the engineering judgment or assumption rather than NSGA or SPEA. This is strong advantages of SPEA2.

The next step is the environmental selection. The non-dominated individuals should be copied in P_t and $\overline{P}_t$ to $\overline{P}_{t+1}$. If the size of $\overline{P}_{t+1}$ exceeds predefined size $\overline{N}$ because of the added non-dominated individuals, the truncation operator is carried out to sustain the archive size even. In this case, the individual which has the minimum distance to another individual is chosen for removal, but if there are several individuals with minimum distance, the tie is broken by considering the second smallest distances. This process is repeated until the size of $\overline{P}_{t+1}$ is same as defined number $\overline{N}$. If the size is smaller than the size $\overline{N}$, the $\overline{P}_{t+1}$ is filled with dominated individuals which exist in P_t (Zitzler et al 2002a). For the last step, the new mating pool is selected from the $\overline{P}_{t+1}$, and then the genetic operator is performed as shown in Figure 6.3.

6.3 The Optimal Results of the 3-story Benchmark Problem

The optimization methods are suggested to find the optimal Pareto fronts about the 3-, and 20-story ASCE control benchmark problem in the previous paragraphs. To test the performances of the defined optimization methods, the two methods are applied to 3-story benchmark problem. With the NSII-IRR GA, the Pareto front is obtained as showed in Figure 6.4. The crossover rate is 0.9, mutation rate is 0.01, the population size is 40, and string length is 175. For the gene locator, the continuous three ‘1’ bits are used.

The minimal maximum drift is 0.6544 with the summation number of dampers and sensors, 11. The Pareto front is well distributed in the feasible area. When the crossover rate is big enough, the optimal results is also better than with small crossover rate as shown in Figure 6.5. Although the optimal results are not big different, the optimal Pareto front is converged to local minima with crossover rate 0.8 and short running time.

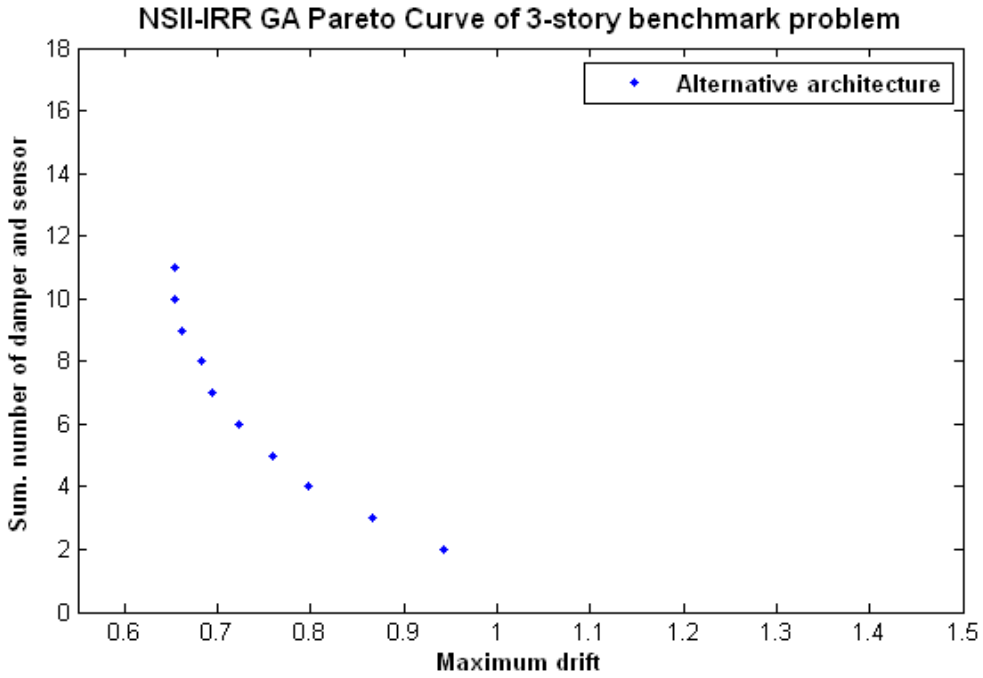

Figure 6.4. The Pareto front obtained using NSII-IRR GA of the 3-story problem

With the SP2-IRR GA, the same optimal analysis was carried out as shown in Figure 6.6. The whole properties for the GA is same as the NSII-IRR GA, but the current mating pool size is 40 and archive size is also assigned as 40.

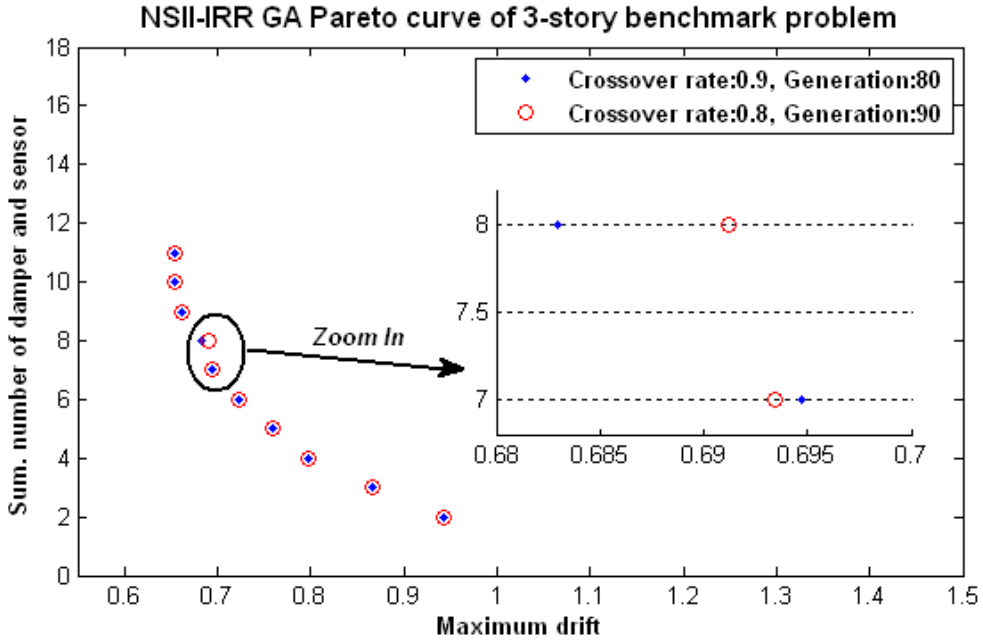

Figure 6.5. Comparison of the Pareto front obtained from different crossover probability

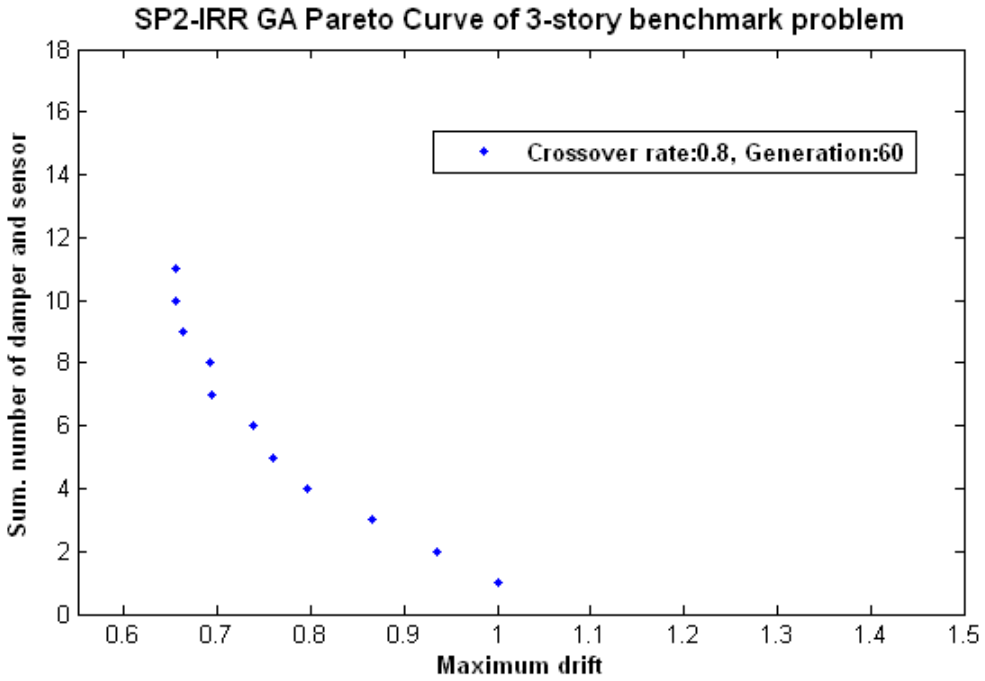

Figure 6.6. The Pareto front obtained using SP2-IRR GA of the 3-story problem

What the interesting things are that there was no difference in optimal result by changing the crossover rate unlike the NSII-IRR GA case. Thus, in order to consider the advantages of the SP2-IRR GA, the bigger archive size is used as 60, but this investigation also can not protect some of the Pareto front individuals from converging to local optima as shown in Figure 6.6.

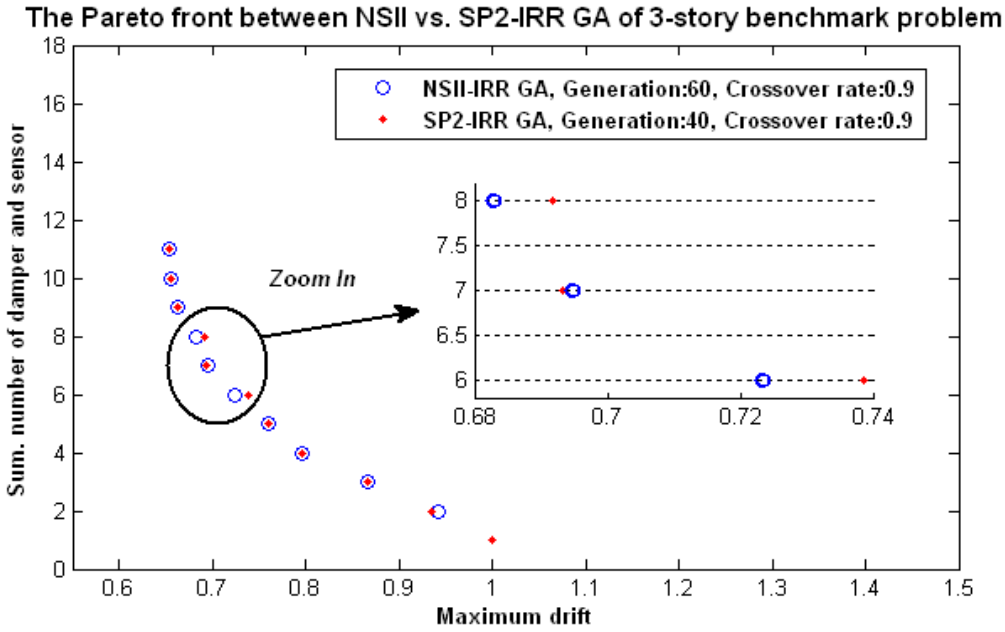

Figure 6.7. Comparison of the Pareto front between NSII- vs. SP2-IRR GA of the 3-story benchmark problem

From the 3-story control benchmark building, the two genetic optimization methods have good performance in finding Pareto front of placement architectures of the dampers and sensors in each story. The NSII-IRR GA has better performance than the SP2-IRR GA in the 3-story problem even though more running time is required as shown in Figure 6.7. Unlike my expectation, the SP2-IRR GA needs some more running

time because of the archive set and converges to the local minima, and it is hard to protect the optimal values from converging to local optima.

From these diverse trials using NS-, NSII-, and SP2-IRR GA, the best Pareto front is selected for the 3-story problem. The best one is the Figure 6.4 with NSII-IRR GA. The detail data are shown in Table 6.1. The installation architectures of the dampers and sensor are very interesting because the available maximum installation number of the damper is 4 in each story, but the efficient maximum numbers of the dampers are not all 4. What the installation of damper in first or lower story is the best policy to reduce the maximum drift is the general engineering common knowledge of the dynamics and control areas. However, from the Table 6.1, the first 3 cases show that the damper locations are very different dependent on the sensor locations. So from these data analysis, although the sensor price is cheap rather than damper price, the locations and numbers of the sensors are significantly important.

Table 6.1. Data of the Pareto front obtained using NSII-IRR GA for the 2-D 3-story benchmark building

Sum.number of dampers and sensors	Maximum drift (J)	1st story	2nd story	3rd story	1st story	2nd story	3rd story
		Damper			Sensor		
11	0.6544	2	3	4	1	1	0
10	0.6546	1	3	4	1	1	0
9	0.6628	0	3	4	1	1	0
8	0.6830	4	3	0	0	0	1
7	0.6947	4	2	0	0	0	1
6	0.7234	3	2	0	0	0	1
5	0.7601	3	1	0	0	1	0
4	0.7970	2	1	0	0	1	0
3	0.8656	1	1	0	0	1	0
2	0.9422	0	1	0	0	1	0

6.4 The Optimal Results of the 2-D 20-story Benchmark Problem

The two suggested optimization methods had good performance in 3-story building. With these fundamental optimization performances, the NSII-IRR GA and SP2-IRR GA are applied to 2-D 20-story building benchmark problem. The near initial population distribution is displayed in Figures. 6.8 and 6.9 with the genetic properties as shown in Table 6.2.

Table 6.2. Comparison genetic properties of NSII-IRR GA and SP2-IRR GA

	NSII-IRR GA	*SP2-IRR GA*
String length	600	600
Population size	100	100, 200
Mutation rate	0.01	0.01
Crossover rate	0.8	0.8
Generation	700	910
Tournament Selection	Tournament size:2	Tournament size:2
CPU running time (day)	90	114

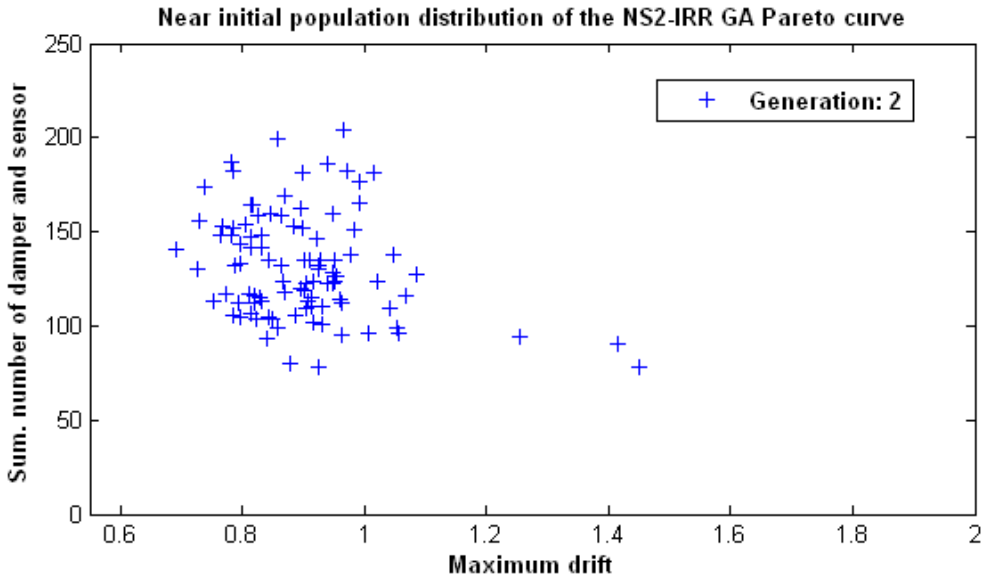

Figure 6.8. Initial population distribution of the NSII-IRR GA of 2-D 20-story benchmark problem

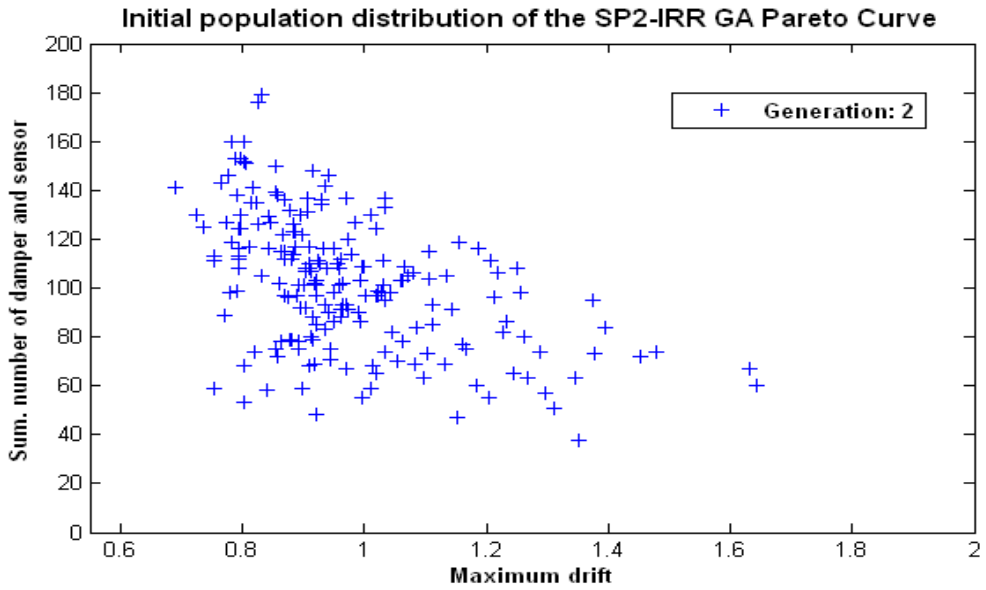

Figure 6.9. Initial population distribution of the SP2-IRR GA of 2-D 20-story benchmark problem

The archive size of the SP2-IRR GA and the parent population size of the NSII-IRR GA is different, then as shown in Figures 6.8 and 6.9, the SP2-IRR GA has more diverse and well-distributed near initial population than the NSII-IRR GA. Because of the big enough archive size of SP2-IRR GA, this optimization method can keep the diverse non-dominated individuals, but it also can be the reason of late convergence to optimal solutions as shown in Table 6.2. The running time of the SP2-IRR GA is a little bit larger than the NSII-IRR GA. Without the archive size, the same genetic properties are used to compare the performance of these two optimization methods.

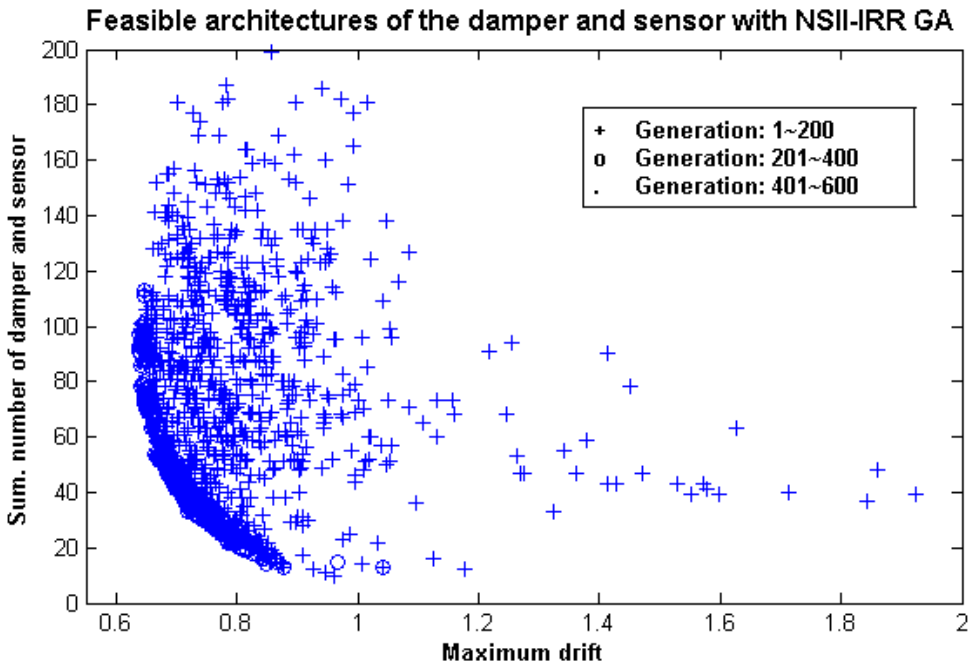

Figure 6.10. Feasible architectures of the damper and sensor using NSII-IRR GA of 2-D 20-story benchmark problem

The feasible architectures of the dampers and sensors with two optimization methods were suggested as shown in Figs. 6.10, and 6.11. The SP2-IRR GA has some not dense area in feasible domain because the only each 10^{th} generation archive set are plotted in the Figure 6.11. Each 10^{th} generation of the parent population was plotted, but the generations 2~20 also plotted in Figure 6.10. Thus the comparison of the two feasible, there is no big difference in feasible architectures of the dampers and sensors.

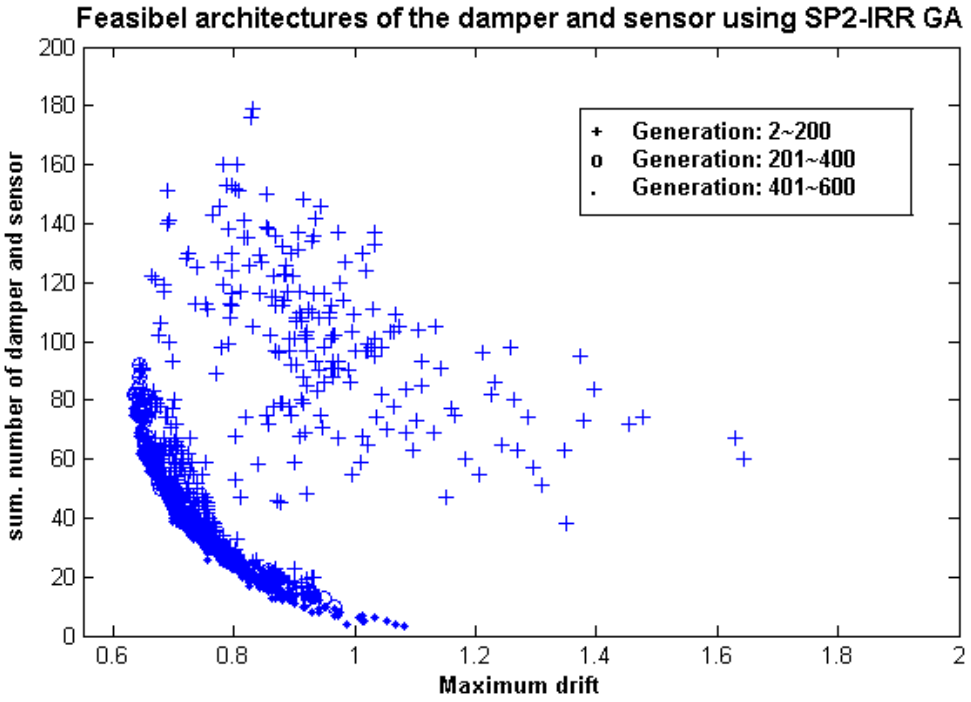

Figure 6.11. Feasible architectures of the dampers and sensor using SP2-IRR GA of 2-D 20-story benchmark problem

The near optimal Pareto fronts obtained using the NSII-IRR GA and SP2-IRR GA with generation 600 were suggested as shown in Figures 6.12, and 6.13. The diversity of the near-optimal front obtained using SP2-IRR GA is better than NSII-IRR

GA, but the running time is smaller with NSII-IRR GA than SP2-IRR GA. As shown in Figure 6.14, the comparison of the two methods, it is hard to determine that which optimization method has priority in finding near-optimal Pareto front. The Pareto front of SP2-IRR GA is better than NSII-IRR GA in area A, but NSII-IRR GA is better than in are B in Figure 6.14. Thus, in case of 3-D 20-story building problem, these two optimization methods are applied and the results also compared to each other in next paragraph.

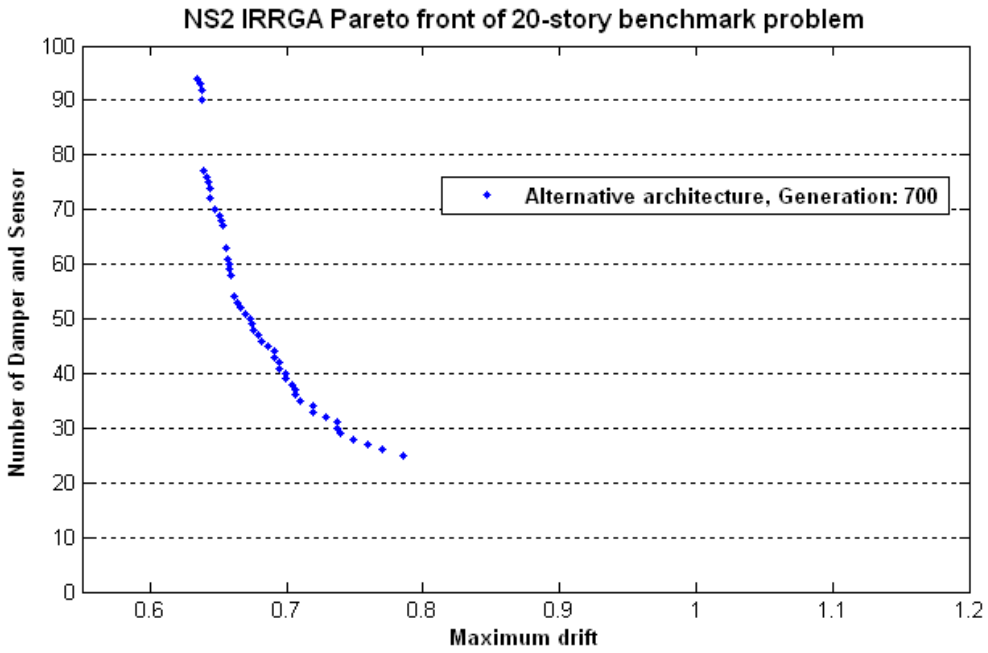

Figure 6.12. The Pareto front obtained using NSII-IRR GA of 2-D 20-story problem

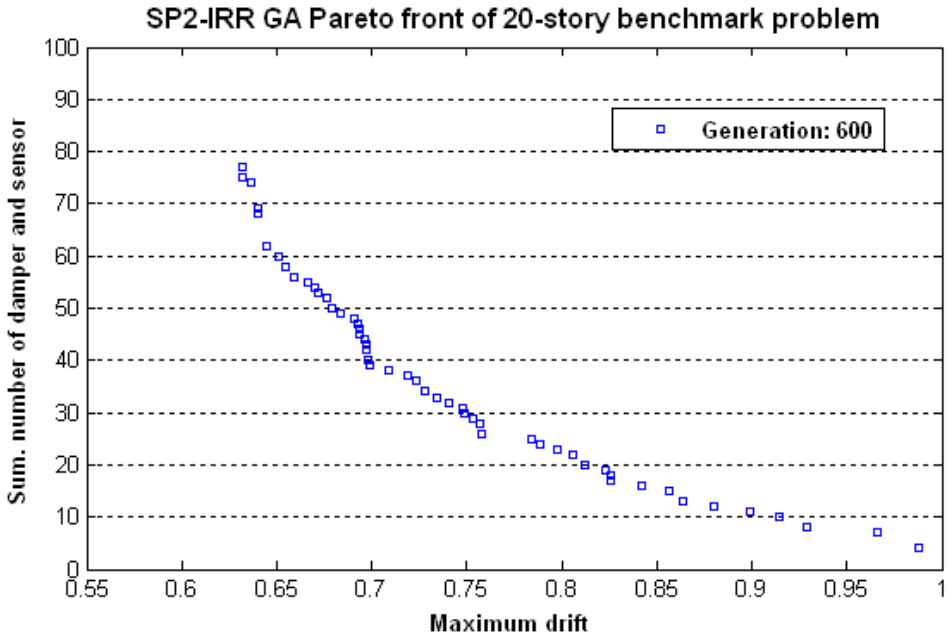

Figure 6.13. The Pareto front obtained using SP2-IRR GA of 2-D 20-story problem

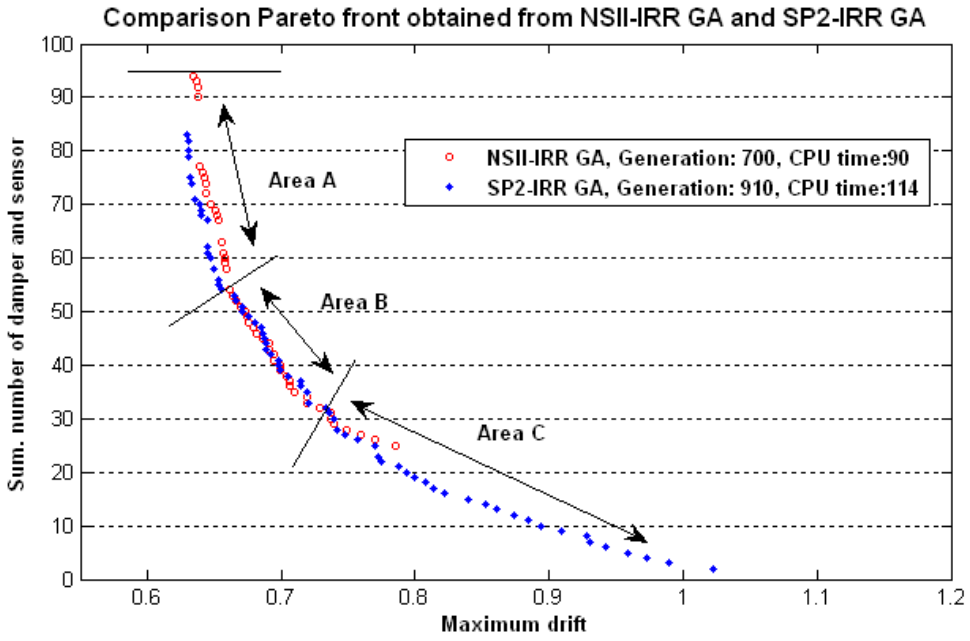

Figure 6.14. Comparison Pareto front obtained from NSII-IRR GA and SP2-IRR GA of 2-D 20-story problem

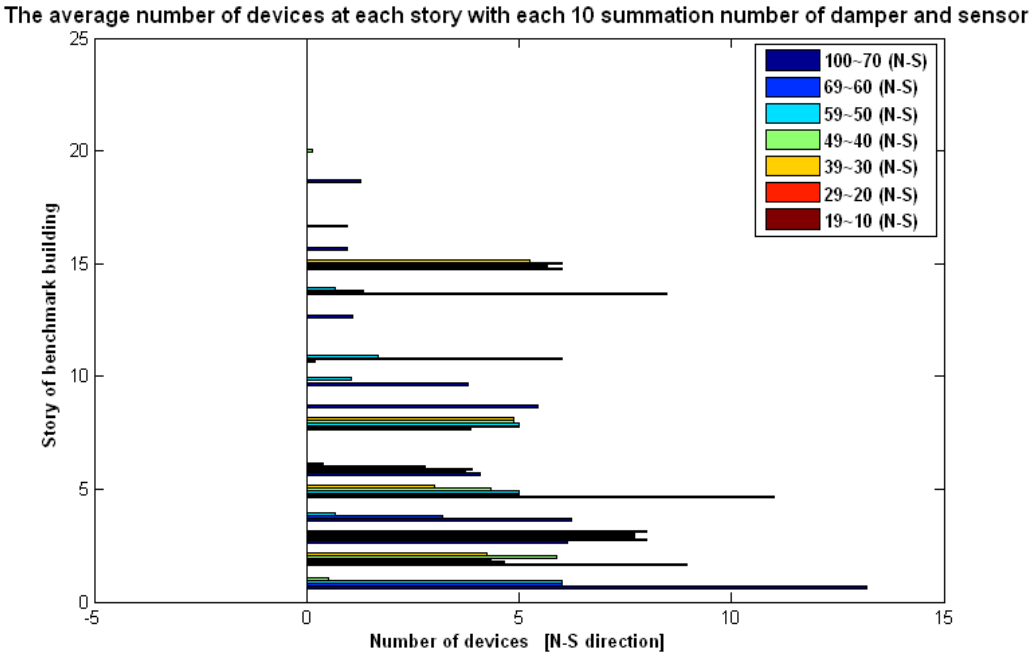

Figure 6.15. The average number of devices at each story using NSII-IRR GA of 2-D 20-story problem in all sections

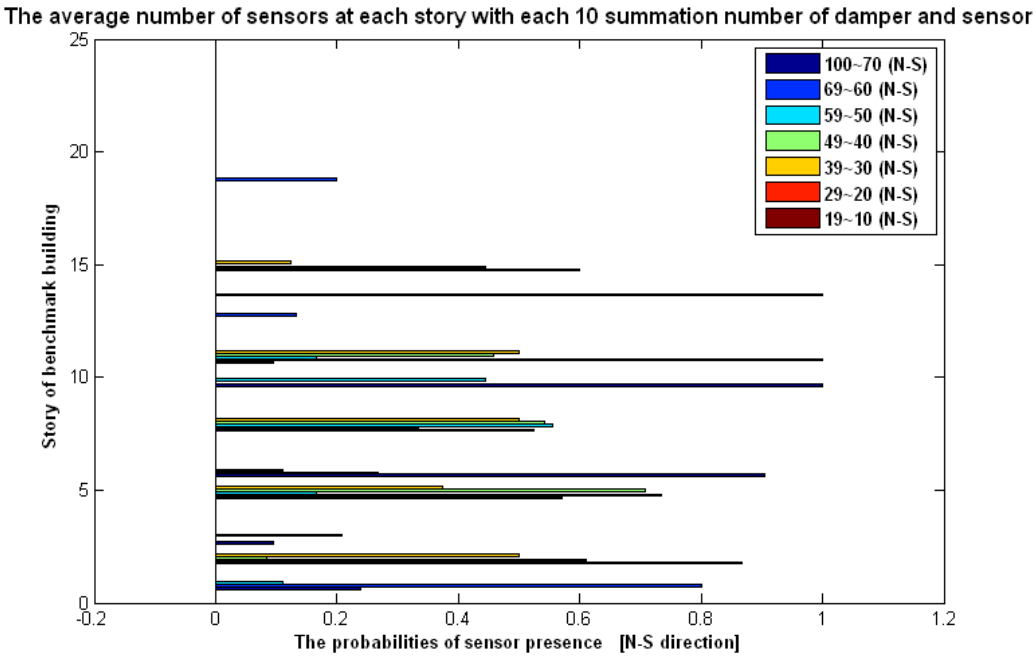

Figure 6.16. The probabilities of sensor presence at each story using NSII-IRR GA of 2-D 20-story problem in all sections

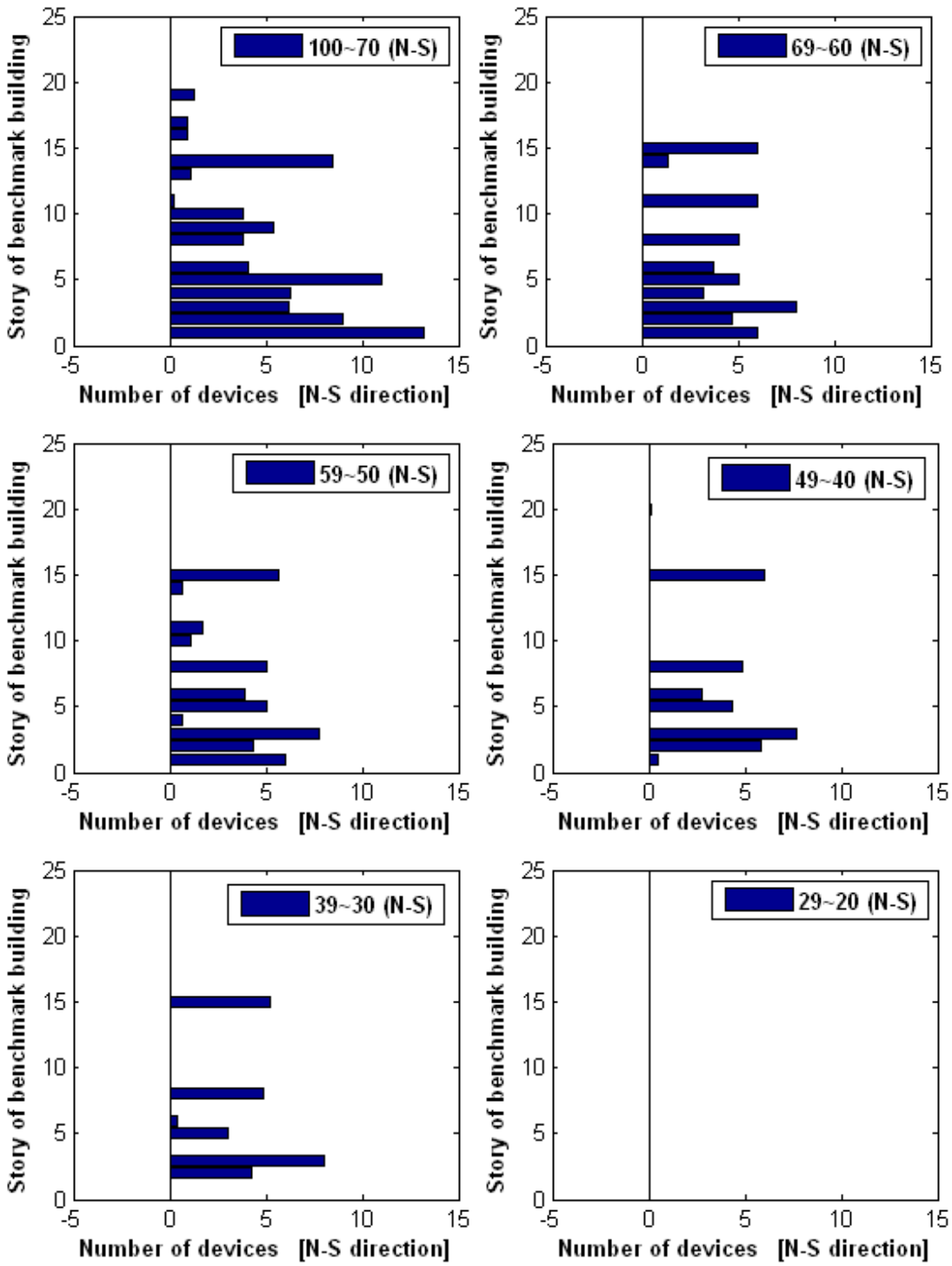

Figure 6.17. The average number of devices at each story using NSII-IRR GA of 2-D 20-story problem

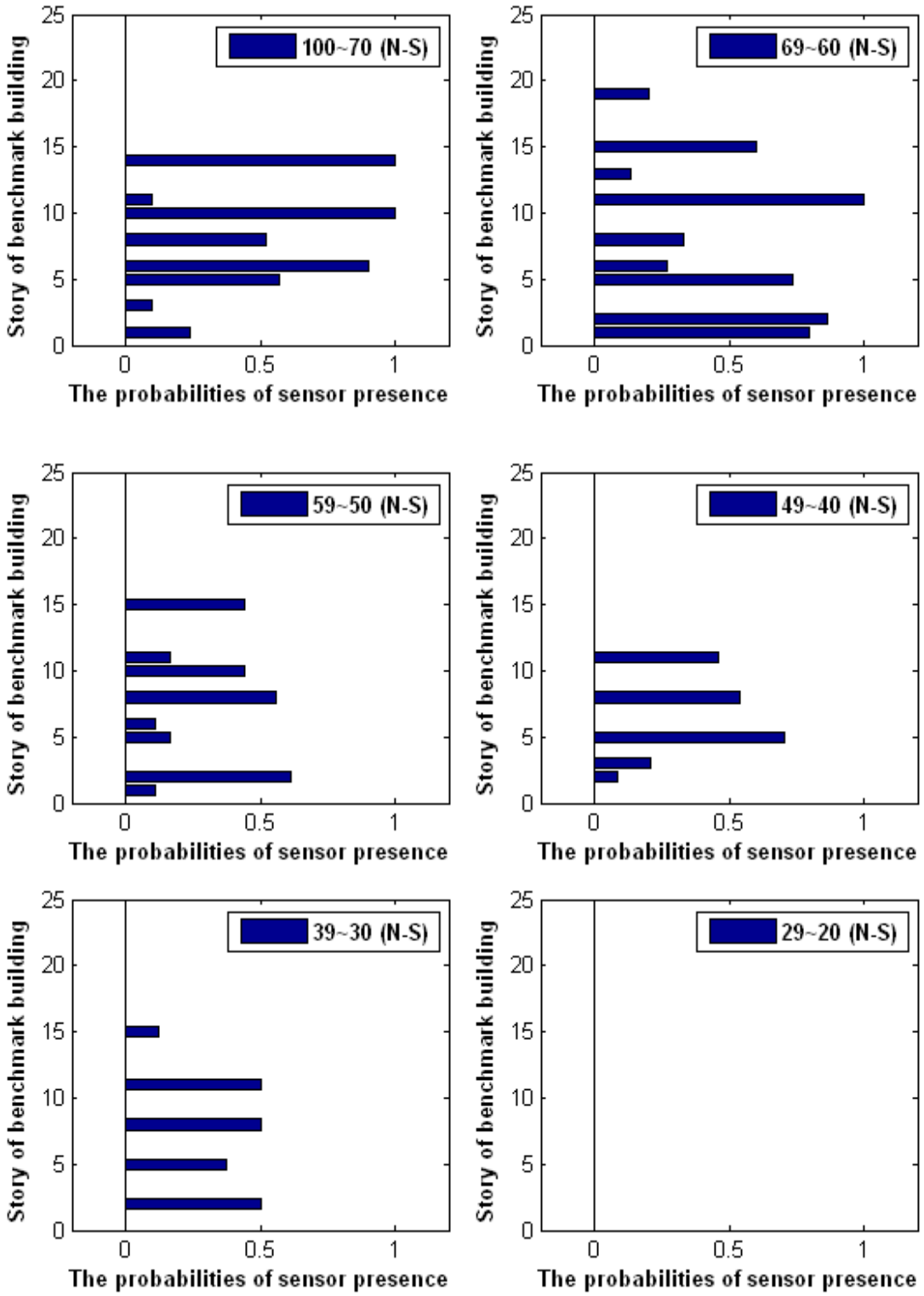

Figure 6.18. The probabilities of sensor at each story using NSII-IRR GA of 2-D 20-story problem

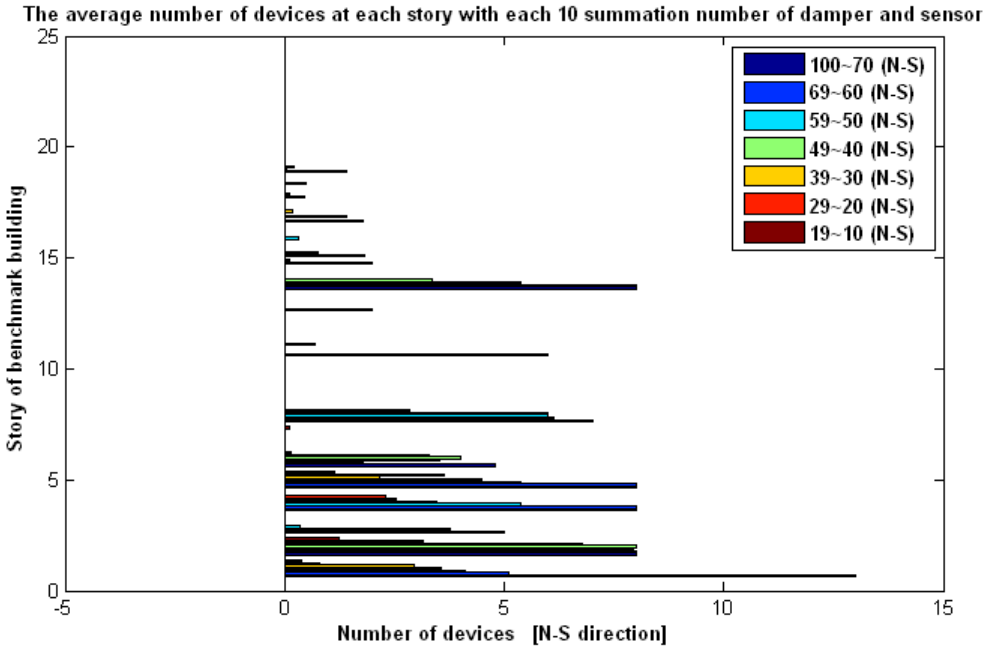

Figure 6. 19. The average number of devices at each story using SP2-IRR GA of 2-D 20-story problem in all sections

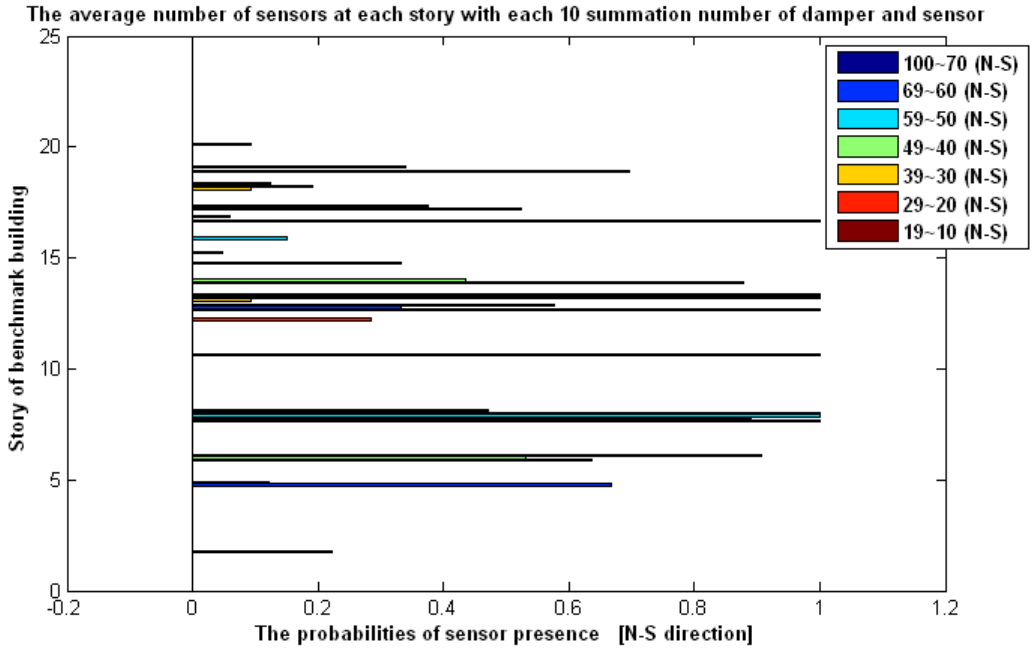

Figure 6.20. The probabilities of sensor presence at each story using SP2-IRR GA of 2-D 20-story problem in all sections

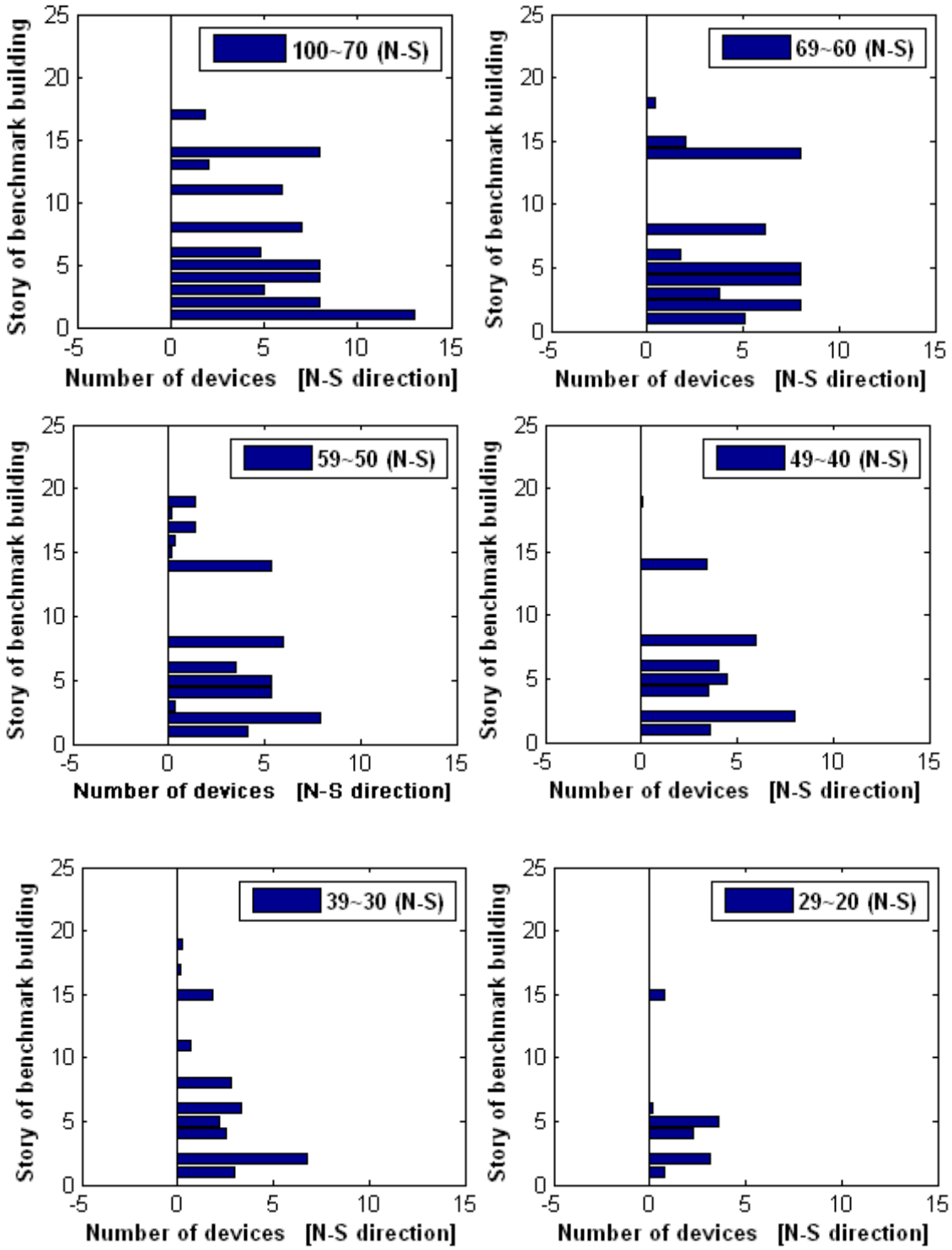

Figure 6.21. The average number of devices at each story using SP2-IRR GA of 2-D 20-story problem

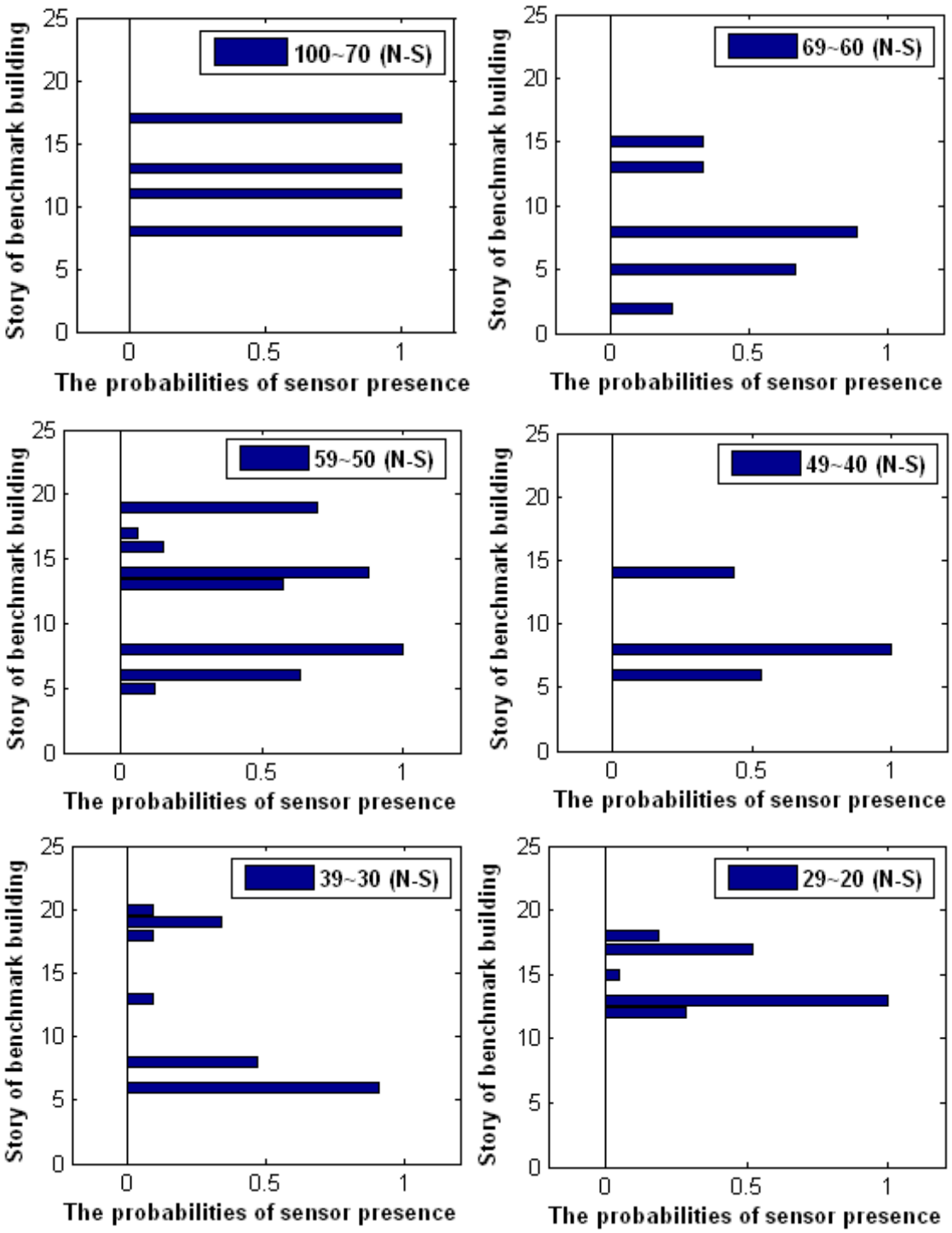

Figure 6.22. The probabilities of sensor at each story using SP2-IRR GA of 2-D 20-story problem

The two defined optimization method NSII-, and SP2-IRR GA are worked really good in the 2-D 20-story model even though the CPU running times are huge as nearly 3 months to get a fully converged final Pareto fronts. From the detail analyzed data of the two methods such as Figure from 6.15 to 6.22. Figure 6.13 is expressed with each 10 summation number of control devices and sensors in Figure from 6.15 to 6.18, and Figure 6.14 is also expressed same way. There are not much similarity in the poisons of the dampers and sensors in each story. Although there are some general pattern in the damper locations and numbers in each story such as 5th, and 15th floor for the control devices locations and 3rd, 5th, 7th, and 12th floors for the sensor locations about the Figure 6.13, but it is hard to say that the poisons and numbers of dampers and sensors are possible to be patterned. Thus, If the optimization software or these graphs are offered to the structural engineers, the structural engineers can choose possible locations and numbers of the control devices and sensor locations.

6.5 The Optimal Results of the 3-D 20-story Benchmark Problem

From the previous study about the 2-D 3-, and 20-story building cases with the defined NSII- and SP2 IRR GA, the genetic properties are redefined to consider the complexity of 3-D 20-story building case, and then 3-D 20-story building problems are investigated to find the optimal Pareto front of the architectures of the control devices and sensors using two suggested optimization method. The genetic properties are shown in Table 6.3.

Table 6.3. Comparison of genetic properties of NSII-IRR GA and SP2-IRR GA for 3-D 20-story problem

	NSII-IRR GA	*SP2-IRR GA*
String length	1000	1000
Population size	200	200, 300
Mutation rate	0.01	0.01
Crossover rate	0.9	0.9
Generation	118	118
Tournament Selection	Tournament size 2	Tournament size 2
CPU running time (day)	91	100

The Pareto fronts obtained with limited generation and CPU running time are suggested as shown in Figures. 6.23 and 6.24. From the comparison of two Pareto fronts, the NS2-IRR GA has better performance in finding the optimal architectures of placement of dampers and sensors with smaller CPU running time and iterations as shown in Figure 6.23. With the diverse optimal candidates, finally the SP2-IRR GA would find the better results than the NSII-IRR GA, the CPU running time for the 1st generation is nearly one day. This huge CPU running time makes it difficult to find the final optimal Pareto fronts using the suggested NSII-IRR GA and SP2-IRR GA with limited time. Thus, a more efficient or better optimization method is required to finish this 3-D 20-story benchmark problem. In the next section, the advanced optimization method is developed and applied to the 3-D 20-story building control devices and sensor architecture problem.

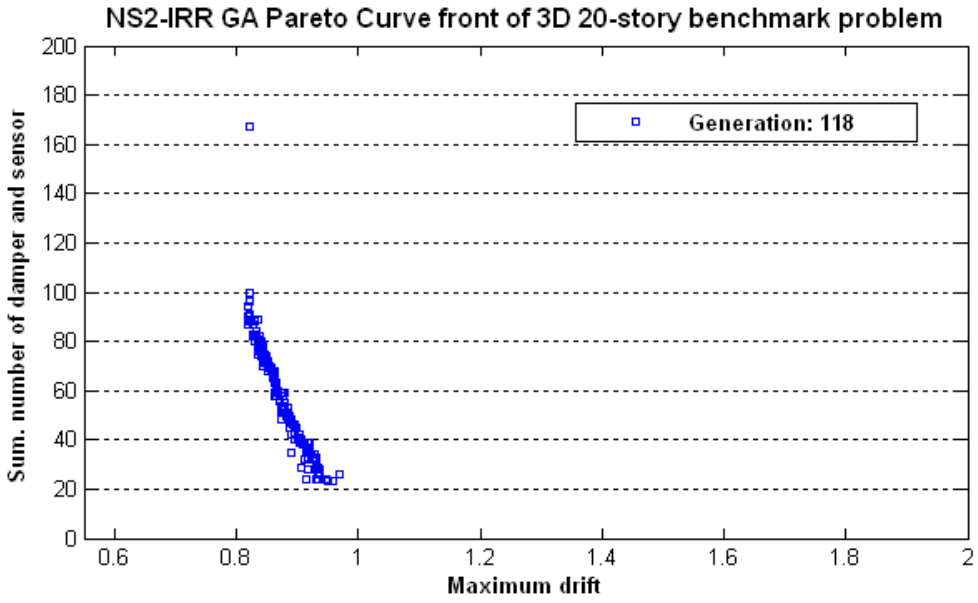

Figure 6.23. The Pareto front obtained using NS2-IRR GA of 3-D 20-story problem

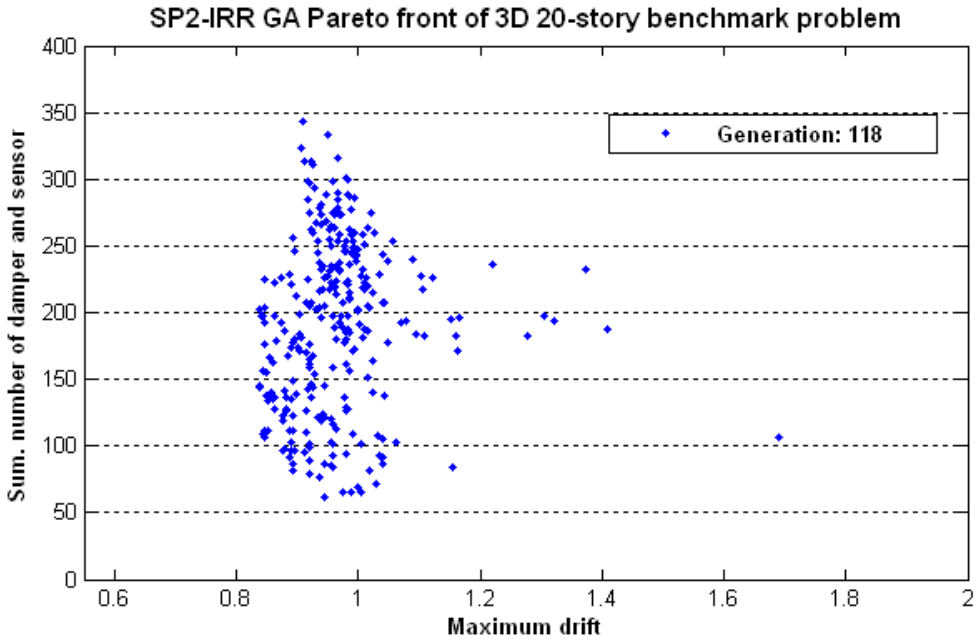

Figure 6.24. The Pareto front obtained using SP2-IRR GA of 3-D 20-story problem

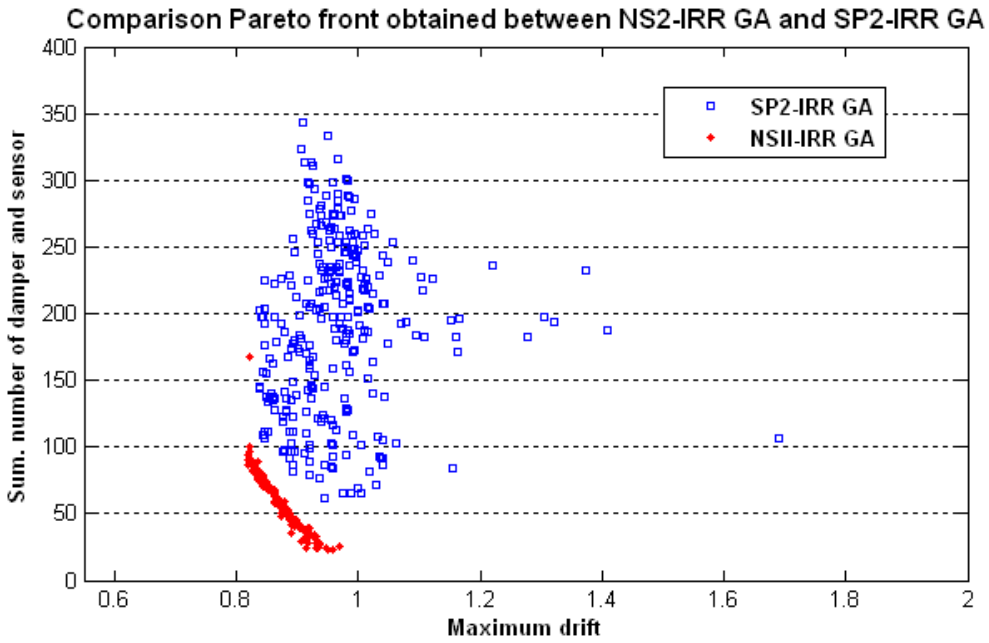

Figure 6.25. The Pareto fronts obtained using NS2- and SP2-IRR GA of 3-D 20-story problem

7 GENE MANIPULATION GENETIC ALGORITHMS

7.1 Introduction

NSII-IRR GA based on the non-dominated sorting genetic algorithm-II (NSGA-II) and implicit redundant representation genetic algorithm (IRR GA) and the SP2-IRR GA based on the strength Pareto evolutionary algorithm2 (SPEA2) and IRR GA were defined to perform finding the optimal architectures of the placement of dampers and sensors in the previous section. These two optimization methods showed that they have are able to find the optimal Pareto front with two objectives in the 2 dimensional (2-D) 3- and 20-story American Society of Civil Engineering (ASCE) control benchmark building problems. However, the central processing unit (CPU) running times of the two suggested methods are too long and they did not demonstrate good performance in 3-D 20-story building problem with limited time.

To date, the more advanced optimization is required because the required optimization problem will be complicated by the development of technological developments of science and engineering. Furthermore most engineering problems such as structural dynamic analysis or nonlinear behavior of the whole structure model problem generally require large CPU running time. Although the NSII-IRR and SP2-IRR GA are still very competitive methods, the new and more highly-advanced optimization method is required in the 3-D 20-story problem. From the previous trial, we

know that the 3-D 20-story optimization in installing control devices and sensors is a highly complex non-continuous and non-patternable problem. Thus, the new advanced genetic algorithm is suggested to investigate these problems in this section.

7.2 Gene Manipulation Genetic Algorithms

In real world systems, new offspring are made by genetic operations such as crossover, mutation, and selection. The simple genetic algorithm (SGA) uses this genetic operation mechanism to solve complex problems which are hard to establish by exact mathematical equation. Many GAs are developed based on SGA. For example, Hajelar and Lin's genetic algorithm (HLGA) (Hajela and Lin 1992), two level genetic algorithm (TLGA) (Li et al, 2004), SPEA (Zitzler and Thiele 1999), niched Pareto genetic algorithm (NPGA) (Horn and Nafpliotis 1993), NSGA (Srinivas and Deb 1994), NSGA-II (Deb at al. 2000), and SPEA2 (Zitzler et al. 2001) used the same mechanism and encoding policy, but were different in selection steps. The SGA mechanism is a kind of passive search method: randomly generate the population, evaluate the population, select good individuals, do crossover and do mutation. The main concept in SGA mechanism is that the random numbers are used in making a new population for the next generation without using engineering judgment. In crossover and mutation, the crossover point and mutation bit are selected randomly without any consideration of problem characters, even though the crossover and mutation probabilities are defined by the user. Thus, to apply engineering judgment in generating new mating, the gene manipulation concept is adapted to SGA and IRR GA mechanisms.

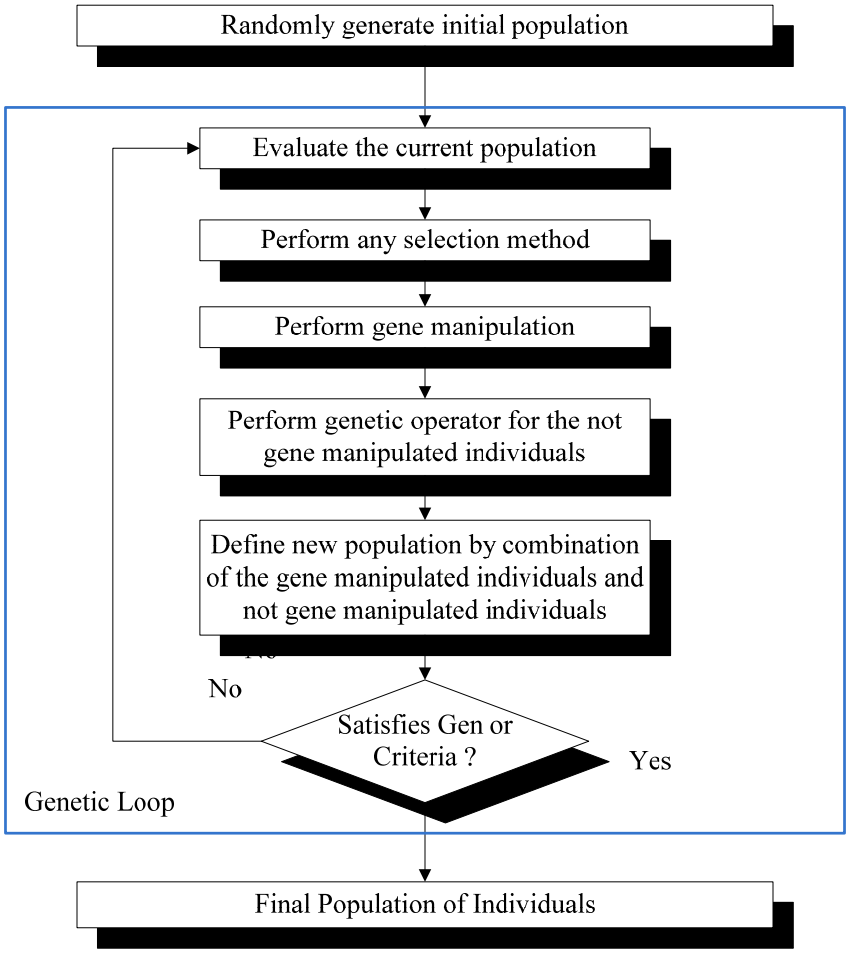

Figure 7.1. Main flowchart of GMGA

As we know, gene manipulation in genetic engineering is the recombination of deoxyribonucleic acid (DNA) technology. DNA contains the genetic instructions used in the development and functioning of all known living organisms. An organism can transfer its information through DNA. DNA gives directions on how to make an organism's offspring. The engineering judgment can be the role of DNA in gene manipulation genetic algorithms (GMGA). By adaption, the engineering judgment in making new individuals using best offspring such as non-dominated individuals, this

new genetic algorithm will find optimal Pareto front or optimal solutions more effectively. The main procedure of the GMGA is shown in Figure 7.1.

As shown in Figure 7.1, for the first step, the GMGA is started by randomly generating or initializing the population. This current population is evaluated by the fitness function, and then any selection method such as SPEA, NSGA, SPEA2 and NSGA-II can be carried out for the amount of the number defined by the Equation (7.1). This is a strong advantage of the GMGA. For the next step, the genetic operator is carried out about these selected individuals. The non-dominated sorting is performed in the current population to create gene manipulated individuals with a defined number by Equation (7.2).

$$\text{Selected number of individuals} = (1 - \text{GMR}) \times \text{current population size} \quad (7.1)$$

$$\text{The number of gene manipulated individuals} = \text{GMR} \times \text{current population size} \quad (7.2)$$

where

GMR : gene manipulation rate about the current population size

The GMR will be defined by considering the mating pool size and the characters of the problem, and usually the range can be 0.1~0.4. The crossover and mutation will be carried out about the selected number of individuals defined by Equation (7.1). The next step is the gene manipulation. The detailed gene manipulation procedure is as follows:

1. Select best fitted individuals defined by the GMR in single objective problem or non-dominated individuals in multi-objective problem
2. Determine the number of new individuals in the Pareto front
3. Make new gene instances to make new individuals

4. Insert new gene instances to current best individuals defined by GMR or non-dominated individuals
5. Make a new child population by adding new gene manipulated individuals those selected, crossed-over and mutated by the generic genetic operator.

7.2.1 Determination of the Number of New Individuals in the Pareto Front

In step 2 of the previous paragraph, the number of new individuals in the best individuals defined by GMR or non-dominated Pareto front is determined by the following Equation (7.3):

$$\text{Number of new string (i)} = \text{round}\left(\frac{\text{Each distance (i)}}{\text{Total distance}} \times \text{Pop. Size} \times \text{GMR} \right) \tag{7.3}$$

As shown in Figure 7.2 and Equation (7.3), the number of new string is dependent on the Euclidean distance of the nearest two non-dominated points or two best individual points. If the distance of the nearest two non-dominated points is big, the largest number of new strings will be assigned to find and fill the Pareto front between two non-dominated or two best individual points. From this mechanism, we can expect that the non-dominated Pareto front will be evenly found without losing any non-dominated points. From this example, we know that the new artificial individuals will not be created in all the sections of the non-dominated front or best individuals, but created between two points which have Euclidean long distance sections of the Pareto front.

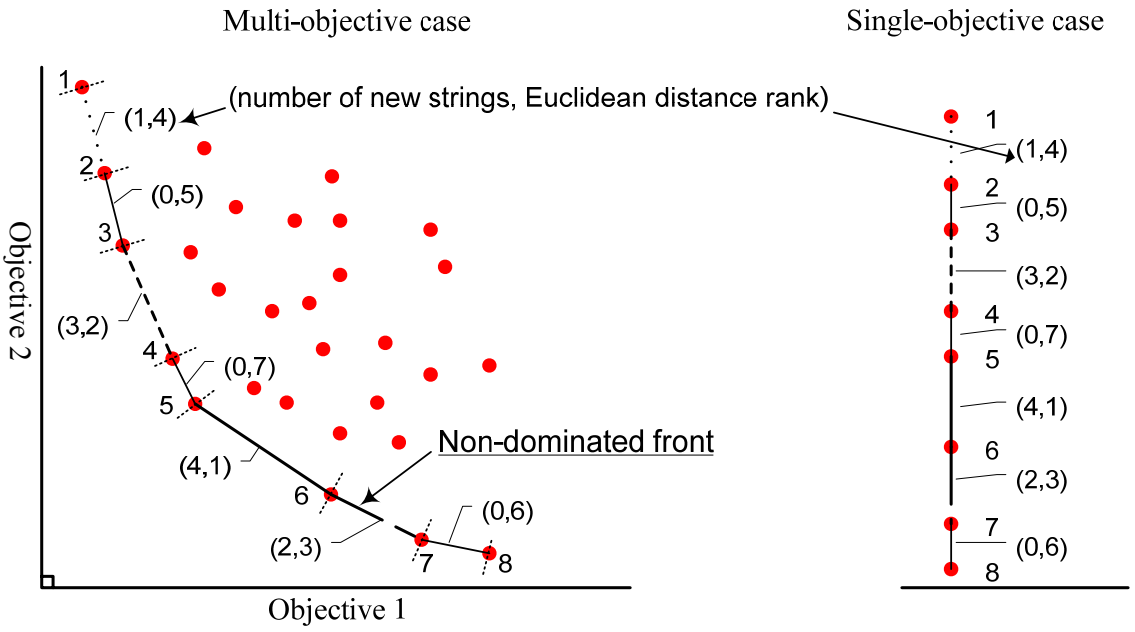

Figure 7.2. Determination of the number of new strings between two non-dominated or two best individual points

7.2.2 Creation New Gene Instances for the New Individuals

The number of new strings is defined between each two points in the non-dominated front. The gene instances for the new individuals will be created by the gene manipulation mechanism. The mechanisms to generate new gene instances are different and dependent on the encoding policies such as SGA and IRR GA. The new gene instance creation rule is simple. When there are two non-dominated individuals like points 5 and 6 in Figure 7.2, the two individuals should be decoded, the optimization parameter values should be compared with each other, and new parameter values should

then be determined by the new parameter creation rules. The mechanism to create new parameter value is:

1. Mean value.
2. Maximum value.
3. Minimum value.
4. Random value.

For example, if there are n parameters, the j objectives, and the defined number of new string is k from the Equation (7.3), the new parameter value for the first new string can be determined as:

$$P_n(k,n) = \frac{1}{j}\left(\sum_{n=1}^{j} P_c(k,n)\right) \tag{7.4}$$

where

$P_n(k,n)$: new parameter value for the k-th new string and n-th parameter
$P_c(k,n)$: current parameter value for the k-th new string and n-th parameter

The new parameter value for the second new string can be determined as:

$$P_n(k,n) = \max_{i=1}^{j}(P_c(k,i)) \tag{7.5}$$

The new parameter value for the third new string can be determined as:

$$P_n(k,n) = \min_{i=1}^{j}(P_c(k,i)) \tag{7.6}$$

The new parameter value for the fourth or later new string can be determined as:

$$P_n(k,n) = \operatorname{rand}_{i=1}^{n}(P_c(k,i)) \tag{7.7}$$

where the $P_n(k,n)$ is randomly generated between max parameter $P_c(k,n)$ and min parameter $P_c(k,n)$ of j objectives. These processes will be repeated until all the gene instances in all sections of non-dominated front are made.

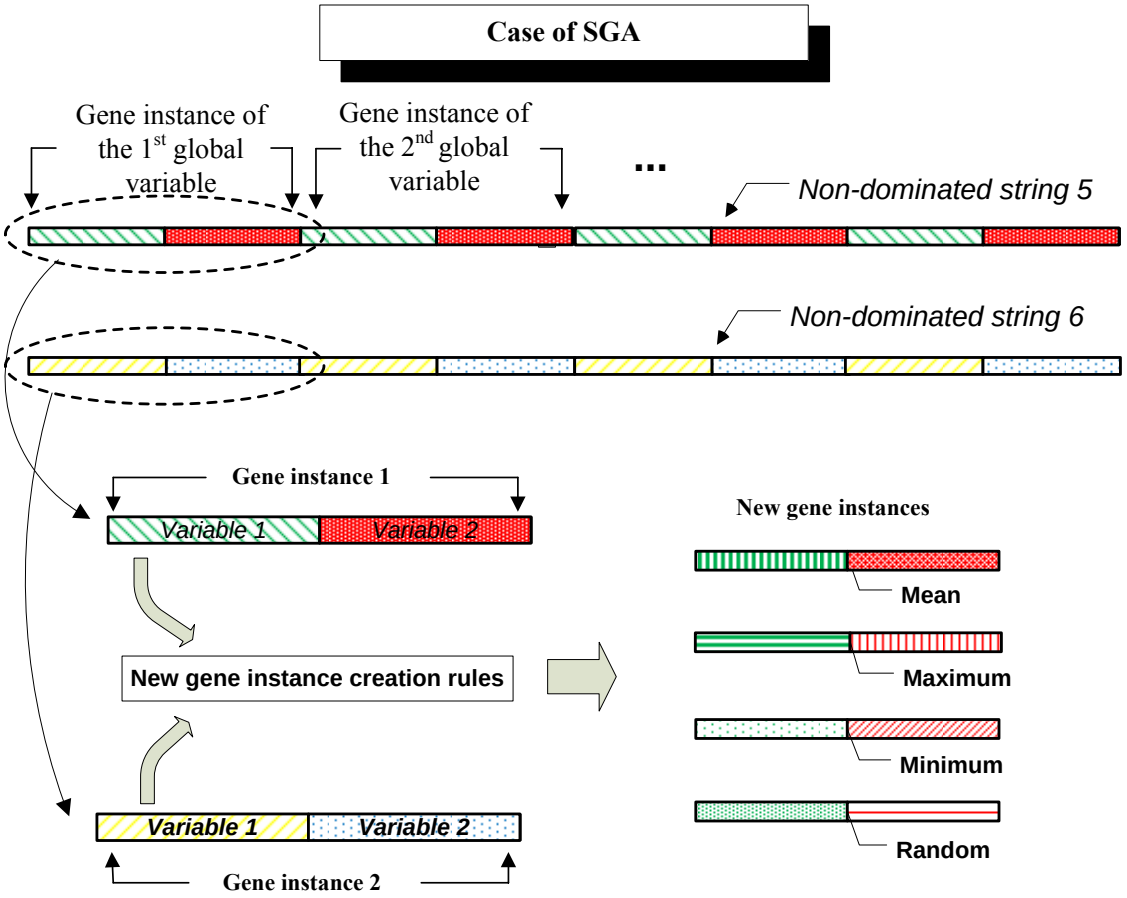

Figure 7.3. Mechanism of creation of new gene instance using SGA

This mechanism is dependent on the engineering judgment as shown in Figure 7.3. If the number of new string is defined as '4' from the Equation (7.3) in between two non-dominated or best individual points, the new four gene instances are generated by

the creation mechanism as shown in Figures 7.3 and 7.4. In the case of SGA, the two parameter variables can be directly compared, and then the new gene instances are created with followed by the gene instance creation mechanism. However, in case of IRR GA, the positions of GL (GL) are needed to be memorized, and the flags are also needed to be compared to determine new parameter values which are used to create new gene instances, because there are special cases in which the two gene instances have the same parameter information.

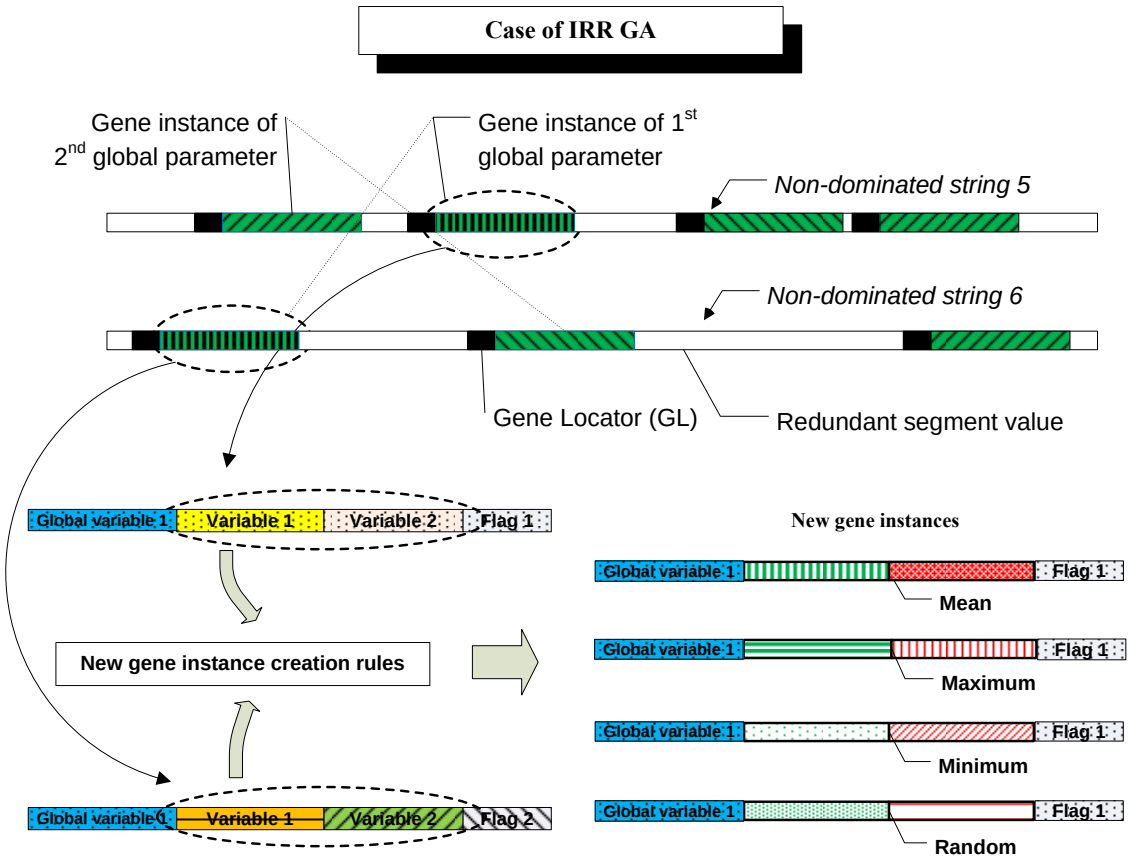

Figure 7.4. Mechanism of creation of new gene instance using IRR GA

For example, from the Figure 7.4, non-dominated string 5 has two gene instances for the 2nd global parameter. Thus, the GL position should be memorized and the flag values of the two same gene instances are should be compared, and the bigger flag valued gene instance is selected to create new gene instance, because the biggest flag value gene instance is used for the control analysis parameter. The Figure 7.3 and 7.4 are only expressed for the 1st global parameter. Thus the other left global parameter also needs to be compared and then the new gene instances are created by following the creation mechanism.

7.2.3 Insertion New Gene Instances to Current Non-dominated Individuals

All the new gene instances are created by using the current non-dominated Pareto front or best individuals by the creation mechanism. These new gene instances should be inserted into the non-dominated or best individuals. In the case of GMGA based on SGA, the new instances are inserted into one of the current best individuals as shown in Figure 7.5. This case is only for the 1st global parameter. This insertion procedure will be continued until the insertion about the other global parameters is finished. For example, the GMGA is applied to 3-story control benchmark building case, and the 1st global parameter is the first story and the local variable 1 and 2 are each the number of the damper and the sensor's absence/presence.

In the case of GMGA based on IRR GA, at fist, select the non-dominated individual which has the larger number of GL between 2 individuals because it is easier to express diverse case of new gene instances in the individual that has many GL. For

example, when the non-dominated string 6 in Figure 7.4 does not have the third global parameter gene instance, but non-dominated string 5 has the global parameter gene instance, the non-dominated string 5 is selected as the base string for the new individual because when base string do not have enough number of GL it is impossible to insert new created gene instance for the last global parameter.

Even though the new gene instance is inserted in the non-dominated string 6 by making new GL artificially, this trial did not show good performance because the building block was collapsed by the artificial insertion of new GL. Thus, the non-dominated string 5 is selected from the Figure 7.6. Then, the locations of the GL of the same global parameter gene instances can be found. In case of Figure 7.6, the non-dominated string 5 has gene instances expressing the 2^{nd} global parameter, thereafter, the two flag values of the gene instances that express 2^{nd} global parameter are needed to be compared, and the bigger flag valued gene instance is selected for the insertion of the new created gene instance, because the bigger flag valued gene instance will be used in the next generation. As already mentioned, the first global parameter is the 1^{st} story of the case of the control benchmark building case and the variable 1 and 2 express the number of dampers and sensor's presence/absence.

The remnant GLs which are not inserted in the new gene instance such as the overlapped 2^{nd} global parameter in non-dominated string 5 is used for the other remnant global parameter with the biggest flag to be used in the next generation. For example, if the re is 5 global parameters, and the selected non-dominated string has 5 GLs, but one global parameter has two gene instances that express the same global parameter.

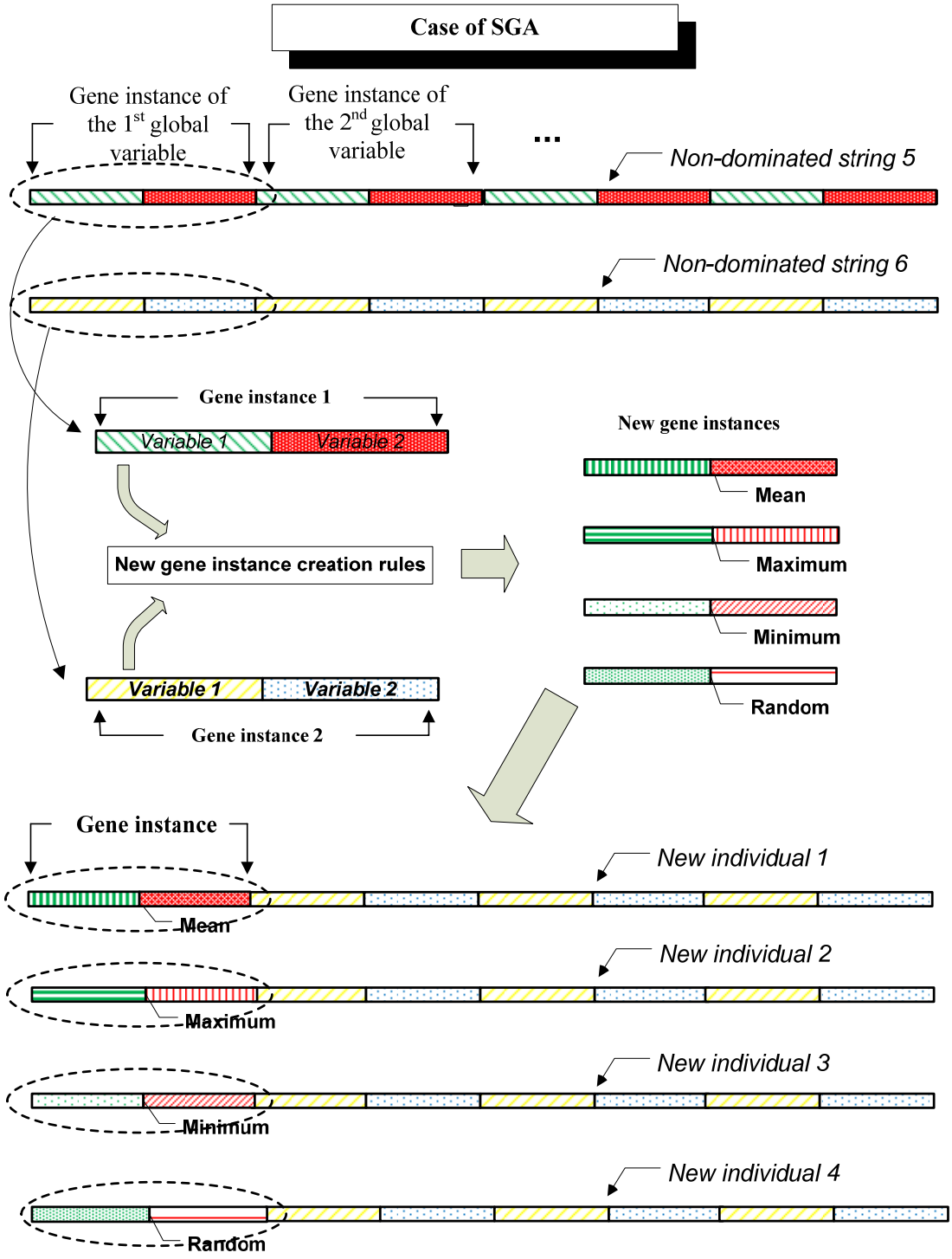

Figure 7. 5. Insertion of new gene instances to current non-dominated individuals in SGA

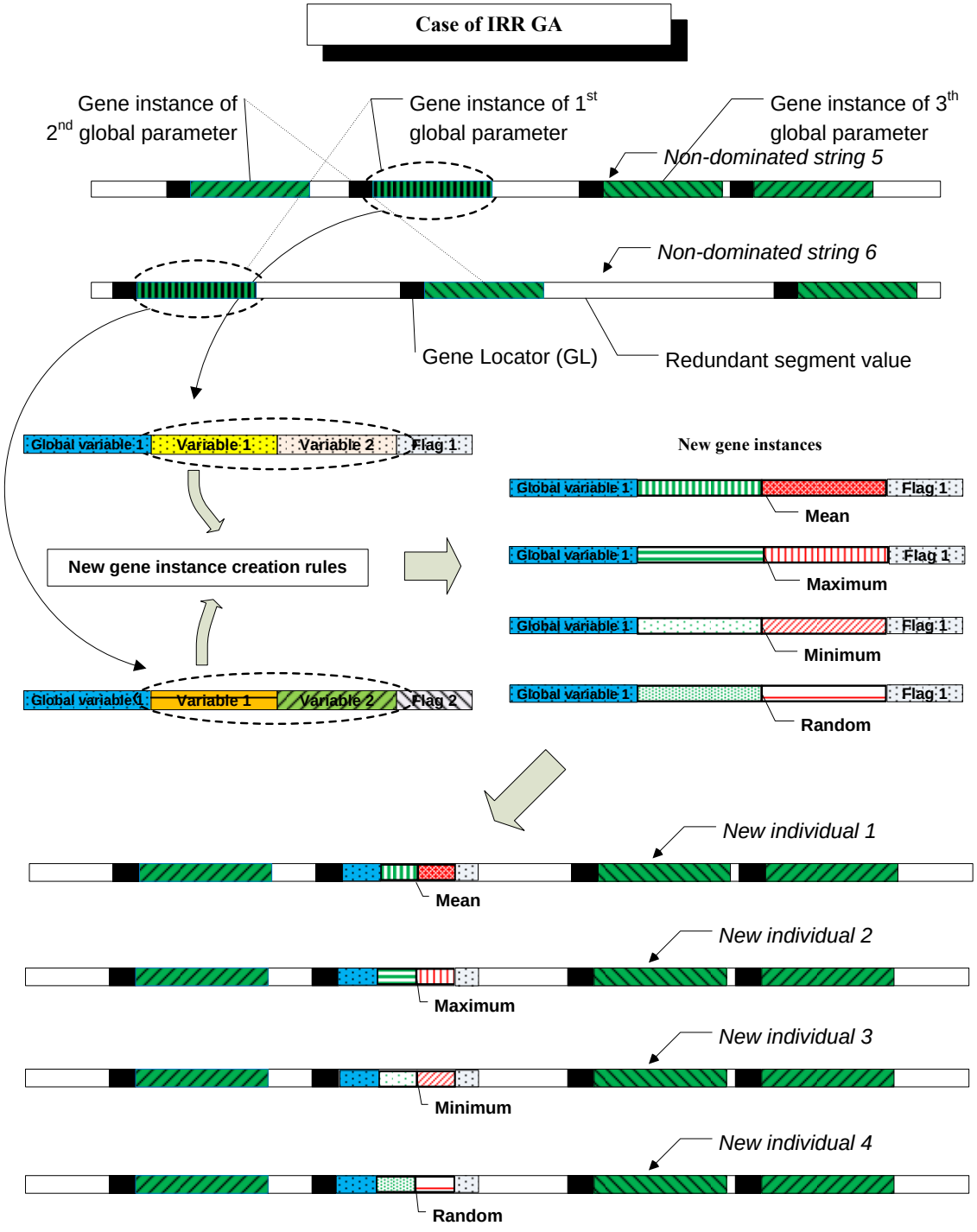

Figure 7.6. Insertion of new gene instances to current non-dominated individuals in IRR GA

One of the parameters did not express in one of the two individuals. The overlapped but not selected GL for the gene insertion is used for the one parameter that is not expressed in individual with biggest flag value.

When the new strings with the defined number by the Equation (7.2) are created by the creation mechanism, these created strings should be kept and then the other individuals for the next generation from the Equation (7.1), are needed to be added to gene manipulated individuals, and then the final child population is defined by following equation:

$$C_{new} = C_{GM} + C_{Tradition} \tag{7.8}$$

7.3 Comparisons of GMGA with Traditional Multi-objective GA

GMGA, proposed new approach of multi-objective optimization, adopts the advantages of suggested multi-objective GA mechanism and new gene instance creation mechanism. The characteristics of the GMGA are;

- Possibility to apply any pre-suggested multi-objective optimization mechanism such as SPEA, SPEA2, NSGA, and NSGA-II.
- Available to the single objective optimization problem.
- Diverse encoding policy of GA such as SGA and IRR GA are also available.
- The defined portion of the next child population is generated by the new creation rule which is not a traditional passive genetic operator such as crossover and mutation.

- This new individual creation rule suggests the foundation of active method to generate the child population.
- The engineering judgment is applied to make a child population by the new creation rule.
- The creation rule is also very flexible to dependent on the optimization problem characters.

7.4 Performance of Gene Manipulation Genetic Algorithm

The potential advantages of the proposed GMGA over generic GAs, including the multi-objective GA methods are:

- The engineering judgment concepts based on the non-dominated individuals in the multi-objective genetic algorithm (MOGA) or superior strings in single objective GA reduce the fundamental running time of the optimization process.
- GMGA offers the broad adoption possibilities of the traditional GA mechanisms to the user.
- Enhanced optimization performance.
- The possibility of pre-convergence to local maximum caused by the using only non-dominated or superior individuals will be reduced due to systemically organized creation rule.
- By dynamically changing the GMR, powerful performance is expected.

- The insertion of new gene instances in the existing GL position does not disrupt building blocks.
- Without crossover and mutation, the new individuals respect the full engineering judgment.

7.5 Performances of GMGA in 2-D 3- and 20-story Control Benchmark Buildings

The new optimization methods are proposed to find the optimal Pareto fronts about the 3-, and 20-story building benchmark problems in the previous paragraphs. To test the performances of the developed optimization method, for the first step, GMGA is applied to 3-story benchmark problem. The detailed process of the optimization is shown in Figure 7.7. The global parameters in GMGA are the system's stories, and the local variables are the number of control devices and the absence/presence of sensors. The GA properties for the 2-D 3-story building problem of the GMGA are shown in Table 7.1.

Table 7.1. The properties of the GMGA for the 2-D 3-story building problem

	SP2-IRR GA based	*NSII-IRR GA based*
String length	54	54
Population size	40, 40	40, 40
Mutation probability	0.01	0.01
Crossover probability	0.9	0.9
Generation	35	32
Tournament Selection	Tournament size: 2	Tournament size: 2
CPU running time	5 hr 46 min	5 hr 10 min

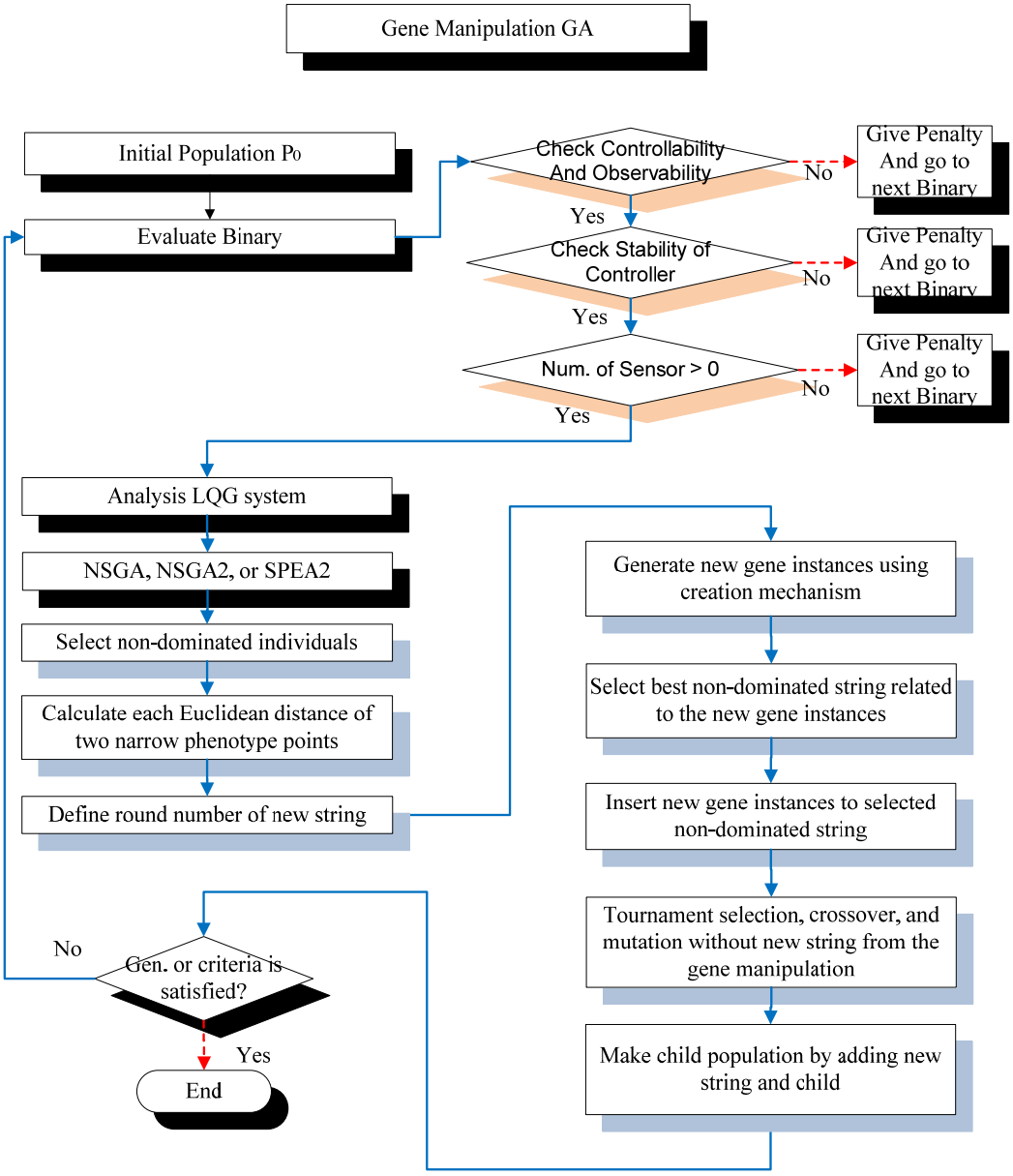

Figure 7.7. Flowchart of the GMGA to optimize the control device architectures.

With adoption of the GMGA based on the NSII-IRR GA, the Pareto front for the 3-story benchmark problem is obtained as shown in Figure 7.8. The crossover rate is 0.9, mutation rate is 0.01, the population size is 40, and string length is 54. For the GL, the

continuous three ‘1’ bits are used. The Pareto front is well distributed in a feasible area. The CPU running time is 5 hours 10 minutes. The GMGA based on the SP2-IRR GA shows that it converges to the local minima at specific points. With several parametric regulations, the GMGA based on SP2-IRR GA did not perform well in finding the non-dominated Pareto front in the 3-story problem rather than GMGA based on NSII-IRR GA. One of the reasons can be as shown in the performance of the SPEA2 in the same problem in the previous section, that SPEA2 mechanism does work well in a small domain of the optimization problem.

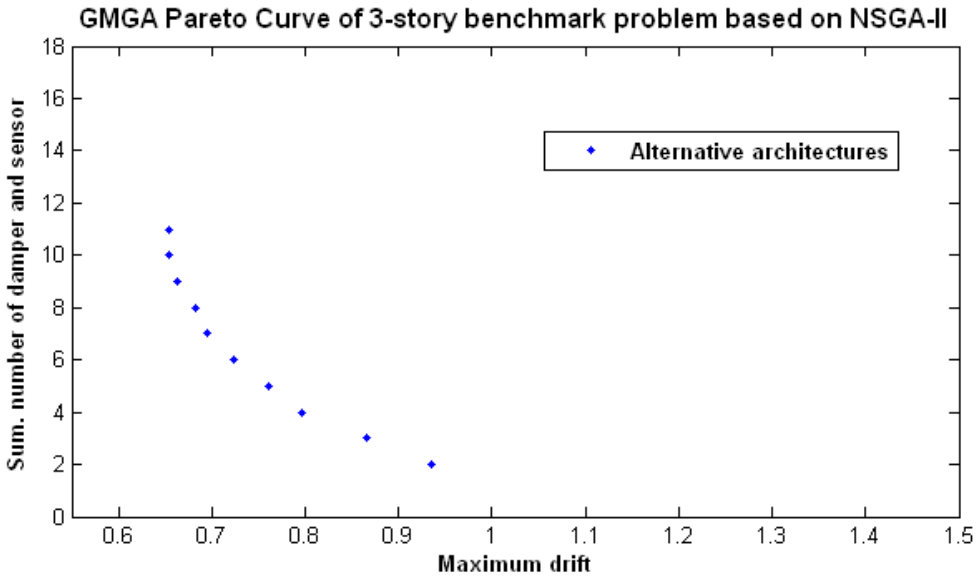

Figure 7. 8. The Pareto front obtained using GMGA of 3-story problem based on NSII-IRR GA

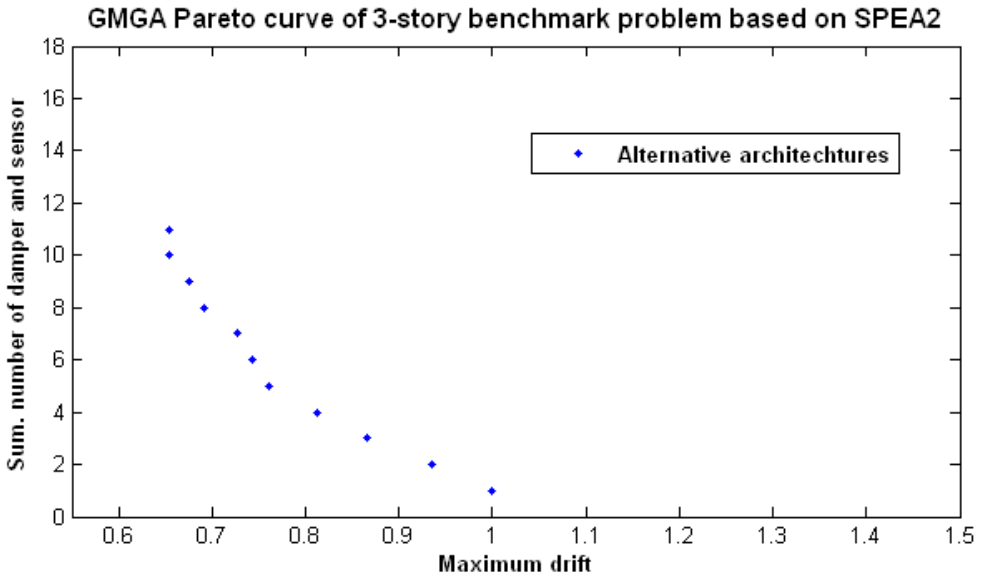

Figure 7.9. The Pareto front obtained using GMGA of 3-story problem based on SP2-IRR GA

The properties of the GMGA for the 20-story control benchmark building problem are shown in Table 7.2. The same values of string length, population size, mutation and crossover probability, and tournament selection size are used from the previously proposed NSII-IRR GA and SP2-IRR GA to compare GMGA's performance. 0.1 is also used for the GMR value. As we know, the GMGA has broad applicability in adoption of the multi-objective optimization selection methods. Thus the most current advanced MOGAs; SPEA2 and NSGA-II, are adopted to the GMGA. The performances of the two proposed GMGA are very good in reduction of CPU running time and search performance in finding optimal Pareto front as shown in Figures 7.8 and 7.9. The final results of the suggested two GMGA based on NSII- and SP2-IRR GA are a little bit different. From the Figures 7.8, 7.9, 7.10, 7.11 and Table 7.2, the GMGA based on the NSII-IRR GA version has the best performance in the 2-D 20-story control benchmark

problem. The whole architecture data for the optimal Pareto front of the Figure 7.10 are attached in the Appendix section.

Table 7.2. The properties of the GMGA for the 20-story building problem

	SP2-IRR GA based	*NSII-IRR GA based*
String length	600	600
Population size	100, 200	100, 100
Mutation probability	0.01	0.01
Crossover probability	0.9	0.9
GMR	0.1	0.1
Generation	700	580
Tournament Selection	Tournament size: 2	Tournament size: 2
CPU running time	56 days	53 days

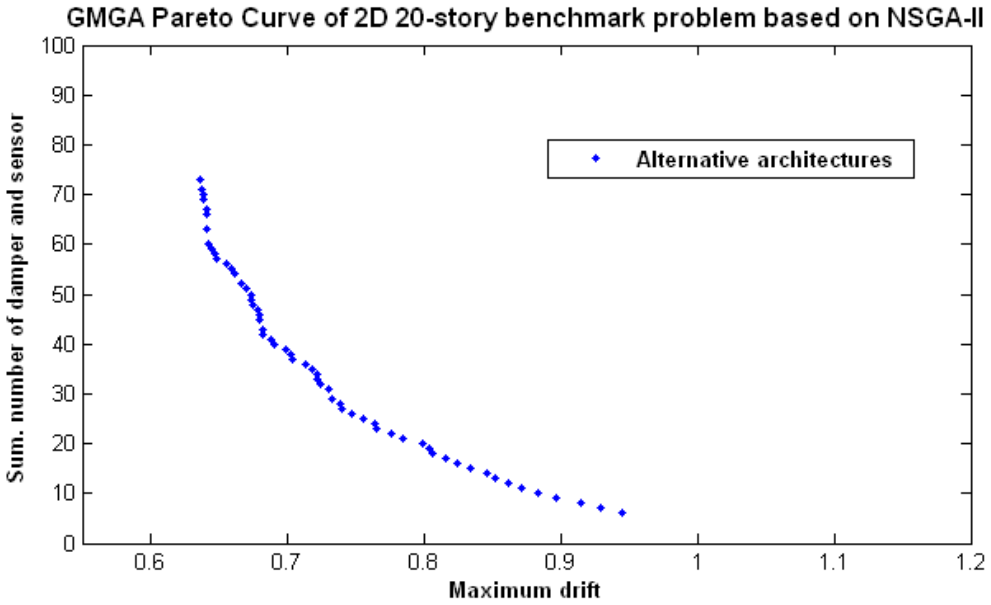

Figure 7.10. The Pareto front obtained using GMGA of 2-D 20-story problem based on NSII-IRR GA

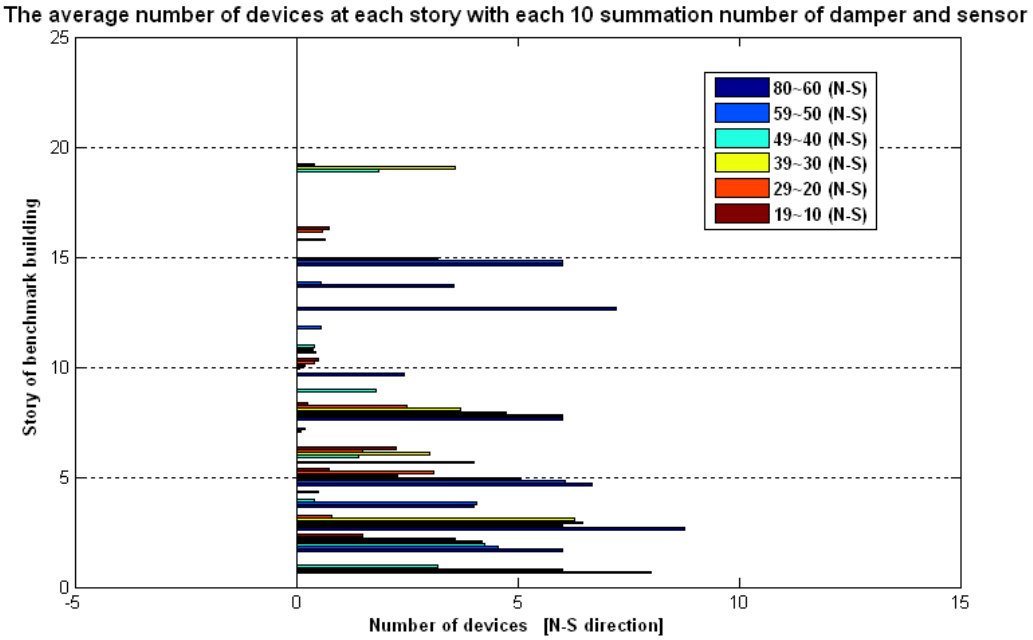

Figure 7.11. The average number of devices at each story using GMGA based on NSGA-II of 2-D 20-story problem in all sections

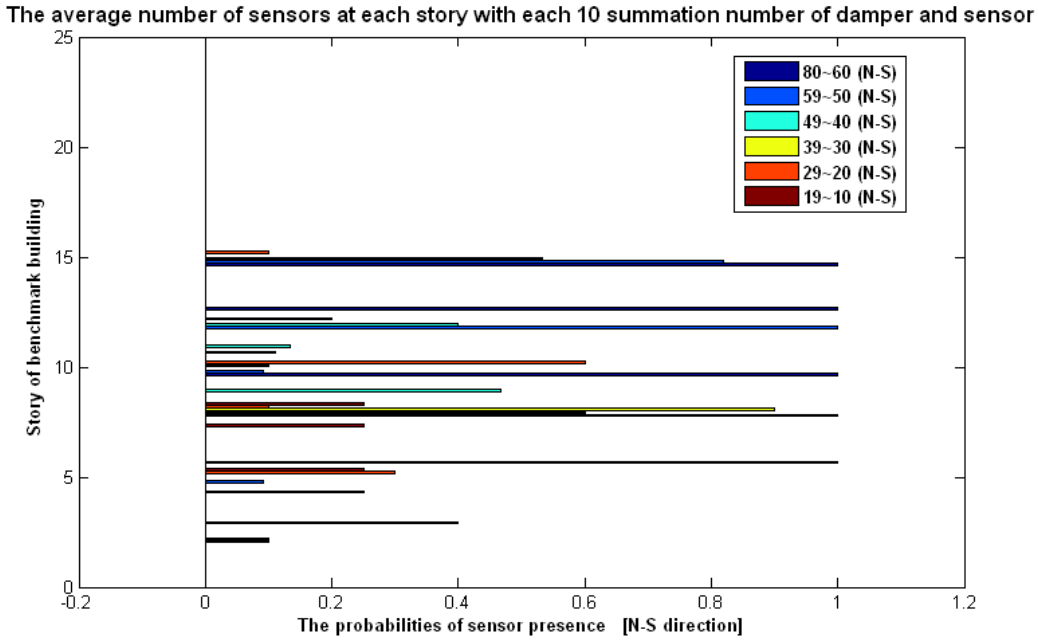

Figure 7.12. The probabilities of sensor presence at each story using GMGA based on NSGA-II of 2-D 20-story problem in all sections

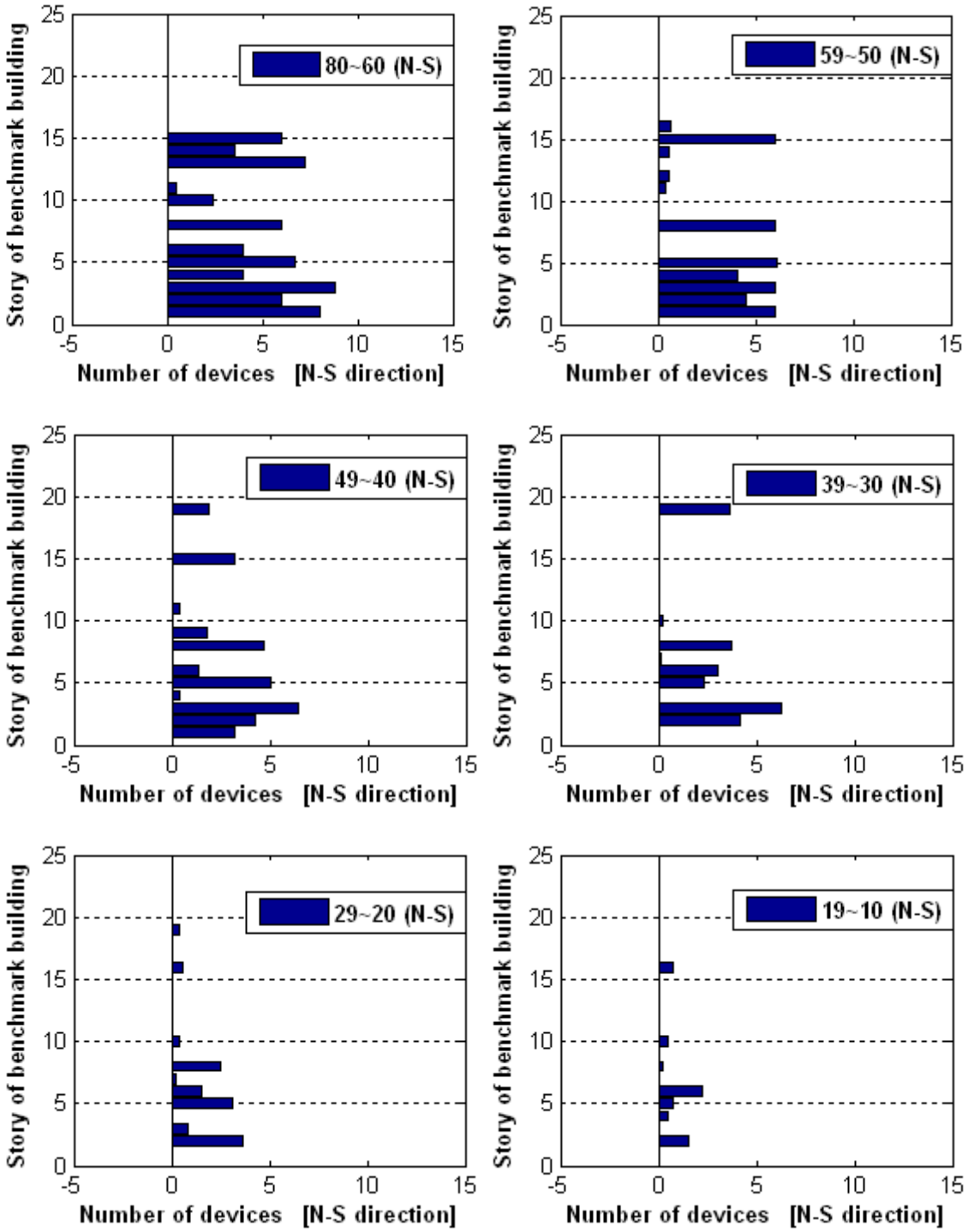

Figure 7.13. The average number of devices at each story using GMGA based on NSGA-II of 2-D 20-story problem

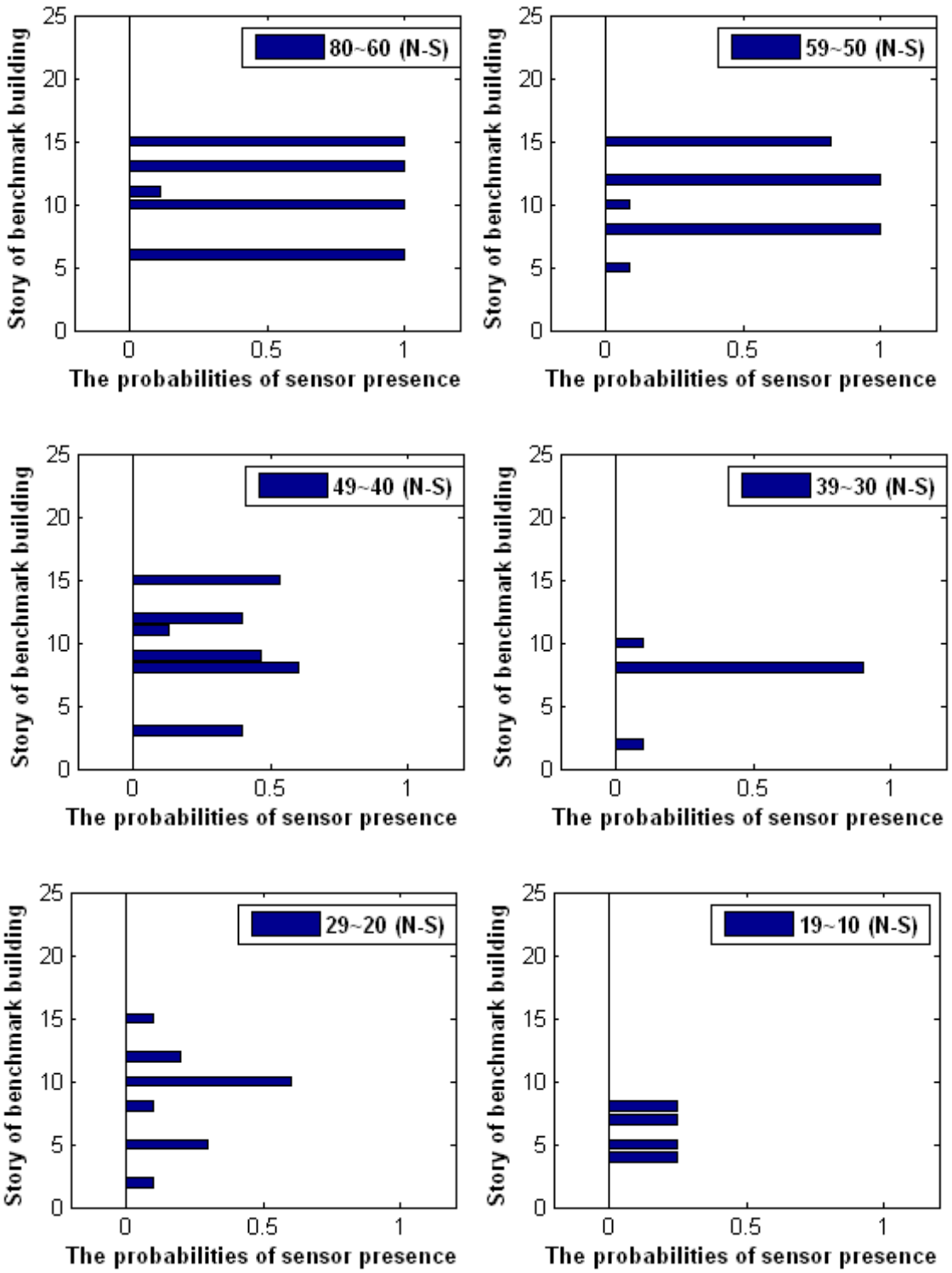

Figure 7.14. The probabilities of sensor at each story using GMGA based on NSGA-II of 2-D 20-story problem

Figure 7.10 is expressed with each 10 summation number of control devices and sensors in Figures from 7.11 to 7.14. The locations and numbers of the control devices from the Figure 7.11 and 7.13 do not have any patterned placement, but the locations and the probabilities of the sensors have some patterns as shown in Figure 7. 12 and 7.14. For the sections of the 80~60 and 59~50, the floor levels 6th, 10th, 13th, and 15th are good choice for the location of the sensors

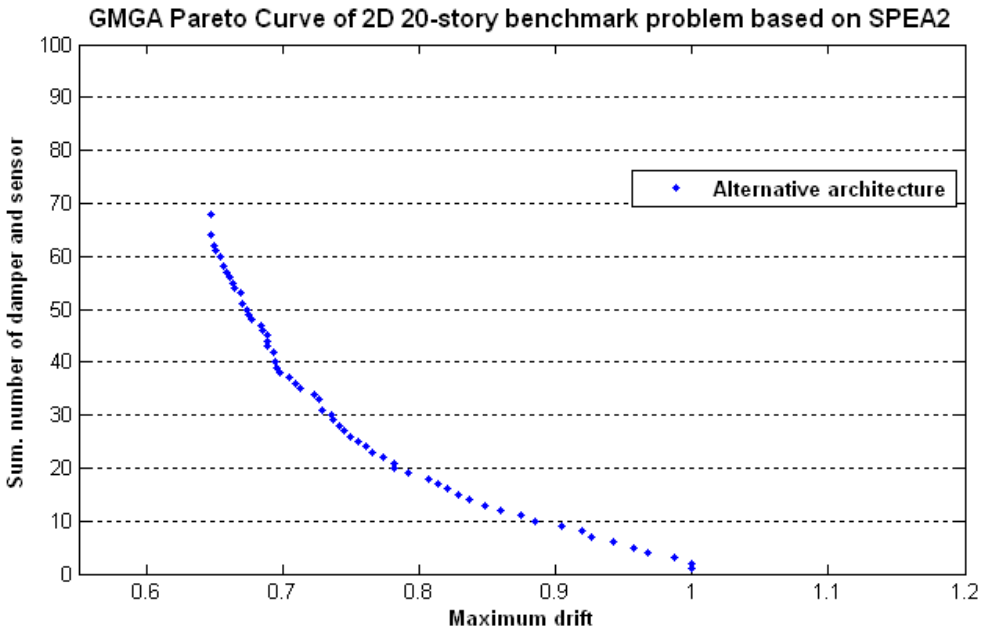

Figure 7.15. The Pareto front obtained using GMGA of 2-D 20-story problem based on SP2-IRR GA

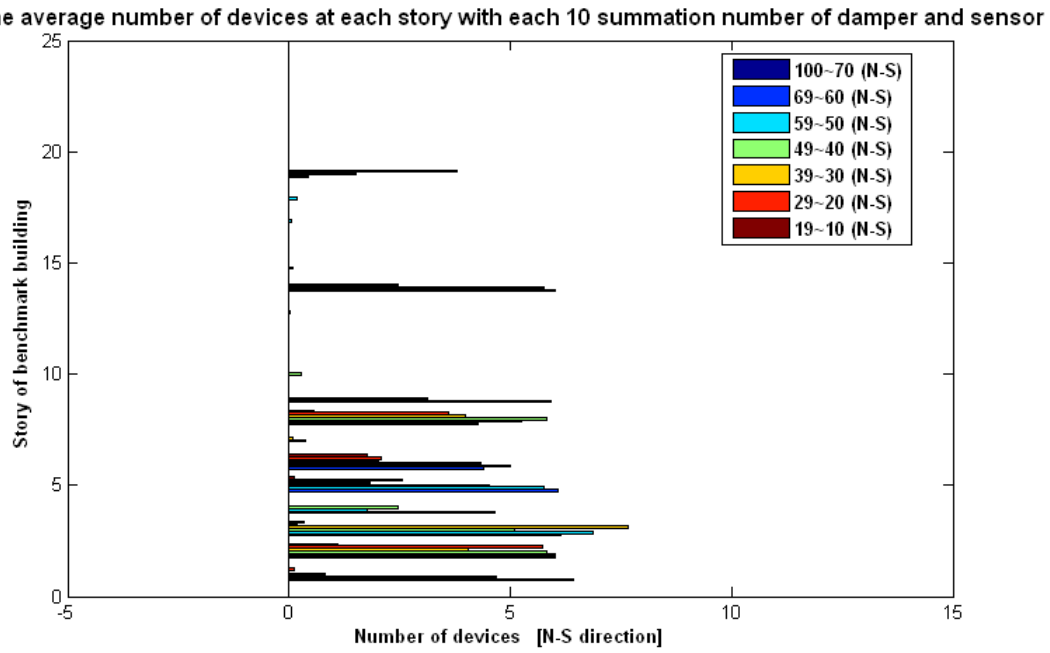

Figure 7.16. The average number of devices at each story using GMGA based on SPEA2 of 2-D 20-story problem in all sections

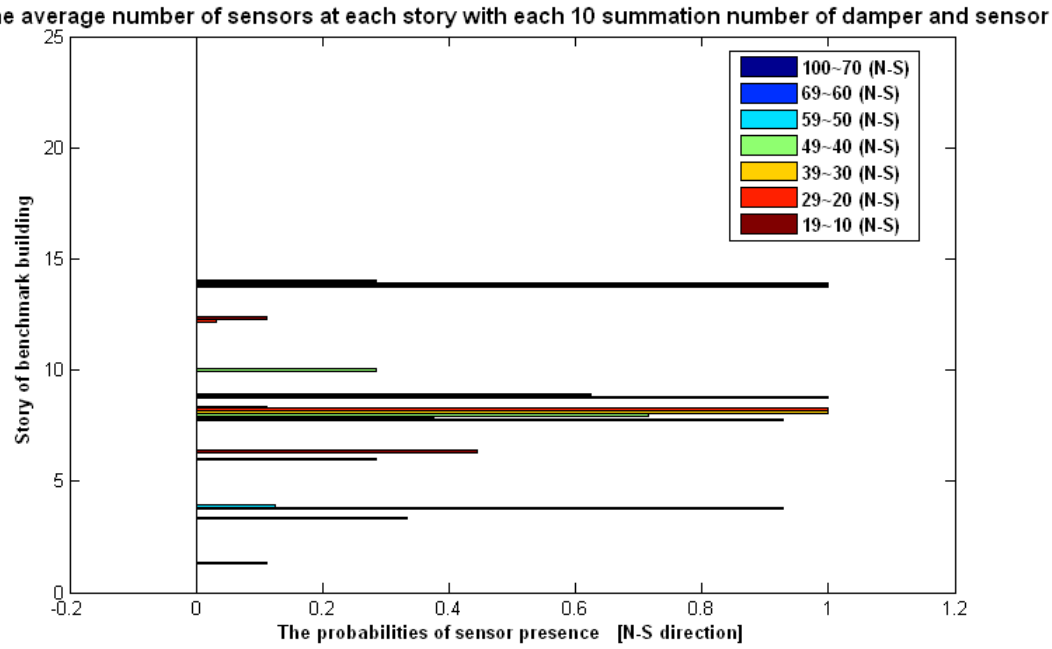

Figure 7.17. The probabilities of sensor presence at each story using GMGA based on SPEA2 of 2-D 20-story problem in all sections

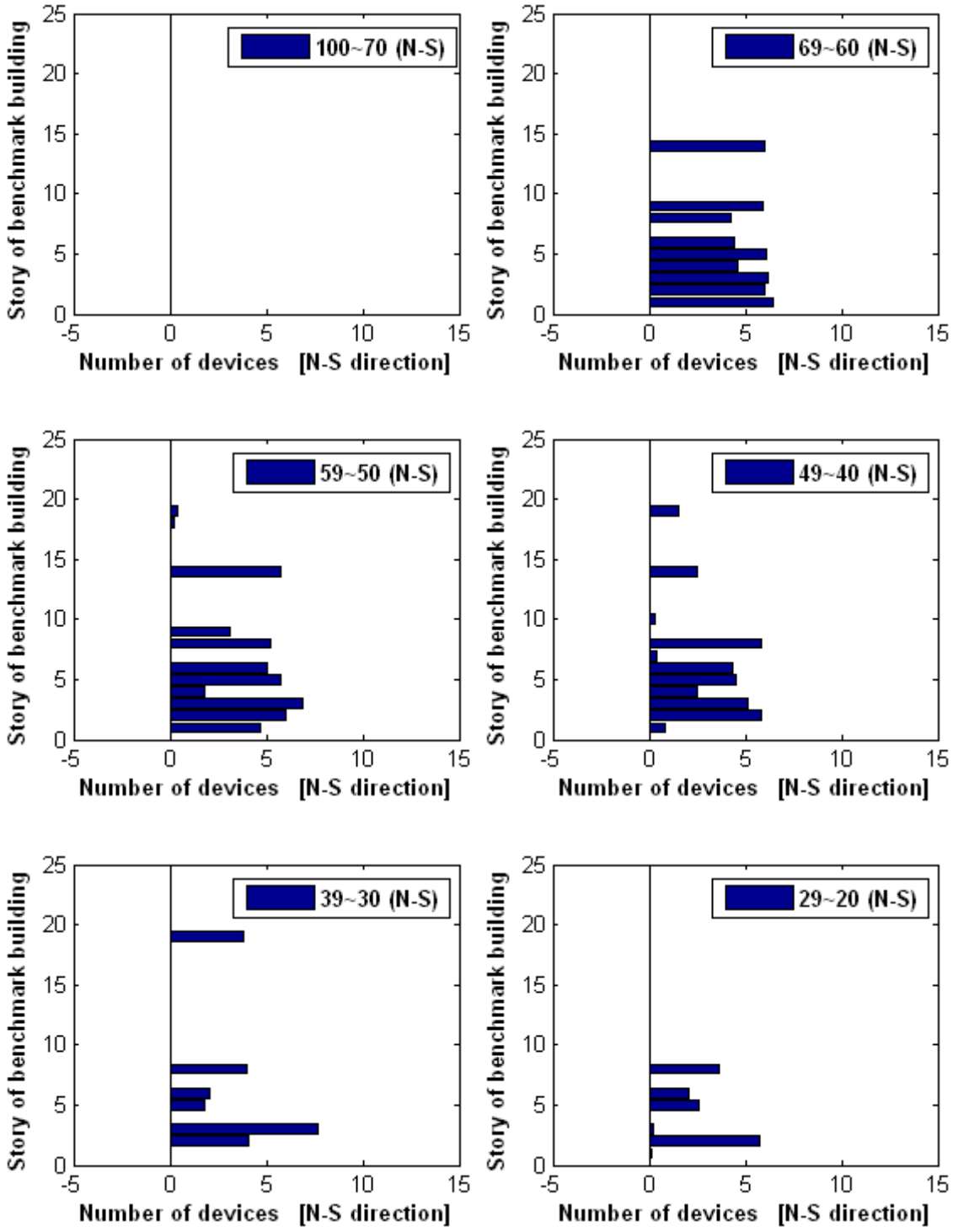

Figure 7.18. The average number of devices at each story using GMGA based on SPEA2 of 2-D 20-story problem

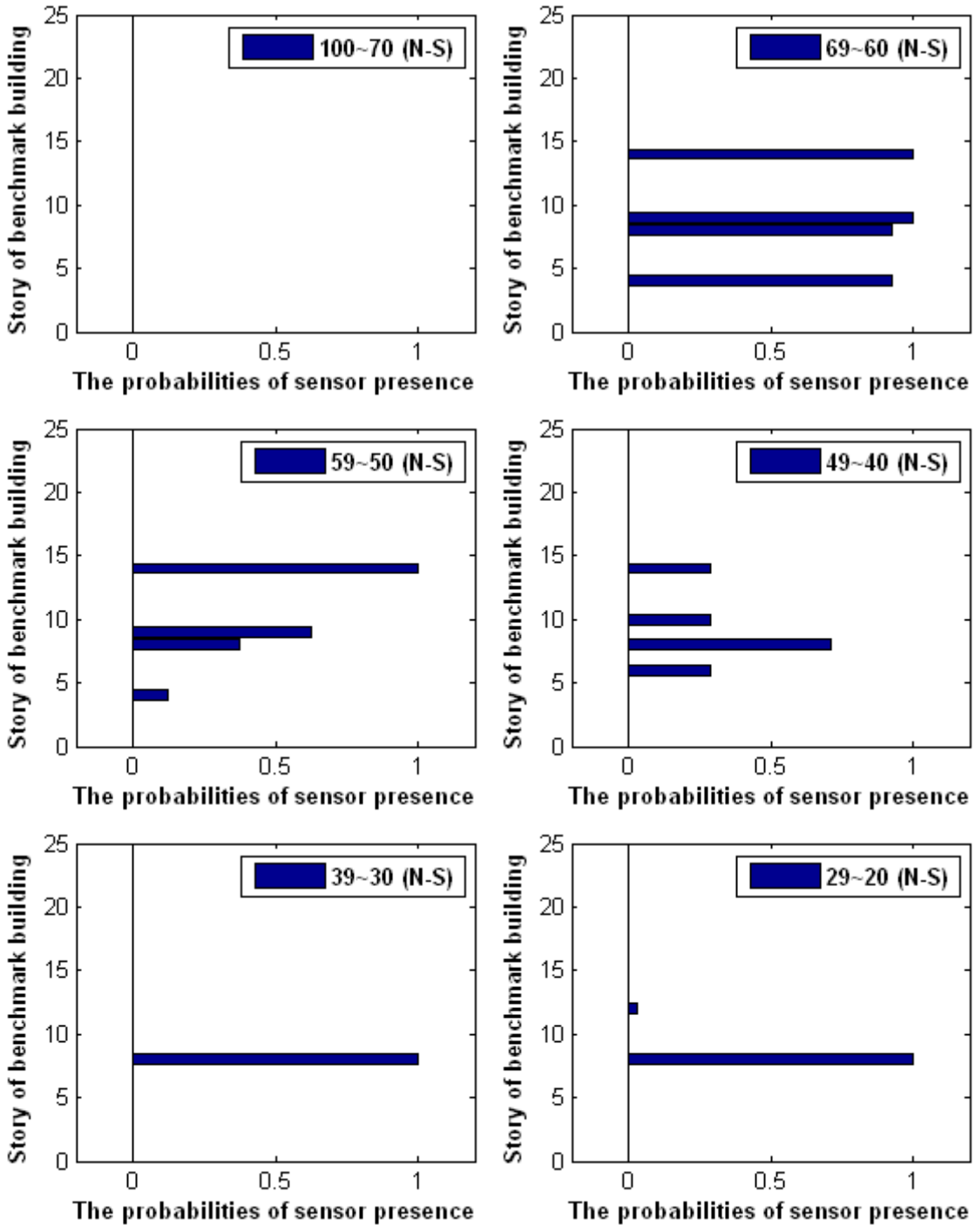

Figure 7.19. The probabilities of sensor presence at each story using GMGA based on SPEA2 of 2-D 20-story problem

From the comparisons of the Figures from 7.11 to 7.19, the damper locations of the each story are very diverse. The two suggested methods have generally similar but their own characteristic architectures of the control devices. Dependent on the locations of the sensor and the aimed range of the maximum drift, the number and locations of the dampers are very different. Thus the average numbers of dampers and the probability of sensor presence for the each story with grouped each 10 summation of number of devices are good data to control engineers. From the graphs, the control engineers can decide the goal of the maximum drift and its possible numbers and locations of the devices.

To verify the performance of the developed GMGA based on the NSII-, and SP2-IRR GA, the Pareto fronts are compared. From the Table 7.3, GMGA based on the NSII-IRR GA show the best performance in finding the near-optimal Pareto front with small iteration and CPU time. From the Figures from 7.20 to 7.23, the gene manipulation genetic operator enhances the performances of the NSII-, and SP2-IRR GA. Specially, these graphs demonstrate that GMGA can significantly reduce the computational analysis time. The large consumption of analysis time makes problem in engineering sites with retardation of the construction process.

Table 7.3. Comparison of the CPU times and generation

	NSII-IRR GA	SP2-IRR GA	GMGA based on NSII-IRR GA	GMGA based on SP2-IRR GA
Generation	700	910	580	700
CPU time	90 days	114 days	53 days	56 days

From the Figure 7.21, GMGA results are better than the SP2-IRR GA in whole Pareto front without the edges. The reason of the weakness of the GMGA in finding at each edge is the assignment of the creation rule. New gene instances in all the sections of the Pareto front are generated by the same rule as the following mean, maximum, minimum, and random values. To solve this weakness at each edge, the creation rule at each edge should be different. The maximum value should be used as the first mechanism at the right edge side of the Pareto front, and the minimum value should be used as the first mechanism at the left edge side of the Pareto front. With this changing mechanism, the weakness in GMGA at each edge will be removed.

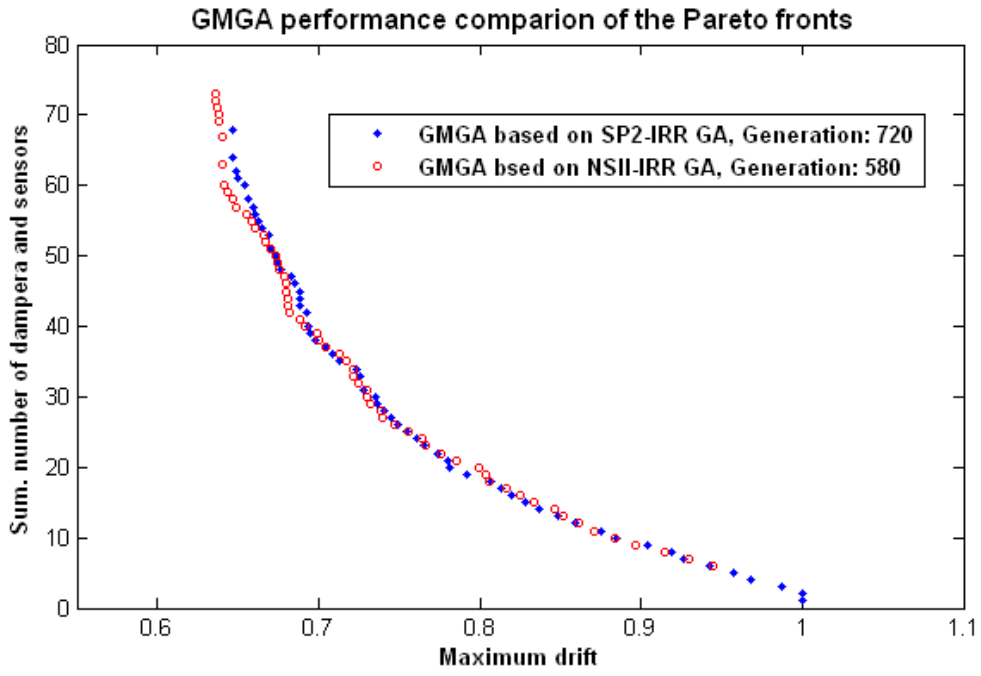

Figure 7.20. GMGA performance comparison of the Pareto fronts to NSII-IRR GA in 2-D 20-story building problem

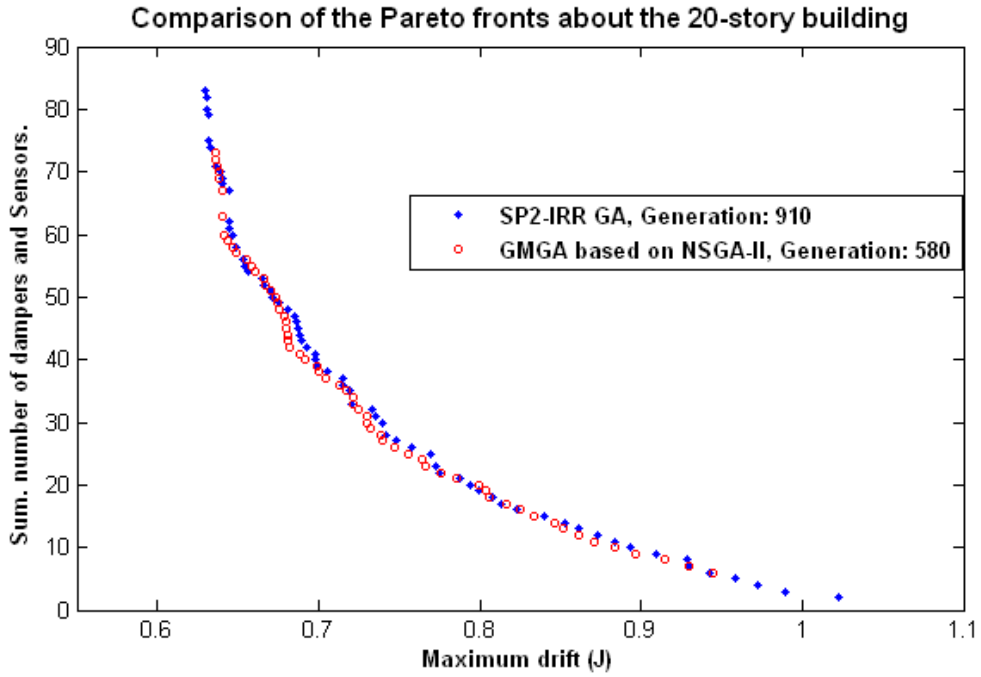

Figure 7.21. GMGA performance comparison of the Pareto fronts to SP2-IRR GA in 2-D 20-story building problem

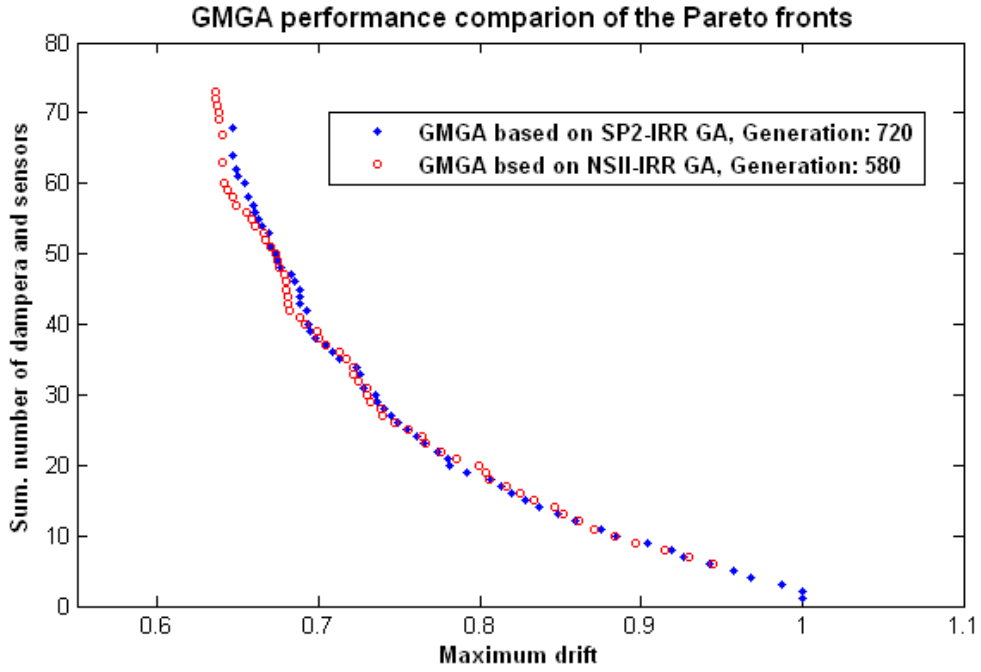

Figure 7.22. GMGA performance comparison of the Pareto fronts based on NSII-IRR GA and SP2-IRR GA in 2-D 20-story building problem

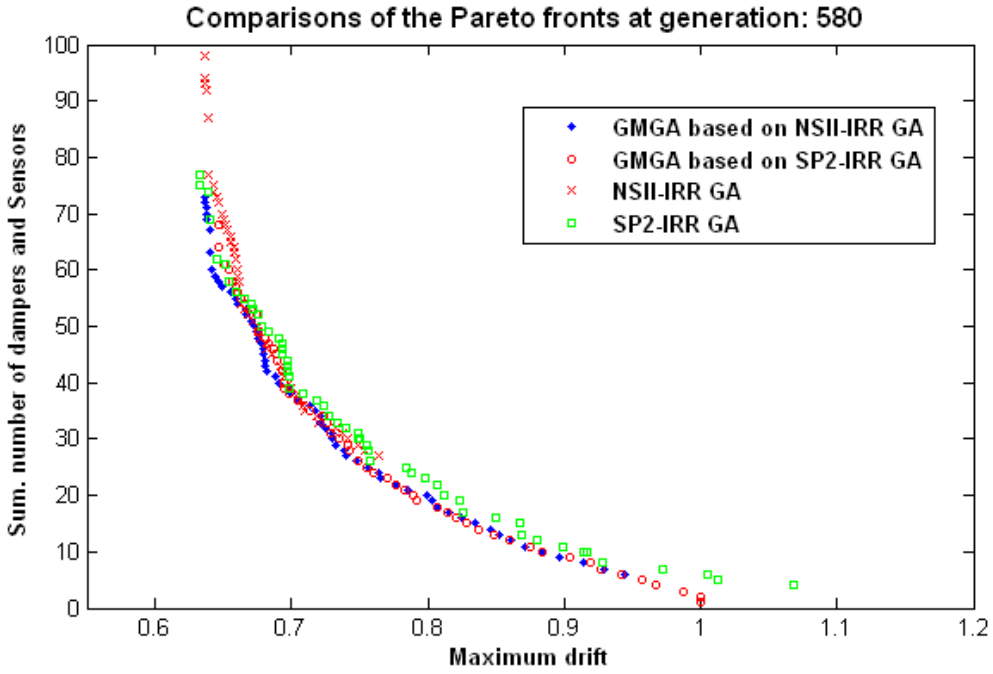

Figure 7.23. Comparison of Pareto fronts to NSII- and SP2-IRR GA and GMGAs at generation: 580 in 2-D 20-story building problem

With the comparison of the CPU running time based on the generation which have fully converged Pareto fronts, the GMGA based on the NSII-IRR GA has better performance in searching the near near-optimal Pareto front in 2-D 20-story control benchmark problem.

7.6 Performances of GMGA in 3-D 20-story Control Benchmark Buildings

Until now, the 3-D model control with LQG has not been highly researched. From this 3-D model control problem, the availability of practical control of the 3-D model is shown. Although the control performance with the 2-D model of civil structure is good, reality of the 3-D control is that it is not easy to control and reduce the seismic responses. Thus for the structural engineer or control engineer, the architectures of the control devices of the 3-D model should be offered. And this study will show the reliable construability of the LQG control algorithm about the civil building structural system.

The properties of the GMGA for the 3-D 20-story control benchmark building problem based on the NSGA-II are expressed in Table 7.4. The defined string lengths are applied as different values to find the reasonable redundancy ratio of the redundant segment of the IRR GA, but the other properties of GMGA such as population size, mutation and crossover probability and tournament selection sizes are the same as the previously defined NSII-IRR GA and SP2-IRR GA to compare the performances of the GMGA to the two defined GAs. The performances of the GMGA based on NSII- and SP2-IRR GA show excellent ability in finding the Pareto front in the 3-D problem and in reducing of the CPU time consumption and iteration. From the comparisons of the GMGA and the two defined GAs: NSII- and SP2-IRR GA, GMGA has the ability to get out of near stability and controllability areas requiring longer CPU analysis time by assigning new and good architectures of control devices faster than the two defined GAs.

Table 7.4. The properties of the GMGA for the 3-D 20-story building problem based on NSII –IRR GAwith different redundancy ratio

	NSII-IRR GA based	*NSII-IRR GA based*
String length	1200 (Redundancy ratio: 0.3)	1600 (Redundancy ratio: 0.5)
Population size	200, 200	200, 200
Mutation probability	0.01	0.01
Crossover probability	0.9	0.9
GMR	0.3	0.3
Generation	155	96
Tournament Selection	Tournament size 2	Tournament size 2
CPU running time	95 days	71 days

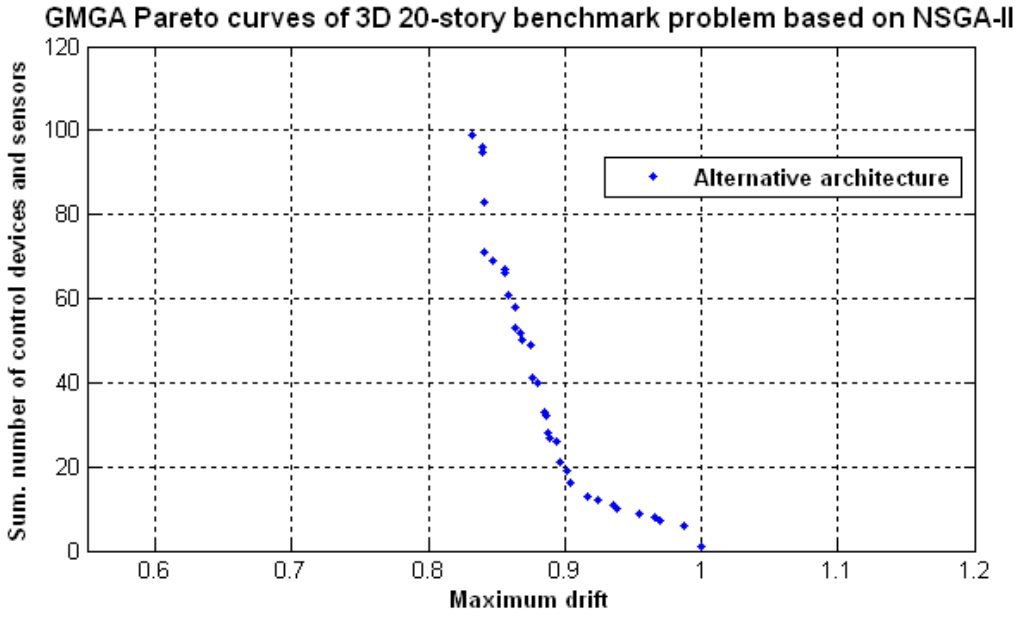

Figure 7.24. The Pareto front obtained using GMGA based on NSII-IRR GA of 3-D 20-story problem with redundancy ratio 0.3

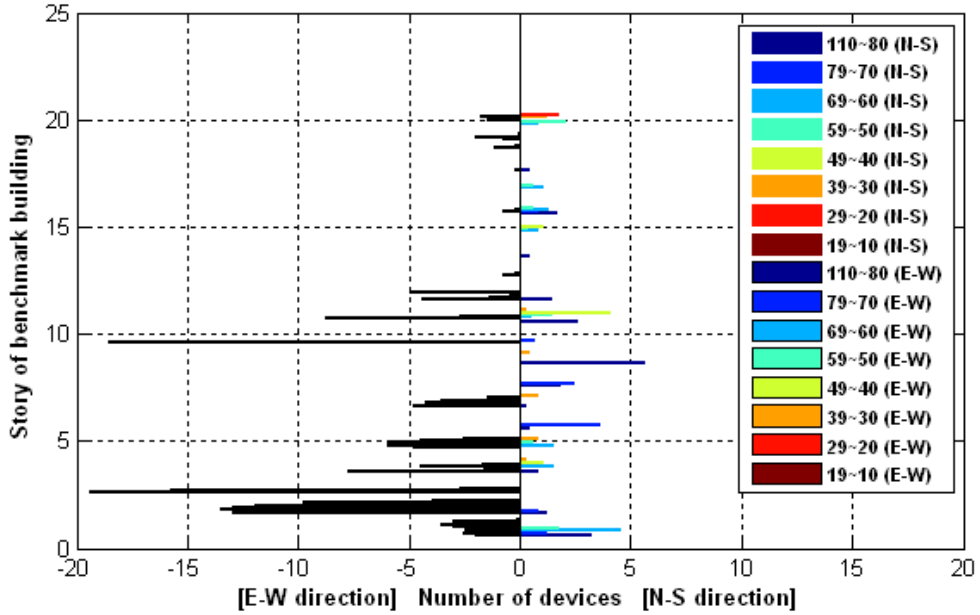

Figure 7.25. The average number of devices at each story using GMGA based on NSII-IRR GA with redundancy ratio 0.3 of 3-D 20-story building in all sections

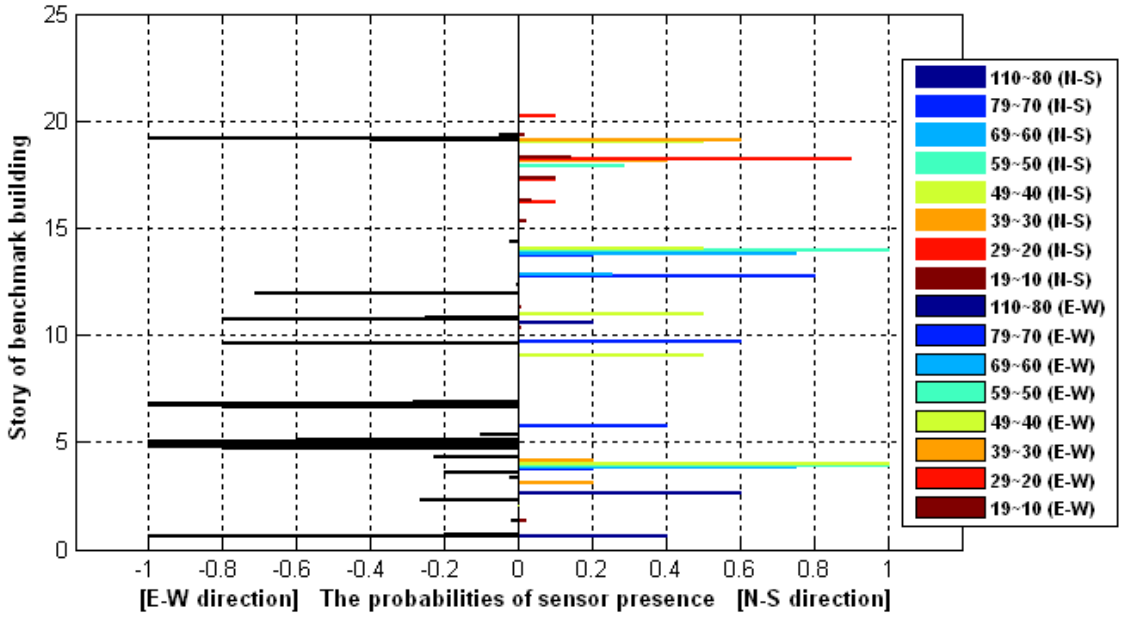

Figure 7.26. The probabilities of sensor presence at each story using GMGA based on NSII-IRR GA with redundancy ratio 0.3 of 3-D 20-story building in all sections

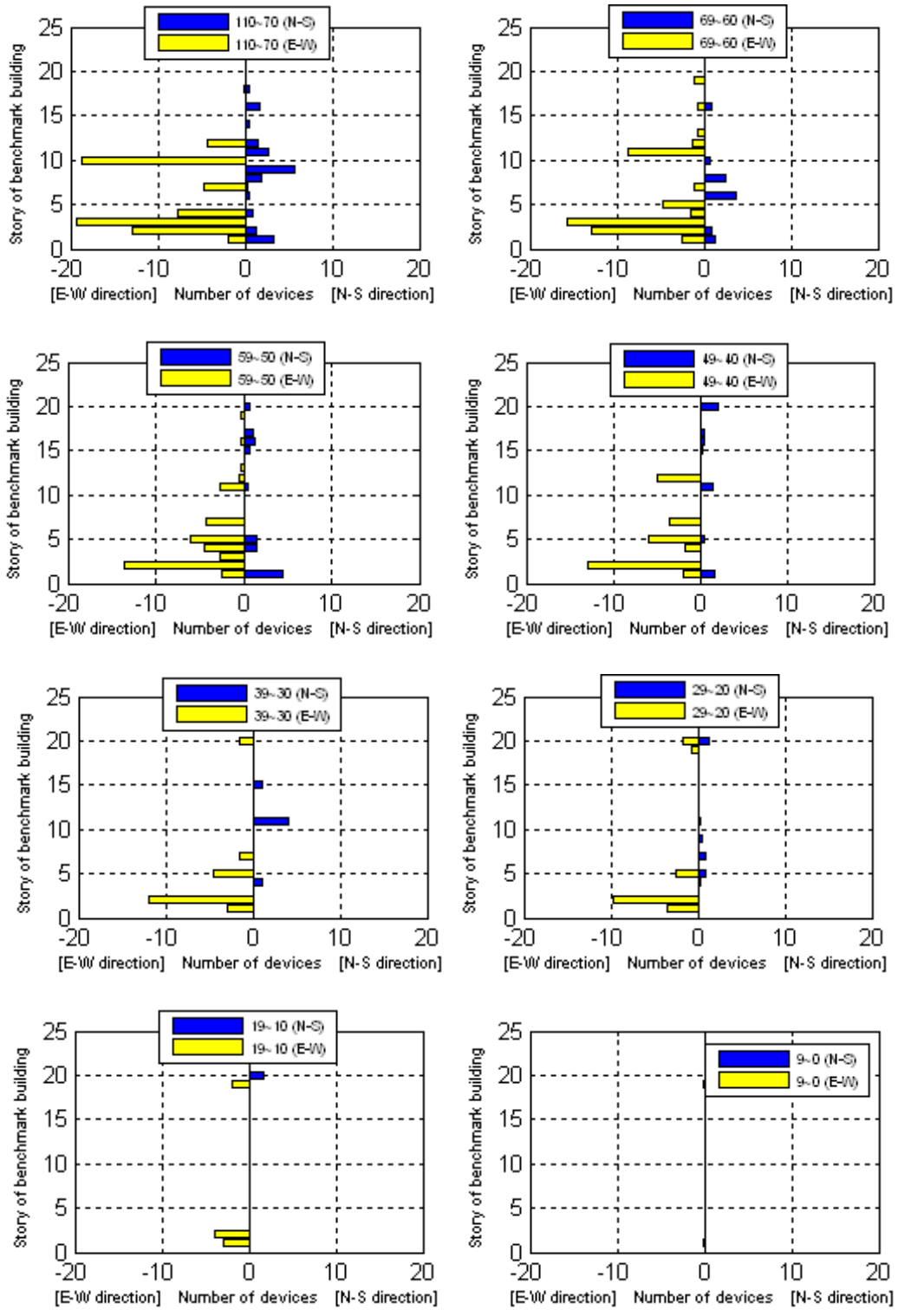

Figure 7.27. The average number of devices at each story using GMGA based on NSII-IRR GA with redundancy ratio 0.3 of 3-D 20-story building

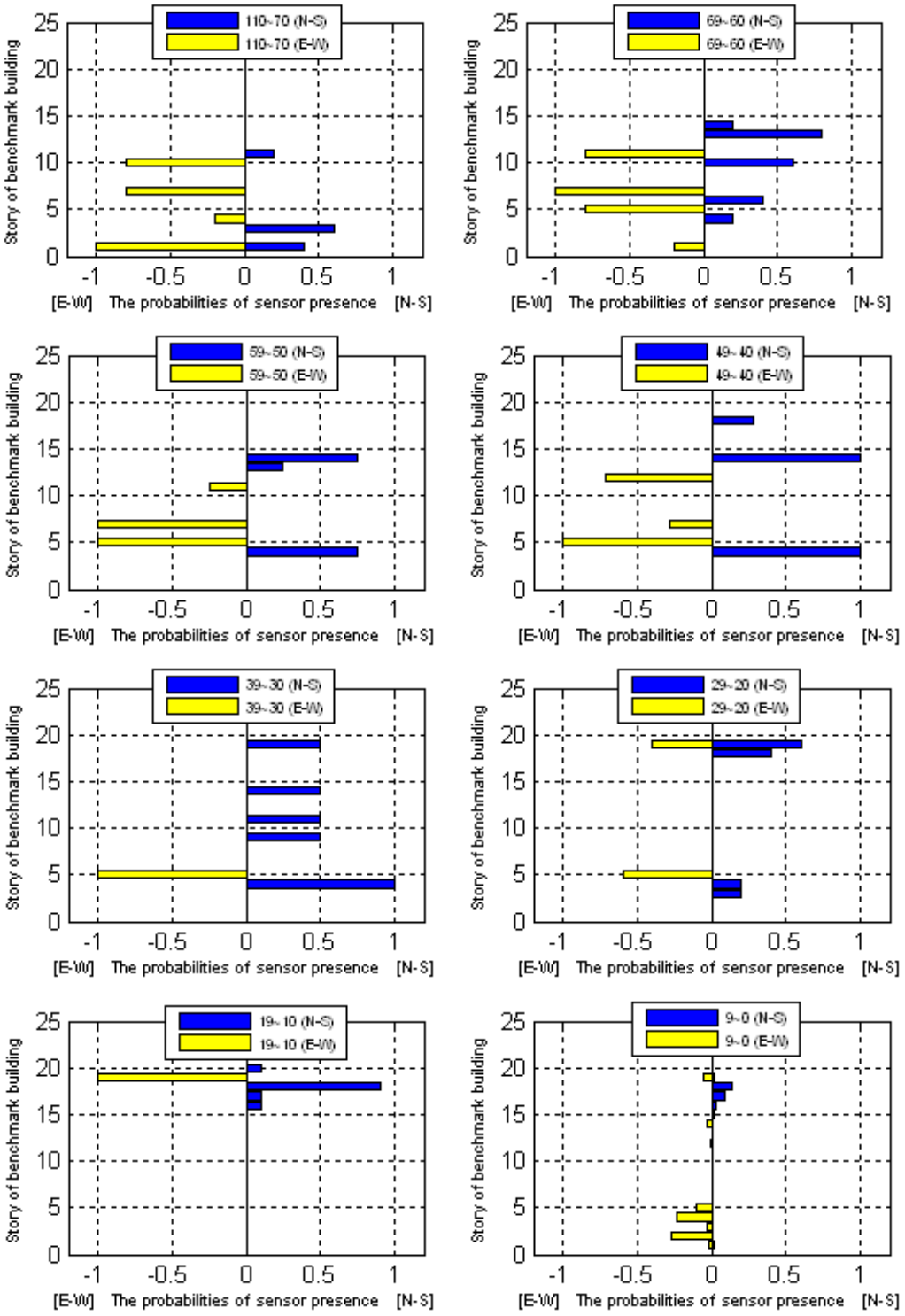

Figure 7.28. The probabilities of sensor at each story using GMGA based on NSII-IRR GA with redundancy ratio 0.3 of 3-D 20-story building

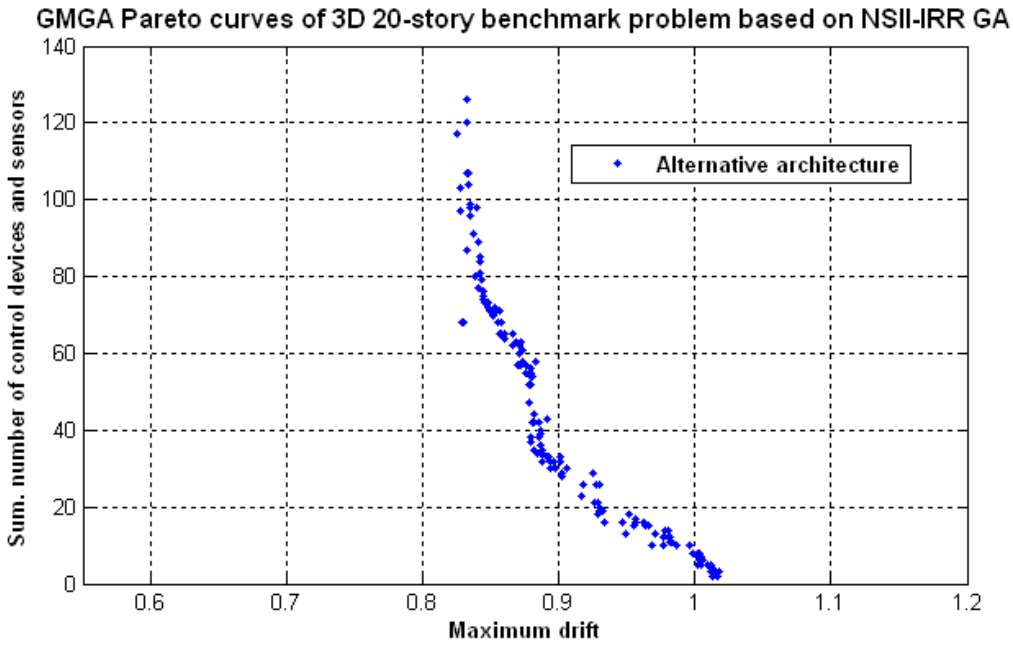

Figure 7.29. The Pareto front obtained using GMGA based on NSII-IRR GA of 3-D 20-story problem with redundancy ratio 0.5

From the detailed data analysis of the Figures from 7.24 to 7.33, the Pareto front is fully converged to its near-optimal values. The interesting facts are the numbers of dampers in the E-W direction of the structure and the locations of dampers and sensors in the two directions. The number of dampers in the E-W direction is generally larger than in the N-S direction. In the engineering concept, the N-S direction of the structure is more flexible in strength, so bigger seismic responses are expected and more control devices are subsequently required. However, the near-optimal architecture of the 3-D model control analysis shows that to reduce the responses of the whole building structure, more dampers should be installed in the E-S direction. The locations of the control devices do not generally overlap in order to reduce the collisions of the control force.

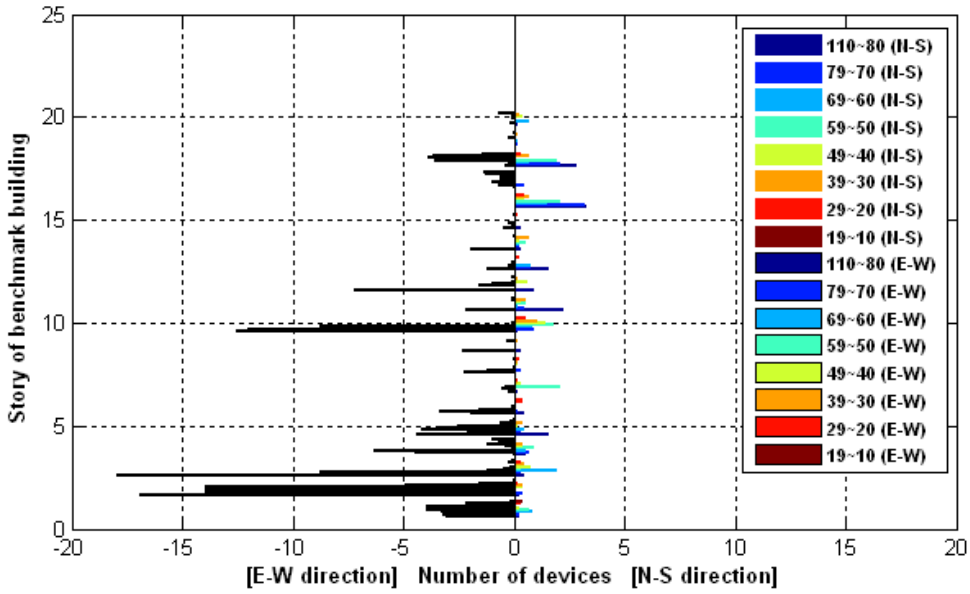

Figure 7.30. The average number of devices at each story using GMGA based on NSII-IRR GA of 3-D 20-story problem with redundancy ratio 0.5 in all sections

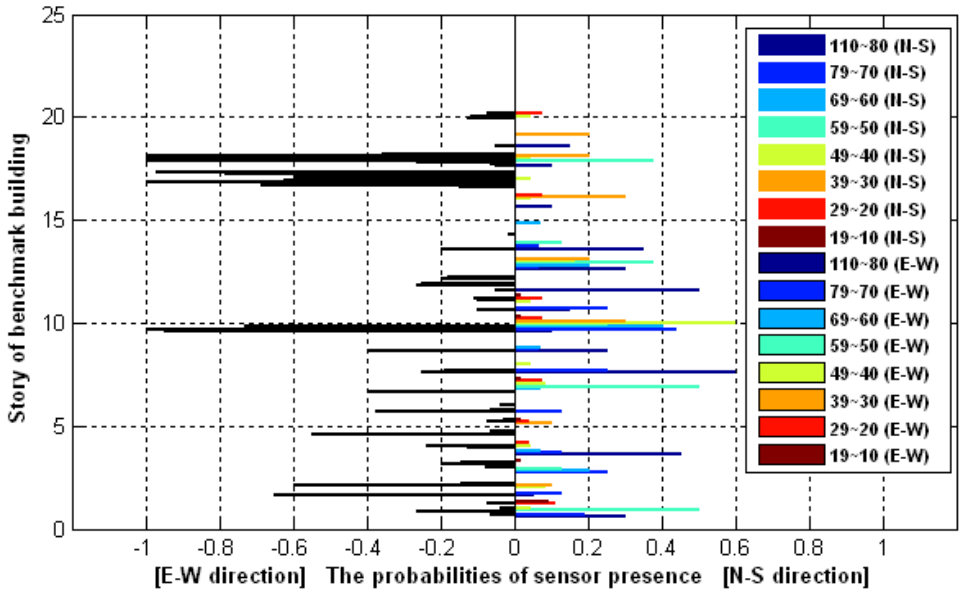

Figure 7.31. The probabilities of sensor presence at each story using GMGA based on NSII-IRR GA of 3-D 20-story problem with redundancy ratio 0.5 in all sections

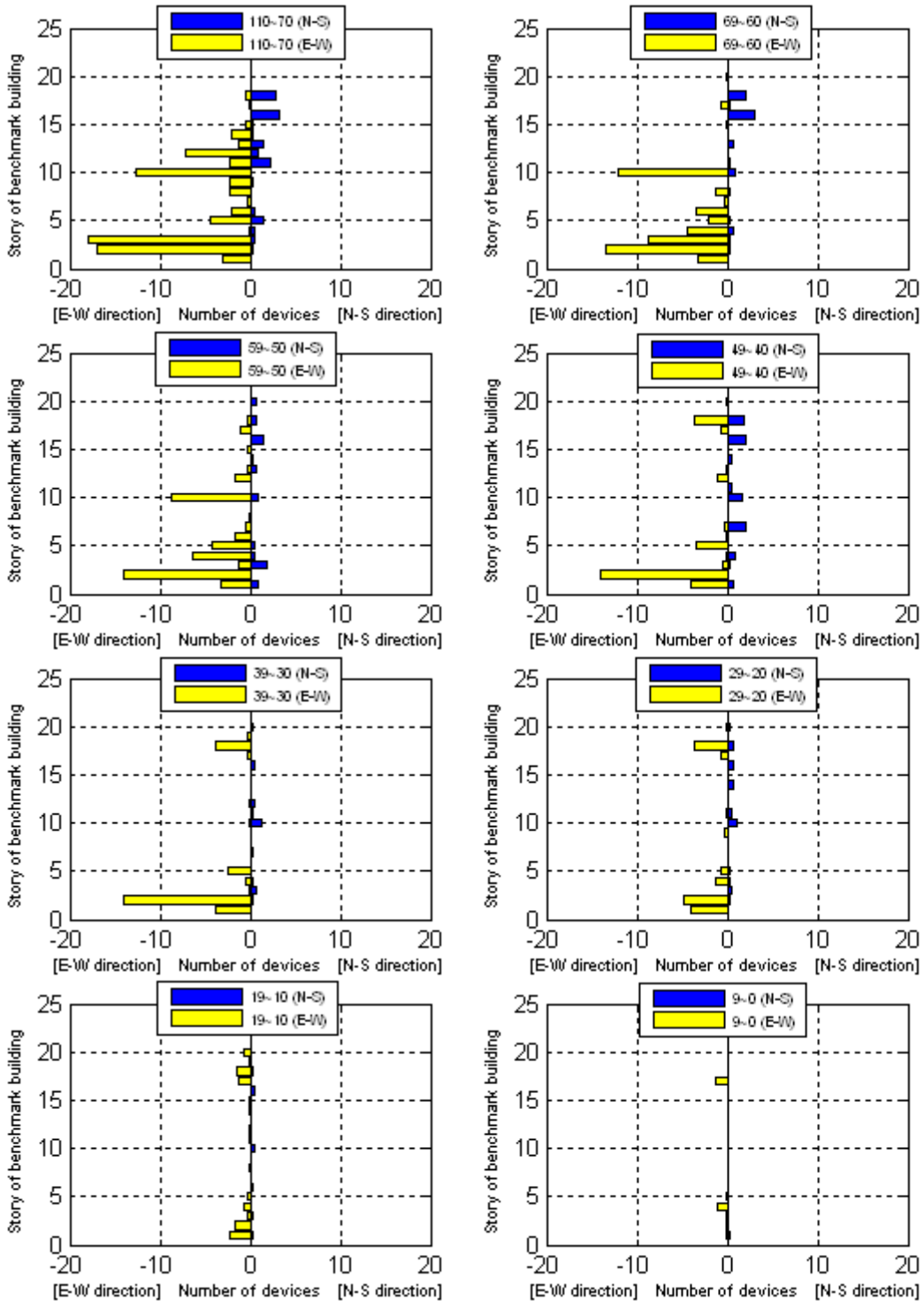

Figure 7.32. The average number of devices at each story using GMGA based on NSII-IRR GA of 3-D 20-story problem with redundancy ratio 0.5

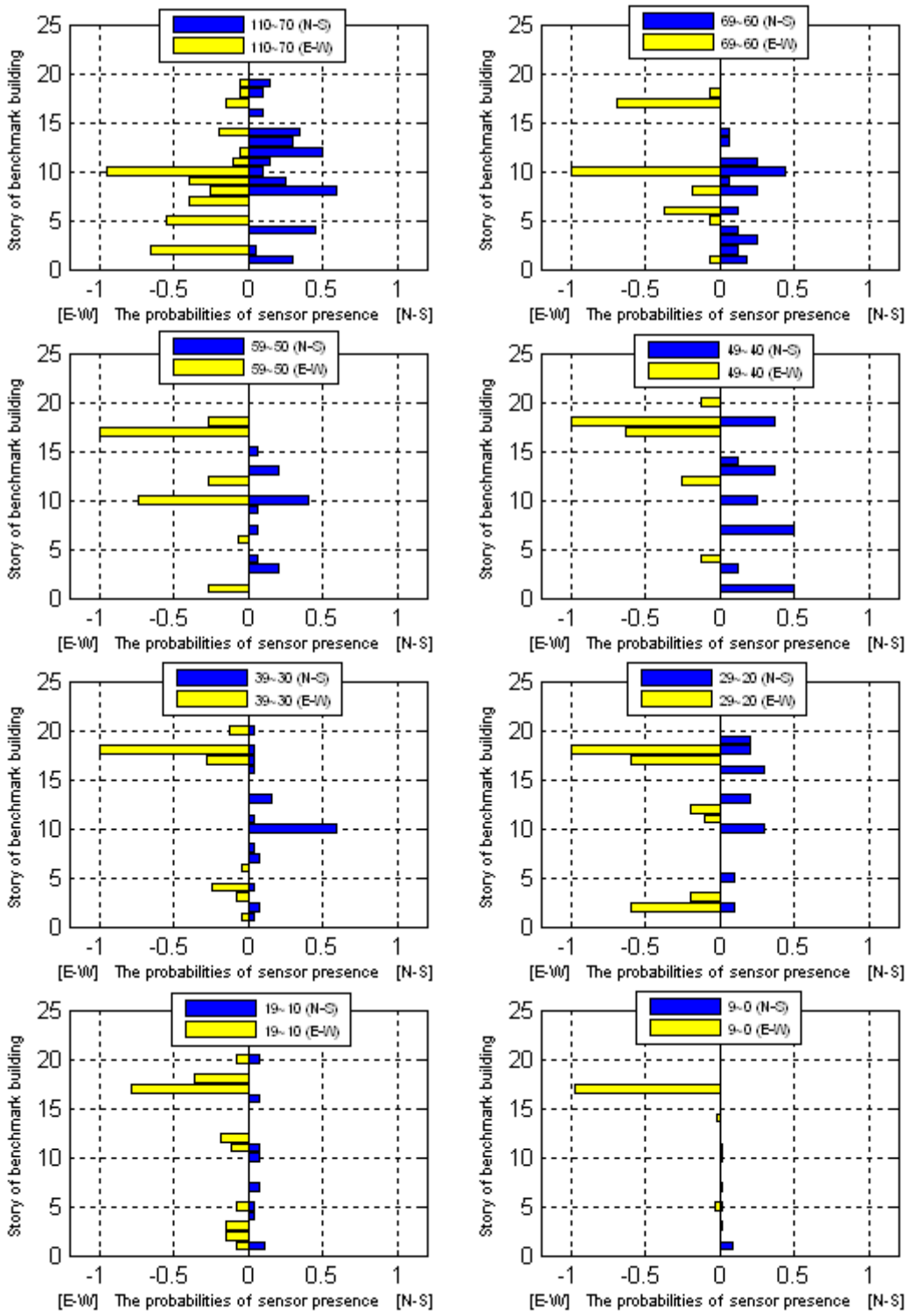

Figure 7.33. The probabilities of sensor at each story using GMGA based on NSII-IRR GA of 3-D 20-story problem with redundancy ratio 0.5

The properties of the GMGA for the 3-D 20-story control benchmark building problem based on the SPEA2 are expressed in Table 7.5. The different string lengths are applied to find the reasonable redundancy ratio of the redundant segment of the IRR GA, but the other properties such as population size, mutation and crossover probability and tournament selection sizes are the same as the previously proposed SP2-IRR GA. Thus these performances can be reasonably compared other methods. The performances of the GMGA based on SPEA2 show excellent ability to find the Pareto front and reduce the CPU time in the 3-D problem. From the comparison of the Figures from 7.34 to 7.43, the redundancy ratio 0.3 is a good value for this 3-D control optimization problem.

Table 7.5. The properties of the GMGA for the 3-D 20-story building problem based on SPEA2 with different redundancy ratio

	SPEA2 based	*SPEA2 based*
String length	1200 (redundancy ratio: 0.3)	1600 (redundancy ratio: 0.5)
Population size	200, 300	200, 300
Mutation probability	0.01	0.01
Crossover probability	0.9	0.9
GMR	0.3	0.3
Generation	166	155
Tournament Selection	Tournament size 2	Tournament size 2
CPU running time	69 days	68 days

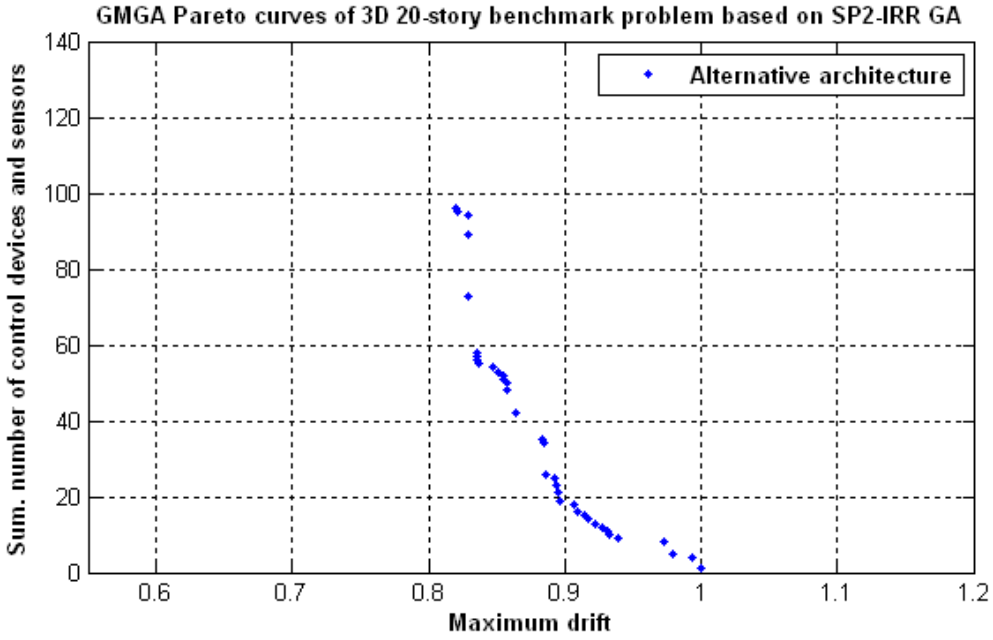

Figure 7.34. The Pareto front obtained using GMGA based on SP2-IRR GA of 3-D 20-story problem with redundancy: 0.3

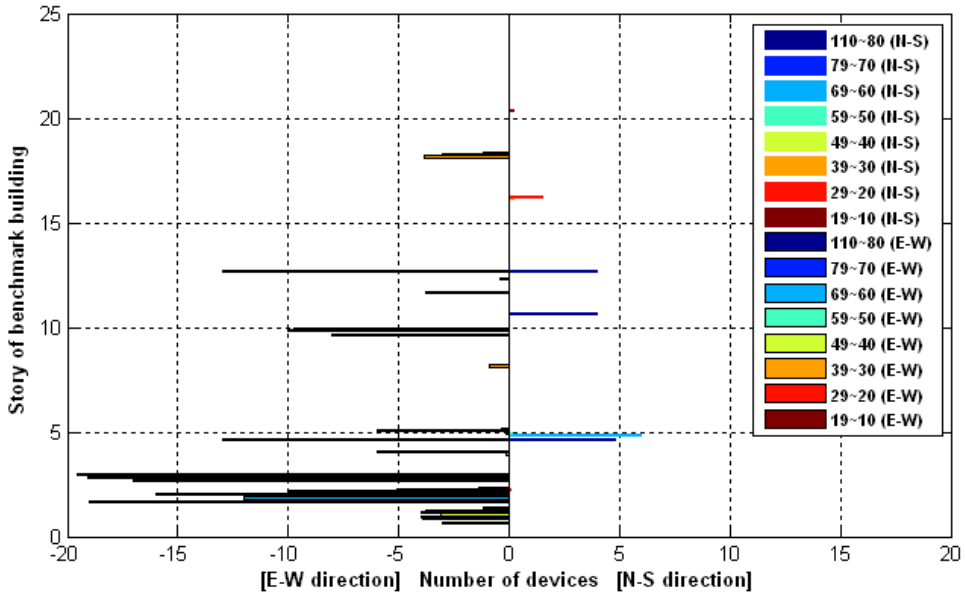

Figure 7.35. The average number of devices at each story using GMGA based on SP2-IRR GA of 3-D 20-story problem with redundancy: 0.3 in all sections

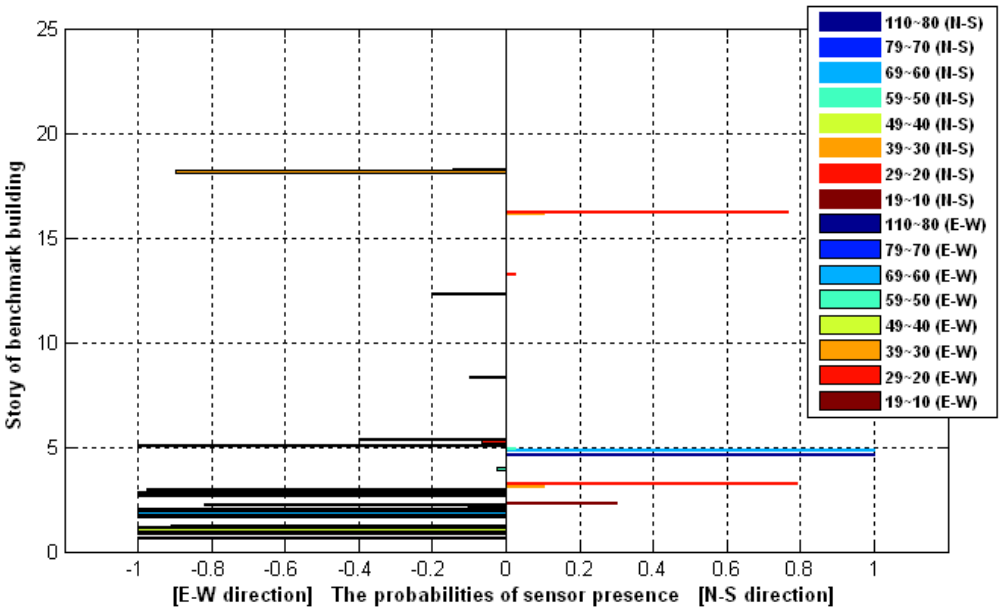

Figure 7.36. The probabilities of sensor presence at each story using GMGA based on SP2-IRR GA of 3-D 20-story problem with redundancy: 0.3 in all sections

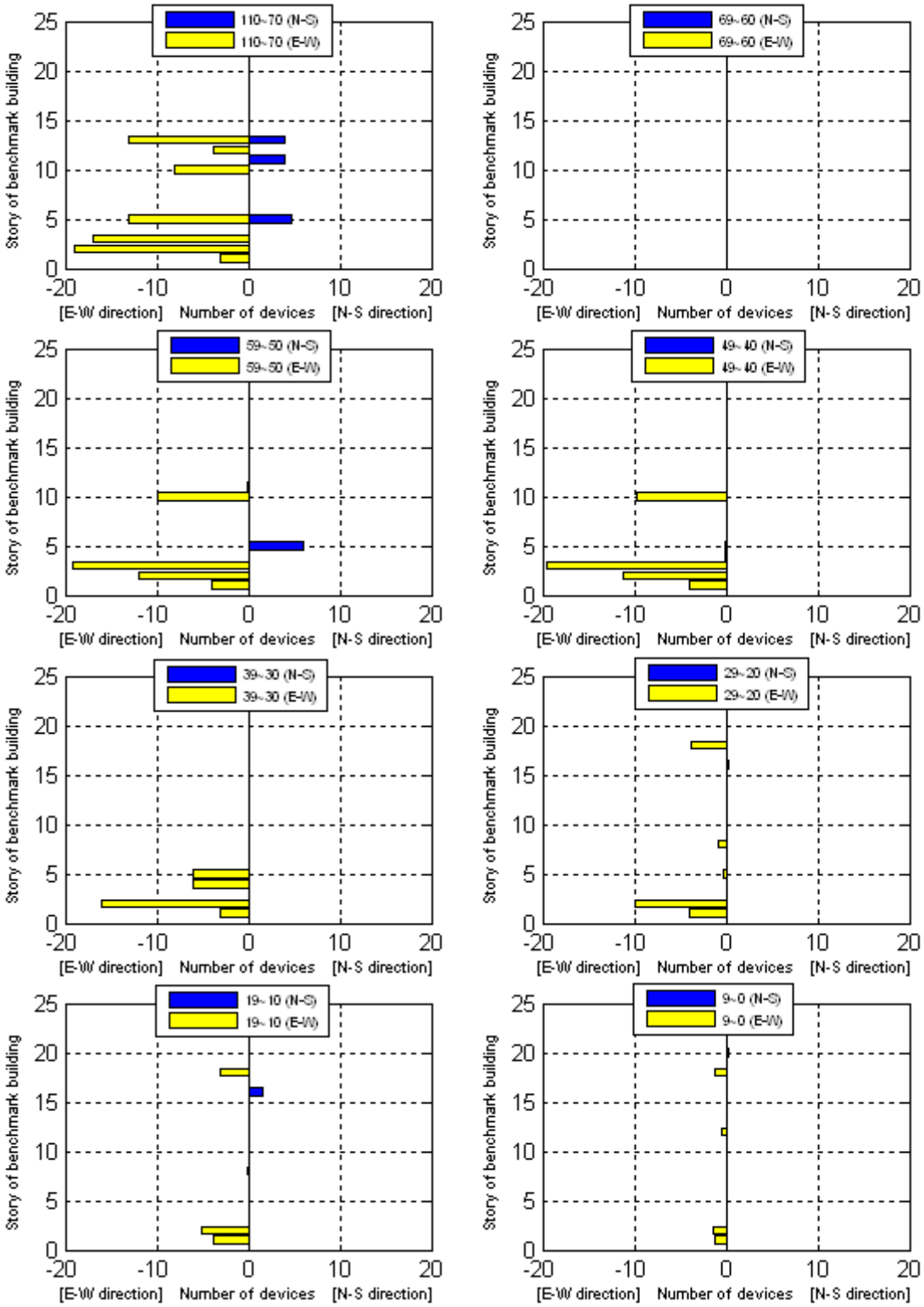

Figure 7.37. The average number of devices at each story using GMGA based on SP2-IRR GA of 3-D 20-story problem with redundancy: 0.3 in all sections

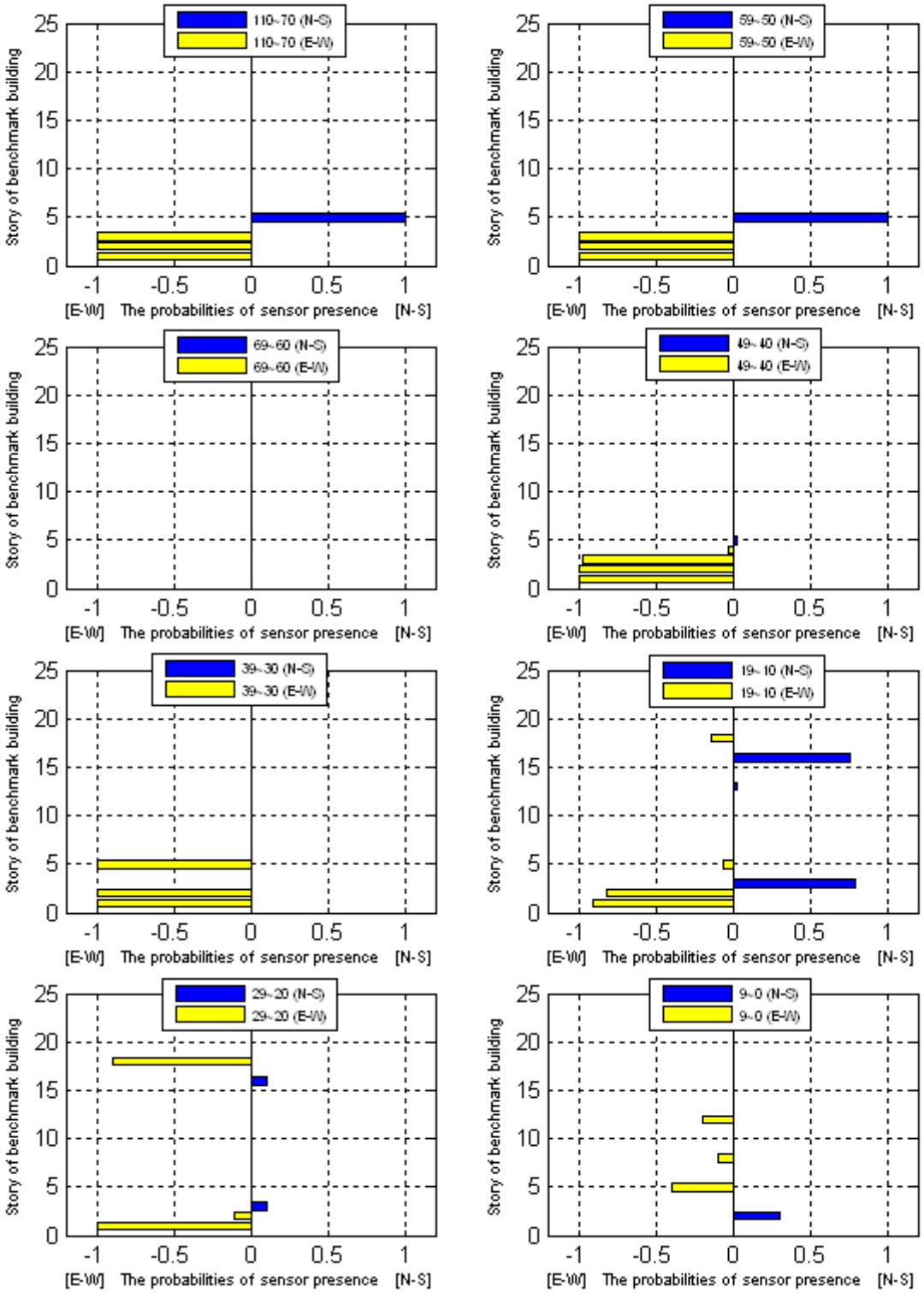

Figure 7.38. The probabilities of sensor at each story using GMGA based on SP2-IRR GA of 3-D 20-story problem with redundancy: 0.3

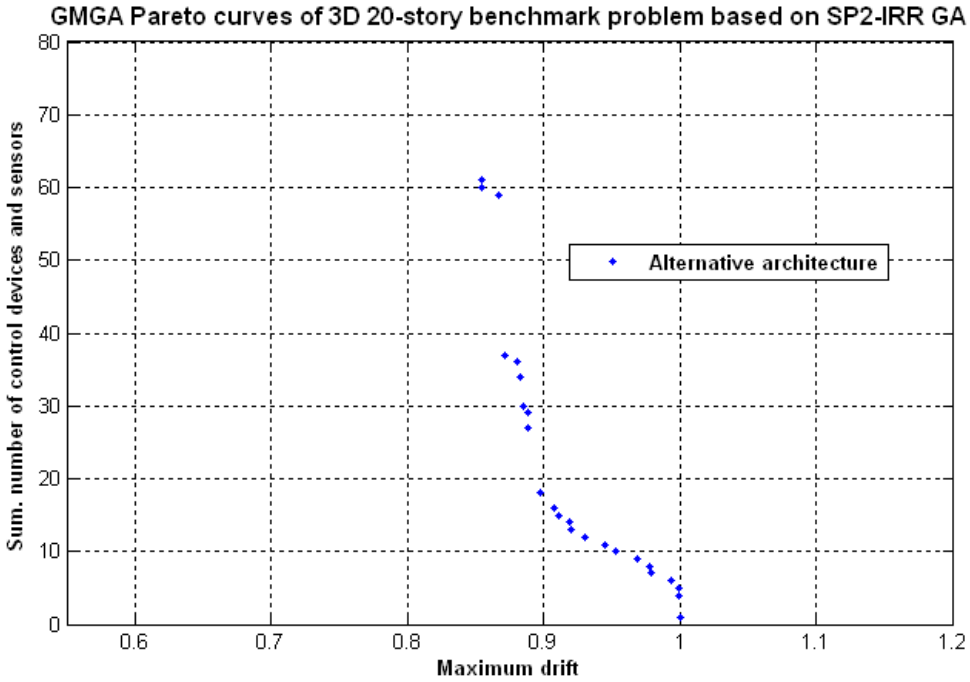

Figure 7.39. The Pareto front obtained using GMGA based on SP2-IRR GA of 3-D 20-story problem with redundancy: 0.5

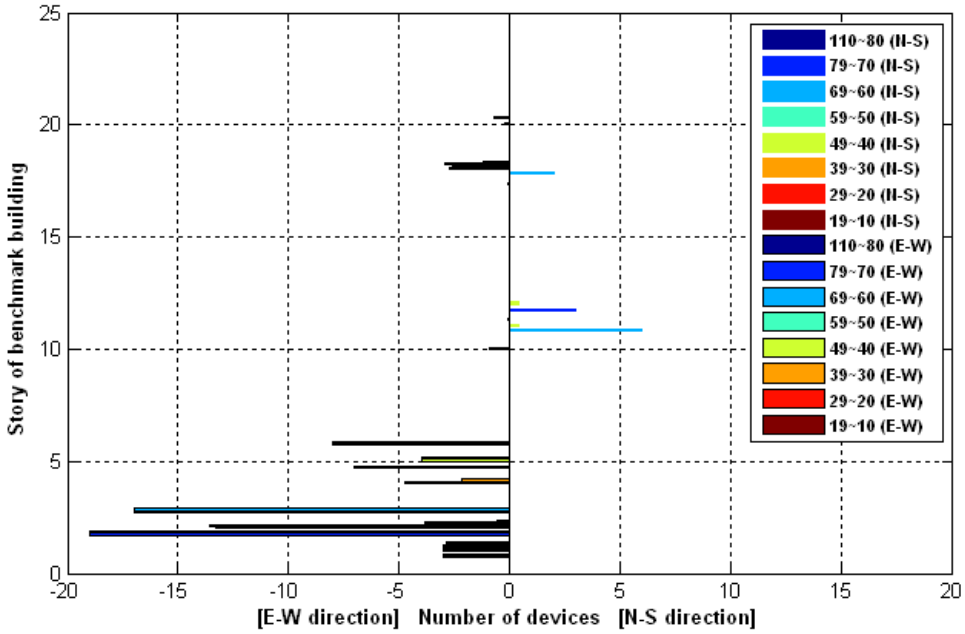

Figure 7.40. The average number of devices at each story using GMGA based on SP2-IRR GA of 3-D 20-story problem with redundancy: 0.5 in all sections

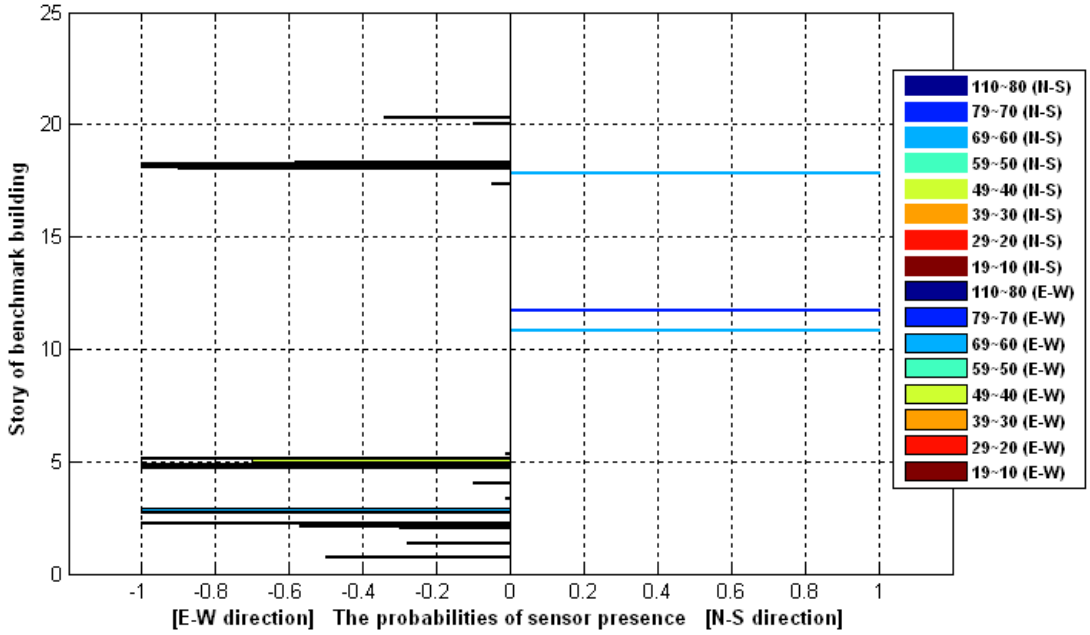

Figure 7.41. The probabilities of sensor presence at each story using GMGA based on SP2-IRR GA of 3-D 20-story problem with redundancy: 0.5 in all sections

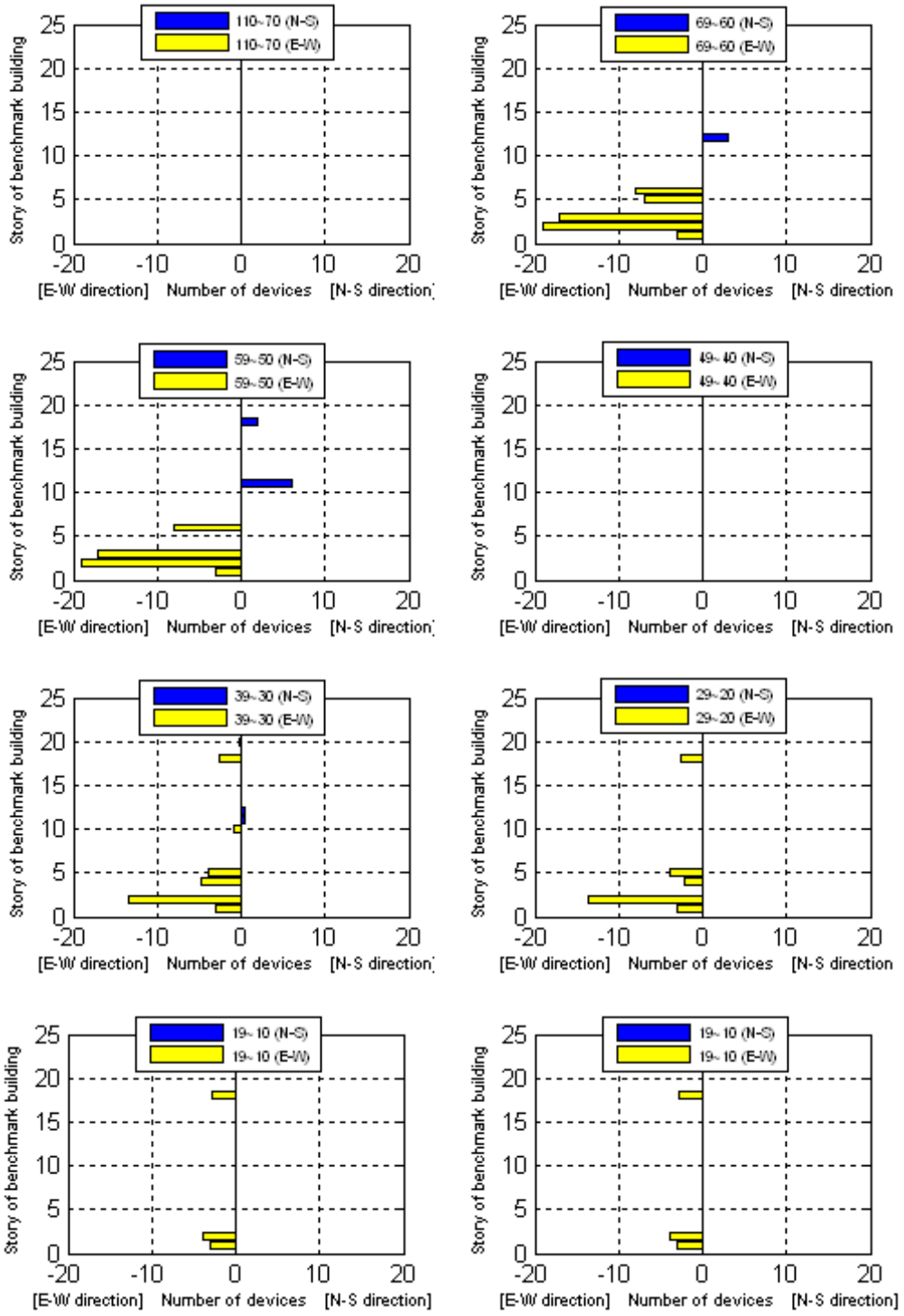

Figure 7.42. The average number of devices at each story using GMGA based on SP2-IRR GA of 3-D 20-story problem with redundancy: 0.5

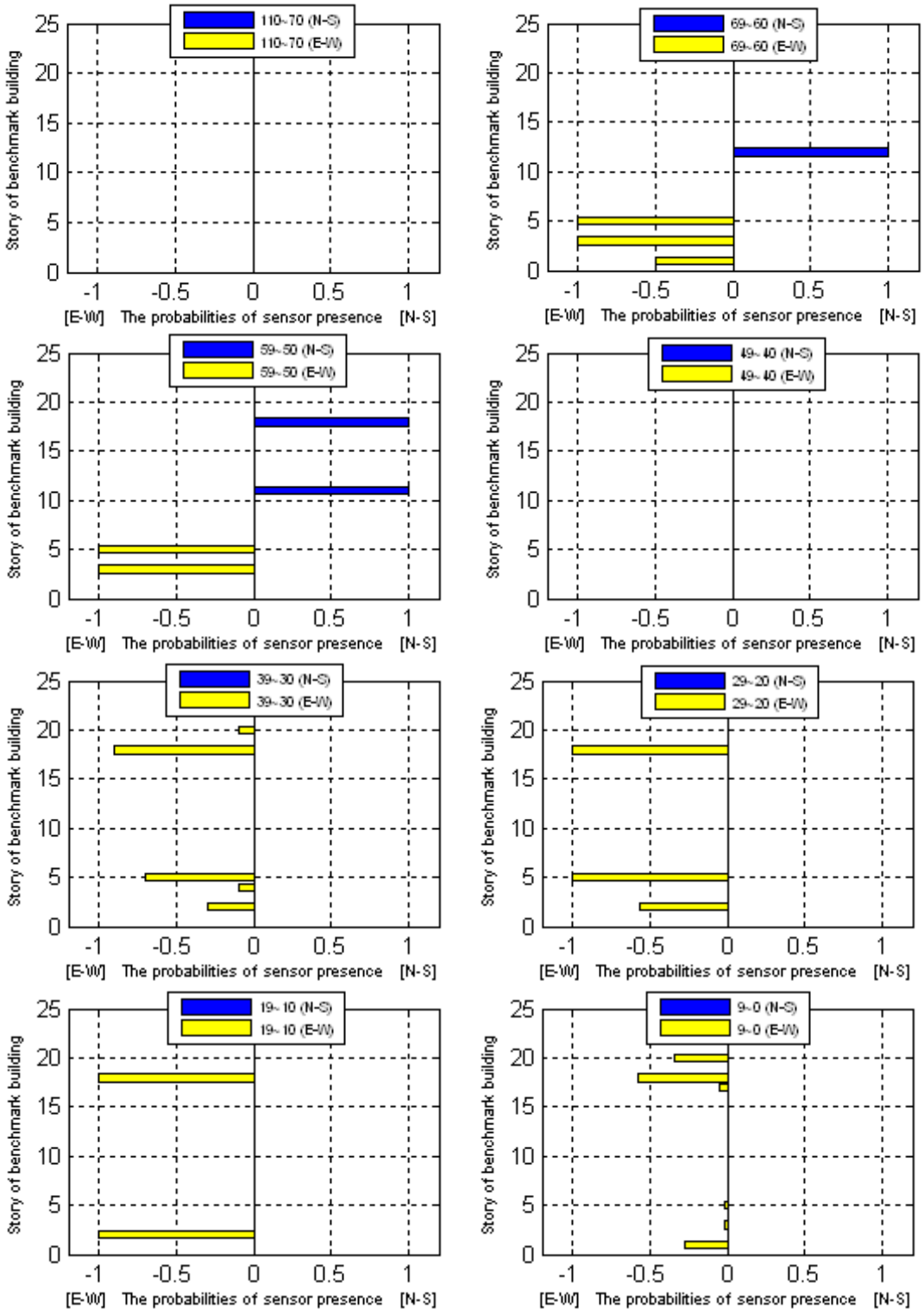

Figure 7.43. The probabilities of sensor at each story using GMGA based on SP2-IRR GA of 3-D 20-story problem with redundancy: 0.5

From the overall comparisons of the Pareto fronts and their redundancy ratio and the number of the iterations to find fully converged Pareto fronts, the redundancy ratio 0.3 is good choice for the 3-D 20-story building near-optimal architecture problem. Although the GMGA based on NSII-IRR GA with redundancy ratio 0.5 is not yet fully converged, the Figure 7.44 shows big difference of the performance. For the GMGA based on the SP2-IRR GA also shows the same trend of performance related to the redundancy ratio as shown in Figure 7.45. The Figure 7.46 shows that the GMGA based on SP2-IRR GA has better performance by comparison the fully converged Pareto fronts with different method. Figure 7.47 shows that the performances of the GMGA and the traditional NSII-IRR GA and SP2-IRR GA have big difference in 3-D 20-story problem and GMGA's big potentiality as optimization method.

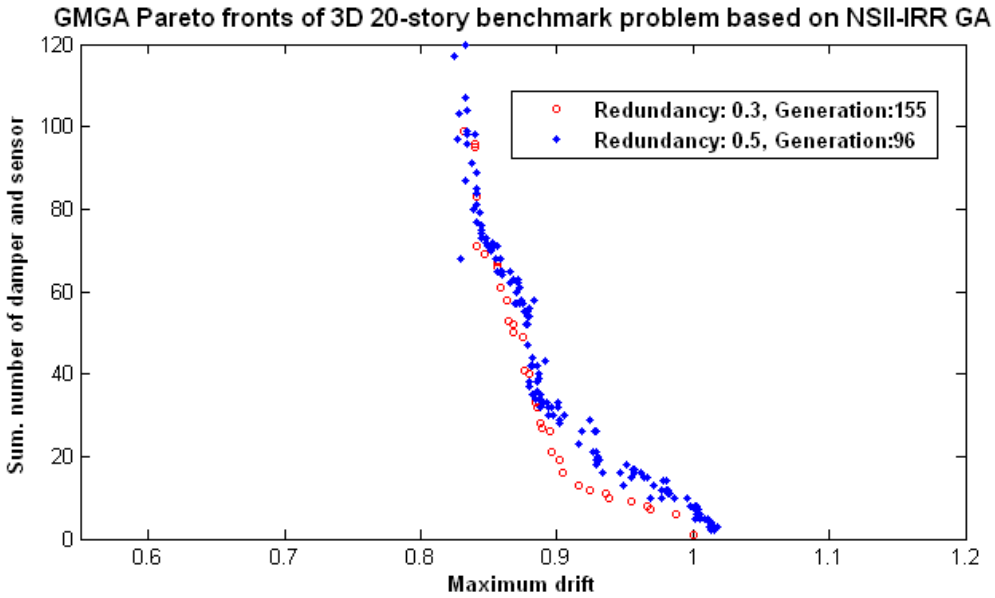

Figure 7.44. The probability comparison of sensor at each story using GMGA based on SPEA2 with redundancy ratio 0.5

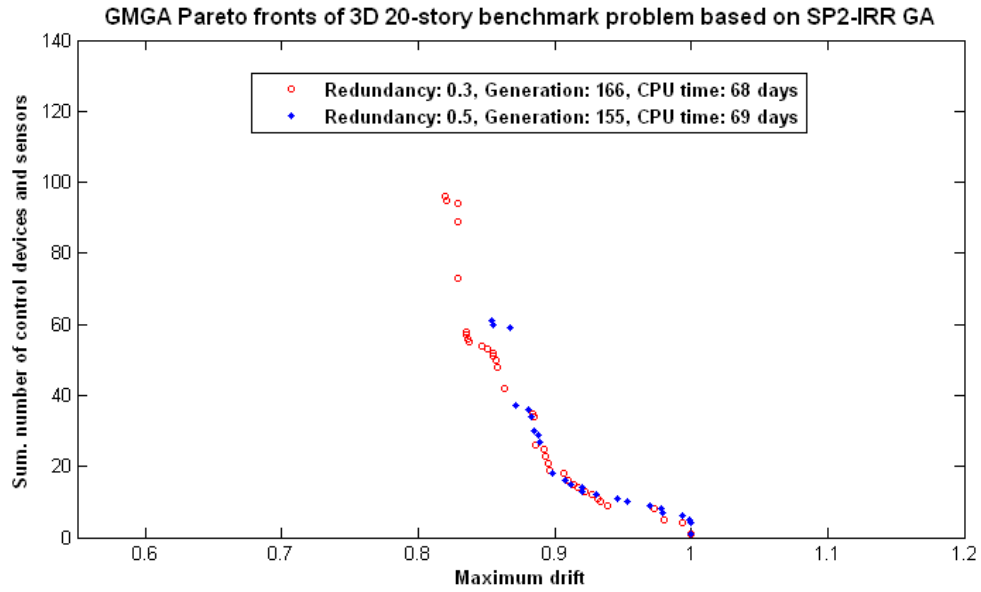

Figure 7.45. The probability comparison of sensor at each story using GMGA based on SPEA2 with redundancy ratio 0.5

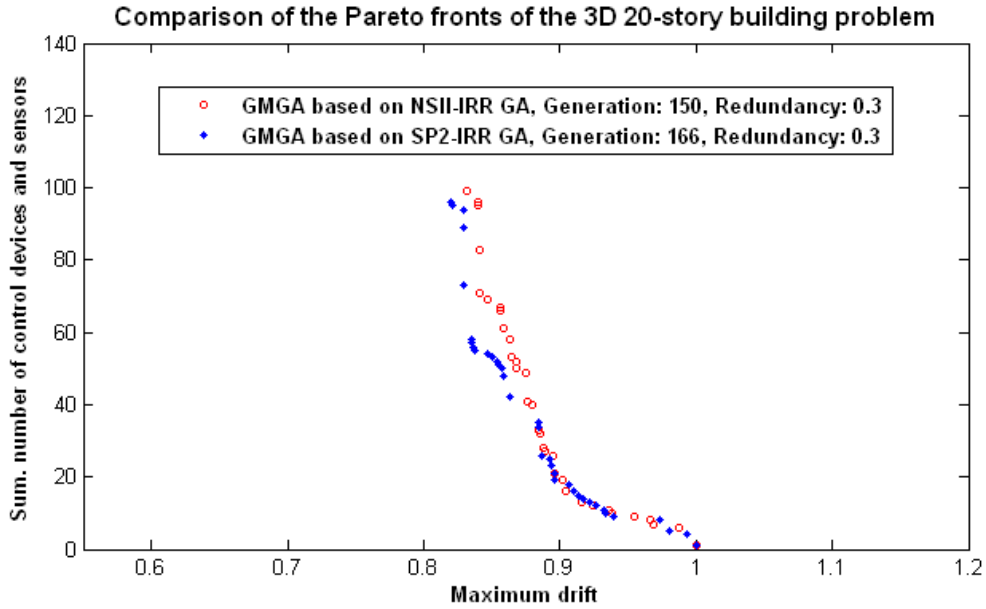

Figure 7.46. The comparison of the Pareto fronts of the 3-D 20-story building problem using GMGA

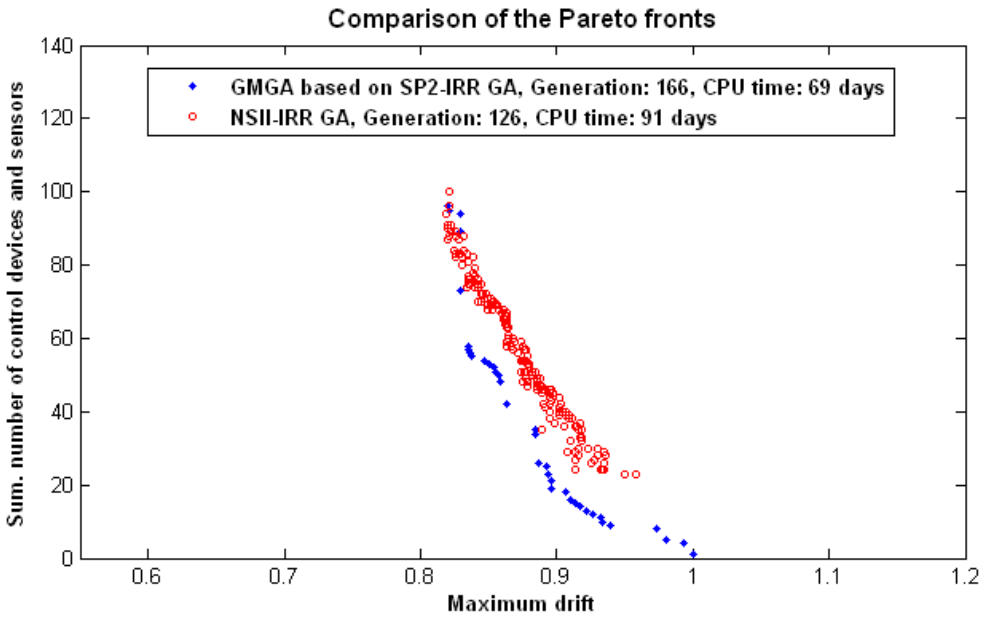

Figure 7.47. The comparison of the Pareto fronts of the 3-D 20-story building problem using GMGA and NSII-IRR GA

8 SUMMARY OF RESEARCH

8.1 Summary

The structural control and optimization knowledge based on genetic algorithm has been developed significantly during the last two decades. In the structural control area, the American Society of Civil Engineers (ASCE) control division suggests the control benchmark buildings to test diverse control strategies developed by civil engineers. To reduce structural damage and protect human property and lives from strong earthquakes, approximately 3 main ways of structural control have been developed: passive control with energy dissipation device, active control and semi-active control. The passive control strategies are openly applied to mild strength or rare probability of earthquake regions. The active or semi-active control strategies will be applied to areas which have the possibility of occurrence of strong earthquakes or structures needing big control efforts that passive control devices cannot support.

Although good control strategies have been developed by structural or control engineers, the installation of devices are another problem needing to be considered carefully because of the random direction, strength and unexpected time of the occurrence of earthquakes. Especially in an active control strategy, the Liner Quadratic Gaussian (LQG) control needs to define the exact locations and numbers of active dampers and sensors. Thus the objective of this research is to find near near-optimal

Pareto fronts and offer practical architecture of the control devices and sensors to civil structural designers.

To perform the objective of this study, various advanced optimization methods are defined and developed. The structure modeling methods of 2 dimensional (2-D) and 3-D are suggested and the detailed LQG controller design process is also expressed. The performances of the simple genetic algorithms (SGA) and implicit redundant representation (IRR) GA about the control optimization problem are compared to choose the better GA engine. NS-IRR GA was defined based on the traditional and well-known multi-objective GA, non-dominated sorting (NS) GA with IRR GA. NSII-IRR GA was defined based on the NSGA-II which is a more advanced multi-objective GA and IRR GA as the encoding policy. SP2-IRR GA was defined based on the SPEA2 and IRR GA. Furthermore, to overcome the NSII- and SP2-IRR GA's disadvantages, the new gene manipulation genetic algorithm (GMGA) was developed. With these diverse methods, the 2-D 3- and 20-story ASCE control benchmark building were studied in full to find the best Pareto front in each structure about 4 major earthquakes. These 4 well-known earthquakes were simultaneously analyzed to find generalized near-optimal architectures of installation of control devices and sensors.

8.2 Results

The 2-D and 3-D modeling results of the ASCE control benchmark buildings are reasonable with the comparison to other researcher's results such as mode shapes and frequencies. The results of the comparison to the performances of the SGA and IRR GA

with single objective were offered. The IRR GA with dynamic search ability showed better performance in finding best locations of the active control devices and sensors with total minimum number of control devices and sensors. Thus, the proposed and developed multi-objective optimization methods adopt the IRR GA as the encoding policy. The first investigation with NS-IRR GA showed the reasonable results, but the central processing unit (CPU) running time was too large and some of the non-dominated individuals were lost during the generations.

The advanced multi-objective optimization methods, such as NSII- and SP2-IRR GA showed good performances in the 2-D 3- and 20-story problems, but the performance in the 3-D problem was not good due to a large CPU running time. The two advanced GA methods did not get out of the near unstable control area in the 3-D problem with small iterations. The NSII-IRR GA's performance is better than the SP2-IRR GA in finding Pareto fronts in the 2-D 3-story building problem, but in case of the 2-D and 3-D 20-story building problem the SP2-IRR GA's performance overwhelmed the NSII-IRR GA. Although the NSII-IRR GA can find and keep diverse near-optimal Pareto front individuals, the final Pareto fronts of SP2-IRR GA are better than the Pareto fronts of NSII-IRR GA with small iterations in 2-D and 3-D 20-story problem. Its redundancy ratio of 0.3 is better than the 0.5 of the two proposed GAs.

To enhance the performance of the NSII- and SP2-IRR GA, the new genetic algorithm was proposed. The gene manipulation genetic algorithm (GMGA) adopted the genetic manipulation concept to generate some defined portion of child population with gene manipulation ratio (GMR). The new individuals were created from the non-

dominated individuals in the multi-objective optimization problem or superior individuals in the single objective optimization problem. With the systemical and engineering judgmental concept, the created gene instances were inserted to the non-dominated individuals. These new individuals did not use the generic genetic operators such as crossover and mutation. This GMGA was carried out to offer the near near-optimal Pareto fronts of the architecture of the installation of the active control devices and sensors to civil structural engineers. The suggested Pareto fronts are better than the NS-, NSII-, and SP2-IRR GA in optimization and CPU running time consumption. With the new gene instance creation mechanism, the near stability and controllability individuals are well removed from the population.

8.3 Conclusion

The suggested diverse architecture of the installation of active control devices and sensors is well matched to LQG in the ASCE control benchmark building. From this research, I verified that there are many kinds of available installation architectures. The selection methods of the advanced multi-objective GAs such as NSGA-II and SPEA2 have competitive abilities in finding and keeping non-dominated individuals. Furthermore, the IRR GA as the encoding policy is also a good mechanism. Finally, the newly developed GMGA has strong ability to find the non-dominated individuals compared to traditional NSGA-II or SPEA2.

From the survey of the control device locations and the numbers and sensor locations, there are some near regularities, but it is hard to apply to find an another

architecture based on the suggested architecture by the developed GAs. Thus, these several proposed GAs to find the non-dominated front will be used for practical control design and active damper and sensor installation because the near-optimal architecture is highly dependent on the structural types.

8.4 Future Works

With successful performances in the ASCE control benchmark building problems, the several proposed advanced multi-objective optimization methods will also be applicable to the benchmark bridge problems. The control optimizations in the installments of the control devices such as lead rubber bearing (LRB) and hybrid control strategies with active control devices can be good studies. And with health monitoring and damage detection areas also needing the optimization method, these areas also can be good experimental study areas with GMGA.

There are many possibilities for developing the GMGA in the creation rule. More mathematical or probabilistic concepts are applicable to the creation rule of the GMGA. Finally, the application of the neural network to GMGA is another good way to develop a high performance GA.

REFERENCES

Arbel, A. (1981). "Controllability measures and actuator placement in oscillatory systems." *International Journal of Control*, 33(3), 565-574.

Abdullah, M. M., Richardson, A., and Hanif, J. (2001). "Placement of sensors/actuators on civil structures using genetic algorithms." *Earthquake Engineering and Structural Dynamics*, 30(8), 1167-1184.

Ahlawat, A. S., and Ramaswamy, A. (2002). "Multi-objective optimal design of FLC driven hybrid mass damper for seismically excited structures." *Earthquake Engineering and Structural Dynamics*, 31, 1459-1479.

Balas, G. J. (1998). "Synthesis of controllers for the active mass driver system in the presence of uncertainty." *Earthquake Engineering and Structural Dynamics*, 27(11), 1189-1202.

Barroso, L. R. (1999). *Performance evaluation of vibration controlled steel structures under seismic loading.* Ph.D. dissertation, Department of Civil and Environmental Engineering, Stanford University, Stanford, CA.

Chen, G. S., Rruno, R. J., and Salama, M. (1991). "Optimal placement of active/passive members in truss structures using simulated annealing." *AIAA Journal*, 29(8), 1327-1334.

Cheng, F. Y. and Li, D. (1997). "Multiobjective optimization design with Pareto genetic algorithm." *Journal of Structural Engineering,* 123(9), 1252-1261.

Chopra, A. K. (2000). *Dynamics of structures theory and applications to earthquake engineering*, Prentice Hall, Upper Saddle River, NJ.

Chung, L. L., Lin, C. C., and Chu, S. Y. (1992). "Optimal direct output feedback of structural control." *Journal of Engineering Mechanics*, 119(11), 2157-2173.

Coello, C. A., Veldhuizen, D. A., and Lamont, G. B. (2001). *Evolutionary algorithms for solving multi-objective problem*, Kluwer Academic Publisher, New York.

Corne, D. W., Knowles, J. D., and oates, M. J. (2000). "The Pareto envelope-based selection algorithm for multiobjective optimization." Schoenauer, M., Deb, K., Rudolph, G., Yao, X., Lutton, E., Merelo, J. J., and Schwefel, H.-P., Ed, *Proc. of the Parallel Problem Solving from Nature VI Conference*, 839-848, Paris France. Springer. Lecture Notes in Computer Science No. 1917.

Crossley, W. A., Cook, A. M., Fanjoy, D. W. (1999). "Using two branch tournament genetic algorithm for multiobjective design." *AIAA Journal*, 37(2), 261–267.

Cunha, A. G., Oliviera, P., and Covas, J. (1997). "Use of genetic algorithms in multicriteria optimization to solve industrial problems." *Proc. 7th Int. Conf. Genetic Algorithms*, T. B¨ack, Ed. San Francisco, CA: Morgan Kaufmann, 682–688.

Dasgupta, D. (1994). "Handling deceptive problems using the structured genetic algorithm." *Proc. of the First IEEE Conference on Evolutionary Computation: IEEE World Congress on Computational Intelligence*, IEEE Service Center, Piscataway, NJ, 1, 807-813.

De Jong, A. K. (1975). *An analysis of the behavior of a class of genetic adaptive systems.* PhD Dissertation, University of Michigan.

Deb, K., Agrawal, S., Pratab, A., and Meyarivan, T. (2000). "*A fast elitist non-dominated sorting genetic algorithm for multi-obejective optimization: NSGA-II*, KanGAL Report 200001, Indian Institute of Technology, Kanpur, India.

DeLorenzo, M. L. (1990). "Sensor and actuator selection for large space structure control." *Journal of Guidance, Control, and Dynamics*, Vol. 13, No. 2, 249-257.

Dhingra, A. K and Lee, B. H. (1995). "Multiobjective design of actively controlled structures using a hybrid optimization method." *International Journal for Numerical Methods in Engineering*, 38(20), 3383-3401.

Dyke, S. J. (1996). *Acceleration feedback control strategies for active and semiactive control systems: modeling, algorithm development, and experimental verification.* PhD Dissertation. University of Notre Dame, Notre Dame, IN.

Erickson, M., Mayer, A., and Horn, J. (2001). "The niched Pareto genetic algorithm 2 applied to the design of groundwater remediation systems." Zitzler, E., Deb, K., Thiele, L., Coello Coello, C. A., and Corne, D., Ed. *First International Conference on Evolutionary Multi-Criterion Optimization*, Zurich, Switzerland, 681-695. Springer-Verlag. Lecture Notes in Computer Science No. 1993.

Eshelman, L. J., Caruana, R. A., and Schaffer, J. D. (1989). "Bases in the crossover landscape." *Proc. 3rd Int. Conf. on Genetic Algorithms, J. Scahffer(Ed.)*, Morgan Kaufmann Publishers, San Francisco, CA, 10-19.

Fonseca, C. M., and Fleming, P. J. (1993). "Genetic algorithms for multiobjective optimization formulation, discussion and generalization." *Proc. 5th Int. Conf. on*

Genetic Algorithms, S. Forrest (Ed.), Morgan Kaufmann Publishers, San Mateo, CA, 416-423.

Fonseca, C. M., and Fleming, P. J. (1995). "An overview of evolutionary algorithms in multiobjective optimization.", *Evolutionary Computation*, 3(1):1-16, Spring

Franklin, G. F., Powell, J. D, and Abbas, E. N. (2002). *Feedback control of dynamic systems*, Prentice Hall, Upper Saddle River, NJ.

Furuya, H., and Haftka, R. T. (1995). "Placing actuators on space structures by genetic algorithms and effectiveness indices." *Structural Optimization*, 9(2), 69-75.

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*, Addison-Wesley, Reading, MA.

Goldberg, D. E. and Richardson, J. (1987). "Genetic algorithm with sharing for multimodal function optimization." *Genetic algorithms and their applications:* Grefenstette, J. J., Ed. *Proc. of the Second International Conference on Genetic Algorithms*, pages 41-49, Hillsdale, NJ. Lawrence Erlbaum.

Gupta, A., and Krawinkler, H. (1999). *Seismic demands for performance evaluation of steel moment resisting frame structures (SAC Task 5.4.3*), Report. No. 132, The John A. Blume Earthquake Engineering Center, Stanford, CA.

Hajela, P. and Lin, C. Y. (1992). "Genetic search strategies in multicriterion optimal design." *Structural Optimization*, 4, 99-107.

Hans, A. E. (1988). "Multicriteria optimization for highly accurate systems.", *Multicirteria optimization in engineering and sciences,* W. Stadler(ed.), Mathematical concepts and methods in science and engineering, 19, 309-352. New York: Plenum press.

Hibbeler, R. C. (1999). *Structural analysis*, Prentice-Hall International, INC, Upper Saddle River, NJ.

Holland, J. (1975). *Adaptation in natural and artificial systems*, The University of Michigan Press, Ann Arbor, MI.

Horn, J. and Nafpliotis, N. (1993). *Multiobjective optimization using the niched Pareto genetic algorithm*, IlliGAL Report 93005, Illinois Genetic Algorithms Lab., Univ. Illinois, Urbana-Champaign.

Horn, J., Nafpliotis, N., and Goldberg, D. E. (1994), "A niched Pareto genetic algorithm for multiobjective optimization." *Proc. 1st IEEE Conf. Evolutionary*

Computation, IEEE World Congr. Computational Computation, Piscataway, NJ, June 27–29, 1, 82–87.

Jones, G., Brown, R. D, Clark, D. E., Willett, P., and Glen, R. C. (1993) "Searching databases of two-dimmensional and three-dimmensional chemical structures using genetic algorithms." *Proc. of the Fifth International Conference on Genetic Algorithms*, Morgan Kaufmann Publishers, San Francisco, CA, 597-602.

Knowles, J. D. and Corne, D. W. (2000). "Approximating the nondominated front ssing the Pareto archived evolution strategy." *Evolutionary Computation*, 8(2), 149-172.

Kursawe, F. (1991). A variant of evolution strategies for vector optimization. Parallel Problem solving from Nature. 1st Workshop, PPSN I, Dortmund, Germany, Springer-Verlag. Lecture Notes in Computer Science No.496.

Li, Q. S., Liu, D. K., Fang, J. Q., and Tam, C. M. (2000). "Multi-level optimal design of buildings with active control under winds using genetic algorithms.", *Journal of Wind Engineering and Industrial Aerodynamics*, 86(1), 65-86.

Li, Q. S., Liu, D. K., Tang, J., Zhang, N., and Tam, C. M. (2004). "Combinatorial optimal design of number and positions of actuators in actively controlled structures using genetic algorithms." *Journal of Sound and Vibration*, 270(4-5), 611-624.

Lu, J., and Skelton, R.E. (1998). "Covariance control using closed loop modeling for structures." *Earthquake Engineering of Structures and Dynamics*, 27(11), 1367-1383.

MATLAB/Simulink (2007), *The mathworks*, Natick, MA.

Moita, J. M., Correia, V. M., Martins, P. G, Soares, C. M., and Soares, C. A. (2006). "Optimal design in vibration control of adaptive structures using a simulated annealing algorithm." *Thirteenth International Conference on Composite Structures - ICCS/13,* Melbourne, Australia, 75(1-4), 79-87.

Morse, J. (1980). "Reducing the size of the nondominated set: Pruning by clustering." *Computers and Operations Research*, 7(1-2), 55-66.

Ohtori, Y., Christenson, R. E., Spencer Jr., B. F., and Dyke, S. J. (2004). "Benchmark control problems for seismically excited nonlinear buildings." *Journal of Engineering Mechanics*, 130(4), 366-385.

Olariu, I. (1995). "Passive control and base isolation: State-of-the-art lecture." *10th European Conference Earthquake Engineering,* Balkema, Rotterdam, 703-713.

Oliver, I. M., Smith, D. J. and Holland, J. H. (1987) "A study of permutation crossover operators on the traveling salesman problem." *Proc. 2nd Int. Conf. on Genetic Algorithms and Their Applications*, Cambridge, MA, 224-230.

Onoda, J. and Hanawa, Y. (1992). "Actuator placement optimization by genetic and improved simulated annealing algorithms." *AIAA Journal*, 31(6), 1167-1169.

Quagliarella, D. and Vicini, A. (1997). *Coupling genetic algorithms and gradient based optimization techniques. Genetic Algorithms and Evolution Strengies in Engineering and Computer Science. Recent Advances and Industrial Applications*, John Wiley and Sons, West Sussex, England.

Raich, A. M. (1999). "An evolutionary based methodology for representing and evolving structural design solutions." Ph.D. Dissertation, Department of Civil and Environmental Engineering, University of Illinois, Urbana-Champaign, IL.

Raich, A. M. and Ghaboussi, J. (1998). "Evolving structural design solution using an implicit redundant genetic algorithm." *Structural Multidisciplinary Optimization*, 20, 222-231.

Rao, S. S., and Pan, T., and Venkayya, V. B. (1991). "Optimal placement of actuators in actively controlled structures using genetic algorithms." *AIAA Journal*, 29(6), 942-943.

Richardson, A. and Abdullah, M. M. (2002). "Sensor/actuators placement on civil structures using a real coded genetic algorithm." *Proc. of the SPIE-The International Society for Optimal Engineering*, San Diego, CA, 4696, 244-255.

Richardson, J. T., Palmer, M. R., Liepins, G., and Hilliard, M. (1989). "Some guidelines for genetic algorithms with penalty functions.", *Proc. of the Third International conference on Genetic Algorithms*, J.D. Schaffer (Eds.), San Mateo, CA: Morgan Kaufmann Publishers, 91-197.

Rutenbar, R. A. (1989). "Simulated annealing algorithms: An overview." *IEEE Circuits and Devices Magazine,* 5(1), 19-26.

Schaffer, J. D. (1985). "Multiple objective optimization with vector evaluated genetic algorithms." *Proc. of the First International Conference on Genetic Algorithms,* Lawrence Erlbaum Associates, Hillsdale, NJ, 93-100.

Simulink-4.0 User's Guide. The mathworks Inc., Natick, MA, USA, 2000.

Silverman, B. W. (1986). *Desity estimation for statistics and data analysis*. Chapman and Hall, London.

Soong, T. T. and Spencer, B. F. Jr. (2000). "Active, semi-active and hybrid control of structures." *Bulletin of the New Zealand National Society for Earthquake Engineering,* 33(3), 387 -402.

Soong, T. T. and Spencer, B. F. Jr. (2002). "Supplemental energy dissipation: State-of-the-art and state-of-the practice." *Engineering Structures,* 24, 243 -259.

Spencer, Jr., B. F., Christenson, R. E., and Dyke, S. J. (1999). "Next generation benchmark control problem for seismically excited buildings." *Proc., 2nd World Conf. on Structural Control,* T. Kobori et al., eds., Wiley, New York, 2, 1135-1360.

Spencer, Jr., B. F., Dyke, S. J. and Deoskar, H. S. (1997). "Benchmark Problems in Structural Control – Part I: Active Mass Driver System." *Proc. of the ASCE Structures Congress XV*, 2, 1265–1269.

Srinivas, N. and Deb, K. (1994), "Multiobjective optimization using nondominated sorting in genetic algorithms." *Journal of Evolutionary computation*, 2(3), 221-248.

Stefani, R. T., Shahian, B, Savant Jr, C. J., Hostetter, G. H. (2002). *Design of feedback control systems*, Oxford University Press, New York.

Symans, M. D. and Constantinou, M. C. (1999). "Semi-active control systems for seismic protection of structure: A state-of-the-art review." *Engineering Structures*, 21, 469-487.

Syswerda. G. (1989). "Uniform crossover in genetic algorithms." *Proc. 3rd Int. Conf. on Genetic Algorithms*, J. Schaffer(Ed.), Morgan Kaufmann Publishers. San Francisco, CA, 2-9.

Syswerda, G. and Palmucci, J. (1991). "The application of genetic algorithms to resource scheduling." *Proc. of the Fourth International conference on Genetic Algorithms: University of California, San Diego*. 502-508.

Tewari, A. (2002). *Modern control design with matlab and simulink*, John Wiley & Sons, LTD, New York.

Van Veldhuizen, D. A. and Lamont, G. B. (2000). "Multiobjective evolutionary algorithms: Analyzing the state-of-the-art." *Evolutionary Computation*, 7(3), 1-26.

Wu, B., Ou, J. P., and Soong, T. T. (1997). "Optimal placement of energy dissipation devices for three-dimensional structures." *Engineering Structures*, 19(2), 113-125.

Wu, S. J. and Lindsay, R. K. (1996). "A comparison of the fixed and floating building block representation in the genetic algorithm." *Evolutionary Computation,* 4(2).

www.wikipedia.org, "Evolutionary algorithm", March 01, 2008.

Yang, J. N., and Samali, B. (1983). "Control of tall buildings in along-wind motion." *Journal of Structural Engineering ASCE*, 109(1), 50-68.

Yao, James T. P. (1972). "Concept of structural control." *ASCE Journal of Structural Division*, 98(ST7), 1567-1574.

Zitzler, E., Deb, K., and Thiele, L. (1999). "Comparison of multiobjective evolutionary algorithms: Empirical results." *Evolutionary Computation*, 8(2), 173-195.

Zitzler, E., Laumanns, M., and Thiels, L. (2001). *SPEA2: Improving the strength Pareto evolutionary algorithm*, Technical Report 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Gloriastrasse 35, CH-8092 Zurich, Switzerland.

Zitzler, E., Laumanns, M., and Thiele, L., (2002a). *SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjecitve optimization*. Technical Report 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Gloriastrasse 35, CH-8092 Zurich, Switzerland.

Zitzler, E., Laumanns, M., and Thiele, L., (2002b). "SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjecitve optimization." *Evolutionary methods for design, optimization and control*. Ed.s. Giannakoglou, K., Tsahalis, D., Periaux, J., Papailiou, K., and Fogarty, T., Barcelona, Spain.

Zitzler, E. and Thiele, L. (1999). "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach." *IEEE Transactions on Evolutionary Computation,* 3(4), 257-271.

Zydallis, J. B., Van Veldhuizen, D. A., and Lamont, G. B. (2001). "A statistical comparison of multiobjective evolutionary algorithms including the MOMGA-

II." Zitzler, E., Deb, K., Thiele, L., Coello, C. A. C., and Corne, D., Ed.s. *First International Conference on Evolutionary Multi-Criterion Optimization*, Zurich, Switzerland, 226-240. Springer-Verlag-Verlag. Lecture Notes in Computer Science No 1993.

APPENDIX

Data for Figure 7.10

Sum. Number of Dampers and sensors	73	71	70	69	67	66	66	63	60	59	59	58	58	57	57
Maximum Drifts	0.6362	0.6374	0.6381	0.6383	0.6405	0.6411	0.6411	0.6413	0.6418	0.6444	0.6444	0.6471	0.6471	0.6489	0.6489
1st story	10	9	9	9	7	7	7	7	7	7	7	7	7	7	7
2nd	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
3rd	9	9	9	9	9	9	9	8	8	9	9	9	9	9	9
4th	4	4	4	4	4	4	4	4	4	2	2	1	1	0	0
5th	7	7	7	7	6	6	6	7	7	7	7	7	7	7	7
6th	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
7th	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8th	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
9th	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10th	4	3	3	3	3	3	3	0	0	0	0	0	0	0	0
11th	1	2	1	0	0	0	0	0	0	0	0	0	0	0	0
12th	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13th	8	7	7	7	7	7	7	7	8	8	8	8	8	8	8
14th	4	4	4	4	4	4	4	4	0	0	0	0	0	0	0
15th	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
16th	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17th	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18th	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19th	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20th	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1st story	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2nd	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3rd	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4th	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5th	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6th	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
7th	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8th	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9th	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10th	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
11th	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
12th	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13th	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
14th	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15th	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
16th	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17th	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18th	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19th	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20th	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
56	56	55	55	54	54	52	52	51	51	50	49	48	47	46	45	
0.6554	0.6554	0.6588	0.6588	0.661	0.661	0.6668	0.6668	0.6702	0.6702	0.6733	0.6742	0.675	0.6783	0.6795	0.6799	
7	7	7	7	6	6	6	6	5	5	3	6	6	6	6	6	
6	6	6	6	6	6	6	6	6	6	6	4	5	5	5	5	
9	9	9	9	6	6	6	6	8	8	9	6	6	6	6	6	
0	0	0	0	6	6	6	6	0	0	0	5	4	4	4	4	
6	6	6	6	6	6	6	6	6	6	6	7	6	6	6	6	
4	4	4	4	2	2	0	0	4	4	4	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	5	5	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	1	1	0	0	0	4	0	0	
0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	
8	8	7	7	0	0	0	0	0	0	7	0	0	0	0	0	
0	0	0	0	5	5	5	5	0	0	0	5	0	0	1	0	
6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	
0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	2	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	
1	1	1	1	0	0	0	0	1	1	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	1	1	1	1	1	1	0	1	1	1	1	1	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
1	1	1	1	1	1	1	1	0	0	1	0	0	0	1	0	
0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	
1	1	1	1	0	0	0	0	0	0	1	0	0	0	0	0	
0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	
1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
43	42	41	41	40	40	39	39	38	38	37	37	36	36	35	35
0.6816	0.682	0.6882	0.6882	0.691	0.691	0.6985	0.6985	0.7025	0.7025	0.7042	0.7042	0.7132	0.7132	0.7176	0.7176
6	6	6	6	6	6	6	6	6	6	6	6	6	6	0	0
5	5	4	4	4	4	4	4	4	4	4	4	4	4	5	5
6	6	6	6	6	6	6	6	6	6	6	6	6	6	7	7
4	4	4	4	4	4	2	2	0	0	0	0	0	0	0	0
6	6	6	6	6	6	6	6	6	6	6	6	6	6	4	4
0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	6	6	6	6	6	6	6	6	6	6	6	6	6	3	3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	5
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	6	6	6	6	6	6	6	6	6	6	6	6	6	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	4
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	0	0	0	0	0	0	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	0	0	1	1	1	1	1	1	1	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

34	33	32	31	31	29	29	28	28	27	27	26	26	25	25	24
0.7216	0.7216	0.7247	0.7304	0.7304	0.7322	0.7322	0.7383	0.7383	0.74	0.74	0.7471	0.7471	0.7561	0.7561	0.7642
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	5	4	4	4	4	4	4	4	4	4	4	4	4	4	4
7	7	7	7	7	7	7	7	7	7	7	7	7	6	6	7
0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	0
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
3	3	5	3	3	5	5	4	4	4	4	3	3	3	3	4
4	5	0	4	4	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	1	1	1	1	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

24	23	23	22	22	21	21	20	20	19	19	18	18	17	17	16	16	15
0.7642	0.7657	0.7657	0.7764	0.7764	0.7851	0.7851	0.7993	0.7993	0.8036	0.8036	0.8062	0.8062	0.8162	0.8162	0.8248	0.8248	0.8342
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	4	4	4	4	4	4	6	6	2	2	6	6	5	5	5	5	4
7	7	7	7	7	6	6	2	2	6	6	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	0	0	3	3	3	3	3	3	3
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	4	4	3	3	3	3	4	4	3	3	4	4	4	4	4	4	3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	4	4	4	4	4	4	0	0	4	4	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	1	0	0	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	0	0	1	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	1	0	0	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	14	13	13	12	12	11	11	10	10	9	9	8	8	7	7	6	6
0.8342	0.8458	0.8458	0.8519	0.8519	0.8614	0.8614	0.8711	0.8711	0.8839	0.8839	0.8962	0.8962	0.9146	0.9146	0.9292	0.9292	0.9449	0.9449
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	4	4	4	4	4	4	2	2	0	0	0	0	2	2	2	2	2	2
0	0	0	0	0	0	0	0	0	2	2	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	0	0	0	0
3	3	3	3	3	3	3	3	3	4	4	3	3	0	0	0	0	0	0
3	3	3	0	0	0	0	0	0	0	0	0	0	3	3	3	3	3	3
0	0	0	0	0	2	2	0	0	0	0	0	0	0	0	0	0	0	0
3	3	3	4	4	0	0	0	0	0	0	0	0	0	0	1	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	2	2	2	2	0	0	2	2	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	3	3	3	3	3	3	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VITA

Young Jin Cha received his Bachelor of Science degree in civil engineering from The Kumoh National Institute of Technology in Gumi, South Korea in 2002. He entered the civil engineering program at Yonsei University in March 2002 and received his Master of Science degree in February 2004. His research interests include special structural dynamics and seismic behavior of bridges. He enrolled in the Doctor of Philosophy program in civil engineering at Texas A&M University, where he worked as a research assistant. He worked with Dr. Raich and Dr. Barroso, and his research focused on heuristic optimization for 3-D structural control architectures of active control devices and sensors. He graduated in December 2008 with a Ph.D. degree in civil engineering from Texas A&M University (College Station, Texas). He may be contacted through the following addresses:

Name: Young Jin Cha

Address: 491 Sindam Gumgok-myun,
JinJu, South Korea 663-830

Email Address: youngjin.cha@gmail.com

Education: B.S., Civil Engineering, Kumoh National Institute of Technology, 2002
M.S., Civil Engineering, Yonsei University, 2004